UNIVERSIDAD MAYOR DE SAN SIMÓN

POSGRADO FACULTAD DE CIENCIAS Y TECNOLOGÍA

**RECONOCIMIENTO DE VOZ CON MACHINE LEARNING**

Presentado por:

Eguivar Vilca Shirley Carminia

COCHABAMBA - BOLIVIA

Cochabamba 19 de Enero de 2019

INDICE

[1 CAPITULO I - INTRODUCCION 4](#_Toc32754503)

[1.1. ANTECEDENTES 4](#_Toc32754504)

[1.2. DEFINICION DEL PROBLEMA 4](#_Toc32754505)

[1.3 OBJETIVOS 5](#_Toc32754506)

[1.3.1 Objetivo General 5](#_Toc32754507)

[1.3.2 Objetivos Específicos 5](#_Toc32754508)

[1.4. AREA DE CONOCIMIENTO: Machine Learning 5](#_Toc32754509)

[1.5. ALCANCE 5](#_Toc32754510)

[1.6. JUSTIFICACION 5](#_Toc32754511)

[2 CAPITULO II - MARCO TEORICO 7](#_Toc32754512)

[2.1. INTELIGENCIA ARTIFICIAL 7](#_Toc32754513)

[2.2. MACHINE LEARNING 7](#_Toc32754514)

[2.2.1. Tipos de aprendizaje 7](#_Toc32754515)

[2.2.1.1. Aprendizaje Supervisado 7](#_Toc32754516)

[2.2.1.2. Aprendizaje No Supervisado 8](#_Toc32754517)

[2.3. DEEP LEARNING 9](#_Toc32754518)

[2.4. REDES NEURONALES 10](#_Toc32754519)

[2.4.1.Redes Monocapa 10](#_Toc32754520)

[2.4.1.Redes Multicapa 11](#_Toc32754521)

[3 CAPITULO III 12](#_Toc32754522)

[RECONOCIMIENTO DE VOZ 12](#_Toc32754523)

[3.1. MACHINE LEARNIG CON PYTHON 12](#_Toc32754524)

[3.1.1.Scikit-Learn 12](#_Toc32754525)

[3.1.2. TensorFlow 12](#_Toc32754526)

[3.1.3. Keras 12](#_Toc32754527)

[3.1.4. PyTorch 13](#_Toc32754528)

[3.1.5. Theano 13](#_Toc32754529)

[3.2. ANTECEDENTES - HISTORIA RECONOCIMIENTO DE VOZ 13](#_Toc32754530)

[3.3.MECANISMO VOCAL 15](#_Toc32754531)

[La voz consiste en el sonido producido por un ser humano haciendo uso de sus cuerdas vocales para hablar, cantar, reírse, gritar, chillar, etc. Su frecuencia oscila entre alrededor de 60 a 7000 Hz. 15](#_Toc32754532)

[3.4. PROCESAMIENTO DE AUDIO CON PYTHON 16](#_Toc32754533)

[3.4.1. Librerias de Audio 16](#_Toc32754534)

[3.4.1.1.Librosa 16](#_Toc32754535)

[3.4.1.2. IPython.display.Audio 16](#_Toc32754536)

[3.4.2. Cargando un archivo de audio con Python 17](#_Toc32754537)

[3.4.3. Reproduciendo audio 17](#_Toc32754538)

[3.4.4. Vizualizando Audio 17](#_Toc32754539)

[3.4.4.1.Waveform/ Ondas 17](#_Toc32754540)

[3.4.4.2.Espectograma 18](#_Toc32754541)

[3.4.5. Extraccion de Caracteristicas 19](#_Toc32754542)

[3.4.5.1. Zero Crossing Rate 19](#_Toc32754543)

[3.4.5.2. Spectral Centroid 20](#_Toc32754544)

[3.4.5.3. Spectral Rolloff 21](#_Toc32754545)

[3.4.5.4. Mel-Frequency Ceptral Coefficients 22](#_Toc32754546)

[3.4.5.5. Chroma Frequencies 24](#_Toc32754547)

[4 CAPITULO IV 25](#_Toc32754548)

[RECONOCIMINETO DE COMANDOS DE VOZ 25](#_Toc32754549)

[4.1. INTRODUCCION 25](#_Toc32754550)

[4.2. DATASET 25](#_Toc32754551)

[4.3. PRE-PROCESAMIENTO DE ARCHIVOS DE AUDIO 27](#_Toc32754552)

[4.3.1. Extraccion de espectograma por cada audio 27](#_Toc32754553)

[4.3.1. Extraccion de Caracteristicas desde los espectogramas 28](#_Toc32754554)

[4.4. ENSAMBLADO DEL DATASET 29](#_Toc32754555)

[4.5. ANALIZANDO LOS DATOS EN PANDAS 31](#_Toc32754556)

[4.5.1. Limpieza de datos 32](#_Toc32754557)

[4.6. DIVIDIENDO EL DATASET EN TRAINING AND TEST DATA 33](#_Toc32754558)

[4.7. CLASIFICACION CON KERAS- CONSTRUCCION DE LA REDE NEURONAL 33](#_Toc32754559)

[4.7.1. Ajustes al modelo 34](#_Toc32754560)

[4.8. ENTRENAMIENTO DE LA RED 34](#_Toc32754561)

[4.8.1. Interpretacion de los resultados del entrenamiento 35](#_Toc32754562)

[4.8.2. Evaluacion 36](#_Toc32754563)

[4.9. PREDICCIONES 36](#_Toc32754564)

[4.8.2. Analisis de Resultados 36](#_Toc32754565)

[5 CONCLUSIONES Y RECOMENDACIONES 38](#_Toc32754566)

[6 BIBLIOGRAFIA 39](#_Toc32754567)

## 1 CAPITULO I - INTRODUCCION

*"Machine learning is a core, transformative way by which we’re rethinking everything we’re doing". –* Sundar Pichai

## 1.1. ANTECEDENTES

En los ultimos años el termino de machine learningesta adquiriendo mas notoriedad en el ambitotecnologico.Se ha producido un fuerte crecimiento de la inversion en este campo, el auge de los smartphones ha permitido que inicien a desarrollarse ssoluciones comerciales basadas en machine learning que son usados por una gran cantidad de usuarios como ejemplo tenemos los llamados asistentes virtuales.

Grandes empresas de la industria tecnologica han incorporado los asistemtes virtuales dentro sus dispositivos tales como Apple(Siri),Google (Google assistan y Google Now), Microsoft (Cortana),Amazon (Alexa), Samsung (Bixby) estas aplicaciones permiten que el usuario realice acciones mediante commandos de voz.

Las aplicaciones de reconocimiento de voz incluyen interfaces de usuario de voz como la marcación por voz , enrutamiento de llamadas, control domótico de dispositivos, búsqueda,etc.

Las interfaces de usuario de voz, son las tecnologías que permiten a las personas interactuar con computadoras y dispositivos a través de la entrada de vozhaciendo que los dispositivos actúen de forma inteligente entendiendo y respondiendo a las diferentes instrucciones.

Las tecnologías que están detrás de estas interfaces de usuario de vozutilizan machine learning que combinados con otras tecnologiaspermiten la comunicación de los usuarios con las computadoras.

## 1.2. DEFINICION DEL PROBLEMA

Con el avance de la tecnología hoy en dia, el usuario puede abrir el navegador, consultar la hora, cerrar una ventana en particular, escribir un texto, realizar cálculos todo esto por medio de comandos de voz. El reconocimiento de voz no cambia lo que la computadora solía hacer, pero sí cambia la forma en que el usuario lo hace.

El presente proyecto plantea como problema el reconocimiento de voz utilizando un algorithmo de machine learning.

Como podría reconocer comandos de voz para ejecutar tareas rutinarias a través del uso de la tecnología?

## 1.3 OBJETIVOS

### 1.3.1 Objetivo General

Implementar un modelo de machine learning para reconocimiento de voz para ejecutar tareas rutinarias a través del uso de la tecnologia.

### 1.3.2 Objetivos Específicos

* Denifir el dataset a utilizar por el modelo
* Preparacion y limpieza de datos
* Desarrollar un modelo machine learning utilizando redes neuronales.
* Interpretar los resultados finales

## 1.4. AREA DE CONOCIMIENTO: Machine Learning

## 1.5. ALCANCE

La deteccion de comandos disparadores Keyword Spotting, consite en la deteccion de determinados comandos en una gravacion de audio, y al reconocer dichos comandos de forma automatica el sistema pueda ejecutar las acciones correspondientes.

Se definene como alcances del proyecto los isguientes puntos:

* Si bien en un entorno real el sistema recibe un streem de audio de forma continua,en el presente proyecto, se considera archivos de audio como entradacon una duracion acotada.
* Ademasmencionar que el objetivo principal del proyecto es el reconocimiento de voz aplicando los conocimientos adquiridos de machine learning, por lo que la ejecucion de comandos luego del reconocimiento de voz no sera parte de este proyecto.

## 1.6. JUSTIFICACION

Hoy en dia las interfaces de usuario de voz estan adquiriendo gran importancia en la vida cotidiana, un ejemplo reciente fue lo ocurrido en diciembre 2019, un hombre fue rescatado después de gritar “Siri, llama al 911" cuando su automóvil había caído en un río congelado.(CNN)

Estas aplicaciones de reconocimiento de voz cumplen un papel muy importante dentro el area tecnológica y cada vez es mas común encontrarlas o escuchar de ellas en situaciones del vivir diario como en teléfonos, televisores, hogares inteligentes y otros productos. Los dispositivos controlados por voz, como Apple HomePod, Google Home y Amazon Echo, están adquiriendo una fuerte demanda en el mercado.

Por estas razones existe la necesidad de crear nuevos recursos y herramientas que faciliten el manejo de estas tecnologías.

## 2 CAPITULO II - MARCO TEORICO

## 2.1. INTELIGENCIA ARTIFICIAL

La inteligencia artificial (IA) hace posible que las máquina aprendan de la experiencia, se ajusten a nuevas aportaciones y realicen tareas como hacen los humanos. La mayoría de los ejemplos de inteligencia artificial de los que usted escucha hoy díadesde computadoras que juegan ajedrez hasta automóviles que se conducen por sí solos, se sustentan mayormente en aprendizaje a fondo (deeplearning) y procesamiento del lenguaje natural. Mediante el uso de estas tecnologías, las computadoras pueden ser entrenadas para realizar tareas específicas procesando grandes cantidades de datos y reconociendo patrones en los datos.

## 2.2. MACHINE LEARNING

El Aprendizaje Automático de las máquinas (Machine Learning) tiene como objetivo desarrollar algoritmos que permitan aprender sobre un conjunto de observaciones, de tal manera que sea posible establecer hipótesis generales o predicciones sobre nuevas observaciones, y de esta forma, tomar decisiones automáticamente, dotando así a un sistema de Inteligencia Artificial.

El Aprendizaje Automático es utilizado en una gran variedad de ámbitos, desde el reconocimiento de voz, hasta la exploración o minería de datos, existiendo una gran variedad de algoritmos. Este tipo de algoritmos tienen dos fases fundamentales: la fase de entrenamiento y la fase de evaluación.

### 2.2.1. Tipos de aprendizaje

### 2.2.1.1. Aprendizaje Supervisado

En el **aprendizaje supervisado**, los algoritmos trabajan con datos “etiquetados” (labeled data), intentado encontrar una función que, dadas las variables de entrada (input data), les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico” de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, **predice** el valor de salida. Ver figura 1.

El aprendizaje supervisado se suele usar en:

* Problemas de *clasificación* (identificación de dígitos, diagnósticos, o detección de fraude de identidad).
* Problemas de*regresión* (predicciones meteorológicas, de expectativa de vida, de crecimiento etc).

Estos dos tipos principales de aprendizaje supervisado, clasificación y regresión, se distinguen por el tipo de variable objetivo. En los casos de clasificación, *es de tipo categórico*, mientras que, en los casos de regresión, la variable objetivo *es de tipo numérico*.

Los algoritmos más habituales que aplican para el aprendizaje supervisado son:

* Árboles de decisión.
* Clasificación de Naïve Bayes.
* Regresión por mínimoscuadrados.
* RegresiónLogística.
* Support Vector Machines (SVM).

Métodos “Ensemble” (Conjuntos de clasificadores).

### 2.2.1.2. Aprendizaje No Supervisado

El aprendizaje no supervisado tiene lugar cuando no se dispone de datos “etiquetados”  para el entrenamiento. Sólo conocemos los datos de entrada, pero no existen datos de salida que correspondan a un determinado input. Por tanto, sólo podemos **describir la estructura de los datos**, para intentar encontrar algún tipo de organización que simplifique el análisis. Por ello, tienen un carácter **exploratorio.**Ver figura 1.

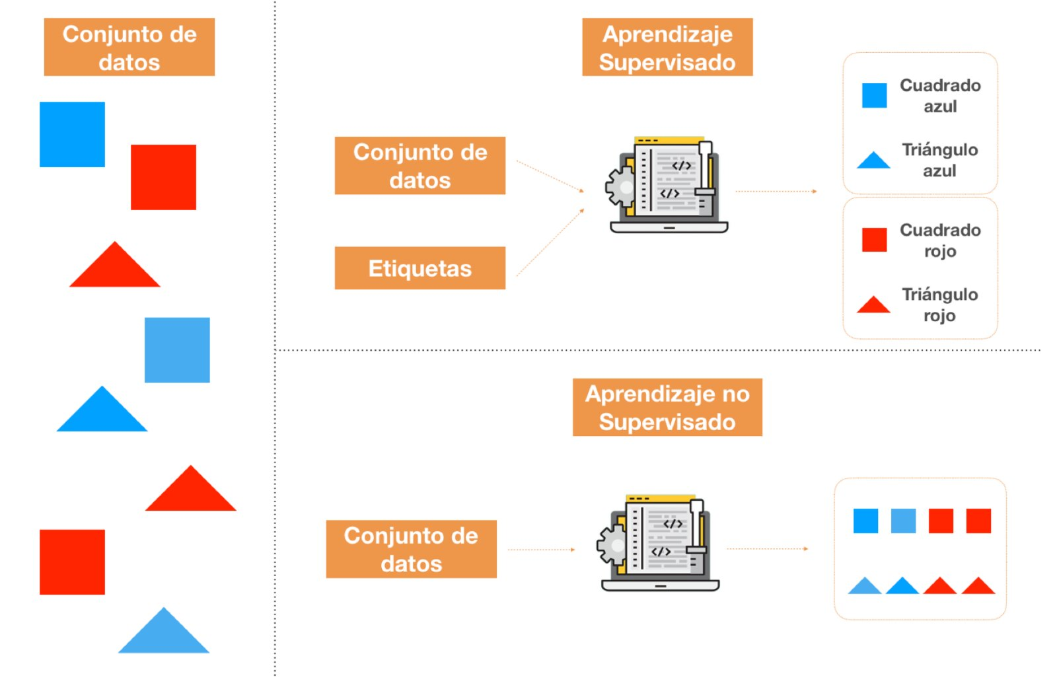
El aprendizaje no supervisado se suele usar en:

* Problemas de clustering.
* Agrupamientos de co-ocurrencias.
* Perfilado o profiling.

Si embargo, los problemas que implican tareas de encontrar similitud, predicción de enlaces o reducción de datos, pueden ser supervisados o no.

Los tipos de algoritmo más habituales en aprendizaje no supervisado son:

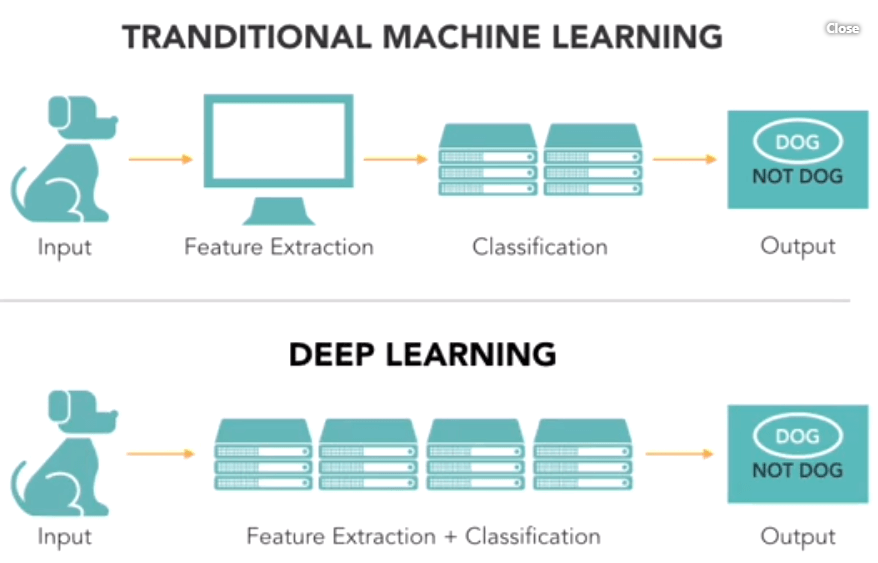
* Algoritmos de clustering.
* Análisis de componentes principales.
* Descomposición en valores singulares (singular valuedecomposition).
* Análisis de componentes principales (IndependentComponentAnalysis).[3]



**Figura 1.Aprendizaje supervisado y Aprendizaje no supervisado.**

## 2.3. DEEP LEARNING

El deeplearning es un tipo de machine learning que entrena a una computadora para que realice tareas como las hacemos los seres humanos, como el reconocimiento del habla, la identificación de imágenes o hacer predicciones. En lugar de organizar datos para que se ejecuten a través de ecuaciones predefinidas, el deeplearning configura parámetros básicos acerca de los datos y entrena a la computadora para que aprenda por cuenta propia reconociendo patrones mediante el uso de muchas capas de procesamiento.[4]



**Figura 2**. Machine learning vs Deep Learning.

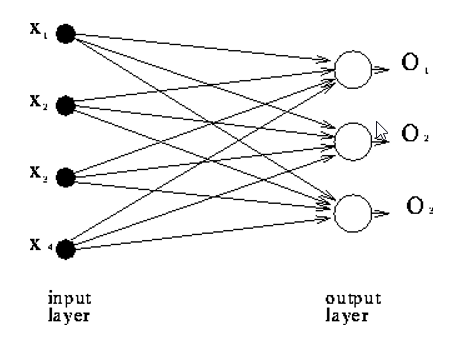
## 2.4. REDES NEURONALES

Una red neuronal consiste en un conjunto de neuronas separadas por capas y comunicadas entre ellas mediante conexiones. Cada neurona recibe un valor inicial, y esta lo transforma y reenvía a todas las neuronas con las que esté conectada en adelante. Estos valores iniciales pueden proceder o de la entrada inicial a la red, o de incluso otra neurona. Cada neurona cuenta con una función de activación y cada conexión con un peso. Las funciones de activación son funciones limitadoras, que o bien modifican el valor que emite la neurona o impone un mínimo para que este valor sea propagado. En cada conexión que comunica una neurona A con una neurona B, se multiplica el valor de salida de la neurona A por el valor del peso de la conexión, quedando como resultado que la entrada que recibe la neurona B es igual a la suma de todas las salidas multiplicadas por todos los pesos conectados con ella. Así, la entrada se va procesando hasta que alcanza la última capa o capa de salida, de la que se extraen los resultados del cómputo. La fortaleza de las redes neuronales reside en que los pesos de las conexiones pueden ajustarse según si los resultados obtenidos en lacapa de salida son correctos o no, en un proceso conocido como entrenamiento. De esta manera, si el resultado obtenido es acertado, se refuerza el comportamiento de la red ante ese tipo de entradas, mientras que, si el resultado es erróneo, se ajustan los pesos para tratar de “acercar” los futuros resultados a una solución satisfactoria. A cada uno de los pasos en el entrenamiento de una red neuronal se le conoce como “época” (o Epoch en inglés)

### 2.4.1.Redes Monocapa

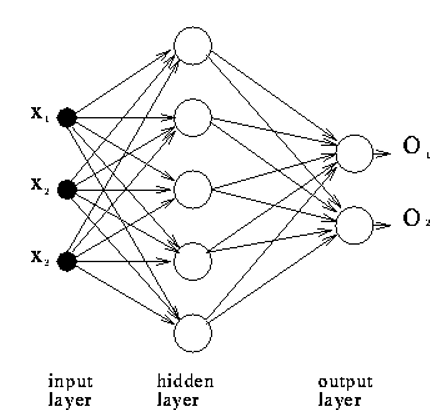
Esta es la arquitectura de red neuronal más simple. En ella, sólo existe una capa de neuronas que recibe las entradas y directamente las emite como una o varias salidas calculadas. Debido a su simpleza, este es un tipo de red neuronal muy adecuado para problemas simples de clasificación, con pocas entradas y pocas clases o etiquetas como salidas, ya que son también redes que obtienen resultados muy rápidos.

Incluso en su simpleza, contando con solo una capa se pueden alcanzar sistemas ligeramente más sofisticados, como una red de Hopfield, en la cual las neuronas tienen a su vez conexiones con neuronas de la misma capa. Este tipo de redes también se conoce como redes de neuronas recurrentes.



### 2.4.1.Redes Multicapa

Añadiendo capas de neuronas, se alcanza una mayor capacidad de cómputo y unos resultados más precisos. La desventaja de estos tipos de redes es que, debido a su mayor tamaño, también acarrean más necesidad de tiempo y recursos para llegar a la solución y ser entrenadas. La diferencia de estas redes con las redes monocapa es que estas incluyen una o más capas intermedias llamadas capas ocultas, que continúan el proceso de emitir resultados a otras neuronas.



## 3 CAPITULO III

## RECONOCIMIENTO DE VOZ

## 3.1. MACHINE LEARNIG CON PYTHON

La mayoría de la documentación encontrada relativa al diseño e implementación de redes de neuronas estaba orientada a Phyton. Todas las librerias de uso extendido en el sector, ya sean de manipulación de datos o de *MachineLearning,* son compatibles con este lenguaje, o direcatamente han sido desarrollados en el.

En adición a la sencillez del propio lenguaje y a su sintaxis simple, que facilita el despliegue y mantenimiento del código, desde un principio se valoro como la opción mas adecuada para el desarrollo del proyecto.

### 3.1.1.Scikit-Learn

Scikit-Learn es un paquete basado en SciPy, haciendo uso intenso de sus operaciones matemáticas. Proporciona una interfaz concisa y consistente para los algoritmos de *Machine Learning*mas comunes, facilitando su incorporación en sistemas en producción.

La librería combina un código de calidad y una buena documentación, así como una sencillez de uso y un alto rendimiento. Todo esto la ha convertido en la librería estándar para desarrollar modelos de *Machine Learning*en Python.

### 3.1.2. TensorFlow

Desarrollada inicialmente por Google, desde la liberación de su código ha ganado popularidad rápidamente, eclipsando en gran medida a las otras alternativas.

Se basa en grafos de computación de flujo de datos, y destaca por su sistema de nodos multicapa que permite un rápido entrenamiento de redes de neuronas en grandes *dataset.* Además, se ha usado para una variedad de aplicaciones del mundo real, como los servicios de Google para reconocimiento de vos (Google VoiceRecognition) y de identificación de objetos en imágenes (Google ImageRecognition).

### 3.1.3. Keras

Keras es una libreria que permite construir redes de neuronas a alto nivel, de forma sencilla y minimalista. Esta escrita en Python y permite trabajar sobre Theano, TensorFlow y Microsoft Toolking (CNTK).

Keras es altamente modular y escalable, además de permitir el prototipado rápido de redes de neuronas por su sencillez. Los datos se preparan en tensores y sus modelos se basan en capas, con una capa para los tensores de entrada, otra para los de salida y un numero de indeterminado de capas ocultas.

### 3.1.4. PyTorch

PyTorch es un *framework*de *Deep Learning*para una experimentación rápida y flexible. Permite la computación con tensores con una potente aceleración por GPU.

Usualmente, se utiliza como sustituida a NumPy para aprovechar la técnica de computo de las GPUs, proporcionando integración con librerias de aceleración de GPU como Intel MKL y NVIDIA CuDNN.

### 3.1.5. Theano

Theano es una librería que define arrays multidimensionales de una forma similar a NumPy, así como operaciones y expresiones matemáticas. Se proporciona compilada, de forma que puede ejecutarse de forma eficiente en todas las arquitecturas.

Destaca por su estrecha integración con NumPy a bajo nivel para sus operaciones, así como sus optimización de uso de GPU y CPU, incrementando el rendimiento para la computación intensiva de datos. Además, sus ajustes de eficiencia y estabilidad permiten obtener resultados muchsmas precisos, incluso con valores muy pequeños.

## 3.2. ANTECEDENTES - HISTORIA RECONOCIMIENTO DE VOZ

El reconocimiento de voz tal y como lo conocemos empezo en 1952, ese año Bell Labs desarrollo un sistema capaz de identiﬁcar los digitos de forma automatica, a este sistema lo denominaron.

Mas adelante, los años 60, aparecieron numerosos algoritmos y tecnicas. Por un lado, los sovieticos inventaron el algoritmo de “Dynamic Time Warping”(DTW) capaz de identiﬁcar 200 palabras segmentando el audio en fragmentos de 10ms. En paralelo, Leonard Baun desarrollo las matematicas de las cadenas de markov, siendo la base de los modelos probabilisticos desarrollados por primera vez por James & Janet Bakerpoco tiempo despues.

A ﬁnales de los 70, principios de los 80, IBM mejoro las tecnicas existentes presentando “Tangora”, un sistema de reconocimiento de voz basado en HMM que permitia reconocer 20.000 palabras, desplazando a las tecnicas existentes que empleaban DTW; este sistema contaba con un elaborado modelo de lenguaje que increment´o notablemente la precision de los algoritmos y tecnicas de la epoca.

A ﬁnales de los años 80, Slava M. Katz desarrollo los modelos de lenguaje N-gram, que modelan la probabilidad condicionada de una palabra basandose en N-1 palabras de la secuencia; estos modelos de lenguaje mejoraron en gran medida los sistemas de reconocimiento de voz existentes hasta la fecha.

A principios de los años 90, aparecio en el mercado Dragon Dictation, siendo una de las primeras empresas en comercializar a nivel general un sistema de reconocimiento de voz, a mediados de los 90, Dragon Dictation fue comprada por Lernout&Hauspie, dando pie al desarrollo posterior de uno de los algoritmos de reconocimientos mas extendidos de la historia, el algoritmo de reconocimiento de voz integrado con Windows XP que, debido a su masiﬁcacion internacional, convirtio a Lernout&Hauspie en los lideres de la industria en tecnologias de reconocimiento de voz.

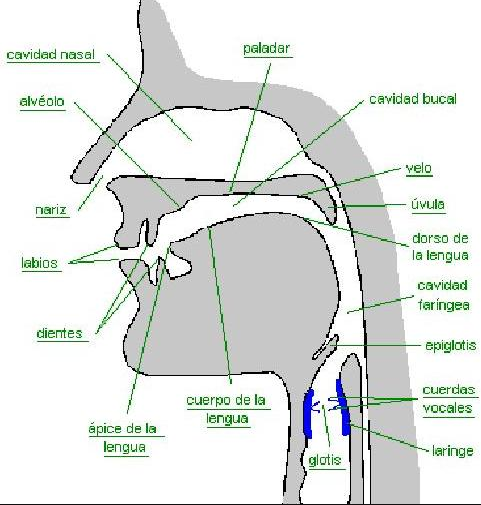
Mas adelante, Lernout&Hauspiedesaparecio debido a un escandalo contable en 2001, vendiendo su tecnologıa de reconocimiento de voz a ScanSoft, quienes pasaron a desarrollar y mantenerel algoritmo de reconocimiento de voz hasta su cambio de nombre en 2005 por Nuance.

En 2005, ScanSoft, ahora Nuance, se alio con Apple para ofrecer las tecnologias de reconocimiento de voz en los productos de la empresa de la manzana, fruto de esta union comercial, Apple obtuvo la licencia del software para proporcionar la capacidad de reconocimiento de voz a su asistente de voz virtual Siri.

En 2007, Google empezo a contratar investigadores de Nuance,la ﬁnalidad de ello fue desarrollar un servicio de directorio telefonico que permitio a Google recopilar cientos de horas de voz que posteriormente emplearia para desarrollar “Google VoiceSearch”, con su propio sistema de reconocimiento de voz; posteriormente, Google integro su tecnologia en la mayoria de sus productos: Maps, Earth, Google Translateapp y muchos mas.

## 3.3.MECANISMO VOCAL

## La voz consiste en el sonido producido por un ser humano haciendo uso de sus cuerdas vocales para hablar, cantar, reírse, gritar, chillar, etc. Su frecuencia oscila entre alrededor de 60 a 7000 Hz.



La voz comienza con la inspiración (toma de aire), el aire llena los pulmones y luego sale a través de la espiración. Cuando la espiración ocurre, el aire pasa de los pulmones a la tráquea y de ésta a la laringe y ahí, pasa por en medio de las cuerdas vocales; dos músculos pequeños que se tensan para producir sonidos agudos y se relajan para los sonidos graves. De este modo la voz adquiere su tono.

El flujo de aire continúa su camino hacia la boca, en donde la lengua, mandíbula, paladar y labios se encargan de darle forma al sonido y se producen los fonemas (vocales y consonantes). Ahí es donde la voz adquiere su forma, lo que llamamos dicción o articulación.

Al mismo tiempo el sonido producido por la laringe y la boca adquiere su timbre característico al resonar dentro de las cajas de resonancia ubicados en los huesos, pecho, laringe, boca, nariz y cráneo. A través de este mecanismo, la voz también se amplifica naturalmente.

## 3.4. PROCESAMIENTO DE AUDIO CON PYTHON

El sonido se representa en forma de una señal de audio que tiene caracteristicas como frecuencia, ancho de banda, decibeles, función de la amplitud , tiempo , etc.

Los sonidos están disponibles en muchos formatos digitales lo que permite que sean leidos y analizados porla computadora. Ejemplo de formatos: mp3 ,wma , wav.En el presente documento analizaremos archivos en formato wav.

Python tiene excelentes librerias para el procesamiento de audio como Librosa, PyAudio.

A continuación se detallan las utilizadas en el presente proyecto.

### 3.4.1. Librerias de Audio

### 3.4.1.1.Librosa

Es un módulo de Python para analizar señales de audio en general,

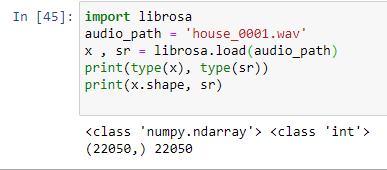
pipinstalllibrosa

Para aumentar más potencia de decodificación/analisis de audio, también se puede instalar *ffmpeg*[[1]](#footnote-2) que incluye muchos decodificadores de audio.

### 3.4.1.2. IPython.display.Audio

IPython.display.Audio le permite reproducir audio directamente en un jupyter notebook.

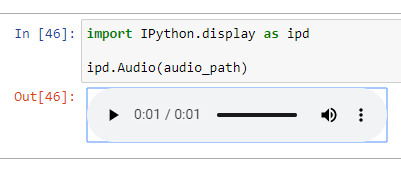
### 3.4.2. Cargando un archivo de audio con Python



Las anteriores instrucciones devuelven una serie temporal de audio como una matriz numpy con una frecuencia de muestreo.

### 3.4.3. Reproduciendo audio

Se usa IPython.display.Audio para reproducir audio.

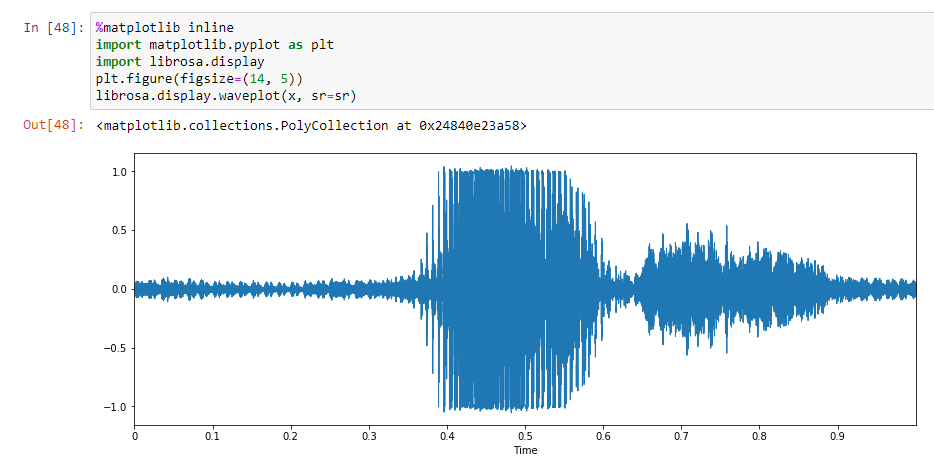


La instrucción ipd.Audio(audio\_path) devuelve un widget de audio en el notebookjupytercomo se vio en la imagen anterior.

### 3.4.4. Vizualizando Audio

### 3.4.4.1.Waveform/ Ondas

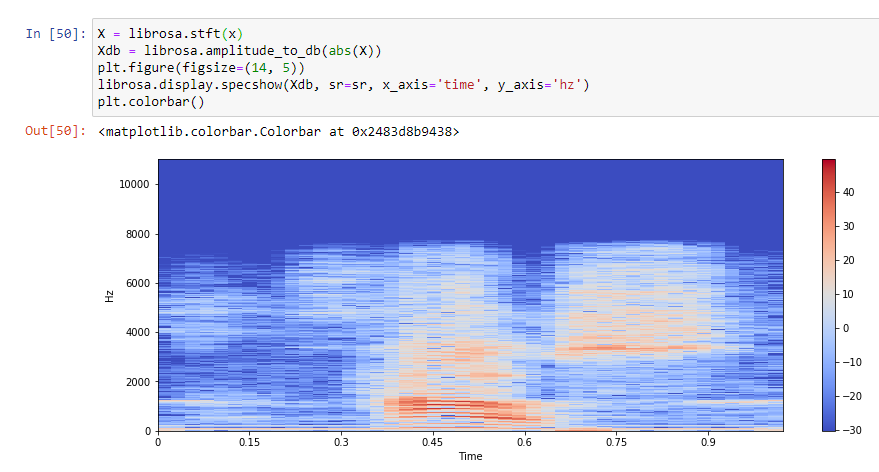
Se puede trazar la matriz de audio usando librosa.display.waveplot



### 3.4.4.2.Espectograma

Un espectrograma es una representación visual del espectro de frecuencias de sonido que varían con el tiempo. Un espectograma se utiliza para el análisis de la señales eléctricas, de comunicaciones, y cualquier señal audiovisual en su contenido frecuencial. Es una representación en tres dimensiones, temporal, frecuencial y amplitud de la distribución de energía de una señal en este caso del sonido. Los espectrogramas a veces se llaman ecografías, o diagramas de voz.

Se puede mostrar un espectrograma usando. librosa.display.specshow.



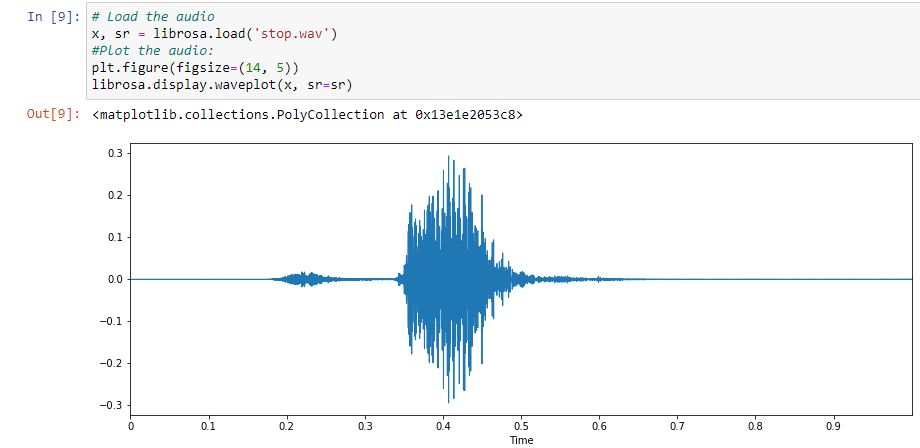
El eje vertical muestra frecuencias (de 0 a 10 kHz), y el eje horizontal muestra el tiempo del clip.

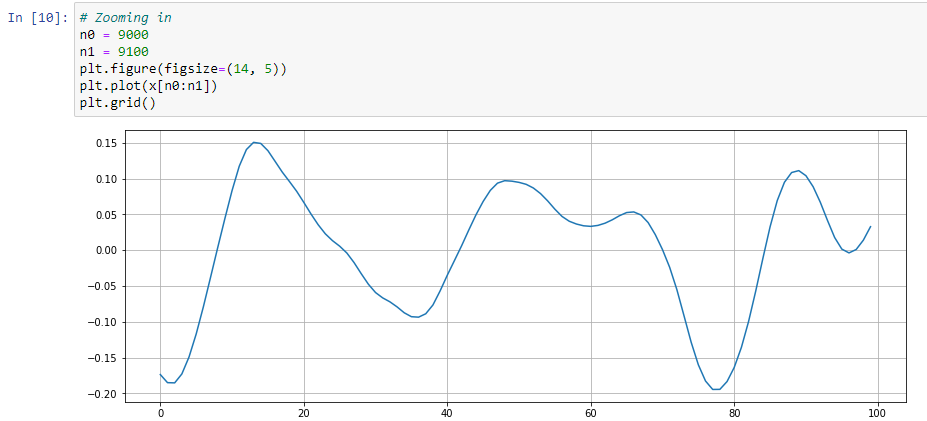
### 3.4.5. Extraccion de Caracteristicas

Cada señal de audio consta de muchas características o parametrosSe debe extraer las características que sean relevantes al objetivo del proyecto.A continuación se mencionan algunas de ellas.

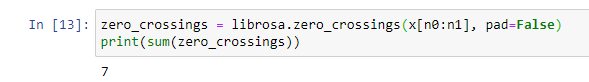
### 3.4.5.1. Zero Crossing Rate

El “zero crossign rate” es la tasa de cambios de signos a lo largo de una señal de audio, es decir, la tasa a la que la señal cambia de positivo a negativo o de regreso. Este este atributo es unos de los comunente utilizado para el reconocimiento de voz pero también en muchos casos es utilizado en la recuperación de información musical. En las imagenes siguientes se puede ver el resultado aplicado al clip de audio *stop.wav*



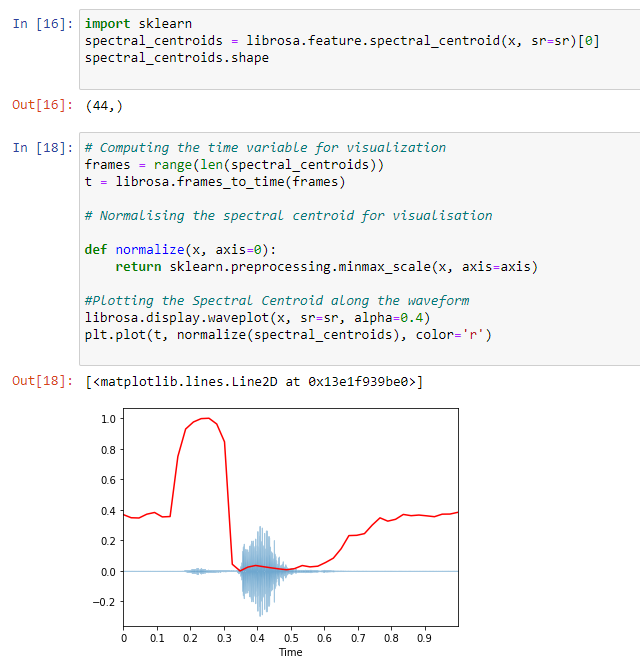


Tenemos 7 zero crossing para el audio ejemplo



### 3.4.5.2. Spectral Centroid

Indica dónde se encuentra el centro del espectro para un sonido y se calcula como la media ponderada de las frecuencias presentes en el sonido.



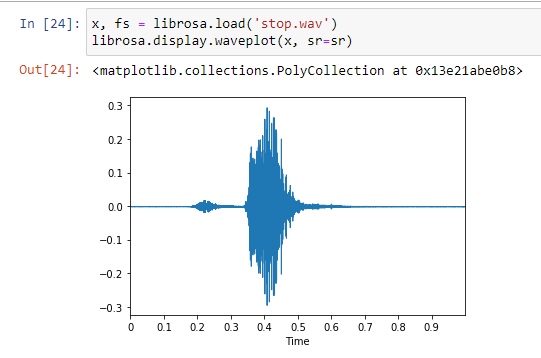
### 3.4.5.3. Spectral Rolloff

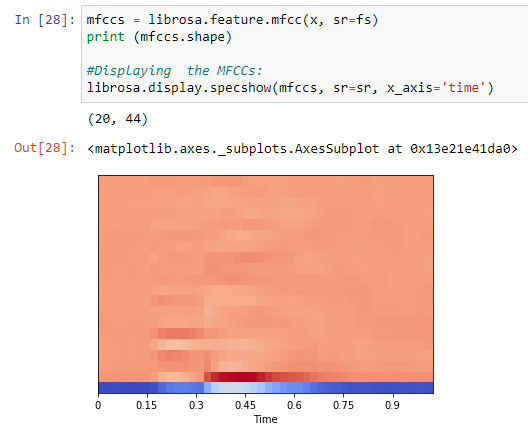
Es una medida de la forma de la señal. Representa la frecuencia por debajo de la cual un porcentaje especificado de la energía espectral total.

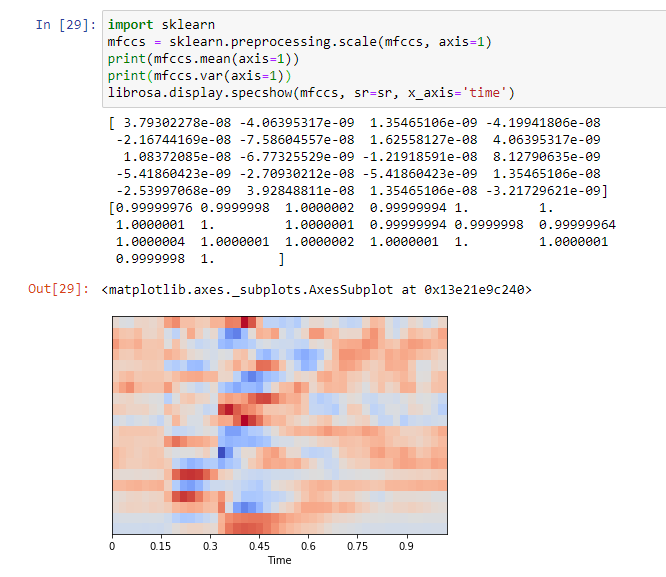
### 

### 3.4.5.4. Mel-Frequency Ceptral Coefficients

Mas conocidos como los MFCCs ,los coeficientes cepstrales de frecuencia Mel de una señal son un pequeño conjunto de características alrededor de 10 a 20 que describen de manera concisa la forma general de una envoltura espectral. Modela las características de la voz humana.

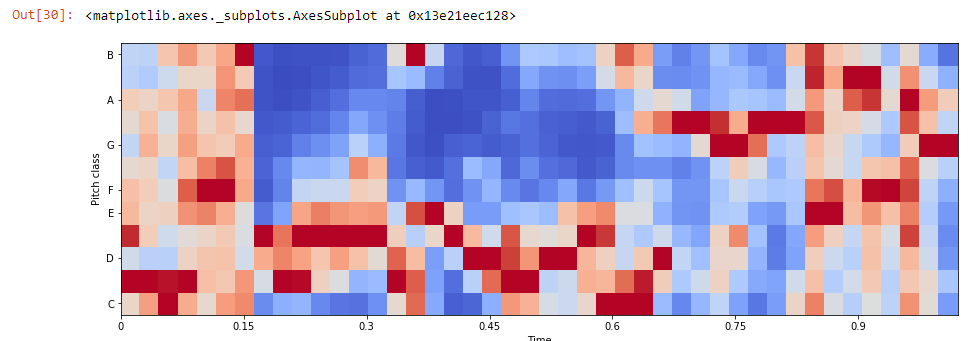






### 3.4.5.5. Chroma Frequencies

Los atributos de croma son una representación interesante para el audio especialmente para la música en la que todo el espectro se proyecta en 12 contenedores que representan los 12 semitonos (o croma) distintos de la octava musical.



## 4 CAPITULO IV

## RECONOCIMINETO DE COMANDOS DE VOZ

## 4.1. INTRODUCCION



Luego de tener una visión general sobre las señales acústicas y el proceso de extracción de características , en esta sección se utilizara los conocimientos adquiridos poder resolver el problema de reconocimiento de voz/comandos con machine learning.

La propuesta es desarrollar un modelo de machine learning usando redes neuronales, antes de que el dataset sea procesado por la red neuronal, este debe pasar por una fase pre-procesado, dado que la red no puede trabajar con los archivos de audio como entradas.

Al momento de preprocesar los archivos de audio se pueden optar por dos enfoques

* Obtener los espectogramas(ya descritos anteriormente) y diseñar una red que trabaje con imagenes como entradas.
* Obtener los caracteristicas y tratarlos como vectores de numeros y usar esos como entradas de la red neuronal.

En el presente proyecto se opto por el segundo enfoque.El procesado de audio requiere una serie de operaciones a realizar de forma secuencial en cada uno de los archivos de audio que permita extraer los atributos o caracteristicasmas relevantes que en su conjunto compondran el dataset.

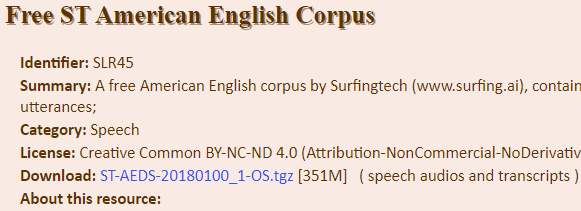
## 4.2. DATASET

Durante el diseño de un modelo de Machine Learning es fundamental elegir correctamente el dataset a utilizar ya que esto infuye directamente en los resultados obtenidos.

En el caso de procesado de lenguaje natural NLP los dataset son llamados corpus por la reclacion que tienen con la linguistica.

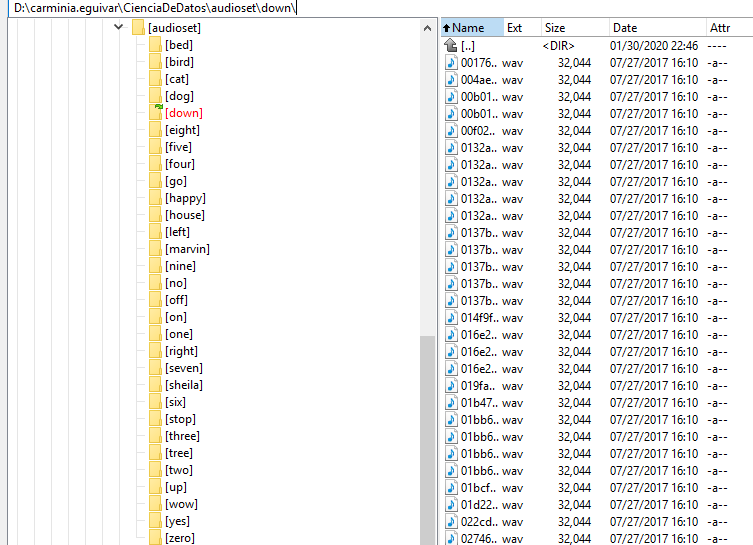
En funcion del objetivo del proyecto el tipo de dataset a escoger ha sido diferente de los normalmente trabajados, en este caso tenemos un conjunto de archivos de audio/corpus en idioma ingles en formatto WAV

El audioset ha sido descargado de la pagina *Open Speach and Language Resources[[2]](#footnote-3)*  el identificador del audioset es el SLR45[[3]](#footnote-4)



El audioset seleccionado contiene 30 carpetas, cada una de ellas representa una palabra y dentro se encuentran los archivos de audios que corresponden a dicha palabra.

Cada carpeta en promedio contiene 2300 archivos de audio por lo que se tendrá un total de 69000 archivos en promedio para trabajar.



## 4.3. PRE-PROCESAMIENTO DE ARCHIVOS DE AUDIO

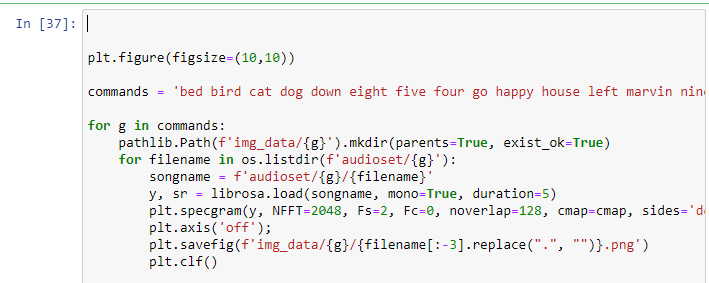
Antes de entrenar el modelo de clasificación,se transformara las muestras de audio en representaciones más significativas/relevantes y de interes para objetivo planteado. Los archivos de audio necesitan un procesamiento previo para que puedan ser usadas como entradas de la red neuronal. Se realizara este pre-procesamiento con librerías python para leer archivos de audio.(Librerias ya mencionadas en detalle enel capitulo anterior)

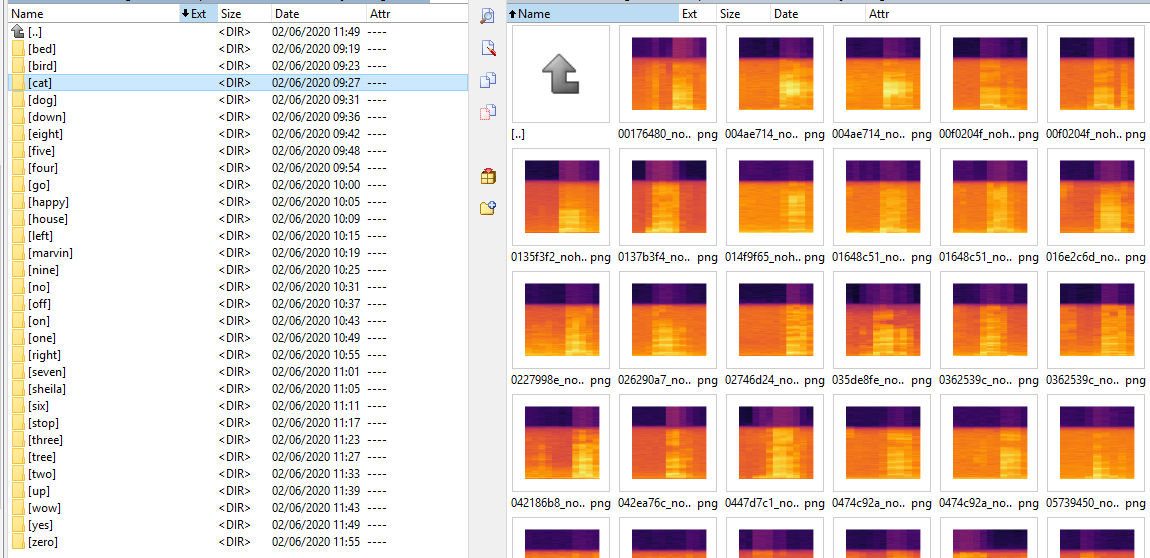
Se realizara la extracion de los atributos massiginificativos de los archivos de audio. Se han elegido 5 atributos: coeficientes cepstrales de frecuencia de mel, centroide espectral, velocidad de cruce cero, frecuencias de croma, roll-off espectral.

### 4.3.1. Extraccion de espectograma por cada audio

Como ya se había encionado en el capitulo anterior un espectograma es la representación visual acerca de qué frecuencias integran un sonido y cuáles son las respectivas amplitudes y fases, constituye lo que se denomina espectro del sonido.

En resumen el espectograma extraido por cada audio es una imagen con extensión .png que será guardada en disco.Se crearan nuevas carpetas replicando la misma estructura del audioset para guardar las imágenes/espectogramas.

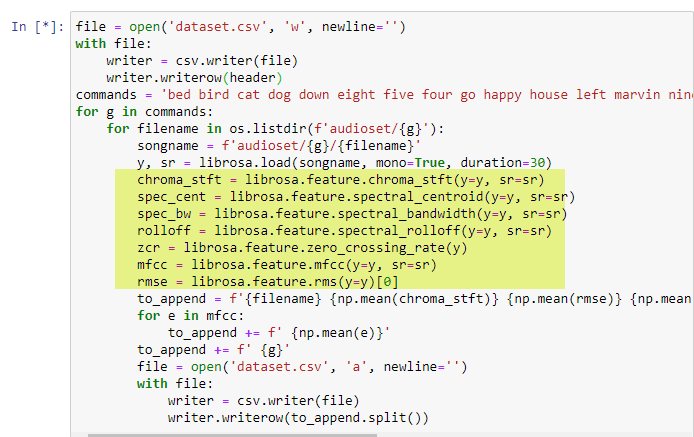




### 4.3.1. Extraccion de Caracteristicas desde los espectogramas

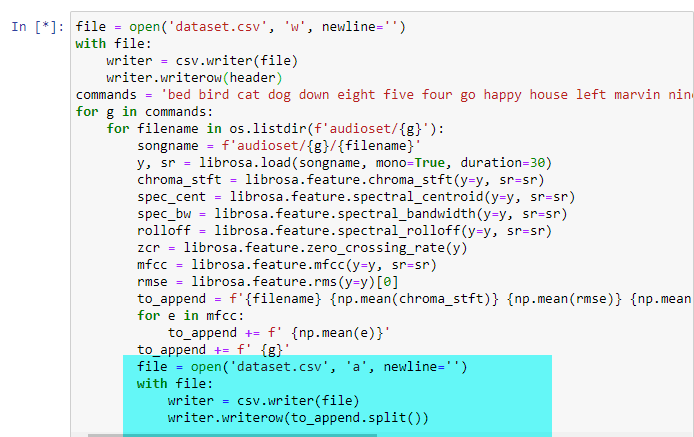
Los atributos a extraer por cada expactograma(imagen) serán los siguientes:

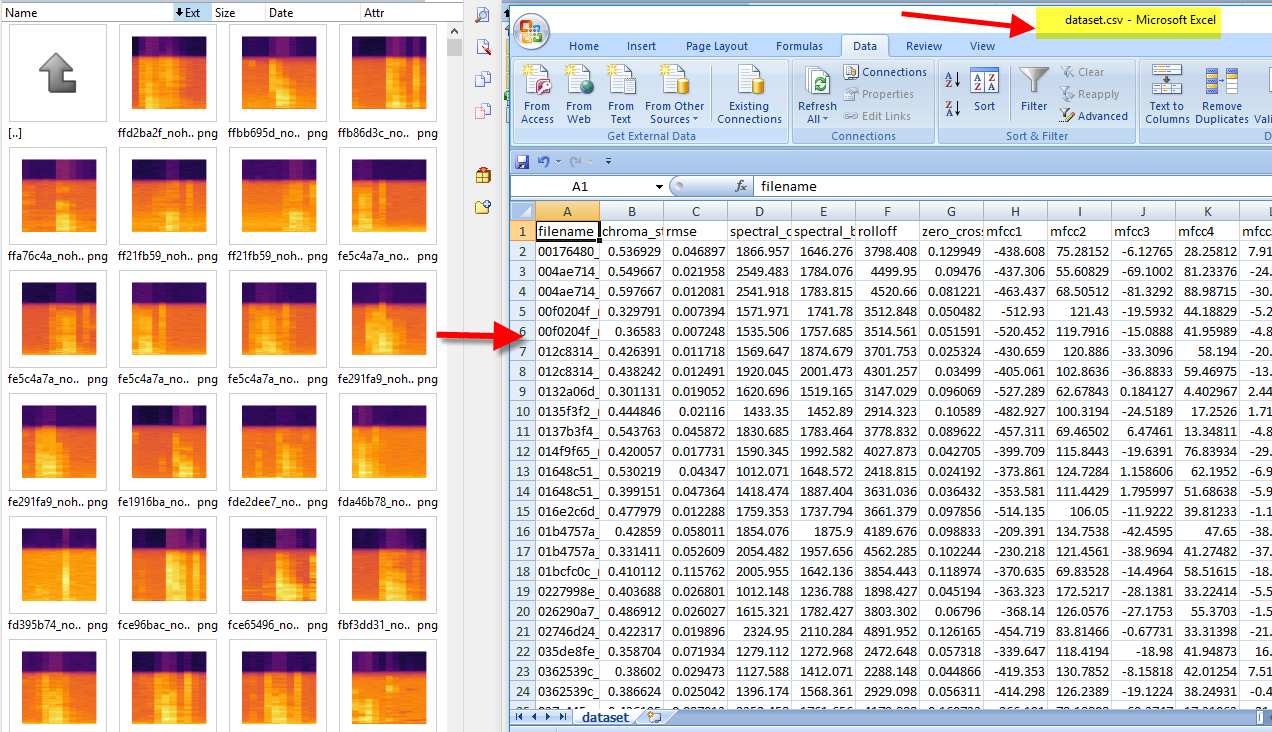
* Coeficientes cepstrales de frecuencia de mel (MFCC) (en un número de 20)
* Centroide espectral,
* Tasa de cruce cero
* Frecuencias de croma
* Desplazamiento espectral.

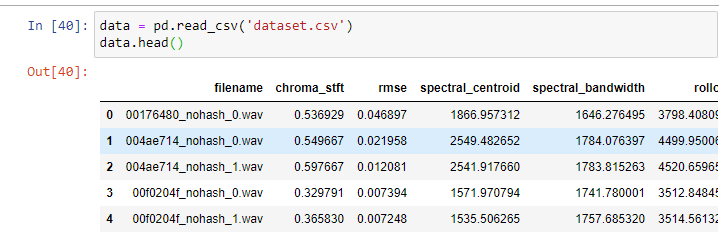


## 4.4. ENSAMBLADO DEL DATASET

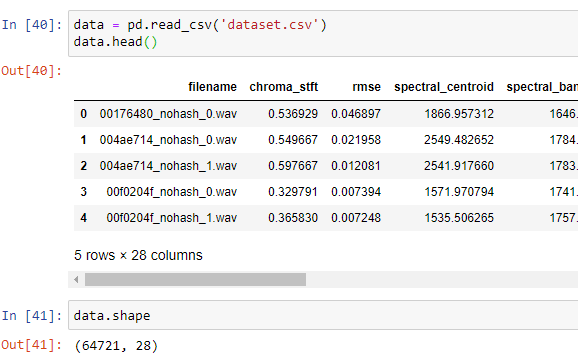
Una vez realizadas la extracción de atributos del archivo de audio , estas características se añaden a un archivo .csv y el conjunto de todos los archivos de audio procesados componen el dataset que luego será utilizado por algoritmo de clasificación en este caso la red neuronal.

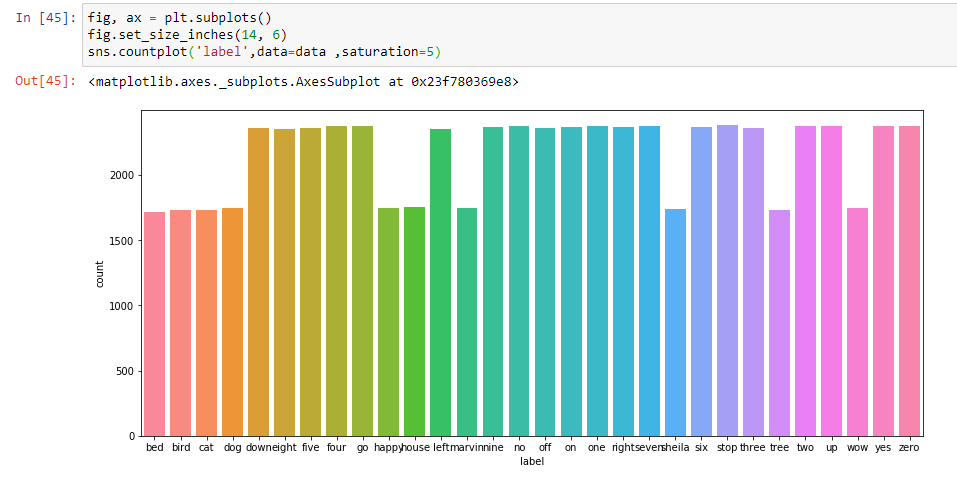






## 4.5. ANALIZANDO LOS DATOS EN PANDAS



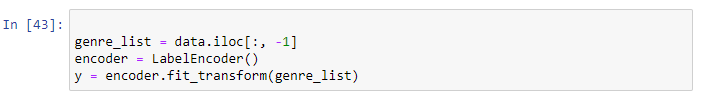


## 4.5.1. Limpieza de datos

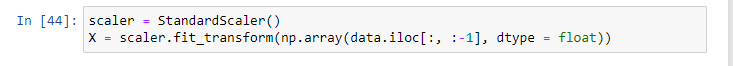
* Eliminamos columnas innecesarias



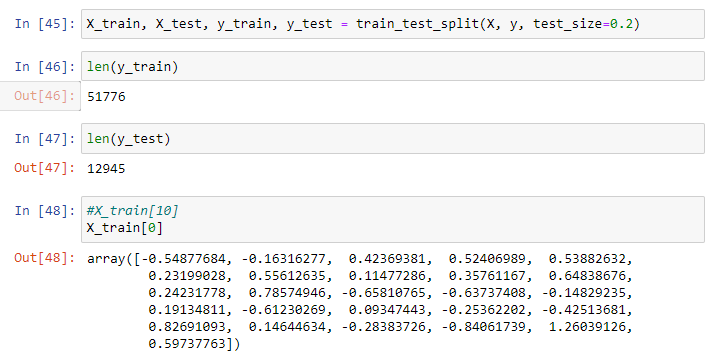
* Codificacion de las etiquetas



* Escalando las columnas



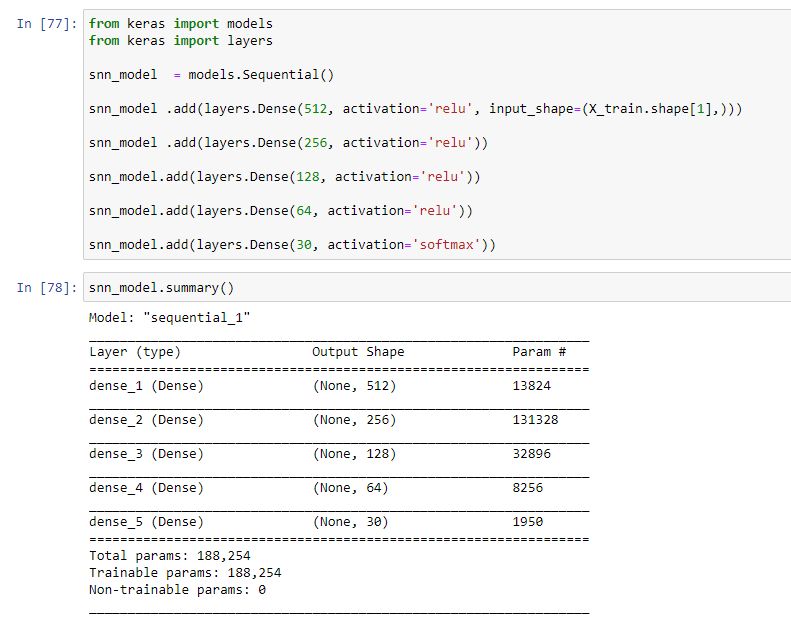
## 4.6. DIVIDIENDO EL DATASET EN TRAINING[[4]](#footnote-5) AND TEST[[5]](#footnote-6) DATA



## 4.7. CLASIFICACION CON KERAS- CONSTRUCCION DE LA REDE NEURONAL

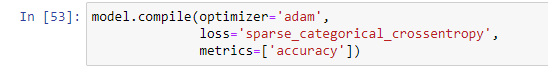
Una vez que se han extraído las características y se tiene ensamblado/definido el dataset, se procede a usar los algoritmos de clasificación existentes para reconocimiento de voz en el presente proyecto se usa redes neuronales.Como anteriormente se ha mencionado, de acuerdo a las investigaciones realizadas se pueden usar las imágenes del espectrograma directamente para la clasificación o puede extraer las características y usar los modelos de clasificación en ellas. Se ha elegido usar la extracción de características y utilizar esas como entradas de la red neuronal.

Se crea el modelo tipo secuencial y las capas de neuronas secuenciales.



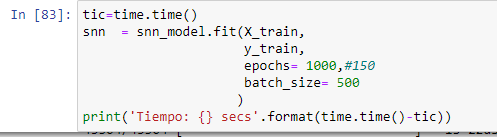
### 4.7.1. Ajustes al modelo

Con esto indicamos el tipo de pérdida (loss) que se utilizara ,el optimizador de los pesos de las conexiones de las neuronas y las métricas que queremos obtener



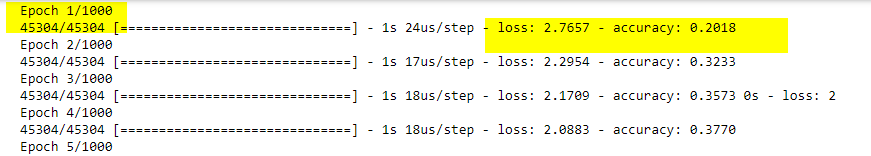
### 4.8. ENTRENAMIENTO DE LA RED

Indicamos con model.fit() las entradas y sus salidas y la cantidad de iteraciones de aprendizaje (epochs) de entrenamiento.

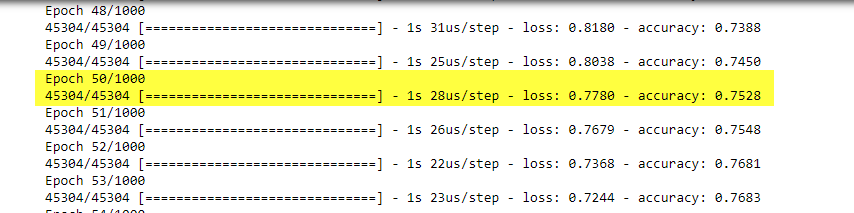


### 4.8.1. Interpretacion de los resultados del entrenamiento

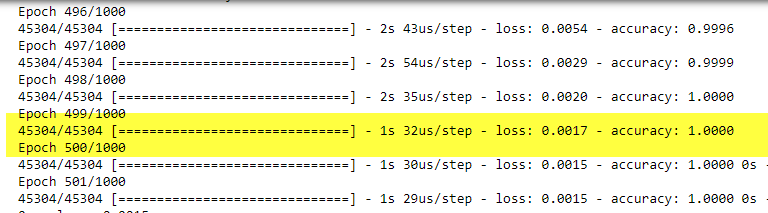
Si vemos las salidas del entrenamiento, vemos que la primer iteración tuvo un acierto de 0.2 , es decir sólo acierta 1 de cada 5 .



Luego en la epoch[[6]](#footnote-7) 50 recupera el 0.7 de acierto ,es decir 7 de cada 10.

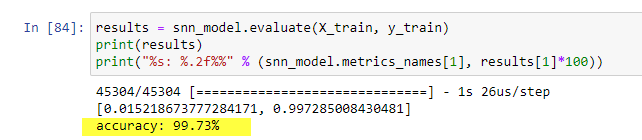


En el epoch 500 se verifica que la iteración tuvo un acierto de 1.00



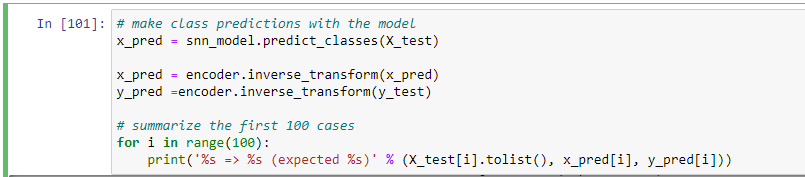
### 4.8.2. Evaluacion

Luego del etrenamiento podemos ver que el modelo tuvo una precisión de 99.73% .



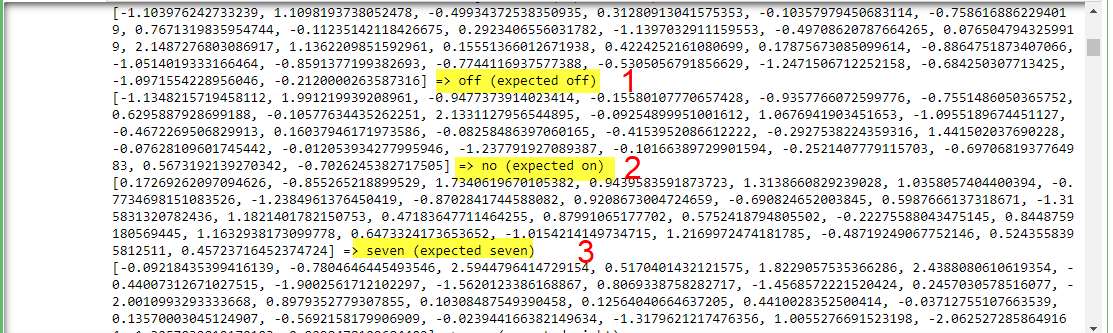
### 4.9. PREDICCIONES

Se utiliza el dataset de test para las predicciones, y mostramos el resultado de los 100 primeros casos.



### 4.8.2. Analisis de Resultados

Si vemos las parte resaltadas en amarillo observamos los resultados de las predicciones :

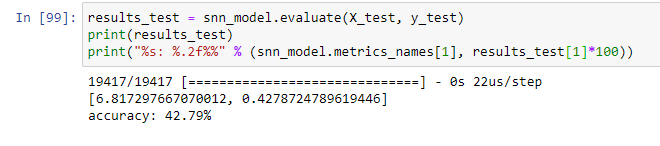


1 => off (expected off) : El resultado esperado era la palabra *off*  el predecido fue *off*  lo cual es correcto.

2 => no (expected on) : El resultado esperado era la palabra *on* y el predecidon fue *no* est preddiccion fue incorrecta.

3 => seven (expected seven) : El resultado esperado era la palabra *seven*  el predecido fue *seven*  lo cual es correcto.

Si evaluamos el modelo con los datos de test obtenemos que la precisión es de un 42.79 , lo cual explica el por que de nuestras predicciones incorrectas.Se puedeir mejorando el modelo para subir esta precisión.



## 5 CONCLUSIONES Y RECOMENDACIONES

* Haciendo uso del reconocimiento de voz se puede reducir la brecha que existe en la comunicación entre una persona y la maquina.
* A pesar que en la evaluacion para en entrenamiento se optuvo una precision del 99.9% se pudo observar que en la etapa de test la precision solo era del 42% Por tanto aun no se ha explotado al maximo el potencial de las redes neuronales, por lo que se espera mejorar esta precision en iteraciones futuras.
* Contrario a la inicialmente previsto,la fase mas compleja del presente proyecto no ha sido el diseño y entrenamiento de la red neuronal sino la elecion y el pre-procesdo de audio por lo que ha sido necesaio incluir este proceso como parte de los objetivos y alcances del proyecto. Para cumplir esta etapa satisfactoriamente la fase de investigacion ha sido extensa en cuanto a la tecnologia, librerias, audioset, etc.

## 6 BIBLIOGRAFIA

<https://scipy2015.scipy.org/>

<https://www.ffmpeg.org/about.html>

<https://docs.google.com/presentation/d/1zzgNu_HbKL2iPkHS8-qhtDV20QfWt9lC3ZwPVZo8Rw0/pub?start=false&loop=false&delayms=3000&slide=id.p>

Dataset Audio:

<http://www.openslr.org/45/>

<https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales>

<https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>

PREDICTIONES

<https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>

recpnocimiento facial

<https://towardsdatascience.com/from-raw-images-to-real-time-predictions-with-deep-learning-ddbbda1be0e4>

1. FFmpeg es frameworkcapaz de decodificar, codificar, transcodificar, filtrar y reproducir un sinfín de formatos. [↑](#footnote-ref-2)
2. Open Speech and Languages Resources o mas conocido como OpenSLR es una pagina web que contiene un sin fín de audiosets para libre descarga. [↑](#footnote-ref-3)
3. Dentro la pagina OpenSLR cada audio set tiene un identificador SLR45 es el que corresponde al utilizado en el presente proyecto. [↑](#footnote-ref-4)
4. Training Data: En un aprendizaje supervisado, utiliza un conjunto de datos de entrenamiento, que contiene resultados, para entrenar la máquina. [↑](#footnote-ref-5)
5. Test Data: Lueego del entrenamiento se utiliza un conjunto de datos de prueba para predecir los resultados. [↑](#footnote-ref-6)
6. Este es el numero de veces que se ejecutaran los algoritmos de forwardpropagation y backpropagation. En cada ciclo (epoch) todos los datos de entrenamiento pasan por la red neuronal para que esta aprenda sobre ellos [↑](#footnote-ref-7)