

2014

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE ESTADÍSTICA



TESIS DE GRADO

**CLASIFICACIÓN MEDIANTE REDES NEURONALES Y
CONGLOMERADOS NO JERÁRQUICOS DE LAS CONDICIONES DE
VIDA DE LOS HOGARES DE BOLIVIA**

Postulante: Univ. Luis Fernando Flores Rivero
Tutor: M.Sc. Fernando Oday Rivero Suguira

La Paz - Bolivia

A Dios, por regalarme la vida y acompañarme cada día.

A mis papás, por ser mi guía, mi fortaleza y mi ejemplo de vida.

A mi esposa, por todo su apoyo y comprensión en estas largas jornadas de trabajo.

A mis bebés: Diego, Fernanda y Franco por ser la alegría y la razón de mi vida.

AGRADECIMIENTOS

A Dios, por darme la oportunidad de existir así, aquí y ahora; por mi vida y por acordarse siempre de mí. Gracias por iluminarme y darme fuerzas para seguir adelante.

A ti Papá, por tu incondicional apoyo, tanto al inicio como al final de mi carrera; por estar pendiente de mí a cada momento. Gracias por ser ejemplo de arduo trabajo y lucha constante en la vida, gracias por ser mi maestro y por todo lo que me das cada día.

A ti Mamá, por la inmensidad de tu amor, gracias por ser mi ángel guardián y por tus incansables cuidados. Gracias por no faltarme nunca, por todo tu esfuerzo y dedicación a lo largo de mi vida y mi formación profesional. Gracias por creer en mí y darme fuerza en los momentos de flaqueza.

A ustedes papás que me lo han dado todo, mi más grande admiración y mi eterno agradecimiento.

A ti mi amor, que elegiste sacrificar tu tiempo para que yo pudiera cumplir con el mío. Con tu bondad y sacrificio me inspiraste a ser mejor para ti y nuestros hermosos hijos; sin duda alguna, esta tesis lleva mucho de ti. Gracias por tu paciencia, por todo tu apoyo y por estar siempre a mi lado.

A ti hermanita, porque juntos aprendimos a vivir, crecimos como cómplices día a día y somos amigos incondicionales de toda la vida, gracias por todo tu cariño.

A mi tutor de tesis Ms.C. Fernando Rivero, por su presencia incondicional, sus apreciados y relevantes aportes, críticas, comentarios y sugerencias durante el desarrollo de esta investigación. Su motivación y su gran desprendimiento han sido fundamentales en la elaboración de la presente tesis. Gracias por todo el tiempo dispensado y por superar los límites habituales del tutor convencional.

A mis tribunales de tesis, Ms.C. Nicolás Chávez y Lic. Juan Carlos Flores, por todo el apoyo y los valiosos aportes realizados. Agradecerles también por la confianza, el interés y la motivación prestada durante este largo camino.

A Paúl Arce, por su desprendimiento y su valiosa amistad; por todo el apoyo brindado en el desarrollo e implementación de la red neuronal. Sin su ayuda esto no hubiese sido posible.

A toda mi familia, porque todos sus consejos y su apoyo moral me han incentivado a seguir adelante a lo largo de mi vida, gracias a todos ustedes.

A todo el plantel docente, que a lo largo de mi carrera, me han transmitido sus amplios conocimientos y la riqueza de sus experiencias.

A todos mis amigos, compañeros de oficina y a todas las personas que de una u otra forma estuvieron conmigo aportando con un granito de arena; a todos ustedes mi más grande reconocimiento.

CONTENIDO

1.	INTRODUCCIÓN.....	1
1.1	Objetivo	8
1.2	Problema de investigación	9
1.3	Hipótesis del trabajo	9
2	REDES NEURONALES ARTIFICIALES	10
2.1	Introducción biológica	11
2.2	Origen y transferencia de señales nerviosas	13
2.3	Emulación de los sistemas nerviosos	15
2.3.1	Procesamiento paralelo	15
2.3.2	Memoria distribuida	15
2.3.3	Aprendizaje adaptativo	16
2.4	Modelos neuronales artificiales.....	16
2.5	Modelo general de neurona artificial	20
2.5.1	Conjunto de entradas.....	21
2.5.2	Pesos sinápticos.....	21
2.5.3	Reglas de propagación	21
2.5.4	La función de transferencia.....	22
2.5.5	La función de salida.....	24
2.6	Clasificación de modelos de RNA.....	24
2.6.1	Clasificación según el tipo de arquitectura	25
2.6.2	Clasificación según el tipo de aprendizaje	28
2.7	Red backpropagation.....	37
2.7.1	Estructura y aprendizaje de la red backpropagation	38
2.7.2	Algoritmo de entrenamiento de la red backpropagation	39
2.7.3	Número de capas y cantidad de neuronas de la red backpropagation	42
3	ANÁLISIS POR CONGLOMERADOS	44
3.1	Fundamentos de análisis conglomerado.....	45
3.2	Proceso del análisis de conglomerados.....	46
3.2.1	Fase de preparación.....	47
3.2.2	Similitud entre objetos.....	47
3.2.3	Formación de conglomerados.....	51
3.2.4	Número de conglomerados finales.....	62

4	EJEMPLO DE APLICACIÓN	64
4.1	Descripción del conjunto de datos	66
4.2	Metodología NBI	68
4.2.1	Fijación de niveles normativos mínimos.....	70
4.2.2	Evaluación de las carencias.....	70
4.3	Análisis previo de los datos.....	72
4.3.1	Estadísticos descriptivos.....	73
4.4	Resultados obtenidos con el método NBI	75
4.5	Resultados obtenidos con el método de conglomerados no jerárquicos .	80
4.5.1	Número de conglomerados o grupos	81
4.5.2	Selección de centros iniciales	81
4.5.3	Asignación de objetos a los grupos.....	82
4.5.4	Optimalidad de los grupos	82
4.5.5	Implementación del algoritmo de k-medias	83
4.6	Resultados obtenidos con el método de redes neuronales BPN.....	92
4.6.1	Interfaz de la red BPN.....	93
4.7	Análisis comparativo de resultados.....	100
4.8	Resumen comparativo de metodologías	104
5.	CONCLUSIONES.....	106
6.	RECOMENDACIONES	108
7	ANEXOS	110
7.1	ANEXO 1 - Demostración desigualdad triangular	110
7.2	ANEXO 2 – Metodología NBI.....	112
7.2.1	Fijación de niveles normativos mínimos.....	114
7.2.2	Evaluación de las carencias.....	114
7.2.3	Agregación de carencias	129
7.2.4	Estratos de pobreza.....	129
7.3	ANEXO 3 - Gráficas de dispersión de valores atípicos	131
7.4	ANEXO 4 – Código NBI en SPSS.....	137
7.4.1	Código NBI materiales de la vivienda	137
7.4.2	Código NBI servicios básicos.....	142
7.4.3	Código NBI salud.....	145
7.4.4	Código NBI educación	149
7.4.5	Código NBI hogar	153
7.5	ANEXO 5 – Código red neuronal en C#.....	154

7.5.1	Código formulario BPN	154
7.5.2	Código neurona BPN	167
7.5.3	Código red BPN.....	168
7.6	ANEXO 6 – Código análisis cluster en SPSS	176
7.7	ANEXO 7 – Código errores de muestreo	177
7.8	ANEXO 8 – Estadístico Kappa.....	178
7.9	ANEXO 9 – Código del estadístico Kappa	180
8	BIBLIOGRAFÍA.....	181

ÍNDICE DE TABLAS

Tabla 1.1.	COMPARACIÓN ENTRE MODELOS ESTADÍSTICOS Y RNA	7
Tabla 4.1.	HOGARES Y PERSONAS SEGÚN DEPARTAMENTO	66
Tabla 4.2.	SECCIONES DEL CUESTIONARIO MULTITEMÁTICO	67
Tabla 4.3.	COMPONENTES DEL INDICADOR NBI	68
Tabla 4.4.	LISTADO DE VARIABLES DE COMPONENTES Y VARIABLES	73
Tabla 4.5.	ESTADÍSTICOS DESCRIPTIVOS	74
Tabla 4.6.	CLASIFICACIÓN CON EL MÉTODO NBI.....	76
Tabla 4.7.	POBREZA SEGÚN DEPARTAMENTO - MÉTODO NBI.....	77
Tabla 4.8.	CENTROS INICIALES SEGÚN ESTRATO DE POBREZA	82
Tabla 4.9.	CLASIFICACIÓN CON EL MÉTODO DE CONGLOMERADOS (CENTROS INICIALES).....	83
Tabla 4.10.	COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI	84
Tabla 4.11.	ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI.....	85
Tabla 4.12.	CENTROS INICIALES Y FINALES.....	85
Tabla 4.13.	CLASIFICACIÓN CON EL MÉTODO DE CONGLOMERADOS (CENTROS FINALES)	86
Tabla 4.14.	COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS FINALES) Y ESTRATOS NBI.....	86
Tabla 4.15.	ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS FINALES) Y ESTRATOS NBI	87
Tabla 4.16.	ANÁLISIS DE MEDIAS DE LOS CONGLOMERADOS FINALES	87
Tabla 4.17.	CARACTERÍSTICAS DEL CONGLOMERADO 1 (K-MEDIAS)	89
Tabla 4.18.	CARACTERÍSTICAS DEL CONGLOMERADO 2 (K-MEDIAS)	89
Tabla 4.19.	CARACTERÍSTICAS DEL CONGLOMERADO 3 (K-MEDIAS)	90
Tabla 4.20.	CARACTERÍSTICAS DEL CONGLOMERADO 4 (K-MEDIAS)	91
Tabla 4.21.	CARACTERÍSTICAS DEL CONGLOMERADO 5 (K-MEDIAS)	91
Tabla 4.22.	CLASIFICACIÓN CON LA RED BPN	96
Tabla 4.23.	COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI	96
Tabla 4.24.	ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI.....	97
Tabla 4.25.	CARACTERÍSTICAS DEL GRUPO 1 (BPN).....	98
Tabla 4.26.	CARACTERÍSTICAS DEL GRUPO 2 (BPN).....	98
Tabla 4.27.	CARACTERÍSTICAS DEL GRUPO 3 (BPN).....	99

Tabla 4.28.	CARACTERÍSTICAS DEL GRUPO 4 (BPN)	99
Tabla 4.29.	CARACTERÍSTICAS DEL GRUPO 5 (BPN)	100
Tabla 4.30.	COMPARACIÓN DE RESULTADOS POR ESTRATO DE POBREZA, SEGÚN MÉTODO DE CLASIFICACIÓN	101
Tabla 4.31.	CUADRO COMPARATIVO DE METODOLOGÍAS UTILIZADAS	104
Tabla 7.1.	COMPONENTES DEL INDICADOR NBI	112
Tabla 7.2.	CALIFICACIÓN DE MATERIALES DE PAREDES.....	116
Tabla 7.3.	CALIFICACIÓN DE MATERIALES DE TECHOS.....	117
Tabla 7.4.	CALIFICACIÓN DE MATERIALES DE PISOS.....	117
Tabla 7.5.	CALIFICACIÓN DE TENENCIA DE COCINA	120
Tabla 7.6.	CALIFICACIÓN DEL ABASTECIMIENTO DE AGUA.....	123
Tabla 7.7.	CALIFICACIÓN DEL SERVICIO SANITARIO	124
Tabla 7.8.	CALIFICACIÓN DE ENERGÍA ELÉCTRICA	124
Tabla 7.9.	CALIFICACIÓN DEL COMBUSTIBLE PARA COCINAR.....	125
Tabla 7.10.	NORMAS DE LOS COMPONENTES DE EDUCACIÓN	126
Tabla 7.11.	CALIFICACIÓN DE LA ATENCIÓN DE SALUD.....	128
Tabla 7.12.	CONDICIÓN O ESTRATO DE POBREZA	130

ÍNDICE DE FIGURAS

Figura 1.1	Warren McCulloch.	3
Figura 1.2	Walter Pitts	4
Figura 2.1.	Sistema de circuitos de Cajal.....	11
Figura 2.2.	Estructura de una neurona biológica típica	12
Figura 2.3.	Modelo de neurona McCulloch-Pitts	17
Figura 2.4.	Tipología de las unidades de procesamiento de neuronas	19
Figura 2.5.	Modelo genérico de neurona artificial	20
Figura 2.6.	Ejemplo de reglas de propagación.....	22
Figura 2.7.	Taxonomías de acuerdo al tipo de arquitectura, aprendizaje y aplicaciones.....	25
Figura 2.8.	Ciclo del aprendizaje supervisado.....	31
Figura 2.9.	Ciclo del aprendizaje reforzado.....	32
Figura 2.10.	Ciclo del aprendizaje no supervisado.....	33
Figura 2.11.	Estructura de la red backpropagation	39
Figura 3.1.	Tipos de correlación.....	49
Figura 3.2.	Dendograma.....	52
Figura 3.3.	Resumen de iteraciones	53
Figura 3.4.	Límites aleatorios.....	55
Figura 3.5.	Centros más alejados entre sí	56
Figura 3.6.	Centros con información a priori	56
Figura 3.7.	Distancia de elementos al centro de cada grupo.....	57
Figura 4.1.	Resumen de metodología NBI	75
Figura 4.2.	Pobreza por departamento – Método NBI.....	78
Figura 4.3.	Estratos de pobreza por fuentes de información	79
Figura 4.4.	Resumen de metodología por conglomerados.....	80
Figura 4.5	Clasificación jerárquica (Dendograma)	81
Figura 4.6.	Resumen del algoritmo Backpropagation.....	92
Figura 4.7.	Formulario o interfaz de la red BPN	94
Figura 4.8.	Simulación de la red BPN	95
Figura 4.9.	Estratos de pobreza según metodología de clasificación.....	102
Figura 4.10.	Pobreza en los hogares de Bolivia según metodología de clasificación	103

1

INTRODUCCIÓN

La tecnología y los grandes inventos de la humanidad, se lograron en su mayoría, gracias a la necesidad de conocer el funcionamiento y la estructura de todo nuestro entorno. Tratar de emular la naturaleza fue sin duda la base de muchos inventos y teorías que permiten alcanzar nuevas experiencias y mejorar nuestras necesidades.

El cerebro es un procesador de información con características notables. Es capaz de procesar inmediatamente grandes cantidades de información procedentes de los sentidos, combinarla o compararla con la información almacenada y dar respuestas adecuadas incluso en situaciones nuevas. Logra distinguir una cara en una calle mal iluminada, leer entre líneas en una conversación, o discernir un susurro en una sala ruidosa; pero lo más impresionante de todo es su capacidad aprender a representar la información necesaria para desarrollar tales habilidades sin instrucciones explícitas que dirijan este proceso.

Aunque todavía se ignora mucho sobre la forma en que el cerebro aprende a procesar la información, se han desarrollado modelos que tratan de mimetizar tales habilidades; a este tipo de modelos se les conoce como Redes Neuronales Artificiales (RNA¹). La elaboración de estos modelos supone en primer lugar, la deducción de los rasgos o características esenciales de las neuronas y sus

¹ En adelante se utilizará la abreviación RNA para hacer referencia a las redes neuronales artificiales.

conexiones, y en segundo lugar, la implementación del modelo computacional de forma que se pueda simular.

El estudio de las redes neuronales artificiales ha sido de gran interés para diversas ramas del conocimiento, en un principio desde el punto de vista biológico con la idea de poder explicar el funcionamiento del cerebro y de los procesos neuronales, y posteriormente también con el objetivo de desarrollar sistemas "inteligentes" que fueran capaces de desarrollar numerosas actividades.

Las redes neuronales artificiales son sistemas conexionistas dentro del campo de la Inteligencia Artificial, las cuales, dependiendo del tipo de arquitectura neuronal, pueden tener diferentes aplicaciones. Pueden utilizarse para el reconocimiento de patrones, la comprensión de información, el modelaje y la reducción de la dimensionalidad, el agrupamiento, la clasificación (aplicación del presente trabajo de tesis), etc.

"¿Por qué no se puede construir un computador que piense?, ¿Por qué no se puede esperar que máquinas capaces de realizar cien millones de operaciones por segundo, sean capaces de entender el significado de las formas en imágenes visuales, o incluso distinguir entre distintas clases de objetos similares? ¿Por qué no puede esa misma máquina aprender a partir de la experiencia, en lugar de repetir indefinidamente un conjunto explícito de instrucciones generadas por un programador humano?"²

Los ingenieros y programadores encargados de diseñar computadoras son los que se enfrentan a este tipo de problema y son los que hasta el día de hoy no alcanzan a responder estas preguntas. Si bien se lograron construir artefactos con sencillas instrucciones de control que dan cierta inteligencia a los electrodomésticos, es claro que la limitación principal es la compleja

² James A. Freeman, David M. Skapura. (1993). Redes Neuronales. Wilmington, Delaware, EUA: Addison-Wesley Iberoamericana, S.A.

programación que se requiere para realizar estas tareas. Tradicionalmente un programador o un analista especifican los "códigos" de cada faceta del problema en orden para que la computadora pueda "entender" la situación.

Es precisamente el lenguaje secuencial, el que impide a los programadores emular el funcionamiento cerebral. Una alternativa que no requiere el código explícito del problema y utiliza otro tipo de instrucción o lenguaje es el "aprendizaje" de una RNA.

Las primeras personas que trataron al cerebro como un organismo computacional fueron los científicos McCulloch³ y Pitts⁴. Ellos diseñaron la primera RNA, en su trabajo "A logical calculus of the ideas immanent in nervous activity" (Un cálculo lógico de las ideas inherentes en actividad nerviosa) presentado en el año 1943. En esta investigación, las neuronas artificiales fueron propuestas como modelos de neuronas biológicas y como componentes de circuitos que pueden desarrollar eventos computacionales.

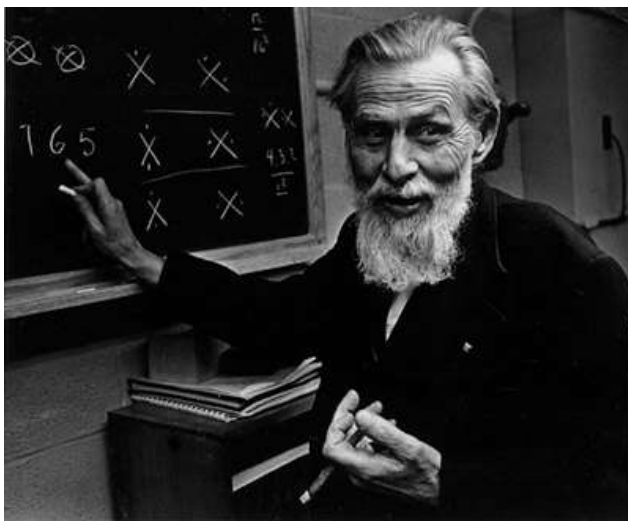


Figura 1.1 Warren McCulloch.

El prototipo de McCulloch y Pitts fue el primer modelo neuronal moderno, y sirvió de inspiración en el desarrollo de otros esquemas neuronales. Sin embargo, muchos de los estudios que citan este modelo, no interpretan correctamente el sentido original de los autores, atribuyéndole características o funciones que no

³ Warren Sturgis McCulloch (16 noviembre 1898 a 24 septiembre 1969). Neurofisiólogo estadounidense, conocido por su trabajo sobre la base de las teorías del cerebro y su contribución a la cibernética.

⁴ Walter Pitts (1926). Matemático estadounidense autodidacta de tortuosa, breve y misteriosa vida.

fueron descritas por McCulloch y Pitts, o restándole importancia a la capacidad de procesamiento original.

Durante el periodo de 1967 a 1982 comienzan los primeros indicios sobre inteligencia artificial. Estas investigaciones se llevan a cabo fuera de Estados Unidos, países como Japón, la Unión Soviética y algunas naciones del continente europeo comienzan este nuevo reto. Sin embargo, unos meses después, una gran empresa estadounidense investiga una cadena neuronal como un proceso adaptativo de reconocimiento de patrones, modelos biológicos e imágenes.

Pese al escepticismo de la mayoría de los analistas, muchos investigadores empezaron a construir audaces propuestas para explorar el desarrollo de neurocomputadoras y las diferentes formas de utilizar las RNA.

La primera aplicación de una RNA surge en la Agencia de Defensa Avanzada de Proyectos (Defense Advanced Research Projects Agency (DARPA)), donde Ira Skurnick (Programa Maestro para la Oficina de Defensa), rechazó continuar con el concepto convencional de trabajo e insistió que escucharan sus argumentos a cerca de sus investigaciones sobre la neurocomputación.

Pocos meses después, el programa maestro Skurnic se consolidaría con mucho éxito, dando con esto un gran salto tecnológico que respaldaría el lanzamiento de la neurocomputación y consolidaría a DARPA'S como uno de los jueces mundiales de la moda tecnológica.



Figura 1.2 Walter Pitts

Una de las grandes contribuciones surge en el año 1982. John Hopfield⁵, presentó un documento a la comunidad científica, en el que afirmó que el enfoque de la inteligencia artificial no alcanzaría a imitar al cerebro humano, pero se podrían utilizar máquinas capaces de resolver problemas dinámicos. Él demostró lo que sus redes fueron capaces de hacer y cómo funcionaban. Su gran conocimiento en el análisis matemático, convenció a los científicos en renovar el interés por la investigación de la inteligencia artificial.

Sus ideas fueron la guía para el desarrollo de nuevas RNA, que con el tiempo, llegaron a ser conocidas como Modelos de Hopfield.

Si bien los conceptos básicos de las RNA fueron planteados en 1943, su aplicación en el campo de la estadística es más reciente y, no puede decirse que, en la actualidad sean una herramienta que forme parte del conjunto de instrumentos utilizados por los investigadores para el análisis estadístico.

La estadística comprende un conjunto de métodos que sirven para recoger, organizar, resumir y analizar datos para la toma de decisiones. Las RNA han sido descritas por algunos como técnicas de ajuste estadístico multivariante⁶ inspiradas en la biología.

El interés por el análisis multivariante se extendió entre autores de ramas muy diversas (además de estadísticos y matemáticos, a psicólogos, sociólogos, economistas, etc.).

La necesidad de diferenciar, clasificar o establecer grupos contribuye también a mejorar el conocimiento de las cosas. Tal necesidad se presenta frecuentemente en todas las ramas relacionadas con el ámbito social, y que habitualmente, tienen un marcado carácter multidimensional.

⁵ John Hopfield. Físico de fama mundial de la Universidad de California. Fue catalogado como el responsable del renacimiento de las Redes Neuronales y la Inteligencia Artificial.

⁶ Se utilizarán los términos "Multivariante" o "Multivariable" indistintamente.

El Análisis Cluster⁷ reúne un conjunto de técnicas del análisis multivariante, cuya principal aplicación reside en la clasificación u obtención de tipologías que no están previamente identificadas.

En la actualidad, las aplicaciones del análisis cluster fueron empleadas en una gran variedad de estudios. Así, en biología se ha utilizado para establecer clasificaciones de seres vivos, ya sean plantas o animales e identificar especies y familias; en psicología, para identificar tipologías de personalidad, de comportamientos paranoicos o esquizofrénicos; en sociología, para definir grupos sociales con arreglo a unas características determinadas; en medicina, para establecer tipos de síntomas o trastornos; en arqueología para describir diferentes objetos líticos⁸ o herramientas de diversa índole; en estudio de mercados (marketing) para estudiar tipologías de comportamientos de compra y de consumidores, en la segmentación de mercados o ciertas necesidades con respecto a un producto; y así en otras ramas científicas.

Se realizaron estudios comparando métodos estadísticos y neuronales, en los cuales, no es posible afirmar con certeza que los modelos neuronales sobrepasen en eficiencia a las técnicas estadísticas. Se necesita evaluar bajo qué condiciones las RNA son procedimientos superiores a los procesos estadísticos convencionales de análisis de datos.

Existe un claro paralelismo entre modelos estadísticos y neuronales, sin embargo, hay algunos modelos de RNA para los que no existe una técnica estadística equiparable. La Tabla 1.1 muestra los modelos más usuales.

⁷ También denominado de grupos, de conglomerados o tipologías.

⁸ Lítico es una pieza tallada en piedra que reúne un conjunto de rasgos característicos que definen un modelo o patrón de algún artefacto.

A inicios de los 90's, se desarrollaron muchos avances en la investigación de las RNA en todo el mundo. El desafío de hoy en día es buscar maneras de implementar electrónicamente esta tecnología.

Alrededor de la neurocomputación se inicia una interesante historia, pero el camino aún se está desarrollando, "*estamos al final del principio*" (Winston Churchill⁹).

El presente trabajo consta de 4 capítulos; en el primero se presenta la introducción, el objetivo y la justificación de la investigación.

En el capítulo 2 se muestran los fundamentos de las redes neuronales artificiales, así como un breve repaso a la historia de la RNA y su paralelismo con los sistemas neuronales biológicos.

Tabla 1.1. COMPARACIÓN ENTRE MODELOS ESTADÍSTICOS Y RNA

Modelo Estadístico	Modelo de Red Neuronal
Regresión Lineal Múltiple	Perceptrón Simple con Función Lineal
Regresión Logística	Perceptrón Simple con Función Logística
Regresión No Lineal Múltiple	Perceptrón Multicapa con Función Lineal en la salida
Función Discriminante Lineal	Perceptrón Simple con Función Umbral
Función Discriminante No Lineal	Perceptrón Multicapa con Función Logística en la salida
Análisis de Componentes Principales	Perceptrón Multicapa Autoasociativo

Se exponen también las características generales de los modelos neuronales artificiales y las motivaciones que desembocan en la introducción de éstas como

⁹ Sir Winston Leonard Spencer Churchill, (1874 - 1965). Político británico considerado el último de los grandes estadistas, siempre será recordado por su rara habilidad para predecir los acontecimientos futuros.

alternativa o complemento a los sistemas de procesamiento más utilizados en la actualidad.

Una vez desarrollados los conceptos básicos relacionados con las RNA, se presenta el modelo *Backpropagation*, su teoría y funcionalidad. Este algoritmo es uno de los sistemas neuronales más conocidos y utilizados en los últimos tiempos.

Posteriormente, en el capítulo 3 se explica el concepto del Análisis Cluster y se desarrolla la metodología de conglomerados no jerárquicos. A continuación se presenta el algoritmo de k-medias y se analiza la formación de los conglomerados finales.

Finalmente, en el capítulo 4, se realiza una aplicación a la información estadística de la Encuesta de Hogares, que mide en una de sus fases, las condiciones de vida en los hogares de Bolivia. Se muestran las soluciones generadas por cada uno de los métodos expuestos, y se resaltan las ventajas y desventajas de cada uno de tal forma que el lector pueda decidirse sobre uno de ellos, o bien, optar por una combinación para sustentar la solución generada.

1.1 Objetivo

El objetivo principal de este trabajo es estudiar, desarrollar, analizar y sobre todo comparar los métodos estadísticos de Redes Neuronales Artificiales y Análisis de Conglomerados no Jerárquicos como instrumentos de clasificación de información estadística y aplicarlos a la Encuesta de Hogares¹⁰, enfocado al tema de la medición de las condiciones de vida en los hogares en Bolivia.

¹⁰ Encuesta Nacional Continua de Hogares (ENCH) – Instituto Nacional de Estadística (INE).

1.2 Problema de investigación

En los últimos años se han desarrollado metodologías para medir condiciones de vida de los hogares bolivianos, mediante indicadores de pobreza, tales como, el NBI¹¹, la incidencia de pobreza y la brecha de pobreza, que son indicadores descriptivos que resumen la información sin mostrar un panorama con la extensión de los datos como lo hace el análisis de conglomerados o las redes neuronales.

El NBI, la incidencia y la brecha de pobreza, son indicadores “clásicos” que miden las condiciones de vida desde un punto de vista descriptivo a base de promedios.

Sin embargo, el análisis de conglomerados no jerárquico clasifica la información mediante técnicas multivariantes y las RNA realizan este proceso utilizando algoritmos de aprendizaje artificial no secuenciales.

En este sentido, se pretende investigar cuál de los dos métodos anteriormente citados (Cluster o RNA) es óptimo, y para esto, se desarrollarán las distintas metodologías con el fin de comprender el enlace y la comparación entre ellos.

1.3 Hipótesis del trabajo

“En relación al método de conglomerados no jerárquicos, las redes neuronales artificiales tienen mayor precisión en la clasificación de condiciones de vida de los hogares de Bolivia”.

¹¹ Indicador de Necesidades Básicas Insatisfechas (NBI)

2

REDES NEURONALES ARTIFICIALES

En el presente capítulo se realiza un breve repaso a la historia del desarrollo de sistemas y máquinas dotadas de cierta inteligencia así como su paralelismo con los sistemas neuronales biológicos. Además, se presentan las características generales de los modelos neuronales artificiales y su funcionamiento.

Gran parte de los inventos del hombre se deben a su capacidad para explicar y emular funciones realizadas por seres vivos. Por ejemplo, se puede citar el radar, que surge como imitación de la forma en la que un murciélago es capaz de detectar los objetos que están en su camino sin necesidad de verlos, por medio de la emisión de una onda ultrasónica, la posterior recepción de la señal de eco y su procesamiento.

Aunque el hombre ha sido capaz de reproducir funciones de los animales, aún se enfrenta con el reto de poder imitar, la llamada por muchos la máquina perfecta, el cerebro humano.

Cuando la neurociencia pudo explicar de forma un poco convincente el funcionamiento de la unidad principal de procesamiento de información que posee el cerebro, la neurona, surge casi de manera automática la idea de poder imitar dicho funcionamiento en un elemento alternativo, "la neurona artificial".

Una de las metodologías con mayor auge en la última década son los modelos de RNA, que en esencia son estructuras formales de carácter matemático y

estadístico con la propiedad de aprendizaje, es decir que, en la mayoría de los casos, la adquisición de conocimiento es desarrollada en base a ejemplos.

Este aprendizaje se produce mediante un estilo de computación que intenta simular algunas de las capacidades que posee nuestro cerebro: la capacidad de memorizar y asociar hechos. Si se examina con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se puede advertir que todos ellos tienen una característica común, la experiencia.

En este sentido, las RNA no son más que un modelo artificial y simplificado del cerebro humano, es decir, un sistema para el tratamiento de información, que es capaz de adquirir conocimiento a través de la experiencia y cuya unidad básica de procesamiento está inspirado en la célula fundamental del sistema nervioso humano, la neurona.

2.1 Introducción biológica

Antes de comenzar el estudio de las RNA es conveniente exponer algunos conceptos básicos de los sistemas neuronales biológicos, para poder establecer más fácilmente el paralelismo entre ambos.

El inicio de las redes neuronales artificiales comienza con el científico Santiago Ramón y Cajal, un gran neuroanatomista español descubridor de la estructura neuronal del sistema nervioso (Figura 2.1). A finales del siglo XIX la teoría reticularista¹² sostenía que el sistema nervioso estaba formado por una red continua de fibras nerviosas.



Figura 2.1. Sistema de circuitos de Cajal.

¹² La Teoría Reticularista negaba la individualidad de las células nerviosas (neuronas). Cajal desmintió la teoría reticularista gracias a sus investigaciones, demostró que las relaciones entre las células nerviosas o neuronas eran de contigüidad y no de continuidad, dejando firmemente establecido que la neurona es la unidad histológica y fisiológica del sistema nervioso. Esta es la llamada Teoría Neuronal, por la que se le concedió el premio Nobel en 1906.

Después de muchos años de trabajo, Ramón y Cajal en el año 1888, demostró que el sistema nervioso en realidad estaba compuesto por una red de células individuales, las neuronas, ampliamente interconectadas entre sí. Pero no sólo observó los pequeños espacios vacíos que separaban unas neuronas de otras, sino que también estableció que la información fluye de una neurona a otra desde las dendritas hasta el axón, atravesando el soma, descubrimiento básico para el desarrollo de las neurociencias en el siglo XX.

Se cree que el sistema nervioso está compuesto por cien mil millones de neuronas, este tipo de células pueden presentarse en múltiples formas, aunque muchas de ellas tienen un aspecto similar muy peculiar (Figura 2.2), con un cuerpo celular o soma (de entre 10 y 80 micras de longitud), del que surge un denso árbol de ramificaciones (árbol dendrítico) compuesto por las dendritas, y del cual parte una fibra tubular denominada axón que también se ramifica en su extremo final para conectarse con otras neuronas.

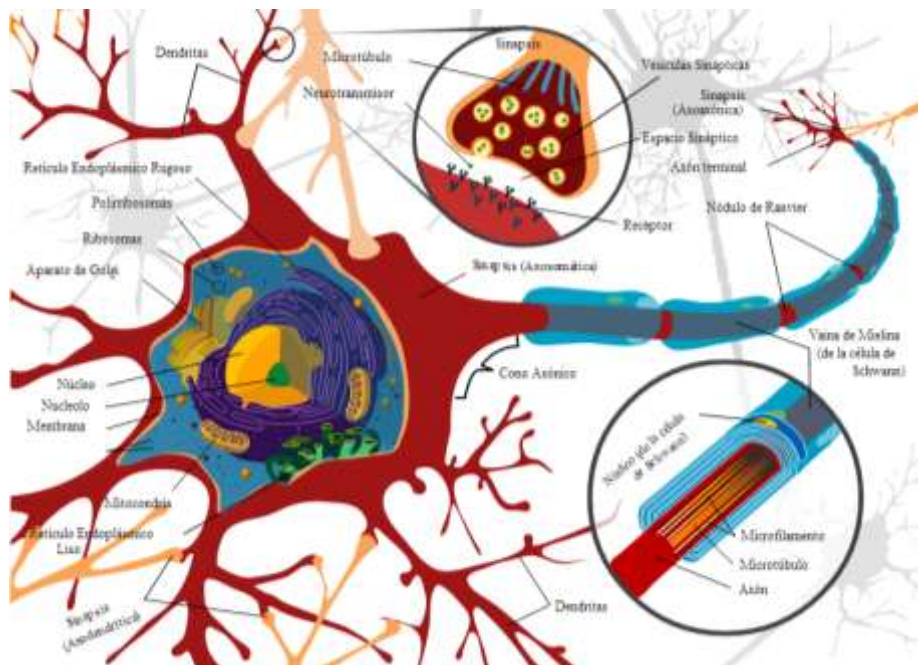


Figura 2.2. Estructura de una neurona biológica típica

Analizando el punto de vista funcional, las neuronas constituyen procesadores de información sencillos. Como todo sistema de este tipo, poseen un canal de entrada de información, las dendritas, un órgano de cómputo, el soma, y un canal de salida, el axón. En el espacio inter neuronal el axón envía la información a otras neuronas, mientras que en las neuronas motoras lo hace directamente al músculo.

Las receptoras o sensoras conforman un tercer tipo de neuronas; que en vez de recibir la información de otras neuronas, la reciben directamente del exterior. Se calcula que una neurona del córtex¹³ cerebral recibe información, por término medio, de unas 10.000 neuronas (convergencia), y envía impulsos a varios cientos de ellas (divergencias).

En la región externa del cerebro (córtex) se aprecia la existencia de una organización horizontal en capas (se suelen señalar unas seis capas), coexistiendo una organización vertical en forma de columnas de neuronas. Hay grupos neuronales, compuestos por millones de neuronas pertenecientes a una determinada región del cerebro, que constituyen unidades funcionales especializadas en ciertas tareas (por ejemplo, existe un área visual, un área auditiva, etc.). Se tiene evidencia de que el procesamiento en el sistema nervioso involucra la actuación de muchos subsistemas, que intercambian continuamente información.

2.2 Origen y transferencia de señales nerviosas

El enlace entre dos neuronas se denomina sinapsis¹⁴. En el tipo de sinapsis más común no existe un contacto físico entre las neuronas, sino que éstas permanecen separadas por un pequeño vacío de unas 0,2 micras. En relación a

¹³ Capa externa del cerebro

¹⁴ La palabra sinapsis viene del griego "synapto" que significa unión o conexión estrecha.

la sinapsis, se habla de neuronas presinápticas (que envían las señales) y postsinápticas (que las reciben).

Las señales nerviosas se pueden transmitir eléctrica o químicamente. La transmisión química prevalece fuera de la neurona y se basa en el intercambio de neurotransmisores, mientras que la eléctrica lo hace en el interior mediante descargas que se producen en el cuerpo celular, y que se propagan por el axón.

El tipo de comunicación más habitual entre dos neuronas es del tipo químico. La neurona presináptica libera unas complejas sustancias químicas denominadas neurotransmisores (como la adrenalina), que atraviesan el vacío sináptico. Si la neurona postsináptica posee en las dendritas o en el soma canales sensibles a los neurotransmisores liberados, los fijarán, y como consecuencia de ello permitirán el paso de determinados iones a través de la membrana. Las corrientes iónicas que de esta manera se crean provocan pequeños potenciales postsinápticos, excitadores o inhibidores, que se integrarán en el soma; éste es el origen de la existencia de sinapsis excitatorias y de sinapsis inhibitorias¹⁵.

Mientras más grande sea el estímulo, mayor será la frecuencia de respuesta, esto hasta que se alcanza una saturación conforme y se acercan a la frecuencia máxima (umbral). La señal que es recibida por la neurona posee diferentes grados de ponderación; cuando la ponderación es alta, el potencial de membrana de la neurona se vuelve positivo, rebasa el umbral y envía la correspondiente señal a la siguiente neurona del circuito; contrariamente cuando la ponderación es baja, el umbral no se rebasa y no se transmite señal a las otras neuronas.

Para establecer una similitud directa entre la actividad sináptica y la analogía con las RNA, es necesario considerar los siguientes aspectos: las señales que llegan a la sinapsis son las entradas a la neurona; éstas son ponderadas (atenuadas o

¹⁵ Existen evidencias experimentales que indican que un axón sólo puede generar sinapsis excitatorias o inhibitorias, pero no de ambos tipos (Ley de Dale).

amplificadas) a través de un parámetro, denominado peso sináptico, asociado a la sinapsis correspondiente.

2.3 Emulación de los sistemas nerviosos

Los tres criterios clave de los sistemas nerviosos que se pretende emular en los artificiales, son: paralelismo de cálculo, memoria distribuida y adaptabilidad al entorno. De esta manera, se puede hablar de las RNA como sistemas paralelos, distribuidos y adaptativos.

2.3.1 Procesamiento paralelo

El cerebro humano promedio tarda aproximadamente 20ms¹⁶ en preprocesar una imagen compuesta por millones de píxeles, extraer sus características, analizarla e interpretarla. Ningún sistema creado por el hombre es capaz de realizar algo semejante. La clave reside en que en este último caso los miles de millones de neuronas que intervienen en el proceso de visión están operando en paralelo sobre la totalidad de la imagen.

2.3.2 Memoria distribuida

Mientras que en un procesador la información ocupa posiciones de memoria establecidas o fijas, en las RNA se encuentra distribuida por las sinapsis de la red, de modo que si una sinapsis resulta dañada, se pierde sólo una parte muy pequeña de la información.

¹⁶ Un milisegundo es el período que corresponde a la milésima fracción de un segundo (0,001s).

Su simbología, al igual que otras milésimas partes de distintas magnitudes, como pudieran ser la masa o la longitud, viene especificada mediante una "m" minúscula antepuesta al símbolo de la magnitud fundamental, que en el caso del segundo es una letra "s" (1 s = 1000 ms).

Los sistemas neuronales biológicos son redundantes, de modo que muchas neuronas y sinapsis pueden realizar un papel similar; en definitiva, el sistema resulta tolerante a fallos. Por ejemplo, cada día mueren miles de neuronas en nuestro cerebro, y sin embargo tienen que pasar muchos años para que se resientan nuestras capacidades. La razón por la que las RNA son tolerantes a fallos es que tienen su información distribuida.

Las RNA son los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Hay dos aspectos distintos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos, ésta es una tolerancia a fallos respecto a los datos. Segundo, pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

2.3.3 Aprendizaje adaptativo

Es una de las características más atractivas de las RNA. Esto significa que aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos, es decir, no es necesario elaborar modelos “a priori” ni especificar funciones de distribución de probabilidad. Una RNA no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de los pesos de los enlaces mediante aprendizaje. Las RNA resuelven problemas mediante autoaprendizaje y autoorganización.

2.4 Modelos neuronales artificiales

Los modelos neuronales asumen muchas simplificaciones del modelo biológico para poder plantear su desarrollo matemático, en esta línea, el primer modelo artificial fue diseñado por McCulloch-Pitts (1943) (Figura 2.3), el cual utilizaba unidades de procesamiento denominadas neuronas que poseían dos estados

discretos. Asociados a cada uno de ellos, se conseguía una salida que se transmitía a lo largo de la estructura vinculada a la red neuronal, pero con la limitación que sólo permitían computar funciones booleanas.

El modelo de McCulloch-Pitts se basa en las siguientes hipótesis. En primer lugar, el estado de una neurona en el tiempo “ $t + 1$ ”, depende solamente del estado que poseía en el período anterior, “ t ”.

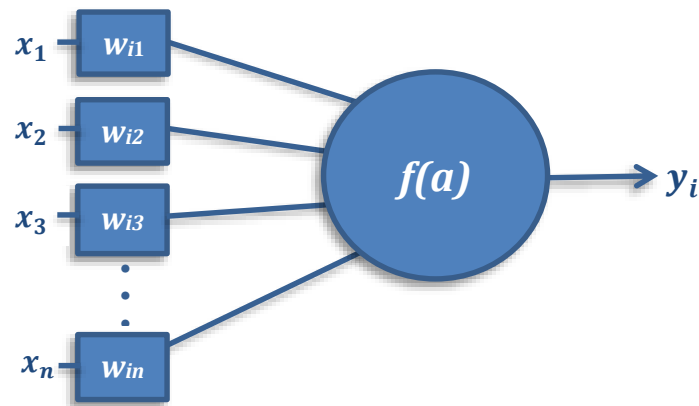


Figura 2.3. Modelo de neurona McCulloch-Pitts

En segundo lugar, una neurona estará activada o no si supera un umbral θ , y en último lugar, se asume la sincronía¹⁷ entre las entradas y las salidas.

La formalización del diseño del modelo de McCulloch-Pitts consiste, en primer lugar, en definir el estado de la entrada, “ x_t ” y en segundo lugar, la salida en el momento t , “ y_t ”. La expresión que describe su funcionamiento es,

$$y_i = f \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right) = f(a)$$

¹⁷ La dinámica que rige la actualización de los estados de las neuronas (evolución de la red neuronal) puede ser de dos tipos: asincrónico y sincrónico. En el primer caso, las neuronas evalúan su estado continuamente, según les va llegando información, y lo hacen de forma independiente. En el caso sincrónico aunque la información llega de forma continua, los cambios se realizan simultáneamente. Los sistemas neuronales biológicos muy probablemente actúan de forma mixta.

$$f(a) = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{e.o.c.} \end{cases}$$

En los modelos neuronales artificiales la información se genera a partir del aprendizaje de la estructura interna de los datos, de forma que son las propias conexiones o pesos donde se retiene el conocimiento. Es de gran importancia notar que no existe a priori una definición explícita de la forma del conocimiento, el propio algoritmo iterativo de estimación de los parámetros (pesos) desconocidos, se encarga de extraer la presencia de regularidades en los datos.

Los temas más relevantes en los modelos neuronales son: su arquitectura o topología¹⁸, el tipo de sus unidades de procesamiento, el tipo de conexiones de estas unidades o neuronas y los tipos de aprendizaje.

La arquitectura de una red neuronal se refiere a la forma de las conexiones entre las unidades neuronales. Su forma genera toda una familia de posibles modelos, cuya gran variedad obliga a la vertebración de los mismos mediante clasificaciones o taxonomías.

En una primera aproximación, se puede encontrar una clasificación en función a los tipos de las salidas que genera el modelo, divididos en: modelos deterministas y modelos estocásticos. Para el caso determinista tenemos que cada neurona sigue una ley del tipo,

$$y = f\left(\sum_{i=1}^n w_i x_i\right)$$

donde $f(x)$ es una función de activación, en cambio para las redes con neuronas estocásticas, la activación de la red se interpreta como una probabilidad de un estado lógico tal y como se expresa en las siguientes ecuaciones,

¹⁸ La topología de la red (forma de la red) muestra cómo los diferentes nodos están conectados entre sí, y la forma de comunicación.

$$P(y = 1) = f\left(\sum_{i=1}^n w_i x_i\right)$$

$$P(y = 0) = 1 - P(y = 1) = 1 - f\left(\sum_{i=1}^n w_i x_i\right)$$

donde la salida es un valor continuo entre [0,1] que se interpreta como una probabilidad.

En relación a la tipología existente en las unidades de procesamiento o neuronas, existen neuronas visibles y neuronas ocultas. Por neuronas visibles se entienden tanto las entradas (variables exógenas) como las salidas (variables endógenas), en cambio las neuronas ocultas, poseen la función de capturar la representación interna de los datos. Éstas pueden no estar conectadas directamente con las neuronas visibles (Figura 2.4).

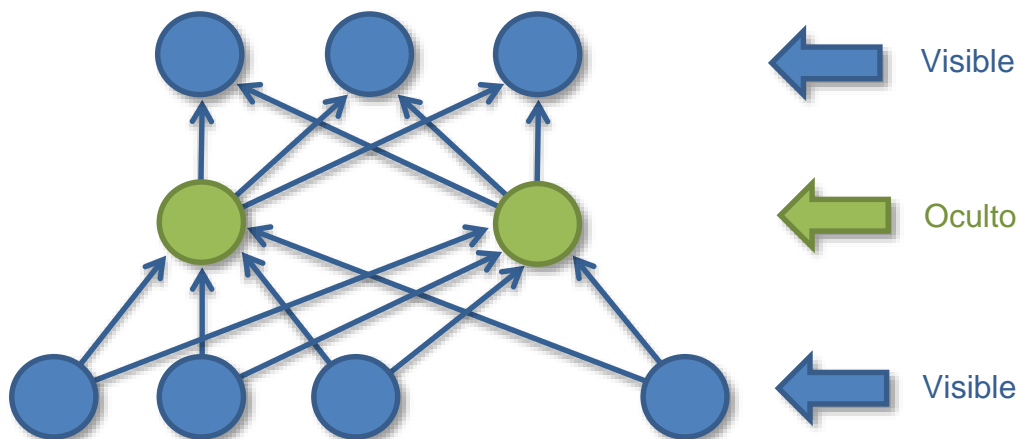


Figura 2.4. Tipología de las unidades de procesamiento de neuronas

El tipo de conexiones que se establecen entre las unidades de procesamiento o neuronas, se dividen en dos grandes grupos, en primer lugar se están los modelos que se propagan en una sola dirección, denominados “*feed-forward*” y en segundo lugar, los modelos recurrentes, cuyas conexiones se establecen en

todas las direcciones incluso con procesos de realimentación, es decir, las propias neuronas consigo mismas.

Existen dos tipos de aprendizaje; supervisado y no supervisado. La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo (supervisor o maestro) que controle el proceso de aprendizaje de la red.

2.5 Modelo general de neurona artificial

Los elementos básicos de un modelo de neurona artificial son (Figura 2.5):

- Un conjunto de entradas, $x_j(t)$
- Los pesos sinápticos de la neurona i , w_{ij}
- La regla de propagación, $h_i(t) = \sigma(w_{ij}, x_j(t))$
- La función de activación o transferencia, $g_i(h_i(t))$
- La función de salida, $f_i(g_i(h_i(t)))$

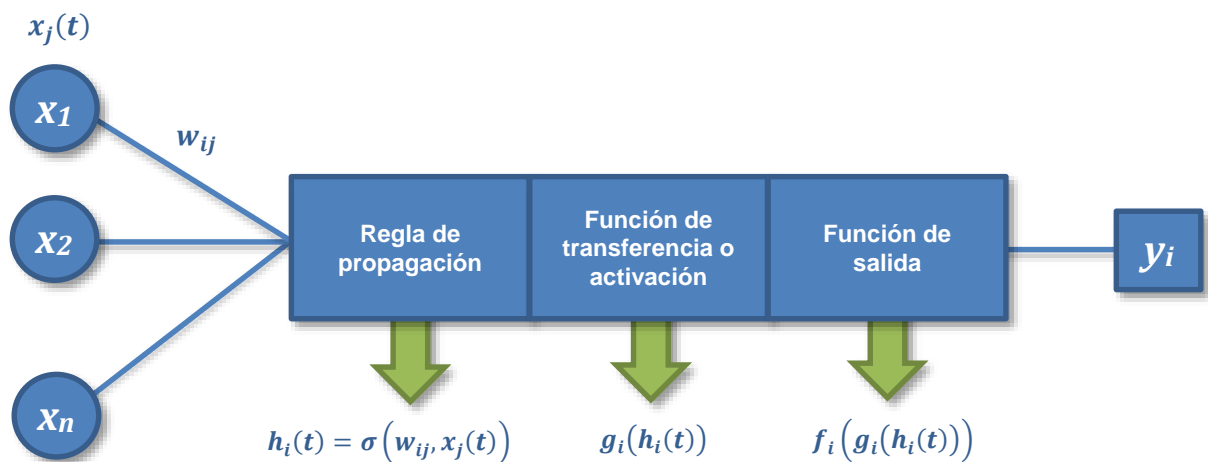


Figura 2.5. Modelo genérico de neurona artificial

Los elementos anteriores posibilitarán una generalización formal, que permitirá describir tanto los modelos supervisados como los no supervisados. A continuación se definen cada uno de estos elementos:

2.5.1 Conjunto de entradas

El conjunto de entradas $x_i(t)$ se refiere a un vector procedente del exterior o de otras neuronas, es decir, son las señales que llegan a la sinapsis.

2.5.2 Pesos sinápticos

Los pesos sinápticos de la neurona i , w_{ij} representan la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i , al igual que en una neurona biológica se establecen sinapsis entre las dendritas de una neurona y el axón de otra. Si el peso es positivo tenderá a excitar a la neurona postsináptica, si es negativo tenderá a inhibirla.

2.5.3 Reglas de propagación

La regla de propagación $h_i(t) = \sigma(w_{ij}, x_j(t))$, también conocida como función de ponderación o de excitación, proporciona el valor del potencial postsináptico de la neurona i en función de sus pesos y entradas. La regla de propagación es un elemento relevante que puede poseer diferentes formas, en la Figura 2.6 se muestran algunas de ellas.

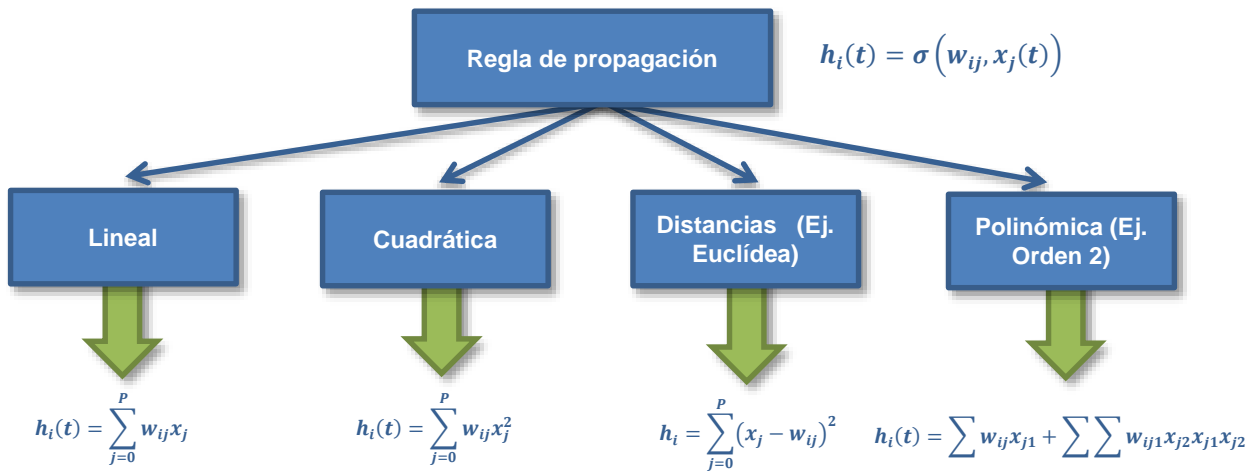


Figura 2.6. Ejemplo de reglas de propagación

La regla de propagación más común, especialmente en los modelos basados en el cálculo de distancias entre vectores (como en los mapas autoorganizados) es la distancia euclídea que representa la distancia (al cuadrado) existente entre el vector de entradas y el de pesos. Cuando ambos vectores son muy similares, la distancia es muy pequeña; cuando son muy diferentes, la distancia crece. Se pueden utilizar también otros tipos de distancias, como la Manhattan o la de Mahalanobis.

2.5.4 La función de transferencia

La función de transferencia o activación, $g_i(h_i(t))$ filtra el valor de la regla de propagación para compararse con algún valor umbral para determinar la salida final de la neurona. Si la suma es mayor que el valor umbral, la neurona generará una señal. Si la suma es menor que el valor umbral, ninguna señal será generada. La función de activación puede o no existir, siendo en este caso la salida la misma función de propagación.

En ciertas situaciones los algoritmos de aprendizaje requieren que la función de activación cumpla con la condición de ser derivable. La más empleada en este sentido es la función back propagation. Otra función clásica es la gaussiana, que se utiliza junto con reglas de propagación que involucran el cálculo de cuadrados de distancias (por ejemplo, la euclídea) entre los vectores de entradas y pesos.

En ocasiones se emplean funciones sinusoidales, como en aquellos casos en los que se requiere expresar explícitamente una periodicidad temporal. Las principales funciones de activación se muestran a continuación:

<p>Función identidad o lineal</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = a$	<p>Función escalón</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \begin{cases} 1 & a \geq 0 \\ 0 & a < 0 \end{cases}$
<p>Función lineal a tramos</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \begin{cases} 0 & a < 0 \\ a & 0 \leq a < 1 \\ 1 & a > 0 \end{cases}$	<p>Función escalón simétrica</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a < 0 \end{cases}$
<p>Función lineal a tramos simétrica</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \begin{cases} -1 & a < 0 \\ a & -1 \leq a < 1 \\ 1 & a > 0 \end{cases}$	<p>Función sinusoidal</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \text{sen}(a)$

<p>Función logística o log sigmoidea</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \frac{1}{1 + \exp(-a)}$	<p>Función tangente hiperbólica sigmoidea</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \frac{\exp(a) - \exp(-a)}{\exp(a) + \exp(-a)}$
<p>Función gaussiana</p> $y_i = g\left(\sum_{j=0}^P w_{ij}x_j\right)$ $g(a) = \exp(-a^2)$	

2.5.5 La función de salida

La función de salida, $f_i(g_i(h_i(t)))$ proporciona la salida global de la neurona i y en función de su estado de activación actual. Muy frecuentemente la función de salida es simplemente la función identidad, de modo que el estado de activación de la neurona se considera la propia salida.

2.6 Clasificación de modelos de RNA

La gran diversidad de modelos de redes neuronales existentes en la actualidad obliga en cierta medida a la realización de clasificaciones o taxonomías. De esta forma, los modelos neuronales se pueden clasificar desde una triple óptica: en función de la arquitectura (“network architecture”), en función del tipo de aprendizaje (“learning paradigm”), y de acuerdo a sus aplicaciones (Figura 2.7).

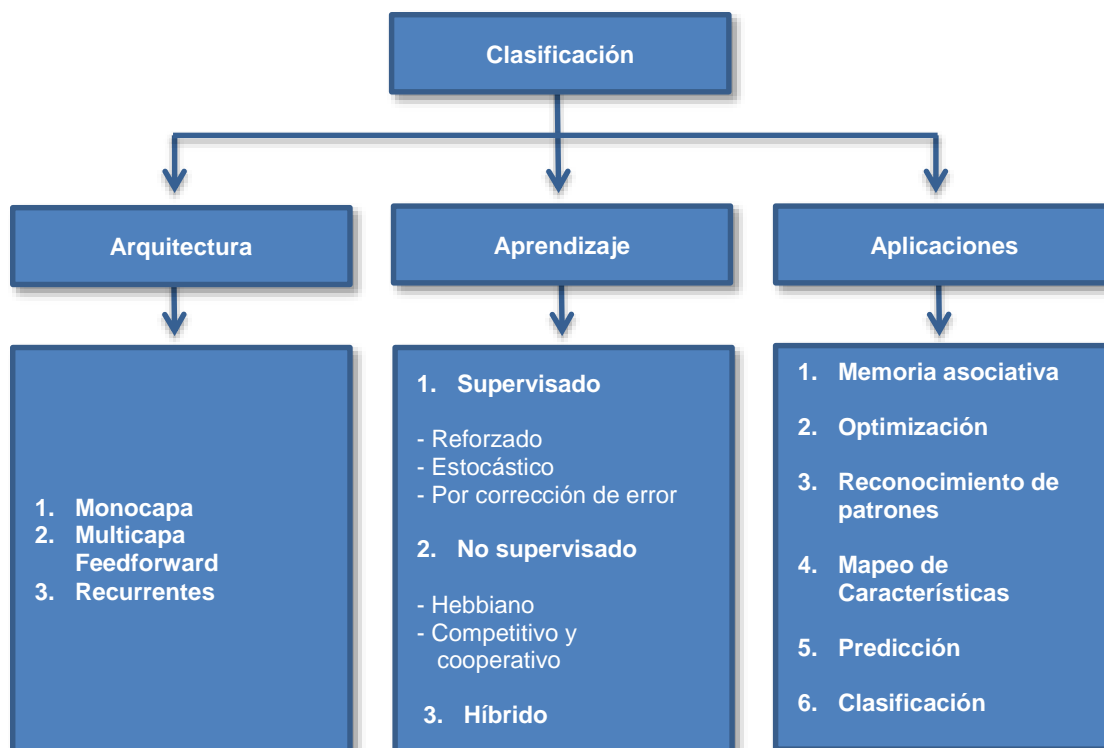


Figura 2.7. Taxonomías de acuerdo al tipo de arquitectura, aprendizaje y aplicaciones

2.6.1 Clasificación según el tipo de arquitectura

La arquitectura o topología de las RNA consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y la salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexión entre neuronas.

A partir de su ubicación dentro de la red, se pueden distinguir tres tipos de capas:

1. De entrada: Es la capa que recibe directamente la información proveniente de las fuentes externas de la red. Esta capa no procesa información, simplemente la distribuye a las demás capas.

2. Ocultas: Son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de RNA.

3. De salida: Transfieren información de la red hacia el exterior.

La conectividad entre los nodos de una RNA está relacionada con la forma en que las salidas de las neuronas están canalizadas para convertirse en entradas de otras neuronas.

Los vínculos entre las neuronas pueden ser excitatorios o inhibitorios: un peso sináptico negativo define una conexión inhibitoria, mientras que uno positivo determina una conexión excitatoria. Habitualmente, no se suele definir una conexión como de un tipo o de otro, sino que por medio del aprendizaje se obtiene un valor para el peso, que incluye signo y magnitud.

Los enlaces entre los nodos de una red se clasifican en conexiones:

1. Intercapa: Es la conexión entre nodos de distintas capas.

2. Intracapa: Cuando se conectan nodos dentro de la misma capa.

3. Autoconectadas: Se refiere a la conexión de un nodo a él mismo.

4. Supracapa: Cuando se conectan nodos de capas no adyacentes.

Una conexión de alto grado es un vínculo que combina entradas de más de un nodo. El número de entradas determina el grado de la conexión mientras que el grado de una RNA es el grado de su conexión más grande.

Si se realiza una clasificación de las redes neuronales en términos topológicos, se suele distinguir entre redes con una sola capa o nivel de neuronas y las redes

con múltiples capas (2, 3, etc.). Se asume que las RNA son de primer grado a menos que se especifique lo contrario.

2.6.1.1 Redes monocapa

En las redes monocapa, se determinan conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. También pueden existir conexiones autorrecurrentes (salida de una neurona conectada a su propia entrada).

Las redes monocapa se usan mayormente en tareas relacionadas con la autoasociación, por ejemplo, para regenerar informaciones de entrada que se presentan a la red incompleta o distorsionada.

2.6.1.2 Redes multicapa

Una red es multicapa si dispone de conjuntos de neuronas agrupados en varios niveles de capas. Normalmente, todas las neuronas de una capa reciben señales de entrada de otra capa anterior, más cercana a la entrada de la red, y envían las señales de salida a una capa posterior, más cercana a la salida de la red. A estas conexiones se les denomina conexiones hacia delante o feedforward.

En las redes feedforward no existen enlaces hacia atrás (ninguna salida de neuronas de una capa i se aplica a la entrada de neuronas de capas $i - 1$, $i - 2$, ...) y normalmente tampoco son autorrecurrentes (salida de una neurona aplicada a su propia entrada), ni laterales.

No obstante, en un gran número de estas redes también existe la posibilidad de conectar las salidas de las neuronas de capas posteriores a las entradas de las capas anteriores, a estas conexiones se les denomina conexiones hacia atrás o feedback.

Finalmente, están las redes que disponen de conexiones tanto hacia delante como hacia atrás (feedforward/feedback).

Generalmente las redes feedforward/feedback suelen ser bicapa, existiendo por tanto dos conjuntos de pesos: los correspondientes a las conexiones feedforward de la primera capa (capa de entrada) hacia la segunda capa (capa de salida) y los de las conexiones feedback de la segunda a la primera.

2.6.2 Clasificación según el tipo de aprendizaje

El término de aprendizaje en las máquinas resulta algo confuso. Existen muchas más definiciones del concepto general de aprendizaje, sin embargo, para los fines del presente trabajo, se toma en cuenta la siguiente: “La modificación del comportamiento es inducido por la interacción con el entorno y como resultado de esta experiencia, se establecen nuevos modelos de respuesta a estímulos externos”¹⁹. Esta definición fue enunciada muchos años antes de que surgieran las RNA, sin embargo puede ser aplicada también a los procesos de aprendizaje de estos sistemas.

En términos biológicos, se suele aceptar que la información memorizada en el cerebro está más relacionada con los valores sinápticos de las conexiones entre las neuronas que con ellas mismas; es decir, el conocimiento se encuentra en las sinapsis. En el caso de las RNA, se puede considerar que el conocimiento se encuentra representado en los pesos de las conexiones entre las neuronas (pesos sinápticos).

De la misma manera que el funcionamiento de una red depende del número de neuronas de las que disponga y de cómo estén conectadas entre sí, cada modelo dispone de sus propias técnicas de aprendizaje.

¹⁹ Hilera, J.R. y Martínez, V.J. (1995). Redes neuronales artificiales. Fundamentos, modelos y aplicaciones, Madrid.

En el entorno de las RNA puede definirse el aprendizaje como “el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada”. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas y en algunos modelos incluso mediante la creación o muerte neuronal (en este caso se modifica la propia arquitectura de la red)²⁰.

En cualquier caso, en un proceso de aprendizaje la información contenida en los datos de entrada queda incorporada en la propia estructura de la red.

Este tipo de situaciones, en especial la modificación de las intensidades sinápticas (plasticidad sináptica) serán las que utilicen los sistemas neuronales artificiales para llevar a cabo el aprendizaje.

Generalmente, se suelen considerar dos tipos de reglas de aprendizaje: las que responden a lo que habitualmente se conoce como aprendizaje supervisado, y las correspondientes a un aprendizaje no supervisado. Ambas modalidades pretenden estimar funciones de entrada/salida multivariante o densidades de probabilidad.

Las reglas de aprendizaje supervisadas suelen ser computacionalmente más complejas, pero también más exactas en sus resultados.

La diferencia principal entre ambos tipos consiste en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

La regla de aprendizaje es una de las características más importantes a especificar para una RNA. Con ella se determina cómo se adaptarán las conexiones de los pesos a fin de optimizar el funcionamiento de la red y cómo

²⁰ La neurona es una célula muy especial que, en general, únicamente posee capacidad para reproducirse en los primeros estados de su vida, de modo que si una neurona muere, no nacerá otra que la reemplace (aunque recientemente se han encontrado evidencias de que en ciertas situaciones sí podría reproducirse).

calcular los ajustes en los pesos durante cada ciclo. Esta regla se suspende después de que el entrenamiento se ha completado.

Cuando se elabora una RNA, se parte de un cierto modelo de neurona y de una determinada arquitectura de red, estableciéndose los pesos sinápticos iniciales como nulos o aleatorios. Para que la red pueda operar es necesario entrenarla, lo que constituye el modo de aprendizaje. El entrenamiento o aprendizaje se puede llevar a cabo en dos niveles. El más convencional es el modelado de las sinapsis, que consiste en modificar los pesos sinápticos siguiendo una cierta regla de aprendizaje, construida normalmente a partir de la optimización de una función de error o costo, que mide la eficacia actual de la operación de la red.

Si se define $w_{ij}(t)$ al peso que conecta la neurona presináptica j con la postsináptica i en la iteración t , el algoritmo de aprendizaje, en función de las señales que llegan procedentes del entorno en el instante t , proporcionará el valor $\Delta w_{ij}(t)$ que da la modificación que se debe incorporar en dicho peso, el cual quedará actualizado de la siguiente forma:

$$\Delta w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t)$$

El proceso de aprendizaje es usualmente iterativo, actualizándose los pesos de la manera anterior, una y otra vez, hasta que la red alcanza el rendimiento deseado.

Se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables, esto es:

$$(dw_{ij}/dt = 0)$$

Es importante conocer cómo se modifican los valores de los pesos; es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las

conexiones cuando se pretende que la red aprenda una nueva información. Estos criterios determinan lo que se conoce como la regla de aprendizaje de la red.

Las reglas de aprendizaje más conocidas son la regla de retropropagación (back propagation) la cual es una generalización de la regla Delta (empleadas para aprendizaje supervisado); la regla de aprendizaje Hebbiano y la regla de aprendizaje competitivo.

2.6.2.1 Aprendizaje supervisado

El aprendizaje supervisado se llama así porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor o maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada.

Este aprendizaje consiste en construir un modelo neuronal que permita estimar relaciones entre las entradas y las salidas sin la necesidad de proponer una cierta forma funcional a priori. La salida no coincidirá generalmente con lo deseado, de forma que se generará un error de salida e_i o residuo del modelo (Figura 2.8).

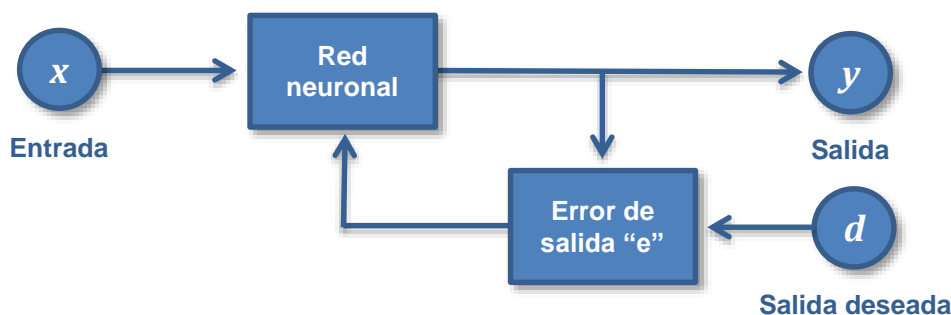


Figura 2.8. Ciclo del aprendizaje supervisado

Habitualmente se consideran tres formas de aprendizajes supervisados:

1. Aprendizaje por corrección del error: Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores

deseados y los obtenidos en la salida de la red, es decir, en función del error cometido en la salida. Ejemplos de este tipo de aprendizaje son: la regla de aprendizaje del perceptrón, utilizada en el aprendizaje de la red perceptrón diseñada por Rosenblatt en 1957; la regla delta o regla del error mínimo cuadrático medio (“Least-Mean-squared Error”(LMS)) propuesta por Widrow en 1960, utilizada en los modelos neuronales Adaline y Madaline (estos modelos mejoran el modelo de perceptrón ya que incorporan la definición de error global cometido y mecanismos para reducirlo con mayor rapidez); y la regla delta generalizada o retropropagación del error.

2. Aprendizaje reforzado: Este método emplea la información del error cometido (calculado en este caso de forma global y no para cada una de las salidas), pero sin poseer la salida deseada. Dicho aprendizaje descansa en la idea dual premio-castigo, donde se refuerza toda aquella acción que permita una mejora del modelo mediante la definición de una señal crítica (Figura 2.9).

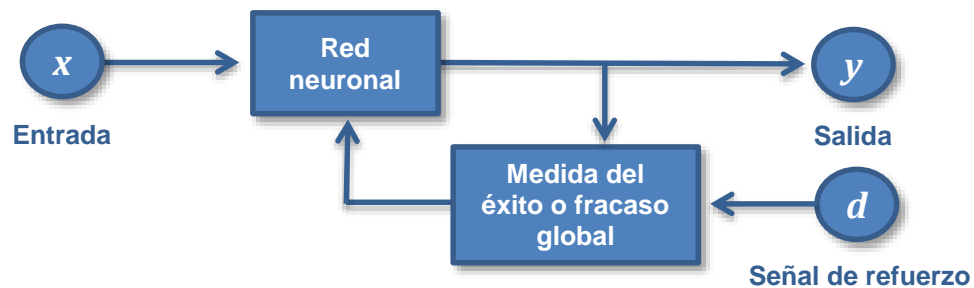


Figura 2.9. Ciclo del aprendizaje reforzado

3. Aprendizaje estocástico: Consiste en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar el efecto a partir del objetivo deseado mediante distribuciones de probabilidad.

2.6.2.2 Aprendizaje no supervisado

Las redes neuronales con aprendizaje no supervisado (también conocidos como auto supervisado o autoorganizado) no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta; por ello, suele decirse que estas redes son capaces de autoorganizarse.

Sus principales beneficios son entre otros, descubrir las regularidades presentes en los datos, extraer rasgos o agrupar patrones según su similitud, a través de la estimación de la función de densidad de probabilidad $p(x)$ que permite describir la distribución de patrones x pertenecientes al espacio de entrada R^n (Figura 2.10).

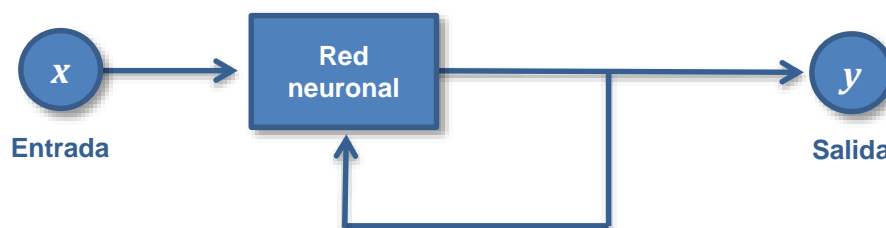


Figura 2.10. Ciclo del aprendizaje no supervisado

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presentan en su entrada. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, las cuales dependen de su estructura y del algoritmo de aprendizaje empleado.

En algunos casos, la salida representa el grado de similitud o proximidad entre la información que se le está presentando a la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso se podría realizar un agrupamiento indicando a la red a qué categoría pertenece la información

presentada a la entrada, siendo la propia red quien deba encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas.

Finalmente, lo que realizan algunas redes es un mapeo de características (feature mapping), obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa de las características de los datos de entrada, de tal forma que si se presentan a la red informaciones similares, siempre serán afectadas neuronas de salida próximas entre sí, en la misma zona del mapa.

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos: aprendizaje Hebbiano y aprendizaje competitivo y cooperativo.

1. Aprendizaje hebbiano: En el ámbito de la estrategia no supervisada, encontramos el aprendizaje Hebbiano, postulado por Hebb (1949), que consiste en el ajuste de los pesos de las conexiones de acuerdo con la correlación de los valores de las dos neuronas conectadas.

Esta forma de aprendizaje se basa en el siguiente postulado formulado por Donald O. Hebb: “Cuando un axón de una celda *A* está suficientemente cerca como para conseguir excitar una celda *B* y repetida o no persistentemente toma parte en su activación, algún proceso de crecimiento o cambio metabólico tiene lugar en una o ambas celdas, de tal forma que la eficiencia de *A*, cuando la celda *A* activa es *B*, aumenta”.

Por celda, Hebb entiende un conjunto de neuronas fuertemente conectadas a través de una estructura compleja. La eficiencia podría identificarse con la intensidad o magnitud de la conexión; es decir, con el peso.

Se puede decir, por tanto, que el aprendizaje Hebbiano consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación

(multiplicación en el caso de los valores binarios +1 y -1) de los valores de activación (salidas) de las dos neuronas conectadas:

$$\Delta w_{ij} = x_i * x_j$$

Así, si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario, cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión. Se trata de una regla de aprendizaje no supervisado, pues la modificación de los pesos se realiza en función de los estados (salidas) de las neuronas obtenidas tras la presentación de cierto estímulo (información de entrada a la red), sin tener en cuenta si se deseaba obtener o no esos estados de activación.

Como característica general de las redes no supervisadas Hebbianas puede señalarse que en ellas un número elevado de neuronas de salida pueden activarse simultáneamente. Algunos modelos utilizan reglas de aprendizaje directamente basadas en la regla de Hebb, como las redes PCA (que realizan análisis de componentes principales).

2. Aprendizaje competitivo y cooperativo: En las redes con aprendizaje competitivo y cooperativo, las neuronas compiten o cooperan unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje, se pretende que cuando se presente a la red cierta información de entrada, sólo una de las neuronas dé salida a la red, o una por cierto grupo de neuronas, se active (alcance su valor de respuesta máximo).

Por tanto, las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona vencedora (winner – take – all), quedando anuladas el resto, que son forzadas a sus valores de respuesta mínimos.

La competición entre neuronas se realiza en todas las capas de la red, existiendo en estas neuronas conexiones recurrentes de auto excitación y conexiones de

inhibición (signo negativo) por parte de neuronas vecinas. Si el aprendizaje es cooperativo, estas conexiones con las neuronas vecinas serán de excitación (signo positivo).

La finalidad de este aprendizaje es categorizar los datos que se introducen en la red. De esta forma, las informaciones similares son clasificadas formando parte de la misma categoría, y por tanto deben activar la misma neurona de salida. En este tipo de redes, cada neurona tiene asignado un peso total, que es la suma de todos los pesos de las conexiones que tiene a su entrada. El aprendizaje afecta sólo a las neuronas ganadoras (activas), redistribuyendo este peso total entre sus conexiones, sustrayendo una porción a los pesos de todas las conexiones que llegan a la neurona vencedora y repartiendo esta cantidad por igual entre todas las conexiones procedentes de unidades activas.

Por tanto, la variación del peso de una conexión entre una unidad i y otra j será nula si la neurona j no recibe excitación por parte de la neurona i (no vence en presencia de un estímulo por parte de i), y se modificará (se reforzará) si es excitada por dicha neurona i .

El aspecto geométrico de la disposición de las neuronas de una red también es la base de un caso particular de aprendizaje competitivo inducido por Kohonen en 1982, conocido como feature mapping, aplicado en redes con una disposición bidimensional de las neuronas de salida, que permiten obtener mapas topográficos o autoorganizados, en los que, de algún modo, estarían representadas las características principales de la información presentada a la red.

2.7 Red backpropagation

En 1986, Rumelhart, Hinton y Williams, formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada y las clases correspondientes, utilizando varios niveles de neuronas.

El método backpropagation (propagación del error hacia atrás), basado en la generalización de la regla delta, a pesar de sus limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales.

El funcionamiento de la red backpropagation (BPN) consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida.

A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación relativa al error total.

Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que en la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada.

La importancia de la red backpropagation consiste en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Es importante la capacidad de generalización, facilidad de dar

salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

La red debe encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento, y que pueda aplicar, además, a entradas no presentadas durante la etapa de aprendizaje para clasificarlas.

2.7.1 Estructura y aprendizaje de la red backpropagation

En una red backpropagation existe una capa de entrada con n neuronas y una capa de salida con m neuronas y al menos una capa oculta de neuronas internas. Cada neurona de una capa (excepto las de entrada) recibe entradas de todas las neuronas de la capa anterior y envía su salida a todas las neuronas de la capa posterior (excepto las de salida). No hay conexiones hacia atrás feedback ni laterales entre las neuronas de la misma capa (Figura 2.11).

La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a través de las capas hasta llegar a la capa de salida.

Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose éstos valores con la salida esperada para así obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una retropropagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera se han modificado los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del que conocíamos su valor de entrada y la salida deseada que debería generar la red ante dicho patrón.

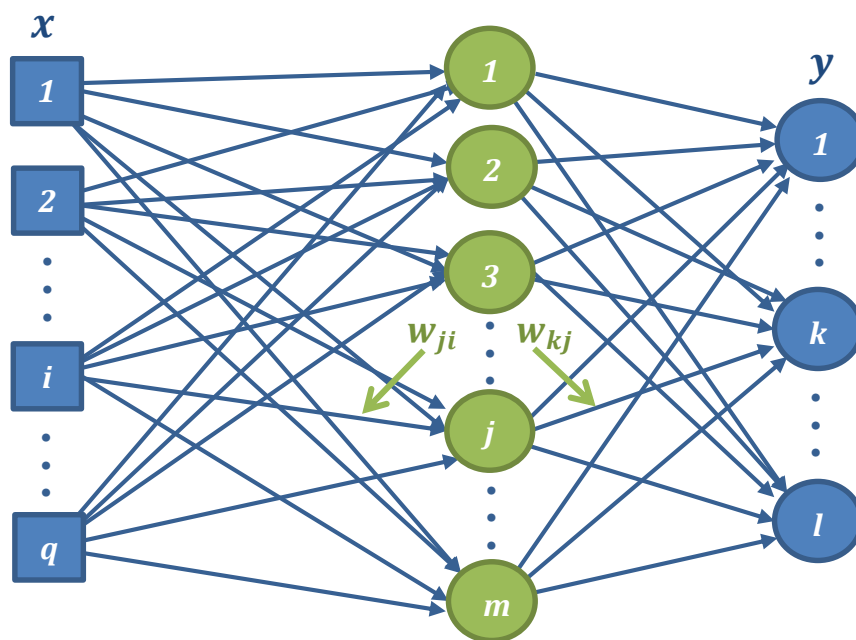


Figura 2.11. Estructura de la red backpropagation

La técnica backpropagation requiere el uso de neuronas cuya función de activación sea continua, y por lo tanto, diferenciable. Generalmente, la función utilizada será del tipo sigmoideal.

2.7.2 Algoritmo de entrenamiento de la red backpropagation

El algoritmo de entrenamiento para la red backpropagation está conformado por seis pasos, los cuales se describen a continuación:

Paso 1. Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2. Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.

Paso 3. Calcular la salida actual de la red. Para ello se presentan las entradas a la red y se calcula la salida de cada capa hasta llegar a la capa de salida.

El procedimiento anterior se detalla a continuación:

- a) Se calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.

Para una neurona j oculta:

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h$$

Donde el índice h se refiere a magnitudes de la capa oculta; el subíndice p , a los vectores de entrenamiento, y j a la j -ésima neurona oculta. El término θ puede ser opcional, pues actúa como una entrada más.

- b) Se calculan las salidas de las neuronas ocultas:

$$y_{pj} = f_j^h(net_{pj}^h)$$

- c) Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida:

$$net_{pk}^0 = \sum_{j=1}^L w_{kj}^0 y_{pj} + \theta_k^0$$

$$y_{pk} = f_k^0(net_{pk}^0)$$

Paso 4. Calcular los términos de error para todas las neuronas.

Si la neurona k es una neurona de la capa de salida, el valor de la delta es:

$$\delta_{pk}^0 = (d_{pk} - y_{pk}) f_k^0'(net_{pk}^0)$$

La función f debe ser derivable. En general disponemos de dos formas de función de salida:

La función lineal:

$$f_k(\text{net}_{jk}) = \text{net}_{jk}$$

La función sigmoïdal:

$$f_k(\text{net}_{jk}) = \frac{1}{1 + e^{-\text{net}_{jk}}}$$

La selección de la función depende de la forma que se decida representar la salida: si se desea que las neuronas de salida sean binarias, se utiliza la función sigmoïdal, en otros casos, la lineal.

Para una función lineal, tenemos: $f_k^{0'} = 1$, mientras que la derivada de una función sigmoïdal es: $f_k^{0'} = f_k^0(1 - f_k^0) = y_{pk}(1 - y_{pk})$ por lo que los términos de error para las neuronas de salida quedan:

$$\delta_{pk}^0 = (d_{pk} - y_{pk}) \text{ para la salida lineal.}$$

$$\delta_{pk}^0 = (d_{pk} - y_{pk})y_{pk}(1 - y_{pk}) \text{ para la salida sigmoïdal.}$$

Si la neurona j no es de salida, entonces la derivada parcial del error no puede ser evaluada directamente, por tanto se obtiene el desarrollo a partir de valores que son conocidos y otros que pueden ser evaluados.

La expresión obtenida en este caso es:

$$\delta_{pj}^h = f_j^{h'}(\text{net}_{pj}^h) \sum_k \delta_{pk}^0 w_{kj}^0$$

donde observamos que el error en las capas ocultas depende de todos los términos de error de la capa de salida. De aquí surge el término propagación hacia atrás.

Paso 5. Actualización de los pesos: para ello se utiliza un algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada, ajustando los pesos de las neuronas de la capa de salida:

$$w_{kj}^0(t+1) = w_{kj}^0(t) + \Delta w_{kj}^0(t+1)$$

$$\Delta w_{kj}^0(t+1) = \alpha \delta_{pk}^0 y_{pj}$$

Para los pesos de las neuronas de la capa oculta:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \Delta w_{ji}^h(t+1)$$

$$\Delta w_{ji}^h(t+1) = \alpha \delta_{pj}^h x_{pi}$$

En ambos casos, para acelerar el proceso de aprendizaje se puede añadir un término (momento).

Paso 6. El proceso se repite hasta que el término de error.

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2$$

2.7.3 Número de capas y cantidad de neuronas de la red backpropagation

No se pueden dar reglas concretas para determinar el número de neuronas o número de capas de una red para resolver un problema concreto.

Respecto al número de capas de la red, en general tres capas son suficientes (entrada, oculta y salida). Sin embargo, hay veces que un problema es fácil de resolver con más de una capa oculta.

El tamaño de las capas, tanto de entrada como de salida, suele estar determinado por la naturaleza de la aplicación. En cambio, decidir cuántas neuronas debe

tener una capa oculta no suele ser tan evidente. El número de neuronas ocultas interviene en la eficiencia de aprendizaje y de generalización de la red. No hay ninguna regla que indique el número óptimo, en cada problema se debe ensayar.

3

ANÁLISIS POR CONGLOMERADOS

En el comienzo de cualquier investigación es necesario realizar un reconocimiento o una exploración que permita mostrar a partir de un concepto o definición, una idea general de la temática de estudio. Luego se intentarán describir y expresar sus características, para posteriormente, distinguir entre tipos o modalidades y establecer clases.

Continuamente se presentan situaciones en las que es necesario establecer categorías.

Por ejemplo, esto es muy frecuente en las decisiones de compra. Cuando se desea comprar un automóvil o una vivienda, se termina categorizando vehículos según precio, kilometraje, potencia del motor, capacidad o diseño; o tipos de vivienda, viendo el precio, ubicación, infraestructura, servicios básicos, etc. Pero, eso sí, antes se debe recabar toda la información necesaria para tener suficiente conocimiento del asunto planteado.

De igual forma, toda investigación que pretenda ser científica sigue esta secuencia. Tras un mínimo de información surge la necesidad de clasificar y de identificar subconjuntos con características diferentes, esto contribuye a mejorar el conocimiento de las cosas.

3.1 Fundamentos de análisis conglomerado

El análisis conglomerado o de conglomerados, es una parte de la estadística multivariante, cuyo objetivo principal es la clasificación o agrupación de objetos en grupos homogéneos en función de las similitudes o similaridades que existen entre ellos.

Normalmente se agrupan observaciones, pero el análisis conglomerado puede también aplicarse para agrupar variables. De igual forma, estas técnicas se conocen con el nombre de *“métodos de clasificación automática o no supervisada”*, y *“reconocimiento de patrones sin supervisión”*.

El análisis de conglomerados, no hace distinción entre variables dependientes e independientes, su propósito es únicamente formar grupos a partir de un conjunto de objetos. Precisamente eso es lo que pretende la taxonomía o ciencia general de la clasificación.

La taxonomía se ocupa de establecer un sistema de clasificación de grupos conformados por objetos los más parecidos posible entre sí (homogeneidad interna) y a la vez, lo más diferentes que sea posible entre grupos (heterogeneidad entre grupos).

Se pretende que los grupos conformados con este procedimiento sean mutuamente exclusivos (que no compartan ningún elemento) y colectivamente exhaustivos (que comprendan a todos los objetos).

Esta técnica puede ser utilizada también como un método de reducción de datos. En la práctica existirán un conjunto de variables innecesarias que no explican ni persiguen los objetivos de una investigación en particular, a menos que se clasifiquen en grupos manejables.

Las principales características del análisis de conglomerados son:

- a) No existe distinción entre variables dependientes e independientes.
- b) Se pretende establecer grupos homogéneos internamente y heterogéneos entre ellos.
- c) Se pueden agrupar casos o individuos pero también variables o características, a diferencia del análisis factorial, que se centra en variables.
- d) Es una técnica exploratoria, no tiene bases para la estadística inferencial.
- e) La influencia del analista en los resultados es considerable.
- f) Las soluciones no son únicas y es necesario realizar varias pruebas para estabilizar los conglomerados finales.

En el análisis de conglomerados es especialmente importante la representatividad de la muestra y vigilar la existencia de multicolinealidad de las variables; sin embargo, no es tan crucial como en otras técnicas el cumplimiento de determinados supuestos como los de linealidad o normalidad.

Las deficiencias se superan trabajando con una muestra representativa y considerando para el análisis un conjunto de variables normalizadas o estandarizadas para evitar conclusiones erróneas.

3.2 Proceso del análisis de conglomerados

Este método de clasificación se debe considerar cuatro aspectos fundamentales:

- a) Fase de preparación
- b) Similitud entre objetos
- c) Formación de conglomerados
- d) Número de conglomerados finales

3.2.1 Fase de preparación

Dada una base de datos, en primer lugar se deben seleccionar y preparar los objetos (variables o casos) con los que se quiere trabajar.

Las conclusiones que se alcancen siempre están supeditadas a las variables con las que se han trabajado. Por tanto, las variables seleccionadas deben describir la relación entre los casos u objetos y deben ser pertinentes para las pretensiones de la investigación. Caso contrario pueden distorsionar los resultados.

Es frecuente que las variables se presenten con diferentes unidades o escalas de medida, por lo que conviene normalizarlas. Cuando vengan en las mismas unidades suele ser mejor no estandarizar, ya que una varianza mucho mayor que el resto puede ser debido precisamente a que existen dos grupos de observaciones en esa variable, que se puede ocultar al estandarizar.

Otras circunstancias poco deseables que se presentan son que las variables estén correlacionadas o incluso que su número sea excesivo, lo que tendría como consecuencia dificultar el análisis.

Por otro lado, también deben especificarse los casos u objetos a considerar, bien todos o bien sólo una parte. Los valores extraños o extremos requieren un tratamiento especial. Estos valores constituyen una verdadera excepción pero, sin embargo distorsionan la estructura, la representación gráfica y hasta la interpretación, por lo que sería aconsejable su exclusión del análisis.

3.2.2 Similitud entre objetos

Para medir la semejanza o parecido entre dos objetos se utilizan medidas de similitud o distancia; dos objetos son más parecidos cuanto más similares son (medidas de correlación), o cuanto más pequeña es la distancia entre ellos (medidas de distancia).

3.2.2.1 Medidas de correlación

El coeficiente de correlación es una medida eminentemente cuantitativa, aunque también pueden utilizarse para variables no métricas con escala nominal u ordinal; es el caso de los coeficientes de correlación de rangos de Spearman o de Kendall.

El coeficiente de correlación entre objetos está definido de la siguiente manera:

$$r_{i,j} = \frac{\text{Cov}(i,j)}{\sigma_i \sigma_j} \quad \forall i,j = 1,2, \dots, n$$
$$i \neq j$$

donde:

i: Objeto *i*

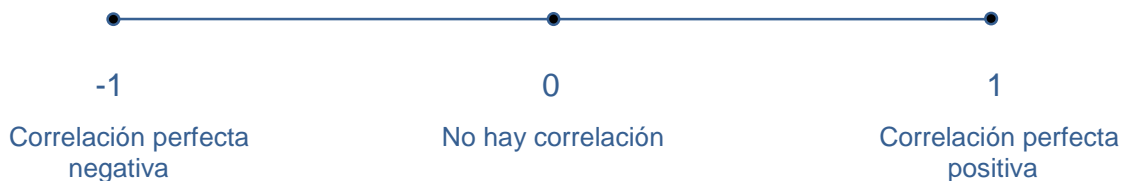
j: Objeto *j*

Cov(*i*, *j*): Covarianza entre los objetos *i* y *j*

σ_i : Desviación estándar del objeto *i*

σ_j : Desviación estándar del objeto *j*

Si r_{ij} es exactamente igual a -1 o 1, quiere decir que existe una perfecta asociación entre los dos objetos. En cambio, si $r_{ij} = 0$, significa que no hay ninguna correlación entre objetos, o de existir, no conforman una relación lineal.



Análogamente, se pueden ver los siguientes diagramas de dispersión (Ver Figura 3.1):

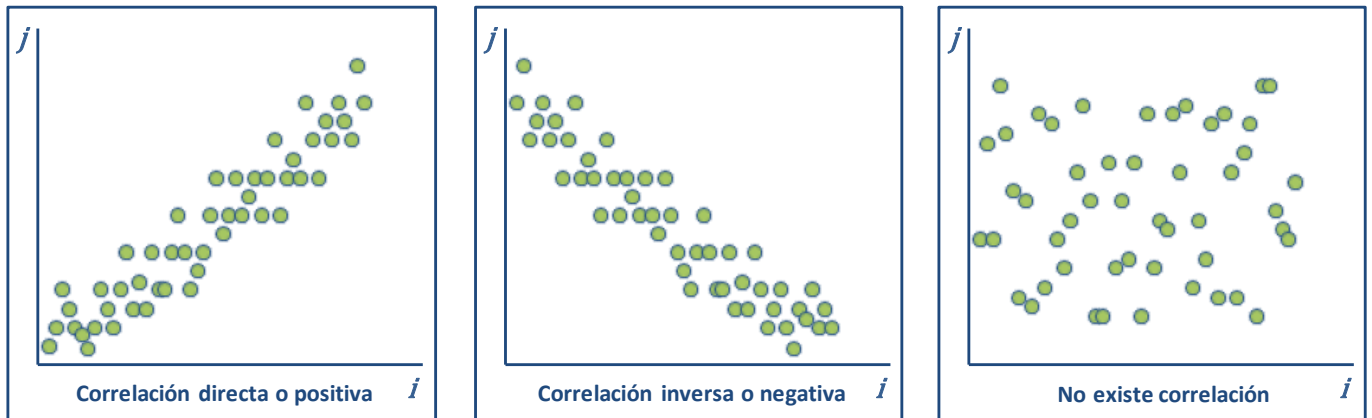


Figura 3.1. Tipos de correlación

Dos objetos o variables serán semejantes siempre y cuando su correlación sea alta y serán disímiles si la correlación entre ellos sea baja. Cabe aclarar que el coeficiente de correlación mide la forma de variación de dos objetos i , j , y no así, la magnitud de las mismas.

3.2.2.2 Medidas de distancia

La distancia entre dos objetos i , j de un mismo conglomerado, se define como una medida que cumple las siguientes propiedades:

- a) La distancia de i a j se denota como $d(i, j)$ que es un valor positivo. Si es cero, entonces i y j son iguales.

$$d(i, j) \geq 0$$

- b) La distancia de i a j es la misma que la de j a i

$$d(i, j) = d(j, i)$$

c) Si j' es un objeto que pertenece al mismo conjunto que i y j , entonces²¹:

$$d(i, j) \leq d(i, j') + d(j', j)$$

Esta medida es la más importante puesto que contrariamente a las medidas de correlación, las medidas de distancia consideran la similitud o semejanza entre las magnitudes de objetos y es una de las herramientas más utilizadas para calcular parentescos entre objetos o variables en el análisis de conglomerados.

Algunas de las medidas de distancia entre objetos i, j se presentan a continuación:

Distancia Euclídea	$d_{ij} = \sqrt{\sum_{K=1}^p (x_{iK} - x_{jK})^2}$
Distancia de Minkowski ²²	$d_{ij} = \sqrt[t]{\sum_{K=1}^p \frac{ x_{iK} - x_{jK} ^t}{S_K^t}}$
Distancia de Manhattan	$d_{ij} = \sum_{K=1}^p \frac{ x_{iK} - x_{jK} }{S_K}$
Distancia de Camberra	$d_{ij} = \sum_{K=1}^p \frac{ x_{iK} - x_{jK} }{ x_{iK} + x_{jK} }$
Distancia de Mahalanobis	$d_{ij}^2 = \sum_{K=1}^p \sum_{m=1}^p W_{Km} (x_{iK} - x_{jK}) (x_{im} - x_{jm})$ <p>W_{Km} : Elemento Km ésimo de la matriz inversa de covarianzas $K, m = 1, 2, \dots, p$ $K \neq m$</p>

²¹ La demostración del punto c) se encuentra en el Anexo 1

²² Para $t=2$, la distancia de Minkowski se transforma en la distancia Euclídeana reducida.

En relación a las demás distancias, la de Mahalanobis elimina el efecto escala, es decir, que la distancia calculada por este método no depende de las unidades de medida. La distancia de Mahalanobis no solo se ocupa de estandarizar, sino que también tiene la capacidad de tener en cuenta la forma de un objeto a partir de su correlación²³.

En el caso del análisis no jerárquico, la distancia utilizada es la euclídeana; las otras distancias son empleadas exclusivamente para el análisis conglomerado jerárquico.

3.2.3 Formación de conglomerados

Los procedimientos de agrupación o formación de conglomerados se clasifican en dos grandes grupos:

- Método jerárquico
- Método no jerárquico

3.2.3.1 Método jerárquico

Como su propio nombre lo indica, se pretende desarrollar grupos en relación a niveles o jerarquías en un proceso secuencial que se representa mediante un gráfico denominado “árbol de clasificación” o “dendograma”.

El método jerárquico divide a la población en subconjuntos de forma que entre ellos no tengan objetos comunes (sean disjuntos), y que cada subconjunto esté incluido en otro (jerarquía).

²³ La distancia de Mahalanobis se utilizó en los años treinta del siglo pasado para comparar las medidas físicas de las razas de la India.

El dendograma es la representación gráfica del proceso iterativo de agrupación jerárquica. La Figura 3.2 muestra su estructura:

- a) En la parte inferior se encuentran los n objetos iniciales. ($n = 6$).
- b) Las uniones entre objetos se forman por tres líneas rectas. Dos dirigidas a los objetos que se unen, y que son perpendiculares al eje de los objetos, y una paralela a este eje, que marca el nivel de similitud.
- c) El proceso se repite hasta que todos los objetos están conectados por líneas rectas.

Si se corta el dendograma a un nivel de similitud dado, se obtiene una clasificación del número de grupos existentes a ese nivel y los objetos que los conforman.

Por ejemplo, según la Figura 3.3 se tienen seis objetos a clasificar en relación a su jerarquía. En primera instancia se determina que los objetos más parecidos entre sí son E1 y E2 agrupados a un nivel de parentesco 2, formando de esta manera el conglomerado (E1,E2). Entonces, en una primera instancia se obtienen los grupos (E1,E2), (E3), (E4), (E5) y (E6).

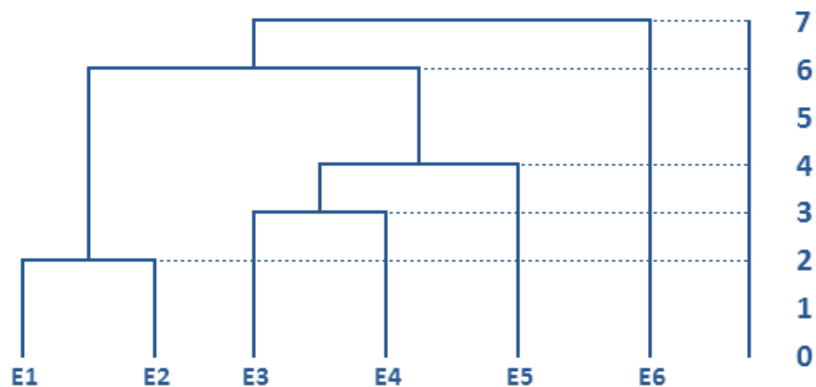


Figura 3.2. Dendograma

Este proceso continúa de manera iterativa hasta obtener un único conglomerado conformado por todos los objetos. La distancia mínima entre objetos o

conglomerados es la que determina un nuevo grupo en cada etapa. En resumen se tienen cinco iteraciones:

Nivel de parentesco	Conglomerados	Iteraciones
0	(E1), (E2), (E3), (E4), (E5), (E6)	
2	(E1,E2), (E3), (E4), (E5), (E6)	1º Iteración
3	(E1,E2), (E3,E4), (E5), (E6)	2º Iteración
4	(E1,E2), (E3,E4,E5), (E6)	3º Iteración
6	(E1,E2,E3,E4,E5), (E6)	4º Iteración
7	(E1,E2,E3,E4,E5,E6)	5º Iteración

Figura 3.3. Resumen de iteraciones

El analista es el que define el nivel máximo al que quiere llegar para obtener un número adecuado de grupos. Sin embargo, se debe tomar en cuenta que a menor nivel de parentesco, existirá mayor grado de similitud entre objetos.

Los procedimientos jerárquicos han perdido gran parte de su popularidad en relación a los métodos no jerárquicos como consecuencia del desarrollo de programas informáticos. Este problema reside en la cantidad de iteraciones necesarias según el número de observaciones u objetos con las que cuenta una base de datos en particular (comparación uno a uno). Es decir, que por fines operativos, es recomendable trabajar con poblaciones o muestras pequeñas para utilizar esta técnica.

Existen varios métodos de agrupación jerárquica, entre ellos se pueden citar los siguientes:

El desarrollar la teoría de conglomerados jerárquicos y explicar a detalle los métodos citados anteriormente no es el interés que persigue el presente trabajo, sin embargo, el método jerárquico es aplicado en el capítulo 4 como una

- a) Método del vecino más cercano o enlace simple
- b) Método del vecino más lejano o enlace completo
- c) Método del promedio entre grupos o enlace promedio
- d) Método del centroide o enlace centroide
- e) Método de Ward o enlace por mínima varianza

herramienta alternativa para determinar el número de conglomerados tomando una muestra aleatoria del total de la información para definir la cantidad de grupos finales.

3.2.3.2 Método no jerárquico

A partir de un número de objetos, n , se deben formar K grupos homogéneos u heterogéneos entre sí, siendo K un número que el analista determina *a priori*. En el caso del procedimiento jerárquico, la cantidad de grupos se decide *a posteriori*.

Como se había mencionado, el procedimiento no jerárquico requiere establecer el número de grupos con antelación. Se aconseja utilizar inicialmente algún método jerárquico para obtener un número de grupos razonable y lógico.

Para llegar a la formación de grupos se sigue un proceso iterativo de asignación. Una vez establecido el número de grupos, se selecciona el origen de cada uno y se clasifican los objetos pertenecientes a los diferentes grupos en base a distancias tomadas desde cada objeto a su correspondiente punto de origen.

El método no jerárquico más conocido, se denomina algoritmo de *K-medias* y se desarrolla a continuación.

3.2.3.2.1 Algoritmo de k-medias

Sea una muestra de n objetos con p variables. El objetivo es agrupar esta muestra en K conglomerados fijados con anticipación. El algoritmo de *k-medias* sigue tres etapas importantes:

1. Selección de los centros iniciales (centroides)
2. Asignación de objetos a los grupos
3. Optimalidad de los grupos

1. Selección de los centros iniciales:

Los centroides de los K grupos iniciales, se pueden obtener:

- Asignando aleatoriamente los objetos a los grupos y tomando los centros de los grupos así formados, tal como muestra la Figura 3.4. Sin embargo, este criterio requerirá saber si los centroides definidos son representativos dentro de cada grupo conformado.

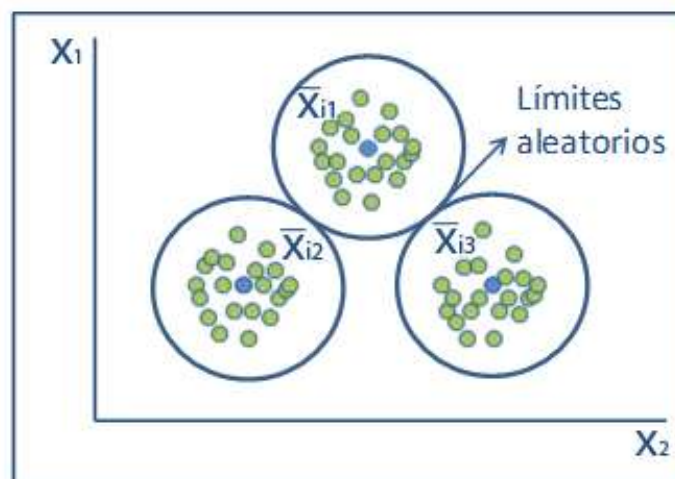


Figura 3.4. Límites aleatorios

Tomando los K puntos más alejados entre sí. (Figura 3.5).

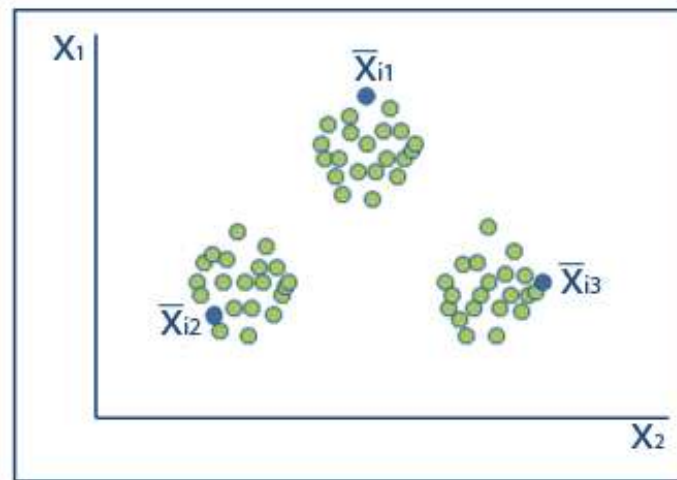


Figura 3.5. Centros más alejados entre sí

- Construyendo grupos iniciales con información *a priori* y calculando sus centros o puntos medios también *a priori*. (Figura 3.6). Esto se puede realizar seleccionando una muestra aleatoria de objetos del conjunto de datos global.

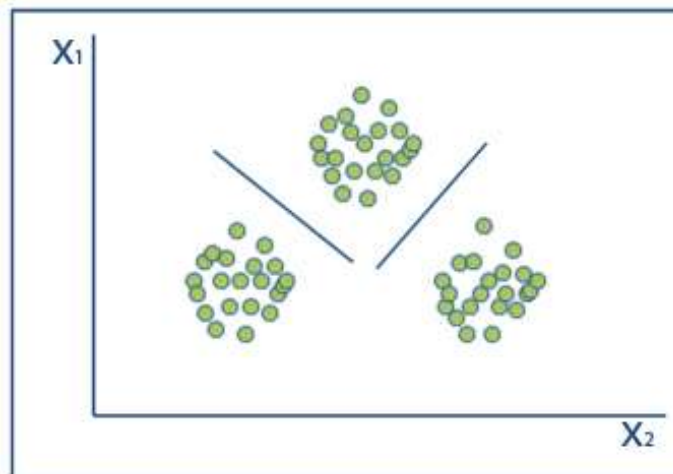


Figura 3.6. Centros con información a priori

2. Asignación de objetos a los grupos:

Para distribuir los objetos a un conglomerado determinado, se calculan las distancias euclidianas de cada elemento a los centros de los K grupos (la distancia requerida debe ser mínima), asignando cada objeto al grupo de cuyo centro \bar{x}_g esté más próximo (Figura 3.7).

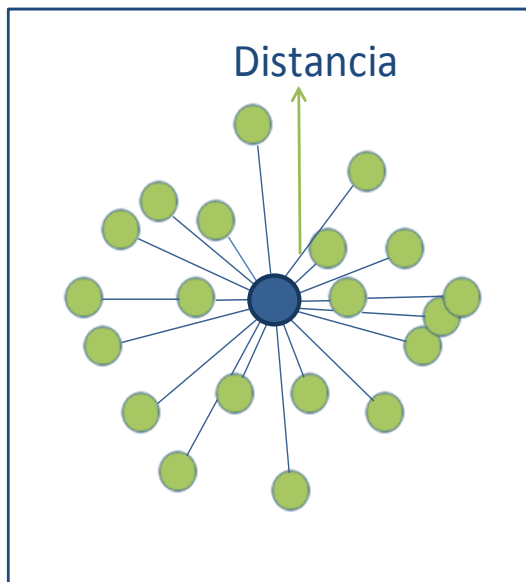


Figura 3.7. Distancia de elementos al centro de cada grupo

La asignación se realiza en forma secuencial y si se incorpora un nuevo elemento a un grupo, se deberán recalculan las coordenadas para definir un centro o centroide de grupo nuevamente.

El centroide y la distancia euclídeana están dados de la siguiente manera:

Centroide	$\bar{x}_{ig} = \frac{\sum_{j=1}^{n_g} x_{igj}}{n_g}$
Distancia Euclidiana respecto al centroide	$d(x_{igj}, \bar{x}_{ig}) = \sqrt{\sum_{i=1}^p (x_{igj} - \bar{x}_{ig})^2}$
<p>Donde:</p> <p>x_{igj}: Valor observado de la variable x_i del conglomerado g para el objeto j</p> <p>$i = 1, 2, \dots, p$</p> <p>$g = 1, 2, \dots, K$</p> <p>$j = 1, 2, \dots, n_g$</p> $n = \sum_{g=1}^K n_g$	

3. Optimalidad de los grupos:

- a) Repetir el procedimiento del análisis de conglomerados no jerárquico utilizando varias medidas de distancia para comparar la estabilidad de los grupos finales.
- b) Dividir la población en dos mitades de manera aleatoria y aplicar el método de conglomerados no jerárquicos para cada mitad. Posteriormente, comparar el grado de coincidencia entre los centroides de ambas muestras.
- c) Una forma de aproximarse a la bondad de la solución es la comparación de la varianza dentro el grupo con la varianza entre grupos.
- d) Repetir el procedimiento del análisis de conglomerados no jerárquico utilizando varias medidas de distancia para comparar la estabilidad de los grupos finales.
- e) Dividir la población en dos mitades de manera aleatoria y aplicar el método de conglomerados no jerárquicos para cada mitad. Posteriormente, comparar el grado de coincidencia entre los centroides de ambas muestras.
- f) Una forma de aproximarse a la bondad de la solución es la comparación de la varianza dentro el grupo con la varianza entre grupos.
- g) Eliminar variables aleatoriamente y realizar la agrupación de conglomerados con el conjunto reducido de variables. Los resultados obtenidos con la base reducida se compararán con los de la base completa con el fin de determinar la validez de las soluciones.
- h) Cabe aclarar que en el método no jerárquico, las soluciones pueden ser afectadas por el orden de los casos del conjunto de datos. Para evitar este problema se deben hacer corridas múltiples para distintos ordenamientos de objetos o casos hasta que la solución se estabilice.
- i) Realizar procesos de validez externa a partir de conocimientos previos. Es decir, comparar los resultados obtenidos con otros similares o de gestiones anteriores

Al definir un criterio de optimalidad se debe considerar lo siguiente:

Si utilizando estas recomendaciones no es posible mejorar la optimabilidad en los conglomerados, se aconseja concluir el proceso con los resultados obtenidos.

3.2.3.2.2 Implementación del algoritmo

El criterio de homogeneidad o de optimalidad que se utiliza en el algoritmo de *K-medias*, es minimizar la suma de cuadrados dentro de los k grupos y maximizar la suma de cuadrados entre grupos para todas las variables objeto de estudio (p variables).

La suma de cuadrados dentro de los grupos se define como:

$$SCDG = \sum_{i=1}^p \sum_{g=1}^K \sum_{j=1}^{n_g} (x_{igj} - \bar{x}_{ig})^2$$

Dónde:

x_{igj} : Valor observado de la variable x_i del conglomerado g para el objeto j

$i = 1, 2, \dots, p$

$g = 1, 2, \dots, K$

$j = 1, 2, \dots, n_g$

Como se había mencionado anteriormente, el propósito es hacer que la *SCDG* sea mínima. Escribiendo de diferente forma se tiene:

$$\min SCDG = \min \sum_{i=1}^p \sum_{g=1}^K \left[\sum_{j=1}^{n_g} \frac{(x_{igj} - \bar{x}_{ig})^2 n_g}{n_g} \right]$$

Entonces:

$$\min SCDG = \min \sum_{i=1}^p \sum_{g=1}^K S_{ig}^2 n_g$$

donde:

n_g : Número de objetos en el grupo g

S_{ig}^2 : Varianza de la variable i en el grupo g

La ecuación anterior es el equivalente a la suma ponderada de las varianzas de las variables en los grupos. Estas varianzas son claramente una medida de heterogeneidad de la clasificación, y al minimizarlas se obtendrán grupos más homogéneos.

Si por otro lado, se escribe la *SCDG* como:

$$\min SCDG = \min \underbrace{\sum_{g=1}^K \sum_{j=1}^{n_g} (x_{ijg} - \bar{x}_{ig})' (x_{ijg} - \bar{x}_{ig})}_{\text{Escalar}}$$

Entonces, como un escalar es igual a su traza:

$$\min SCDG = \min \text{tr} \left[\sum_{g=1}^K \sum_{j=1}^{n_g} (x_{ijg} - \bar{x}_{ig})' (x_{ijg} - \bar{x}_{ig}) \right]$$

Si se llama W a la suma de cuadrados dentro de los grupos:

$$W = \sum_{g=1}^K \sum_{j=1}^{n_g} (x_{ijg} - \bar{x}_{ig})' (x_{ijg} - \bar{x}_{ig})$$

Se tiene que:

$$\min tr(W) = \min SDCG$$

Esta expresión se denomina *criterio de la traza*, y fue propuesto por Ward (1963). La minimización de *SCG* requeriría calcular la *SCDG* para todas las posibles particiones, labor claramente imposible, salvo para valores de *n* muy pequeños.

Si la *SCDG* es mínima, entonces la *SCEG* (suma de cuadrados entre grupos) debe ser máxima definida de la siguiente manera:

$$\begin{aligned} SCEG &= \sum_{i=1}^p \sum_{g=1}^K \sum_{j=1}^{n_g} (\bar{x}_{ig} - \bar{x}_i)^2 \\ &= \sum_{i=1}^p \sum_{g=1}^K n_g (\bar{x}_{ig} - \bar{x}_i)^2 \end{aligned}$$

Dónde:

$$\bar{x}_i = \frac{\sum_{g=1}^K \sum_{j=1}^{n_g} x_{ijg}}{Kn_g} ; \quad \forall i = 1, \dots, p$$

La variación total *SCTG* (suma de cuadrados total de grupos) queda como:

$$SCTG = \sum_{i=1}^p \sum_{g=1}^K \sum_{j=1}^{n_g} (x_{igj} - \bar{x}_i)^2$$

$$= SCDG + SCEG$$

El algoritmo de *k-medias* busca la partición óptima con la restricción de que en cada iteración sólo se permite mover un elemento de un grupo a otro.

El algoritmo funciona de la siguiente forma:

- Paso 1.** Comenzar con una asignación inicial.
- Paso 2.** Comprobar si se reduce la $tr(W)$ al trasladar un elemento a otro grupo.
- Paso 3.** Si es posible reducir $tr(W)$ moviendo un elemento, hacerlo y recalcular las medias de los dos grupos afectados por el cambio. Posteriormente regresar al paso 2.
Si no es posible reducir la $tr(W)$ finalizar el proceso.

El resultado del algoritmo puede depender de la asignación inicial y del orden de los objetos. Conviene siempre repetir el algoritmo con distintos valores iniciales y permutando los objetos de la muestra.

El criterio de la traza no es invariante ante cambios de escala, por lo que se sugiere considerar los puntos citados en el acápite 3.2.1.

3.2.4 Número de conglomerados finales

En el método de conglomerados no jerárquicos y precisamente en el algoritmo de *k-medias*, el número de grupos es fijado con antelación. Este número no se puede estimar con un criterio de homogeneidad, ya que la forma de conseguir

grupos muy homogéneos y minimizar la *SCDG*, es hacer tantos grupos como observaciones, con lo que la *SCDG* siempre será igual a cero.

La cantidad de grupos a formar debe ser adecuada. Al seleccionar pocos conglomerados o grupos, la homogeneidad del conglomerado disminuirá. Sin embargo la técnica mejora a menor número de conglomerados elegidos.

Un test estadístico *F* de reducción de variabilidad en la *SCDG* considerando *K* grupos con las de *K+1*, y calculando la reducción relativa de variabilidad al aumentar un grupo adicional, se define como:

$$F = \frac{SCDG(K) - SCDG(K + 1)}{SCDG(K + 1)/(n - K - 1)}$$

Este valor de contraste de hipótesis se compara con una F-Fisher con *p*, *p*(*n-K-1*) grados de libertad y determinado nivel de confianza. Sin embargo, esta regla no siempre es confiable, ya que los datos no tienen por qué verificar las hipótesis necesarias para aplicar la distribución *F*.

Un criterio empírico que da resultados razonables, sugerida por Hartigan (1975), es introducir un conglomerado más, si este cociente es mayor que diez.

4

EJEMPLO DE APLICACIÓN

En la práctica es necesario evaluar diferentes alternativas cuando se desea encontrar una solución a un problema de clasificación, es decir, obtener resultados mediante distintas técnicas y con diversos escenarios a fin de probar la consistencia de las soluciones o bien, elegir la que mejor se adecue al problema planteado.

En el presente capítulo se mostrará mediante un ejemplo práctico que las redes neuronales y el análisis por conglomerados no jerárquico, pueden utilizarse como métodos de clasificación de las condiciones de vida de los hogares de Bolivia, como métodos alternativos a los tradicionales (NBI, línea de pobreza y brecha de pobreza). Además se identificarán las ventajas y desventajas de cada metodología, así como las principales similitudes y diferencias desde su implementación hasta la interpretación de los resultados generados.

En la actualidad una gran variedad de paquetes informáticos son utilizados para implementar los métodos estadísticos tradicionales, de la misma manera, existen también varios paquetes para la aplicación de redes neuronales artificiales, de hecho algunos son de distribución gratuita. Sin embargo, son pocos los que tienen integrados ambos métodos (estadísticos y redes neuronales).

Se evaluaron diferentes alternativas para elegir el software con el que se realizaría la implementación tomando en cuenta la limitada oferta de paquetes que cuentan

tanto con módulos estadísticos como redes neuronales. Se evaluaron las siguientes opciones obteniendo los siguientes resultados:

- a)** “SPSS Modeler”: Anteriormente llamado “Clementine”, contiene un módulo Base (análisis por conglomerados en sus diferentes variantes), un módulo de Clasificación (redes neuronales), un módulo de Segmentación y un módulo de Asociación.

Mediante este paquete estadístico es posible desarrollar cada uno de los métodos propuestos en el presente trabajo, sin embargo, no cuenta con versiones de prueba o versiones para estudiantes, teniendo un costo bastante elevado .

- b)** “SPSS”: cuenta con diversos métodos para el análisis por conglomerados; sin embargo, la sección de redes neuronales no incluye la Red Backpropagation (sólo incluye modelos de Función de Base Radial y perceptrón multicapa).
- c)** “MatLab”: ofrece la posibilidad (en su versión completa) de realizar redes neuronales; sin embargo, no incluye métodos para análisis por conglomerados.
- d)** “C#”: El lenguaje C# (pronunciado “C Sharp”) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET que permite crear, configurar y entrenar todo tipo de redes, sin ningún tipo de restricción.

De acuerdo a la revisión anterior, se puede concluir que ninguno de los paquetes evaluados cumple con las características deseables para poder realizar la implementación de ambos métodos en un mismo software. En este sentido, se optó realizar el ejercicio en dos de ellos (C# y SPSS), de tal forma que el desarrollo de la Red Backpropagation fue elaborado con el lenguaje de programación C# y el análisis por conglomerados con el programa estadístico SPSS.

4.1 Descripción del conjunto de datos

El conjunto de datos a utilizar contiene información de la Encuesta de Hogares del Instituto Nacional de Estadística (INE). La Encuesta de Hogares es un instrumento que tiene como objetivo suministrar información sobre las condiciones de vida de los hogares, a partir de la recopilación de datos de variables socioeconómicas y demográficas de la población boliviana.

La base de datos contiene información de 9.627.078 personas pertenecientes a 2.382.499 hogares distribuidos en los nueve departamentos del territorio nacional (Tabla 4.1.).

Tabla 4.1. HOGARES Y PERSONAS SEGÚN DEPARTAMENTO

DEPARTAMENTO	HOGARES	PERSONAS
Chuquisaca	154.194	611.540
La Paz	702.146	2.672.709
Cochabamba	400.088	1.709.849
Oruro	120.330	437.090
Potosí	191.077	772.648
Tarija	116.285	471.512
Santa Cruz	595.495	2.467.466
Beni	86.172	414.786
Pando	16.712	69.478
Total	2.382.499	9.627.078

FUENTE: Encuesta de Hogares (Elaboración propia)

El cuestionario de la Encuesta a Hogares es una boleta multitemática y consta de ocho secciones, las cuales se describen a en la Tabla 4.2.:

Tabla 4.2. SECCIONES DEL CUESTIONARIO MULTITEMÁTICO

SECCIONES	
SECCIÓN 1: Características generales del hogar y sus miembros.	PARTE A: Características sociodemográficas.
SECCIÓN 2: Migración.	PARTE A: Migración.
SECCIÓN 3: Salud.	PARTE A: EDAs e IRAs. PARTE B: Vacunas. PARTE C: Fecundidad. PARTE D: Acceso a servicios de salud.
SECCIÓN 4: Educación.	PARTE A: Formación educativa. PARTE B: Repitencia y causas de inasistencia. PARTE C: Uso individual de TICs.
SECCIÓN 5: Empleo.	PARTE A: Condición de actividad. PARTE B: Ocupación y actividad principal. PARTE C: Ingresos del trabajador asalariado. PARTE D: Ingresos del trabajador independiente. PARTE E: Ocupación y actividad secundaria. PARTE F: Ingreso laboral de la ocupación secundaria. PARTE G: Subutilización de mano de obra.
SECCIÓN 6: Ingresos no laborales del hogar.	PARTE A: Ingresos no laborales. PARTE B: Ingresos por transferencias. PARTE C: Remesas.
SECCIÓN 7: Gastos	PARTE A: Gastos en alimentos y bebidas fuera del hogar. PARTE B: Gastos en educación. PARTE C: Gastos en alimentación dentro del hogar. PARTE D: Gastos no alimentarios. PARTE E: Equipamiento de hogar.
SECCIÓN 8: Vivienda	PARTE A: Características de la vivienda. PARTE B: Acceso a TICs en hogares.

Mediante la estructura anterior, la Encuesta de Hogares captura información específica y completa de las familias bolivianas, sin embargo, no es propósito del presente trabajo utilizar todas y cada una de las variables o secciones de la encuesta.

En este sentido, y siguiendo la línea de investigación, se tomarán solamente cuatro secciones del cuestionario (características generales, salud, educación y vivienda). Estas secciones permitirán calcular el indicador de Necesidades

Básicas Insatisfechas (NBI) para clasificar las condiciones de vida de los hogares en Bolivia.

4.2 Metodología NBI

El método de Necesidades Básicas Insatisfechas concibe la pobreza como “necesidad”. Es así que se analizan las carencias de los bienes que permiten a un hogar satisfacer sus necesidades esenciales. Es un método de medición directo, puesto que observa y evalúa si un hogar cuenta o no con los bienes y servicios que le permitirán satisfacer efectivamente sus necesidades.

Se caracteriza, por identificar la pobreza de tipo estructural, debido a que las variables que lo conforman varían lentamente a lo largo del tiempo; no captura situaciones de pobreza reciente o coyuntural puesto que no incorpora variables como el ingreso o el empleo que pueden ser muy volátiles. Los componentes del indicador se describen en la Tabla 4.3.

Tabla 4.3. COMPONENTES DEL INDICADOR NBI

LISTA DE COMPONENTES		
VIVIENDA	Materiales de construcción de la vivienda	Pared
		Techo
		Piso
	Disponibilidad de espacios de la vivienda	Dormitorios por persona
		Habitaciones Multiuso por persona
		Tenencia de Cocina
INSUMOS Y SERVICIOS BÁSICOS	Saneamiento Básico	Agua
		Servicio Sanitario
	Insumos Energéticos	Energía Eléctrica
		Combustible para Cocinar
EDUCACIÓN	Asistencia Escolar	
	Años aprobados	
	Alfabetización	
SALUD	Atención Médica	

El componente de **Vivienda** estudia los materiales de construcción, cuya función es aislar al hogar del medio natural, brindando protección de factores externos como animales e insectos transmisores de enfermedades, así como de la intemperie. Por otra parte, estudia los espacios de la vivienda que deben aislar al hogar del medio social, dando privacidad y comodidad para realizar actividades biológicas y sociales.

El grupo de **Insumos y Servicios Básicos**, estudia el Saneamiento Básico y los Insumos Energéticos con los que cuenta el hogar. El saneamiento básico considera la necesidad de agua de buena calidad para la alimentación e higiene, así como la disponibilidad de un servicio sanitario que permita privacidad, salubridad e higiene. Por su parte, el estudio de Insumos Energéticos analiza los combustibles que se utilizan para cocinar y la disponibilidad de energía eléctrica.

La **Educación** es el componente que relaciona la necesidad de socialización de los individuos, además de mejorar sus oportunidades de incorporación en la vida productiva y laboral. Para ello, considera los años de escolarización de las personas en relación con su edad, la asistencia para aquellos en edad escolar, y la condición de alfabetización para todos los mayores de diez años.

La **Salud** está relacionada con las capacidades de las personas, ya que gozar de buena salud permite el desenvolvimiento adecuado dentro del medio social. En la medición se considera si la atención de la salud es adecuada, es decir, a cargo se personal calificado.

Una vez definidos los componentes, se siguen cuatro pasos para el cálculo del NBI: 1) se determinan normas mínimas para cada uno de los componentes, 2) en relación a la norma, se elabora un índice de carencia que refleje la necesidad de un hogar de cada componente, 3) se procede a agregar todos estos componentes en un sólo índice de carencia (NBI) para cada hogar, 4) se agregan todos los hogares en cinco estratos de pobreza según su valor de NBI. A continuación se describen cada uno de estos pasos.

4.2.1 Fijación de niveles normativos mínimos

La definición de las normas, de cada componente considera dos criterios:

- Establecer como norma el requerimiento mínimo para llevar una vida digna, según los derechos humanos, considerados mundialmente.
- Reconocer como nivel mínimo, la mayor frecuencia observada para la variable, siempre que esté de acuerdo con la norma establecida en el anterior criterio.

La inclusión de ambos criterios permite reflejar en la medición tanto los niveles mínimos a nivel mundial, como las características propias observadas en el país.

4.2.2 Evaluación de las carencias

Para determinar la carencia de cada componente, se evalúa si la observación del hogar es menor o mayor que la norma, en el primer caso se considera que el hogar presenta un estado de insatisfacción o carencia; de lo contrario, el hogar tiene la necesidad satisfecha.

Si se trata de una variable cualitativa (como por ejemplo, los materiales de construcción o el sistema de distribución de agua), la evaluación requiere que se asignen valores o puntajes de calificación a cada una de las alternativas de respuesta de la boleta censal. Esta calificación se realiza en función de la calidad que represente cada opción como satisfactor de una necesidad.

La cuantificación de la necesidad requiere el cálculo de los **Índices de carencia (NBI)**:

$$NBI(x) = \frac{N_x - C_x}{N_x}$$

Donde:

$NBI(x)$: Índice de carencia del componente x

N_x : Norma del componente x

C_x : Calificación del componente x en el hogar

Entonces, el indicador $NBI(x)$ se puede reescribir de la siguiente manera:

$$NBI(x) = 1 - \frac{C_x}{N_x}$$

Por fines prácticos se conocerá como **Índice de Logro (Lx)** al cociente del Logro observado entre la norma:

$$Lx = \frac{C_x}{N_x}$$

Entonces,

$$NBI(x) = 1 - Lx$$

El índice de carencia muestra el nivel o grado de insatisfacción respecto a los valores normativos. Por construcción, puede tomar los valores del rango $[-1, 1]$, donde:



Valores positivos reflejan niveles de insatisfacción y mientras más cercanos a la unidad indican una carencia mayor; por el contrario, valores negativos muestran una satisfacción por encima del nivel mínimo y mientras más tiendan a “-1”, mayor grado de satisfacción.

Una vez calculados los NBI's para cada componente (vivienda, insumos y servicios básicos, educación y salud), se agregan todas las carencias de cada hogar en un solo indicador NBI.

Posteriormente, se construyen los estratos de pobreza dividiendo el recorrido del NBI [-1,1] en cinco segmentos: necesidades básicas satisfechas, umbral de pobreza, pobreza moderada, indigencia y marginalidad.

Finalmente y como resultado del anterior proceso, se obtiene el indicador de incidencia de pobreza, que muestra porcentaje de población pobre respecto al total de población²⁴.

4.3 Análisis previo de los datos

Antes de aplicar cualquier técnica de análisis multivariante es necesario realizar un análisis previo de los datos. Es importante examinar las variables individuales y relacionarlas entre ellas, así como también solucionar los problemas de información faltante y valores atípicos (outliers).

En este sentido, y con el fin de trabajar con una base de datos sólida, se analizaron los casos atípicos y los valores faltantes mediante un proceso de consistencia y validación de datos²⁵.

El conjunto de variables de la Encuesta de Hogares necesarias para el cálculo del indicador de Necesidades Básicas Insatisfechas se presentan en la siguiente tabla:

²⁴ La metodología completa del indicador NBI se la presenta en Anexo 2.

²⁵ Los resultados del análisis para datos atípicos se encuentran en el Anexo 3.

Tabla 4.4. LISTADO DE VARIABLES DE COMPONENTES Y VARIABLES

COMPONENTE	VARIABLES
VIVIENDA	Material predominante en las paredes de la vivienda
	Paredes con revoque
	Material más utilizado en los techos
	Material más utilizado en los pisos
	Cuarto solo para cocinar
	Número de cuartos que ocupa el hogar sin contar baño y cocina
	Número de dormitorios
	Número de miembros en el hogar
AGUA Y SANEAMIENTO	Procedencia del agua para beber y cocinar
	Distribución del agua para beber y cocinar
	Tenencia de baño, water o letrina
	Uso del baño, water o letrina común o independiente
	Desagüe del baño, water o letrina
	Tenencia de energía eléctrica
	Tipo de combustible o energía utiliza para cocinar
EDUCACIÓN	Edad en años
	Relación o parentesco con el jefe de hogar
	Sabe leer y escribir
	Nivel o ciclo máximo de aprobación (Sistema actual)
	Curso o grado máximo de aprobación (Sistema actual)
	Nivel o ciclo máximo de aprobación (Sistema anterior)
	Curso o grado máximo de aprobación (Sistema anterior)
SALUD	Diarrea en las últimas cuatro semanas
	Tos en las últimas cuatro semanas
	Otra enfermedad en las últimas cuatro semanas
	Número de hijos nacidos vivos
	Lugar de atención de enfermedades
	Lugar de atención del último parto

4.3.1 Estadísticos descriptivos

Antes de analizar los valores atípicos, se calculan algunos estadísticos descriptivos que ayuden a brindar un mayor panorama de la información cuantitativa de la base de datos. La tabla siguiente muestra el detalle:

Tabla 4.5. ESTADÍSTICOS DESCRIPTIVOS

DESCRIPCIÓN	MÍNIMO	MÁXIMO	MEDIA	DESVIACIÓN ESTÁNDAR
Número de miembros en el hogar	1	17	4,0	2,1
Número de cuartos sin baño y cocina	1	11	2,4	1,5
Número de dormitorios	0	9	1,5	1,1
Edad en años	0	98	25,7	19,9
Número de hijos nacidos vivos	0	17	2,1	2,5

FUENTE: Encuesta de Hogares (Elaboración propia)

Aquellas variables que presenten mayor dispersión, no serán representadas adecuadamente por la media aritmética. Para quitar el efecto de que ejercen las unidades de medida, se utilizará el indicador NBI a fin de estandarizar los datos.

La Tabla 4.5 muestra que la variable “edad en años”, es la que posee mayor dispersión, con desviación estándar de 19,9. Dado este valor y observando la media, tanto como el rango (0 a 98 años) se puede decir que es la variable que presenta mayor heterogeneidad de todas.

4.4 Resultados obtenidos con el método NBI

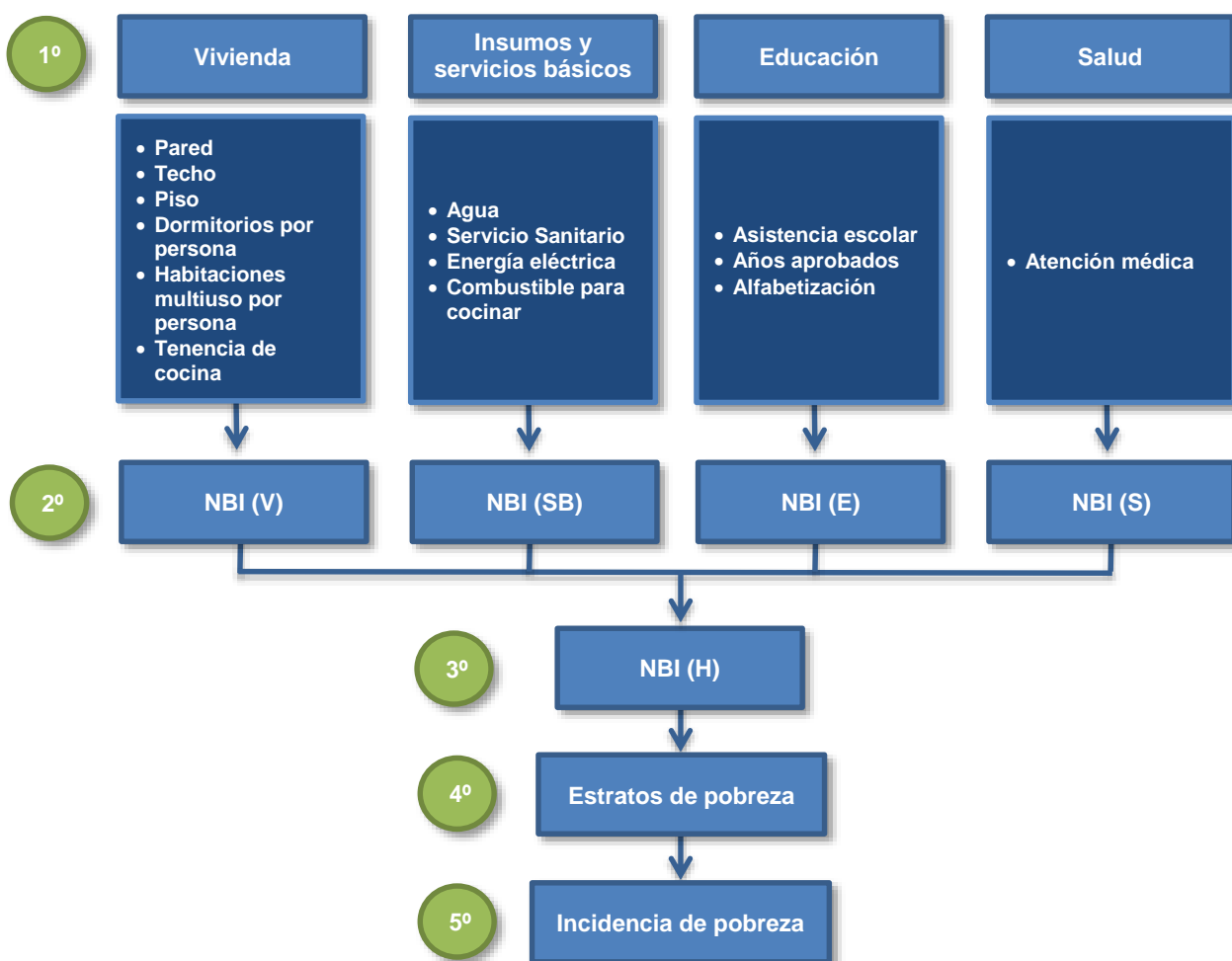


Figura 4.1. Resumen de metodología NBI

Una vez implementada la metodología del indicador de necesidades básicas insatisfechas (NBI)²⁶ sobre la base de datos de la encuesta de hogares, se calcularon además los intervalos de confianza, errores de muestreo relativos y efectos de diseño para cada componente (vivienda, servicios e insumos básicos,

²⁶ El código fuente del proceso detallado por componente se encuentra en el Anexo 4.

educación y salud), Los resultados de este procedimiento se detallan en la Tabla 4.6.

Tabla 4.6. CLASIFICACIÓN CON EL MÉTODO NBI

ESTRATO DE POBREZA	RECUESTO	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Necesidades Básicas Satisfechas	386.679	16,2%	14,9%	17,7%	4,4%	1,5
Umbral de Pobreza	537.357	22,6%	21,0%	24,2%	3,6%	1,5
Pobreza Moderada	897.983	37,7%	35,6%	39,8%	2,8%	2,0
Indigencia	505.316	21,2%	19,3%	23,2%	4,7%	2,4
Marginalidad	55.164	2,3%	1,6%	3,3%	18,6%	3,4
Total	2.382.499					

FUENTE: Encuesta de Hogares (Elaboración propia)

A nivel nacional, los resultados muestran que el 61,2% de los hogares bolivianos son pobres²⁷. De este porcentaje, el 37,7% tienen pobreza moderada, el 21,2% son indigentes y el 2,3% son hogares marginales.

Contrariamente, el 38,8% de los hogares del territorio nacional no son pobres y en referencia a este grupo, se puede decir que el 16,2% cubre todas sus necesidades básicas y el 22,6% lo hace parcialmente.

En la mayoría de los casos, los errores de muestreo relativos son menores al 5%. Asimismo se aprecia que los errores de muestreo están ligados al tamaño de los estratos de pobreza; a medida que las frecuencias van disminuyendo, se incrementará el error de muestreo. El estrato de pobreza que representa a los hogares marginales es un claro ejemplo. Sin embargo y a pesar de que el error de muestreo es considerable, presenta un intervalo de confianza pequeño.

²⁷ De acuerdo a la metodología NBI, los hogares pobres lo conforman los hogares con pobreza moderada, indigencia o marginalidad. El resto de los estratos de pobreza (necesidades básicas satisfechas y umbral de pobreza), corresponden a los hogares no pobres.

En consecuencia y según los errores relativos de muestreo, los resultados con este tipo de desagregación (estratos de pobreza) pueden presentarse como máximo a nivel nacional.

Por otro lado, se observa que el efecto de diseño utilizado en la Encuesta de Hogares es claramente superior al muestreo aleatorio simple. No obstante, cabe aclarar que el diseño muestral para la encuesta de hogares requiere un diseño complejo y no sería adecuado encararlo con un muestreo aleatorio simple.

Sabiendo que el diseño muestral de la Encuesta de Hogares no incluye en su planteamiento la desagregación por departamento, y con el objetivo analizar la posibilidad de presentar resultados a este nivel, es necesario observar los errores de muestreo para cada uno de ellos.

La Tabla 4.7 revela errores de muestreo adecuados en cada uno de los departamentos y clasificaciones de pobreza; lo que demuestra y respalda la presentación de resultados a estos niveles de desagregación.

Tabla 4.7. POBREZA SEGÚN DEPARTAMENTO - MÉTODO NBI

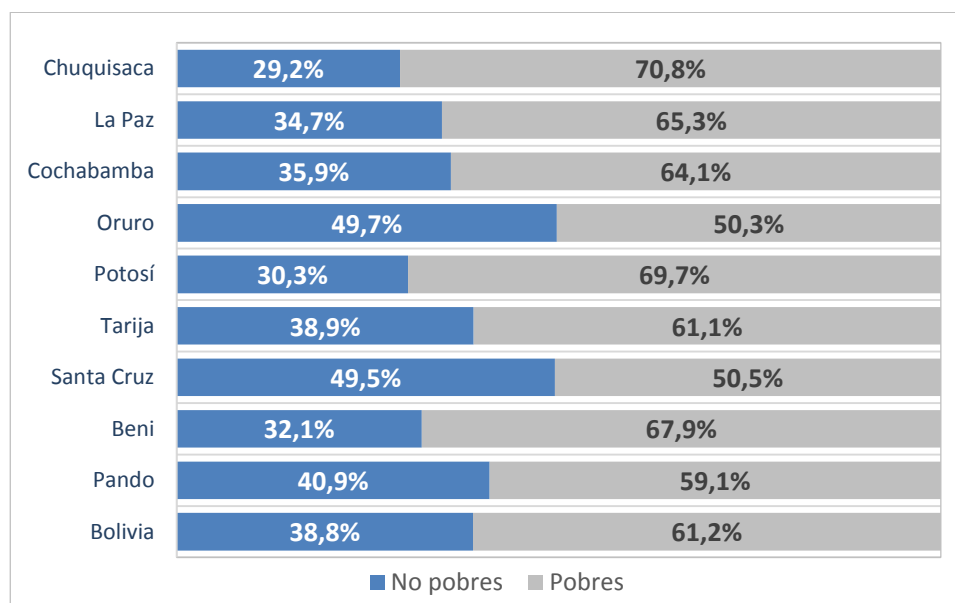
DEPARTAMENTO	POBREZA	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Bolivia	Pobres	61,2%	59,2%	63,2%	1,6%	1,7
	No pobres	38,8%	36,8%	40,8%	2,6%	1,7
Chuquisaca	Pobres	70,8%	63,7%	73,4%	3,6%	0,8
	No pobres	29,2%	26,6%	36,3%	7,9%	0,8
La Paz	Pobres	65,3%	56,3%	65,0%	3,7%	2,6
	No pobres	34,7%	35,0%	43,7%	5,7%	2,6
Cochabamba	Pobres	64,1%	55,2%	64,0%	3,8%	1,4
	No pobres	35,9%	36,0%	44,8%	5,6%	1,4
Oruro	Pobres	50,3%	38,7%	50,3%	6,6%	0,7
	No pobres	49,7%	49,7%	61,3%	5,3%	0,7
Potosí	Pobres	69,7%	62,5%	71,7%	3,5%	0,8
	No pobres	30,3%	28,3%	37,5%	7,1%	0,8

Continuación...

DEPARTAMENTO	POBREZA	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Tarija	Pobres	61,1%	49,8%	62,0%	5,6%	0,8
	No pobres	38,9%	38,0%	50,2%	7,1%	0,8
Santa Cruz	Pobres	50,5%	41,8%	50,4%	4,8%	2,0
	No pobres	49,5%	49,6%	58,2%	4,1%	2,0
Beni	Pobres	67,9%	58,9%	71,5%	4,9%	0,7
	No pobres	32,1%	28,5%	41,1%	9,3%	0,7
Pando	Pobres	59,1%	39,8%	59,8%	10,4%	0,3
	No pobres	40,9%	40,2%	60,2%	10,3%	0,3

FUENTE: Encuesta de Hogares (Elaboración propia)

De la misma manera, se evaluó la posibilidad de presentar resultados por departamento y estratos de pobreza (necesidades básicas satisfechas, umbral de pobreza, pobreza moderada, indigencia y marginalidad). Dado que el resultado del análisis reportó errores de muestreo por encima del 20,0%, no es recomendable generar datos a este nivel de desagregación.



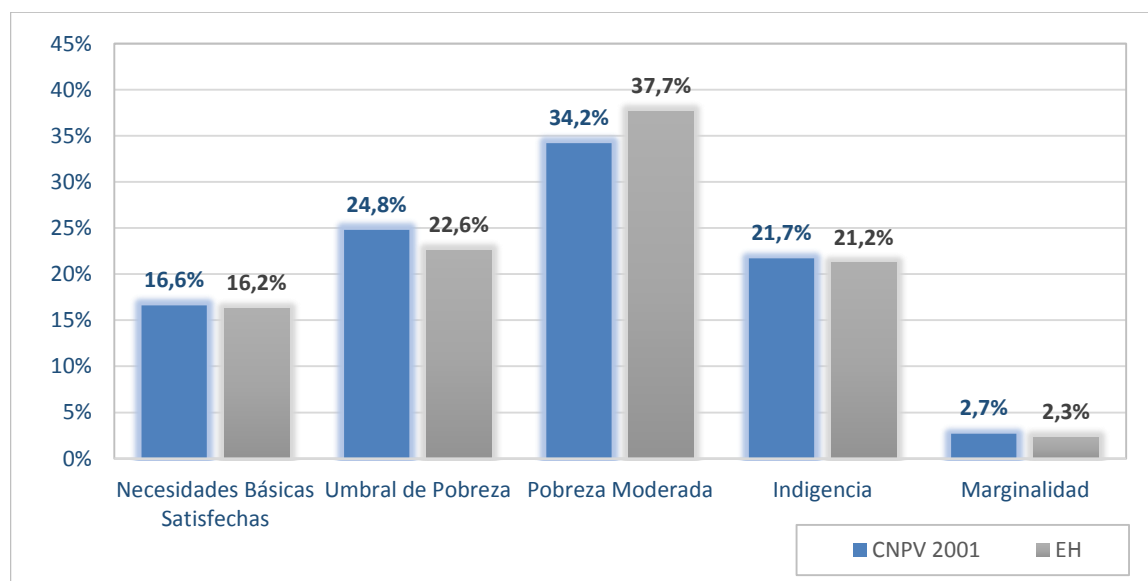
Fuente: Encuesta de Hogares. Elaboración propia.

Figura 4.2. Pobreza por departamento – Método NBI

Según la Figura 4.2., Chuquisaca es el departamento más pobre del país, con un 70,8% de hogares que no cubren sus necesidades básicas; seguido de Potosí y Beni con niveles de pobreza del 69,7% y 67,9% respectivamente.

De acuerdo a los resultados procesados en base a la metodología NBI, Oruro es el departamento que presenta la menor cantidad de hogares pobres en el territorio nacional (50,3%); seguido de Santa Cruz y Pando.

Por otro lado, a manera de contrastar la información obtenida y para fines únicamente comparativos, se presentan los resultados oficiales del CNPV 2001 y los calculados con el método NBI desarrollado en el presente trabajo. La Figura 4.3, muestra resultados y tendencias similares entre ambas fuentes de información. Lo cual permite, en alguna forma, validar los datos obtenidos con la base de la Encuesta de Hogares.



Fuente: CNPV 2001, Encuesta de Hogares. Elaboración propia.

Figura 4.3. Estratos de pobreza por fuentes de información

4.5 Resultados obtenidos con el método de conglomerados no jerárquicos

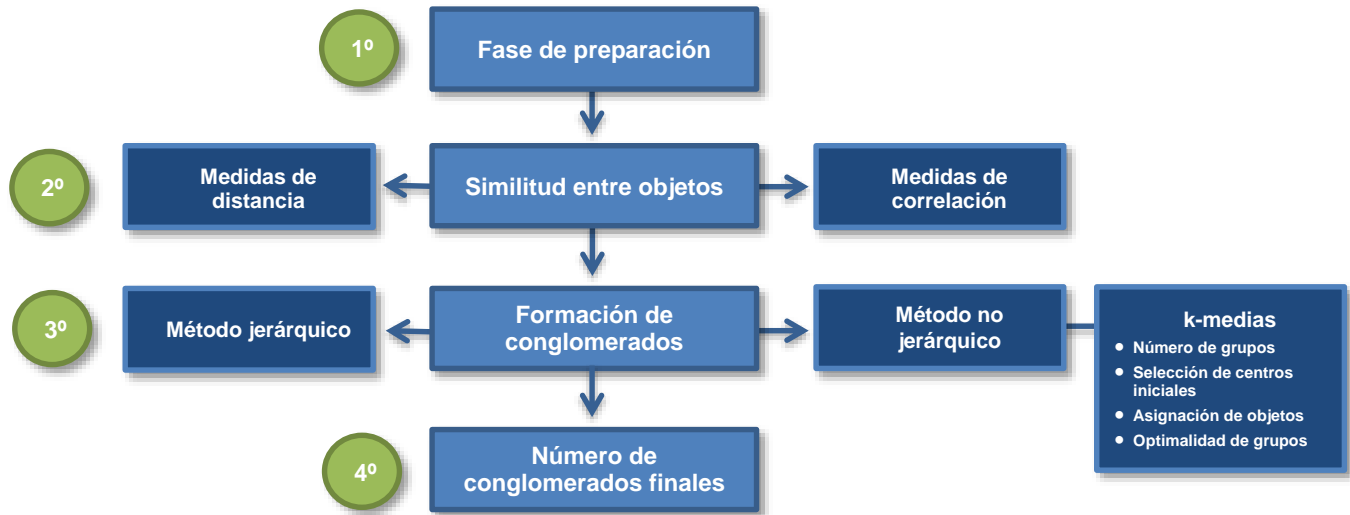


Figura 4.4. Resumen de metodología por conglomerados

Según a lo mencionado en el capítulo 3, se aplicó la técnica de k-medias para la clasificación de las condiciones de vida de los hogares de Bolivia. Este procedimiento está dividido en cuatro etapas: 1) Establecer el número de conglomerados, 2) Seleccionar los centros iniciales, 3) Asignar los objetos a los grupos y 4) Optimizar o validar los resultados.

Las variables de entrada para el análisis son los cuatro componentes del indicador NBI: El índice de carencia de vivienda (NBI(V)), el índice de carencia de servicios básicos e insumos energéticos (NBI(SB)), el índice de carencia de servicio de educación (NBI(E)) y el índice de carencia de servicio de salud (NBI(S)).

4.5.1 Número de conglomerados o grupos

Dado que en el procedimiento no jerárquico es necesario definir con antelación la cantidad de grupos en los que se clasificarán las observaciones, y con el propósito de utilizar la teoría de clasificación jerárquica citada en el capítulo 3, se utilizó la representación gráfica (dendograma) de este método como alternativa para establecer el número de conglomerados finales.

La Figura 4.5 muestra que existen cuatro o cinco grupos bien determinados de clasificación. En consecuencia, para alcanzar los objetivos del trabajo y considerando que los resultados del análisis por conglomerados deben ser comparados con los obtenidos bajo la metodología de necesidades básicas insatisfechas, se determinan cinco grupos finales, los cuales coinciden con el número de estratos de pobreza NBI (necesidades básicas satisfechas, umbral de pobreza, pobreza moderada, indigencia y marginalidad).

4.5.2 Selección de centros iniciales

La teoría para seleccionar los centros iniciales indica seguir uno de tres pasos para conseguir estos puntos de partida: 1) asignando aleatoriamente los objetos a los grupos y tomando los centros de los grupos así formados, 2) tomando los K puntos más alejados entre sí, y 3) construyendo grupos iniciales con información a priori.

En este sentido y teniendo presente como objetivo principal la comparación de resultados entre diferentes métodos, se tomará como información a priori, los puntos medios de los recorridos NBI de cada estrato de pobreza, tal como se muestra en la Tabla 4.8.

Figura 4.5 Clasificación jerárquica (Dendograma)



Tabla 4.8. CENTROS INICIALES SEGÚN ESTRATO DE POBREZA

ESTRATO DE POBREZA	RECORRIDO NBI	CENTROS INICIALES
Necesidades Básicas Satisfechas	-1,0 ≤ NBI < -0,1	-0,5
Umbral de Pobreza	-0,1 ≤ NBI ≤ 0,1	0,0
Pobreza Moderada	0,1 < NBI ≤ 0,4	0,2
Indigencia	0,4 < NBI ≤ 0,7	0,5
Marginalidad	0,7 < NBI ≤ 1,0	0,8

Fuente: Encuesta de Hogares. Elaboración propia.

4.5.3 Asignación de objetos a los grupos

Para dosificar los objetos a un conglomerado determinado, se calculan las distancias euclidianas de cada elemento a los centros de los K grupos (la distancia requerida debe ser mínima), asignando cada objeto al grupo de cuyo centro \bar{x}_{ig} esté más próximo. La asignación se realiza en forma secuencial y si se incorpora un nuevo elemento a un grupo, se deberán recalculan las coordenadas para definir un centro o centroide de grupo nuevamente.

El centroide y la distancia euclidiana definidos en el capítulo 3 son los siguientes:

Centroide	$\bar{x}_{ig} = \frac{\sum_{j=1}^{n_g} x_{igj}}{n_g}$
Distancia Euclidiana respecto al centroide	$d(x_{igj}, \bar{x}_{ig}) = \sqrt{\sum_{i=1}^p (x_{igj} - \bar{x}_{ig})^2}$

4.5.4 Optimalidad de los grupos

Los criterios de optimización utilizados en este punto, se detallan a continuación:

- 1) realizar procesos de validez externa a partir de conocimientos previos, es decir, comparar los resultados obtenidos con otros similares o de gestiones anteriores,
- 2) dividir la población en dos mitades de manera aleatoria y aplicar el método de

conglomerados no jerárquicos para cada mitad, 3) hacer corridas múltiples para distintos ordenamientos de objetos o casos hasta que la solución se estabilice.

4.5.5 Implementación del algoritmo de k-medias

Una vez seleccionados los centros de los conglomerados, cada caso es asignado al cluster de cuyo centro se encuentra más próximo y comienza un proceso de ubicación iterativa de los centros. En la primera iteración se reasignan los casos por su distancia al nuevo centro y, tras la reasignación, se vuelve a actualizar el valor del centro y así sucesivamente.

Los resultados con los primeros centros se muestran en la Tabla 4.9²⁸.

**Tabla 4.9. CLASIFICACIÓN CON EL MÉTODO DE CONGLOMERADOS
(CENTROS INICIALES)**

CONGLOMERADO	RECUENTO	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Necesidades Básicas Satisfechas	386.679	16,2%	14,9%	17,7%	4,4%	1,5
Umbral de Pobreza	537.357	22,6%	21,0%	24,2%	3,6%	1,5
Pobreza Moderada	897.983	37,7%	35,6%	39,8%	2,8%	2,0
Indigencia	505.316	21,2%	19,3%	23,2%	4,7%	2,4
Marginalidad	55.164	2,3%	1,6%	3,3%	18,6%	3,4
Total	2.382.499					

Fuente: Encuesta de Hogares. Elaboración propia.

Observando los valores de la Tabla 4.9, se puede notar diferencias considerables con las calculadas anteriormente (Tabla 4.6). En este sentido y recordando que los centros iniciales se calcularon en base a los puntos medios de los estratos de pobreza del indicador NBI, se esperaría que las diferencias entre los resultados de ambas tablas sean mínimas.

²⁸ El código fuente de la clasificación por conglomerados no jerárquicos (k-medias) se presenta en el Anexo 6.

De manera similar a la clasificación NBI, los errores relativos de muestreo permiten presentar los resultados a este nivel de desagregación.

La Tabla 4.10 ayuda a identificar rápidamente los casos mal clasificados. En este ejemplo, los valores en la diagonal principal (117.973, 537.357, 825.836, 505.316 y 55.164) representan aquellos casos que fueron clasificados de manera adecuada. Los valores de la matriz fuera de la diagonal (resaltados en color celeste) representan casos de clasificación errónea. En total se clasificaron 2.041.646 casos correctamente (85,7%), lo que deja un 14,3% de error.

Tabla 4.10. COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI

ESTRATOS DE POBREZA	CONGLOMERADOS INICIALES					
	Necesidades Básicas Satisfechas	Umbral de Pobreza	Pobreza Moderada	Indigencia	Marginalidad	Total
Necesidades Básicas Satisfechas	117.973	268.706	0	0	0	386.679
Umbral de Pobreza	0	537.357	0	0	0	537.357
Pobreza Moderada	0	72.147	825.836	0	0	897.983
Indigencia	0	0	0	505.316	0	505.316
Marginalidad	0	0	0	0	55.164	55.164
Total	117.973	878.210	825.836	505.316	55.164	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

El problema de utilizar este porcentaje como índice de acuerdo es que no tiene en cuenta la probabilidad de obtener acuerdos por azar y solamente se observan los acuerdos observados. Para determinar si la concordancia observada es superior a la que se esperaría encontrar por el azar, resulta conveniente utilizar el índice de concordancia de kappa²⁹.

²⁹ La definición y la teoría del índice de concordancia Kappa se presenta en el Anexo 7.

Tabla 4.11. ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI

Medida de acuerdo	Valor	Significación aproximada	Número de casos válidos
Kappa	0,805	0,000	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

El índice Kappa muestra un 80,5% de concordancia entre los resultados de la clasificación por conglomerados y los estratos de pobreza. Asimismo, se observa que el nivel de significación es muy pequeño, por tanto se rechaza la hipótesis de acuerdo nulo.

Para corregir los errores de clasificación es necesario modificar los puntos medios o centros de cada conglomerado. Observando nuevamente la diagonal principal de la Tabla 4.10, se ve claramente que los primeros tres conglomerados tuvieron problemas en la asignación de objetos. Por tanto son los centros de estos conglomerados los que deben recalcularse.

La Tabla 4.12 resume este procedimiento y define los nuevos centros iniciales para una mejor clasificación.

Tabla 4.12. CENTROS INICIALES Y FINALES

ESTRATOS DE POBREZA	INICIALES	FINALES
Necesidades Básicas Satisfechas	-0,55	-0,15
Umbral de Pobreza	0,00	-0,03
Pobreza Moderada	0,25	0,30
Indigencia	0,55	0,55
Marginalidad	0,85	0,85

Fuente: Encuesta de Hogares. Elaboración propia.

Una vez establecidos los nuevos centros, se generan los resultados de clasificación desglosados en la Tabla 4.13.

Tabla 4.13. CLASIFICACIÓN CON EL MÉTODO DE CONGLOMERADOS (CENTROS FINALES)

CONGLOMERADO	RECUENTO	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Necesidades Básicas Satisfechas	406.381	17,1%	15,7%	18,5%	4,3%	1,6
Umbral de Pobreza	618.661	26,0%	24,3%	27,7%	3,3%	1,6
Pobreza Moderada	852.513	35,8%	33,7%	37,9%	3,0%	2,0
Indigencia	449.780	18,9%	17,0%	20,9%	5,1%	2,5
Marginalidad	55.164	2,3%	1,6%	3,3%	18,6%	3,4
Total	2.382.499					

Fuente: Encuesta de Hogares. Elaboración propia.

De esta manera y repitiendo los pasos trabajados con los centros iniciales, se obtiene la Tabla 4.14 con los nuevos centros para validar la información y medir los avances alcanzados.

Tabla 4.14. COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS FINALES) Y ESTRATOS NBI

ESTRATOS DE POBREZA	CONGLOMERADOS FINALES					
	Necesidades Básicas Satisfechas	Umbral de Pobreza	Pobreza Moderada	Indigencia	Marginalidad	Total
Necesidades Básicas Satisfechas	386.679	0	0	0	0	386.679
Umbral de Pobreza	19.702	517.655	0	0	0	537.357
Pobreza Moderada	0	101.006	796.977	0	0	897.983
Indigencia	0	0	55.536	449.780	0	505.316
Marginalidad	0	0	0	0	55.164	55.164
Total	406.381	618.661	852.513	449.780	55.164	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

Los resultados recientes, revelan una notable mejoría en la clasificación de condiciones de vida de los hogares bolivianos. Los valores obtenidos en la diagonal principal (386.679, 517.655, 796.977, 449.780 y 55.164) representan los casos clasificados de forma correcta. Los valores que no pertenecen a la diagonal

(resaltados en color celeste) representan todos los casos clasificados erróneamente.

En resumen, de 2.382.499 casos clasificados, el 92,6% (2.206.255 casos) fue asignado adecuadamente y un 7,4% (176.244 casos) tuvo problemas en la distribución.

Sin embargo y con el objetivo de considerar los valores clasificados por el azar, se calcula el estadístico Kappa. El valor del estadístico indica que el 90,0% de los objetos clasificados por el método de conglomerados y los estratos de pobreza, tienen concordancia casi perfecta.

Tabla 4.15. ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS FINALES) Y ESTRATOS NBI

Medida de acuerdo	Valor	Significación aproximada	Número de casos válidos
Kappa	0,900	0,000	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

En base a los resultados de las Tablas 4.10. y 4.14., se puede concluir que la reasignación de los centros de los conglomerados permite obtener grupos mejor clasificados. El error de clasificación se redujo en 9,5% gracias a la readecuación de centros.

Tabla 4.16. ANÁLISIS DE MEDIAS DE LOS CONGLOMERADOS FINALES

CONGLOMERADOS	ESTADÍSTICO	NBI (Carencia de la vivienda)	NBI (Carencia de servicios básicos e insumos energéticos)	NBI (Carencia de educación por hogar)	NBI (Carencia de salud por hogar)
Necesidades Básicas Satisfechas	Media	-0,27	-0,04	-0,52	-0,10
	N	406.381	406.381	406.381	406.381
	Desv. típica	0,24	0,14	0,36	0,63
Umbral de Pobreza	Media	-0,07	0,03	-0,22	0,36
	N	618.661	618.661	618.661	618.661
	Desv. típica	0,24	0,18	0,37	0,51
Pobreza Moderada	Media	0,13	0,23	0,16	0,53
	N	852.513	852.513	852.513	852.513
	Desv. típica	0,22	0,24	0,30	0,41

Continuación...

CONGLOMERADOS	ESTADÍSTICO	NBI (Carencia de la vivienda)	NBI (Carencia de servicios básicos e insumos energéticos)	NBI (Carencia de educación por hogar)	NBI (Carencia de salud por hogar)
Indigencia	Media	0,29	0,61	0,52	0,76
	N	449.780	449.780	449.780	449.780
	Desv. típica	0,24	0,24	0,29	0,25
Marginalidad	Media	0,36	0,84	0,89	0,95
	N	55.164	55.164	55.164	55.164
	Desv. típica	0,18	0,12	0,13	0,12
Total	Media	0,05	0,22	0,03	0,43
	N	2.382.499	2.382.499	2.382.499	2.382.499
	Desv. típica	0,30	0,32	0,49	0,54

Fuente: Encuesta de Hogares. Elaboración propia.

Según el análisis de medias, es claro que la clasificación está conformada por grupos diferentes entre sí. Los dos primeros conglomerados son los considerados “no pobres” ya que satisfacen en algún porcentaje las escalas NBI de carencia.

El primer conglomerado está constituido por los hogares que satisfacen ampliamente sus necesidades y alcanzan niveles de vida adecuados (ver Tabla 4.17). En la misma línea, cabe recalcar que sobre todo los hogares con personas que gozan de buena educación y que además viven en viviendas apropiadas, pertenecen a este grupo.

El segundo grupo está conformado por hogares que cumplen con lo necesario para catalogarse como hogares no pobres (ver Tabla 4.18). Esta parte de la población es la que divide a los pobres de los no pobres (umbral de pobreza).

Al tercer conglomerado pertenecen los hogares con viviendas construidas con materiales de menor calidad, algunos servicios básicos no tienen condiciones adecuadas y además los miembros de estos hogares cuentan con un servicio de salud regular (ver Tabla 4.19).

Tabla 4.17. CARACTERÍSTICAS DEL CONGLOMERADO 1 (K-MEDIAS)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Ladrillo/bloques de cemento/hormigón
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Mosaico/baldosas/cerámica
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	3
	Número de dormitorios	2
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería dentro de la vivienda
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	Alcantarillado
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Buena
SALUD		Buena

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.18. CARACTERÍSTICAS DEL CONGLOMERADO 2 (K-MEDIAS)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Ladrillo/bloques de cemento/hormigón
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Cemento
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería dentro de la vivienda
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	Alcantarillado
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Buena
SALUD		Regular

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.19. CARACTERÍSTICAS DEL CONGLOMERADO 3 (K-MEDIAS)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Cemento
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería fuera de la vivienda, pero dentro del lote o terreno
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	A un pozo ciego
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Regular
SALUD		Regular

Fuente: Encuesta de Hogares. Elaboración propia.

El cuarto grupo lo componen los hogares con características de vivienda mínimas, son hogares que no tienen todos los servicios básicos, que además tienen un servicio de salud regular y mala educación (ver Tabla 4.20).

Por último, el grupo cinco está formado por hogares de extrema pobreza, con viviendas construidas con materiales de baja calidad y sin servicios básicos. Los miembros o las personas de este conglomerado de hogares tienen mala atención de salud y mala educación (ver Tabla 4.21).

Tabla 4.20. CARACTERÍSTICAS DEL CONGLOMERADO 4 (K-MEDIAS)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	Si
	Material más utilizado en los techos	Paja/caña/palma/barro
	Material más utilizado en los pisos	Tierra
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Río/vertiente/acequia
	Distribución del agua para beber y cocinar	No se distribuye por cañería
	Tenencia de baño, water o letrina	No
	Uso del baño, water o letrina común o independiente	No tiene baño
	Desagüe del baño, water o letrina	No tiene baño
	Tenencia de energía eléctrica	No
	Tipo de combustible o energía utiliza para cocinar	Leña
EDUCACIÓN		Regular
SALUD		Mala

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.21. CARACTERÍSTICAS DEL CONGLOMERADO 5 (K-MEDIAS)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	No
	Material más utilizado en los techos	Paja/caña/palma/barro
	Material más utilizado en los pisos	Tierra
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Río/vertiente/acequia
	Distribución del agua para beber y cocinar	No se distribuye por cañería
	Tenencia de baño, water o letrina	No
	Uso del baño, water o letrina común o independiente	No tiene baño
	Desagüe del baño, water o letrina	No tiene baño
	Tenencia de energía eléctrica	No
	Tipo de combustible o energía utiliza para cocinar	Leña
EDUCACIÓN		Mala
SALUD		Mala

Fuente: Encuesta de Hogares. Elaboración propia.

4.6 Resultados obtenidos con el método de redes neuronales BPN

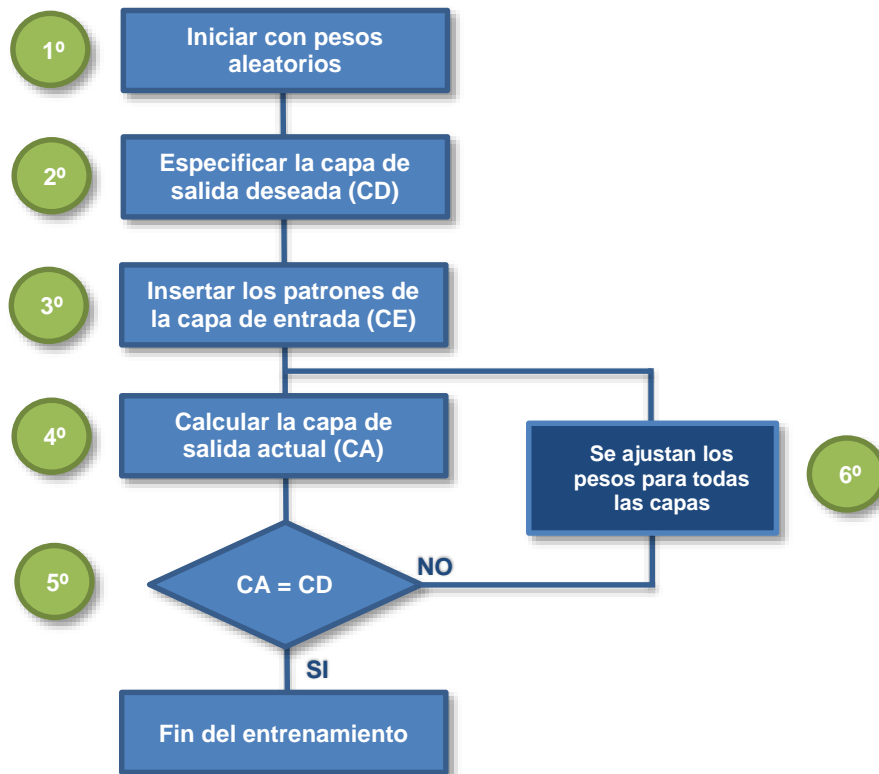


Figura 4.6. Resumen del algoritmo Backpropagation

Como se había mencionado en el Capítulo 2, el funcionamiento de la red backpropagation (BPN) consiste en el aprendizaje de un conjunto predefinido de entradas-salidas de datos.

Como en el caso de clasificación por conglomerados, los patrones de entrada de la red BPN son los componentes del indicador NBI, el índice de carencia de vivienda (NBI(V)), el índice de carencia de servicios básicos e insumos energéticos (NBI(SB)), el índice de carencia de servicio de educación (NBI(Ed)) y el índice de carencia de servicio de salud (NBI(S)).

Estos patrones de entrada se aplican como un estímulo para la primera capa de neuronas de la red, se va propagando a través de todas las capas superiores hasta generar una salida, se compara el resultado de las neuronas de salida con el resultado que se desea obtener y se calcula un valor de error para cada neurona de salida.

Los valores deseados son el resultado NBI de un conjunto reducido de datos y que a su vez, sirvieron para la fase de entrenamiento de la red BPN.

Posteriormente, los errores obtenidos fueron transmitidos hacia atrás. Esta iteración se repite, capa por capa, hasta conseguir que todas las neuronas de la red hayan recibido un error mínimo. En base al valor del error recibido, se recalculan los pesos para cada neurona, de manera que en la siguiente vez, la salida esté más cercana a la deseada.

Si las diferencias entre la capa de salida y la capa deseada son nulas, o ya no mejoran considerablemente en las últimas iteraciones, el proceso se detiene.

4.6.1 Interfaz de la red BPN

El diseñar una interfaz o un formulario para la red neuronal backpropagation permite introducir de una forma más amigable los criterios en la estructura de la red (cantidad de nodos ocultos), como también establecer el número de iteraciones deseadas, la velocidad de simulación y la selección de los datos de entrenamiento. De otra forma, la definición de estos criterios tendría que establecerse directamente en el código fuente.

El formulario de la red BPN se muestra a continuación:

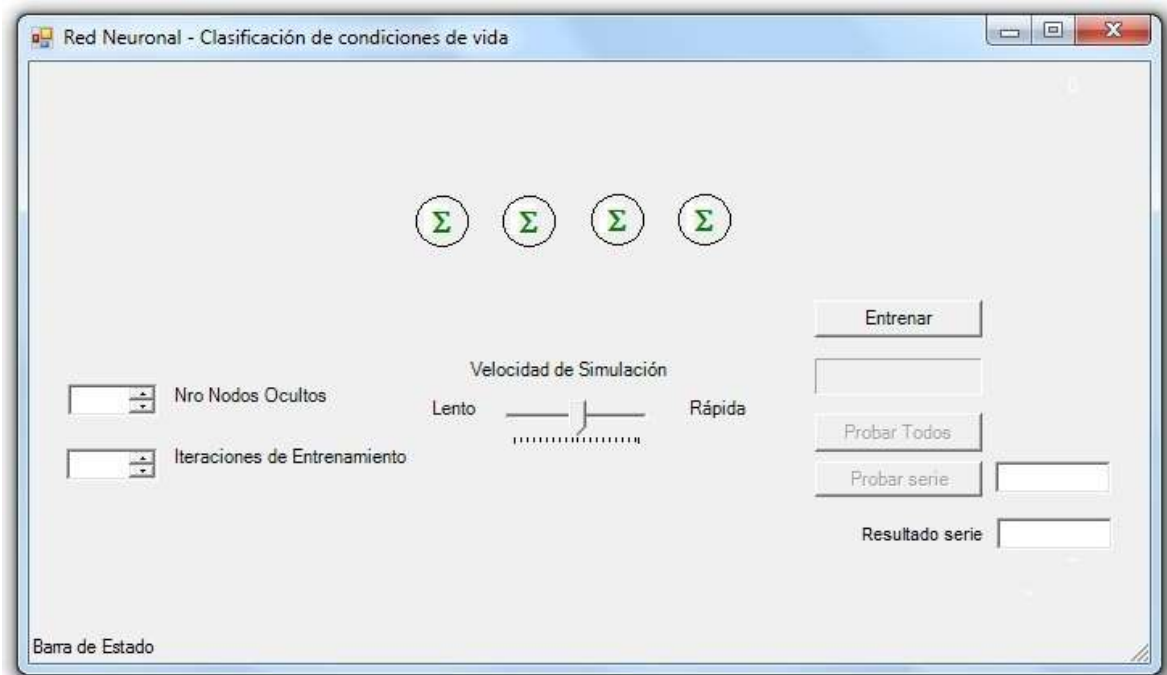


Figura 4.7. Formulario o interfaz de la red BPN

La interfaz determina por defecto cuatro neuronas de entrada, que corresponden a los componentes NBI. Permite además introducir el número de neuronas de la capa oculta y la cantidad de iteraciones. La mayoría de los autores recomiendan utilizar en la capa oculta, el doble de neuronas de la capa de entrada. Sin embargo, se debe considerar que por cada nodo oculto añadido, aumenta considerablemente el tiempo de procesamiento.

La interfaz brinda dos modalidades de entrenamiento, la primera con una base de datos externa, y la segunda con un conjunto de datos reducido (E_1, E_2, E_3, E_4), donde, E_i equivale a los componentes resultantes del indicador NBI. La última opción de entrenamiento resulta muy útil para realizar correcciones de diseño al inicio y fallas en el entrenamiento de la red.

La estructura final de la red BPN la componen tres capas: una capa de entrada conformada por cuatro neuronas, una capa oculta de ocho nodos y la capa de salida con una neurona. Luego de tres mil iteraciones, la clasificación final

convergió aproximadamente a las tres horas y media de ejecución, a una velocidad media de procesamiento (la Figura 4.8 muestra la simulación).

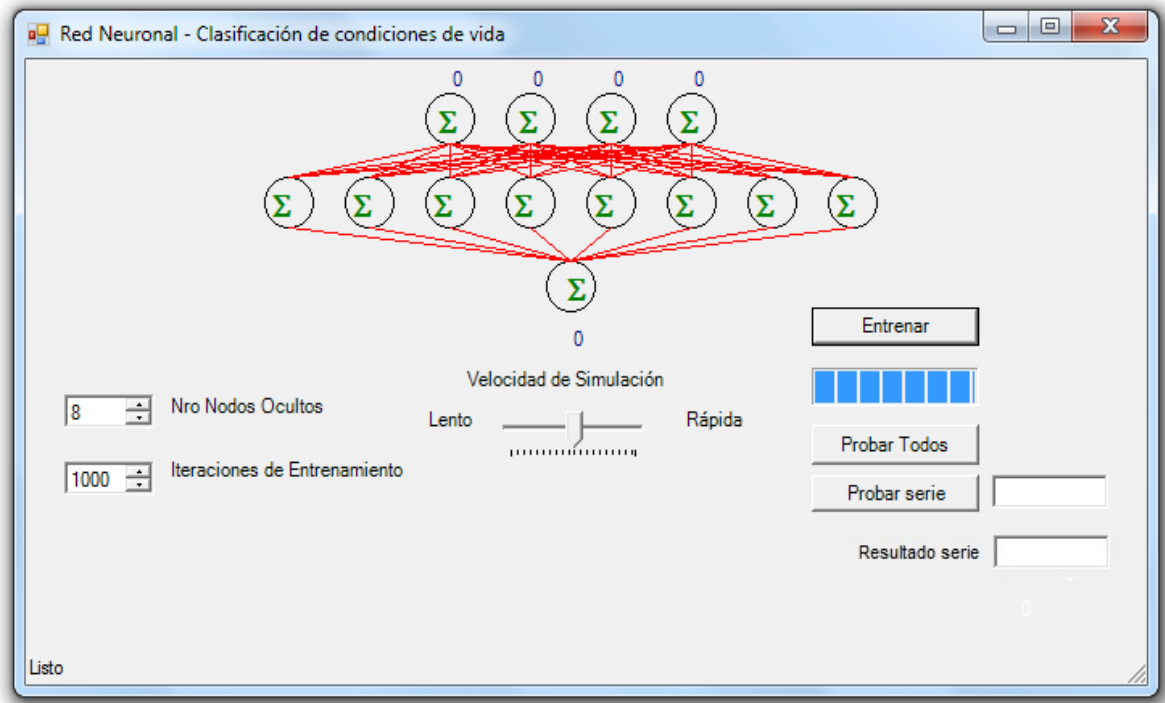


Figura 4.8. Simulación de la red BPN

Los resultados generales de la red neuronal se muestran en la Tabla 4.22 y la comparación con los estratos de pobreza, se reportan en la Tabla 4.23.³⁰

³⁰ El código fuente del diseño e implementación de la red neuronal se detallan en el Anexo 5.

Tabla 4.22. CLASIFICACIÓN CON LA RED BPN

CONGLOMERADO	RECUENTO	%	INTERVALO AL 95%		ERROR DE MUESTREO RELATIVO	EFECTO DE DISEÑO
			INFERIOR	SUPERIOR		
Necesidades Básicas Satisfechas	395.612	16,6%	15,2%	18,1%	4,4%	1,6
Umbral de Pobreza	566.140	23,8%	22,2%	25,4%	3,5%	1,5
Pobreza Moderada	884.037	37,1%	35,0%	39,2%	2,9%	2,0
Indigencia	481.546	20,2%	18,4%	22,2%	4,9%	2,5
Marginalidad	55.164	2,3%	1,6%	3,3%	18,6%	3,4
Total	2.382.499					

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.23. COMPARACIÓN DE RESULTADOS - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI

ESTRATOS DE POBREZA	CLASIFICACIÓN RED NEURONAL					
	Necesidades Básicas Satisfechas	Umbral de Pobreza	Pobreza Moderada	Indigencia	Marginalidad	Total
Necesidades Básicas Satisfechas	386.679	0	0	0	0	386.679
Umbral de Pobreza	8.933	528.424	0	0	0	537.357
Pobreza Moderada	0	37.716	860.267	0	0	897.983
Indigencia	0	0	23.770	481.546	0	505.316
Marginalidad	0	0	0	0	55.164	55.164
Total	395.612	566.140	884.037	481.546	55.164	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

Analizando los valores de la anterior tabla y observando la diagonal principal (386.679, 528.424, 860.267, 481.546 y 55.164), se tienen 2.312.080 (97,0%) objetos clasificados eficazmente. Por otro lado y en referencia a los valores fuera de la diagonal principal (8.933, 37.716 y 23.700) representan los casos mal clasificados y que no pudieron ser corregidos en el proceso. Esta cifra representa un 2,9% (70.419 casos) de error en la clasificación general.

Los porcentajes anteriores representan únicamente los valores observados, dejando de lado los valores clasificados por el azar. En este sentido, el estadístico

Kappa brinda un mejor panorama del grado de relación entre los dos métodos de clasificación. El valor del estadístico muestra un 96,0% de concordancia.

Tabla 4.24. ÍNDICE KAPPA - MÉTODOS DE CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS NBI

Medida de acuerdo	Valor	Significación aproximada	Número de casos válidos
Kappa	0,960	0,000	2.382.499

Fuente: Encuesta de Hogares. Elaboración propia.

Las características de los grupos clasificados por la red neuronal BPN se presentan desde la Tabla 4.25 a la Tabla 4.29.

De acuerdo a los resultados de las tablas citadas y revisando los resultados obtenidos con el método de conglomerados no jerárquicos (ver Tablas 4.17 a la Tabla 4.21), se puede evidenciar que ambas técnicas logran reportes idénticos.

Sin embargo, se debe considerar que el propósito específico de estas tablas, es el de ver las composiciones generales de cada grupo clasificado. Por tanto y suponiendo que la clasificación cumple con los criterios de: homogeneidad interna y heterogeneidad entre grupos; es razonable mostrar la frecuencia máxima de cada variable, como un indicador válido de las características más representativas de cada grupo.

Tabla 4.25. CARACTERÍSTICAS DEL GRUPO 1 (BPN)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Ladrillo/bloques de cemento/hormigón
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Mosaico/baldosas/cerámica
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	3
	Número de dormitorios	2
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería dentro de la vivienda
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	Alcantarillado
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Buena
SALUD		Buena

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.26. CARACTERÍSTICAS DEL GRUPO 2 (BPN)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Ladrillo/bloques de cemento/hormigón
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Cemento
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería dentro de la vivienda
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	Alcantarillado
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Buena
SALUD		Regular

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.27. CARACTERÍSTICAS DEL GRUPO 3 (BPN)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	Si
	Material más utilizado en los techos	Calamina o plancha
	Material más utilizado en los pisos	Cemento
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Cañería de red
	Distribución del agua para beber y cocinar	Por cañería fuera de la vivienda, pero dentro del lote o terreno
	Tenencia de baño, water o letrina	Si
	Uso del baño, water o letrina común o independiente	Usado solo por su hogar
	Desagüe del baño, water o letrina	A un pozo ciego
	Tenencia de energía eléctrica	Si
	Tipo de combustible o energía utiliza para cocinar	Gas licuado (garrafa)
EDUCACIÓN		Regular
SALUD		Regular

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.28. CARACTERÍSTICAS DEL GRUPO 4 (BPN)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	Si
	Material más utilizado en los techos	Paja/caña/palma/barro
	Material más utilizado en los pisos	Tierra
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Río/vertiente/acequia
	Distribución del agua para beber y cocinar	No se distribuye por cañería
	Tenencia de baño, water o letrina	No
	Uso del baño, water o letrina común o independiente	No tiene baño
	Desagüe del baño, water o letrina	No tiene baño
	Tenencia de energía eléctrica	No
	Tipo de combustible o energía utiliza para cocinar	Leña
EDUCACIÓN		Regular
SALUD		Mala

Fuente: Encuesta de Hogares. Elaboración propia.

Tabla 4.29. CARACTERÍSTICAS DEL GRUPO 5 (BPN)

COMPONENTE NBI	VARIABLE	PREDOMINANCIA
VIVIENDA	Material predominante en las paredes de la vivienda	Adobe/tapial
	Paredes con revoque	No
	Material más utilizado en los techos	Paja/caña/palma/barro
	Material más utilizado en los pisos	Tierra
	Cuarto solo para cocinar	Si
	Número de cuartos que ocupa el hogar sin contar baño y cocina	1
	Número de dormitorios	1
SERVICIOS BÁSICOS	Procedencia del agua para beber y cocinar	Río/vertiente/acequia
	Distribución del agua para beber y cocinar	No se distribuye por cañería
	Tenencia de baño, water o letrina	No
	Uso del baño, water o letrina común o independiente	No tiene baño
	Desagüe del baño, water o letrina	No tiene baño
	Tenencia de energía eléctrica	No
	Tipo de combustible o energía utiliza para cocinar	Leña
EDUCACIÓN		Mala
SALUD		Mala

Fuente: Encuesta de Hogares. Elaboración propia.

4.7 Análisis comparativo de resultados

Con el objetivo de verificar la precisión de cada método en la clasificación de las condiciones de vida de los hogares bolivianos, se comparan en primera instancia los resultados obtenidos de cada metodología y se analizan las posibles diferencias de manera general. Posteriormente se ven los resultados por estratos de pobreza y finalmente se contrastan los mismos por condición de pobreza a nivel departamental.

Este análisis termina con un resumen metodológico, en el que se contrastan los puntos más importantes de cada procedimiento.

Tabla 4.30. COMPARACIÓN DE RESULTADOS POR ESTRATO DE POBREZA, SEGÚN MÉTODO DE CLASIFICACIÓN

MÉTODO	ESTRATOS DE POBREZA	Necesidades Básicas Satisfechas	Umbral de Pobreza	Pobreza Moderada	Indigencia	Marginalidad
K-MEDIAS	Necesidades Básicas Satisfechas	386.679	0	0	0	0
	Umbral de Pobreza	19.702	517.655	0	0	0
	Pobreza Moderada	0	101.006	796.977	0	0
	Indigencia	0	0	55.536	449.780	0
	Marginalidad	0	0	0	0	55.164
TOTAL		406.381	618.661	852.513	449.780	55.164
RED BPN	Necesidades Básicas Satisfechas	386.679	0	0	0	0
	Umbral de Pobreza	8.933	528.424	0	0	0
	Pobreza Moderada	0	37.716	860.267	0	0
	Indigencia	0	0	23.770	481.546	0
	Marginalidad	0	0	0	0	55.164
TOTAL		395.612	566.140	884.037	481.546	55.164

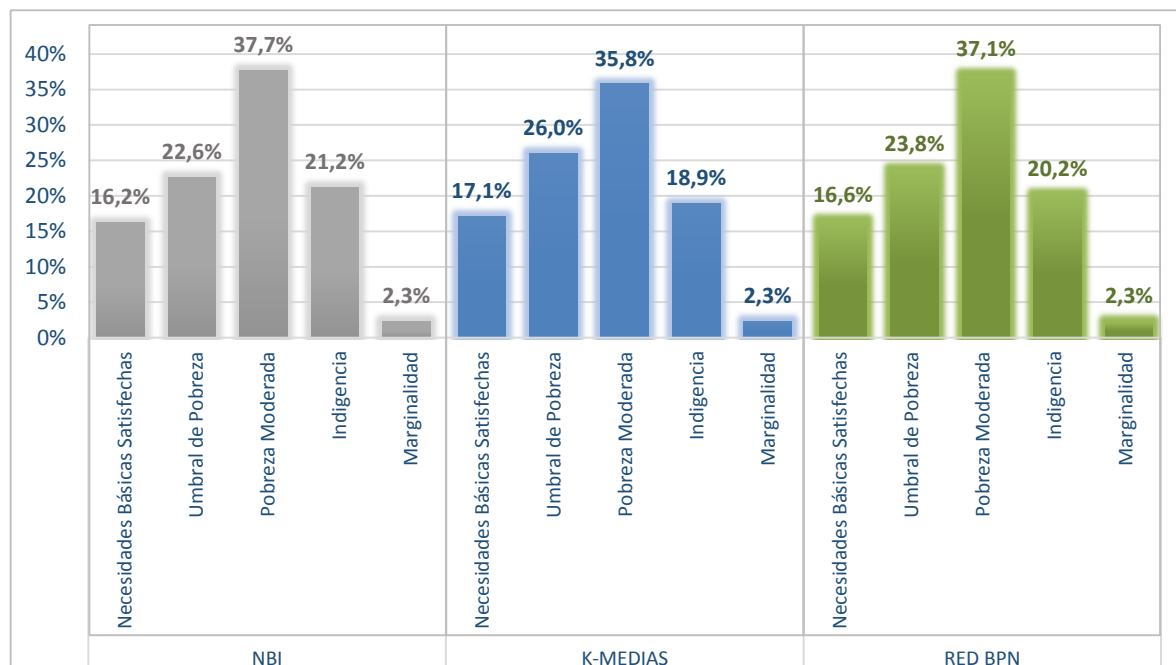
Fuente: Encuesta de Hogares. Elaboración propia.

Para cada metodología, los valores de la diagonal principal, representan los casos clasificados de forma correcta y los valores fuera de la diagonal principal (resaltados en color celeste) representan errores de clasificación.

De acuerdo al estadístico Kappa, el método de conglomerados no jerárquicos alcanza un 90,0% de objetos correctamente clasificados, mientras que la clasificación por redes neuronales muestra un 96,0% de éxito. Por lo que se puede concluir que la técnica de clasificación por redes neuronales es 6 puntos porcentuales más efectiva.

Según la Figura 4.9, la variable de control (estratos de pobreza calculados en base al indicador de necesidades básicas insatisfechas) muestra que la mayoría (37,7%) de los hogares en Bolivia tienen pobreza moderada, seguido de un 22,6% de hogares que cumplen con lo necesario para satisfacer sus necesidades

básicas. Un 21,2% está compuesto por hogares indigentes, tan solo el 16,2% del total tiene satisfacción plena y el 2,3% vive en condiciones marginales.



Fuente: Encuesta de Hogares. Elaboración propia.

Figura 4.9. Estratos de pobreza según metodología de clasificación

En relación a la variable control, los resultados del método de conglomerados (k-medias) disminuye la cantidad de hogares con pobreza moderada e indigencia y aumenta considerablemente el umbral de pobreza en más de tres puntos porcentuales. Bajo esta clasificación, existirían menos hogares pobres en el territorio nacional.

De acuerdo a la gráfica, los resultados obtenidos por la red neuronal BPN son bastante similares a los de la variable control. Lo que confirma en alguna forma, el análisis de concordancia realizado anteriormente (90,0% de correspondencia).

Por otro lado y en referencia a la Figura 4.10, se puede observar que según la clasificación NBI, el 61,0% de los hogares en Bolivia son pobres y el 39,0% no sufren de pobreza. Situación similar ocurre con los resultados de la red neuronal (40,0% no pobres y 60,0% pobres). Sin embargo la clasificación por

conglomerados muestra mayor cantidad de hogares no pobres (43,0%) y menor porcentaje de hogares pobres (57,0%).

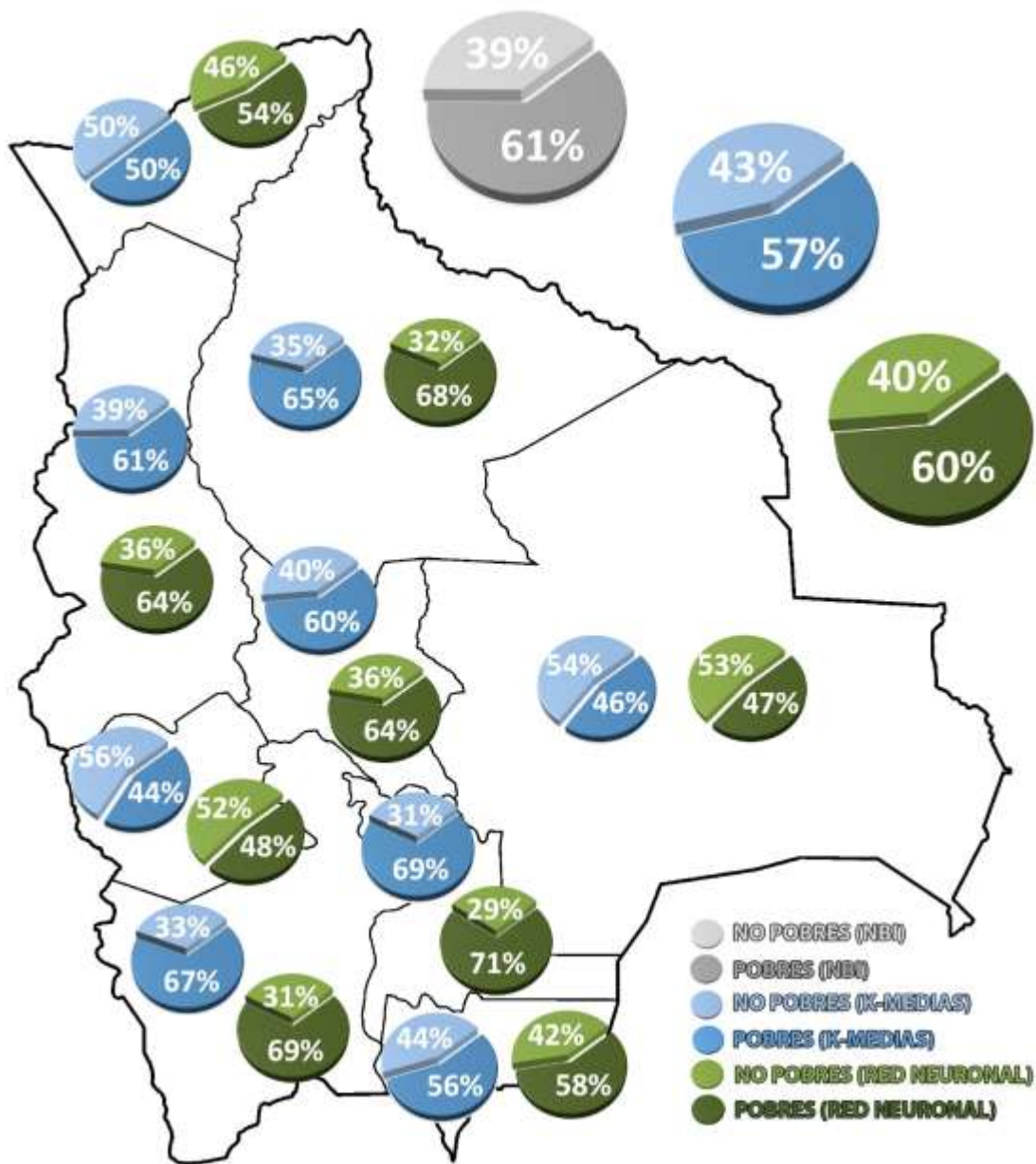


Figura 4.10. Pobreza en los hogares de Bolivia según metodología de clasificación

Las mayores diferencias de clasificación se presentan en los departamentos de Cochabamba y Pando, seguidos de Oruro, La Paz, Beni y Tarija.

Chuquisaca es el departamento más pobre del país, con un 71,0% de hogares que no cubren sus necesidades básicas; seguido de Potosí y Beni con niveles de pobreza del 69,0% y 68,0% respectivamente. Oruro es el departamento menos pobre del país, aproximadamente la mitad de sus hogares (48,0%) son no pobres.

4.8 Resumen comparativo de metodologías

En la tabla siguiente se resumen las diferencias de cada método por aspecto evaluado y se consideran básicamente los atributos más importantes de ambas metodologías.

Tabla 4.31. CUADRO COMPARATIVO DE METODOLOGÍAS UTILIZADAS

ASPECTO EVALUADO	ALGORITMO K-MEDIAS	RED BACKPROPAGATION
Conocimientos previos Recomendados	Análisis multivariante Análisis cluster Conocimientos medios de SPSS.	Análisis multivariante Análisis cluster Conocimientos medios de lenguaje C#.
Tipo de aprendizaje utilizado	Supervisado	Supervisado
Dificultad de Implementación	Baja. El método puede implementarse y modificarse desde el menú de herramientas de SPSS.	De media a alta. Se requieren niveles de programación avanzados para la red y la modificación de los parámetros manejados por defecto.
Tipos de datos con los que trabaja (continuos / discretos)	Cualquiera	Cualquiera

ASPECTO EVALUADO	ALGORITMO K-MEDIAS	RED BACKPROPAGATION
Procesamiento previo de los datos	Estandarización de las variables a incluir en el modelo (NBI).	Estandarización de las variables a incluir en el modelo (NBI). Sin embargo, en las redes neuronales no es imprescindible este procedimiento.
Especificación a priori del número de grupos	Si	Si
Dificultad de ejecución	Baja	Media a alta. Instalación de varios paquetes computacionales. Conjunto de datos pequeño. Se requirieron 3.000 iteraciones para el entrenamiento de la red.
Error de clasificación	10,0 %	4,0%
Interpretación de resultados	Sencilla	Sencilla

De lo anterior, se puede resaltar dos puntos: 1º) Los dos procedimientos necesitan conocimientos avanzados en análisis multivariante y manejo de software, y 2º) la dificultad en la implementación y ejecución de la red es considerable. Fuera de estas consideraciones, las diferencias metodológicas son similares.

5

CONCLUSIONES

En resumen, para la clasificación de las condiciones de vida de los hogares bolivianos, se utilizaron tres metodologías, estas metodologías son muy diferentes entre sí desde el punto de vista de su construcción o su estructura.

El NBI, es un indicador clásico que mide las condiciones de vida desde un punto de vista descriptivo en base a promedios. El análisis de conglomerados no jerárquico clasifica la información mediante técnicas multivariantes, y las redes neuronales realizan este proceso mediante algoritmos de aprendizaje.

Por otra parte, se aplicó la metodología NBI con el fin de estandarizar las variables y evitar los problemas en las unidades de medida. Además de lo anterior, se utilizó los estratos de pobreza como variable externa o de control, que permita supervisar de mejor forma la clasificación resultante del análisis de conglomerados y las redes neuronales.

Se evaluaron las posibles diferencias y se compararon las características de cada uno de los grupos obtenidos con cada procedimiento.

Ambas técnicas proporcionan conglomerados similares en cuando a sus características, salvo pequeñas diferencias cuantitativas.

El primer conglomerado está constituido por los hogares que satisfacen ampliamente sus necesidades y alcanzan niveles de vida adecuados. Son

hogares compuestos con personas que gozan de buena educación, salud y que además viven en viviendas apropiadas.

El segundo grupo está conformado por hogares que cumplen con lo necesario para catalogarse como hogares no pobres. Esta parte de la población es la que divide a los pobres de los no pobres (umbral de pobreza).

El tercer conglomerado presenta viviendas construidas con materiales de menor calidad, algunos servicios básicos no reúnen las condiciones adecuadas, los miembros de estos hogares cuentan con un servicio de salud regular y educación intermedia.

El cuarto grupo lo componen los hogares con características de vivienda mínimas, son hogares que no tienen todos los servicios básicos, que además tienen un servicio de salud regular y mala educación.

Finalmente, el quinto grupo lo conforman hogares con extrema pobreza, son viviendas construidas con materiales de baja calidad y sin servicios básicos. Los miembros o las personas de este conglomerado tienen mala atención de salud y educación, o simplemente no gozan de ellas.

Por otro lado y en relación a la hipótesis del trabajo, se puede concluir que los resultados obtenidos por las RNA mostraron mayor capacidad de clasificación según el análisis de concordancia de Kappa, el método por conglomerados no jerárquicos obtuvo un error global del 10,0%, mientras que, el modelo de red neuronal BPN registró una tasa de error general del 4,0%.

Si bien, lo anterior inclinaría la balanza a favor de las RNA, la dificultad en la implementación y ejecución de la red, es claramente superior al método estadístico.

6

RECOMENDACIONES

Antes de analizar las ventajas y desventajas de cada método, y así recomendar alguno de ellos; es necesario mencionar otros aspectos y consideraciones de la metodología NBI.

En referencia al indicador, no se debe perder de vista que la pobreza encierra un concepto amplio y complejo, el cual implica muchas dimensiones, ya sea insuficiencia de recursos, exclusión, desigualdad, carencia, estándares de vida, inseguridad, etc.; y que el NBI se aproxima a la pobreza sólo a través de una de sus dimensiones: la necesidad o carencia.

Los análisis de pobreza realizados a partir del NBI, deben considerar que está compuesto por variables (como el nivel educativo, la calidad de los servicios sanitarios, los materiales de la vivienda y sus espacios disponibles) que se modifican lentamente a lo largo del tiempo, por lo tanto refleja la pobreza de tipo estructural, y no así, la pobreza reciente, como lo hace el método de línea de pobreza, que se calcula en base a los ingresos de los hogares (Variable que puede tener muchas fluctuaciones en un periodo corto de tiempo).

Por otro lado y volviendo al contexto del trabajo (la clasificación mediante redes neuronales y conglomerados no jerárquicos), se puede decir que ambos métodos muestran resultados similares en la composición de cada grupo. Sin embargo existe una clara ventaja a favor de las redes neuronales por la optimización de sus resultados (4,0% de error).

La superioridad de las RNA viene acompañada de recursos temporales y computacionales considerables. Por el contrario, y con menor nivel de precisión en la clasificación (10,0%); la fácil accesibilidad, los pocos recursos informáticos y temporales, benefician el método por conglomerados no jerárquicos.

Finalmente y habiendo descrito brevemente los pros y contras de cada procedimiento, se deja a consideración del lector la elección de un método ante otro y se recomienda la utilización de ambos para un mayor enriquecimiento y complementación en el análisis de resultados.

7

ANEXOS

7.1 ANEXO 1 - Demostración desigualdad triangular

$$d(A, B) \leq d(A, C) + d(C, B) \quad \forall A, B, C \in X$$

Contextualizando la definición, se tiene que:

$$A = (x_1, y_1), \quad B = (x_2, y_2), \quad C = (x_3, y_3)$$

Por tanto, se pretende demostrar:

$$d((x_1, y_1), (x_2, y_2)) \leq d((x_1, y_1), (x_3, y_3)) + d((x_3, y_3), (x_2, y_2))$$

$$\forall A, B, C \in \mathfrak{R} \times \mathfrak{R}$$

Para demostrar lo anterior, se debe considerar las siguientes propiedades:

Propiedad 1. Si $a, b, c \in \mathfrak{R}$, entonces $|a - c| \leq |a - b| + |b - c|$

Demostración propiedad 1.

Como $|a - c| = |a - c + b - b|$

$$= |a - b + b - c|, \text{ por la desigualdad triangular, se tiene que:}$$

$$|a - c| \leq |a - b| + |b - c|$$

Propiedad 2. Si $a, b, c, d \in \mathfrak{R}$, entonces $|a - b| \leq \text{máx}\{|a - b|, |c - d|\}$

Demostración propiedad 2.

Se supone que el $\text{máx}\{|a - b|, |c - d|\} = |a - b|$ por lo que se cumple que

$$|a - b| \leq |a - b|,$$

por otra parte si el $\text{máx}\{|a - b|, |c - d|\} = |c - d|$

entonces $|a - b| \leq |c - d|$ por definición de máximo.

Volviendo a la demostración de interés, y considerando lo anterior, se debe demostrar que:

$$\text{máx}\{|x_1 - x_3|, |y_1 - y_3|\} \leq \text{máx}\{|x_1 - x_2|, |y_1 - y_2|\} + \text{máx}\{|x_2 - x_3|, |y_2 - y_3|\}$$

Se supone que:

$$\text{máx}\{|x_1 - x_3|, |y_1 - y_3|\} = |x_1 - x_3|$$

Por la propiedad 1, se tiene que:

$$|x_1 - x_3| \leq |x_1 - x_2| + |x_2 - x_3|$$

Por la propiedad 2, se sabe que:

$$|x_1 - x_2| \leq \text{máx}\{|x_1 - x_2|, |y_1 - y_2|\}$$

y que además:

$$|x_2 - x_3| \leq \text{máx}\{|x_2 - x_3|, |y_2 - y_3|\}$$

Sumando estas dos desigualdades:

$$|x_1 - x_3| \leq \text{máx}\{|x_1 - x_2|, |y_1 - y_2|\} + \text{máx}\{|x_2 - x_3|, |y_2 - y_3|\}$$

Y por lo supuesto anteriormente, se concluye que:

$$\text{máx}\{|x_1 - x_3|, |y_1 - y_3|\} \leq \text{máx}\{|x_1 - x_2|, |y_1 - y_2|\} + \text{máx}\{|x_2 - x_3|, |y_2 - y_3|\}$$

7.2 ANEXO 2 – Metodología NBI

El método de Necesidades Básicas Insatisfechas concibe la pobreza como “necesidad”. En este sentido, analiza las carencias de los bienes que permiten a un hogar satisfacer sus necesidades esenciales. Es un método de medición directo, puesto que observa y evalúa si un hogar cuenta o no con los bienes y servicios que le permitirán satisfacer efectivamente sus necesidades.

Se caracteriza, por identificar la pobreza de tipo estructural, debido a que las variables que lo conforman varían lentamente a lo largo del tiempo; no captura situaciones de pobreza reciente o coyuntural puesto que no incorpora variables como el ingreso o el empleo que pueden ser muy volátiles.

Tabla 7.1. COMPONENTES DEL INDICADOR NBI

LISTA DE COMPONENTES		
VIVIENDA	Materiales de construcción de la vivienda	Pared
		Techo
		Piso
	Disponibilidad de espacios de la vivienda	Dormitorios por persona
		Habitaciones Multiuso por persona
Tenencia de Cocina		
SERVICIOS E INSUMOS BÁSICOS	Saneamiento Básico	Agua
		Servicio Sanitario
	Insumos Energéticos	Energía Eléctrica
		Combustible para Cocinar
EDUCACIÓN	Asistencia Escolar	
	Años aprobados	
	Alfabetización	
SALUD	Atención Médica	

El componente de **vivienda** estudia los materiales de construcción, cuya función es aislar al hogar del medio natural, brindando protección de factores externos como animales e insectos transmisores de enfermedades, así como de la

intemperie. Por otra parte, estudia los espacios de la vivienda que deben aislar al hogar del medio social, dando privacidad y comodidad para realizar actividades biológicas y sociales.

El grupo de **Insumos y servicios básicos**, estudia el Saneamiento Básico y los Insumos Energéticos con los que cuenta el hogar. El saneamiento básico considera la necesidad de agua de buena calidad para la alimentación e higiene, así como la disponibilidad de un servicio sanitario que permita privacidad, salubridad e higiene. Por su parte, el estudio de Insumos Energéticos analiza los combustibles que se utilizan para cocinar y la disponibilidad de energía eléctrica.

La **Educación** es el componente que relaciona la necesidad de socialización de los individuos, además de mejorar sus oportunidades de incorporación en la vida productiva y laboral. Para ello, considera los años de escolarización de las personas en relación con su edad, la asistencia para aquellos en edad escolar, y la condición de alfabetización para todos los mayores de diez años.

La **Salud** está relacionada con las capacidades de las personas, ya que gozar de buena salud permite el desenvolvimiento adecuado dentro del medio social. En la medición se considera si la atención de la salud es adecuada, es decir, a cargo se personal calificado.

Una vez definidos los componentes, se siguen cuatro pasos para el cálculo del NBI: 1) se determinan normas mínimas para cada uno de los componente, 2) en relación a la norma, se elabora un índice de carencia que refleje la necesidad de un hogar de cada componente, 3) se procede a agregar todos estos componentes en un sólo índice de carencia (NBI) para cada hogar. 4) se agregan todos los hogares en cinco estratos de pobreza según su valor de NBI. A continuación se describen cada uno de estos pasos.

7.2.1 Fijación de niveles normativos mínimos

La definición de las normas, de cada componente considera dos criterios:

- Establecer como norma el requerimiento mínimo para llevar una vida digna, según los derechos humanos, considerados mundialmente.
- Reconocer como nivel mínimo, la mayor frecuencia observada para la variable, siempre que esté de acuerdo con la norma establecida en el anterior criterio.

La inclusión de ambos criterios permite reflejar en la medición tanto los niveles mínimos a nivel mundial, como las características propias observadas en el país.

7.2.2 Evaluación de las carencias

Para determinar la carencia de cada componente, se evalúa si la observación del hogar es menor o mayor que la norma, en el primer caso se considera que el hogar presenta un estado de insatisfacción o carencia; de lo contrario, el hogar tiene la necesidad satisfecha.

Si se trata de una variable cualitativa (como por ejemplo, los materiales de construcción o el sistema de distribución de agua), la evaluación requiere que se asignen valores o puntajes de calificación a cada una de las alternativas de respuesta de la boleta censal. Esta calificación se realiza en función de la calidad que represente cada opción como satisfactor de una necesidad.

La cuantificación de la necesidad requiere el cálculo de los **Índices de carencia (NBI)**:

$$NBI(x) = \frac{N_x - C_x}{N_x}$$

Donde:

$NBI(x)$: Índice de carencia del componente x

N_x : Norma del componente x

C_x : Calificación del componente x en el hogar

El $nbi(x)$ se puede reescribir de la siguiente manera:

$$NBI(x) = 1 - \frac{C_x}{N_x}$$

Por fines prácticos se conocerá como **Índice de logro (Lx)** al cociente del Logro Observado entre la Norma:

$$Lx = \frac{C_x}{N_x}$$

Entonces,

$$NBI(x) = 1 - Lx$$

El índice de carencia muestra el nivel o grado de insatisfacción respecto a los valores normativos. Por construcción, puede tomar los valores del rango $[-1, 1]$, donde:



Valores positivos reflejan niveles de insatisfacción y mientras más cercanos a la unidad indican una carencia mayor; por el contrario, valores negativos muestran

una satisfacción por encima del nivel mínimo y mientras más tiendan a –1 mayor grado de satisfacción.

El índice de carencia debe ser contabilizado para cada variable del NBI. A continuación se describen los cálculos de este índice y las calificaciones asignadas (en caso de tratarse de variables cualitativas) para cada componente del NBI.

7.2.2.1 Vivienda

7.2.2.1.1 Materiales de la vivienda

Este indicador está calculado en base a tres componentes que son las paredes, los pisos y los techos:

a) Paredes de la vivienda

Se considera como norma el adobe revocado. Al tratarse de una variable cualitativa, se asignan calificaciones a los materiales más utilizados en las paredes de la vivienda, las cuales se detallan en el siguiente cuadro:

Tabla 7.2. CALIFICACIÓN DE MATERIALES DE PAREDES

LISTADO DE MATERIALES	
Ladrillo, bloques de cemento, hormigón	1.5
Adobe o tapial sin revocar	0.5
Adobe o tapial con revoque ³¹	1
Tabique, quinche sin revocar	0.5
Tabique, quinche con revoque	1
Piedra	0.5
Madera	1
Caña, palma, tronco	0.5
Otro	0

³¹ Con propósito didáctico y en relación a todas las tablas de puntuación NBI, se tomará como referencia de "norma" a las celdas pintadas de color plomo.

b) Techos de la vivienda

La norma empleada para el material de techos es la calamina y la plancha. Sus valores de calificación se presentan en el siguiente cuadro:

Tabla 7.3. CALIFICACIÓN DE MATERIALES DE TECHOS

LISTADO DE MATERIALES	
Calamina o plancha	1
Teja (cemento, arcilla, fibrocemento)	1.5
Losa de hormigón armado	1.5
Paja, caña, palma, barro	0.5
Otro	0

c) Pisos de la vivienda

La norma especificada corresponde al piso de cemento. Los materiales de construcción de pisos y sus respectivas calificaciones se detallan en el siguiente cuadro.

Tabla 7.4. CALIFICACIÓN DE MATERIALES DE PISOS

LISTADO DE MATERIALES	
Tierra	0
Tablón de madera	2
Machimbre, parquet	2
Alfombra, tapizón	2
Cemento	1
Mosaico, baldosa, cerámica	2
Ladrillo	1
Otro	0

Con estas calificaciones, se calcula el Índice de carencia, para paredes, pisos y techos. **La carencia de materiales de la vivienda (NBI(M))** es:

$$NBI(M) = \frac{NBI(pa) + NBI(te) + NBI(pi)}{3}$$

Donde:

NBI(pa): Índice de carencia de pared

NBI(pa): Índice de carencia de techo

NBI(pi): Índice de carencia de piso

7.2.2.1.2 Espacios disponibles de la vivienda

En 1992 el tamaño promedio del hogar particular en Bolivia era de cinco personas, por esta razón, las normas de espacios de la vivienda consideran cinco habitantes en cada hogar. Actualmente, por fines de comparabilidad, se mantienen dichas normas a pesar que el tamaño promedio del hogar particular disminuyó a cuatro personas.

En un hogar de dos o más miembros, se requerirán espacios con diferentes funcionalidades (dormitorios, cocina y habitaciones multiuso) para no estorbarse entre sí; sin embargo, un hogar de una sola persona puede realizar todas sus actividades en una sola habitación sin causar inconveniente a nadie. Por esta razón, el cálculo de hacinamiento es diferente para los hogares unipersonales de los conformados por más de una persona.

a) Hogares de dos o más personas

Se consideran los dormitorios, habitaciones multiuso y la cocina.

Dormitorios por persona: La norma establece que en un hogar de cinco personas existan por lo menos dos dormitorios, vale decir que existan $2/5$ (0.4) dormitorios por cada miembro del hogar.

Habitaciones multiuso³² por persona: La norma establece que en un hogar de cinco personas existan por lo menos una habitación multiuso, vale decir que existan 1/5 (0.2) cuartos multiuso por cada miembro del hogar.

Los cálculos del Índice de carencia de dormitorios y de habitaciones multiuso siguen el siguiente procedimiento:

Cálculo del **Índice de logro en dormitorios (Ld) y habitaciones multiuso (Lh):**

$$Ld = \frac{\#D/\#Per}{2/5}; Lh = \frac{\#HM/\#Per}{1/5}$$

Donde:

#Per: Número de personas en el hogar

#D: Número de dormitorios

#HM: Número de habitaciones multiuso

2/5; 1/5: Norma de dormitorios y habitaciones multiuso por persona

El índice de logro, al ser una función del número de personas y de habitaciones observadas en cada hogar, no tiene un rango definido, pudiendo tomar cualquier valor mayor o igual a cero. Por esta dificultad, es necesario acotar el límite superior del rango. Para ello se construye una nueva escala que considera como máximo, para dormitorios y habitaciones multiuso, un logro de cinco (equivalente a dos dormitorios por persona y a un cuarto multiuso por persona, respectivamente). Este procedimiento se describe a continuación:

Si $Ld,h > 5$ entonces, $LRd,h = 2$

Si $1 < Ld,h \leq 5$ entonces, $LRd,h = 1 + \frac{Lx - 1}{5 - 1}$

Si $0 \leq Ld,h \leq 1$ entonces, $LRd,h = Lx$

³² La medición considera habitaciones multiuso a los cuartos de la vivienda no usados como dormitorio (sin contar cuartos de baño ni cocina).

Donde:

Ld: Índice de logro es dormitorios

Lm: Índice de logro en habitaciones multiuso

LRd: Índice de logro reescalado en dormitorios

LRh: Índice de logro reescalado en habitaciones multiuso

El índice de carencia de dormitorios (**NBI(d)**) y habitaciones multiuso (**NBI(h)**) son:

$$NBI(d) = 1 - LRd ; NBI(h) = 1 - LRh$$

Tenencia de cocina: La norma es que el hogar cuente con cocina, independientemente del número de componentes del hogar. Se considera como cualidad la tenencia de cocina, por tanto se asignan las siguientes calificaciones:

Tabla 7.5. CALIFICACIÓN DE TENENCIA DE COCINA

TENENCIA DE COCINA	
Tiene cocina	1
No Tiene cocina	0

La agregación de los tres componentes de hacinamiento en un solo índice de carencia de espacios, introduce la definición de **dormitorios equivalentes (DE)**, según la que valora en proporción de dormitorios a la cocina y habitaciones multiuso de la siguiente manera³³:

Cocina = 0.5 dormitorios

Habitación multiuso = 1.5 dormitorios

³³ Estas equivalencias se basan en la aproximación de las dimensiones establecidas para las habitaciones de las viviendas de carácter social.

Los espacios requeridos de la vivienda dependen del número de personas que compongan el hogar, por tanto el **Índice de carencia de espacios (NBI(Es))** introduce esta variable en su cálculo:

$$NBI(Es) = \frac{(1 * NBI(d) * \frac{2}{5} * NPer) + (1.5 * NBI(h) * \frac{1}{5} * NPer) + (0.5 * NBI(c))}{(1 * \frac{2}{5} * NPer) + (1.5 * \frac{1}{5} * NPer) + (0.5 * 1)}$$

Donde:

NBI(d): Índice de carencia de dormitorios

NBI(h): Índice de carencia de habitaciones multiuso

NBI(c): Índice de carencia de cocina

NPer: Número de personas en el hogar

b) Hogares unipersonales

La norma para un hogar unipersonal, es que cuente con una habitación, sin importar la funcionalidad que ésta tenga (dormitorio o cuarto multiuso). El **logro de espacios del hogar unipersonal (LE₁)** se cuantifica de la siguiente manera:

$$LE_1 = \frac{\#h + (0.5)(Cc)}{1}$$

Donde:

#h: Total de habitaciones

Cc: Calificación de cocina

0.5: Equivalencia de la cocina

Como el rango de LE_1 puede variar grandemente, precisa ser reescalado (teniendo como máximo el valor de 3.5, correspondiente a una cocina y tres habitaciones):

Si $LE_1 > 3.5$ entonces, $LER_1 = 2$

Si $1 \leq LE_1 \leq 3.5$ entonces, $LER_1 = 1 + \frac{Lx - 1}{3.5 - 1}$

Si $0 \leq LE_1 \leq 1$ entonces, $LER_1 = Lx$

Donde:

LER_1 : Índice de logro de espacios reescalado para el hogar

LE_1 : Índice de logro de espacios para el hogar unipersonal

El **Índice de carencia de espacios de hogar unipersonal (nbi(Es))** será:

$$NBI(Es) = 1 - LER_1$$

Una vez obtenidos los índices de carencia de materiales y de espacios, se calcula el **Índice de carencia de la vivienda (NBI(V))**:

$$NBI(V) = \frac{NBI(M) + NBI(Es)}{2}$$

Donde:

$NBI(M)$: Índice de carencia de materiales de construcción

$NBI(Es)$: Índice de carencia de espacios de la vivienda

7.2.2.2 Servicios e insumos básicos

7.2.2.2.1 Saneamiento básico

Comprende la calidad del agua que depende en gran medida de la procedencia y del modo de distribución de la misma en la vivienda.

a) Abastecimiento y procedencia de agua

Las calificaciones y normas para el abastecimiento y procedencia de agua se observan en el siguiente cuadro.

Tabla 7.6. CALIFICACIÓN DEL ABASTECIMIENTO DE AGUA

LISTADO DE PROCEDENCIA Y DISTRIBUCIÓN DE AGUA			
PROCEDENCIA DEL AGUA	DISTRIBUCIÓN DE AGUA		
	Por cañería dentro de la vivienda	Por cañería fuera de la vivienda pero dentro del lote o terreno	No se distribuye por cañería
Cañería de red	4	3	2
Pileta pública	4	3	2
Carro repartidor (aguatero)	3	2	0
Pozo o noria con bomba	3	2	1
Pozo o noria sin bomba	3	2	1
Río, Vertiente, Acequia	2	1	0
Lago, Laguna, Curiche	2	1	0
Otra	2	1	0

b) Servicio sanitario

En el caso del servicio sanitario, se hace una distinción entre las normas para medio rural y el urbano.

La norma del área rural, es de 1.5, que corresponde a sanitario de uso privado del hogar con desagüe a pozo ciego.

La norma para el área urbana es la tenencia de baño compartido con desagüe a alcantarillado.

Tabla 7.7. CALIFICACIÓN DEL SERVICIO SANITARIO

ELIMINACIÓN DE EXCRETAS Y TENENCIA DE BAÑO			
ELIMINACIÓN DE EXCRETAS	TIENE BAÑO		NO TIENE BAÑO
	Privado	Compartido	
Alcantarillado	4	3.5 u	0
Cámara Séptica	4	3	0
Pozo ciego	1.5 r	1	0
Superficie (calle, quebrada, río)	0.5	0.5	0

El **Índice de carencia de saneamiento básico (NBI(Sa))** es:

$$NBI(Sa) = \frac{NBI(aa) + NBI(ss)}{2}$$

Donde:

NBI(aa): Índice de carencia de abastecimiento de agua

NBI(ss): Índice de carencia de servicio sanitario

7.2.2.2 Insumos energéticos

Agrupan a la disponibilidad de energía eléctrica de los hogares y el principal combustible que usan para cocinar.

a) Energía eléctrica

La norma es que la vivienda cuente con energía eléctrica. Su calificación es binaria, como muestra el siguiente cuadro:

Tabla 7.8. CALIFICACIÓN DE ENERGÍA ELÉCTRICA

TENENCIA DE SERVICIO ELÉCTRICO	
Tiene Energía eléctrica	1
No Tiene Energía eléctrica	0

b) Combustible para cocinar

La norma del combustible para cocinar es la utilización de gas. Las calificaciones de los diferentes combustibles se muestran a continuación.

Tabla 7.9. CALIFICACIÓN DEL COMBUSTIBLE PARA COCINAR

LISTADO DE COMBUSTIBLES	
Leña	1
Guano/Bosta o taquia	1
Kerosene	2
Gas (garrafa por cañería)	3
Electricidad	3
Otro	1
No utiliza	4

El **Índice de carencia de insumos energéticos (NBI(IE))** se calcula como:

$$NBI(IE) = \frac{NBI(ee) + NBI(cc)}{2}$$

Donde:

NBI(ee): Índice de carencia de energía eléctrica

NBI(cc): Índice de carencia de combustible para cocinar

Es así que, el **Índice de carencia de servicios e insumos básicos (NBI(SB))** será:

$$NBI(SB) = \frac{NBI(Sa) + NBI(IE)}{2}$$

Donde:

NBI(Sa): Índice de carencia de saneamiento básico

NBI(IE): Índice de carencia de insumos energéticos

7.2.2.3 Educación

El cálculo del componente educación, comprende tres aspectos: asistencia escolar, años de estudio aprobados o escolaridad y condición de alfabetismo.

Para el cálculo de este índice se excluyen a los menores de siete años de edad. También quedan fuera del cálculo la(el) empleada(o) del hogar y sus parientes, puesto que sesgarían la medición incrementando la carencia del hogar.

El requerimiento mínimo o normas de estas tres variables está diferenciado para cada miembro del hogar en función su edad.

Tabla 7.10. NORMAS DE LOS COMPONENTES DE EDUCACIÓN

NORMAS POR COMPONENTE SEGÚN EDAD			
Edad del Individuo	Años aprobados	Asistencia ⁽²⁾	Alfabetismo ⁽³⁾
7	0	1	0
8	1	1	0
9	2	1	0
10	3	1	1
11	4	1	1
12	5	1	1
13	6	1	1
14	7	1	1
15	8	1	1
16	9	1	1
17 - 38	10	0	1
39 - 53	8	0	1
Más de 53	5	0	1

(2) Los valores 1 corresponden a la condición de asistencia; los valores 0 reflejan inasistencia.

(3) Los valores 1 corresponden a la condición de alfabetismo; los valores 0 reflejan analfabetismo.

La adecuación del nivel educativo individual (**Índice de logro educativo: LEdi**), se calcula de la siguiente manera:

$$LEd_i = \frac{(ap_i + as_i)}{(ap^* + as^*)} * al_i$$

Donde:

ap_i : Años de escolaridad del individuo i

as_i : Condición de asistencia del individuo i

al_i : Condición de alfabetización del individuo i

ap^* : Norma años de escolaridad

as^* : Norma de asistencia

El LEdi, permite el cálculo de **Rezago educativo (Índice de carencia de educación NBI(Ed))** de cada individuo se calcula como:

$$NBI(Ed)_i = 1 - LE_i$$

para luego agregarse en un solo índice del hogar:

$$NBI(Ed) = \frac{1}{m} \sum_{i=1}^m NBI(Ed)_i$$

Donde:

$NBI(Ed)_i$: Índice de rezago educativo (carencia de educación) del individuo i

m : Número de personas del hogar (excuyendo a la empleada y sus parientes)

7.2.2.4 Salud

El índice de salud se calcula en base al lugar de atención de la salud de los miembros del hogar. Dependiendo de la calidad del servicio que se brinda en dichos lugares se asignan las siguientes calificaciones para cada opción, donde la norma es la atención de salud en dependencias del Ministerio de Previsión Social y Salud Pública (Hospitales generales, puestos médicos o sanitarios).

Tabla 7.11. CALIFICACIÓN DE LA ATENCIÓN DE SALUD

LUGAR DE ATENCIÓN		
Dependencias del Ministerio de salud		2
Cajas de Seguro social		3
Dependencia de ONG's o Iglesia		1.5
Servicios Privados ⁽¹⁾	Pobres	1,5
	No pobres	3
Farmacia		0,5
Jampiri, Yatiri, Curandero, Kallawayaya, Naturista (medicina tradicional)		1
Otro		0
No atiende su salud		0

(1) La distinción de los servicios de salud privados en "Pobres" y "No Pobres" se realiza por la heterogeneidad de cobertura, calidad y costos de estos servicios. El criterio considera que los pobres no tienen los recursos monetarios para acceder a servicios costosos, que por lo general son de mejor calidad; mientras que los no pobres sí tienen la capacidad de acceder a ellos.

Con las calificaciones asignadas, se calculan los índices de logro individual, y a partir de estos, se calcula el **Índice de logro de salud por hogar (LSa)** (promedio de los logros individuales).

$$LS_a = \frac{1}{\#per} \sum_{i=1}^{\#per} LS_{a_i}$$

Donde:

LS_{a_i}: Índice de logro de salud del individuo *i*

#per: Total de personas del hogar

Entonces, el **Índice de carencia de salud del hogar (NBI(S))** será:

$$NBI(S) = 1 - Sa$$

7.2.3 Agregación de carencias

Se agregan todas las carencias de cada hogar en un solo indicador NBI, se procede de la siguiente manera:

$$NBI = \frac{NBI(V) + NBI(SB) + NBI(Ed) + NBI(S)}{4}$$

Donde:

NBI(V): Índice de carencia de vivienda

NBI(SB): Índice de carencia de servicios básicos e insumos energéticos

NBI(Ed): Índice de carencia de servicio de educación

NBI(S): Índice de carencia de servicio de salud

El NBI promedia los índices de carencia de sus componentes y a su vez es un índice de carencia. Su recorrido es el mismo, al igual que sus rangos de satisfacción e insatisfacción.

7.2.4 Estratos de pobreza

La construcción de estos estratos pretende dar a conocer la intensidad de la pobreza. Consisten en dividir el recorrido del NBI [-1,1] en cinco segmentos, que forman cinco grupos y cada uno de ellos representa un estrato de pobreza. Cada hogar se clasifica en el estrato que le corresponda según su NBI alcanzado.

Tabla 7.12. CONDICIÓN O ESTRATO DE POBREZA

ESTRATO DE POBREZA	RECORRIDO NBI
Necesidades Básicas Satisfechas	$-1 \leq \text{NBI} < -0,1$
Umbral de Pobreza	$-0,1 \leq \text{NBI} \leq 0,1$
Pobreza Moderada	$0,1 < \text{NBI} \leq 0,4$
Indigencia	$0,4 < \text{NBI} \leq 0,7$
Marginalidad	$0,7 < \text{NBI} \leq 1$

Los dos primeros estratos (Necesidades básicas satisfechas y Umbral de pobreza) corresponden a los no pobres. Los estratos de Pobreza moderada, Indigencia y Marginalidad agrupan a la población pobre.

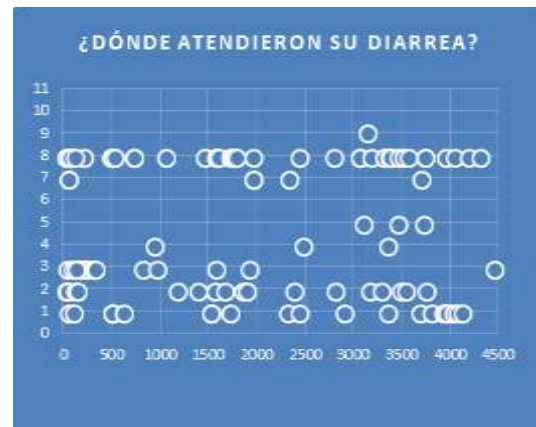
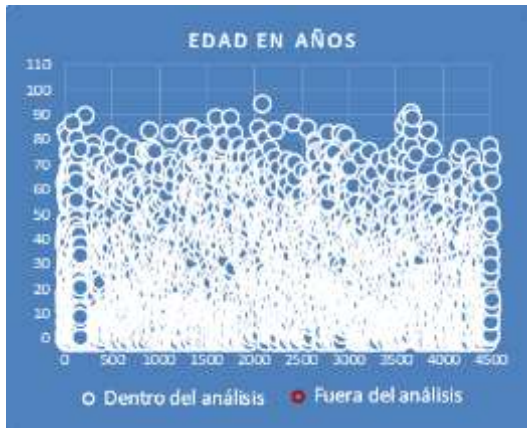
A partir de esta clasificación se obtiene el indicador de **Incidencia de pobreza**:

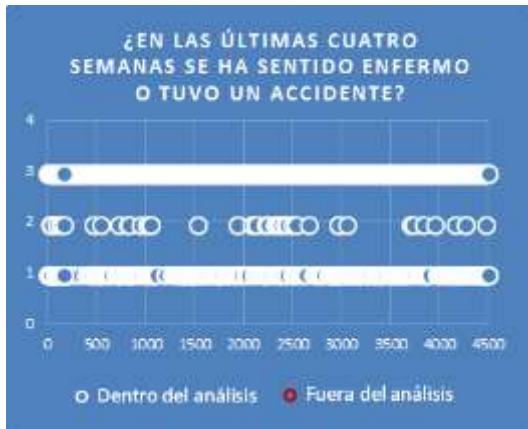
$$\text{Incidencia} = \frac{\text{Población pobre}}{\text{Población total}}$$

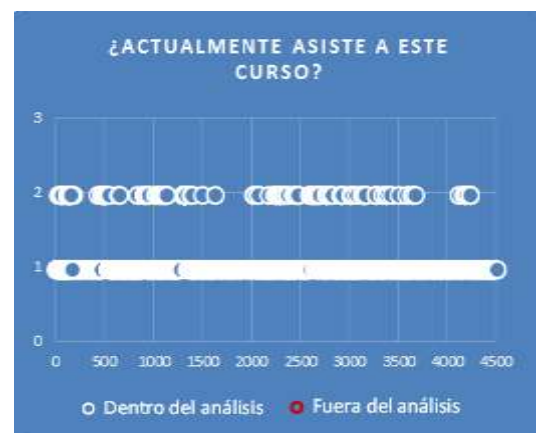
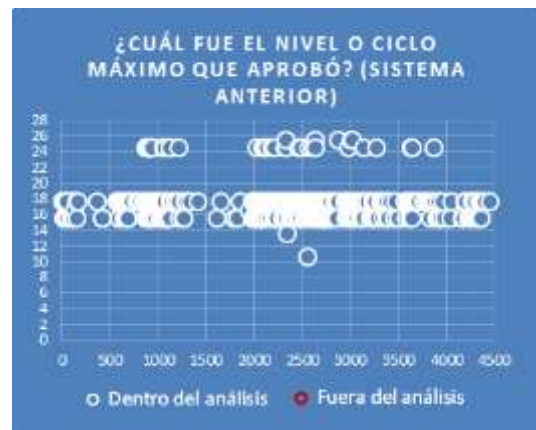
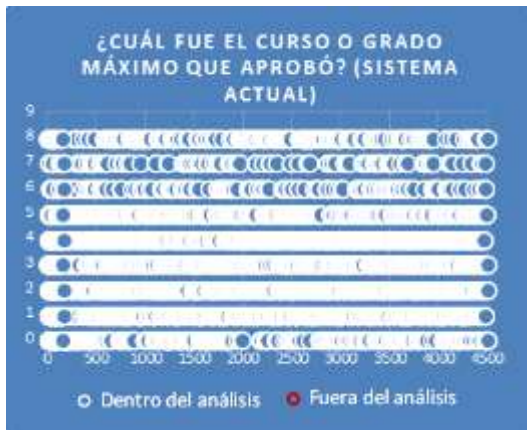
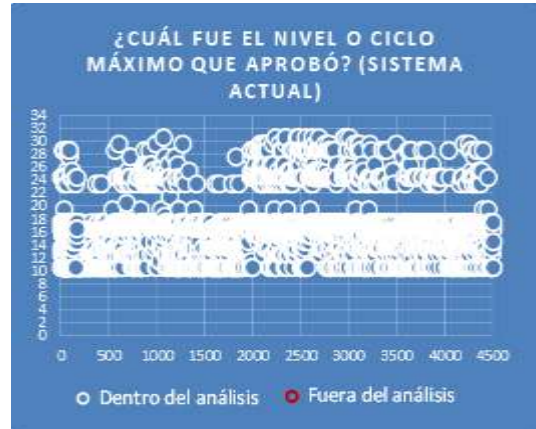
Que muestra el porcentaje de población pobre respecto al total de población.

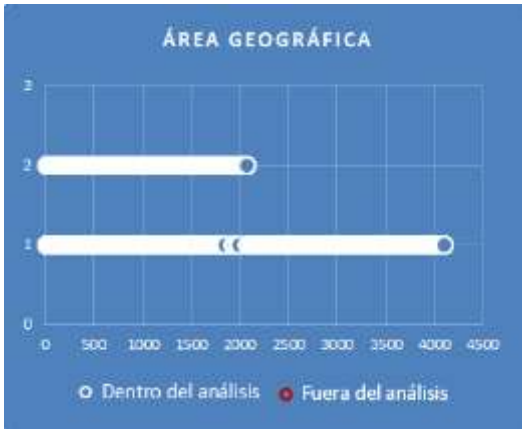
7.3 ANEXO 3 - Gráficas de dispersión de valores atípicos

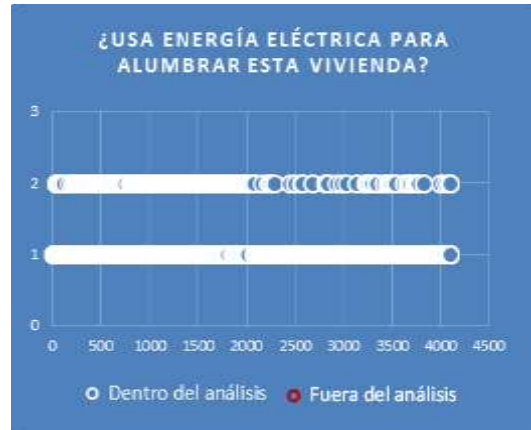
A continuación se muestran las gráficas de dispersión de todas las variables consideradas en el presente trabajo y se detallan los valores que serán parte del análisis como también los valores eliminados de la muestra.

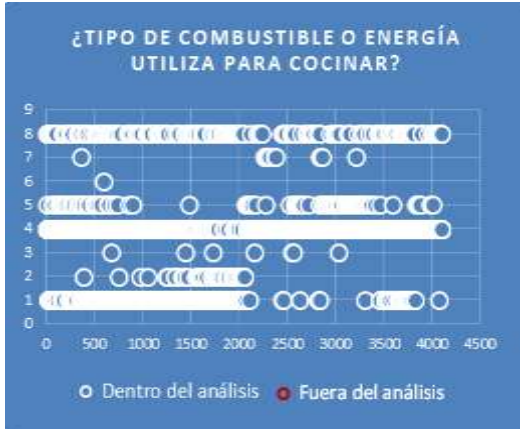












7.4 ANEXO 4 – Código NBI en SPSS

7.4.1 Código NBI materiales de la vivienda

* NBI - VIVIENDA*

* CALIFICACIÓN PAREDES *

```
IF (s8_05=1) c_paredes=1.5.
IF (s8_05=2 & s8_06=2) c_paredes=0.5.
IF (s8_05=2 & s8_06=1) c_paredes=1.
IF (s8_05=3 & s8_06=2) c_paredes=0.5.
IF (s8_05=3 & s8_06=1) c_paredes=1.
IF (s8_05=4) c_paredes=0.5.
IF (s8_05=5) c_paredes=1.
IF (s8_05=6) c_paredes=0.5.
IF (s8_05=7) c_paredes=0.
EXECUTE.
```

VARIABLE LABELS c_paredes 'Calificación NBI para material más utilizado en las paredes de la vivienda'.

* CALIFICACIÓN TECHOS *

```
IF (s8_07=1) c_techos=1.
IF (s8_07=2) c_techos=1.5.
IF (s8_07=3) c_techos=1.5.
IF (s8_07=4) c_techos=0.5.
IF (s8_07=5) c_techos=0.
EXECUTE .
```

VARIABLE LABELS c_techos 'Calificación NBI para material más utilizado en los techos de la vivienda'.

* CALIFICACIÓN PISOS *

```
IF (s8_08=1) c_pisos=0.
IF (s8_08=2) c_pisos=2.
IF (s8_08=3) c_pisos=2.
```

```

IF (s8_08=4) c_pisos=2.
IF (s8_08=5) c_pisos=1.
IF (s8_08=6) c_pisos=2.
IF (s8_08=7) c_pisos=1.
IF (s8_08=8) c_pisos=0.
EXECUTE .

```

VARIABLE LABELS c_pisos 'Calificación NBI para material más utilizado en los pisos de la vivienda'.

```

* CARENCIA DE MATERIALES DE LA VIVIENDA *
*****

```

```

* A) NBI PAREDES *
*****

```

```

COMPUTE nbi_paredes=(1 - (c_paredes/1)).
EXECUTE .

```

VARIABLE LABELS nbi_paredes 'NBI para material más utilizado en las paredes de la vivienda'.

```

* B) NBI TECHOS *
*****

```

```

COMPUTE nbi_techos=(1 - (c_techos/1)).
EXECUTE .

```

VARIABLE LABELS nbi_techos 'NBI para material más utilizado en los techos de la vivienda'.

```

* C) NBI PISOS *
*****

```

```

COMPUTE nbi_pisos=(1 - (c_pisos/1)).
EXECUTE .

```

VARIABLE LABELS nbi_pisos 'NBI para material más utilizado en los pisos de la vivienda'.

```

* D) NBI CARENCIA DE MATERIALES DE LA VIVIENDA *
*****

```

```

COMPUTE nbi_M=((nbi_paredes + nbi_techos + nbi_pisos)/3).
EXECUTE .

```

VARIABLE LABELS nbi_M 'NBI para la carencia de materiales de la vivienda'.

* HOGARES DE DOS O MÁS PERSONAS *

* A) INDICE DE LOGRO EN DORMITORIOS *

IF (miembros>1) ld=((s8_21/miembros)/0.4).
EXECUTE.

VARIABLE LABELS ld 'Indice de logro en dormitorios'.

* B) INDICE DE LOGRO REESCALADO EN DORMITORIOS *

IF (1<ld<=5) lrd=(1+((ld-1)/(5-1))).
EXECUTE .

IF (ld>5) lrd=2.
EXECUTE .

IF (ld<=1) lrd=ld.
EXECUTE .

VARIABLE LABELS lrd 'Indice de logro reescalado en dormitorios'.

* HABITACIONES MULTIUSO *

COMPUTE hm=s8_20-s8_21.
VARIABLE LABELS hm 'Número de habitaciones multiuso'.
EXECUTE .

* INDICE DE LOGRO EN HABITACIONES MULTIUSO *

IF (miembros>1) lh=((hm/miembros)/0.2).
EXECUTE .

VARIABLE LABELS lh 'Indice de logro en habitaciones multiuso'.

* INDICE DE LOGRO REESCALADO EN HABITACIONES MULTIUSO *

IF (1<lh<=5) lrh=(1+((lh-1)/(5-1))).
EXECUTE .

IF (lh>5) lrh=2.
EXECUTE .

IF (lh<=1) lrh=lh.
EXECUTE .

VARIABLE LABELS lrh 'Indice de logro reescalado en habitaciones multiuso'.

* CALIFICACIÓN COCINA *

IF (s8_17=1) c_cocina=1.
EXECUTE .

IF (s8_17=2) c_cocina=0.
EXECUTE .

VARIABLE LABELS c_cocina 'Calificación NBI para tenencia de cocina en la vivienda'.

* CARENCIA DE ESPACIOS *

* A) NBI DE DORMITORIOS PARA HOGARES DE DOS O MÁS PERSONAS *

IF (miembros>1) Enbi_d=(1-lrd).
EXECUTE.

VARIABLE LABELS Enbi_d 'NBI para la carencia de dormitorios en hogares con dos o más personas'.

* B) NBI HABITACIONES MULTIUSO PARA HOGARES DE DOS O MÁS PERSONAS *

IF (miembros>1) Enbi_h=(1-lrh).
EXECUTE.

VARIABLE LABELS Enbi_h 'NBI para la carencia de habitaciones multiuso en hogares con dos o más personas'.

* C) NBI PARA LA TENENCIA DE COCINA *

COMPUTE Enbi_c=(1-(c_cocina/1)).
EXECUTE .

VARIABLE LABELS Enbi_c 'NBI para la tenencia de cocina en la vivienda'.

* D) LOGRO DE ESPACIOS DEL HOGAR UNIPERSONAL *

IF (miembros=1) leu=(((s8_20)+(0.5*c_cocina))/1).
EXECUTE.

VARIABLE LABELS leu 'Logro de espacios del hogar unipersonal'.

* D) INDICE DE LOGRO DE ESPACIOS DEL HOGAR UNIPERSONAL *

IF (1<leu<=3.5) leur=(1+((leu-1)/(3.5-1))).
EXECUTE .

IF (leu>3.5) leur=2.
EXECUTE .

IF (leu<=1) leur=leu.
EXECUTE .

VARIABLE LABELS leur 'Indice de logro reescalado de espacios del hogar unipersonal'.

* E) NBI CARENCIA DE ESPACIOS *

IF (miembros=1) nbi_es=(1-leur).
EXECUTE .

IF (miembros>1)
nbi_es=(((Enbi_d*0.4*miembros)+(1.5*Enbi_h*0.2*miembros)+(0.5*Enbi_c))/((0.4*miembros)+(1.5*0.2*miembros)+(0.5))).
EXECUTE .

VARIABLE LABELS nbi_es 'NBI para la carencia de espacios en la vivienda'.

* NBI CARENCIA DE LA VIVIENDA *

COMPUTE NBI_V=((nbi_M+nbi_es)/2).
EXECUTE .

VARIABLE LABELS NBI_V 'NBI para la carencia de la vivienda'.

7.4.2 Código NBI servicios básicos

* CALIFICACIÓN DEL ABASTECIMIENTO DE AGUA *

```
IF (s8_09=1 & s8_10=1) c_agua=4.
IF (s8_09=2 & s8_10=1) c_agua=4.
IF (s8_09=3 & s8_10=1) c_agua=3.
IF (s8_09=4 & s8_10=1) c_agua=3.
IF (s8_09=5 & s8_10=1) c_agua=3.
IF (s8_09=6 & s8_10=1) c_agua=2.
IF (s8_09=7 & s8_10=1) c_agua=2.
IF (s8_09=8 & s8_10=1) c_agua=2.
IF (s8_09=1 & s8_10=2) c_agua=3.
IF (s8_09=2 & s8_10=2) c_agua=3.
IF (s8_09=3 & s8_10=2) c_agua=2.
IF (s8_09=4 & s8_10=2) c_agua=2.
IF (s8_09=5 & s8_10=2) c_agua=2.
IF (s8_09=6 & s8_10=2) c_agua=1.
IF (s8_09=7 & s8_10=2) c_agua=1.
IF (s8_09=8 & s8_10=2) c_agua=1.
IF (s8_09=1 & s8_10=3) c_agua=2.
IF (s8_09=2 & s8_10=3) c_agua=2.
IF (s8_09=3 & s8_10=3) c_agua=0.
IF (s8_09=4 & s8_10=3) c_agua=1.
IF (s8_09=5 & s8_10=3) c_agua=1.
IF (s8_09=6 & s8_10=3) c_agua=0.
IF (s8_09=7 & s8_10=3) c_agua=0.
IF (s8_09=8 & s8_10=3) c_agua=0.
EXECUTE .
```

VARIABLE LABELS c_agua 'Calificación NBI para el abastecimiento de agua de la vivienda'.

* NBI ABASTECIMIENTO DE AGUA *

```
COMPUTE nbi_aa=(1-(c_agua/3)).
EXECUTE .
```

VARIABLE LABELS nbi_aa 'NBI para el abastecimiento de agua de la vivienda'.

* VALORES EXTRAÑOS EN SERVICIO SANITARIO DEL AREA RURAL*

```
RECODE s8_13 (ELSE=Copy) INTO uso_baño.
VARIABLE LABELS uso_baño 'El baño, water o letrina es privado o compartido'.
EXECUTE .
```

```
RECODE s8_14 (ELSE=Copy) INTO desagüe.
VARIABLE LABELS desagüe 'El baño, water o letrina tiene desagüe...'.
EXECUTE .
```

```
IF ((urb_rur=2 & s8_13=1 & s8_14 <= 2) | (urb_rur=2 & s8_13=2 & s8_14 = 1))
ss_mal=1.
IF (ss_mal=1) desagüe=2.
IF (ss_mal=1) uso_baño=2.
EXECUTE .
```

```
VALUE LABELS
uso_baño
1 "Usado sólo por su hogar"
2 "Compartido con otros hogares"
.
EXECUTE .
```

```
VALUE LABELS
desagüe
1 "Alcantarillado"
2 "Cámara séptica"
3 "Pozo ciego"
4 "Superficie (calle, quebrada, río. etc.)"
.
EXECUTE .
```

* CALIFICACIÓN DE SERVICIO SANITARIO *

```
IF (desagüe=1 & s8_12=1 & uso_baño=1) c_sanitario=4.
IF (desagüe=2 & s8_12=1 & uso_baño=1) c_sanitario=4.
IF (desagüe=3 & s8_12=1 & uso_baño=1) c_sanitario=1.5.
IF (desagüe=4 & s8_12=1 & uso_baño=1) c_sanitario=0.5.
IF (desagüe=1 & s8_12=1 & uso_baño=2) c_sanitario=3.5.
IF (desagüe=2 & s8_12=1 & uso_baño=2) c_sanitario=3.
IF (desagüe=3 & s8_12=1 & uso_baño=2) c_sanitario=1.
IF (desagüe=4 & s8_12=1 & uso_baño=2) c_sanitario=0.5.
EXECUTE .
```

```
IF (s8_12=2) c_sanitario=0.
EXECUTE .
```

VARIABLE LABELS c_sanitario 'Calificación NBI para el servicio sanitario de la vivienda'.

* NBI SERVICIO SANITARIO *

IF (urb_rur=1) nbi_ss=(1-(c_sanitario/3.5)).
IF (urb_rur=2) nbi_ss=(1-(c_sanitario/1.5)).
EXECUTE .

VARIABLE LABELS nbi_ss 'NBI para el servicio sanitario de la vivienda'.

* NBI SANEAMIENTO BÁSICO *

COMPUTE nbi_SA=((nbi_aa + nbi_ss)/2).
EXECUTE .

VARIABLE LABELS nbi_SA 'NBI para la carencia de saneamiento básico de la vivienda'.

* CALIFICACIÓN DE ENERGÍA ELÉCTRICA *

IF (s8_15=1) c_eléctrica=1.
IF (s8_15=2) c_eléctrica=0.
EXECUTE .

VARIABLE LABELS c_eléctrica 'Calificación NBI para la energía eléctrica de la vivienda'.

* NBI ENERGÍA ELÉCTRICA *

COMPUTE nbi_ee=(1-(c_eléctrica/1)).
EXECUTE .

VARIABLE LABELS nbi_ee 'NBI para la energía eléctrica de la vivienda'.

* CALIFICACIÓN DEL COMBUSTIBLE PARA COCINAR *

IF (s8_18=1) c_combustible=1.
IF (s8_18=2) c_combustible=1.
IF (s8_18=3) c_combustible=2.
IF (s8_18=4) c_combustible=3.
IF (s8_18=5) c_combustible=3.
IF (s8_18=6) c_combustible=1.

```
IF (s8_18=7) c_combustible=3.
IF (s8_18=8) c_combustible=4.
EXECUTE .
```

VARIABLE LABELS c_combustible 'Calificación NBI del combustible para cocinar'.

```
* NBI COMBUSTIBLE PARA COCINAR *
*****
```

```
COMPUTE nbi_cc=(1-(c_combustible/3)).
EXECUTE .
```

VARIABLE LABELS nbi_cc 'NBI del combustible para cocinar'.

```
* NBI CARENCIA DE INSUMOS ENERGÉTICOS *
*****
```

```
COMPUTE nbi_IE=((nbi_ee+nbi_cc)/2).
EXECUTE .
```

VARIABLE LABELS nbi_IE 'NBI carencia de insumos energéticos'.

```
* NBI CARENCIA DE SERVICIOS E INSUMOS BÁSICOS *
*****
```

```
COMPUTE NBI_SB=((nbi_SA+nbi_IE)/2).
EXECUTE .
```

VARIABLE LABELS NBI_SB 'NBI carencia de servicios básicos e insumos energéticos'.

7.4.3 Código NBI salud

```
* NBI - CARENCIA EN SALUD *
*****
```

```
* IDENTIFICACIÓN DE ENFERMOS *
*****
```

```
IF (s3_01=1) enfermo=1.
IF (s3_06=1) enfermo=1.
IF (s3_14=1) enfermo=1.
IF (s3_14=2) enfermo=1.
IF ((s3_20>0)&(s3_20<99)) enfermo=1.
EXECUTE.
```

```
RECODE enfermo (MISSING=2).
EXECUTE.
```

VARIABLE LABELS enfermo 'Identificación de enfermos o atención de parto'.

VALUE LABELS

enfermo

1 "Enfermo"

2 "Sano"

.

EXECUTE .

* CALIFICACIÓN NBI PARA SALUD *

* EDA *

DO IF (enfermo=1).

RECODE S3_04a (4=3) (7=0.5) (10=1) (11=1.5) (1 thru 3=2) (8 thru 9=0) (ELSE=Copy)

INTO atención_EDA.

END IF.

VARIABLE LABELS atención_EDA 'Lugar de atención de EDAS'.

EXECUTE.

DO IF ((enfermo=1) & (P0=1)).

RECODE atención_EDA (5 thru 6=1.5).

END IF.

EXECUTE.

DO IF ((enfermo=1) & (P0=0)).

RECODE atención_EDA (5 thru 6=3).

END IF.

EXECUTE.

DO IF ((enfermo=1)).

RECODE atención_EDA (MISSING=0).

END IF.

EXECUTE .

* IRA *

DO IF (enfermo=1).

RECODE S3_10a (4=3) (7=0.5) (10=1) (11=1.5) (1 thru 3=2) (8 thru 9=0) (ELSE=Copy)

INTO atención_IRA.

END IF.

VARIABLE LABELS atención_IRA 'Lugar de atención de IRAS'.

EXECUTE.

```
DO IF ((enfermo=1) & (p0=1)).
RECODE atención_IRA (5 thru 6=1.5).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1) & (P0=0)).
RECODE atención_IRA (5 thru 6=3).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1)).
RECODE atención_IRA (MISSING=0).
END IF.
EXECUTE .
```

```
* TODOS LOS MIEMBROS DEL HOGAR *
*****
```

```
DO IF (enfermo=1).
RECODE S3_16a (4=3) (7=0.5) (10=1) (11=1.5) (1 thru 3=2) (8 thru 9=0) (ELSE=Copy)
INTO atención_TODOS.
END IF.
VARIABLE LABELS atención_TODOS 'Lugar de atención para todos los miembros del
hogar'.
EXECUTE.
```

```
DO IF ((enfermo=1) & (p0=1)).
RECODE atención_TODOS (5 thru 6=1.5).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1) & (p0=0)).
RECODE atención_TODOS (5 thru 6=3).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1)).
RECODE atención_TODOS (MISSING=0).
END IF.
EXECUTE .
```

* PARTOS *

```
DO IF (enfermo=1).
RECODE S3_24a (4=3) (7=0.5) (10=1) (11=1.5) (1 thru 3=2) (8 thru 9=0) (ELSE=Copy)
INTO atención_PARTO.
END IF.
VARIABLE LABELS atención_PARTO 'Lugar de atención de PARTOS'.
EXECUTE.
```

```
DO IF ((enfermo=1) & (p0=1)).
RECODE atención_PARTO (5 thru 6=1.5).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1) & (p0=0)).
RECODE atención_PARTO (5 thru 6=3).
END IF.
EXECUTE.
```

```
DO IF ((enfermo=1)&(s3_20>0)).
RECODE atención_PARTO (MISSING=0).
END IF.
EXECUTE .
```

* LOGRO DE SALUD INDIVIDUAL *

```
COMPUTE
LSa=MAX(atención_EDA,atención_IRA,atención_TODOS,atención_PARTO).
EXECUTE.
```

```
VARIABLE LABELS LSa 'Logro de salud individual'.
EXECUTE .
```

* AGREGAR POR HOGAR *

* NBI DE SALUD POR HOGAR *

```
COMPUTE NBI_S_P=(1-LSa_hogar).
EXECUTE .
```

```
RECODE NBI_S_P (Lowest thru -1=-1) (ELSE=Copy) INTO NBI_S.
EXECUTE.
```

```
VARIABLE LABELS NBI_S 'NBI carencia de salud por hogar'.
EXECUTE .
```

7.4.4 Código NBI educación

* NBI - CARENCIA DE EDUCACIÓN *

* AÑOS DE ESTUDIO *

IF (s4_02a=11) años_e=0.
 IF (s4_02a=12) años_e=s4_02b.
 IF (s4_02a=13) años_e=0.
 IF (s4_02a=14) años_e=s4_02b.
 IF (s4_02a=15) años_e=s4_02b+5.
 IF (s4_02a=16) años_e=s4_02b+8.
 IF (s4_02a=17) años_e=s4_02b.
 IF (s4_02a=18) años_e=s4_02b+8.
 IF (s4_02a=19) años_e=s4_02b.
 IF (s4_02a=20) años_e=s4_02b.
 IF (s4_02a=21) años_e=s4_02b.
 IF (s4_02a=22) años_e=s4_02b.
 IF (s4_02a=23) años_e=s4_02b+8.
 IF ((s4_02a=24)&(s4_02b=8)) años_e=17.
 IF ((s4_02a=24)&(s4_02b~=8)) años_e=s4_02b+12.
 IF ((s4_02a=25)&(s4_02b=8)) años_e=18.
 IF ((s4_02a=25)&(s4_02b~=8)) años_e=s4_02b+12.
 IF ((s4_02a=26)&(s4_02b=8)) años_e=18.
 IF ((s4_02a=26)&(s4_02b~=8)) años_e=s4_02b+12.
 IF ((s4_02a=27)&(s4_02b=8)) años_e=23.
 IF ((s4_02a=27)&(s4_02b~=8)) años_e=s4_02b+18.
 IF (s4_02a=28) años_e=s4_02b+12.
 IF (s4_02a=29) años_e=s4_02b+12.
 IF (s4_02a=30) años_e=s4_02b+12.
 IF (s4_02a=31) & (s4_03a=14) años_e=s4_03b.
 IF (s4_02a=31) & (s4_03a=15) años_e=s4_03b+5.
 IF (s4_02a=31) & (s4_03a=16) años_e=s4_03b+8.
 IF (s4_02a=31) & (s4_03a=17) años_e=s4_03b.
 IF (s4_02a=31) & (s4_03a=18) años_e=s4_03b+8.
 EXECUTE .

VARIABLE LABELS años_e 'Años de estudio'.

* RECODIFICACIÓN DE EDAD SEGÚN NBI *

RECODE s1_03 (7 thru 16=Copy) (17 thru 38=17) (39 thru 53=18) (54 thru 98=19)

INTO edad_r.

VARIABLE LABELS edad_r 'Edad recodificada según metodología NBI'.

VALUE LABELS

edad_r

7 "7 años"

8 "8 años"

9 "9 años"

10 "10 años"

11 "11 años"

12 "12 años"

13 "13 años"

14 "14 años"

15 "15 años"

16 "16 años"

17 "De 17 a 38 años"

18 "De 39 a 53 años"

19 "De 54 a 98 años"

.

EXECUTE.

* CALIFICACIÓN DE LA NORMA DE AÑOS APROBADOS SEGÚN LA EDAD *

IF (edad_r=7) nap=0.

IF (edad_r=8) nap=1.

IF (edad_r=9) nap=2.

IF (edad_r=10) nap=3.

IF (edad_r=11) nap=4.

IF (edad_r=12) nap=5.

IF (edad_r=13) nap=6.

IF (edad_r=14) nap=7.

IF (edad_r=15) nap=8.

IF (edad_r=16) nap=9.

IF (edad_r=17) nap=10.

IF (edad_r=18) nap=8.

IF (edad_r=19) nap=5.

EXECUTE .

VARIABLE LABELS nap 'Norma para años de estudio aprobados'.

EXECUTE.

* CALIFICACIÓN DE LA NORMA DE ASISTENCIA ESCOLAR SEGÚN LA EDAD *

IF (edad_r <= 16) nas=1.
 IF (edad_r > 16) nas=0.

VARIABLE LABELS nas 'Norma para asistencia escolar'.
 EXECUTE.

* ALFABETISMO *

IF (s4_01=2) al=0.
 IF (s4_01=1) al=1.
 EXECUTE.

VALUE LABELS
 al
 0 "Analfabeto"
 1 "Alfabeto"

.
 EXECUTE .

VARIABLE LABELS al 'Condición de alfabetismo'.
 EXECUTE.

* ASISTENCIA *

IF (s4_08=1) asiste=1.
 EXECUTE .

RECODE asiste (MISSING=0).
 EXECUTE .

VALUE LABELS
 asiste
 1 "asiste"
 0 "no asiste"

.
 EXECUTE .

VARIABLE LABELS asiste 'Condición de asitencia educativa'.
 EXECUTE.

* LOGRO EDUCATIVO INDIVIDUAL *

COMPUTE LEdi=((años_e+asiste)/(nap+nas))*al.
EXECUTE.

VARIABLE LABELS LEdi 'Logro educativo individual'.
EXECUTE.

* NBI - CARENCIA DE EDUCACIÓN PRELIMINAR *

COMPUTE nbi_Ed1=1-LEdi.
EXECUTE.

VARIABLE LABELS nbi_Ed1 'NBI para carencia de educación con fallas en la norma'.
EXECUTE.

RECODE nbi_Ed1 (Lowest thru -1=-1) (ELSE=Copy) INTO nbi_Ed2.
VARIABLE LABELS nbi_Ed2 'NBI para la carencia de educación individual preliminar'.
EXECUTE.

* SELECCIÓN DE DATOS NO CORRESPONDIENTES DEL NBI_ED *

(Menores de 7 años, empleadas y parientes de las mismas)

IF ((s1_03 >= 7) & (s1_05 <= 10)) Casos_Educación=1.
EXECUTE.

RECODE Casos_Educación (MISSING=0).
EXECUTE.

VARIABLE LABELS Casos_Educación 'Casos útiles para el cálculo del NBI_ED'.

VALUE LABELS

Casos_Educación

1 "Seleccionado"

0 "No seleccionado"

.
EXECUTE .

* NBI - CARENCIA DE EDUCACIÓN INDIVIDUAL*

IF (Casos_Educación=1) nbi_EDi=nbi_Ed2.
EXECUTE.

VARIABLE LABELS nbi_EDi 'NBI PARA LA CARENCIA EN EDUCACIÓN INDIVIDUAL'.
EXECUTE .

```
* POR HOGAR
* NBI_ED (PROMEDIO POR NÚMERO DE MIEMBROS SELECCIONADOS) Y NBI_S
*****
```

```
DATASET DECLARE NBI_EDUCACION_SALUD_POR_HOGAR.
AGGREGATE
  /OUTFILE='NBI_EDUACION_SALUD_POR_HOGAR'
  /BREAK=Folio
  /NBI_ED_sum=SUM(NBI_ED)
  /Casos_Educación_sum=SUM(Casos_Educación)
  /NBI_S_first=FIRST(NBI_S).
```

```
* CALCULANDO EL NBI_ED FINAL PARA LA BASE AGREGADA
*****
```

```
COMPUTE NBI_ED=NBI_ED_sum/Casos_Educación_sum.
EXECUTE .
VARIABLE LABELS NBI_ED 'NBI PARA LA CARENCIA EN EDUACIÓN DEL
HOGAR'.
EXECUTE .
```

```
RENAME VARIABLES NBI_S_first=NBI_S.
EXECUTE .
```

```
DELETE VARIABLES
NBI_ED_sum
Casos_Educación_sum
.
EXECUTE .
```

7.4.5 Código NBI hogar

```
* NBI FINAL (POR HOGAR)
*****
```

```
COMPUTE NBI=(NBI_V+NBI_SB+NBI_S+NBI_ED)/4 .
EXECUTE .
```

```
VARIABLE LABELS NBI 'NBI DEL HOGAR'.
```

```
* CONDICION O ESTRATO DE POBREZA
*****
```

```
IF (NBI>=-1 & NBI<-.1) ESTRATO_P=1.
IF (NBI>=-.1 & NBI<=0.1) ESTRATO_P=2.
IF (NBI>0.1 & NBI<=0.4) ESTRATO_P=3.
IF (NBI>0.4 & NBI<=0.7) ESTRATO_P=4.
```

```
IF (NBI>0.7 & NBI<=1) ESTRATO_P=5.
```

```
VARIABLE LABELS ESTRATO_P 'CONDICION O ESTRATO DE POBREZA'.
```

```
VALUE LABELS
```

```
ESTRATO_P
```

```
1 'NECESIDADES BÁSICAS SATISFECHAS'
```

```
2 'UMBRAL DE POBREZA'
```

```
3 'POBREZA MODERADA'
```

```
4 'INDIGENCIA'
```

```
5 'MARGINALIDAD'
```

```
.'
```

```
EXECUTE .
```

7.5 ANEXO 5 – Código red neuronal en C#

7.5.1 Código formulario BPN

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace red_Neuronal
{
    /// <summary>
    /// Summary description for Form1.
    /// </summary>

    //Controles que componen la interfaz neuronal
    //y los inputs de los parámetros de control

    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.Button BotonEntrenamiento;
        private System.Windows.Forms.Button BotonTesteo;
        private System.Windows.Forms.StatusBar barraEstado1;
        private System.Windows.Forms.NumericUpDown numArribaAbajo1;
        private System.Windows.Forms.NumericUpDown numArribaAbajo2;
        private System.Windows.Forms.TrackBar barraRastreo1;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;
```

```

private System.Windows.Forms.Label label5;
private System.Windows.Forms.ProgressBar barraProgreso1;
private System.Windows.Forms.Button botonSimple;
private System.Windows.Forms.TextBox primerTextBox;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label lblResultadosFalsos;
private System.Windows.Forms.Label lblResultadosPerdidos;
private Label label8;
private System.ComponentModel.IContainer components;

public Form1()
{
    InicializarComponente();
    Initialized();

    Invalidate();
}

Nred red = null;

//Datos de definición de entradas de la red

int[] nbis = new float[]{inputs};
double[,] inputs = new double[4, 1];
double[,] outputs = new double[1,1];

void InicializarRed()
{
    for (int i = 0; i < inputs.GetLength(0); i++)
    {
        int num = i;
        int mask = 0x200;
        for (int j = 0; j < 10; j++)
        {
            if ((num & mask) > 0)
                inputs[i, j] = 1;
            else
                inputs[i, j] = 0;

            mask = mask >> 1;
        }

        if (Array.BinarySearch(nbis, i) >= 0)
        {

```

```

        outputs[i,0] = 1;
    }
    else
    {
        outputs[i,0] = 0;
    }
}

private void DoStats(int num, int outputValue)
{
    if (Array.BinarySearch(nbis, num) >= 0)
    {
        if (outputValue == 0)
        {
            int val =
Convert.ToInt32(lblResultadosPerdidos.Text);
            lblResultadosPerdidos.Text = (val + 1).ToString();
        }
        else
        {
            if (outputValue == 1)
            {
                int val =
Convert.ToInt32(lblResultadosFalsos.Text);
                lblResultadosFalsos.Text = (val + 1).ToString();
            }
        }
    }

    void entrenarRed()
    {
        red = new Nred(inputs.GetLength(1), (int)numArribaAbajo1.Value,
(int)outputs.GetLength(1));

        red.FirstTimeSettings();

        red.entrenarRed(inputs, outputs, (int)numArribaAbajo2.Value,
barraProgreso1);
    }
}

```

```

    }

    //Función de simulación de proceso de la red neuronal

    private void comenzarSimulacion()
    {
        count = 0;
        lblResultadosPerdidos.Text = "0";
        lblResultadosFalsos.Text = "0";
        timer1.Interval = barraRastreo1.Value;

        timer1.Start();
        SimulacionComenzada = true;
    }

    int count = 0;

    //Función de descarte de datos erróneos

    protected override void Dispose( bool disposing )
    {
        if( disposing )
        {
            if (components != null)
            {
                components.Dispose();
            }
        }
        base.Dispose( disposing );
    }

    //Inicialización de los controles de componentes de la interfaz red
    //de red de acuerdo

    private void InicializarComponente()
    {
        this.components = new System.ComponentModel.Container();
        this.label1 = new System.Windows.Forms.Label();
        this.label2 = new System.Windows.Forms.Label();
        this.timer1 = new System.Windows.Forms.Timer(this.components);
        this.BotonEntrenamiento = new System.Windows.Forms.Button();
        this.BotonTesteo = new System.Windows.Forms.Button();
        this.barraEstado1 = new System.Windows.Forms.StatusBar();
        this.numArribaAbajo1 = new System.Windows.Forms.NumericUpDown();
        this.numArribaAbajo2 = new System.Windows.Forms.NumericUpDown();
        this.barraRastreo1 = new System.Windows.Forms.TrackBar();
        this.label3 = new System.Windows.Forms.Label();
        this.label4 = new System.Windows.Forms.Label();
    }

```



```

this.label5 = new System.Windows.Forms.Label();
this.barraProgreso1 = new System.Windows.Forms.ProgressBar();
this.botonSimple = new System.Windows.Forms.Button();
this.primerTextBox = new System.Windows.Forms.TextBox();
this.label6 = new System.Windows.Forms.Label();
this.lblResultadosFalsos = new System.Windows.Forms.Label();
this.lblResultadosPerdidos = new System.Windows.Forms.Label();
this.label8 = new System.Windows.Forms.Label();

((System.ComponentModel.ISupportInitialize)(this.numArribaAbajo1)).BeginInit();

((System.ComponentModel.ISupportInitialize)(this.numArribaAbajo2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.barraRastreo1)).BeginInit();
this.SuspendLayout();

this.label1.Location = new System.Drawing.Point(88, 238);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(160, 24);
this.label1.TabIndex = 3;
this.label1.Text = "Iteraciones de Entrenamiento";

this.label2.Location = new System.Drawing.Point(88, 200);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(112, 24);
this.label2.TabIndex = 4;
this.label2.Text = "Nro Nodos Ocultos";

this.timer1.Interval = 2000;
this.timer1.Tick += new System.EventHandler(this.timer1_Tick);

this.BotonEntrenamiento.Location = new System.Drawing.Point(496, 144);
this.BotonEntrenamiento.Name = "BotonEntrenamiento";
this.BotonEntrenamiento.Size = new System.Drawing.Size(104, 23);
this.BotonEntrenamiento.TabIndex = 6;
this.BotonEntrenamiento.Text = "Entrenar";
this.BotonEntrenamiento.Click += new
System.EventHandler(this.BotonEntrenamiento_Click);

this.BotonTesteo.Enabled = false;
this.BotonTesteo.Location = new System.Drawing.Point(488, 208);
this.BotonTesteo.Name = "BotonTesteo";
this.BotonTesteo.Size = new System.Drawing.Size(104, 24);
this.BotonTesteo.TabIndex = 7;
this.BotonTesteo.Text = "Probar Todos";
this.BotonTesteo.Click += new System.EventHandler(this.BotonTesteo_Click);

this.barraEstado1.Location = new System.Drawing.Point(0, 351);
this.barraEstado1.Name = "barraEstado1";
this.barraEstado1.Size = new System.Drawing.Size(696, 22);

```

```

this.barraEstado1.TabIndex = 8;
this.barraEstado1.Text = "barraEstado1";

this.numArribaAbajo1.Location = new System.Drawing.Point(24, 200);
this.numArribaAbajo1.Name = "numArribaAbajo1";
this.numArribaAbajo1.Size = new System.Drawing.Size(56, 20);
this.numArribaAbajo1.TabIndex = 9;
this.numArribaAbajo1.Value = new decimal(new int[] {
    15,
    0,
    0,
    0});
this.numArribaAbajo1.ValueChanged += new
System.EventHandler(this.numArribaAbajo1_ValueChanged_1);
this.numArribaAbajo1.Leave += new
System.EventHandler(this.numArribaAbajo1_Leave);

this.numArribaAbajo2.Increment = new decimal(new int[] {
    10,
    0,
    0,
    0});
this.numArribaAbajo2.Location = new System.Drawing.Point(24, 240);
this.numArribaAbajo2.Maximum = new decimal(new int[] {
    100000,
    0,
    0,
    0});
this.numArribaAbajo2.Name = "numArribaAbajo2";
this.numArribaAbajo2.Size = new System.Drawing.Size(56, 20);
this.numArribaAbajo2.TabIndex = 10;
this.numArribaAbajo2.Value = new decimal(new int[] {
    1000,
    0,
    0,
    0});

this.barraRastreo1.Location = new System.Drawing.Point(288, 208);
this.barraRastreo1.Maximum = 2000;
this.barraRastreo1.Minimum = 50;
this.barraRastreo1.Name = "barraRastreo1";
this.barraRastreo1.RightToLeft = System.Windows.Forms.RightToLeft.Yes;
this.barraRastreo1.Size = new System.Drawing.Size(104, 45);
this.barraRastreo1.TabIndex = 11;
this.barraRastreo1.TickFrequency = 100;
this.barraRastreo1.Value = 1000;
this.barraRastreo1.Scroll += new
System.EventHandler(this.barraRastreo1_Scroll);

```

```

this.label3.Location = new System.Drawing.Point(248, 208);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(40, 23);
this.label3.TabIndex = 12;
this.label3.Text = "Lento";
//
// label4
//
this.label4.Location = new System.Drawing.Point(408, 208);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(40, 23);
this.label4.TabIndex = 13;
this.label4.Text = "Rápido";
//
// label5
//
this.label5.Location = new System.Drawing.Point(272, 184);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(128, 16);
this.label5.TabIndex = 14;
this.label5.Text = "Velocidad de Simulación";
//
// barraProgreso1
//
this.barraProgreso1.Location = new System.Drawing.Point(496, 176);
this.barraProgreso1.Name = "barraProgreso1";
this.barraProgreso1.Size = new System.Drawing.Size(104, 23);
this.barraProgreso1.TabIndex = 15;
//
// botonSimple
//
this.botonSimple.Enabled = false;
this.botonSimple.Location = new System.Drawing.Point(488, 240);
this.botonSimple.Name = "botonSimple";
this.botonSimple.Size = new System.Drawing.Size(104, 32);
this.botonSimple.TabIndex = 16;
this.botonSimple.Text = "Probar un solo Numero";
this.botonSimple.Click += new System.EventHandler(this.botonSimple_Click);
//
// primerTextBox
//
this.primerTextBox.Location = new System.Drawing.Point(600, 248);
this.primerTextBox.Name = "primerTextBox";
this.primerTextBox.Size = new System.Drawing.Size(72, 20);
this.primerTextBox.TabIndex = 17;
//
// label6
//
this.label6.Location = new System.Drawing.Point(496, 296);

```

```

this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(112, 16);
this.label6.TabIndex = 20;
this.label6.Text = "Resultados Falsos";
//
// lblResultadosFalsos
//
this.lblResultadosFalsos.BackColor = System.Drawing.Color.White;
this.lblResultadosFalsos.ForeColor = System.Drawing.Color.Red;
this.lblResultadosFalsos.Location = new System.Drawing.Point(616, 296);
this.lblResultadosFalsos.Name = "lblResultadosFalsos";
this.lblResultadosFalsos.Size = new System.Drawing.Size(40, 16);
this.lblResultadosFalsos.TabIndex = 22;
this.lblResultadosFalsos.Text = "0";
//
// lblResultadosPerdidos
//
this.lblResultadosPerdidos.BackColor = System.Drawing.Color.Transparent;
this.lblResultadosPerdidos.ForeColor = System.Drawing.Color.Transparent;
this.lblResultadosPerdidos.Location = new System.Drawing.Point(616, 320);
this.lblResultadosPerdidos.Name = "lblResultadosPerdidos";
this.lblResultadosPerdidos.Size = new System.Drawing.Size(40, 16);
this.lblResultadosPerdidos.TabIndex = 23;
this.lblResultadosPerdidos.Text = "0";
//
// label8
//
this.label8.ForeColor = System.Drawing.SystemColors.ButtonFace;
this.label8.Location = new System.Drawing.Point(496, 320);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(96, 16);
this.label8.TabIndex = 21;
this.label8.Text = "Primos Omitidos";
//
// Form1
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(696, 373);
this.Controls.Add(this.lblResultadosPerdidos);
this.Controls.Add(this.lblResultadosFalsos);
this.Controls.Add(this.label8);
this.Controls.Add(this.label6);
this.Controls.Add(this.primerTextBox);
this.Controls.Add(this.botonSimple);
this.Controls.Add(this.barraProgreso1);
this.Controls.Add(this.label5);
this.Controls.Add(this.label4);
this.Controls.Add(this.label3);
this.Controls.Add(this.barraRastreo1);

```

```

        this.Controls.Add(this.numArribaAbajo2);
        this.Controls.Add(this.numArribaAbajo1);
        this.Controls.Add(this.barraEstado1);
        this.Controls.Add(this.BotonTesteo);
        this.Controls.Add(this.BotonEntrenamiento);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.label1);
        this.Name = "Form1";
        this.Text = "Simulacion de Red Neuronal Cluster Luchito";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.Paint += new
System.Windows.Forms.PaintEventHandler(this.Form1_Paint);
        this.Resize += new System.EventHandler(this.Form1_Resize);
        ((System.ComponentModel.ISupportInitialize)(this.numArribaAbajo1)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.numArribaAbajo2)).EndInit();
        ((System.ComponentModel.ISupportInitialize)(this.barraRastreo1)).EndInit();
        this.ResumeLayout(false);
        this.PerformLayout();

    }
    #endregion

    [STAThread]
    static void Main()
    {
        Application.Run(new Form1());
    }

    private void Form1_Paint(object sender,
System.Windows.Forms.PaintEventArgs e)
    {
        // Dibujar la red neuronal
        int input = inputs.GetLength(1);
        int hidden = (int)numArribaAbajo1.Value;
        int output = outputs.GetLength(1);

        Graphics g = e.Graphics;

        // Dibujar valores de ingreso
        for (int i = 0; i < input; i++)
        {
            g.DrawString(inputs[CurrentCount, i].ToString(), Font,
Brushes.Navy,
            (ClientRectangle.Width - input * 50)/2 + i * 50 + 15, 5, new
StringFormat());
        }

        g.DrawString(CurrentCount.ToString(), Font, Brushes.Green,

```

```

ClientRectangle.Width - 20, 5, new StringFormat());

// Dibujar la capa de valores de ingreso
for (int i = 0; i < input; i++)
{
    //
    g.DrawEllipse(Pens.Black, (ClientRectangle.Width - input *
50)/2 + i * 50, 20, 30, 30);
    g.DrawString("S", SymbolFont, Brushes.Green,
                (ClientRectangle.Width - input * 50)/2 + i * 50 + 5,
20 + 5);
}

// Dibujar la capa oculta
for (int i = 0; i < hidden; i++)
{
    g.DrawEllipse(Pens.Black,
                (ClientRectangle.Width - hidden * 50)/2 + i * 50, 70,
30, 30);
    g.DrawString("S", SymbolFont, Brushes.Green,
                (ClientRectangle.Width - hidden * 50 + 5)/2 + i * 50,
70 + 5);
}

// Dibujar la capa de salida
for (int i = 0; i < output; i++)
{
    g.DrawEllipse(Pens.Black,
                (ClientRectangle.Width - output * 50)/2 + i * 50,
120, 30, 30);
    g.DrawString("S", SymbolFont, Brushes.Green,
                (ClientRectangle.Width - output * 50)/2 + i * 50 +
10, 120 + 5);
}

// Dibujar los valores de salida
for (int i = 0; i < output; i++)
{
    if (SimulacionComenzada)
    {
        g.DrawString(CurrentOutputValue[i], Font,
Brushes.Purple,
                (ClientRectangle.Width - output * 50)/2 + i *
50 + 15, 160, new StringFormat());
    }
    else
    {
        g.DrawString(outputs[CurrentCount, i].ToString(),
Font, Brushes.Navy,

```

```

                    (ClientRectangle.Width - output * 50)/2 + i *
50 + 15, 160, new StringFormat());
                }
            }

            // Conectar cada capa de datos con la capa oculta
            for (int i = 0; i < input; i++)
            {
                for (int j = 0; j < hidden; j++)
                //
                {
                    g.DrawLine(Pens.Red,
                        (ClientRectangle.Width - input * 50)/2 + i*50 + 15,
50,
                        (ClientRectangle.Width - hidden * 50)/2 + j*50 +
15, 70);
                }
            }

            // Conectar cada capa oculta con la capa de salida
            for (int i = 0; i < hidden; i++)
            {
                for (int j = 0; j < output; j++)
                //
                {
                    g.DrawLine(Pens.Red,
                        (ClientRectangle.Width - hidden * 50)/2 +
i*50 + 15, 100,
                        (ClientRectangle.Width - output * 50)/2 +
j*50 + 15, 120);
                }
            }

        }

        //Función de ajuste parámetros de input de nodos de red
        private void numArribaAbajo1_ValueChanged(object sender,
System.EventArgs e)
        {
            Invalidate();
        }

        string[] CurrentOutputValue;
        int CurrentCount = 0;

```

```

Font SymbolFont = new Font("Symbol", 14, FontStyle.Bold);

//Función de ajuste del temporizador de simulación

private void timer1_Tick(object sender, System.EventArgs e)
{
    CurrentCount = count;
    CurrentOutputValue = red.Test_Drive(CurrentCount,
inputs).Split(new char[]{' '});

    DoStats(CurrentCount, Convert.ToInt32(CurrentOutputValue[0]));

    count++;
    if (CurrentCount >= inputs.GetUpperBound(0))
    {
        Invalidate();
        Update();
        SimulacionComenzada = false; //
        timer1.Stop();
    }
    else
    {
        Invalidate();
    }
}

//Funciones de ajuste de valores de parámetros de entrada
//para la simulación de entrenamiento de la red

e) private void BotonEntrenamiento_Click(object sender, System.EventArgs
{
    barraEstado1.Text = "Entrenando...";
    entrenarRed();
    BotonTesteo.Enabled = true;
    this.botonSimple.Enabled = true;
    barraEstado1.Text = "Listo";
}

bool SimulacionComenzada = false;
private void BotonTesteo_Click(object sender, System.EventArgs e)
{
    comenzarSimulacion();
}

private void numArribaAbajo1_ValueChanged_1(object sender,
System.EventArgs e)
{

```



```

        Invalidate();
    }

e) private void numArribaAbajo1_Leave(object sender, System.EventArgs
    {
        Invalidate();
    }

private void Form1_Load(object sender, System.EventArgs e)
{
}

private void barraRastreo1_Scroll(object sender, System.EventArgs e)
{
    timer1.Interval = barraRastreo1.Value;
}

private void botonSimple_Click(object sender, System.EventArgs e)
{
    CurrentCount = Convert.ToInt32(this.primerTextBox.Text);
    CurrentOutputValue = red.Test_Drive(CurrentCount,
inputs).Split(new char[]{' '});
    Invalidate();
    Update();
}

private void Form1_Resize(object sender, System.EventArgs e)
{
    Invalidate();
}
}
}

```

7.5.2 Código neurona BPN

```

using System;

// Definición de la clase nodo (dato de entrada y capa oculta)
// y su comportamiento

namespace RedNeuronal
{
    ///
    public class node
    {
        private double activation;
        private double threshold;
        public double []weights;
        public double error;

        public node()
        {
            this.activation = 0;
            this.error = 0;
        }

        public node(double act,double thr,int numOfweights)
        {
            this.activation = act;
            this.threshold = thr;
            this.weights = new double[numOfweights];
        }

        public double Activation
        {
            set {activation = value ;}
            get {return activation;}
        }

        public double Threshold
        {
            set {threshold = value;}
            get {return threshold;}
        }
    }
}

```

7.5.3 Código red BPN

```

using System;
using System.Windows.Forms;

// Definición de las funciones de
// la red neuronal

namespace Red_neuronal
{
    public class Nred
    {
        private int i;
        private int o;
        private int h;

        private double LH=0.15f; // parametro de la capa oculta
        private double LO=0.2f; // parametro de la salida

        private double []inputsGuardados;// array de datos de entranamiento
"inputs"
        private double []Outputsdeseados;// array de datos de entranamiento
"salidas deseadas"

        private node []input; // array neuronas input
        private node []output;// array neuronas output
        private node []hidden;// array neuronas capas ocultas

        private double error_total;
        public double error_comparado_de_tolerancia=0;

        //Función sigmoidea de simulación para la red neuronal

        private double sigmoid(double x)
        {
            return 1/(1+Math.Exp(-x));
        }

        //Funciones de ajuste de las condiciones iniciales
        //de las variables de la red neuronal

        public Nred(int inp , int hide , int outp)
        {
            i = inp ; o = outp ; h = hide;
            int ahm = 0;

```

```

input = new node[i];
output = new node[o];
hidden = new node[h];
Random rand = new
Random(unchecked((int)DateTime.Now.Ticks));
// inicializar la red arrays y los pesos
for(int x=0 ; x<inp ; x++)
{
    input[x] = new node();
    input[x].weights = new double[this.h];
    for(int j=0 ; j<hide ; j++)// loop para inicializar los pesos
    {
        ahm = rand.Next()& 1;
        input[x].weights[j] = rand.NextDouble() ;
        if(ahm == 0)
            input[x].weights[j] *= -1;
    }
}
for(int y=0; y<hide ; y++)
{
    hidden[y] = new node();
    hidden[y].weights = new double[this.o];
    for(int j=0 ; j<outp ; j++)// loop para inicializar los pesos
        hidden[y].weights[j] = rand.NextDouble();
}
for(int z=0 ; z<outp ; z++)
{
    output[z] = new node();
}
}

```

// Establecimiento del umbral de las neuronas de la capa oculta y la capa de salida

```

public void AjustesIniciales()
{
    Random x = new Random(unchecked((int)DateTime.Now.Ticks));
    for(int i=0 ; i<h ; i++)
        hidden[i].Threshold = x.NextDouble();
    for(int i=0 ; i<o ; i++)
        output[i].Threshold = x.NextDouble();
}

```

// Paso de datos de entrenamiento a la red (inputs y outputs deseados)

```

public void AntesEntrenamiento(params double []list)
{
    int counter=0;
    int j=0;
    int k=0;
    inputsGuardados = new double[this.i];
    Outputsdeseados = new double[this.o];

    while(counter < list.Length)
    {
        if( counter < this.i)
        {
            inputsGuardados[j] = list [counter];
            j++;
        }
        else
        {
            Outputsdeseados[k] = list[counter];
            k++;
        }
        counter++;
    }
}

//Función de entrenamiento de red

public void EntrenarRed(double[,] inputlist, double[,] outputlist, int
iterations, ProgressBar progressBar1)
{
    inputsGuardados = new double[this.i];
    Outputsdeseados = new double[this.o];

    int outputlistSampleLength = outputlist.GetUpperBound(0) + 1;
    int outputlistLength = outputlist.GetUpperBound(1) + 1;
    int inputlistLength = inputlist.GetUpperBound(1) + 1;

    progressBar1.Minimum = 0;
    progressBar1.Maximum = iterations;
    progressBar1.Value = 0;

    for (int i = 0; i < iterations; i++)
    {
        progressBar1.Value ++;
        for (int sampleindex = 0; sampleindex <
outputlistSampleLength; sampleindex++)
        {

```

```

        for (int j = 0; j < inputlistLength; j++)
        {
            inputsGuardados[j] = inputlist[sampleindex,
j]);
        }
        for (int k = 0; k < outputlistLength; k++)
        {
            Outputsdeseados[k] =
outputlist[sampleindex, k];
        }
        // siguiente entrenamiento
        Entranamiento_patrones();
    }
}

void PopulateInputList(double[,] inputlist, int index)
{
    for (int j = 0; j < inputlist.GetUpperBound(1) + 1; j++)
    {
        inputsGuardados[j] = inputlist[index, j];
    }
}

// Función de testeo de red

public string Test(int index, double[,] inputlist)
{
    PopulateInputList(inputlist, index);
    this.Calc_Activation();
    string temp = this.print_output();
    return temp;
}

// Funciones de entrenamiento

public void Entranamiento_patrones()
{
    this.Calc_Activation();
    this.Calc_error_output();
    this.Calc_error_hidden();
    this.Calc_new_Thresholds();
    this.Calc_new_weights_in_hidden();
}

```

```

        this.Calc_new_weights_in_input();
    }

    // Cálculo de la activación de la capa oculta y la capa de salida
    private void Calc_Activation()
    {
        //
        int ch=0;
        while(ch<this.h)
        {
            for(int ci=0 ; ci<this.i ; ci++)
                hidden[ch].Activation += inputsGuardados[ci] *
input[ci].weights[ch];

            ch++;
        }//

        // calcular output de la capa oculta
        for(int x=0 ; x<this.h ; x++)
        {
            hidden[x].Activation += hidden[x].Threshold ;
            hidden[x].Activation = sigmoid(hidden[x].Activation ) ;
        }

        int co=0;
        while(co<this.o)
        {
            for(int chi=0 ; chi<this.h ; chi++)
                output[co].Activation += hidden[chi].Activation *
hidden[chi].weights[co];

            co++;
        }//

        //
        for(int x=0 ; x<this.o ; x++)
        {
            output[x].Activation += output[x].Threshold ;
            output[x].Activation = sigmoid(output[x].Activation ) ;
        }

    }

    // Obtención de salidas
    public string print_output()
    {
        string outp="";
    }

```

```

// calcular la salida de la capa de resultados
for(int x=0 ; x<this.o ; x++)
{
    if(output[x].Activation>0.5)
        outp += "1" + " ";
    else
        outp += "0" + " ";
}
return outp;
}

// Cálculo del error de cada neurona de salida

private void Calc_error_output()
{
    for(int x=0 ; x<this.o ; x++)
        output[x].error = output[x].Activation * (1 -
output[x].Activation ) * (Outputsdeseados[x] - output[x].Activation );
}

// Cálculo del error de cada neurona oculta

private void Calc_error_hidden()
{
    int y=0;
    while(y<this.h)
    {
        for(int x=0 ; x<this.o ; x++)
        {
            hidden[y].error += hidden[y].weights[x] *
output[x].error ;
        }
        hidden[y].error *= hidden[y].Activation * (1 -
hidden[y].Activation );
        y++;
    }
}

// Cálculo de los nuevos umbrales para cada neurona (nodo)

private void Calc_new_Thresholds()
{
    // siguiente iteración

```



```

for(int x=0 ; x<this.h ; x++)
    hidden[x].Threshold += hidden[x].error * this.LH ;
// siguiente iteración
for(int y=0 ; y<this.o ; y++)
    output[y].Threshold += output[y].error * this.LO;
}

// Cálculo de los nuevos pesos de la capa oculta y de salida

private void Calc_new_weights_in_hidden()
{
    int x=0;
    double temp=0.0f;
    while(x < this.h)
    {
        temp = hidden[x].Activation * this.LO;
        for(int y=0 ; y<this.o ; y++)
        {
            hidden[x].weights[y] += temp * output[y].error;
        }
        x++;
    }
}

```

// Cálculo de los nuevos pesos entre la capa de entrada y la capa oculta

```

private void Calc_new_weights_in_input()
{
    int x=0;
    double temp=0.0f;
    while(x < this.i)
    {
        temp = inputsGuardados[x] * this.LH;
        for(int y=0 ; y<this.h ; y++)
        {
            input[x].weights[y] += temp * hidden[y].error;
        }
        x++;
    }
}

// Cálculo del error total

```

```

public double Calc_error_total_in_pattern()
{

```

```
        double temp=0.0f;
        for(int x=0 ; x<this.o ; x++)
            temp+=output[x].error;
        return this.error_total;
    }

    // Inicialización del error a 0 después de un ciclo de proceso

    public void reset_error_total()
    {
        this.error_total = 0;
    }
}
}
```

7.6 ANEXO 6 – Código análisis cluster en SPSS

* CLASIFICACIÓN – KMEDIAS *

* CLASIFICACIÓN CON CENTROS INICIALES *

```
QUICK CLUSTER NBI_V NBI_SB NBI_ED NBI_S
/MISSING=LISTWISE
/CRITERIA=CLUSTER(5)
/METHOD=CLASSIFY
/SAVE CLUSTER DISTANCE
/PRINT INITIAL ANOVA CLUSTER DISTAN
/FILE='centroides iniciales.sav'.
```

* COMPARACION CON LOS ESTRATOS DE POBREZA *

CTABLES

```
/VLABELS VARIABLES=ESTRATO_P QCL_1 DISPLAY=LABEL
/TABLE ESTRATO_P [C][COUNT F40.0] BY QCL_1
/CATEGORIES VARIABLES=ESTRATO_P ORDER=A KEY=VALUE
EMPTY=INCLUDE TOTAL=YES POSITION=AFTER
/CATEGORIES VARIABLES=QCL_1 ORDER=A KEY=VALUE EMPTY=EXCLUDE
TOTAL=YES POSITION=AFTER
/TITLES
TITLE='COMPARACIÓN DE CONGLOMERADOS Y ESTRATOS DE POBREZA (1)'.
```

* CLASIFICACIÓN CON CENTROS FINALES *

```
QUICK CLUSTER NBI_V NBI_SB NBI_ED NBI_S
/MISSING=LISTWISE
/CRITERIA=CLUSTER(5)
/METHOD=CLASSIFY
/SAVE CLUSTER DISTANCE
/PRINT INITIAL ANOVA CLUSTER DISTAN
/FILE='centroides finales.sav'.
```

* COMPARACION CON LOS ESTRATOS DE POBREZA *

CTABLES

```
/VLABELS VARIABLES=ESTRATO_P QCL_2 DISPLAY=LABEL
/TABLE ESTRATO_P [C][COUNT F40.0] BY QCL_2
/CATEGORIES VARIABLES=ESTRATO_P ORDER=A KEY=VALUE
EMPTY=INCLUDE TOTAL=YES POSITION=AFTER
/CATEGORIES VARIABLES=QCL_2 ORDER=A KEY=VALUE EMPTY=EXCLUDE
TOTAL=YES POSITION=AFTER
/TITLES
TITLE='COMPARACIÓN DE CONGLOMERADOS Y ESTRATOS DE POBREZA (2)'.
```

7.7 ANEXO 7 – Código errores de muestreo

* ASISTENTE DE PREPARACIÓN DEL ANÁLISIS *

CSPLAN ANALYSIS

/PLAN FILE='C:\Users\LUIS FLORES\Desktop\T2014\errores de muestreo\errores1.csaplan'

/PLANVARS ANALYSISWEIGHT=factor

/PRINT PLAN

/DESIGN

/ESTIMATOR TYPE=EQUAL_WOR

/POPSIZE VALUE=2382499.

* FRECUENCIAS DE MUESTRAS COMPLEJAS *

CSTABULATE

/PLAN FILE='C:\Users\LUIS FLORES\Desktop\T2014\errores de muestreo\errores1.csaplan'

/TABLES VARIABLES=POBREZA_CLUSTER POBREZA_RED POBREZA_NBI
ESTRATO_P CLUSTER_1 CLUSTER_2 RED

/SUBPOP TABLE=dep DISPLAY=LAYERED

/CELLS TABLEPCT

/STATISTICS SE CV CIN(95) COUNT DEFF

/MISSING SCOPE=TABLE CLASSMISSING=EXCLUDE.

7.8 ANEXO 8 – Estadístico Kappa

Los coeficientes de concordancia son especialmente útiles para medir el grado de asociación entre dos conjuntos de datos. La medida más popular para evaluar la concordancia entre variables cualitativas es el coeficiente de Kappa de Cohen, formulado inicialmente para dos variables binarias en 1960. Este estadístico se desarrolló originalmente para medir la concordancia entre observadores, aplicándose particularmente a problemas de fiabilidad entre diagnósticos. En este contexto se han presentado sucesivas generalizaciones a lo largo del tiempo y su aplicación se ha extendido lejos del problema inicial, de hecho, en la actualidad se emplea como una medida de similaridad para datos categóricos (Fleiss 1981).

Supongamos que dos observaciones distintas se clasifican de forma independientemente en n ítems y un conjunto de k categorías nominales. El resultado de esta clasificación se puede resumir en la siguiente tabla:

	MÉTODO 2				
MÉTODO 1	1	2	...	k	Total
1	X_{11}	X_{12}	...	X_{1k}	X_{1*}
2	X_{21}	X_{22}	...	X_{2k}	X_{2*}
...
k	X_{k1}	X_{k2}	...	X_{kk}	X_{k*}
Total	X_{*1}	X_{*2}	...	X_{*k}	n

Cada valor x_{ij} representa el número de observaciones clasificadas por ambos métodos y sus diferentes categorías ($i, j=1, \dots, k$).

La proporción de coincidencias observadas está conformada por la diagonal principal y se representa de la siguiente manera:

$$P_0 = \frac{\sum_{i,j=1}^k x_{ij}}{n}; i = j$$

Por otro lado, la proporción de coincidencias esperadas por azar se define como:

$$P_e = \frac{1}{n^2} \sum_{i,j=1}^k x_{i*} x_{*j}; i = j$$

Entonces, para determinar si la concordancia observada es superior a la que se esperaría encontrar por azar resulta conveniente utilizar el índice de concordancia de kappa, que se define como:

$$K = \frac{P_0 - P_e}{1 - P_e}$$

El estadístico kappa presenta un valor de 1 cuando la concordancia observada es perfecta, un valor de 0 cuando la concordancia observada es igual a la concordancia esperada por azar y valores inferiores a 0 cuando la concordancia observada es inferior a la concordancia esperada por azar.

Para interpretar orientativamente qué significado tiene el valor del coeficiente que obtengamos al calcular sobre nuestros datos, Landis y Koch propusieron la siguiente clasificación:

Kappa	Grado de acuerdo
<0,0	Sin acuerdo
0,0 – 0,2	Insignificante
0,2 – 0,4	Bajo
0,4 – 0,6	Moderado
0,6 – 0,8	Bueno
0,8 - 1,0	Muy bueno

7.9 ANEXO 9 – Código del estadístico Kappa

* KAPPA – CONGLOMERADOS (CENTROS INICIALES) Y ESTRATOS DE POBREZA

```
CROSSTABS
/TABLES=ESTRATO_P BY CLUSTER_1
/FORMAT=AVALUE TABLES
/STATISTICS=KAPPA
/CELLS=COUNT
/COUNT ROUND CELL.
```

* KAPPA – CONGLOMERADOS (CENTROS FINALES) Y ESTRATOS DE POBREZA

```
CROSSTABS
/TABLES=ESTRATO_P BY CLUSTER_2
/FORMAT=AVALUE TABLES
/STATISTICS=KAPPA
/CELLS=COUNT
/COUNT ROUND CELL.
```

* KAPPA – RED NEURONAL BPN Y ESTRATOS DE POBREZA

```
CROSSTABS
/TABLES=ESTRATO_P BY RED
/FORMAT=AVALUE TABLES
/STATISTICS=KAPPA
/CELLS=COUNT
/COUNT ROUND CELL.
```

8

BIBLIOGRAFÍA

- Choque Aspiazu, G., Loza Herrera, R. y Mendez Quintanilla, R. (2009). Redes Neuronales Artificiales, Aplicaciones en MATLAB. La Paz, Bolivia.
- Freeman, J.A. and Skapura, D.M. (1993). Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación. Addison Wesley/Diaz de Santos. USA.
- Hecht-Nielsen, R. (1988). Neurocomputing: picking the human brain. IEEE. Spectrum, 25, No.3, 36-41.
- Hilera, J. y Martínez, V. (1995). Redes Neuronales Artificiales. Fundamentos, Modelos y aplicaciones. Rama. Madrid, España.
- Johnson, D.E. (2000). Métodos multivariados aplicados al análisis de datos. International Thomson Editores.
- Luque Martínez, T. (2000). Técnicas de análisis de datos en investigación de mercados. Ediciones Pirámide. Madrid, España.
- Martín, B. y Sanz, A. (2002). Redes Neuronales y Sistemas Difusos, 2ª ed. Alfaomega Rama. Madrid, España.
- Peña, D. (2002). Análisis de datos multivariantes. McGRAW-HILL. Madrid, España.
- Pérez, C. (2004). Técnicas de Análisis Multivariante de Datos, Aplicaciones con SPSS. Pearson Educación S. A. Madrid, España.