

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



TESIS DE GRADO

DETECTOR DE MENTIRAS MEDIANTE EL ANALISIS DE
ESTRÉS DE VOZ USANDO REDES NEURONALES
RECURRENTES

Tesis de grado para obtener el Título de Licenciatura en Informática
Mención Ingeniería de Sistemas Informáticos

POR: CARMELA SANDRA CALLICONDE CARRASCO
TUTOR: M.SC. ALDO RAMIRO VALDEZ ALVARADO

LA PAZ-BOLIVIA

2022

HOJA DE CALIFICACIONES

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

Tesis de grado:

DETECTOR DE MENTIRAS MEDIANTE EL ANALISIS DE ESTRÉS DE VOZ
USANDO REDES NEURONALES RECURRENTE

Presentado por: Carmela Sandra Calliconde Carrasco

Para optar por el grado Académico de Licenciada en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral.....

Nota Literal.....

Ha sido.....

Director de la carrera de informática: Ph.D. Hermenegildo Nogales Quispe

Tutor: M.Sc. Aldo Ramiro Valdez Alvarado

Tribunal: M.Sc. Franz Cuevas Quiroz

Tribunal: M.Sc. German Huanca Ticona

Tribunal: M.Sc. Franz Ramiro Gallardo Portanda

Tribunal: M.Sc. Jorge Humberto Teran Pomier



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON EXTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso del ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionado la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS, EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mis padres Limber y Carmela, por apoyarme, confiar y preocuparse en mí en todo momento.

A mi hermana Rosio, por apoyarme y sacarme una sonrisa en todo el transcurso de mi vida colegial y universitaria.

A mi entera familia, mis primos, tíos y abuelos. Cada integrante de mi familia, que con su granito de arena que me contribuyeron, pude culminar el presente proyecto. Por los consejos y el apoyo incondicional que me brindaron.

A mis amigos por todo el apoyo y su amistad incondicional.

AGRADECIMIENTOS

Agradezco a mi familia, por sus palabras de aliento, consejos, por todo su apoyo y amor que me brindan día a día.

A mi Docente tutor M.Sc. Aldo Ramiro Valdez Alvarado, quien con su gran calidad de persona, profesionalismo y experiencia acompañó en el desarrollo del presente proyecto, gracias por la paciencia y por los consejos que nos guiará a lo largo de nuestra vida profesional.

Al personal de la biblioteca, por su ayuda y colaboración.

A mis amigos, amigas que me acompañaron en todos estos años de estudio y sobre todo por su amistad sincera.

Muchas gracias a todos de todo corazón y que Dios siempre los bendiga.

carmelacallicondecarrasco@gmail.com

RESUMEN

Las Redes Neuronales Artificiales están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano.

Las Redes Neuronales Recurrentes son capaces de realizar una amplia variedad de tareas computacionales incluyendo el tratamiento de secuencias, la continuación de una trayectoria, la predicción no lineal y la modelación de sistemas dinámicos. Estas redes también se conocen como redes espacio-temporales o dinámicas, son un intento de establecer una correspondencia entre secuencias de entrada y de salida que no son más que patrones temporales.

El detector de mentiras analiza el estrés de voz con redes neuronales recurrentes para la detección de la mentira. , no existe un dataset con audios de estrés para lo cual se construye primeramente un dataset en base a SAVEE Y CREMA-D tomando en cuenta las emociones de enojo y miedo por la presencia de estrés en los mismos, para ser leído por la librería librosa, procesado por 40 coeficientes cepstrales de frecuencia de mel con lo cual obtiene un vector por audio, pasa al modelo para ser entrenado y validado. Se construye una aplicación donde se incluye el modelo para realizar pruebas, el cual examina el audio en el smartphone y luego realiza su respectiva predicción.

El proyecto emplea la metodología fundamental para la ciencia de datos, presenta 10 etapas.

Al culminar con las 10 etapas al proyecto se obtiene un 80% de exactitud en la detección de estrés.

Palabras claves: Redes neuronales recurrentes, exactitud, entrenamiento, Coeficientes cepstrales de frecuencia de mel, estrés, detector de mentiras.

Metodología: Metodología Fundamental para la Ciencia de Datos

ABSTRACT

Artificial Neural Networks are inspired by the biological neural networks of the human brain. They are made up of elements that behave in a similar way to the biological neuron in its most common functions. These elements are organized in a way similar to that of the human brain.

Recurrent Neural Networks are capable of performing a wide variety of computational tasks including sequence processing, trajectory continuation, nonlinear prediction, and modeling of dynamical systems. These networks are also known as space-time or dynamic networks, they are an attempt to establish a correspondence between input and output sequences that are nothing more than temporal patterns.

The lie detector analyzes voice stress with recurrent neural networks for lie detection. , there is no dataset with stress audios, for which a dataset based on SAVEE AND CREMA-D is first built, taking into account the emotions of anger and fear due to the presence of stress in them, to be read by the bookstore librosa , processed by 40 mel frequency cepstral coefficients with which it obtains an audio vector, passes to the model to be altered and validated. An application is built that includes the model for testing, which examines the audio on the smartphone and then makes its respective prediction.

The project uses the fundamental methodology for data science, presents 10 stages. At the end of the 10 stages of the project, an 80% accuracy in stress detection is obtained.

Keywords: Recurrent neural networks, accuracy, training, mel frequency cepstral coefficients, stress, lie detector.

Methodology: Foundational Methodology for Data Science

ÍNDICE

CAPÍTULO 1	MARCO INTRODUCTORIO	1
1.1.	INTRODUCCIÓN	1
1.2.	ANTECEDENTES	2
1.3.	PLANTEAMIENTO DEL PROBLEMA	4
1.3.1.	PROBLEMA CENTRAL	5
1.3.2.	PROBLEMAS SECUNDARIOS.....	5
1.4.	DEFINICIÓN DE OBJETIVOS	5
1.4.1.	OBJETIVO GENERAL	5
1.4.2.	OBJETIVOS ESPECÍFICOS.....	5
1.5.	HIPÓTESIS.....	6
1.5.1.	OPERACIONALIZACIÓN DE VARIABLES	6
1.6.	JUSTIFICACIÓN	6
1.6.1.	JUSTIFICACIÓN ECONÓMICA	6
1.6.2.	JUSTIFICACIÓN SOCIAL	6
1.6.3.	JUSTIFICACIÓN CIENTÍFICA	7
1.7.	ALCANCES Y LÍMITES	7
1.7.1.	ALCANCES.....	7
1.7.2.	LÍMITES.....	7
1.8.	APORTES	8
1.8.1.	PRÁCTICO.....	8
1.8.2.	TEÓRICO	8
1.9.	METODOLOGÍA	8
1.9.1.	METODOLOGÍA DE INVESTIGACIÓN.....	8
1.9.2.	METODOLOGÍA DE INGENIERÍA.....	8
CAPÍTULO 2	MARCO TEÓRICO	9
2.1.	INTRODUCCIÓN	9
2.2.	INTELIGENCIA ARTIFICIAL.....	9
2.3.	REDES NEURONALES	10
2.3.1.	VENTAJAS QUE OFRECEN LAS REDES NEURONALES	11
2.3.2.	LA NEURONA BIOLÓGICA.....	12
2.3.3.	LA NEURONA ARTIFICIAL.....	14
2.3.3.1.	ENTRADAS Y SALIDAS EN LA RED NEURONAL	15
2.3.3.2.	PESOS SINÁPTICOS.....	15
2.3.3.3.	REGLA DE PROPAGACIÓN.....	15
2.3.4.	FUNCIONES DE ACTIVACIÓN	16
2.3.4.1.	FUNCIÓN SOFTMAX.....	16
2.3.5.	ARQUITECTURA DE REDES NEURONALES	17
2.3.5.1.	FASE DE APRENDIZAJE.....	19
2.3.5.2.	FASE DE ENTRENAMIENTO	20

2.3.5.3. FASE DE OPERACIÓN.....	21
2.4. DEEP LEARNING	22
2.4.1. FUNCIONAMIENTO DEL DEEP LEARNING	23
2.5. REDES NEURONALES RECURRENTE.....	23
2.5.1. REDES LSTM	25
2.6. COEFICIENTES CEPSTRALES DE FRECUENCIA DE MEL.....	26
2.6.1. PROCESO DE EXTRACCIÓN DE MFCC	27
2.6.2. TRANSFORMADA DE FOURIER	29
2.6.3. BANCO DE FILTROS EN LA ESCALA DE MEL	30
2.6.4. LOGARITMO.....	31
2.6.5. TRANSFORMADA DISCRETA DEL COSENO	31
2.7. METODOLOGÍA FUNDAMENTAL PARA LA CIENCIA DE DATOS.....	32
2.7.1. INTRODUCCIÓN	32
2.7.2. METODOLOGÍA Y ETAPAS	33
2.7.2.1. ETAPA 1: COMPRESIÓN DEL NEGOCIO	34
2.7.2.2. ETAPA 2: ENFOQUE ANALÍTICO	34
2.7.2.3. ETAPA 3: REQUISITOS DE DATOS.....	35
2.7.2.4. ETAPA 4: RECOPIACIÓN DE DATOS.....	35
2.7.2.5. ETAPA 5: COMPRESIÓN DE DATOS	35
2.7.2.6. ETAPA 6: PREPARACIÓN DE DATOS	36
2.7.2.7. ETAPA 7: MODELADO.....	36
2.7.2.8. ETAPA 8: EVALUACIÓN	37
2.7.2.9. ETAPA 9: IMPLEMENTACIÓN.....	37
2.7.2.10. ETAPA 10: RETROALIMENTACIÓN.....	38
2.8. LA MENTIRA Y EL ENGAÑO.....	38
2.9. PSICOFISIOLOGÍA DE LAS EMOCIONES.....	39
2.10. POLÍGRAFO	40
2.11. ANÁLISIS DE ESTRÉS DE VOZ	41
2.12. PRESENCIA DE ESTRÉS EN LAS EMOCIONES DE ENOJO Y MIEDO ..	43
2.13. LA VOZ HUMANA	44
2.13.1. CARACTERÍSTICAS DE LA VOZ HUMANA	46
2.14. SONIDO.....	46
2.14.1. CARACTERÍSTICAS DEL SONIDO	47
2.15. PYTHON	48
2.16. TENSORFLOW.....	48
2.17. ANDROID STUDIO.....	49
2.18. FLASK.....	50
CAPÍTULO 3 MARCO APLICATIVO	51
3.1. INTRODUCCIÓN	51
3.2. ETAPAS DE LA METODOLOGÍA FUNDAMENTAL PARA LA CIENCIA	

DE DATOS	53
3.2.1. ETAPA 1: COMPRESIÓN DEL NEGOCIO	53
3.2.2. ETAPA 2: ENFOQUE ANALÍTICO	56
3.2.3. ETAPA 3: REQUISITOS DE DATOS.....	57
3.2.4. ETAPA 4: RECOPIACIÓN DE DATOS.....	58
3.2.5. ETAPA 5: COMPRESIÓN DE DATOS	60
3.2.6. ETAPA 6: PREPARACIÓN DE DATOS	63
3.2.6.1. SELECCIONAR LOS DATOS	63
3.2.6.2. LIMPIEZA DE DATOS	63
3.2.7. ETAPA 7: MODELADO.....	66
3.2.7.1. TÉCNICA DE MODELADO	66
3.2.7.2. PLAN DE PRUEBA	66
3.2.7.3. CONSTRUCCIÓN DEL MODELO.....	68
3.2.8. ETAPA 8: EVALUACIÓN	69
3.2.8.1. EVALUAR LOS RESULTADOS	70
3.2.9. ETAPA 9: IMPLEMENTACIÓN.....	71
3.2.9.1. PROTOTIPO.....	72
3.2.10. ETAPA 10: RETROALIMENTACIÓN.....	75
CAPÍTULO 4 PRUEBA DE HIPOTESIS.....	76
4.1 DATOS DE VALIDACIÓN.....	76
4.2. ANÁLISIS DE RESULTADOS DE VALIDACIÓN.....	76
4.3. DEMOSTRACIÓN DE LA HIPÓTESIS	77
CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES.....	79
5.1. CONCLUSIONES	79
5.2. RECOMENDACIONES.....	80
BIBLIOGRAFÍA	81

ÍNDICE DE FIGURAS

Figura 2.1. Estructura de una neurona biológica.....	13
Figura 2.2. Neurona artificial.....	14
Figura 2.3. Softmax y entropía cruzada categórica.....	13
Figura 2.4. Función softmax.....	14
Figura 2.5. Capas en una red neuronal artificial.....	18
Figura 2.6. Deep Learning.....	22
Figura 2.7. Estructura de una red neuronal recurrente.....	25
Figura 2.8. Proceso de extracción MFCC.....	28
Figura 2.9. Ventana Hamming.....	45
Figura 2.10. Banco de filtros de escala de Mel.....	45
Figura 2.11. Escala de Mel.....	31
Figura 2.12. Proceso y fórmulas para obtener MFCC.....	34
Figura 2.13. Metodología fundamental para la ciencia de datos.....	34
Figura 2.14. Gráfica sin estrés.....	34
Figura 2.15. Gráfica con estrés.....	34
Figura 2.16. La voz humana.....	34
Figura 3.1. Proceso para realizar el proyecto.....	53
Figura 3.2. Pre-procesamiento y entrenamiento.....	53
Figura 3.3. Formato de audios de la base de datos SAVEE.....	53
Figura 3.4. Formato de audios de la base de datos SAVEE.....	53
Figura 3.5. Uso del detector de mentiras.....	55
Figura 3.6. Estadística de la importancia del detector de mentiras en distintas áreas.	55
Figura 3.7. Recursos de hardware.....	56
Figura 3.8. Tiempo estimado del desarrollo del proyecto.....	57
Figura 3.9. WAV vs MP3.....	58
Figura 3.10. Estructura de los audios SAVEE.....	59

Figura 3.11. Cantidad total de datos en el Dataset a usar	61
Figura 3.12. Audios con estrés y sin estrés	61
Figura 3.13. Espectrograma del estado con estrés del primer caso.....	62
Figura 3.14. Espectrograma del estado sin estrés del primer caso.....	62
Figura 3.15. Importación de paquetes para visualización del audio	63
Figura 3.16. Extracción de características MFCC.	64
Figura 3.17. Proceso de selección y búsqueda de cada archivo para extraer MFCC. 65	
Figura 3.18. Resultado de la extracción de MFCC para cada archivo	66
Figura 3.19. Código para la división del dataset.....	67
Figura 3.20. División de datos y forma de los datos.....	67
Figura 3.21. Modelo LSTM.	68
Figura 3.22. Model Summary	69
Figura 3.23. Entrenamiento.....	69
Figura 3.24. Evaluación del modelo	70
Figura 3.25. Matriz de confusión	70
Figura 3.26. Pantalla Lie Detector.	72
Figura 3.27. Selección de archivos.	73
Figura 3.28. Pantalla del detector cuando existe estrés de voz	73
Figura 3.29. Pantalla del detector cuando no existe estrés de voz.	74
Figura 3.30. Aceptación del detector de mentiras.....	75
Figura 4.1. Matriz de confusión positivos y negativos	76
Figura 4.2. Matriz de confusión del modelo	77

INDICE DE TABLAS

Tabla 3.1. Base de datos SAVEE.....	59
Tabla 3.2. Resultados.....	74

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

El presente trabajo de tesis tiene como finalidad realizar un detector de mentiras analizando el estrés de voz con redes neuronales recurrentes para la detección del engaño. Saber la verdad y descubrir el engaño son objetivos compartidos por numerosas instituciones entre las cuales está la justicia, empresas privadas como públicas, en lo profesional, sociales o familiar. Las Redes Neuronales Recurrentes (RNR) permitirán el análisis de sonidos para poder determinar si una persona miente o dice la verdad realizando primeramente un pre-procesamiento y después un entrenamiento hasta lograr un modelo factible. Esto se implementará en Python con la ayuda de la librería Keras de TensorFlow para la creación del modelo que se implementará en una aplicación móvil.

El diccionario de la Real Academia de la lengua española define la mentira como: “la expresión o manifestación contraria a lo que se sabe, se cree o se piensa”, lo que abarca: una falsedad genuina, una verdad selectiva, exagerar una verdad, o incluso, manifestar una verdad cuando la intención es engañar o causar una acción en contra de los intereses de quien (quienes) son objeto de la mentira. La mentira se ha convertido en un objeto de estudio de numerosas investigaciones, intentado entender su origen, así como el efecto de ésta en las relaciones sociales. (RAE, 2019).

La sociedad tiene la necesidad de establecer la verdad en situaciones graves o delicadas: por ejemplo, para saber quién ha cometido un crimen o dónde está el dinero producto de un fraude. Las personas, en momentos de su vida, experimentan también la necesidad de saber si lo que escuchan, en temas que consideran personalmente muy importantes, es mentira o verdad. Por otro lado, el mundo de los negocios se basa en la confianza mutua, que puede destruirse fácilmente con una mentira, intento de engaño. Una mala información puede llevar a una merma en los beneficios e incluso a la pérdida de la empresa. Saber si nos dicen la verdad nos preocupa a todos. Está comprobado el estrés y la emoción intensa se manifiesta a través del aumento en la actividad fisiológica general, que es más acusada en el sistema nervioso vegetativo, en las funciones corporales que dependen de él. La voz es una herramienta

importante para poder transmitir y comunicar el mensaje por lo que con los parámetros adecuados se puede descubrir si una persona miente. El análisis de estrés de voz se usa como reemplazo de los polígrafos tradicionales y se basa en la medición de la frecuencia de la voz humana en el rango de 8 a 12 hercios Hz, para considerar que científicamente está demostrado que dicha voz es sensible a la honestidad en el rango aludido. (Martínez, 2005).

1.2. ANTECEDENTES

El primer intento de detector de mentiras concebido tecnológicamente fue patentado en 1905 por Henry Lavery en Estados Unidos siguiendo los principios de la frenología del austríaco Franz Joseph Gall (1758-1828), que era una teoría psicológica que pretendía estudiar el carácter y las funciones intelectuales del hombre basándose en el análisis de la conformación exterior del cráneo. Inspirado en ella, Lavery inventó el psicógrafo. El aparato consistía en 1.954 piezas montadas en un bastidor de metal, conectadas, mediante una correa movida por un motor, con el interior de una caja de madera que contenía las definiciones de 32 rasgos psicológicos. Cada uno de estos rasgos estaba clasificado de 1 (deficiente) a 5 (muy elevado), de modo que había 160 estados posibles, pero un número casi ilimitado de combinaciones. El sujeto se sentaba en una silla conectada con la máquina, el casco se bajaba y ajustaba. El casco tenía 32 sondas, cada una con cinco puntos de contacto en la cabeza. El operador tiraba de una palanca que activaba el motor, el aparato recibía las señales de baja tensión del casco y se obtenía la puntuación imprimiendo la definición apropiada para cada rasgo. (Muñoz, 2011).

Si el invento de Lavery se ha convertido hoy día en una anécdota, el aparato yace en el Museum of Questionable Medical Devices, no sucede lo mismo con el instrumento que en 1938 presentó Leonard Keeler, del Departamento de Policía de Berkeley (California), cuya ideación original es atribuida a J.A. Larson, del mismo Departamento. Este instrumento, denominado polígrafo y pronto popularizado como detector de mentiras, es un equipo fisiológico electrónico consistente en un aparato de medición utilizado para el registro de respuestas fisiológicas. Generalmente registra las variaciones de la presión arterial, el ritmo cardíaco, la frecuencia respiratoria y la respuesta galvánica o conductancia de la piel. Estas

respuestas fisiológicas son medidas por el aparato simultáneamente a un interrogatorio verbal muy específico que se administra a la persona sometida a la prueba. (Muñoz, 2011).

Después de desarrollarse el polígrafo se desarrolló el VSA. Son instrumentos tecnológicos desarrollados por las Fuerzas Armadas de los Estados Unidos. El VSA recomendado al igual que el polígrafo por el Congreso Americano después de los hechos de septiembre 11, en acuerdo al Acta e implementación de recursos de Seguridad de Aviación y Transporte del 16 de noviembre 2001. El VSA es un programa que detecta, mide y despliega el nivel de estrés psicológico a través de la voz humana y el polígrafo lo hace a través de diferentes mediciones, entre ellas el cardio, los neumas y la respuesta galvanometría. (Muñoz, 2011).

Existen investigaciones sobre detectores de mentiras entre ellos podemos mencionar:

- “DETECTAR LA MENTIRA A TRAVÉS DE LA VOZ”. Sandra Aller Arias, 2018. Universidad de Salamanca. El objetivo del trabajo consiste en confirmar, de manera contrastable, la hipótesis de que la mentira se puede detectar analizando la voz, ya que el habla es el instrumento principal utilizado por los seres humanos en sus relaciones interpersonales.
- “DETECCIÓN DE MENTIRAS POR ANÁLISIS DE IMÁGENES TÉRMICAS DE ROSTROS”. Hernán Belalcazar Ramírez, Sebastian Bedoya Echevery, 2015. Universidad del Valle, Facultad de Ingeniería, Escuela de Ingeniería Eléctrica y Electrónica, programa de Ingeniería Electrónica. Santiago de Cali. En este trabajo se presenta el diseño, desarrollo e implementación de un sistema de visión artificial para la detección de mentiras basado en el análisis del rostro por medio de imágenes térmicas.
- “MEDICIÓN PSICOFISIOLÓGICA: ANÁLISIS DE ESTRÉS DE VOZ EN CREDIBILIDAD DE TESTIMONIO EN INTERNOS DEL CE.RE.SO. DE TEKAX, YUCATÁN”. Jose Luis Chi Zaldivar, 2012. Universidad Autónoma Yucatán, Facultad de Psicología. México. En este trabajo se empleó el equipo de Análisis de Estrés de Voz (VSA; por sus siglas en ingles) para la evaluación de la

credibilidad de testimonio en una muestra de 24 internos del Centro de Reinserción Social de Tekax, Yucatán; los cuales se encuentran internados por haber cometido uno de los siguientes delitos: violación, homicidio y robo.

- “APLICACIONES DE REDES NEURONALES RECURRENTE Y MODELOS DE SERIES DE TIEMPO BAYESIANOS A LA PREDICCIÓN DE RENTABILIDAD DE FONDOS DE PENSIONES”. Sergio Esteban Altamirano Pontigo, 2020. Universidad de Concepción, Facultad de Ciencias Físicas y Matemáticas. Chile. En este trabajo se busca desarrollar un estudio comparativo sobre la aplicación de técnicas de predicción para series de tiempo a la rentabilidad de los fondos de pensiones. Específicamente, implementar y evaluar el rendimiento de las redes neuronales recurrentes y modelos bayesianos de series temporales sobre los fondos A y E con dos metodologías distintas.
- “REDES DE NEURONAS RECURRENTE PARA EL RECONOCIMIENTO DE PATRONES TEMPORALES”. Francisco Javier Segovia Pérez, 1992. Universidad Politécnica de Madrid, Facultad de Informática. España. El propósito fundamental de esta Tesis es el estudio de las redes de neuronas recurrentes aplicadas al reconocimiento de patrones temporales. A través de una revisión de los conocimientos que existen actualmente sobre redes recurrentes, se llegarán a unas conclusiones fundamentales respecto a su capacidad y limitaciones actuales.

1.3. PLANTEAMIENTO DEL PROBLEMA

Todas las personas tienen interés en encontrar un método eficaz para detectar cuándo una persona miente o cuándo dice la verdad. El campo donde es utilizado y estudiado con más interés es en el ámbito forense. Aunque hay numerosos procedimientos para detectar la mentira, éstos se han centrado en investigar la fiabilidad del polígrafo en el ámbito judicial, dejando de lado la investigación de otros métodos alternativos. (Aller, 2018).

Según Martínez (2005) la mentira y el engaño forman parte, se quiera o no, de las relaciones humanas en todos los ámbitos, tanto privados como públicos, y en todas sus vertientes, ya sean relativas a la amistad, el afecto, la política o la economía.

1.3.1. PROBLEMA CENTRAL

¿Cómo detectar si una persona miente?

1.3.2. PROBLEMAS SECUNDARIOS

- No hay herramientas tecnológicas confiables para detectar la mentira a un costo razonable, lo que ocasiona que sea difícil acceder a ellas.
- No hay ninguna aplicación móvil para detectar la mentira que sea confiable, lo que ocasiona que den resultados erróneos.
- No hay algoritmo disponible para el análisis de estrés de voz, lo que ocasiona que no se pueda detectar la mentira mediante el estrés.
- No se cuenta con una estructura definida para el análisis de estrés de voz, lo que ocasiona que no se pueda realizar una correcta predicción.
- No hay evaluación ágil e inmediata en la declaración de testimonios, lo que ocasiona la demora en la solución de un caso.

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un detector de mentiras mediante el análisis de estrés de voz.

1.4.2. OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación móvil que tendrá un costo razonable.
- Entrenar las redes neuronales recurrentes para obtener datos correctos y confiables.
- Diseñar una red neuronal recurrente para detectar si una persona miente o dice la verdad.
- Realizar el pre-procesamiento de las grabaciones extraídas, para el análisis de contenido.
- Agilizar la toma de declaraciones mediante el modelo con RNR para obtener la

veracidad del testimonio de manera inmediata.

1.5. HIPÓTESIS

El uso de Redes Neuronales Recurrentes, permite al detector de mentiras realizar un análisis de estrés en la voz y coadyuvar a la evaluación si una persona dice o no lo verdad con un grado de confiabilidad del 80%.

1.5.1. OPERACIONALIZACIÓN DE VARIABLES

De la hipótesis planteada se obtienen las variables independientes y las variables dependientes:

- **Variables dependientes.** - Detector de mentiras.
- **Variables independientes.** - Estrés de voz.
- **Variables intervinientes.** - Redes neuronales recurrentes

1.6. JUSTIFICACIÓN

1.6.1. JUSTIFICACIÓN ECONÓMICA

El desarrollo del detector de mentiras económicamente es factible, los gastos no son elevados en comparación con los beneficios que podrá brindar a los usuarios que lo utilicen, principalmente por su contribución a las empresas y departamentos de investigación judicial o policial. El uso de dispositivos móviles representa bajo presupuesto por lo que el detector de mentiras reflejaría el 5% de costos con relación al uso del VSA o de un polígrafo los cuales son más costosos, la implementación del modelo con RNR tendrán el mismo resultado e impacto.

1.6.2. JUSTIFICACIÓN SOCIAL

El detector de mentiras basado en el análisis de estrés de voz, será una herramienta importante para la detección de engaños, el cual ayudará a instituciones como: la justicia como en departamentos de investigación judicial o policial. En empresas, para seleccionar a su personal, comprobando la honestidad de las personas que ingresen, o bien evitando la contratación de personal susceptible a cometer actos ilícitos. También en el ámbito familiar,

profesional o social.

1.6.3. JUSTIFICACIÓN CIENTÍFICA

El presente trabajo pretende contribuir a la sociedad con ayuda de la tecnología, desarrollando un detector de mentiras que será una herramienta alternativa, rápida y eficiente, el cual permitirá optimizar la evaluación si una persona dice o no la verdad. El motivo por el cual se realizará el detector de mentiras mediante el análisis de estrés de voz, se centra en su uso amplio en todos los campos de investigación, además no presentará incomodidad por el uso de cables como es el caso de un polígrafo. Por lo que se usará Redes Neuronales Recurrentes ya que son muy potentes al analizar secuencias como el sonido.

1.7. ALCANCES Y LÍMITES

1.7.1. ALCANCES

- Para el manejo de Redes Neuronales Recurrentes se utilizará la librería keras de Tensorflow.
- Podrá analizar el estrés de voz de cualquier persona.
- Procesará e interpretará las grabaciones de manera inmediata.
- Crear un dataset con audios de emociones en base al enojo y miedo.
- Diseñar una aplicación para que interactue con el modelo dando predicciones.
- Lograr un modelo óptimo para el análisis de estrés de voz.

1.7.2. LÍMITES

- El detector solo estará disponible para celulares Android.
- El detector solo estará disponible en el idioma español.
- El detector solo interpretará grabaciones de voz.
- Las grabaciones permitidas son en extensión wav.
- El detector no permitirá la manipulación de resultados.

1.8. APORTES

1.8.1. PRÁCTICO

Se desarrollará una aplicación móvil que detectará la mentira mediante el análisis de estrés de voz con ayuda de un modelo a base de RNR, el cual se desarrollará en Python utilizando librerías de TensorFlow.

1.8.2. TEÓRICO

El desarrollo del detector de mentiras se llevará a cabo la combinación de dos metodologías: el científico y la metodología fundamental para la ciencia de datos. El primero para la investigación del estrés de voz y el segundo para el desarrollo del modelo con redes neuronales.

1.9. METODOLOGÍA

1.9.1. METODOLOGÍA DE INVESTIGACIÓN

En la investigación se emplea el Método Científico, para realizar diferentes pruebas controladas que permitan el buen diseño de la Red Neuronal y el modelado ideal en el pre-procesado de los sonidos, es así que la investigación responde a una investigación de tipo exploratorio a descriptivo.

1.9.2. METODOLOGÍA DE INGENIERÍA

Para el diseño y desarrollo del detector de mentiras se utilizará lo siguiente:

- Para el diseño y desarrollo del detector de mentiras se utilizará la metodología fundamental para la ciencia de datos usada con grandes volúmenes de datos, la incorporación de la analítica de texto en el modelado predictivo y la automatización de algunos procesos.
- Las Redes Neuronales Recurrentes se utilizará como técnica para realizar el modelo para la detección de mentiras.
- Las herramientas que se utilizarán para el modelo serán Python y TensorFlow.
- Por último, se utilizará Android Studio y Flask para la aplicación de prueba.

CAPÍTULO 2

MARCO TEÓRICO

2.1. INTRODUCCIÓN

2.2. INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial (IA) ha despertado bastante interés en la actualidad, debido a su enorme campo de aplicación. La búsqueda de mecanismos que nos ayuden a comprender la inteligencia y realizar modelos y simulaciones de estos, es algo que ha motivado a muchos científicos a elegir esta área de investigación. (Ponce et al., 2014).

Es una de las ramas de las ciencias de la computación que más interés ha despertado en la actualidad, debido a su enorme campo de aplicación. La búsqueda de mecanismos que nos ayuden a comprender la inteligencia y realizar modelos y simulaciones de estos, es algo que ha motivado a muchos científicos a elegir esta área de investigación. (Ponce et al., 2014).

El origen inmediato del concepto y del desarrollo de la IA se remonta a la intuición del genio matemático inglés Alan Turing y el apelativo Inteligencia Artificial se debe a McCarthy quien organizó una conferencia en el Dartmouth College (Estados Unidos) para discutir la posibilidad de construir máquinas inteligentes; a esta reunión asistieron científicos investigadores de conocida reputación en el área de las ciencias computacionales como: Marvin Minsky, Nathaniel Rochester, Claude Shannon, Herbert Simon y Allen Newell. (Ponce et al., 2014).

Como resultado de la conferencia, se establecieron los primeros lineamientos de la hoy conocida como Inteligencia Artificial; aunque anteriormente ya existían algunos trabajos relacionados. (Ponce et al., 2014).

Según Ponce (2014) La IA tuvo que lidiar con el conflicto de que no existía una definición clara y única de inteligencia; así es que no es de sorprender que aún en la actualidad, no exista una definición única de ella. Así como la Psicología ha identificado diferentes tipos de inteligencia humana (emocional, interpersonal, musical, lingüística, kinestésica, espacial, etc.), las distintas definiciones de la inteligencia artificial hacen énfasis en diferentes aspectos; aunque existen similitudes entre ellas. A continuación, se presentan algunas de las definiciones iniciales de esta área.

- Estudio sobre la computación donde se observa que una maquina es capaz de percibir, razonar y actuar.
- Ciencia de la obtención de máquinas que logren hacer cosas que requerirían inteligencia si las hiciesen los humanos.
- Nuevo esfuerzo excitante que logre que la computadora piense, máquinas con mentes, en el sentido completo y literal.
- Rama de la ciencia computacional preocupada por la automatización de la conducta inteligente.
- Maquina Inteligente es la que realiza el proceso de analizar, organizar, y convertir los datos en conocimiento, donde el conocimiento del sistema es información estructurada adquirida y aplicada para reducir la ignorancia o la incertidumbre sobre una tarea específica a realizar por esta.

Originalmente la Inteligencia Artificial se construyó en base a conocimientos y teorías existentes en otras áreas del conocimiento. Algunas de las principales fuentes de inspiración y conocimientos que nutrieron a esta área son las ciencias de la computación, la filosofía, la lingüística, las matemáticas y la psicología. Cada una de estas ciencias contribuyó no solamente con los conocimientos desarrollados en ellas, sino con sus herramientas y experiencias también; contribuyendo así a la gestación y desarrollo de esta nueva área del conocimiento. (Ponce et al., 2014).

2.3. REDES NEURONALES

Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks) están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano. (Basogain, s.f.).

Según Basogain (s.f.) las ANN al margen de parecerse al cerebro presentan una serie de características propias del cerebro. Por ejemplo, las ANN aprenden de la experiencia,

generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

- **Aprender.** Adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las ANN pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes.
- **Generalizar.** Extender o ampliar una cosa. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión.
- **Abstraer.** Aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos.

2.3.1. VENTAJAS QUE OFRECEN LAS REDES NEURONALES

Según Matich (2001) debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante entre otros. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

- **Aprendizaje Adaptativo.** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- **Auto-organización.** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- **Tolerancia a fallos.** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso

sufriendo un gran daño.

- **Operación en tiempo real.** Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- **Fácil inserción dentro de la tecnología existente.** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.

2.3.2. LA NEURONA BIOLÓGICA

Una neurona biológica es una célula nerviosa eléctricamente excitable e interconectada dentro del cerebro que procesan y transmiten información a través de señales eléctricas y químicas. Fue Ramón Cajal quien descubrió la estructura celular (neurona) del sistema nervioso. Defendió la teoría de que las neuronas se interconectaban entre sí de forma paralela, y no formando un circuito cerrado como el sistema sanguíneo. Una neurona cerebral puede recibir unas 10.000 entradas y enviar a su vez su salida a varios cientos de neuronas. (García, 2019).

La conexión entre neuronas se llama sinapsis. No es una conexión física, sino que hay unos 2 mm de separación. Son conexiones unidireccionales, en la que la transmisión de la información se hace de forma eléctrica en el interior de la neurona y de forma química entre neuronas; gracias a unas sustancias específicas llamadas neurotransmisora. No todas las neuronas son iguales, existen muchos tipos diferentes según el número de ramificaciones de sus dendritas, la longitud del axón y otros detalles estructurales. Sin embargo, como hemos visto, todas ellas operan con los mismos principios básicos. Estas neuronas están conectadas entre sí para formar una red neuronal en el cerebro. Una neurona se compone de tres partes: cuerpo celular, las dendritas y el axón, como se puede ver en la Figura 2.1. (García, 2019).

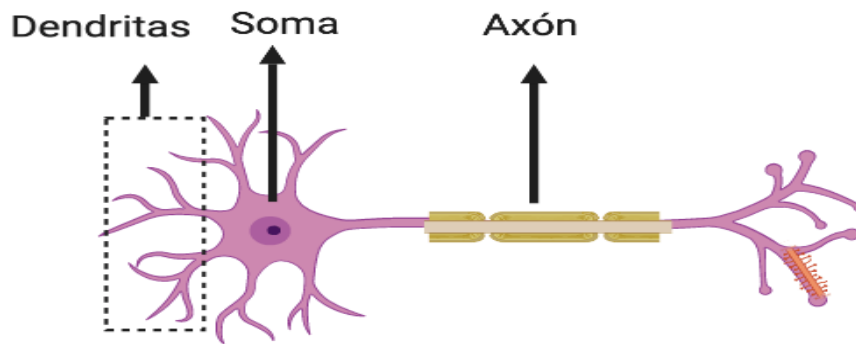


Figura 2.1. Estructura de una neurona biológica.

Fuente: García, 2019.

- Cuerpo Celular (Soma): Tiene como función principal sintetizar la mayoría de las proteínas que se encuentran en la neurona.
- Núcleo: Contiene el material genético en forma de cromosomas.
- Dendritas: Son prolongaciones protoplasmáticas, cortas y ramificadas que surgen desde el cuerpo celular. Estas prolongaciones se encargan de recibir y procesar los impulsos que llegan de otras neuronas para transmitirlos al cuerpo celular.
- Axón: Se trata de una prolongación caracterizada por ser única y larga, pues en algunas ocasiones tiende a medir 1 metro de longitud. Esta prolongación se origina en el área del cuerpo celular. Tiene como función extraer el impulso resultante desde cuerpo celular y transmitirlo a otra zona del sistema

La comunicación entre neuronas se llama sinapsis la cual se da en la terminación del axón y las dendritas de la siguiente neurona liberando una sustancia, esta cambia el balance de iones (átomos cargados electrónicamente) entre el interior y el exterior de la membrana celular. Cuando este cambio alcanza un nivel umbral, este efecto se expande a través de la membrana de la célula hasta el axón. Cuando alcanza al axón, se inicia un potencial de acción. Cuando el potencial de acción alcanza la terminación del axón, causa diminutas burbujas químicas llamadas vesículas que descargan su contenido en el salto sináptico. (García, 2019).

2.3.3. LA NEURONA ARTIFICIAL

Las neuronas artificiales son modelos que tratan de simular el comportamiento de las neuronas biológicas. Cada neurona se representa como una unidad de proceso que forma parte de una entidad mayor, la red neuronal. (Aguilar, 1999).

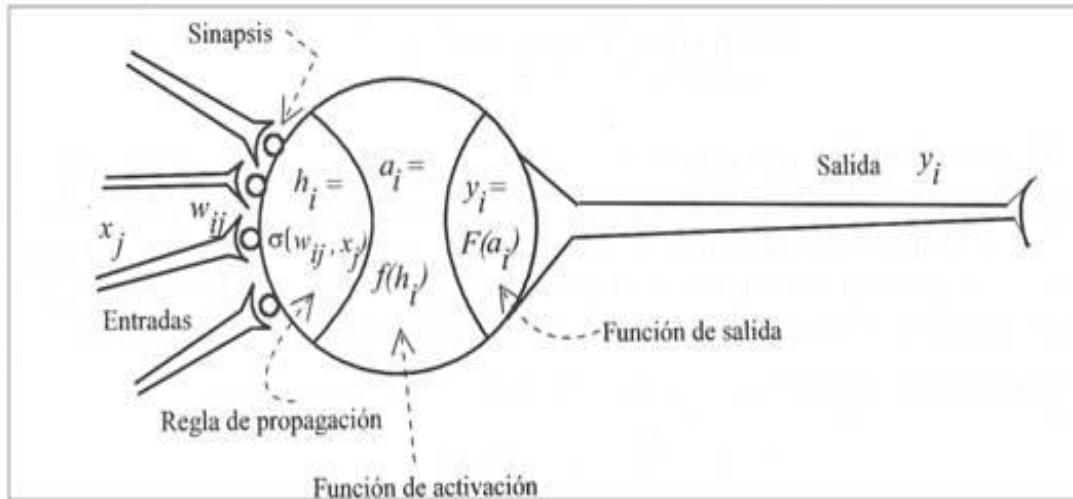


Figura 2.2. Neurona artificial.

Fuente: Aguilar, 1999.

La Figura 2.2. Muestra los elementos que componen una neurona artificial.

- Conjunto de entradas, $x_j(t)$. Estas pueden ser provenientes del exterior o de otras neuronas artificiales.
- Pesos sinápticos, w_{ij} . Representa el grado de comunicación entre la neurona artificial j y la neurona artificial i . Pueden ser excitadores o inhibidores.
- Regla de propagación, $\sigma_i(w_{ij}, x_j(t))$. Integra la información proveniente de las distintas neuronas artificiales y proporciona el valor del potencial postsináptico de la neurona i .
- Función de activación, $f_i(a_i(t-1), h_i(t))$. Provee el estado de activación actual de la neurona i .
- Función de salida, $F_i(a_i(t))$. Representa la salida actual de la neurona i . De esta forma,

la salida producida por una neurona i , para un determinado instante de tiempo t puede ser escrita en forma general de la siguiente manera:

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma_i(w_{ij}, x_j(t))])$$

2.3.3.1. ENTRADAS Y SALIDAS EN LA RED NEURONAL

Las entradas y salidas de una neurona pueden ser clasificadas en dos grandes grupos, binarias o continuas. Las neuronas binarias (digitales) sólo admiten dos valores posibles. En general en este tipo de neurona se utilizan los siguientes dos alfabetos $\{0,1\}$ o $\{-1,1\}$. Por su parte, las neuronas continuas (analógicas) admiten valores dentro de un determinado rango, que suelen definirse como $[-1,1]$. La selección del tipo de neurona a utilizar depende de la aplicación y del modelo a construir. (Aguilar, 1999).

2.3.3.2. PESOS SINÁPTICOS

El peso sináptico w_{ij} define la fuerza de una conexión sináptica entre dos neuronas, la neurona presináptica i y la neurona postsináptica j . Los pesos sinápticos pueden tomar valores positivos, negativos o cero. En caso de una entrada positiva, un peso positivo actúa como excitador, mientras que un peso negativo actúa como inhibidor. En caso de que el peso sea cero, no existe comunicación entre el par de neuronas. Mediante el ajuste de los pesos sinápticos la red es capaz de adaptarse a cualquier entorno y realizar una determinada tarea. (Aguilar, 1999).

2.3.3.3. REGLA DE PROPAGACIÓN

Según Aguilar (1999) la regla de propagación determina el potencial resultante de la interacción de la neurona i con las N neuronas vecinas. El potencial resultante h_i se puede expresar de la siguiente manera:

$$h_i(t) = \sigma_i(w_{ij}, x_j(t))$$

La regla de propagación más simple y utilizada consiste en realizar una suma de las entradas ponderadas con sus pesos sinápticos correspondientes.

$$h_i(t) = \sum w_{ij} * x_j(t)$$

2.3.4. FUNCIONES DE ACTIVACIÓN

La función de activación determina, como su propio nombre indica, el nivel de activación que alcanza cada neurona una vez que ha recibido los impulsos aferentes. Estas funciones ostentan un rol muy importante en la determinación del poder computacional de la red neuronal. (Rodríguez, 2021).

2.3.4.1. FUNCIÓN SOFTMAX

La función softmax se usa exclusivamente en las neuronas de salida de la red, y cuyo objeto es que la salida de todas ellas sume uno. A este proceso se le denomina normalización.

Esta función por lo general es utilizada en sistemas que clasifican datos de entrada (por ejemplo, imágenes de flores) en categorías (la especie a la que pertenecen). En ellos, a cada categoría le corresponde a una neurona de salida diferente, y los valores softmax de esta última señalan la probabilidad de que los datos de entrada pertenezcan a la misma. Las entradas de la función softmax reciben el nombre de puntuaciones (*scores*), que se miden en *logits*. (Rodríguez, 2021).

Matemáticamente, tiene una expresión tal que: simple:

$$f(x_z) = \frac{e^{x_z}}{\sum e^{x_i}}$$

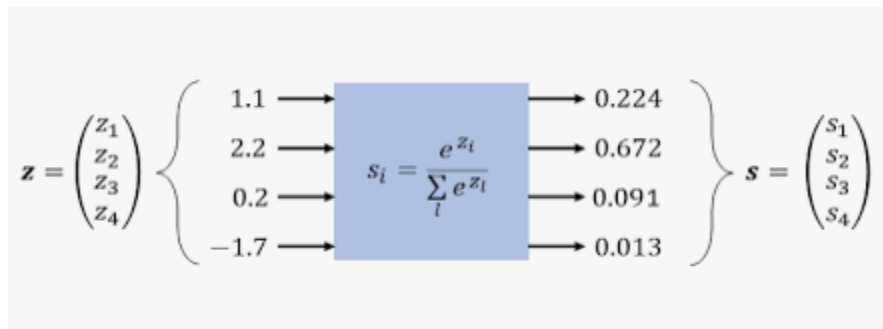


Figura 2.3. Softmax y entropía cruzada categórica.

Fuente: Rodríguez, 2021.

Siendo xz la función de excitación de la neurona que el entrenador determina como correcta y x_i la de cualquier neurona de la capa de salida.

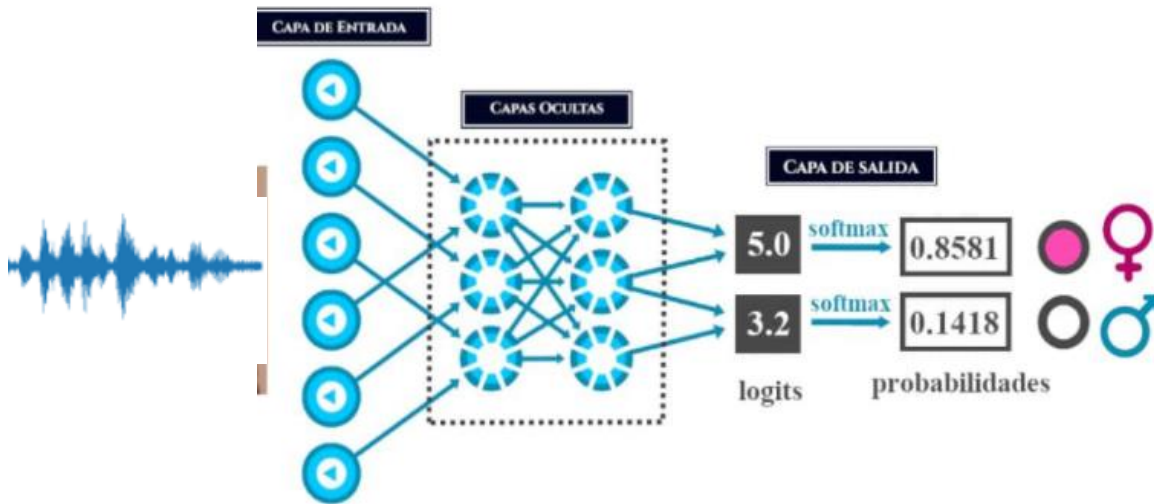


Figura 2.4. Función Softmax.

Fuente: Rodríguez, 2021.

2.3.5. ARQUITECTURA DE REDES NEURONALES

La topología de la red está determinada por el patrón de conexión entre las neuronas, su eventual jerarquización por capas, la intensidad y sentido de las distintas interconexiones. De acuerdo a la topología de la red se clasifican por el número de capas o niveles de neuronas, número de neuronas por capa y grado, tipo de conectividad entre las mismas. Dentro de una red neuronal, los elementos de procesamiento se encuentran agrupados por capas, una capa es una colección de neuronas. (Aguilar, 1999).

En la Figura 2.5. Se muestra la estructura de las capas y su ubicación en una RNA.

- Capa de entrada: Recibe las señales de la entrada de la red, algunos autores no consideran el vector de entrada como una capa porque no se lleva a cabo ningún proceso.
- Capas ocultas: Son aquellas que no tienen contacto con el medio exterior, sus elementos pueden tener diferentes conexiones y son estas las que determinan las

diferentes topologías de la red.

- Capas de salida: Recibe la notificación de la capa oculta y transmite la respuesta al medio externo.

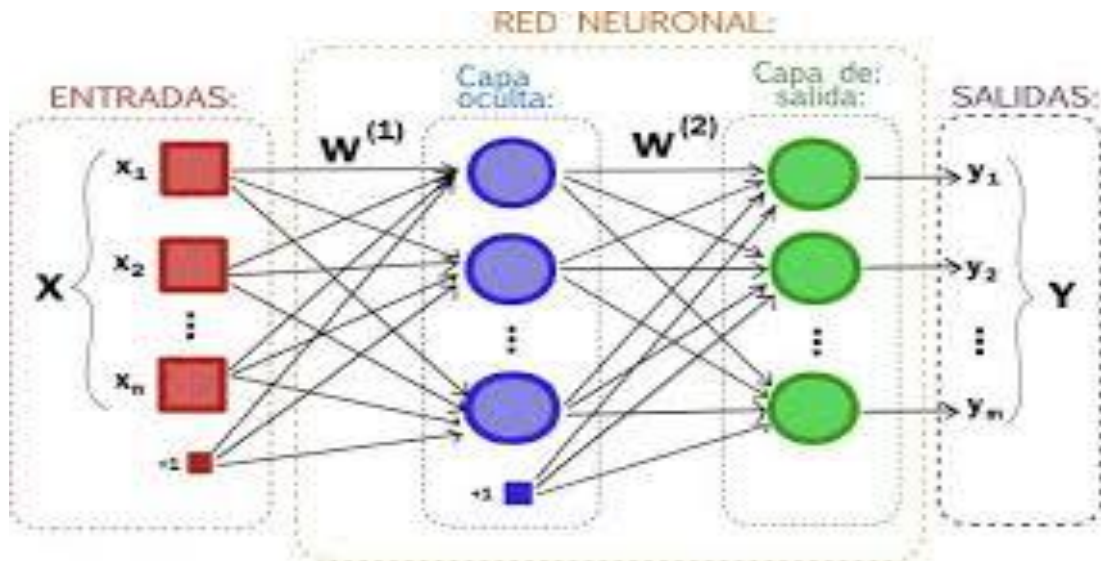


Figura 2.5. Capas en una red neuronal artificial.

Fuente: Maurnu, 2018.

Una vez definida el tipo de neurona que se utilizará en un modelo de redes neuronales artificiales es necesario definir la topología de la misma. La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas. (Aguilar, 1999).

Las neuronas suelen agruparse en unidades funcionales denominadas capas. Una capa de entrada es aquella que está compuesta por neuronas de entrada y por lo tanto recibe información procedente desde el exterior. Análogamente, se denomina capa oculta y capa de salida a aquellas capas que están compuestas por neuronas ocultas y de salida respectivamente. (Aguilar, 1999).

Una red neuronal artificial está compuesta por una o más capas, las cuales se encuentran interconectadas entre sí. Entre un par de neuronas de la red neuronal artificial pueden existir

conexiones. Estas conexiones son las sinapsis, tienen asociadas un peso sináptico, y son direccionales. Cuando la conexión se establece entre dos neuronas de una misma capa hablamos de conexiones laterales o conexiones intra-capa. Por el contrario, si la conexión se establece entre neuronas de distintas capas se les denomina conexión inter-capa. Si la conexión se produce en el sentido inverso al de entrada-salida la conexión se llama recurrente o realimentada. (Aguilar, 1999).

Una red puede estar formada por una única capa de neuronas. En este caso hablamos de redes monocapa, y las neuronas que conforman dicha capa cumplen la función de neuronas de entrada y salida simultáneamente. Cuando la red está compuesta por dos o más capas hablamos de redes multicapa. A su vez hablamos de redes neuronales con conexión hacia adelante (redes feedforward) cuando las conexiones entre las distintas neuronas de la red siguen un único sentido, desde la entrada de la red hacia la salida de la misma. Cuando las conexiones pueden ser tanto hacia adelante como hacia atrás hablamos de redes recurrentes (redes feedback). (Aguilar, 1999).

2.3.5.1. FASE DE APRENDIZAJE

Durante la operatoria de una red neuronal podemos distinguir dos fases o modos de operación: la fase de aprendizaje o entrenamiento, y fase de operación o ejecución. En la fase de aprendizaje, la red es entrenada para realizar un determinado tipo de procesamiento, alcanzando un nivel de entrenamiento adecuado, pasando a la fase de operación, donde la red es utilizada para llevar a cabo la tarea para la cual fue entrenada. (Aguilar, 1999).

El aprendizaje es un proceso de adaptación de los parámetros libres de la red a través de la estimulación continua por parte del entorno. El tipo de aprendizaje viene determinado por el modo de adaptar los parámetros o pesos en la red.

$$W_{kj} (n + 1) = W_{kj} (n) + \Delta W_{kj} (n)$$

Donde:

$W_{kj} (n + 1)$: Vector del peso del neurón k al neurón j después del ajuste.

$W_{kj} (n)$: Vector del peso del neurón k al neurón j antes del ajuste.

El conjunto de reglas preestablecidas para la solución de un poco problema de aprendizaje se denomina algoritmo de aprendizaje. La regla de aprendizaje consistirá en el procedimiento de modificación del patrón de conexión en función de la experiencia, conduciendo, en última instancia al establecimiento de nuevos modelos de respuesta del sistema a los estímulos externos. (Aguilar, 1999).

2.3.5.2. FASE DE ENTRENAMIENTO

Una vez seleccionada el tipo de neurona artificial que se utilizara en una red neuronal y determinada su topología es necesario entrenarla para que la red pueda ser utilizada. Partiendo de un conjunto de pesos sinápticos aleatorio, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red desarrollar correctamente una determinada tarea. Durante el proceso de aprendizaje se va refinando iterativamente la solución hasta alcanzar un nivel de operación suficientemente bueno. (Aguilar, 1999).

Según Viñuela (2004) el proceso de aprendizaje se puede dividir en tres grandes grupos de acuerdo a sus características.

- Aprendizaje supervisado. Se presenta a la red un conjunto de patrones de entrada junto con la salida esperada. Los pesos se van modificando de manera proporcional al error que se produce entre la salida esperada.
- Aprendizaje no supervisado. Se presenta a la red un conjunto de patrones de entrada. No hay información disponible sobre la salida esperada. El proceso de entrenamiento en este caso deberá ajustar sus pesos en base a la correlación existente entre los datos de entrada.
- Aprendizaje por refuerzo. Este tipo de aprendizaje se ubica al medio de los dos anteriores. Se presenta a la red un conjunto de patrones de entrada y se indica a la red si la salida obtenida es o no correcta. Sin embargo, no se proporciona el valor de la salida esperada. Este tipo de aprendizaje es muy útil en aquellos casos en que se desconoce cuál es la salida exacta que debe proporcionar la red.

a) APRENDIZAJE SUPERVISADO

En este tipo de aprendizaje supervisado se debe proporcionar a la red pares de patrones entrada-salida y la red neuronal aprende a asociarlos. Estadísticamente equivale a los modelos en los que hay vectores de variables independientes y dependientes. (Viñuela, 2004). Según Viñuela (2004) los algoritmos de aprendizaje supervisado tienen como procedimiento el minimizar una función objetivo o error. Esta función es monótona creciente de la diferencia entre la señal deseada, señal que debería dar la red y la salida proporcionada por la red, entonces el problema es de optimización, para esto los procedimientos genéricos son:

- Métodos de búsqueda global. Buscan el mínimo global de la función objetivo. Proporcionan los pesos que dan el valor más pequeño de dicha función sobre todo en su dominio.
- Métodos de búsqueda local. Buscan el mínimo más cercano de la función objetivo en relación al estado inicial de los pesos al comenzar dichos algoritmos; lógicamente son más rápidos en la obtención de la solución. Entre estos métodos se encuentran los basados en descenso por gradiente, gradiente conjugado entre otros.

b) APRENDIZAJE NO SUPERVISADO

Si el entrenamiento es no supervisado, únicamente se debe suministrar a la red los datos de entrada, para que extraiga los rasgos características esenciales. En terminología estadística equivale a los modelos en los que sólo hay vectores de variables independientes y buscan el agrupamiento de los patrones de entrada. (Viñuela, 2004).

El entrenamiento no supervisado se caracteriza porque no requiere de un vector de salida. Los pesos de la RNA se van modificando de manera que produzcan vectores de salida que sean consistentes, por consiguiente, se espera que entradas similares produzcan las mismas salidas. (Viñuela, 2004).

2.3.5.3. FASE DE OPERACIÓN

Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos. De esta manera, una red neuronal puede tratar con información que no le fue

presentada durante la fase de entrenamiento. (Viñuela, 2004).

2.4. DEEP LEARNING

El Deep Learning es un subconjunto del aprendizaje automático donde las redes neuronales artificiales, algoritmos inspirados en el cerebro humano, aprenden de grandes cantidades de datos. Nuestro encéfalo está compuesto por distintos núcleos de redes neuronales que tienen características de percepción y respuesta muy específicas, es decir, cada red se especializa en tareas especiales y diferentes a las otras redes. Por ejemplo, la percepción de un olor y la respuesta de agrado o desagrado a dicho olor; la percepción de un sonido y el paso de ese sonido a sensaciones. (Zuniga, 2019).

Si esto lo traducimos en Inteligencia Artificial veremos que los modelos computacionales de Deep Learning son una imitación de esas características arquitecturales de nuestro sistema nervioso, lo que permite que cada red neuronal construida de forma artificial esté entrenada para tareas especiales. El campo de la inteligencia artificial es, esencialmente, cuando las máquinas pueden realizar tareas que generalmente requieren de inteligencia humana. Abarca el aprendizaje automático, donde las máquinas pueden aprender por experiencia y adquirir habilidades sin la participación de los humanos. La forma en la que funciona en el cerebro humano esta arquitectura de redes neuronales es que la información recibida del exterior pasa a través de un número grande de capas antes de convertirse en una respuesta. (Zuniga, 2019).

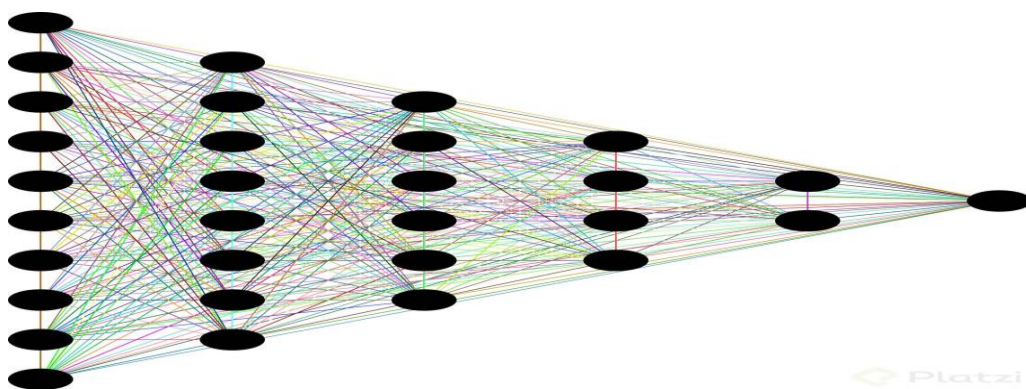


Figura 2.6. Deep Learning.

Fuente: Zuniga, 2019.

Deep Learning es un conjunto de algoritmos no lineales que pueden aplicarse para la modelización de datos y el reconocimiento de patrones. Cuando nos referimos a una forma no lineal estamos hablando de que las capas de las redes neuronales artificiales se apilan en una jerarquía que va desde las características de nivel de abstracción bajo hasta las de nivel de abstracción más compleja. (Zuniga, 2019).

2.4.1. FUNCIONAMIENTO DEL DEEP LEARNING

La forma en la que trabaja el Deep Learning es usando cascadas de capas con unidades de procesamiento que permiten la extracción y transformación de variables. Cada red, dentro de su jerarquía, aplica una transformación en su capa de entrada y utiliza esa información de aprendizaje para crear un modelo estadístico de salida que itera las veces necesarias hasta lograr un nivel de precisión en el aprendizaje y respuesta aceptable. (Zuniga, 2019).

El Deep Learning es una rama que ha mostrado resultados increíbles en tareas muy difíciles como reconocer objetos de una imagen, comprender el habla y los idiomas y, por supuesto, jugar juegos de mesa. Un grupo de personas muy inteligentes han estado trabajando en ello durante décadas, y es absolutamente de última generación. (Zuniga, 2019).

2.5. REDES NEURONALES RECURRENTE

Una red neuronal puede ser caracterizada por el modelo de la neurona, el esquema de conexión que presentan sus neuronas, o sea su topología, y el algoritmo de aprendizaje empleado para adaptar su función de cómputo a las necesidades del problema particular. Existe una amplia variedad de modelos de neuronas, cada uno se corresponde con un tipo determinado de función de activación y de salida de la neurona. (Bonet, 2007).

La topología es la forma específica de conexión (arquitectura) y la cantidad de neuronas conectadas (el número de parámetros libres) que describen una red. En los últimos años se ha producido una amplia variedad de topologías de redes neuronales, sin embargo, la mayoría de ellas se encuentran ubicadas en dos grandes grupos: las redes multicapa de alimentación hacia adelante (feed-forward) y las redes recurrentes (RNR). Las redes feed-forward no tienen ciclos, las neuronas están organizadas en capas que se conectan de manera unidireccional. Generalmente estas redes son denominadas estáticas, pues producen una

única salida para un conjunto de entrada, o sea, el estado de una red es independiente del estado anterior. Por otro lado, las redes recurrentes son sistemas dinámicos. El cálculo de una entrada, en un paso, depende del paso anterior y en algunos casos del paso futuro. (Bonet, 2007).

Según Bonet (2007) las RNR son capaces de realizar una amplia variedad de tareas computacionales incluyendo el tratamiento de secuencias, la continuación de una trayectoria, la predicción no lineal y la modelación de sistemas dinámicos. Estas redes también se conocen como redes espacio-temporales o dinámicas, son un intento de establecer una correspondencia entre secuencias de entrada y de salida que no son más que patrones temporales. Existen tres tipos de tareas esenciales que se pueden realizar con este tipo de redes:

- Reconocimiento de secuencias: Se produce un patrón de salida particular cuando se especifica una secuencia de entrada.
- Reproducción de secuencias: La red debe ser capaz de generar el resto de una secuencia cuando ve parte de ella.
- Asociación temporal: En este caso una secuencia de salida particular se debe producir en respuesta a una secuencia de entrada específica.

Una RNN se puede clasificar en parcial y/o totalmente recurrente. Las totalmente recurrentes son aquellas que cada neurona puede estar conectada a cualquier otra y sus conexiones recurrentes son variables. Las redes parcialmente recurrentes son aquellas que sus conexiones recurrentes son fijas. Estas últimas son la forma usual para reconocer o reproducir secuencias. Generalmente tienen la mayoría de las conexiones hacia delante, pero incluyen un conjunto de conexiones retroalimentadas. (Bonet, 2007).

Una red neuronal recurrente no tiene una estructura de capas definida, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria.

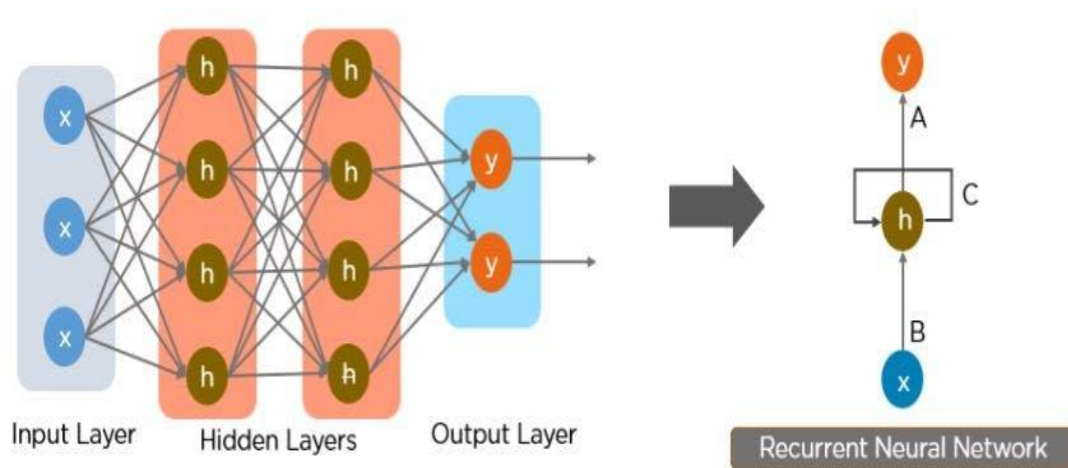


Figura 2.7. Estructura de una red neuronal recurrente.

Fuente: Borjas, 2021.

En el lado izquierdo de la Figura 2.7., veremos lo siguiente:

- X – vendría a ser la capa de entrada.
- H – es la capa oculta que tiene la información de salida anterior y la retroalimenta en sí misma.
- Y – es la capa de salida.
- En el caso de A, B, C estos son los parámetros para mejorar la salida del modelo.

Las redes neuronales recurrentes son muy potentes para todo lo que tiene que ver con el análisis de secuencias, como puede ser el análisis de textos, sonido o video. Existe multitud de tipos de redes neuronales dependiendo del número de capas ocultas y la forma de realizar la retro propagación, entre las cuales están las redes LSTM.

2.5.1. REDES LSTM

Las redes LSTM (Long Short Term Memory) están compuestas por unidades LSTM y son un tipo especial de red neuronal recurrente descritas en 1997 por Hochreiter & Schmidhuber. Las redes neuronales recurrentes convencionales presentan problemas en su entrenamiento

debido a que los gradientes retro propagados tienden a crecer enormemente o a desvanecerse con el tiempo debido a que el gradiente depende no solo del error presente sino también los errores pasados. La acumulación de errores provoca dificultades para memorizar dependencias a largo plazo. Estos problemas son solventados por las redes LSTM, para ello incorporan una serie de pasos para decidir qué información va a ser almacenada y cual borrada. (Calvo, 2018).

La unidad de memoria LSTM contiene tres puertas que controlan el modo en que la información fluye dentro o fuera de la unidad.

- Puerta de **entrada** controla cuando la información nueva puede entrar en la memoria.
- Puerta del **olvido** controla cuando se olvida una parte de la información, lo que permite a la celda discriminar entre datos importantes y superfluos, dejando así sitio para nuevos datos.
- Puerta de **salida** controla cuando se utiliza en el resultado de los recuerdos almacenados en la celda.

La celda dispone de un mecanismo de optimización de las ponderaciones basado en el error de salida de la red resultante, que controla cada puerta. Este mecanismo se suele implementar con el algoritmo de capacitación, BPTT. Las redes LSTM se utilizan para compresión de textos de lenguajes naturales, reconocimiento de escritura a mano, reconocimiento de voz, reconocimiento de gestos, captura de imágenes. (Calvo, 2018).

2.6. COEFICIENTES CEPSTRALES DE FRECUENCIA DE MEL

Los coeficientes cepstrales de frecuencia de mel (MFCC) se usaron originalmente en varias técnicas de procesamiento del habla; sin embargo, a medida que el campo de recuperación de información musical (MIR) comenzó a desarrollarse como complemento del aprendizaje automático, se descubrió que los MFCC podían representar el timbre bastante bien. (Yumpu, 2015).

Un método más eficiente para extraer características y que es el más utilizado actualmente en reconocedores comerciales son los MFCC, este método es robusto, además hace uso de la

Transformada de Fourier para obtener las frecuencias de la señal. El objetivo es desarrollar un conjunto de características basadas en criterios perceptuales, diversos experimentos muestran que la percepción de los tonos en los humanos no está dada en una escala lineal, esto hace que se trate de aproximar el comportamiento del sistema auditivo. (Yumpu, 2015).

Según Programmerclick (2020) Davis y Mermelstein en 1980 demostraron que los MFCC son beneficiosos para el reconocimiento automático del habla. Los MFCC son una representación definida como el cepstrum de una señal ventaneada en el tiempo que ha sido derivada de la aplicación de una Transformada rápida de Fourier, pero en una escala de frecuencias no lineal, las cuales se aproximan al comportamiento del sistema auditivo humano. MFCC es un conjunto de algoritmos desarrollados para completar el reconocimiento de voz, basado en cómo las personas reconocen las voces.

Primero, se aclara cuatro puntos:

- La mayor parte de la información de la señal de voz está contenida en el componente de baja frecuencia.
- La mayor parte de la información de la señal de voz está contenida en la parte de baja amplitud.
- El nivel del sonido que escucha el oído humano no está relacionado linealmente con la frecuencia del sonido.
- Las personas no pueden distinguir todos los componentes de frecuencia. Solo dos componentes de frecuencia difieren en un cierto ancho de banda, cuando el tiempo (por debajo de 1000 hz, el ancho de banda es constante de 100 hz; por encima de 1000 hz, el ancho de banda y la frecuencia central tienen una relación exponencial), los humanos pueden distinguir, de lo contrario las personas escucharán dos tonos en uno, esto se llama efecto de blindaje, El ancho de banda se llama ancho de banda crítico.

2.6.1. PROCESO DE EXTRACCIÓN DE MFCC

El diagrama de flujo básico es el siguiente:

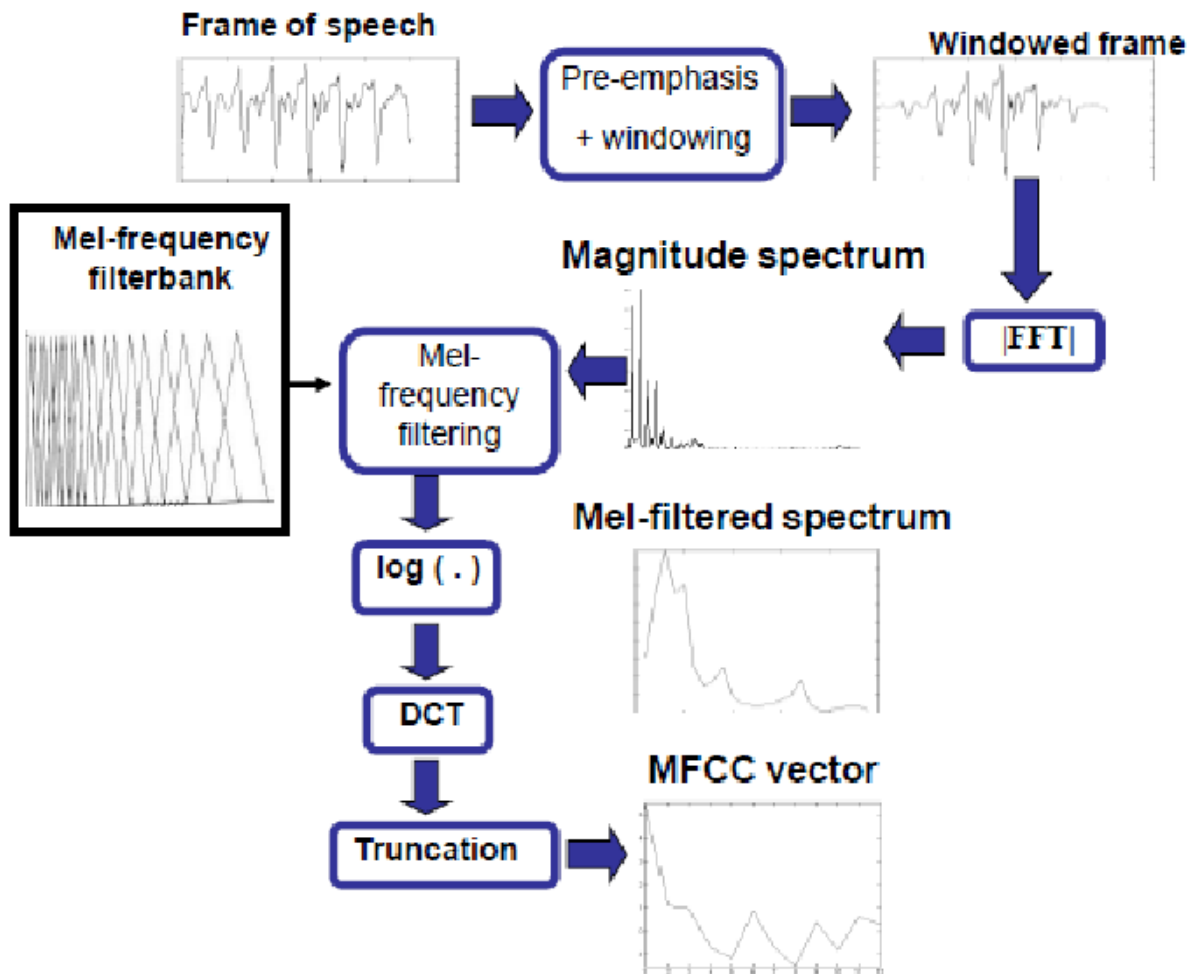


Figura 2.8. Proceso de extracción MFCC.

Fuente: Mathur, 2015.

Para empezar a extraer los parámetros MFCC de un archivo de audio, el audio se inventana. En lugar de extraer los parámetros MFCC de una pieza de 1 segundo (que podría ser aproximadamente la duración de una palabra), se extraen los MFCC de segmentos de 25 ms. La ventana es una ventana de tipo Hanning o Hamming para evitar los armónicos artificiales que aparecen al utilizar ventanas cuadradas. Cada ventana se desplaza 10 ms en tiempo, por lo que existe una superposición entre una ventana y otra. (Sánchez, 2016).

Las ventanas son funciones matemáticas usadas con frecuencia en el análisis y el procesamiento de señales para evitar las discontinuidades al principio y al final de los bloques

analizados. Cuando una secuencia de muestras a procesar digitalmente, es demasiado grande, tal procesamiento puede sobrepasar las capacidades del equipo de cómputo disponible. Así entonces se toman pequeños espacios muestrales y a cada uno se le aplica el mismo procesamiento. Las ventanas se usan entonces para evitar las discontinuidades al principio y al final de los bloques analizados. Sin el uso de las ventanas, el resultado de unir las secuencias procesadas implica nuevas componentes espectrales: artefactos en imágenes y ruido de alta frecuencia o gis en audio. (Gonzales, 2017).

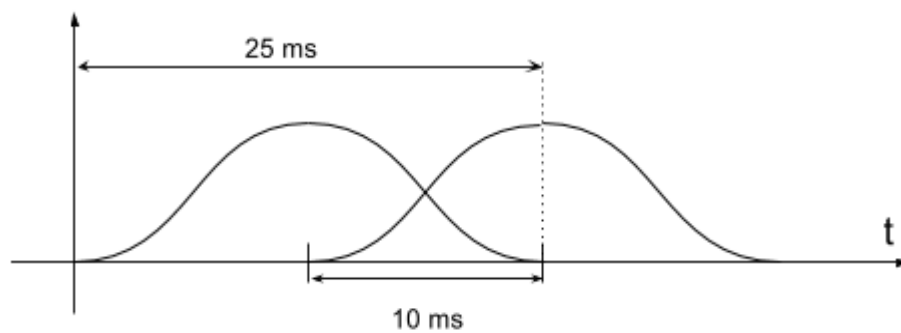


Figura 2.9. Ventana Hamming.

Fuente: Sánchez, 2016.

- A cada una de estas ventanas se aplica la FFT.
- A la FFT de cada ventana se le aplica el banco de filtros MEL.
- Se calcula el logaritmo
- Se hace la transformada discreta del coseno (DCT)
- Al final se queda con los N primeros coeficientes (baja frecuencia). Normalmente 12 coeficientes. O también 12 para bajas frecuencias y 1 para la energía media de la ventana.

2.6.2. TRANSFORMADA DE FOURIER

La FFT es un importante método de medición en la tecnología de medición de **audio** y

acústica. Descompone una señal en sus componentes espectrales individuales y así proporciona información sobre su composición. Dada una Transformada de Fourier de una señal de entrada:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi nk}{N}} \quad K = 0, 1, \dots, N - 1$$

2.6.3. BANCO DE FILTROS EN LA ESCALA DE MEL

Estos filtros calculan el promedio del espectro alrededor de cada frecuencia central.

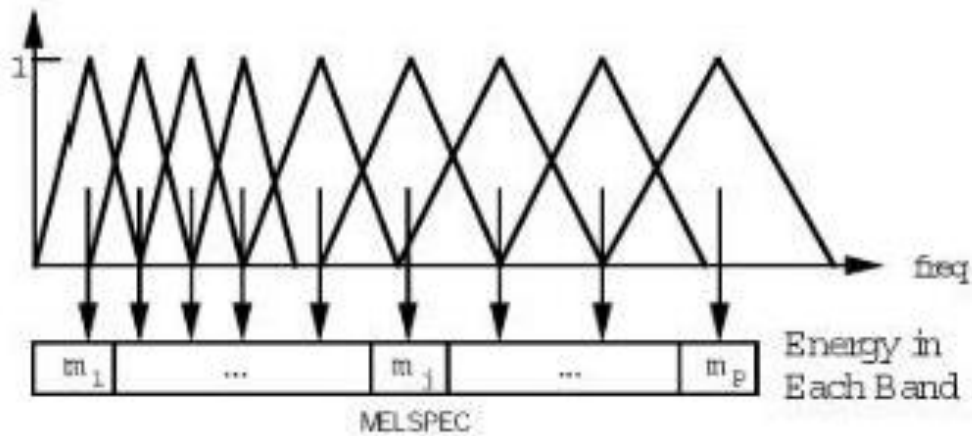


Figura 2.10. Banco de filtros de escala de Mel.

Fuente: Yumpu, 2015.

Se puede definir a f_l como la frecuencia más alta y a f_h como la frecuencia más baja del banco de filtros en Hz, F_s es la frecuencia de muestreo en Hz, M el número de filtros y N el tamaño de la Transformada Rápida de Fourier. Los puntos límite $f(m)$ se encuentran uniformemente espaciados en la escala de Mel. (Yumpu, 2015).

$$f(m) = \frac{N}{F_s} \beta^{-1} + \left(\beta(f_l) + m \frac{\beta(f_h) - \beta(f_l)}{M + 1} \right)$$

Donde la escala de Mel está dada por:

$$\beta(f) = 1125 \ln \left(1 + \frac{f}{700} \right)$$

La grafica de dicha expresión se muestra a continuación:

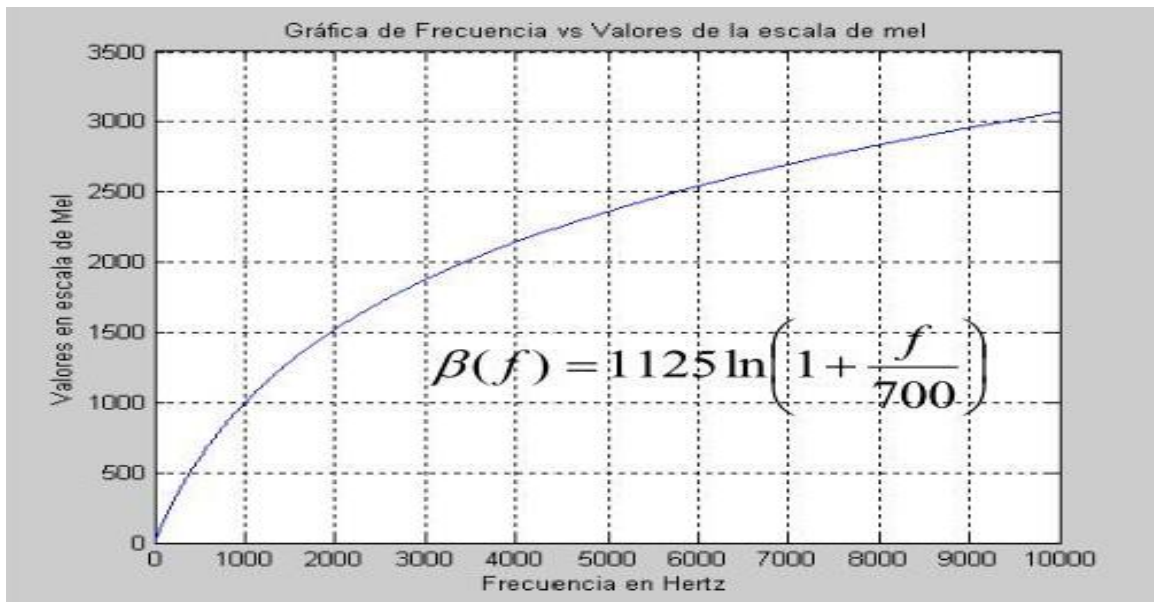


Figura 2.11. Escala de Mel.

Fuente: Yumpu, 2015.

2.6.4. LOGARITMO

Después de obtener el banco de filtros de Mel se calcula el logaritmo de la energía de cada filtro con la siguiente fórmula.

$$S(m) = \ln \left(\sum_{k=0}^{N-1} |X_a(k)|^2 H_m(K) \right) \quad 0 < m < M$$

2.6.5. TRANSFORMADA DISCRETA DEL COSENO

Según Yumpu (2015) El cepstrum en frecuencia de Mel es la Transformada Discreta del Coseno (DCT) de las salidas de los M filtros. Donde M varía para diferentes implementaciones de 24 a 40, para el Reconocimiento Automático del Habla (RAH) generalmente son usados los primeros 13 coeficientes. La DCT tiene una buena capacidad de compactación de la energía al dominio transformado, es decir, que la transformada de coseno

discreta consigue concentrar la mayor parte de la información en pocos coeficientes transformados. Este algoritmo es ampliamente usado para obtener el vector de características en sistemas de RAH. Después de aplicar DCT se obtienen los MFCC y está dado por la siguiente fórmula:

$$c(m) = \sum_{m=0}^{M-1} S(m) \cos \left(\pi n \left(\frac{m - \frac{1}{2}}{M} \right) \right)$$

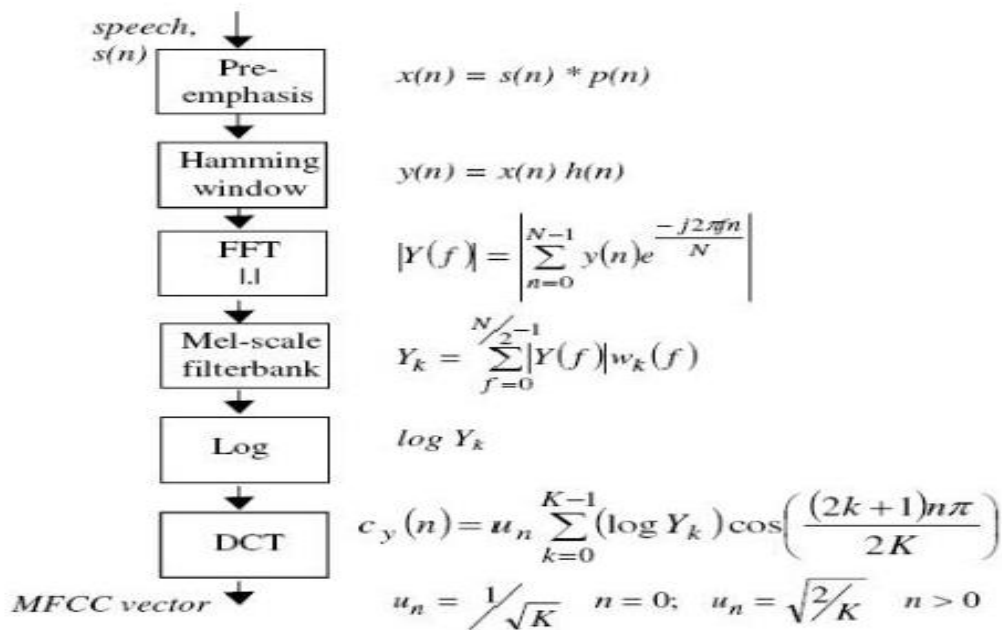


Figura 2.12. Proceso y fórmulas para obtener MFCC.

Fuente: Yumpu, 2015.

2.7. METODOLOGÍA FUNDAMENTAL PARA LA CIENCIA DE DATOS

2.7.1. INTRODUCCIÓN

En el dominio de la ciencia de datos, resolver problemas y responder preguntas a través del análisis de datos es una práctica estándar. A menudo, los científicos de datos construyen modelos para predecir resultados o para descubrir patrones subyacentes, con la meta de

obtener insights. Después, las organizaciones pueden usar estos insights para tomar medidas que mejoren los siguientes resultados. (Rollins, 2015).

Para analizar los datos y construir modelos existen numerosas tecnologías que evolucionan rápidamente. En un tiempo extraordinariamente corto, han pasado de utilizar escritorios a almacenes que están masivamente en paralelo con enormes volúmenes de datos y funcionalidad analítica en las bases de datos relacionales y de Apache Hadoop. La analítica de texto en datos no estructurados o semiestructurados se está volviendo cada vez más importante como forma de incorporar a modelos predictivos la percepción y otra información útil de los textos, lo que a menudo conlleva mejoras significativas en la calidad y precisión del modelo. (Rollins, 2015).

Los enfoques analíticos emergentes buscan automatizar muchos de los pasos de la creación y aplicación de modelos, lo que hace que la tecnología de aprendizaje automático sea más accesible para quienes carecen de profundas habilidades cuantitativas. Además, en contraposición al enfoque de arriba abajo por el que primero se define el problema empresarial y luego se analizan los datos para obtener una solución, algunos científicos de datos pueden usar un enfoque de abajo a arriba. Con este último enfoque, el científico de datos analiza grandes volúmenes de datos para saber cuál es el objetivo empresarial que pueden sugerir los datos y, luego, aborda ese problema. Dado que la mayoría de los problemas se abordan de manera descendente, la metodología de este documento refleja esa visión. (Rollins, 2015).

2.7.2. METODOLOGÍA Y ETAPAS

A medida que las capacidades de analítica de datos se vuelven más accesibles y prevalentes, los científicos de datos necesitan una metodología fundamental capaz de proporcionar una estrategia de orientación, que sea independiente de las tecnologías, los volúmenes de datos o los enfoques involucrados. Esta metodología tiene algunas similitudes con las metodologías reconocidas 1-5 para la minería de datos, pero pone el énfasis en varias de las nuevas prácticas en la ciencia de datos, como el uso de grandes volúmenes de datos, la incorporación de la analítica de texto en el modelado predictivo y la automatización de algunos procesos.

La metodología consta de 10 etapas que forman un proceso iterativo para el uso de datos para descubrir insights. Cada etapa juega un papel vital en el contexto de la metodología general. (Rollins, 2015).

Podemos visualizar la metodología y sus etapas en la Figura 2.11.

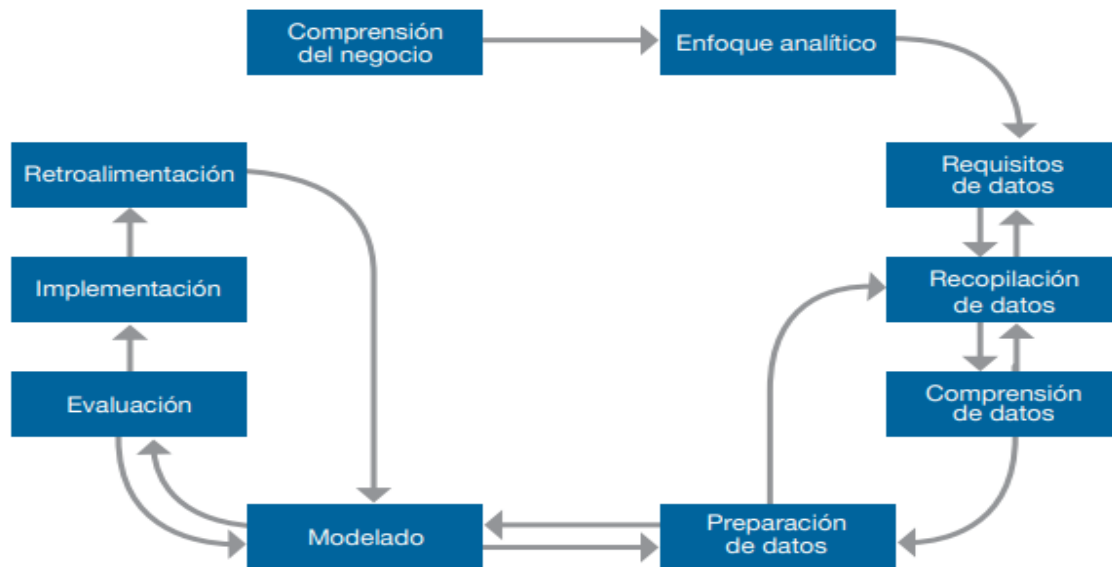


Figura 2.13. Metodología fundamental para la ciencia de datos.

Fuente: Rollins, 2015.

2.7.2.1. ETAPA 1: COMPRENSIÓN DEL NEGOCIO

Todos los proyectos comienzan con la comprensión del negocio. Los promotores de negocios que necesitan la solución analítica desempeñan el papel más importante en esta etapa, al definir el problema, los objetivos del proyecto y los requisitos de la solución desde una perspectiva empresarial. Esta primera etapa sienta las bases para que el problema empresarial sea resuelto con éxito. Para ayudar a garantizar el éxito del proyecto, los promotores deben participar mientras dure el proyecto para proporcionar experiencia en el dominio, revisar los hallazgos intermedios y garantizar que el trabajo siga su curso para generar la solución deseada. (Rollins, 2015).

2.7.2.2. ETAPA 2: ENFOQUE ANALÍTICO

Cuando el problema empresarial se haya establecido claramente, el científico de datos podrá

definir el enfoque analítico para resolver el problema. Esta etapa implica expresar el problema bajo el contexto de las técnicas estadísticas y de aprendizaje automático, para que la organización pueda identificar las más adecuadas para el resultado deseado. Por ejemplo, si el objetivo es predecir una respuesta como sí o no, el enfoque analítico podría definirse como la construcción, las pruebas y la implementación de un modelo de clasificación. (Rollins, 2015).

2.7.2.3. ETAPA 3: REQUISITOS DE DATOS

El enfoque analítico elegido determina los requisitos de datos. Más concretamente, los métodos analíticos a utilizar requieren de determinados contenidos de datos, formatos y representaciones, orientados por el conocimiento en el dominio. (Rollins, 2015).

2.7.2.4. ETAPA 4: RECOPIACIÓN DE DATOS

En la etapa inicial de recopilación de datos, los científicos de datos identifican y reúnen los recursos de datos disponibles (estructurados, no estructurados y semiestructurados) y relevantes para el dominio del problema. Por lo general, deben elegir si realizan inversiones adicionales para obtener elementos informativos menos accesibles. Lo mejor puede ser aplazar la decisión de inversión hasta que se sepa más sobre los datos y el modelo. Si hay algunas lagunas en la recopilación de datos, es posible que el científico tenga que revisar los requisitos de datos y recopilar más datos o nuevos datos. (Rollins, 2015).

Aunque el muestreo y la subdivisión de datos siguen siendo importantes, las plataformas actuales de alto rendimiento y la funcionalidad analítica en la base de datos permiten que los científicos de datos utilicen conjuntos de datos mucho más grandes que contienen gran parte de los datos disponibles, o incluso todos. Al incorporar más datos, los modelos predictivos pueden representar mejor los eventos raros, como la incidencia de una enfermedad o un fallo del sistema. (Rollins, 2015).

2.7.2.5. ETAPA 5: COMPRESIÓN DE DATOS

Después de la recopilación de datos inicial, los científicos de datos suelen utilizar estadísticas descriptivas y técnicas de visualización para comprender el contenido de los datos, evaluar su calidad y descubrir insights iniciales sobre ellos. Para llenar los huecos es posible que sea

necesario volver a recopilar datos. (Rollins, 2015).

2.7.2.6. ETAPA 6: PREPARACIÓN DE DATOS

Esta etapa abarca todas las actividades para construir el conjunto de datos que se utilizará en la subsiguiente etapa de modelado. Entre las actividades de preparación de datos están la limpieza de datos (tratar con valores no válidos o que faltan, eliminar duplicados y dar un formato adecuado), combinar datos de múltiples fuentes (archivos, tablas y plataformas) y transformar los datos en variables más útiles. (Rollins, 2015).

Los científicos de datos utilizan un proceso llamado ingeniería de características para crear variables explicativas adicionales, también conocidas como indicadores o características, a través de una combinación de conocimiento en el dominio y de variables estructuradas existentes. Cuando hay disponibles datos en texto, como los registros del centro de atención al cliente o las observaciones de los médicos en forma no estructurada o semiestructurada, la analítica de texto se puede utilizar para derivar nuevas variables estructuradas y, así, enriquecer el conjunto de indicadores y mejorar la precisión del modelo. (Rollins, 2015).

La preparación de datos suele ser el paso más largo de los proyectos de ciencia de datos. En muchos dominios, algunos pasos de la preparación de datos son comunes para problemas diferentes. La automatización anticipada de determinados pasos de la preparación de datos puede acelerar el proceso al minimizar el tiempo de preparación a medida. Gracias al alto rendimiento, los sistemas masivamente paralelos y la funcionalidad analítica que reside donde se almacenan los datos de hoy en día, los científicos de datos pueden preparar los datos de forma más fácil y rápida utilizando conjuntos de datos muy grandes. (Rollins, 2015).

2.7.2.7. ETAPA 7: MODELADO

La etapa de modelado utiliza la primera versión del conjunto de datos preparado y se enfoca en desarrollar modelos predictivos o descriptivos según el enfoque analítico previamente definido. En los modelos predictivos, los científicos de datos utilizan un conjunto de capacitación (datos históricos en los que se conoce el resultado de interés) para construir el modelo. El proceso de modelado normalmente es muy iterativo, ya que las organizaciones están adquiriendo insights intermedios, lo que deriva en ajustes en la preparación de datos y

en la especificación del modelo. Para una técnica determinada, los científicos de datos pueden probar múltiples algoritmos con sus respectivos parámetros para encontrar el mejor modelo para las variables disponibles. (Rollins, 2015).

2.7.2.8. ETAPA 8: EVALUACIÓN

Durante el desarrollo del modelo y antes de su implementación, el científico de datos evalúa el modelo para comprender su calidad y garantizar que aborda el problema empresarial de manera adecuada y completa. La evaluación del modelo implica el cálculo de varias medidas de diagnóstico y de otros resultados, como tablas y gráficos, lo que permite al científico de datos interpretar la calidad y la eficacia del modelo en la resolución del problema. Para los modelos predictivos, los científicos de datos usan un conjunto de pruebas, que es independiente del conjunto de capacitación, pero sigue la misma distribución de probabilidad y tiene un resultado conocido. El conjunto de pruebas se utiliza para evaluar el modelo para ajustarlo según las necesidades. A veces, el modelo final también se aplica a un conjunto de validación para realizar una evaluación final. (Rollins, 2015).

Además, los científicos de datos pueden asignar al modelo pruebas de significancia estadística como prueba adicional de su calidad. Esta prueba adicional puede ser fundamental para justificar la implementación del modelo o para tomar medidas cuando hay mucho en juego, como un costoso protocolo médico suplementario o un sistema crítico para vuelos en avión. (Rollins, 2015).

2.7.2.9. ETAPA 9: IMPLEMENTACIÓN

Cuando el modelo satisfactorio ha sido desarrollado y aprobado por los promotores del negocio, se implementa en el entorno de producción o en un entorno de pruebas comparable. Por lo general, se implementa de forma limitada hasta que su rendimiento se haya evaluado completamente. Su implementación puede ser tan fácil como generar un informe con recomendaciones, o tan enrevesado como incrustar el modelo en un complejo proceso de puntuación y de flujo de trabajo administrado por una aplicación personalizada. La implementación de un modelo en un proceso operativo empresarial generalmente involucra a grupos, habilidades y tecnologías adicionales dentro de la empresa. Por ejemplo, un grupo

de ventas puede implementar un modelo de propensión a la respuesta a través de un proceso de administración de campañas creado por un equipo de desarrollo y administrado por un grupo de marketing. (Rollins, 2015).

2.7.2.10. ETAPA 10: RETROALIMENTACIÓN

Al recopilar los resultados del modelo implementado, la organización obtiene retroalimentación sobre el rendimiento del modelo y su impacto en el entorno en el que se implementó. Por ejemplo, la retroalimentación puede ser en forma de porcentajes de respuesta a una campaña promocional dirigida a un grupo de clientes que ha sido identificado por el modelo como respondedores de alto potencial. Los científicos de datos pueden analizar esta retroalimentación para ajustar el modelo para mejorar su precisión y utilidad. Pueden automatizar algunos o todos los pasos de la evaluación del modelo y de la recopilación de retroalimentación, el ajuste y la reimplementación del modelo para acelerar el proceso de actualización del modelo para obtener mejores resultados. (Rollins, 2015).

2.8. LA MENTIRA Y EL ENGAÑO

La mentira es la expresión manifiesta contraria a lo que se sabe, se cree o se piensa. Engañar es dar a la mentira apariencia de verdad o inducir a alguien a tener por cierto lo que no es, valiéndose de palabras y de obras aparentes o fingidas. Mientras que la mentira se refiere así al contenido de un mensaje que refleja adecuadamente o no la realidad, el engaño, o la acción de engañar, incluye el concepto clave de intencionalidad o voluntariedad. (Martínez, 2005). Se dice que una persona miente cuando deliberadamente aporta una información de la que sabe conscientemente que no se ajusta a la realidad de los hechos (Massip, 2008).

El engaño no es exclusivo de la especie humana, sino que es también una característica que está presente en los primates y en otros animales que viven en entornos sociales de gran complejidad; y en los reinos animal y vegetal son numerosos los seres vivos que han desarrollado, en el proceso evolutivo, capacidades de camuflaje y de adaptación muy elaboradas, que han prosperado gracias al efecto de confundir a sus competidores o a sus depredadores. (Lemos, 2005)

Como se mencionó anteriormente el fenómeno de mentir está proscrito en todas las culturas

y se han realizado grandes esfuerzos en intentar superarlo pues amenaza el bienestar la sociedad (Benavides, 2007).

Si una persona es creíble y fiable, los demás tenderán a creer lo que se dice y podrán ser convencidos. El mentiroso intentará reforzar a toda costa su credibilidad y utilizarla para que la crean (Martínez, 2005).

Frecuentemente, la credibilidad es una cuestión de percepción, de que alguien sea percibido como una persona fiable; el mentiroso intentará dar una imagen creíble positiva y honesta de sí mismo.

Manzanero (2008) afirma que el desarrollo de procedimientos para discriminar a las personas que mienten en sus declaraciones se remonta al inicio de los sistemas de administración de justicia. Así, podemos encontrar referencias a diferentes métodos para detectar la mentira en escritos Vedas en la India de hace 3000 años, donde se propone la observación de las respuestas no verbales de los sospechosos como técnica. Mientras en occidente, el médico griego Erasistratus (300-250 a. c.), al servicio de Alejandro El Grande, propuso medir algunas respuestas fisiológicas.

El engaño es “un intento deliberado exitoso o no, que se hace sin aviso, para crear en otro una creencia el cual el comunicador considera no ser verdadera” (Vrij, 2000). El engaño es un acto intencional. El engaño se define únicamente desde la perspectiva del que engaña y no si la declaración es factual, un tercer aspecto del engaño es que las personas solo mienten cuando ellos no informan a otros de sus intenciones de mentir.

2.9. PSICOFISIOLOGÍA DE LAS EMOCIONES

En primer término, es importante definir el concepto de emociones, entendidas como las reacciones complejas y estructuradas, de carácter rápido, difícil de controlar, con un fuerte contenido subjetivo y fisiológico, que alteran el comportamiento que se está realizando en ese momento (Martínez, 1995).

Para hablar de emociones es importante que se cumplan tres características básicas: el componente comportamental, el hecho de que es algo automático y difícil de controlar y por último el componente hormonal. Normalmente el factor más notable a simple vista es el

comportamental, pero lo que tiene que ocurrir para que se de algún comportamiento es que algún estímulo del exterior, active el componente hormonal el cual facilita la activación de las respuestas neurovegetativas, lo que lleva a una respuesta comportamental. (Martínez, 1995).

Para conocer las funciones vegetativas que propician cambios comportamentales se propone la siguiente lista: cambios en la electricidad de la piel, velocidad cardiaca, nivel de presión sanguínea, vasoconstricción y vasodilatación, cambios en la velocidad, amplitud y regularidad de la respiración, temperatura de la piel, sudoración, alteraciones en el diámetro de la pupila, cambios en el funcionamiento digestivo, contracción o relajación de los esfínteres, variaciones en la electricidad del cerebro, cambios en la química de la sangre, de la orina y de la saliva, y alteraciones del metabolismo basal. Es importante tener en cuenta que todas estas funciones no se tienen que presentar al mismo tiempo ni en todas las emociones, si no que esto varía según la persona y la emoción experimentada al momento. (Godoy, 2010).

El estrés y la emoción intensos se manifiestan a través del aumento en la actividad fisiológica general, que es más acusada en el sistema nervioso vegetativo y en las funciones corporales que dependen de él. Principalmente se estudian los cambios en la presión arterial, las frecuencias cardiaca y respiratoria y los cambios en la actividad eléctrica de la piel asociados a la sudoración. (Martínez, 2005).

2.10. POLÍGRAFO

A menudo el polígrafo se etiqueta como detector de mentiras, pero no detecta mentiras sino ciertos cambios periféricos que pueden traslucir ansiedad, miedo, culpa o una respuesta de orientación por parte del examinado. El que estos estados internos se asocien o no con el engaño es otra cuestión. Mentir puede o no dar lugar a ansiedad, miedo, etc., y estas emociones pueden deberse al acto de mentir o bien a otros factores. (Garrido, 2006).

El polígrafo es una técnica, basada en las respuestas psicofisiológicas del cuerpo relacionadas con las reacciones fisiológicas del mismo al dar una información inventada cuando se le da la instrucción contraria. Comúnmente el polígrafo se utiliza cuando en las investigaciones

policiales existen datos que son difíciles de clarificar, pues en ocasiones las declaraciones tanto de los testigos como de los victimarios, pueden omitir cierta información, pues creen que no son importantes (Iacono, 2006).

Es importante mencionar el uso del polígrafo como principal antecedente de los analizadores de estrés de voz y por lo tanto abarcar de manera general un poco de la historia de este. Los inicios del polígrafo fueron a principios del siglo XIX con el criminólogo italiano Lombroso, que utiliza el pletismógrafo para medir la velocidad del pulso y el volumen sanguíneo durante los interrogatorios criminales, ya que los cambios en estas áreas se relacionan con cierta deshonestidad en las personas. Después del trabajo de Lombroso el polígrafo tiene dos grandes protagonistas; principalmente en la Universidad de Harvard de los Estados Unidos de América donde tuvo su nacimiento a principios del siglo XX. En 1908 Hugo Munsterberg, profesor de esta universidad e innovador del campo de la psicofisiología y la criminología propuso teóricamente el uso de tres monitores de análisis en la detección del estrés emocional parecidos a los utilizados en los polígrafos actuales. Otro precursor fue William Moulton Marston, el desarrolló el primer detector de mentiras específicamente por utilizar el monitoreo de la presión sanguínea para evaluar la deshonestidad entre sospechosos de ilícitos. (Benavides, 2007).

Normalmente los registros poligráficos comprenden la respuesta psicogalvánica, la presión sanguínea relativa, el ritmo cardíaco y la respiración. Para ello es necesario colocar unos electrodos sobre los dedos del sujeto mediante los cuales se toman las medidas sanguíneas, y unos tubos flexibles alrededor del pecho y del abdomen para las medidas respiratorias. La cuestión fundamental en un examen poligráfico consiste en la comparación entre la reactividad fisiológica ante diversos tipos de preguntas. Existen varios procedimientos poligráficos, entre los cuales destacan la Prueba de Pregunta Control o CQT, reetiquetada como Prueba de la Pregunta de Comparación o CQT, y la Prueba del Conocimiento del Culpable o GKT. (Garrido, 2006).

2.11. ANÁLISIS DE ESTRÉS DE VOZ

El Análisis de Estrés de Voz (VSA por sus siglas en inglés) es una prueba que ha

evolucionado y mejorado durante años, hasta convertirse en un estándar en los exámenes de confiabilidad de empresas y gobiernos en todo el mundo. (Martínez, 2005).

En esta prueba se identifican las frecuencias en las modulaciones de la voz y se exhiben en una gráfica impresa. De esta forma es posible determinar los efectos de ciertos cambios fisiológicos en la voz. El análisis puede hacerse con una simple respuesta monosilábica o mediante una narración. Como el estudio se realiza con la voz de un individuo, este puede hacerse en entrevistas personales, telefónicas; incluso, ser grabadas para su posterior análisis. Está comprobado que el estrés y la emoción intensa se manifiesta a través del aumento en la actividad fisiológica general, que es más acusada en el sistema nervioso vegetativo y en las funciones corporales que dependen de él. Se presenta en las emociones como el enojo o miedo. (Martínez, 2005).

Martínez (2005) también señala que en estudios relacionados a la mentira se estudian los cambios en la presión arterial, las frecuencias cardíaca y respiratoria y los cambios en la actividad eléctrica de la piel asociados a la sudoración ya que se sabe que ciertas variables de personalidad podrían afectar las mediciones psicofisiológicas.

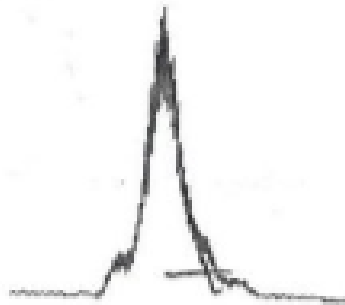


Figura 2.14. Gráfica sin estrés.

Fuente: Gonzales, 2021.

En la Figura 2.8. podemos observar que el pico no muestra una ampliación, a diferencia de la Figura 2.9. que muestra una ampliación lo que denota estrés.



Figura 2.15. Gráfica con estrés.

Fuente: Gonzales, 2021.

2.12. PRESENCIA DE ESTRÉS EN LAS EMOCIONES DE ENOJO Y MIEDO

El enojo se da de forma inmediata y anula la parte más lógica y congruente del cerebro para incrementar la actividad cardiovascular y respiratoria. Se conoce que en la segunda etapa del enojo se da un incremento de los niveles séricos de cortisol, una hormona relacionada con el estrés, y el cambio de glucosa a nivel plasmático. De hecho, el cortisol buscará adaptarnos bajo ciertas condiciones, en consecuencia, el enojo es una respuesta funcional del sistema nervioso central. (Nuñez, 2019).

El enojo es una reacción que tiene el sistema nervioso central para manifestarse, estar en contra, evadir algún tipo de molestia y obtener algún beneficio. De hecho, llevado a cabo adecuadamente nos hace competitivos, y por eso resulta benéfico. ¿Qué sustancias del cerebro se involucran en este proceso? De acuerdo con el especialista el enojo libera noradrenalina (hormona que aumenta la presión arterial y el ritmo cardíaco) y dopamina, al mismo tiempo que glutamato y se da una disminución de los niveles de serotonina y vasopresina. (Nuñez, 2019).

Los investigadores de la detección de la mentira han propuesto algunas teorías para dar cuenta de los supuestos indicadores conductuales, sobre todo no verbales, del engaño. La perspectiva más comprensiva es la de Zuckerman, DePaulo y Rosenthal (1981), que propusieron que al mentir las personas pueden experimentar uno o varios de estos cuatro procesos: (a) activación psicofisiológica (*arousal*), (b) determinadas emociones (como el

sentimiento de culpa por mentir o el miedo a ser descubierto), (c) la dificultad cognitiva que entraña inventar una mentira y (d) el desafío de controlar la propia conducta para inhibir posibles señales delatorias de engaño. A su vez, cada uno de estos cuatro procesos daría lugar a ciertas señales conductuales. Así, por ejemplo, el *arousal* incrementaría los movimientos corporales nerviosos, haría que la voz sonara más aguda. (Masip, 2020).

A menudo, esta orientación no verbal se ha visto ligada a una perspectiva *emocional* (más que cognitiva) de la mentira. Hay diversas maneras de entender la influencia de las emociones sobre la conducta al mentir. Una de ellas, la llamada “hipótesis del filtraje” (Ekman y Friesen, 1969), afirma que al mentir la persona experimenta determinadas emociones que hay que ocultar o enmascarar (puesto que podrían revelar la verdad), pero que estas emociones se “filtran” pese a todo, manifestándose de manera breve o parcial o provocando alteraciones en la conducta observable del mentiroso. Por el contrario, desde orientaciones *cognitivas* se entiende que los posibles indicadores de la mentira obedecen más a factores cognitivos que emocionales. No obstante, esto no implica que las emociones no ejerzan influencia alguna. Existe notable evidencia de que las emociones influyen sobre los procesos cognitivos. Si las emociones alteran los procesos cognitivos y estos a su vez dan lugar a determinados indicadores conductuales, entonces tales conductas pueden estar bajo la influencia indirecta (mediada por lo cognitivo) de las emociones. (Masip, 2020).

2.13. LA VOZ HUMANA

La voz humana es producida en la laringe, cuya parte esencial, la glotis, constituye el verdadero órgano de fonación humano. El aire procedente de los pulmones, es forzado durante la espiración a través de la glotis, haciendo vibrar los dos pares de cuerdas vocales, que se asemejan a dos lengüetas dobles membranáceas. Las cavidades de la cabeza, relacionadas con el sistema respiratorio y nasofaríngeo, actúan como resonadores. (Acústica, 2003).

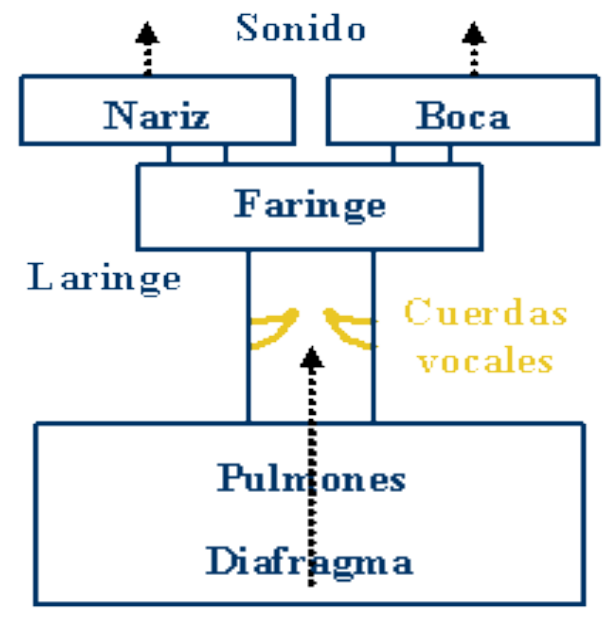


Figura 2.16. La voz humana.

Fuente: Acústica, 2003.

El aparato de fonación puede ser controlado conscientemente por quien habla o canta. La variación de la intensidad depende de la fuerza de la espiración. En el hombre las cuerdas vocales son algo más largas y más gruesas que en la mujer y el niño, por lo que produce sonidos más graves. La extensión de las voces es aproximadamente de dos octavas para cada voz. (Acústica, 2003).

Aunque el tono y la intensidad del habla están determinados principalmente por la vibración de las cuerdas vocales, su espectro está fuertemente determinado por las resonancias del tracto vocal. Los picos que aparecen en el espectro sonoro de las vocales, independientemente del tono, se denominan formantes. Aparecen como envolventes que modifican las amplitudes de los armónicos de la fuente sonora. (Acústica, 2003).

La inteligibilidad oral se debe a las altas frecuencias. Para que el habla sea comprensible, es indispensable la presencia de armónicos cuya frecuencia se halla entre 500 y 3500 Hz. Por otra parte, la energía de la voz está contenida en su mayor parte en las bajas frecuencias y su supresión resta potencia a la voz que suena delgada y con poca energía. (Acústica, 2003).

2.13.1. CARACTERÍSTICAS DE LA VOZ HUMANA

Según Uriarte (2020) los elementos principales que determinan las características de la voz son: el timbre, el tono, la intensidad y la articulación.

- **Timbre:** el modo individual y característico con que la voz es producida pudiendo ser graves o agudas.
- **Tono o entonación:** la modulación en el ritmo de producción de la voz, que permite atribuirle a los sonidos un cariz específico, como la emoción o el acento.
- **Intensidad:** la cantidad de fuerza con que se obtienen los sonidos, pudiendo recaer en algunas partes de la cadena sonora, como en las distintas sílabas de una palabra.
- **Articulación:** Los modos en que la voz se modifica a través de los movimientos de los órganos del aparato fonador, para conseguir sonidos determinados y así formar palabras.

Las principales emociones que manifestamos mediante las características de la voz son:

- **Tristeza:** un volumen bajo y un tono solemne
- **Alegría:** un volumen alto, un tono duro y mayor ritmo
- **Desinterés:** un volumen y un ritmo bajo
- **Nerviosismo:** tono medio-alto y velocidad rápida
- **Sorpresa:** velocidad rápida, tono alto y acentuación pronunciada.
- **Confianza:** volumen alto, ritmo medio y voz decidida.

2.14. SONIDO

El sonido es la propagación de las ondas mecánicas originadas por la vibración de un cuerpo a través de un fluido o un medio elástico. Dichas ondas pueden o no ser percibidas por los seres vivos, dependiendo de su frecuencia. Existen sonidos audibles por el oído humano y otros que solo perciben ciertas especies de animales. Se trata de ondas acústicas producidas por la oscilación de la presión del aire, que son percibidas por el oído y transmitidas al cerebro

para ser interpretadas. En el caso del ser humano, este proceso es esencial para la comunicación hablada. (Coluccio, 2021).

El sonido se produce cuando un cuerpo vibra, y transmite dichas vibraciones al medio circundante en forma de ondas sonoras. Éstas se desplazan expansivamente, a una velocidad promedio (en aire) de 331,5 m/s, y pueden rebotar en distintos tipos de superficies, logrando distintos efectos de eco o de distorsión, que a menudo magnifican su potencia (como en las cajas de resonancia o los parlantes). (Coluccio, 2021).

2.14.1. CARACTERÍSTICAS DEL SONIDO

Según Coluccio (2021) el sonido presenta las siguientes características físicas:

- **Frecuencia:** Es el número de vibraciones completas por segundo que efectúa la fuente del sonido y que se transmite en las ondas. Un sonido audible por los seres humanos tendrá una frecuencia de entre 20 y 20.000 Hz. Por encima de ese rango será un ultrasonido perceptible, a lo sumo, por algunos animales. La siguiente es parte de la terminología relacionada con la frecuencia: hercio (Hz) es igual a un ciclo por segundo. Ciclo es una onda completa de corriente alterna o tensión. Alternancia es la mitad de un ciclo. Período es el tiempo requerido para producir un ciclo completo de una forma de onda. La frecuencia se utiliza normalmente para describir el funcionamiento del equipo eléctrico.
- **Amplitud:** Es la intensidad (potencia acústica), que solemos llamar «volumen». La amplitud se relaciona con la cantidad de energía transmitida por las ondas sonoras.
- **Longitud de onda:** Es la distancia que recorre una onda en un período de oscilación o la distancia entre dos máximos consecutivos de la oscilación.
- **Potencia acústica:** Es la cantidad de energía emitida por las ondas por unidad de tiempo. Se mide en vatios y depende directamente de la amplitud de onda.
- **Espectro de frecuencia:** Es la distribución de amplitudes, o energía acústica, para cada frecuencia de las diversas ondas que componen el sonido.

2.15. PYTHON

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. (Alvarez, 2003).

Según Alvarez (2003) en los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac y otros.

2.16. TENSORFLOW

Machine Learning es una disciplina compleja. Sin embargo, la implementación de modelos es mucho menos desalentadora y difícil de lo que solía ser, gracias a los marcos de Machine Learning, como TensorFlow de Google, que facilitan el proceso de adquisición de datos, el entrenamiento de modelos, la realización de predicciones y el perfeccionamiento de los resultados futuros. (González, 2021).

TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite que los investigadores innoven con el aprendizaje automático y los desarrolladores creen e implementen aplicaciones con tecnología de AA (Aprendizaje Automático) fácilmente. (TensorFlow, 2021).

Utiliza Python para proporcionar una práctica API para crear aplicaciones con el marco de

trabajo, a la vez que ejecuta esas aplicaciones en C++ de alto rendimiento. TensorFlow puede entrenar y ejecutar redes neuronales profundas para la clasificación de dígitos escritos a mano, el reconocimiento de imágenes, la incrustación de palabras, las redes neuronales recurrentes, los modelos secuencia a secuencia para la traducción automática, el procesamiento del lenguaje natural y las simulaciones basadas en ecuaciones diferencias parciales. Lo mejor de todo es que TensorFlow admite predicción de producción a escala, con los mismos modelos utilizados para el entrenamiento. (González, 2021).

2.17. ANDROID STUDIO

Según Developers (2021) Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine

2.18. FLASK

Según Epitech (2021) Flask es un framework que permite desarrollar aplicaciones web de forma sencilla, está especialmente guiado para desarrollo web fácil y rápido con Python. Una de sus características a destacar es el potencial de instalar extensiones o complementos acorde al tipo de proyecto que se desarrolla. Sólo se instala las funcionalidades que en realidad se vaya a utilizar. Las ventajas que presenta son las siguientes:

- Micro Framework: Perfecto si se quiere desarrollar una App básica o que se quiera crear de manera rápida y ágil. Para según qué aplicaciones no se requieren muchas extensiones y con Flask es suficiente.
- Incluye un servidor web de desarrollo: Por lo tanto, no requiere una infraestructura con un servidor web para testar las apps web, simplemente se puede correr un servidor web de forma sencilla con el que se pueden ir observando los resultados que se van obteniendo.
- Tiene un depurador y soporte integrado para pruebas unitarias: Si hay algún error en el código que se está creando, se puede depurar ese error y también se pueden ver los valores de las variables. A su vez, existe la posibilidad de integrar pruebas unitarias.
- Compatibilidad con WSGI: Para servir las páginas web creadas en Python se pueden usar los servidores web con este protocolo, el WSGI.
- Es compatible con Python3.
- Buen manejo de rutas: Cuando se trabaja con Apps Web hechas en Flask Python se tiene el controlador que recibe todas las peticiones que hacen los clientes y se tiene que determinar a qué ruta está accediendo el cliente para ejecutar el código pertinente.
- Soporta de manera nativa el uso de cookies.
- Sin ORMs: No tiene ORMs, pero fácilmente se puede usar una extensión.
- Se pueden usar sesiones.

CAPÍTULO 3

MARCO APLICATIVO

3.1. INTRODUCCIÓN

En el presente capítulo se inicia el análisis y desarrollo del detector de mentiras, para tal propósito se trabajará con la metodología fundamental para la ciencia de datos.

La metodología fundamental para la ciencia de datos nos ayuda a desarrollar el detector de mentiras siguiendo las 10 etapas de la metodología.

El proceso que se tomará para realizar el proyecto es el siguiente, se construye un dataset en base a emociones que son el miedo y enojo para la categoría de estrés y audios neutrales para la categoría de sin estrés, se examina uno por uno para evitar errores y verificar que pertenezcan a dicha emoción.

Una vez construido el dataset se pasa a realizar el preprocesamiento de los audios con MFCC, el entrenamiento con redes neuronales recurrentes. Finalmente se diseña la aplicación y se pasa el modelo a flask para comunicar el modelo con la aplicación.

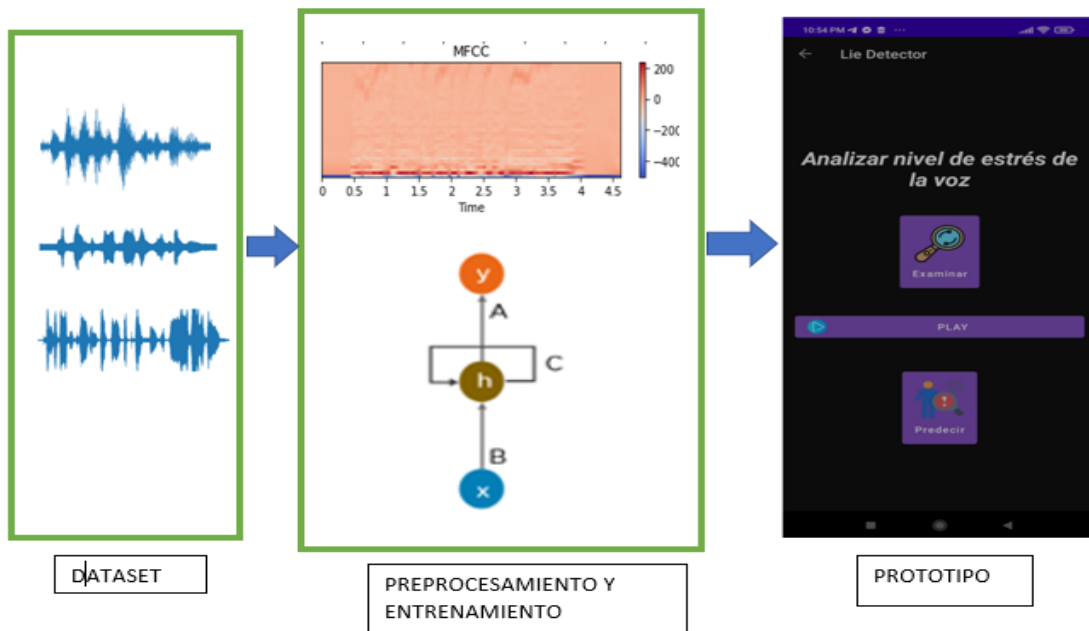


Figura 3.1. Proceso para realizar el proyecto.

En la Figura 3.2. se presenta el proceso de preprocesamiento y entrenamiento. Primeramente, se obtiene con 40 coeficientes el MFCC de cada audio, se tendrá por resultado un vector por cada audio, donde se encuentra las características más importantes del audio. MFCC se encarga de obtener frecuencias que son únicas como la huella digital, por tal motivo que se puede examinar tales características para determinar el estrés. Al obtener los vectores estos pasan a ser divididos para su entrenamiento y validación, 80% para el entrenamiento y 20% para la validación, estos datos divididos ingresaran al modelo secuencial donde se encuentra la primera capa de red neuronal recurrente LSTM y el resultado del modelo arrojara 1 o 0, 0 si existe estrés y 1 si no existe estrés.

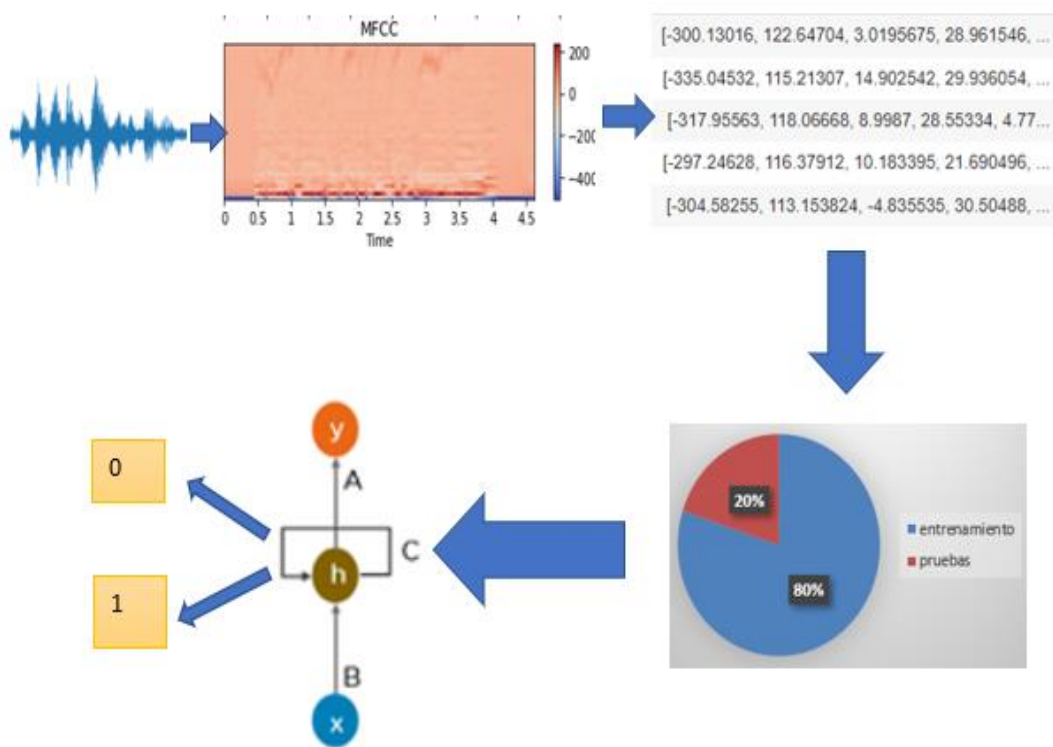


Figura 3.2. Pre-procesamiento y entrenamiento.

3.2. ETAPAS DE LA METODOLOGÍA FUNDAMENTAL PARA LA CIENCIA DE DATOS

3.2.1. ETAPA 1: COMPRENSIÓN DEL NEGOCIO

A continuación, se muestra cada una de las tareas de las que consta esta primera etapa, cuya finalidad es definir el problema, los objetivos del proyecto y los requisitos del proyecto desde una perspectiva de negocio.

3.2.1.1. DEFINICIÓN DEL PROBLEMA

Detectar si una persona miente mediante el análisis de estrés de voz.

3.2.1.2. DETERMINAR LOS OBJETIVOS DEL NEGOCIO

- Realizar predicciones lo más fiable posible del estrés de voz para determinar la mentira a partir de datos recopilados.
- Proporcionar un detector de mentiras que sirva para apoyar a distintas áreas.
- Ayudar al ser humano, mediante la detección de estrés a tomar consciencia y corregir comportamientos en situaciones de estrés.

Contexto: En referencia a la situación de negocio con respecto al detector de mentiras mediante el análisis de estrés de voz, al principio del proyecto se cuenta con dos datasets SAVEE y CREMA-D con sonidos en formato wav para el análisis de estrés de voz que se encontró en Kaggle.

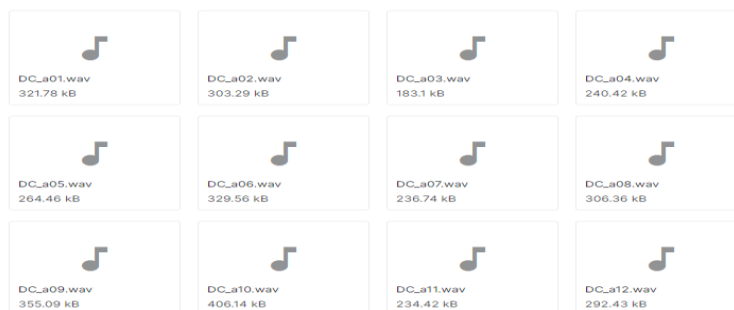


Figura 3.3. Formato de audios de la base de datos SAVEE

En la Figura 3.1. se muestran los audios de SAVEE, en la figura 3.2 se muestran los audios de CREMA-D.

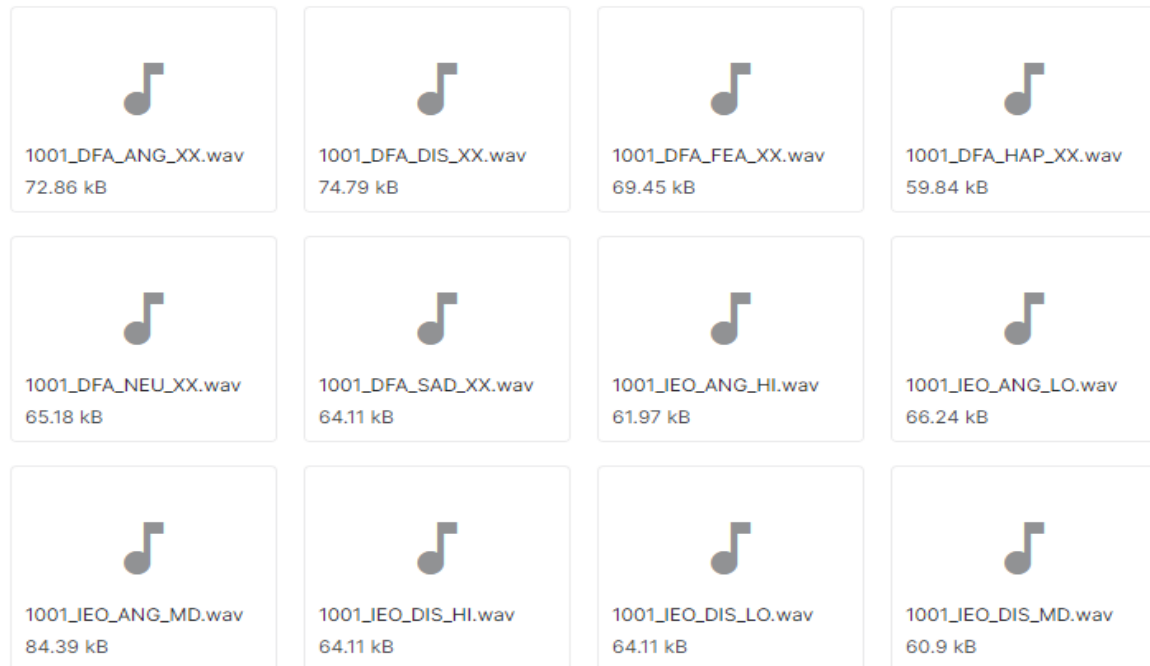


Figura 3.4. Formato de audios de la base de datos CREMA-D

Objetivos del negocio: Como se mencionó anteriormente son la predicción de datos de tal manera que se pueda determinar si una persona miente o dice la verdad.

Criterios de éxito del negocio: Desde el punto de vista del negocio se establece como criterio de éxito la posibilidad de realizar predicciones si una persona miente con un elevado porcentaje de fiabilidad, de tal forma que se pueda aplicar a distintas áreas como: en las empresas, en la parte jurídica entre otras.

Para lo que se realizó una encuesta si usaría el detector de mentiras, como podemos observar en la Figura 3.2. Se tiene que el 92,9 % dice que si lo haría y el 7,1% que no.

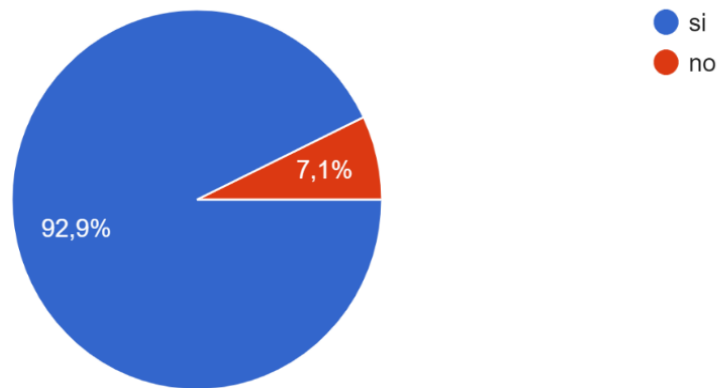


Figura 3.5. Uso del detector de mentiras.

Así también se consultó en qué áreas creen que es más importante o útil el detector de mentiras. Como se puede observar en la Figura 3.3., es más útil en el parte judicial con un 50%, seguido de las empresas con el 28,6 %, con amigos en un 14,3 % y por último con familiares en un 7,1 %.

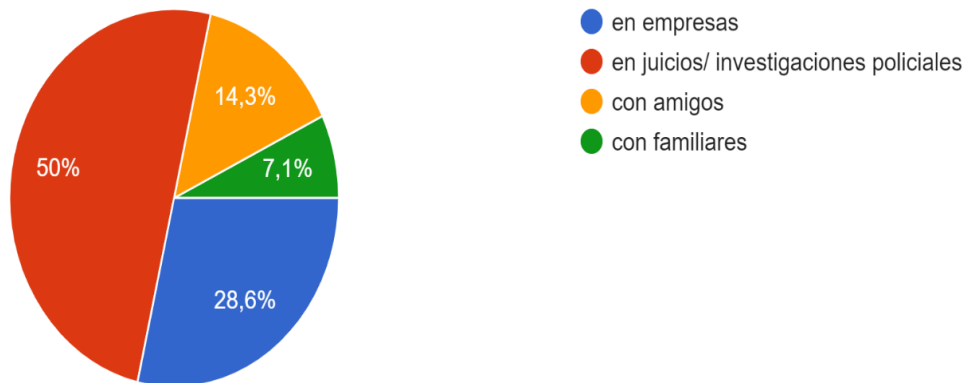


Figura 3.6. Estadística de la importancia del detector de mentiras en distintas áreas.

3.2.1.3. REQUISITOS DEL PROYECTO

En cuanto a recursos externos se obtiene datos que son sonidos clasificados en dos, con estrés y sin estrés, estos sonidos son con extensión.wav.

En cuanto a recursos de software se trabaja en colabory por la facilidad que proporciona,

donde se utiliza Python para realizar el modelo, además para el almacenamiento de datos como carpetas se utiliza Google drive.

En cuanto a recursos para el prototipo se utiliza Android Studio con Java, flask y Python.

Los recursos de hardware con lo que se dispone es un ordenador con las siguientes características:

Series	S145
Marca	Lenovo
Tamaño de pantalla	15.6 Pulgadas
Sistema operativo	Windows 10 Home
Entrada de interfaz humana	Micrófono
Fabricante de CPU	Intel
Descripción de la tarjeta	Integrated
Cantidad de procesadores	4
Marca Chipset	Intel
Modelo de CPU	Core i7

Figura 3.7. Recursos de hardware.

3.2.2. ETAPA 2: ENFOQUE ANALÍTICO

El proyecto se dividirá en las siguientes fases para facilitar su organización y estimar el tiempo de realización del mismo

Fase 1: Análisis de la estructura de los datos y la información de la base de datos. Tiempo estimado: 2 semanas.

Fase 2: Preparación de los datos (selección, limpieza, conversión y formateo, si fuera necesario) para facilitar la minería de datos sobre ellos. Tiempo estimado: 3 semanas.

Fase 3: Elección de las técnicas de modelado y ejecución de las mismas sobre los datos.
Tiempo estimado: 2 semanas.

Fase 4: Análisis de los resultados obtenidos en la etapa anterior. Tiempo estimado: 1 semana.

Fase 5: Producción de informes con los resultados obtenidos en función de los objetivos de negocio y los criterios de éxito establecidos. Tiempo estimado: 2 semanas.

Fase 6: Presentación de los resultados finales. Tiempo estimado: 2 semanas.

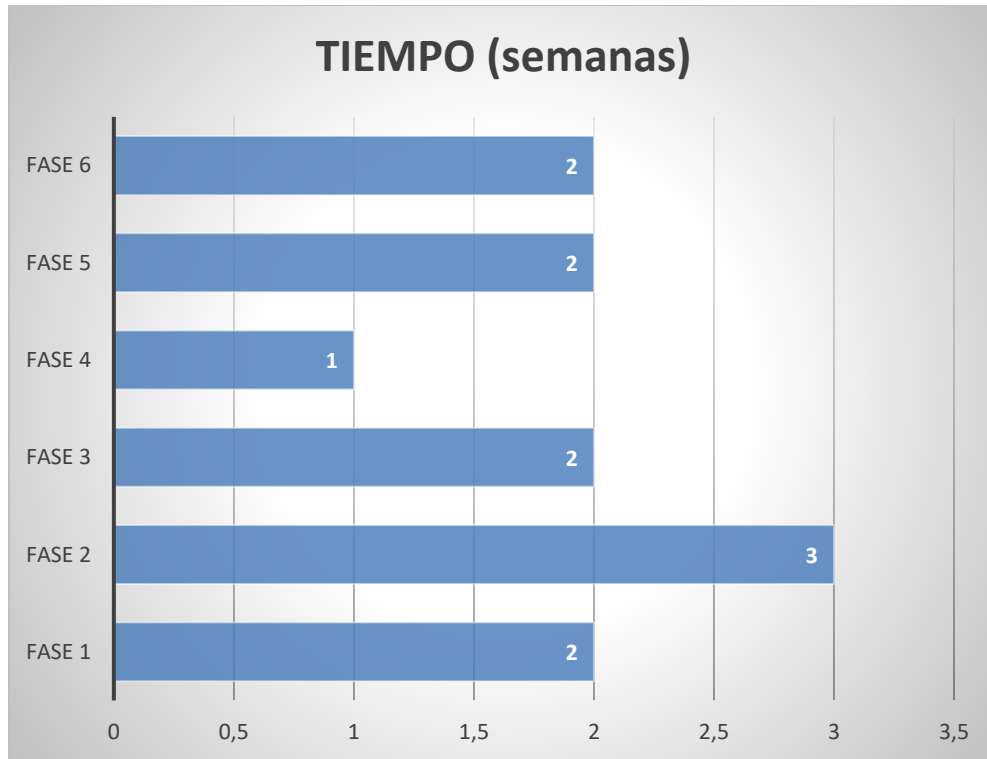


Figura 3.8. Tiempo estimado del desarrollo del proyecto.

3.2.3. ETAPA 3: REQUISITOS DE DATOS

Para realizar el proyecto se debe utilizar audios en formato wav., porque se utilizará la librería librosa para cargar los audios que, para ser leído, deberá ser en un formato sin comprensión de datos, que son en formato wav o ogg.

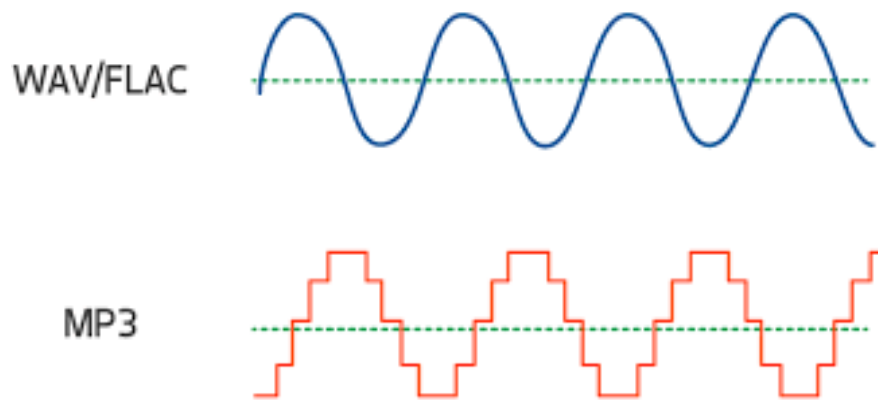


Figura 3.9. WAV vs MP3

Fuente: Tables, 2020.

3.2.4. ETAPA 4: RECOPIACIÓN DE DATOS

En esta etapa se realiza la recolección inicial de los datos para poder establecer un primer contacto con el problema. Los datos que se requieren son audios en formato wav. El formato wav es uno de los formatos de audio más comunes. Fue desarrollado por Microsoft (en colaboración con IBM) y normalmente guarda datos de sonido con modulación por codificación de pulsos (PCM), pero puede utilizarse para almacenar sonido procesado con otros codecs de audio.

Para el desarrollo del trabajo se requirieron muestras de voz como objeto de análisis, partiendo de la condición inicial de que las muestras debían presentar los estados de estrés y sin estrés. Para lo cual se encontró las siguientes bases de datos SAVEE y CREMA-D.

La base de datos SAVEE fue grabada con cuatro hablantes nativos del idioma inglés, estos se identifican como DC, JR, JK y KL, todos son estudiantes e investigadores en la universidad de Surrey y se ubican entre los 27 y 31 años. (Kaggle, 2019).

Las emociones fueron discretizadas en varias categorías como lo son: enfado, disgusto, miedo, felicidad, tristeza, sorpresa y neutral, se decidió usar esta base de datos debido a que las grabaciones están soportadas por estudios sobre reconocimiento automático de emociones.

Tabla 3.1. Base de datos SAVEE

SAVEE			
Nº	SIGLA	EMOCIÓN	CANTIDAD
1	a	enfado	60
2	d	disgusto	60
3	f	miedo	60
4	h	felicidad	60
5	sa	tristeza	60
6	su	sorpresa	60
7	n	neutral	120

En la tabla 3.1. Se muestra las características de la base de datos SAVEE donde se tiene 7 emociones, la sigla de cada emoción y finalmente la cantidad por emoción. La sigla nos sirve para reconocer el tipo de emoción en los audios de la base de datos que presentan la siguiente estructura de la Figura 3.5., donde se tiene primero las siglas del nombre de la persona, seguido de la sigla de la emoción y el número de audio correspondiente al tipo de emoción de la persona.

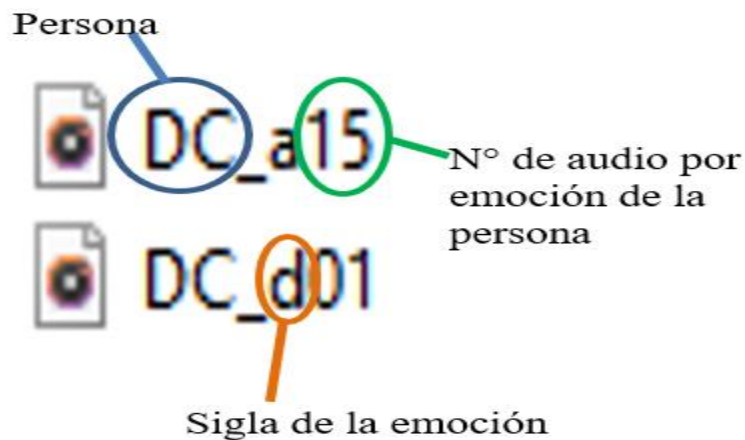


Figura 3.10. Estructura de los audios SAVEE

La base de datos CREMA-D es interesante porque es un conjunto de datos que tiene una gran variedad de datos que ayuda a entrenar un modelo que se puede generalizar a través de nuevos conjuntos de datos. Muchos conjuntos de datos de audio utilizan un número limitado de altavoces, lo que provoca una gran fuga de información. CREMA-D tiene muchos oradores. Por este hecho, CREMA-D es un muy buen conjunto de datos para garantizar que el modelo no se sobreajuste. (Kaggle, 2019).

CREMA-D es un conjunto de datos de 7442 clips originales de 91 actores. Estos clips fueron de 48 actores masculinos y 43 femeninos entre las edades de 20 y 74 años provenientes de una variedad de razas y etnias (afroamericanos, asiáticos, caucásicos, hispanos y no especificados). Los actores hablaron de una selección de 12 frases. Las oraciones se presentaron usando una de seis emociones diferentes (Ira, Disgusto, Miedo, Feliz, Neutral y Triste) y cuatro niveles de emoción diferentes (Bajo, Medio, Alto y Sin especificar). (Kaggle, 2019).

3.2.5. ETAPA 5: COMPRENSIÓN DE DATOS

Los datos que se tienen son audios en formato wav lo cual nos ayuda a optimizar el proyecto porque no es necesario convertirlo a dicho formato.

La base de datos SAVEE posee grabaciones de 15 frases por emoción, de las cuales son 3 frases comunes, 2 específicas de la emoción y 10 frases genéricas, para el caso de las frases neutrales se tienen las tres frases comunes más las dos frases específicas de cada una de las emociones para un total de 30 frases, de esta manera, por cada hablante se tomaron 30 grabaciones correspondientes a 15 frases del estado de enojo y 15 correspondientes a miedo, las cuales son emociones donde se puede percibir un nivel de estrés en la voz y este conjunto se definió como el estado con estrés, mientras que para el estado sin estrés se tomaron las 30 frases del estado neutral, así se obtuvo un dataset con un total de 240 grabaciones, 120 correspondientes al estado sin estrés y 120 correspondientes al estado con estrés, todas distribuidas de manera uniforme entre los cuatro hablantes.

De la misma manera se realizó la selección de datos del dataset CREMA-D donde se tiene para el estado de estrés 1100 archivos y para el estado sin estrés 896 archivos.

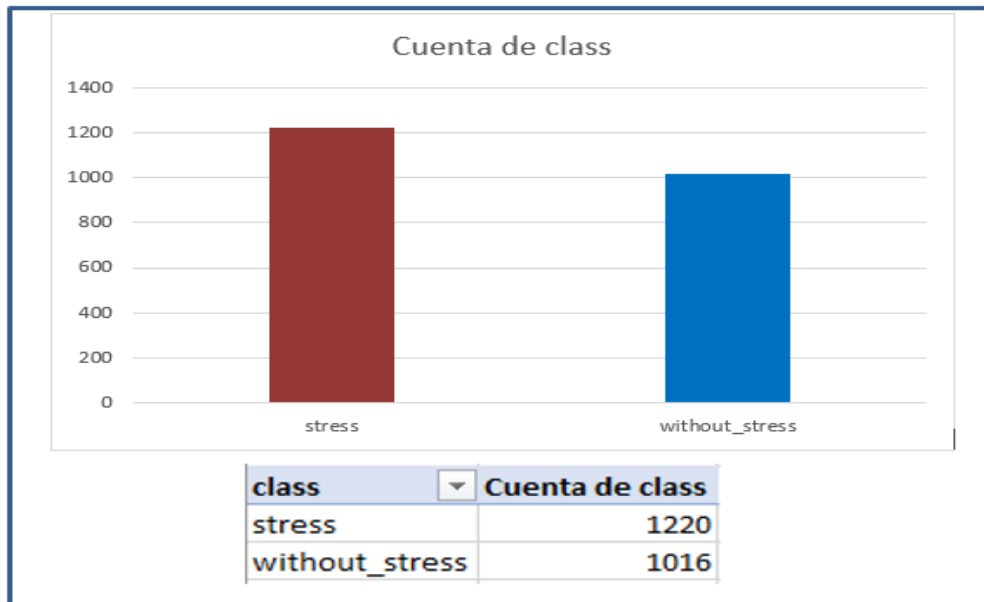


Figura 3.11. Cantidad total de datos en el Dataset a usar.

En la Figura 3.10. Se observa el estado con estrés donde se identifica que los picos son muy elevados además de repetitivos, y el estado sin estrés donde se no se identifica picos elevados tampoco la repetición de ellos.

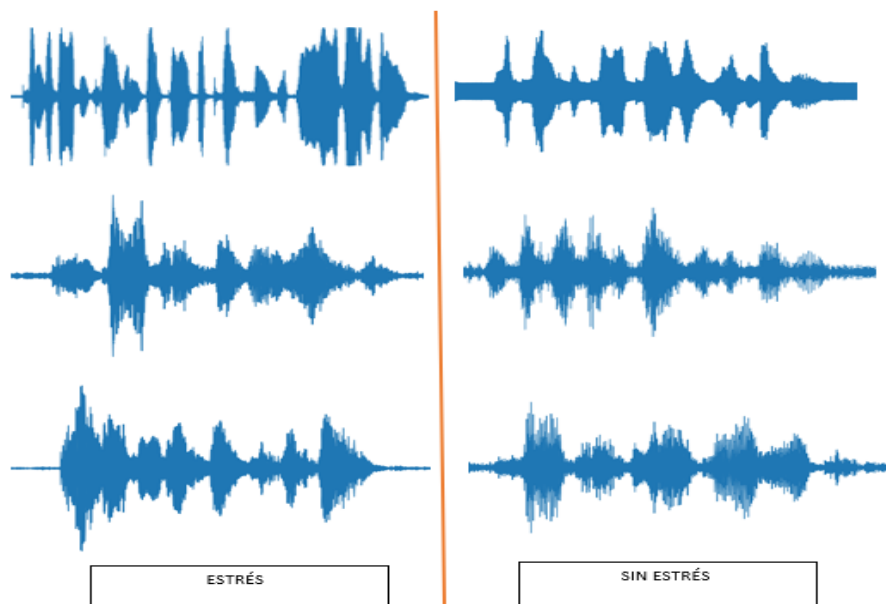


Figura 3.12. Audios con estrés y sin estrés.

A continuación, podemos visualizar comparación de datos con estrés y sin estrés con respecto a espectrogramas.

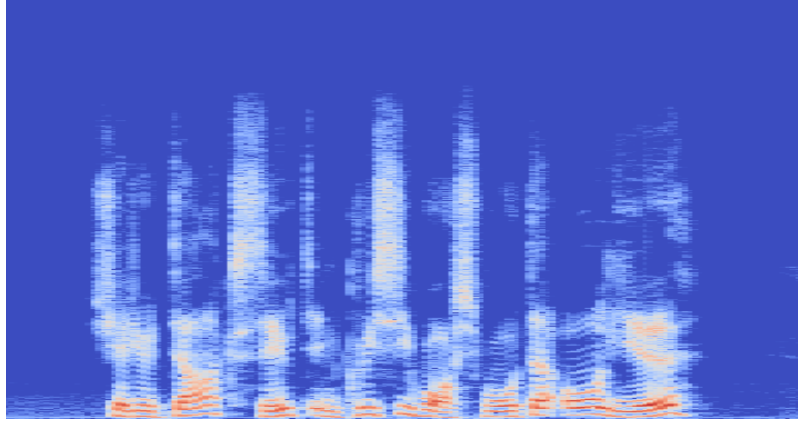


Figura 3.13. Espectrograma del estado con estrés del primer caso.

Realizando una comparación de la Figura 3.10 que es un estado con estrés con la Figura 3.11 del estado sin estrés, se observa en los espectrogramas diferencias con referencia a los picos elevados y repetición de los mismos, En el primero se identifica 8 picos elevados con repetición y en el segundo se observa 3 picos elevados y no hay repetición como en el primero, además que se observa espacios.

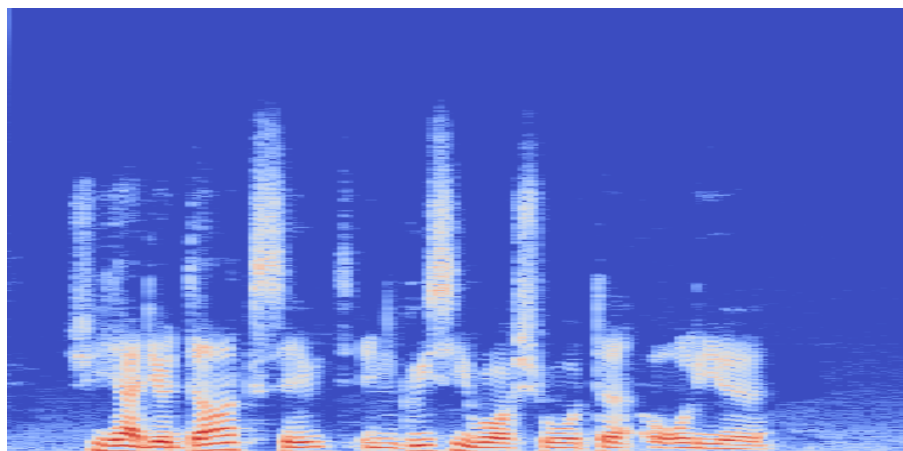


Figura 3.14. Espectrograma del estado sin estrés del primer caso.

Con ayuda del espectrograma de Mel MFCC nos ayuda a realizar un análisis más profundo

del sonido y poder identificar mediante la gráfica las diferencias entre los audios de la base de datos que se tiene para el presente proyecto.

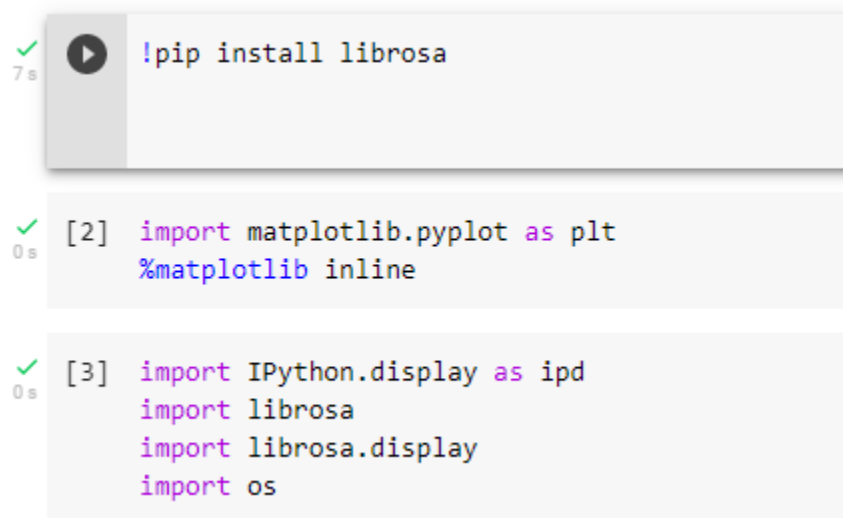
3.2.6. ETAPA 6: PREPARACIÓN DE DATOS

En esta etapa de la metodología se trata de preparar los datos para el modelo que se van a emplear con ellos. Esto implica seleccionar el subconjunto de datos que se va a utilizar, limpiarlos para mejorar su calidad, añadir nuevos datos a partir de los existentes y darles el formato requerido por la herramienta de modelado.

En un principio se tenía 240 datos del dataset SAVEE, sin embargo, la cantidad de datos es muy pequeña por lo cual se selecciona 1996 datos del dataset CREMA-D para robustecer la cantidad de datos que se tiene.

3.2.6.1. SELECCIONAR LOS DATOS

Para este proyecto se utilizan los datos referidos a las emociones de enojo y miedo para la parte del estado con estrés como también el neutral para el estado sin estrés. Los audios están en formato wav.



```
✓ 7s !pip install librosa

✓ 0s [2] import matplotlib.pyplot as plt
      %matplotlib inline

✓ 0s [3] import IPython.display as ipd
      import librosa
      import librosa.display
      import os
```

Figura 3.15. Importación de paquetes para visualización del audio.

3.2.6.2. LIMPIEZA DE DATOS

La base de datos con la que se cuenta para el proyecto contiene toda la información necesaria para poder cumplir los objetivos, son datos limpios y por lo tanto no hay necesidad de hacer

una limpieza más profunda sobre ellos. Tampoco tenemos algún estado en los que nos falten audios, no existe audios que no contengan información relevante al proyecto.

Así también se construye el metadata que se visualiza en Anexo, donde se tendrá la información de cada archivo junto a la clase que pertenece si es estrés o sin estrés en formato csv para que sea leído en Python.

Para generar el modelo relacionado con el primer objetivo, es decir, la predicción de la mentira o la verdad, será necesario utilizar una librería para leer los audios como también para tratarlos como ser librosa y MFCC.

Así mismo para el tratamiento de los datos se debe aplicar MFCC a cada uno para obtener características específicas del audio para diferenciar entre un audio que tiene estrés y otro que no lo tiene. También es importante aplicar MFCC porque el modelo debe recibir datos en forma de vector por lo cual no recibirá directamente al audio, se debe realizar una transformación a los datos.

Se realiza primeramente una función para sacar los MFCC de cada audio, en la Figura se tiene la función para extraer características de los audios, ingresa primeramente el archivo. En la primera línea se saca el audio y la frecuencia de muestreo, cargando el archivo con librosa y con un tipo de remuestreo 'kaiser_fast'. Después se saca MFCC donde se le da los siguientes datos para realizar dicho proceso y será el audio, sr es la frecuencia de muestreo del audio y en número de coeficientes MFCC es de 40 y finalmente obtenemos mfccs.

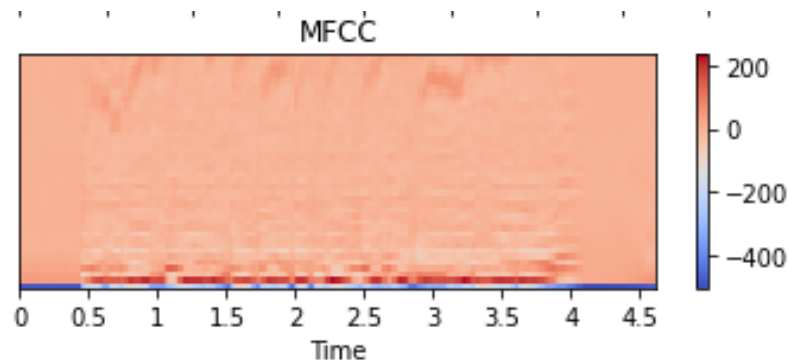


Figura 3.16. Extracción de características MFCC.

A continuación, se accede a la función para la lectura de todos los datos tanto de los archivos .wav y el archivo .csv para obtener las clases o labels. Se realiza iteraciones en la carpeta audio donde se encuentran las dos clases de audios y buscara por el nombre comparando con el nombre y el campo del archivo .csv , sacando el archivo se envía a la función para extraer MFCC ,como se muestra en la Figura 3.22 y se tiene su respectivo código en la Figura 3.23.

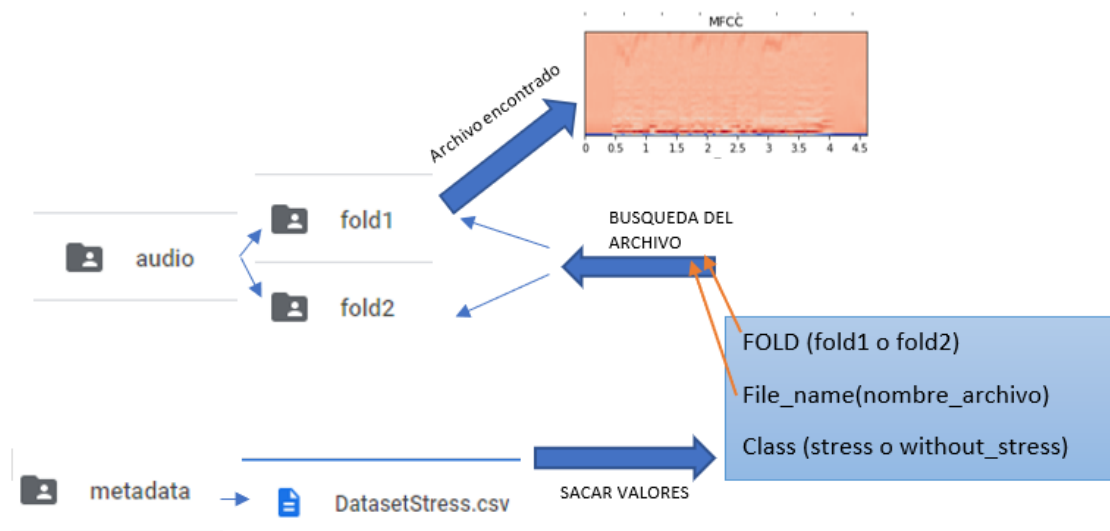


Figura 3.17. Proceso de selección y búsqueda de cada archivo para extraer MFCC.

Se extrajo una característica de cada archivo de audio en el conjunto de datos para su uso en la clasificación: MFCC. Los MFCC se generaron utilizando Librosa (librosa.feature.mfcc), con 40 MFCC generados por archivo de audio, cada MFCC relleno con ceros hasta una longitud fija de 1000.Obteniendo como resultado de cada archivo un vector como se muestra en las Figura 3.24.

	feature	class
0	[-300.13016, 122.64704, 3.0195675, 28.961546, ...	stress
1	[-335.04532, 115.21307, 14.902542, 29.936054, ...	stress
2	[-317.95563, 118.06668, 8.9987, 28.55334, 4.77...	stress
3	[-297.24628, 116.37912, 10.183395, 21.690496, ...	stress
4	[-304.58255, 113.153824, -4.835535, 30.50488, ...	stress
...
195	[-417.63538, 108.74354, 35.42251, 41.251564, 1...	without_stress
196	[-441.4426, 123.60107, 31.101675, 47.32855, 10...	without_stress
197	[-457.14117, 116.23598, 36.340252, 37.334404, ...	without_stress
198	[-465.2257, 151.34636, 28.00042, 38.127228, 9...	without_stress
199	[-443.84122, 137.78188, 32.914318, 44.046074, ...	without_stress

Figura 3.18. Resultado de la extracción de MFCC para cada archivo.

3.2.7. ETAPA 7: MODELADO

En esta etapa de la metodología se escogerá la técnica o técnicas más apropiadas para los objetivos marcados. A continuación, una vez realizado un plan de prueba para los modelos escogidos, se procederá a aplicar dichas técnicas sobre los datos para generar el modelo y por último se tendrá que evaluar si dicho modelo ha cumplido los criterios de éxito o no.

3.2.7.1. TÉCNICA DE MODELADO

En este proyecto se utiliza una red neuronal secuencial llamada red neuronal recurrente LSTM utilizando funciones exclusivamente acústicas en el habla para la detección de mentiras.

3.2.7.2. PLAN DE PRUEBA

Se dividió los datos en dos partes, por un lado, está el conjunto de datos que se van a utilizar para generar el modelo, llamados datos de entrenamiento, y un segundo conjunto de datos que se empleará para realizar las pruebas y medir la calidad del modelo, llamados datos de

prueba o de evaluación. Normalmente se suele utilizar un 60% de los datos para los datos de entrenamiento y el 40% restante para los datos de prueba, pero se utilizó la librería `from sklearn.model_selection import train_test_split` para que pueda seleccionar la cantidad de datos al azar para evitar el overfitting. La función `train_test_split` añade al bloque de entrenamiento el 80% de los registros, y al bloque de pruebas el 20% restante.

```
#Dividiendo train y test
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

Figura 3.19. Código para la división del dataset.

En la Figura 3.26 se visualiza que se está tomando 1788 datos para el entrenamiento y 448 datos para pruebas.

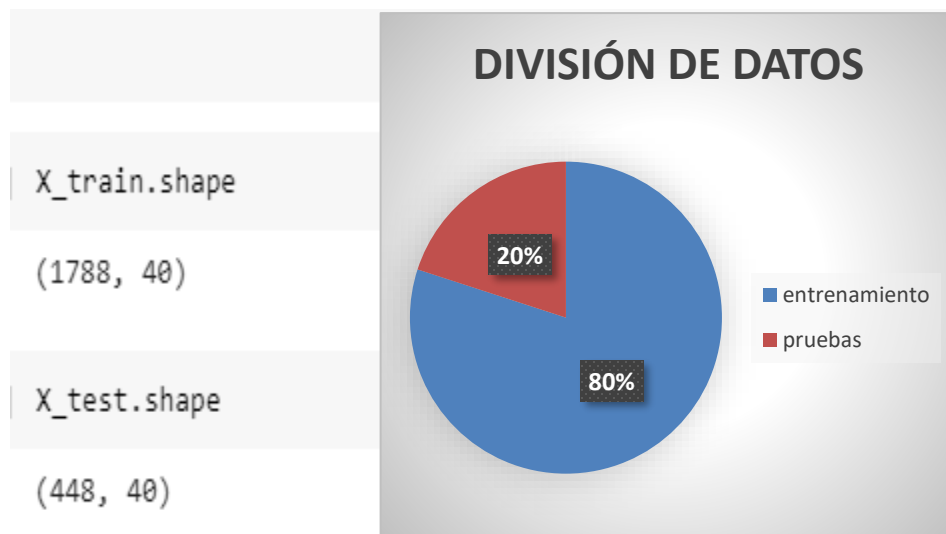


Figura 3.20. División de datos y forma de los datos.

3.2.7.3. CONSTRUCCIÓN DEL MODELO

Aparte de los modelos más simples proporcionados en scikit-learn, los modelos secuenciales, como el neural recurrente La red RNR se ha adoptado ampliamente para el reconocimiento de voz secuencial y el lenguaje natural. tareas de procesamiento. En particular, la memoria a corto plazo (LSTM) es una arquitectura RNR que permite un rendimiento de vanguardia en el reconocimiento de voz debido a su capacidad para recordar patrones a largo plazo mejor que un simple RNR.

Se implementó un modelo secuencial basado en LSTM utilizando la API de Keras para estratificación secuencial. Se utiliza una capa LSTM porque analiza secuencias, con 50 neuronas y una capa Densa con 2 neuronas como salida porque se tiene dos respuestas. El modelo aquí usa la pérdida softmax función que genera probabilidades para cada clase que suman 1, y selecciona la probabilidad más alta como la selección de la clase predicha. Softmax se seleccionó porque se tiene como función de perdida categorical_crossentropy aplicada a clases además que se basa en la suposición de que solo 1 clase es correcta de todas las posibles. El optimizador utilizado fue Adam algoritmo de optimización.

```
model=Sequential()

model.add(LSTM(units=50,input_shape= dim_entrada))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy',metrics=['accuracy'],optimizer='adam')
```

Figura 3.21. Modelo LSTM.

Se puede lograr ver en la Figura 3.22 la estructura del modelo usando LSTM.

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50)	10400
activation (Activation)	(None, 50)	0
dropout (Dropout)	(None, 50)	0
dense (Dense)	(None, 2)	102

=====
Total params: 10,502
Trainable params: 10,502
Non-trainable params: 0
=====

Figura 3.22. Model Summary.

En la Figura 3.23. se observa el entrenamiento del modelo con 50 épocas y un `batch_size` de 32, porque que ayudan a cumplir los objetivos establecidos en el capítulo 1. Con 20 épocas nos da una exactitud del 60% y con 100 del 70%, por lo cual no son factibles ninguno de los dos casos.

```
num_epochs = 50
num_batch_size = 32

start = datetime.now()

model.fit(X_train, y_train, batch_size=num_batch_size, epochs=num_epochs, validation_data=(X_test, y_test))
```

Figura 3.23. Entrenamiento.

3.2.8. ETAPA 8: EVALUACIÓN

En esta etapa de la metodología se intentan evaluar los modelos generados, pero en esta ocasión la evaluación se hace desde el punto de vista de los objetivos de negocio. Una vez realizada esta evaluación, se debe decidir si los objetivos han sido cumplidos y de ser así se puede avanzar a la fase de implementación, de lo contrario se tendría que identificar cualquier factor que se haya podido pasar por alto y hacer una revisión del proceso.

3.2.8.1. EVALUAR LOS RESULTADOS

Las métricas de evaluación, como las matrices de confusión se visualizaron y trazaron con SciKit learn, una biblioteca de aprendizaje automático para python.

Se genero la matriz de confusión para las salidas del clasificador LSTM.

```
test_accuracy=model.evaluate(X_test,y_test,verbose=0)
print(test_accuracy[1])
```

0.8125

Figura 3.24. Evaluación del modelo.

En la Figura 3.24 se llega a evaluar el modelo donde se llega a un accuracy del 0,8125 donde se visualiza la exactitud del modelo.

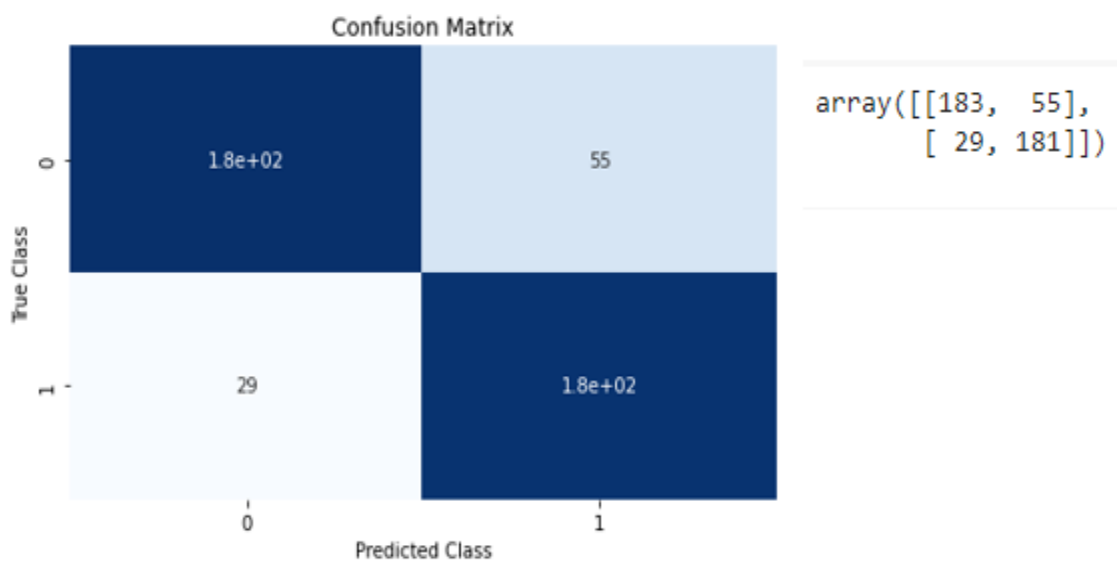


Figura 3.25. Matriz de confusión.

Realizando un análisis de la matriz de confusión de la Figura 3.30 se obtiene:

$$precision = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

$$precision = \frac{183}{183 + 29} = 0,86$$

Se tiene una precisión en el modelo del 86% donde se ve que hay una buena respuesta a la calidad del modelo.

$$Accuracy = \frac{\text{True Positive} + \text{True Negative}}{(\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative})}$$

$$Accuracy = \frac{183 + 181}{183 + 29 + 181 + 55} = 0,81$$

Se tiene una exactitud del 81% esto quiere decir que es el porcentaje en el que el modelo ha acertado.

$$Recall = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

$$Recall = \frac{183}{(183 + 55)} = 0,77$$

Se tiene una exhaustividad del 77% que es la cantidad que el modelo identifica.

$$F1score = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

$$F1score = 2 * \frac{0,86 * 0,77}{(0,86 + 0,77)} = 0,81$$

Se tiene como F1 del 81% que nos ayuda a comparar el rendimiento combinado de Precisión y exhaustividad

3.2.9. ETAPA 9: IMPLEMENTACIÓN

Con todo el estudio realizado hasta este capítulo en el presente trabajo, ya solo queda implementar el prototipo del detector de mentiras.

3.2.9.1. PROTOTIPO

Para tener la ejecución del detector, obligatoriamente se debe tener un smartphone. Consta de una pantalla donde se tiene los siguientes botones: el primero que es examinar que se encarga de recolectar el audio en formato wav, el segundo botón es play para reproducir el audio y de esta manera verificar si es el correcto y el tercer botón es el de predicción que al presionarlo enviará el audio para que pueda ser predicho por el modelo y desplegar un resultado.

El prototipo se implementó en Android Studio donde se incorporó el modelo en Flask para realizar las predicciones con los audios donde muestra dos estados dependiendo de la predicción si es mentira mostrará el mensaje con estrés y si es verdad el mensaje sin estrés.

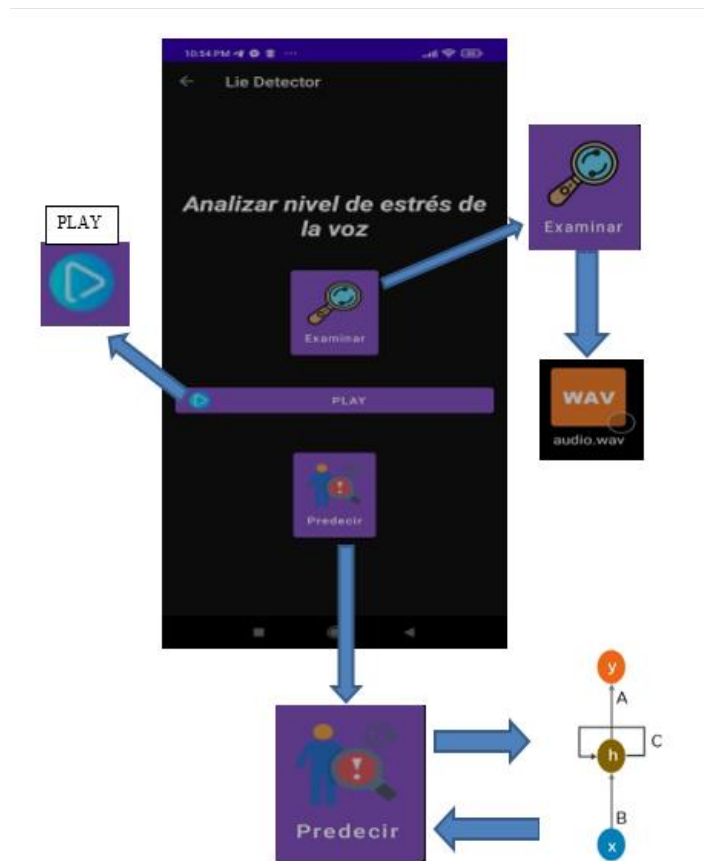


Figura 3.26. Pantalla Lie Detector.

Al presionar el botón predicción que se encuentra en la Figura nos desplegará la pantalla del almacenamiento del celular que se muestra en la Figura.

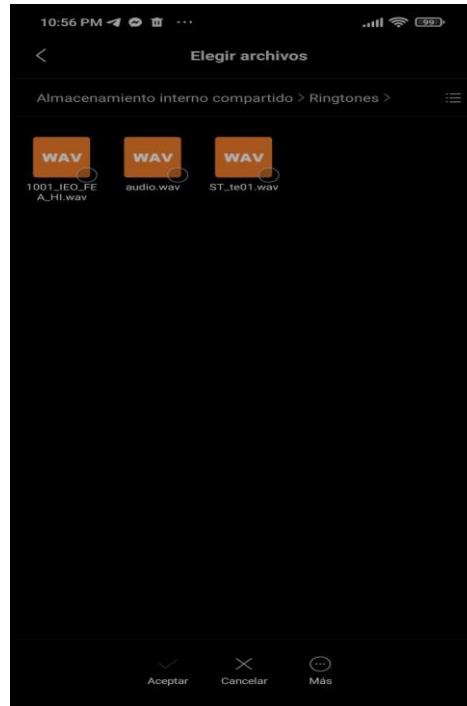


Figura 3.27. Selección de archivos.

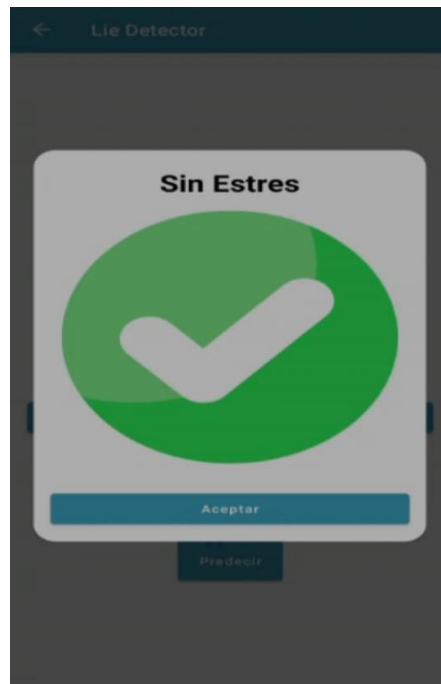


Figura 3.28. Pantalla del detector cuando existe estrés de voz.



Figura 3.29. Pantalla del detector cuando no existe estrés de voz.

Se realizó pruebas con los datos seleccionados para la parte de pruebas en el prototipo donde se obtuvo los siguientes resultados de la Tabla 3.2. donde se tiene el número de prueba, el estado de la predicción y si el modelo acertó o no lo hizo

Tabla 3.2. Resultados.

N°	PREDICIÓN	RESULTADO
1	con estrés	Acertado
2	con estrés	Acertado
3	con estrés	No Acertado
4	con estrés	No Acertado
5	sin estrés	No Acertado
6	con estrés	Acertado
7	sin estrés	Acertado
8	sin estrés	Acertado
9	con estrés	Acertado
10	con estrés	Acertado
11	con estrés	Acertado
12	con estrés	Acertado
13	sin estrés	No Acertado
14	sin estrés	Acertado
15	con estrés	Acertado
16	con estrés	Acertado
17	sin estrés	Acertado
18	sin estrés	Acertado
19	sin estrés	No Acertado

3.2.10. ETAPA 10: RETROALIMENTACIÓN

Al recopilar los resultados del modelo implementado, se obtuvo retroalimentación sobre el rendimiento del modelo y su impacto en el entorno. Se obtuvo las siguientes estadísticas de las pruebas realizadas.

En la figura 3.26. se observa la aceptación que tuvo el detector con un 88,2%, por lo que se puede determinar que se tuvo una buena aceptación al detector como también se pudo visualizar su factibilidad del mismo.

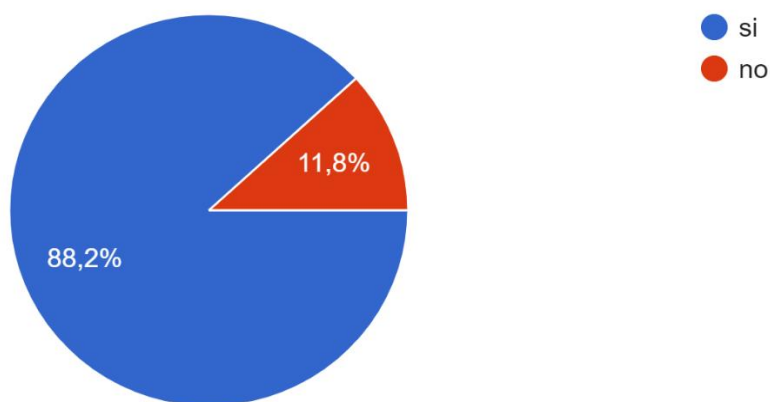


Figura 3.30. Aceptación del detector de mentiras.

CAPÍTULO 4

PRUEBA DE HIPOTESIS

En este apartado se detalla el análisis realizado a los datos de validación del modelo, y sus resultados. Además, en este capítulo se prueba la hipótesis de la tesis expuesta en el capítulo 1.

Hipótesis:

El uso de Redes Neuronales Recurrentes, permite al detector de mentiras realizar un análisis de estrés en la voz y coadyuvar a la evaluación si una persona dice o no lo verdad con un grado de confiabilidad del 80%.

4.1 DATOS DE VALIDACIÓN

El entrenamiento del modelo se realizó con un criterio 80-20, esto quiere decir que el 80% de las muestras del corpus se empleó en el entrenamiento del modelo y el 20% para su validación. Los datos de validación se usan para comparar el rendimiento del modelo como la sensibilidad, precisión, exactitud y demás. Los datos de validación son muestras que se obtienen de los 2236 datos del dataset.

4.2. ANÁLISIS DE RESULTADOS DE VALIDACIÓN

Una herramienta fundamental a la hora de realizar una evaluación a la exactitud del modelo es la matriz de confusión como se muestra en la Figura.

	Positivos	Negativos
Positivos	TP	FP
Negativos	FN	TN

Figura 4.1. Matriz de confusión positivos y negativos.

TP es el número de predicciones correctas.

TN es el número de predicciones negativas correctas.

FN es el número de predicciones negativas cuando los valores reales son positivos.

FP es el número de predicciones negativas cuando los valores reales son positivos.

Con los resultados que despliega la matriz de confusión se puede calcular las métricas como

precisión, exactitud, exhaustividad y F1.

Los resultados obtenidos en la prueba de validación de 2236 muestras son las siguientes:

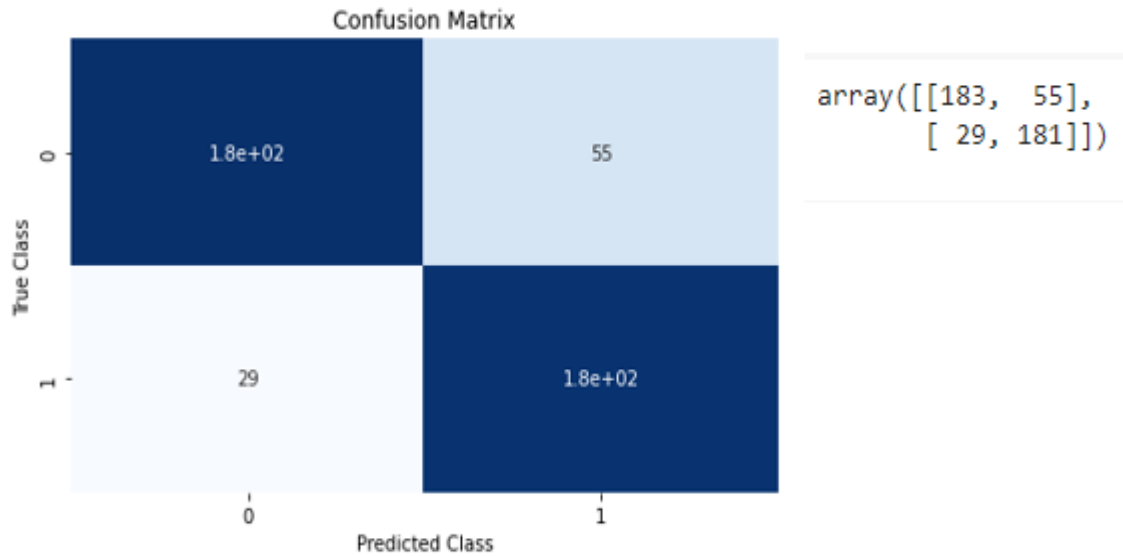


Figura 4.2. Matriz de confusión del modelo.

Donde

TP 183 es el número de predicciones correctas.

TN 181 es el número de predicciones negativas correctas.

FN 29 es el número de predicciones negativas cuando los valores reales son positivos.

FP 55 es el número de predicciones negativas cuando los valores reales son positivos.

4.3. DEMOSTRACIÓN DE LA HIPÓTESIS

Para la demostración de la hipótesis se emplea la métrica de exactitud que proporciona el porcentaje total de los aciertos del modelo para detección del estrés y está definido como:

$$Accuracy = \frac{\text{True Positive} + \text{True Negative}}{(\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative})}$$

Reemplazando los valores obtenidos se obtiene la exactitud:

$$Accuracy = \frac{183+181}{183+29+181+55} = 0,81 \cong 80\%$$

De esta manera se concluye que con redes neuronales recurrentes se logra una exactitud del 80% para la detección del estrés en la voz.

CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Tras el desarrollo del detector de mentiras para detectar si una persona miente o dice la verdad mediante el análisis de estrés de voz, permitió cumplir con el objetivo principal y con los objetivos específicos planteados en el capítulo 1 del presente proyecto.

Del modelo de RNR, se concluye que la detección de estrés es posible. Se ha verificado que una RNR, bien entrenada es capaz de clasificar los datos de manera precisa diferenciando correctamente audios que tienen estrés y sin estrés.

La recolección de datos de audios con estrés y sin estrés es un requerimiento fundamental para realizar el modelo de predicción, la existencia de un conjunto amplio de datos es fundamental para detectar, en ellos características que son únicas extraídas por MFCC para realizar las predicciones. Se encontró que audios de estrés y sin estrés como tal no existen por lo cual se tuvo que construir en base a emociones que presentan estrés como el enojo y miedo.

En cuanto al proceso de entrenamiento se concluye que el modelo inicia con vectores, que presentan las características de cada audio preprocesado por MFCC el cual se encarga de eliminar ruido y frecuencias no necesarias para el audio.

En cuanto a los análisis de resultados obtenidos, se concluye que el objetivo real del entrenamiento es aprender del funcionamiento de las redes neuronales recurrentes LSTM y cómo funciona con los audios, por lo cual se utiliza una capa de LSTM para la entrada de datos y una capa de salida Densa.

Se concluye que las redes neuronales recurrentes LSTM trabajan muy bien con audios y realizan análisis de sus secuencias.

Durante el desarrollo de la investigación, en cada una de sus etapas se pudo evidenciar que el requerimiento fundamental para iniciar el entrenamiento del modelo, es contar con una fuente de datos especializados. A continuación, estos datos deben ser pre-procesados, convertirlos en datos que la red neuronal pueda manejar. En el proceso de entrenamiento del modelo fue entrenado con dos capas, una capa de entrada y una de salida, en la primera con

50 neuronas y la de salida con dos neuronas, como también un dropout solo para evitar overfitting.

En cuanto a la metodología empleada en el proyecto se concluye que es factible para el manejo de datos, sus etapas son completas describiendo paso a paso el proceso que se debe seguir para obtener un modelo.

En cuanto al problema central ¿Cómo detectar si una persona miente? Se concluye que usando redes neuronales recurrentes y procesando los datos con MFCC se puede lograr obtener un modelo que detecte el estrés en la voz determinando que una persona miente.

En cuanto a la hipótesis planteada se logró obtener un 80% de exactitud en el modelo para la detección de estrés en los audios.

En el transcurso de la realización del presente proyecto se optó por realizar una aplicación como prototipo por la facilidad de manejo, además que no existe una aplicación que detecte la mentira sin ser manipulada, por lo cual se concluye que es una herramienta útil y efectiva pero no una herramienta final, solo es de apoyo.

En conclusión, se verifica que el detector logra identificar el estrés de la voz usando redes neuronales recurrentes.

5.2. RECOMENDACIONES

Para la elaboración de futuros trabajos se recomienda:

Implementar en la aplicación la funcionalidad para convertir el audio a wav, esto para facilitar su manejo y no tener la necesidad de utilizar un convertidor de audios extra.

Se recomienda agregar más datos al dataset para obtener un mejor entrenamiento y predicción con el modelo.

Se recomienda agregar una capa oculta LSTM para aumentar la efectividad del mismo modelo.

Se recomienda analizar otras formas de analizar el estrés en los audios aparte de audios con emociones.

Se recomienda complementar el presente trabajo de investigación poder implementar nuevas técnicas de detección de estrés para que el algoritmo sea más preciso.

BIBLIOGRAFÍA

- Acústica. (2003). La voz humana. Recuperado el 20 de octubre de 2021, de:
<http://www.ehu.eus/acustica/espanol/musica/vohues/vohues.html>
- Aguilar, R. (1999). Red Neuronal de Topología Flexible. La Paz, Bolivia.
- Aller, S. (2018). Detectar la mentira a través de la voz. Salamanca, Universidad de Salamanca.
- Alvarez, M. (2003). Qué es Python. Recuperado el 23 de octubre de 2021, de:
<https://desarrolloweb.com/articulos/1325.php>
- Altamirano, S. (2020). Aplicaciones de redes neuronales recurrentes y modelos de series de tiempo bayesianos a la predicción de rentabilidad de fondos de pensiones. Chile, Universidad de Concepción.
- Basogain, X.(s.a.). Redes Neuronales Artificiales y sus Aplicaciones. Escuela Superior de Ingeniería de Bilbao, EHU.
- Belalcazar, H y Bedoya, S. (2015). Detección de mentiras por análisis de imágenes térmicas de rostros. Ingeniero Electrónico. Santiago, Universidad del Valle, Facultad de Ingeniería.
- Benavides, J. (2007). El Arte del Interrogatorio Científico. México.
- Bonet, I. (2007). Redes neuronales recurrentes para el análisis de secuencias. Centro de estudios de Informática, Universidad Central Marta Abreu de las Villas. Cuba.
- Borjas, A. (2021). ¿Qué es una RNN? Red Neuronal Recurrente. Recuperado el 16 de octubre de 2021, de: <https://www.islabit.com/152200/que-es-una-rnn-red-neuronal-recurrente.html>
- Calvo, D. (2018). Red neuronal recurrente-RNN. Recuperado el 15 de junio de 2021, de:
<https://www.diegocalvo.es/red-neuronal-recurrente/>.
- Chi, J. (2012). Medición psicofisiológica:Análisis de estrés de voz en credibilidad de testimonio en internos del CE.RE.SO de Tekax,Yucatán. Licenciado en Psicología.Mérida,Universidad Autónoma Yucatán,Facultad de Psicología.
- Coluccio, E. (2021). Sonido. Recuperado el 18 de noviembre de 2021, de:
<https://concepto.de/sonido/>

- Developers. (2021). Introducción a Android Studio. Recuperado el 10 de noviembre de 2021, de: <https://developer.android.com/studio/intro?hl=es-419>
- Epitech. (2021). ¿Qué es flask (python) y cuáles son sus principales ventajas? Recuperado el 10 de noviembre de 2021, de: <https://www.epitech-it.es/flask-python/>
- Fallas, J. (2012). Intervalos de confianza. Recuperado el 20 de noviembre de 2021, de: https://www.ucipfg.com/Repositorio/MGAP/MGAP-05/BLOQUE-ACADEMICO/Unidad-2/complementarias/intervalo_de_confianza_2012.pdf
- García, U. (2019). Introducción a las Redes Neuronales. Recuperado el 13 de octubre de 2021, de: <https://futurelab.mx/redes%20neuronales/inteligencia%20artificial/2019/06/25/intro-a-redes-neuronales-pt-1/>
- Garrido, E., Masip, J. y Herrero, M. (2006). Psicología Jurídica. Madrid: Pearson.
- Godoy, V. y Dzib, P. (2010). Psicología del Testimonio: Técnicas, instrumentos y procedimientos. En E. García-López, Fundamentos de Psicología Jurídica y forense México: OXFORD.
- Gonzales, F. (2017). Ventanas de Hamming. Recuperado el 25 de octubre de 2021, de : <https://es.scribd.com/document/358481883/Ventanas-de-Hamming>.
- González, L. (2021). ¿Qué es TensorFlow? ¿Cómo funciona? Recuperado el 23 de octubre de 2021, de: <https://aprendeia.com/que-es-tensorflow-como-funciona/>
- Iacond, W. y Patrick, C. (2006). Polygraph ("Lies detector") Testing: Current Status and Emerging Trends. En I. B. Weiner, & A. K. Hess, The handbook of Forensic Psychology. USA.
- Kaggle. (2019). Surrey Audio-Visual Expressed Emotion (SAVEE). Recuperado el 25 de octubre de 2021, de: <https://www.kaggle.com/ejlok1/surrey-audiovisual-expressed-emotion-savee>
- Kaggle. (2019). CREMA-D. Recuperado el 30 de octubre de 2021, de: <https://www.kaggle.com/datasets/ejlok1/cremad>
- Lemos, S. (2005). Simulación, Engaño y Mentira. Revista del consejo General de Colegios Oficiales de Psicólogos. 26, 57-58

- Manzanero, A. (2008). *Psicología del testimonio. Una aplicación de los estudios sobre la memoria*. Madrid: Pirámide.
- Martínez, J. (1995). *Psicofisiología*; Madrid: Editorial Síntesis.
- Martínez, J. (2005). *La Psicología de la Mentira*. México.
- Masip, J., Sporer, S., Garrido, E. y Herrero, C. (2005). The detection of deception with the reality monitoring approach: a review of the empirical evidence. *Psychology, Crime & Law*, 11, 99-122.
- Masip J y Garrido E. (2020). La evaluación de la credibilidad del testimonio en contextos judiciales a partir de indicadores conductuales. *Anuario de Psicología Jurídica*, 93-131.
- Mathur, R. (2015). Performance Comparison of Speaker Identification using vector quantization by MFCC algorithm. Recuperado el 26 de octubre de 2021, de : <https://www.semanticscholar.org/author/nbspDr.Rajeev-Mathur/1405043892>.
- Matich, D. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Universidad Tecnológica Nacional, Facultad Regional Rosario, Departamento de Ingeniería Química.
- Maurnu, B. (2018). ¿Qué son las redes neuronales artificiales? Recuperado el 14 de octubre de 2021, de: <https://brmaurnu201.wordpress.com/2018/01/03/que-son-las-redes-neuronales-artificiales/>
- Muñoz, L. (2011). *Hablemos otra vez del detector de mentiras*.s.l.
- Núñez, M. (2019). ¿Sabes que pasa en tu cerebro cuando te enojas? Recuperado el 20 de agosto de 2021, de: <https://unamglobal.unam.mx/enojarse-resulta-benefico-para-el-cerebro/>
- Ponce, J., Torres, A., Quezada, F., Silva, A., Martínez, E., Casali, A., ...Pedreño, O. (2014). *Inteligencia Artificial. Primera Edición. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn)*.
- Programmerclick. (2020). Extracción de funciones MFCC. Recuperado el 29 de octubre de 2021, de: <https://programmerclick.com/article/9408130436/>
- RAE. (2019). *Diccionario de la lengua española*. Recuperado el 15 de junio de 2021, de:

- <https://www.rae.es/drae2001/mentira>
- Ramos, J. (2014). Prueba de rachas de Wald-Wolfowitz. Recuperado el 29 de noviembre de 2021, de: <https://prezi.com/qf4mwu2csnsm/prueba-de-rachas-y-prueba-de-rachas-de-wald-wolfowitz/>
- Rodríguez, R. (2021). La función de activación. Recuperado el 17 de septiembre de 2021, de: <https://lamaquinaoraculo.com/computacion/la-funcion-de-activacion/>
- Rollins, J. (2015). Metodología Fundamental para la ciencia de datos. EE.UU.: IBM Analytics.
- Sánchez, R. (2016). Extracción de parámetros para reconocimiento de voz. Recuperado el 25 de octubre de 2021, de: <https://rubensm.com/extraccion-de-parametros-para-reconocimiento-de-voz/>
- Segovia, F. (1992). Redes de neuronas recurrentes para el reconocimiento de patrones temporales. España, Universidad Politécnica de Madrid.
- Tables, J. (2020). MP3 versus WAV: Can anyone tell the difference?. Recuperado el 3 de noviembre de 2021, de: <https://www.bhphotovideo.com/explora/pro-audio/features/mp3-versus-wav-can-anyone-tell-the-difference>
- TensorFlow. (2021). Por qué TensorFlow. Recuperado el 23 de octubre de 2021, de: <https://www.tensorflow.org/?hl=es-419>
- Uriarte, J. (2020). Voz. Recuperado el 18 de junio de 2021, de: <https://www.caracteristicas.co/voz/>
- Viñuela, I y León, G. (2004). Redes de Neuronas Artificiales un enfoque práctico. Madrid, España: Editorial Pearson Education, S.A.
- Vrij, A. (2000). Detecting lies and deceit: the psychology of lying and the implications for professional practice. Chichester: John Wiley & Sons.
- Yumpu. (2015). COEFICIENTES MFCC (Mel Frequency Cepstrum Coefficients). Recuperado el 25 de octubre de 2021, de: <https://www.yumpu.com/es/document/read/26670628/coeficientes-mfcc-mel-frequency-cepstrum-coefficients>
- Zuniga, I. (2019). ¿Qué es el Deep Learning? Guía práctica con ejemplos. Recuperado el 16

de octubre de 2021, de: <https://platzi.com/blog/deep-learning-guia-ejemplos/>

ANEXOS

Tabla de una parte del dataset (Metadata)

file_name	Fold	Class
ST_te01.wav	1	Stress
ST_te02.wav	1	Stress
ST_te03.wav	1	Stress
ST_te04.wav	1	Stress
ST_te05.wav	1	Stress
ST_te06.wav	1	Stress
ST_te07.wav	1	Stress
ST_te08.wav	1	Stress
ST_te09.wav	1	Stress
ST_te10.wav	1	Stress
ST_te11.wav	1	Stress
ST_te12.wav	1	Stress
ST_te13.wav	1	Stress
ST_te14.wav	1	Stress
ST_te15.wav	1	Stress
ST_te16.wav	1	Stress
ST_te17.wav	1	Stress
ST_te18.wav	1	Stress
ST_te19.wav	1	Stress
ST_te20.wav	1	Stress
ST_te21.wav	1	Stress
ST_te22.wav	1	Stress
ST_te23.wav	1	Stress
ST_te24.wav	1	Stress
ST_te25.wav	1	Stress
ST_te26.wav	1	Stress
ST_te27.wav	1	Stress
ST_te28.wav	1	Stress
ST_te29.wav	1	Stress
ST_te30.wav	1	Stress
ST_te31.wav	1	Stress
ST_te32.wav	1	Stress
ST_te33.wav	1	Stress
ST_te34.wav	1	Stress

ST_te35.wav	1	Stress
ST_te36.wav	1	Stress
ST_te37.wav	1	Stress
ST_te38.wav	1	Stress
ST_te39.wav	1	Stress
ST_te40.wav	1	Stress
ST_te41.wav	1	Stress
ST_te42.wav	1	Stress
ST_te43.wav	1	Stress
ST_te44.wav	1	Stress
ST_te45.wav	1	Stress
ST_te46.wav	1	Stress
ST_te47.wav	1	stress
ST_te48.wav	1	stress
ST_tr001.wav	1	stress
ST_tr002.wav	1	stress
ST_tr003.wav	1	stress
ST_tr004.wav	1	stress
ST_tr005.wav	1	stress
ST_tr006.wav	1	stress
ST_tr007.wav	1	stress
ST_tr008.wav	1	stress
ST_tr009.wav	1	stress
ST_tr010.wav	1	stress
ST_tr011.wav	1	stress
ST_tr012.wav	1	stress
ST_tr013.wav	1	stress
ST_tr014.wav	1	stress
ST_tr015.wav	1	stress
ST_tr016.wav	1	stress
ST_tr017.wav	1	stress
ST_tr018.wav	1	stress
ST_tr019.wav	1	stress
ST_tr020.wav	1	stress
ST_tr021.wav	1	stress
ST_tr022.wav	1	stress
ST_tr023.wav	1	stress
ST_tr024.wav	1	stress

ST_tr025.wav	1	stress
ST_tr026.wav	1	stress
ST_tr027.wav	1	stress
ST_tr028.wav	1	stress
ST_tr029.wav	1	stress
ST_tr030.wav	1	stress
ST_tr031.wav	1	stress
ST_tr032.wav	1	stress
ST_tr033.wav	1	stress
ST_tr034.wav	1	stress
ST_tr035.wav	1	stress
ST_tr036.wav	1	stress
ST_tr037.wav	1	stress
ST_tr038.wav	1	stress
ST_tr039.wav	1	stress
ST_tr040.wav	1	stress
ST_tr041.wav	1	stress
ST_tr042.wav	1	stress
ST_tr043.wav	1	stress
ST_tr044.wav	1	stress
ST_tr045.wav	1	stress
ST_tr046.wav	1	stress
ST_tr047.wav	1	stress
ST_tr048.wav	1	stress
ST_tr049.wav	1	stress
ST_tr050.wav	1	stress
ST_tr051.wav	1	stress
ST_tr052.wav	1	stress
ST_tr053.wav	1	stress
ST_tr054.wav	1	stress
ST_tr055.wav	1	stress
ST_tr056.wav	1	stress
ST_tr057.wav	1	stress
ST_tr058.wav	1	stress
ST_tr059.wav	1	stress
ST_tr060.wav	1	stress
ST_tr061.wav	1	stress
ST_tr062.wav	1	stress

ST_tr063.wav	1	stress
ST_tr064.wav	1	stress
ST_tr065.wav	1	stress
ST_tr066.wav	1	stress
ST_tr067.wav	1	stress
ST_tr068.wav	1	stress
ST_tr069.wav	1	stress
ST_tr070.wav	1	stress
ST_tr071.wav	1	stress
ST_tr072.wav	1	stress
WS_te01.wav	2	without_stress
WS_te02.wav	2	without_stress
WS_te03.wav	2	without_stress
WS_te04.wav	2	without_stress
WS_te05.wav	2	without_stress
WS_te06.wav	2	without_stress
WS_te07.wav	2	without_stress
WS_te08.wav	2	without_stress
WS_te09.wav	2	without_stress
WS_te10.wav	2	without_stress
WS_te11.wav	2	without_stress
WS_te12.wav	2	without_stress
WS_te13.wav	2	without_stress
WS_te14.wav	2	without_stress
WS_te15.wav	2	without_stress
WS_te16.wav	2	without_stress
WS_te17.wav	2	without_stress
WS_te18.wav	2	without_stress
WS_te19.wav	2	without_stress
WS_te20.wav	2	without_stress
WS_te21.wav	2	without_stress
WS_te22.wav	2	without_stress
WS_te23.wav	2	without_stress
WS_te24.wav	2	without_stress
WS_te25.wav	2	without_stress
WS_te26.wav	2	without_stress
WS_te27.wav	2	without_stress
WS_te28.wav	2	without_stress

WS_te29.wav	2	without_stress
WS_te30.wav	2	without_stress
WS_te31.wav	2	without_stress
WS_te32.wav	2	without_stress
WS_te33.wav	2	without_stress
WS_te34.wav	2	without_stress
WS_te35.wav	2	without_stress
WS_te36.wav	2	without_stress
WS_te37.wav	2	without_stress
WS_te38.wav	2	without_stress
WS_te39.wav	2	without_stress
WS_te40.wav	2	without_stress
WS_te41.wav	2	without_stress
WS_te42.wav	2	without_stress
WS_te43.wav	2	without_stress
WS_te44.wav	2	without_stress
WS_te45.wav	2	without_stress
WS_te46.wav	2	without_stress
WS_te47.wav	2	without_stress
WS_te48.wav	2	without_stress
WS_tr01.wav	2	without_stress
WS_tr02.wav	2	without_stress
WS_tr03.wav	2	without_stress
WS_tr04.wav	2	without_stress
WS_tr05.wav	2	without_stress
WS_tr06.wav	2	without_stress
WS_tr07.wav	2	without_stress
WS_tr08.wav	2	without_stress
WS_tr09.wav	2	without_stress
WS_tr10.wav	2	without_stress
WS_tr11.wav	2	without_stress
WS_tr12.wav	2	without_stress
WS_tr13.wav	2	without_stress
WS_tr14.wav	2	without_stress
WS_tr15.wav	2	without_stress
WS_tr16.wav	2	without_stress
WS_tr17.wav	2	without_stress
WS_tr18.wav	2	without_stress

WS_tr19.wav	2	without_stress
WS_tr20.wav	2	without_stress
WS_tr21.wav	2	without_stress
WS_tr22.wav	2	without_stress
WS_tr23.wav	2	without_stress
WS_tr24.wav	2	without_stress
WS_tr25.wav	2	without_stress
WS_tr26.wav	2	without_stress
WS_tr27.wav	2	without_stress
WS_tr28.wav	2	without_stress
WS_tr29.wav	2	without_stress
WS_tr30.wav	2	without_stress
WS_tr31.wav	2	without_stress
WS_tr32.wav	2	without_stress
WS_tr33.wav	2	without_stress
WS_tr34.wav	2	without_stress
WS_tr35.wav	2	without_stress
WS_tr36.wav	2	without_stress
WS_tr37.wav	2	without_stress
WS_tr38.wav	2	without_stress
WS_tr39.wav	2	without_stress
WS_tr40.wav	2	without_stress
WS_tr41.wav	2	without_stress
WS_tr42.wav	2	without_stress
WS_tr43.wav	2	without_stress
WS_tr44.wav	2	without_stress
WS_tr45.wav	2	without_stress
WS_tr46.wav	2	without_stress
WS_tr47.wav	2	without_stress
WS_tr48.wav	2	without_stress
WS_tr49.wav	2	without_stress
WS_tr50.wav	2	without_stress
WS_tr51.wav	2	without_stress
WS_tr52.wav	2	without_stress
WS_tr53.wav	2	without_stress
WS_tr54.wav	2	without_stress
WS_tr55.wav	2	without_stress
WS_tr56.wav	2	without_stress

WS_tr57.wav	2	without_stress
WS_tr58.wav	2	without_stress
WS_tr59.wav	2	without_stress
WS_tr60.wav	2	without_stress
WS_tr61.wav	2	without_stress
WS_tr62.wav	2	without_stress
WS_tr63.wav	2	without_stress
WS_tr64.wav	2	without_stress

Código de la implementación del modelo en Flask

```

app.py > ...
24
25 app = Flask(__name__)
26 app.config['UPLOADER_FOLDER'] = './Archivos'
27
28 def features_extractor(file_name):
29     audio, sample_rate = librosa.load(file_name, res_type='kaiser_fast')
30     mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
31     mfccs_scaled_features = np.mean(mfccs_features.T, axis=0)
32     return mfccs_scaled_features
33
34 @app.route('/')
35 def main():
36     return "modelo cargado"
37
38 @app.route('/uploader', methods=['POST'])
39 def predict_model():
40     if request.method == "POST":
41         f = request.files['archivo']
42         #se genera el nombre del archivo par aque no se repita en segundos con milisegundos
43         curr_dt = datetime.now()
44         timestamp = int(round(curr_dt.timestamp()))
45         newRoute = str(timestamp) + ".wav"
46         f.save(os.path.join(app.config['UPLOADER_FOLDER'], newRoute))
47
48         ##### Modelo de tesis #####
49
50         audio_dataset_path = './audio/'
51         metadata = pd.read_csv('./metadata/DatasetStress.csv')
52
53         extracted_features = []
54         for index_num, row in tqdm(metadata.iterrows()):
55             file_name = os.path.join(os.path.abspath(audio_dataset_path), 'fold'+str(row["fold"])+ '/' + str(row["file_name"]))
56             final_class_labels = row["class"]
57             data = features_extractor(file_name)
58             extracted_features.append([data, final_class_labels])
59
60         extracted_features_df = pd.DataFrame(extracted_features, columns=['feature', 'class'])

```

```

61     X=np.array(extracted_features_df['feature'].tolist())
62     y=np.array(extracted_features_df['class'].tolist())
63     labelencoder=LabelEncoder()
64     y=to_categorical(labelencoder.fit_transform(y))
65     X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
66
67     #creacion del modelo
68     num_labels=y.shape[1]
69     dim_entrada=(X_train.shape[1],1)
70
71     #definiendo modelo
72     model=Sequential()
73     model.add(LSTM(units=50,input_shape= dim_entrada))
74     model.add(Activation('relu'))
75     model.add(Dropout(0.5))
76     model.add(Dense(2, activation='softmax'))
77     model.compile(loss='categorical_crossentropy',metrics=['accuracy'],optimizer='adam')
78
79     #numero de epocas
80     num_epochs = 50
81     num_batch_size = 32
82     start = datetime.now()
83     model.fit(X_train, y_train, batch_size=num_batch_size, epochs=num_epochs, validation_data=(X_test, y_test)) #, callbacks=[checkpo
84     duration = datetime.now() - start
85
86     test_accuracy=model.evaluate(X_test,y_test,verbose=0)
87     print(test_accuracy[1])
88

```

Predicción del modelo

```

app.py  X  requirements.txt
app.py > predict_model
81
82
83 @app.route('/uploader',methods=['POST'])
84 def predict_model():
85     if request.method == "POST":
86         f= request.files['archivo']
87         #se genera el nombre del archivo par aque no se repita en segundos con milisegundos
88         curr_dt = datetime.now()
89         timestamp = int(round(curr_dt.timestamp()))
90         newRoute=str(timestamp)+".wav"
91         f.save(os.path.join(app.config['UPLOADER_FOLDER'],newRoute))
92
93         ##prediccion del modelo
94         filename2="./Archivos/"+newRoute
95
96         audio, sample_rate = librosa.load(filename2, res_type='kaiser fast')
97         mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
98         mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
99
100        mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
101        predicted_label=model.predict(mfccs_scaled_features)
102        classes_label=np.argmax(predicted_label,axis=1)
103
104        ##capturando valor
105        list1 = classes_label.tolist()
106        print(list1[0],"<===== este es el valor de la prediccion")
107        return jsonify({"value":list1[0]})
108
109

```

DOCUMENTACIÓN

La Paz, 31 de marzo de 2022

Señor

Lic. Ruben Alcon Lopez

Director a.i

Carrera de Informática

Presente

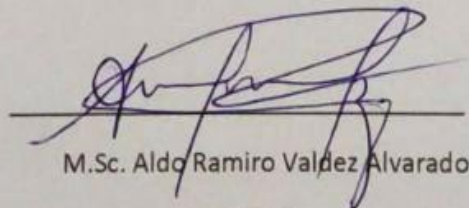
Ref. Aval para Defensa de Tesis de Grado

De mi mayor consideración.

Por intermedio de la presente, y en mi calidad de Tutor, tengo a bien dirigirme a su autoridad, para darle a conocer que luego de efectuar el seguimiento a la estructura y contenido de la Tesis de Grado, titulada "Detector de mentiras mediante el analisis de estrés de voz usando redes neuronales recurrentes ", elaborada por la postulante Carmela Sandra Calliconde Carrasco con C.I. 91866629 LP, me corresponde dar mi CONFORMIDAD Y AVAL, para que el mismo proceda a la DEFENSA PUBLICA, de acuerdo a normas y reglamentos universitarios vigentes.

Sin otro particular me despido de usted con las consideraciones más distinguidas.

Atentamente



M.Sc. Aldo Ramiro Valdez Alvarado
Tutor

av/csc

cc/archivo