UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

PROGRAMA DE FORMACIÓN DUAL

FACULTAD DE INGENIERÍA

REPORTE DE PROYECTO FINAL.

CLASIFICADOR DE COMANDOS DE VOZ.

MENTORES:

Gastelum Ossiel

Continental Automotive - Querétaro Campus

APRENDICES:

Espinosa Bernal Giovanni

Fuentes Flores Lorena

Martínez Olvera Judith

Ugalde Romero Dulce Carolina

Vite González Cynthia

Santiago de Querétaro, Qro., 13 de Diciembre del 2019.

ÍNDICE

I. INTRODUCCIÓN

El proceso de planeación para el desarrollo de un proyecto consiste de etapas fundamentales antes de su ejecución, si bien no se obtienen los resultados esperados en la primera experimentación estos nos aportan información relevante sobre los puntos débiles en la investigación, a lo largo de este reporte se describen los pasos que se siguieron para desarrollar el proyecto de clasificación de comandos de vos en una red neuronal y los resultados que se obtuvieron ante la falta de cuidado en la selección de las características de los datos.

II. OBJETIVOS

Objetivo específico:

Desarrollar un clasificador de comando de voz haciendo uso de los algoritmos de Machine Learning.

Objetivos Particulares:

- Generar una base de datos propia.
- Aplicar los conocimientos de Data Science.
- Clasificar 10 comandos distintos.
- El funcionamiento no dependerá del timbre de voz del usuario.
- Se podrá probar el clasificador mediante una entrada directa de voz.
- El sistema tendrá algún tipo de respuesta en modo texto dependiendo de la entrada obtenida.

III. MARCO TEÓRICO

Aprendizaje automático (Machine Learning)

El *machine learning* es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con mínima intervención.

Redes neuronales

Inspirándose en el comportamiento conocido del cerebro humano (principalmente el

referido a las neuronas y sus conexiones), trata de crear modelos artificiales que solucionen problemas difíciles de resolver mediante técnicas algorítmicas convencionales.

Ciencia de los datos (*Data Science*)

El *Data Scientist* debe explorar y analizar datos de múltiples fuentes, a menudo inmensas (conocidas como Big Data), y que pueden tener formatos muy diferentes. Además, debe tener una fuerte visión de negocio para ser capaz de extraer y transmitir recomendaciones a los responsables de negocio de su empresa.

Transformada de Fourier

El análisis de Fourier es un método para expresar una función como una suma de componentes periódicos, y para recuperar la función de estos componentes. Cuando la función y su transformada de Fourier se reemplazan por las contrapartes discretizadas, se denomina transformada de Fourier discreta (DTF) la cual separa la entrada en componentes que contribuyen a frecuencias discretas, tiene una gran cantidad de aplicaciones en el procesamiento de señales digitales, la entrada discreta a la transformación se conoce como señal, que existe en el dominio del tiempo. La salida se llama espectro o transformación y existe en el dominio de la frecuencia.

Correlación para procesamiento de señales

En el procesado digital de señales se necesita cuantificar el grado de interdependencia entre dos procesos o la similitud entre dos señales $x_1[n]$ y $x_2[n]$. Una medida de la correlación existente entre ambas señales puede efectuarse mediante la suma de los productos de los correspondientes pares de puntos mediante la expresión conocida como correlación cruzada. Un valor cero en la correlación significa que no existe correlación y por lo tanto las dos señales son completamente independientes.

Diagrama de Dispersión y Correlación

Este tipo de diagramas son una herramienta grafica que nos permite mostrar y demostrar la relación existente entre dos tipos de datos y cuantificar de manera grafica la intensidad de la relación existente. Su principal uso es conocer si existe una correlación entre dos magnitudes.

Regresión lineal

Este modelo permite hallar el valor esperado de una variable cuando se establece una relación con otra variable distinta. El análisis de regresión determina la intensidad entre las variables a través de coeficientes de correlación y determinación.

Coeficiente de correlación lineal

Es una medida de asociación entre dos variables aleatoria, este valor varía entre -1 y 1.

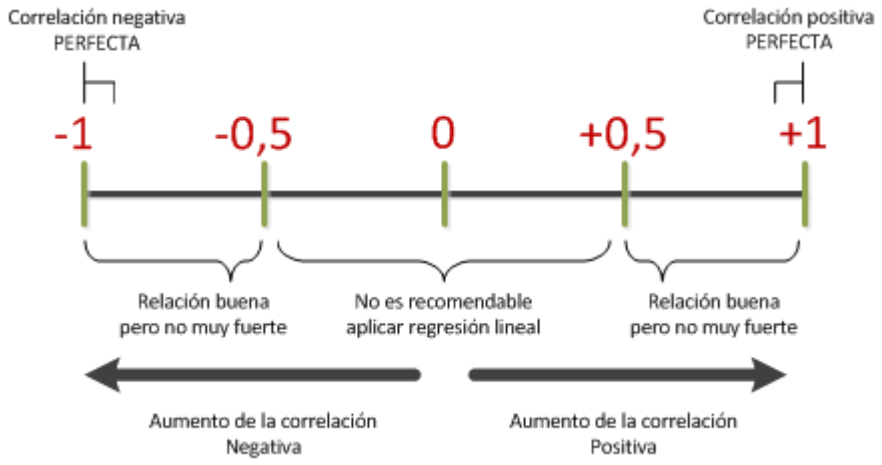


Ilustración 1. Coeficiente de correlación

IV. MATERIALES Y EQUIPO

- Micrófono
- *Workspace* para desarrolladores de *Python.*

V. METODOLOGÍA

Adquisición de audio

Para realizar las grabaciones de cada archivo de audio .wav, se desarrolló el programa "PyAudio_BD.py" el incluye las librerías "pyAudio" y "wave". Este programa permite grabar una señal de audio de tipo mono estéreo de 3 segundos, la grabación se logra por medio de un micrófono externo.

Se solicitó la ayuda a 18 personas para la grabación de todos los audios, cada persona realizó cinco grabaciones por cada comando; los comandos se listan a continuación:

1. Subir ventana.
2. Bajar ventana.
3. Como Rosalía.
4. Una flor para otra flor.
5. Nadaremos.
6. Modo fiesta.
7. Encender auto.
8. Café por favor.
9. Llévame a casa.
10. Hola kit.
11. Turbo.

Resultando 55 grabaciones por persona. Siendo un total de 990 grabaciones en nuestra base de datos.

Cada audio se nombró "C#_P#_G#" siendo "C#" el número de comando, "P#" número de persona y "G#" número de grabación. Para facilitar el nombramiento de los audios se generó automáticamente el nombre de cada grabación para posteriormente almacenarse.

Caracterización de las señales de audio

Se calculó la Transformada Rápida de Fourier (FFT, *Fast Fourier Transform*) en el programa "SignalAnalisisAndDB.py", esto debido a que la FFT permite hacer un análisis

en el dominio de la frecuencia para señales que se encuentran en el dominio del tiempo. Para facilitar el cálculo de FFT se empleó la librería “scipy.fftpack”.

La correlación para el procesamiento de señales nos permite cuantificarla similitud entre dos señales del mismo tipo, esta etapa de codificación se realizó en el programa “SignalAnalisisAndDB.py” y se utilizó la librería “numpy”. En la función “Correlacion” se extrae el canal del audio base para la comparación y el audio a comparar para posteriormente calcular la amplitud máxima de la señal base. Se realiza la división entre los datos del audio y la amplitud máxima de la señal base. Con ayuda de la librería se calcula la correlación entre ambas señales y se calcula el valor de la correlación normalizada.

Crear la base de datos en un archivo .csv

Se generó un archivo con ayuda de las librerías “csv” y “pandas” en el programa “SignalAnalisisAndDB.py” en la función con nombre “createfilescsv”; esta función requiere del nombre del archivo que se va a generar y los encabezados de cada columna. El nombre del archivo de la base de datos generada es “Voice_Commands_DataBase2.csv”.

Adición de los datos al archivo .csv

Para añadir automáticamente FFT, la correlación, el nombre y la etiqueta para cada audio se utilizaron las librerías “Pandas” y “csv” en el mismo programa dentro de la función “addtocsv”, esta función requiere del nombre del archivo al que se va a agregar los datos, en este caso “Voice_Commands_DataBase2.csv” y las características a guardar en cada columna.

Finalmente utilizando la librería “glob” se desplegaron los 990 audios para posteriormente, uno por uno, ir caracterizando y guardando cada uno. Esto se realizó de forma automática con el conjunto de códigos “SignalAnalisisAndDB.py”, “MakeDataBase.py” y “VoiceCommandsDataBase.py”. Siendo “VoiceCommandsDataBase.py” el único ejecutable.

Primera incorporación de la red neuronal

En el programa "NeuronalTest.py" se abrió el archivo de la base de datos, se entrenó la red neuronal con esta base de datos y se calculó el desempeño de la red neuronal con las características de norma y correlación, el total de aciertos y el porcentaje de clasificación correcta.

Realización de gráficos

Se realizaron las gráficas de dispersión y de regresión lineal. Para esto primero se leyó el archivo "Voice_Commands_DataBase2.csv" para posteriormente obtener los valores y etiquetas de cada característica. Una vez que se obtuvieron esos valores se calculó la ecuación de la recta de regresión lineal para posteriormente realizar la gráfica de dispersión lineal general, asignando un color para cada etiqueta de comando.

Para cada comando se generó su propio grafico de dispersión con su respectiva recta de regresión lineal, para mostrar el comportamiento de los datos de cada comando.

Segunda incorporación de la red neuronal

Se realizó una limpieza de los datos, retirando los datos atípicos para cada grupo de valores por comando y se entrenó la red neuronal con esta base de datos para después calcular su desempeño, el total de aciertos y el porcentaje de clasificación correcta.

VI. RESULTADOS

Al ejecutar el programa "PyAudio_BD.py" se muestra lo siguiente, en esta parte de la ejecución del código se debe de introducir el número de persona que se va a grabar.

```
In [1]: runfile('C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/
CONTINENTAL/Algoritmos/pyAudio_BD.py', wdir='C:/Users/DULCE/Desktop/UAQ FI/
8°/OPTATIVA FORMACIÓN DUAL/CONTINENTAL/Algoritmos')

Ingresa el # de persona: 5
```

Ilustración 2. Texto en el compilador al ejecutar "PyAudio_BD.py"

Después de ingresar de número de persona que se está grabando el programa asigna el nombre a la próxima grabación que se va a tomar y se muestra en la terminal, después de esto se muestra un mensaje destinado al usuario que se encuentra grabando los audios. Y por último se procede a tomar la grabación de 3 segundos, esto se hace constantemente hasta terminar de grabar los 55 audios por persona.

```
In [1]: runfile('C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/
CONTINENTAL/Algoritmos/pyAudio_BD.py', wdir='C:/Users/DULCE/Desktop/UAQ FI/
8°/OPTATIVA FORMACIÓN DUAL/CONTINENTAL/Algoritmos')

Ingresa el # de persona: 5
C1_P5_G1
Comienza a hablar cuando veas la leyenda _grabando_
grabando...
grabación terminada


C1_P5_G2
Comienza a hablar cuando veas la leyenda _grabando_
grabando...
grabación terminada


C1_P5_G3
Comienza a hablar cuando veas la leyenda _grabando_
grabando...
grabación terminada


C1_P5_G4
Comienza a hablar cuando veas la leyenda _grabando_
grabando...
grabación terminada


C1_P5_G5
Comienza a hablar cuando veas la leyenda _grabando_
grabando...
grabación terminada
```

Ilustración 3. Proceso de grabación de audios.

Al finalizar la ejecución de código "PyAudio_BD.py" todos los audios grabados se guardan en la misma dirección en la que se encuentra el código nombrados de acuerdo a lo establecido.

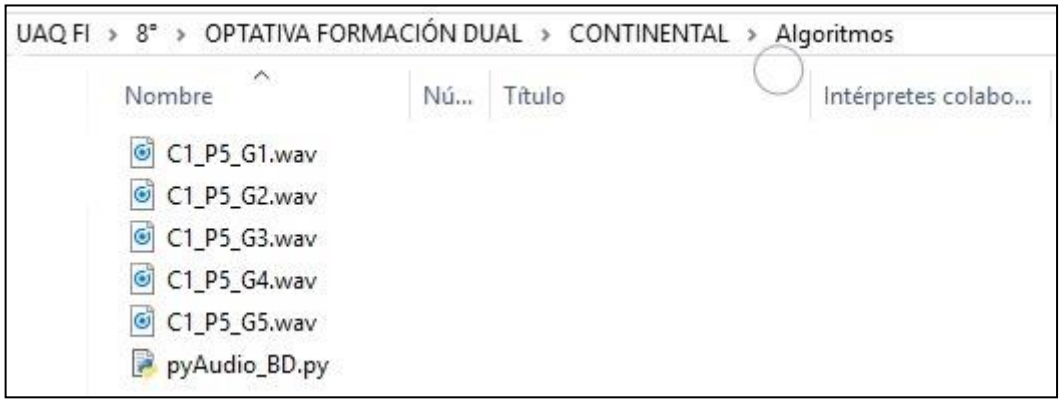


Ilustración 4. Archivos .wav generados.

Al ejecutar el programa “” se observa el mensaje mostrado en la siguiente ilustración, después de esto es posible ver el archivo .csv en la dirección donde se encuentra el código.

```
In [2]: runfile('C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/
PRUEBAS FFT/Ultima Versión para crear BD/Voice_Commands_DataBase.py',
wdir='C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/PRUEBAS FFT/
Ultima Versión para crear BD')
Archivo Voice_Commands_DataBase creado.
```

Ilustración 5. Mensaje presentado en la terminal al ejecutar el código “”.

Durante la ejecución de este programa, después de crear el archivo .csv comienza a indicarnos en la terminal el elemento que se va caracterizando y el elemento que se toma como audio base para realizar la correlación, durante este proceso se va llenando de forma automática nuestra base de datos con las características de interés para cada audio

```
In [2]: runfile('C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/
PRUEBAS FFT/Ultima Versión para crear BD/Voice_Commands_DataBase.py',
wdir='C:/Users/DULCE/Desktop/UAQ FI/8°/OPTATIVA FORMACIÓN DUAL/PRUEBAS FFT/
Ultima Versión para crear BD')
Archivo Voice_Commands_DataBase creado.
Audio Base C10_P16_G1.wav ELEMENTO: C10_P10_G1.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P10_G2.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P10_G3.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P10_G4.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P10_G5.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P11_G1.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P11_G2.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P11_G3.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P11_G4.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P11_G5.wav
Audio Base C10_P16_G1.wav ELEMENTO: C10_P12_G1.wav
```

Ilustración 6. Proceso de caracterización y creación de la base de datos de forma automática.

Al finalizar la ejecución del código se tiene como resultado, el archivo de la base de datos con la información de cada audio.

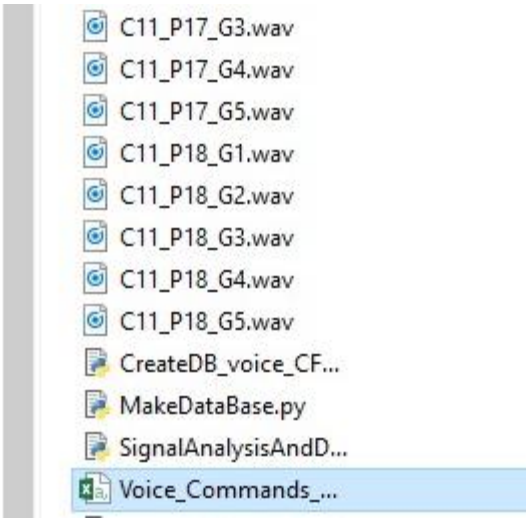
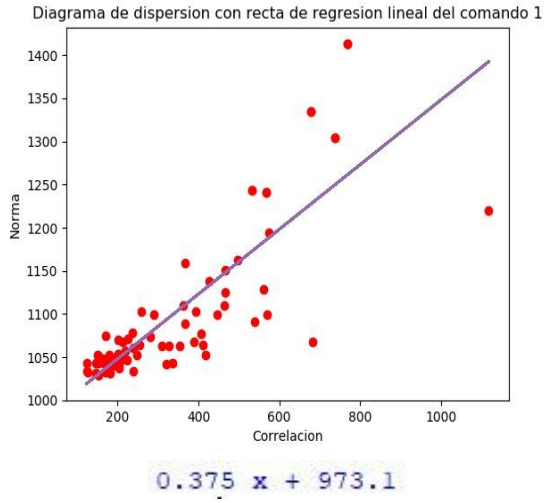
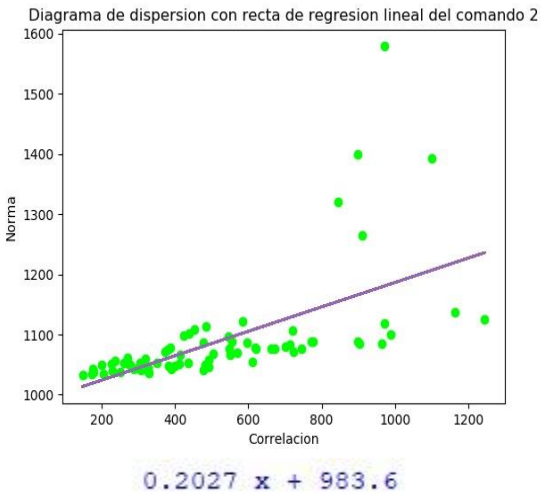


Ilustración 7. Archivo .csv con la base de datos.

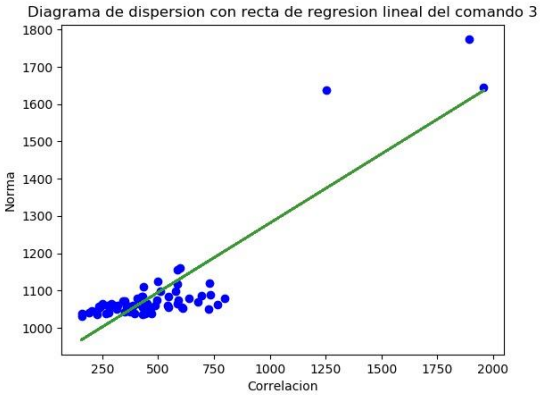
Los resultados de las gráficas de dispersión para cada comando se muestran en las siguientes figuras, así como la regresión lineal y la ecuación característica correspondiente para cada caso.

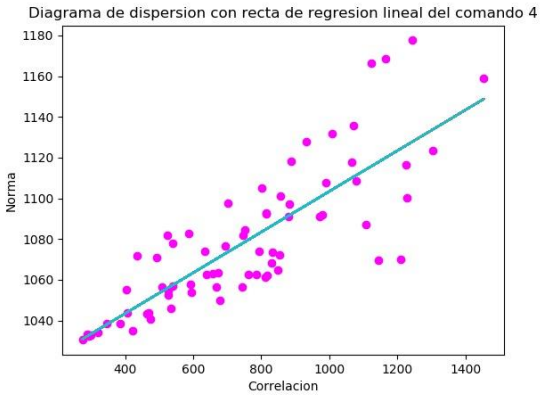


0.375 x + 973.1

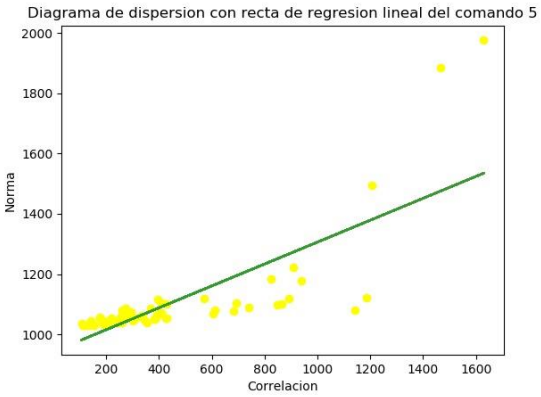


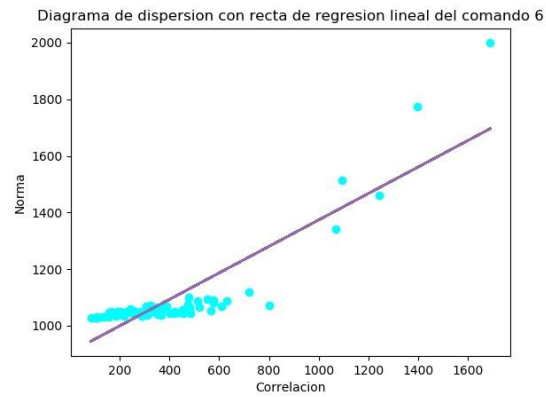
0.2027 x + 983.6

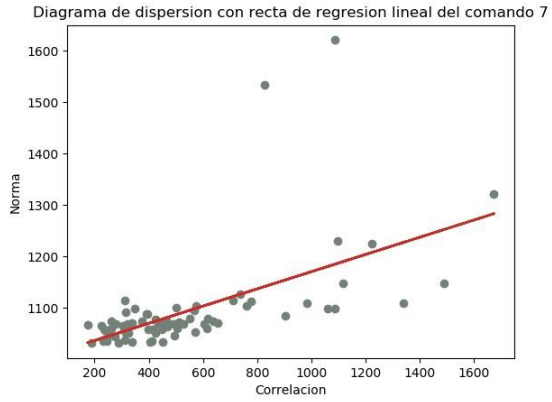




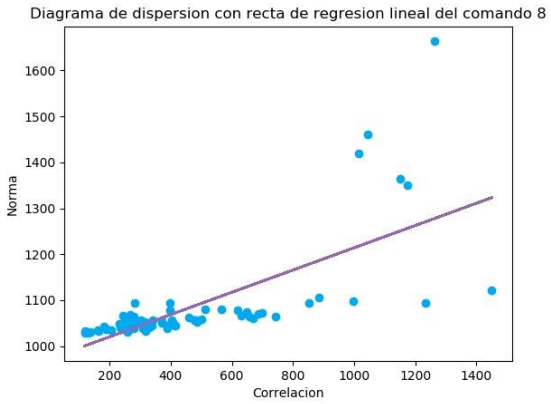
Conforme se observa en cada grupo de datos para cada comando, hasta el caso de la dispersión del comando 4 se puede observar que la dispersión de los grupos es diferente entre ellos, para algunos casos hay poca dispersión y se encuentran pocos *outliner como* es el caso del comando 3 y el comando 2, los datos se encuentran concentrados en un área específica, esto es bueno ya que no debería existir dispersión porque se trata de datos que pertenecen al mismo comando, en el caso de comando 4 sucede algo diferente, la dispersión es muy notable en todo el rango de valores de la gráfica, no se presenta una concentración por lo que se puede decir que las características de este grupo de audios no nos aporta información útil para nuestra base de datos.





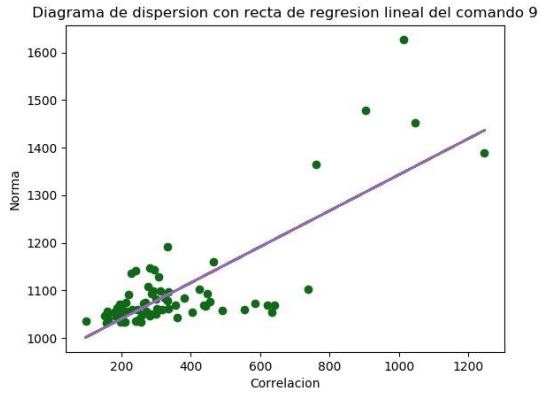


0.1674 x + 1003

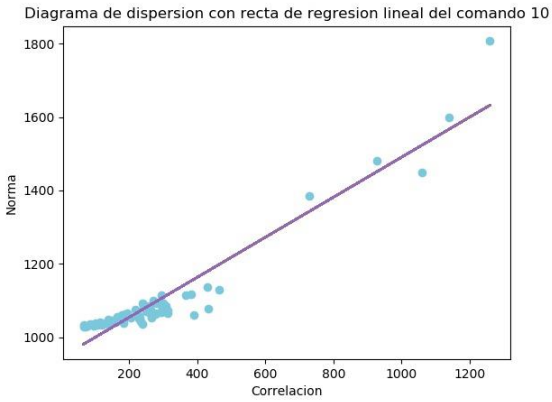


0.2422 x + 971.8

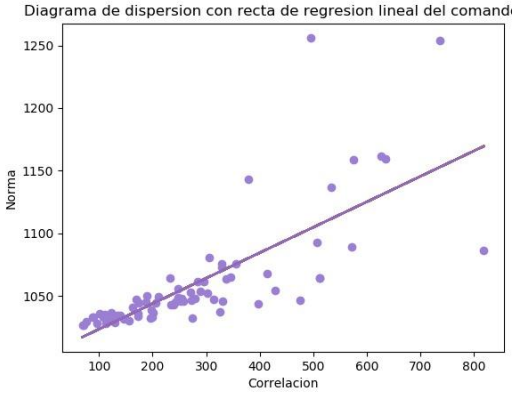
Se puede observar en cada gráfica que la mayoría de los grupos de comandos presentan algunos *outliner* que hasta el momento se creyó que podría ser posible evitar su efecto en la regresión lineal con algún método estadístico, sin embargo, también se observa que las concentraciones de los grupos se encuentran casi en la misma área, esto no es un resultado satisfactorio ya que deberían diferenciarse los comando por los valores de las características que presenta cada comando. Lo mismo sucede para los diagramas de los comandos 9, 10 y 11 que se muestran en las siguientes figuras.



0.3788 x + 964.4

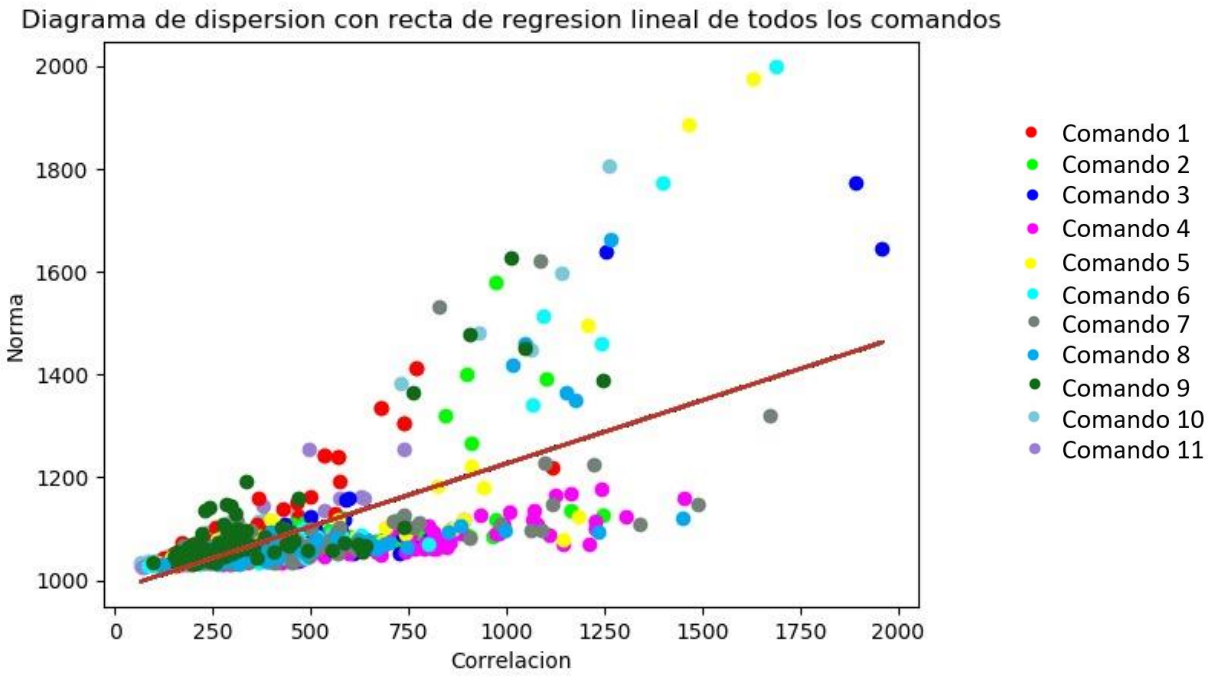


0.5441 x + 946.1



0.2035 x + 1003

La agrupación de la dispersión de todos los comandos se muestra en la siguiente figura. En ella finalmente se comprueba que todos los comandos se encuentran agrupados casi en su totalidad en una misma área, esto nos indica que no hay una diferencia notable entre los valores de las características para ninguno de los grupos lo que nos impide hacer una separación e identificación correcta de los comandos por medio de las características (FFT y Correlación) que se tomaron para este caso.



El resultado al entrenar la red neuronal con esta base de datos que presenta malas características es un desempeño del 11.11% con 110 coincidencias de un total de 990.

Al eliminar los *outliner* de la base de datos el desempeño de la red neuronal fue de 15.09% con 123 coincidencias de un total de 990.

Es notable que el desempeño de la red neuronal es afectado por los tipos de datos que la integran ya que como se comprobó las características no fueron buenas, una solución a este problema es realizar otra caracterización de las señales con mayor relevancia que permitan la clasificación, hacer uso de filtros y otro tipo de procesamiento a las señales.

VII. CODIGOS

Todos los códigos utilizados para el desarrollo del proyecto, hasta los resultados presentados en este reporte, se encuentran en el siguiente link de repositorio en GitHub.

https://github.com/ELGeb/Control-de-versiones-de-proyecto-formacion-dual/tree/master/Algoritmos/C%C3%B3digos%20Para%20Generar%20Base%20de%20Datos

VIII. CONCLUSIONES

Es importante seleccionar las características de acuerdo al tipo de datos que se tiene para generar la base de datos, estas características deben representar de forma significativa a la señal o dato al que pertenecen, en nuestro caso las características seleccionadas no fueron útiles para el buen desempeño del proyecto. El proceso de *Data Science* es la etapa más importante durante el desarrollo del proyecto ya que con un análisis básico de los datos se puede identificar si estos son de utilidad para nuestra aplicación en la red neuronal y con esto tener una aproximación del porcentaje de desempeño que se obtendrá tendrá o como sucedió en este caso, demostrar que las características no aportan información útil.

Para continuar con el desarrollo de este proyecto se sugiere realizar una nueva caracterización de los datos, hacer un procesamiento significativo en las señales y seleccionar las características que sean de mayor utilidad para la clasificación, además la selección de elementos que constituirán nuestro resultado final debe analizarse de forma minuciosa y evaluar sus efectos en caso de no hacerlo.

IX. REFERENCIAS

- Códigos proporcionados por el mentor durante las sesiones de la clase de algoritmos.
- Fuentes Lorena., Cinta Israel., Ramírez Luis, Sánchez Jorge & Sánchez Verónica (2018, 19 junio). Detección de grosor de placas mediante análisis de frecuencias. Reporte Final Análisis de señales, 1–12.
- Anónimo. (2019). Aprendizaje automático. Diciembre 11, 2019, de SAS Institute Inc. Sitio web: https://www.sas.com/es_mx/insights/analytics/machine-learning.html
- Anónimo. (2019). Redes Neuronales. Diciembre 11, 2019, de Universidad de Salamanca. Sitio web: http://avellano.fis.usal.es/~lalonso/RNA/index.htm
- Obiols, A. (2015). ¿Qué es un data scientist?. Diciembre 11, 2019, de inLab FIB.Sitio web: https://inlab.fib.upc.edu/es/blog/que-es-un-data-scientist
- La comunidad SciPy. (2019, 26 julio). Discrete Fourier Transform (numpy.fft) — NumPy v1.17 Manual. Recuperado 1 diciembre, 2019, de ttps://docs.scipy.org/doc/numpy/reference/routines.fft.html

- Escuela Universitaria de Ingeniería Técnica Industrial Bilbao. (s.f.). Correlación cruzada y autocorrelación. Recuperado 3 diciembre, 2019, de http://www.ehu.eus/Procesadodesenales/tema8/corre1.html