Actividad: Clasificación con máquina de vectores de soporte y redes de neuronas

Introducción

A lo largo del nuestro día se toman variadas decisiones basadas en clasificación, lectores de huellas dactilares, reconocimiento facial, filtros de spam para correo electrónico, etc.

En la inteligencia artificial uno de los métodos más utilizados son las redes neuronales y las Maquinas de Soporte Vectorial (SVM) para realizar problemas tanto de regresión como de clasificación.

El objetivo de esta práctica es conocer mejor cómo funcionan estos algoritmos para realizar un problema de clasificación basado en el precio de los celulares, utilizando un set de datos tomado de la página de kaggle con múltiples datos recolectados.

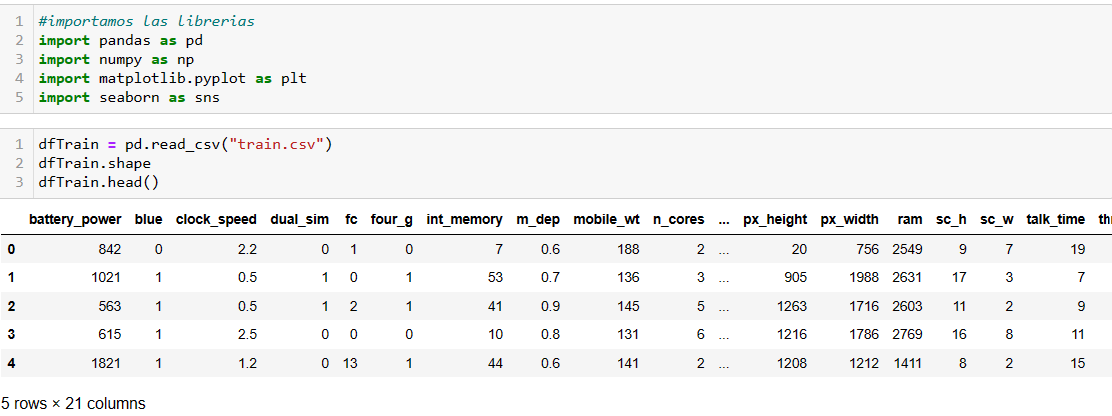
Desarrollo

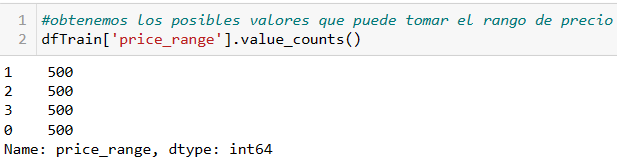
**Máquina de soporte vectorial (SVM)**

La primera parte de la práctica se centra en los SVM, los cuales funcionan correlacionando los datos en un plano de características de grandes dimensiones de forma que los puntos en el plano se puedan categorizar correctamente, se añade un separador el cual es una línea que se encargara de dividir y categorizar los datos correctamente.

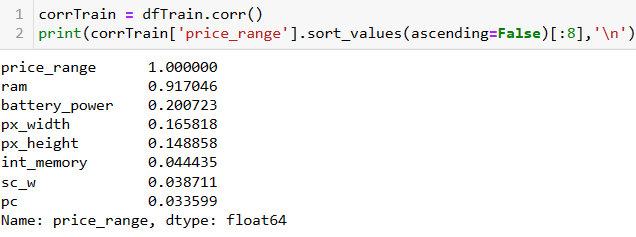
*Análisis de los datos*

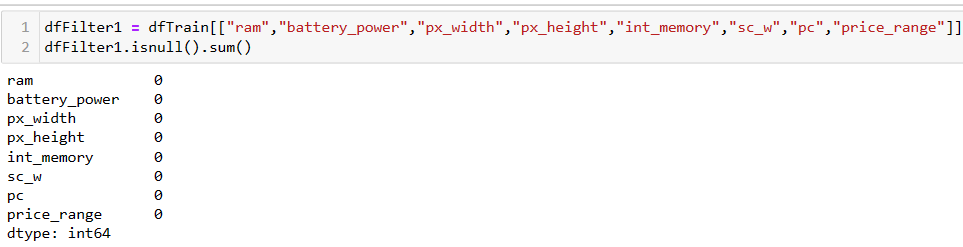
Para iniciar la practica primero se deben importar las librerías necesarias y conocer los datos con los cuales se va a trabajar, para esto cargamos el dataset y con la función head () de pandas, observamos algunas de nuestras variables con su respectivo valor.

Se procede a obtener los posibles valores que puede tomar nuestra variable de interés, en este caso la variable de interés es ‘Price\_range’.

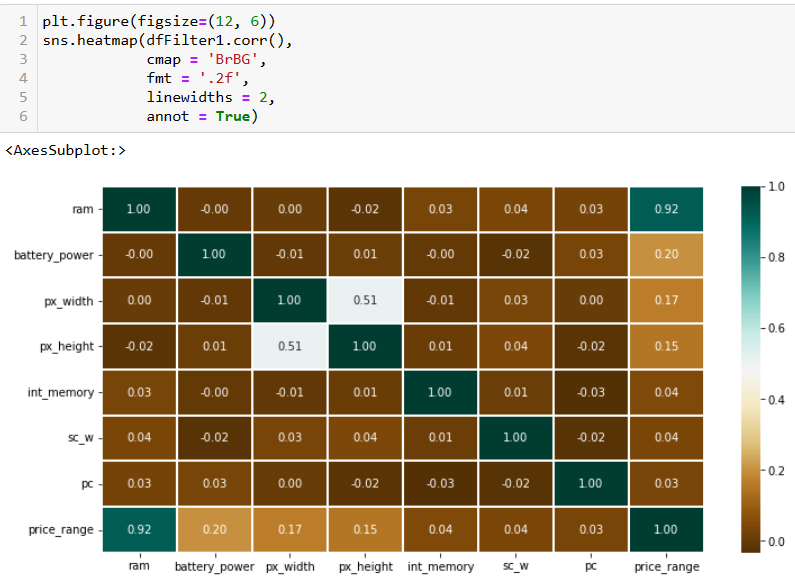


Con la realización de una correlación con la variable de interés, se obtienen las variables mas relevantes que influyen en el precio de un celular, en la siguiente imagen se muestran las variables correlacionadas.



Como se observa en la imagen, una de las variables que mas influye en el precio de un teléfono es la ram y las demás variables influyen en menor medida, se procede a realizar un filtrado con estas variables que influyen directamente en el precio, además se realiza un proceso para asegurarnos de que no vengan valores nulos en estas variables.

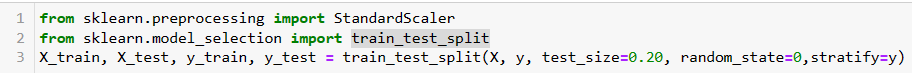
Una vez realizado el filtrado, se procede nuevamente a realizar la correlación para ver mejor como se relacionan los datos, pero se realiza el proceso de una manera mas grafica para entender mejor nuestras variables respecto al precio de los teléfonos.



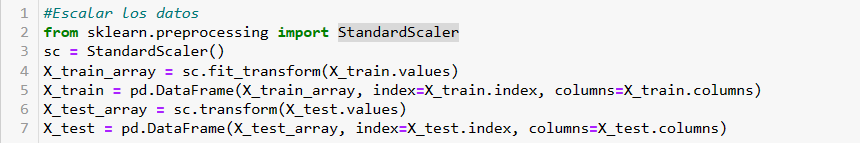
Una vez que se entiende la relación de los datos seleccionados con los de la variable objetivo, se realiza una división entre los datos a analizar y la variable de interés que en este caso es ‘price\_range’, se dividen los datos y se guardan en las variables (X,y), donde la “y” tiene la variable de interés y “X” conserva los demás datos excluyendo la variable “Price\_range”.



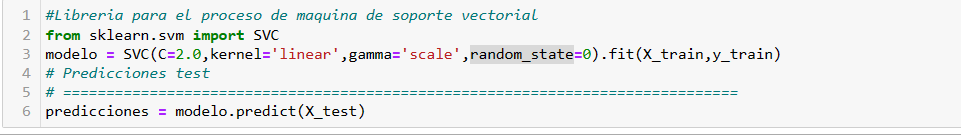
Se procede a dividir los datos para las pruebas y entrenamiento con la función train\_test\_split.



La máquina de soporte vectorial funciona mejor con datos escalados, por lo tanto, se realiza un proceso de escalamiento con la librería StandardScaler, se escalarán tanto los sets de test y train.

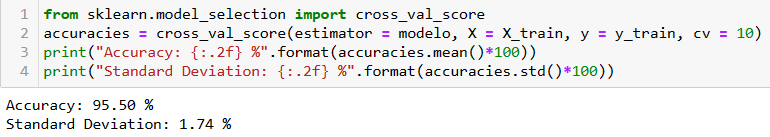


Se procede a realizar el modelo de máquina de soporte vectorial, para esto se utiliza la librería sklearn y se importa el SVC, se especifica el kernel, el cual puede recibir 5 posibles instrucciones *‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’ y ‘precomputed’,* para la practica se probo con los 5 y el que dio mejores resultados fue el “linear”, también recibe el parámetro gamma para esta practica se utilizo el valor de “scale”, otro parámetro importante es el de random\_state, el cual se indico con 0 para que no presente aleatoriedad. Una vez realizado el modelo, con la función predict, se realizan las predicciones necesarias en nuestro algoritmo.

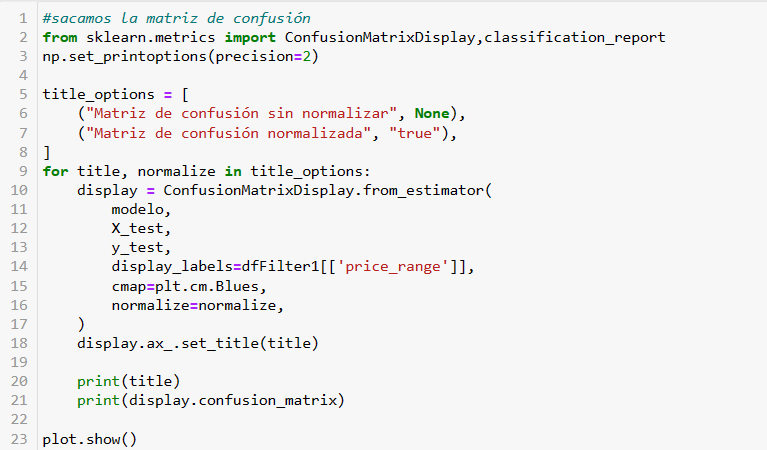


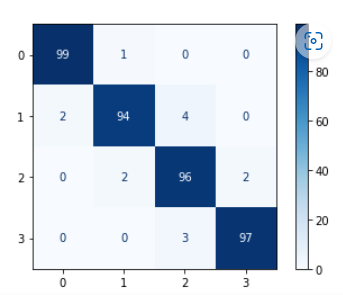
*Rendimiento del modelo MSV*

Para evaluar el desempeño de nuestro modelo se realiza el cálculo del accuracy, la cual nos indica que tan cerca están nuestros resultados del valor autentico de nuestro objetivo o meta, también se utiliza la desviación estándar, cual nos indica la dispersión o desviación típica en el conjunto de datos, es decir que tanto se equivocó porcentualmente nuestro modelo al momento de realizar la clasificación.



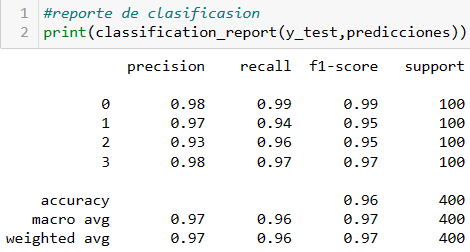
Como se puede observar con este modelo de MSV se obtuvo un accuracy de 95.50 % y una desviación estándar de 1.74%, lo cual es bastante bueno, pues se puede decir que el modelo esta lo suficientemente capacitado para clasificar correctamente los datos con un margen de error muy poco.

Otra forma de evaluar el modelo es utilizar una matriz de confusión, la cual es ampliamente utilizada en el campo de la IA, ya que es una herramienta que sirve para evaluar el desempeño de un algoritmo clasificatoria que se emplee en aprendizaje supervisado.

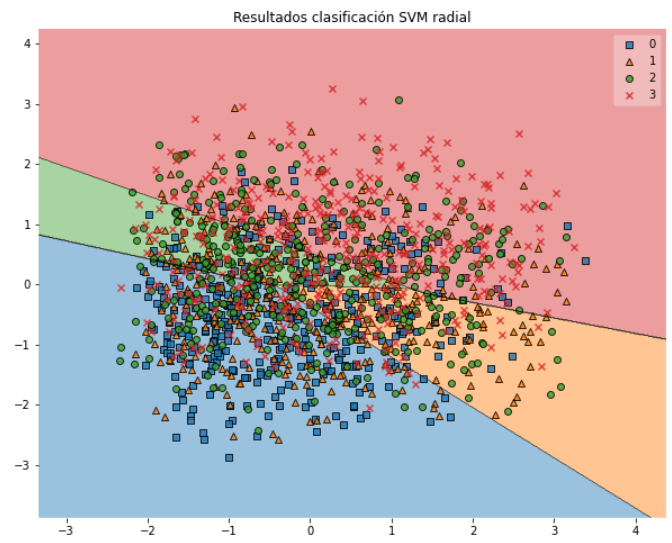


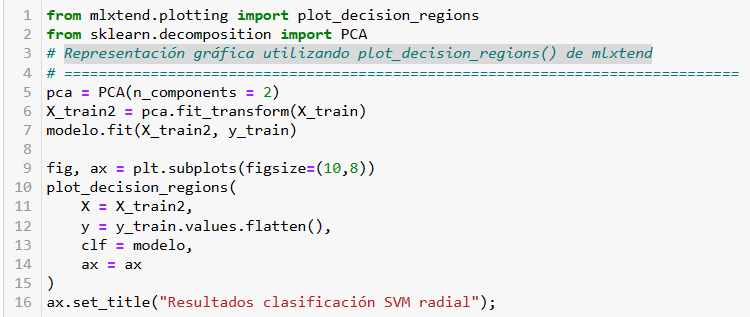
En la matriz se puede ver como clasifica los rangos de precios 0,1,2 o 3, se observan pequeños errores en el que por ejemplo se equivoca 2 veces al querer poner el precio en el rango de 0 y lo asigna en el valor de 1, pero estos son errores menores y se muestra como presenta un gran desempeño el modelo.

Por último, para ver mejor el desempeño se realiza un reporte de clasificación, en el cual se muestran los datos mas relevantes como la precisión, el re-call e incluso el f-Score.



*Resultados del modelo MSV*

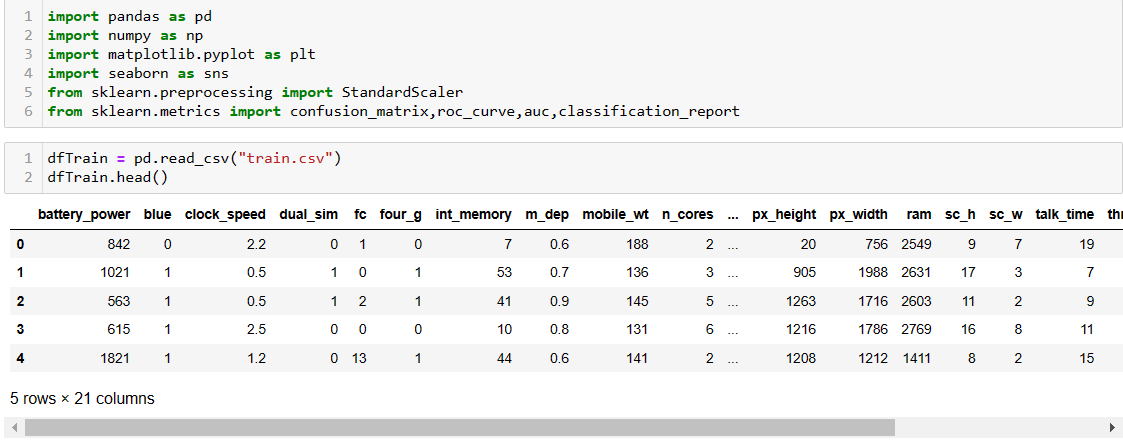
****Para ver el proceso de clasificación que utiliza el modelo se realiza una representación gráfica utilizando plot\_decision\_regions() de mlxtend.

****En este grafico se observa como clasifica los valores de rango de precio 0, 1, 2 o 3 en su respectiva zona, al ser muchos datos no se alcanza a apreciar tan bien como realiza la clasificación.

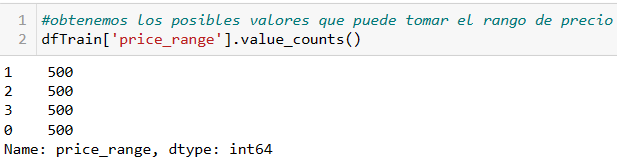
**Redes neuronales**

La segunda parte de la práctica se centra en las redes neuronales de clasificación, los cuales funcionan con un nivel de neuronas interconectadas entre si, tal y como funcionan nuestro cerebro, son un conjunto de nodos que se configuran de tal modo que aportan una solución al problema de clasificación.

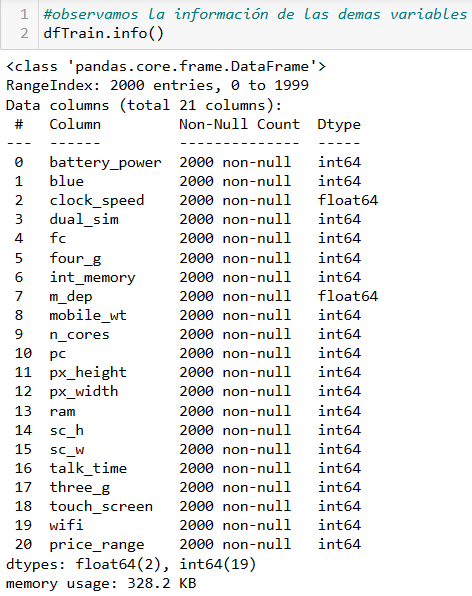
*Análisis de los datos*

Para comenzar el ejercicio, primero importa las librerías necesarias y conoce los datos con los que quieres trabajar, para esto cargamos un dataset y usando la función pandas head() observamos algunas variables y sus valores correspondientes.

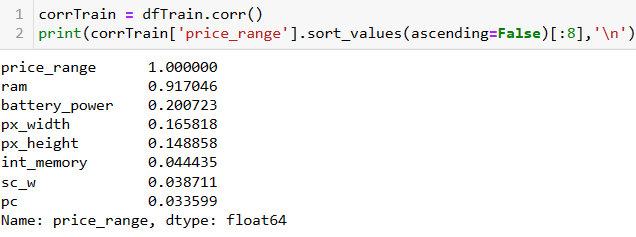
Seguimos obteniendo los posibles valores que puede tomar nuestra variable de interés, en este caso la variable de interés es ‘Price\_range’.

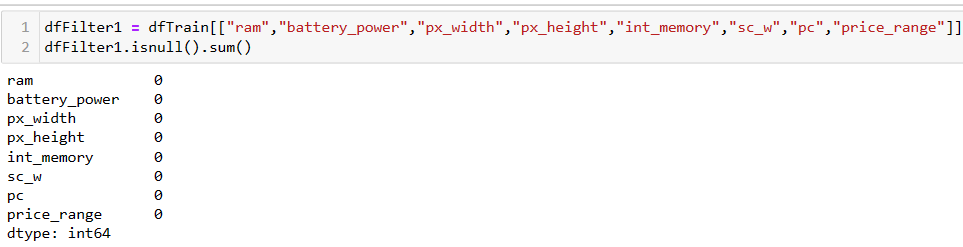


Se observan las demás variables con su respectiva información.

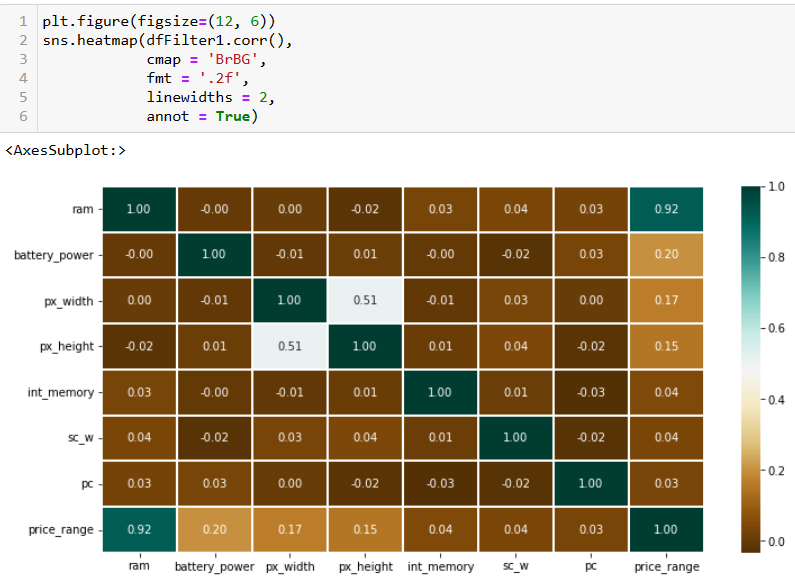


Correlacionando con la variable de interés se pueden obtener las variables más relevantes que afectan al precio de un teléfono móvil. La siguiente figura muestra las variables relevantes.



Como se observa en la imagen, una de las variables que mas influye en el precio de un teléfono es la ram y las demás variables influyen en menor medida, se procede a realizar un filtrado con estas variables que influyen directamente en el precio, además se realiza un proceso para asegurarnos de que no vengan valores nulos en estas variables.

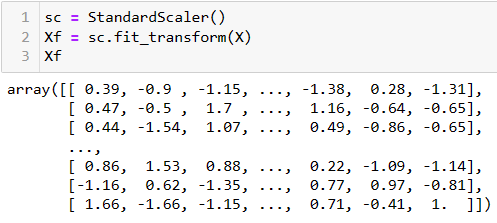
Una vez hecho el filtrado, volvemos a hacer la correlación para tener una mejor idea de la relevancia de los datos, pero de forma más gráfica para tener una mejor idea de las variables de precio del teléfono.



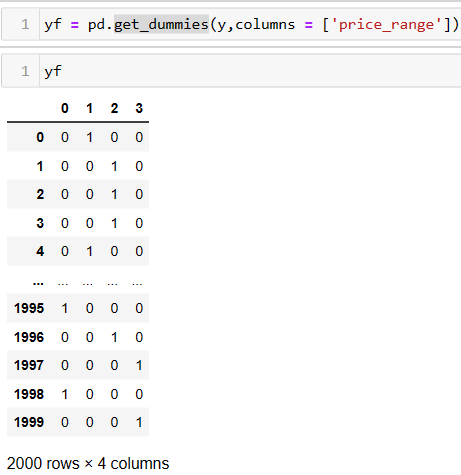
Una vez que se entiende la relación de los datos seleccionados con los de la variable objetivo, se realiza una división entre los datos a analizar y la variable de interés que en este caso es ‘price\_range’, se dividen los datos y se guardan en las variables (X,y), donde la “y” tiene la variable de interés y “X” conserva los demás datos excluyendo la variable “Price\_range”.



Se procede a realizar un escalamiento de los datos para que el algoritmo funcione mejor.

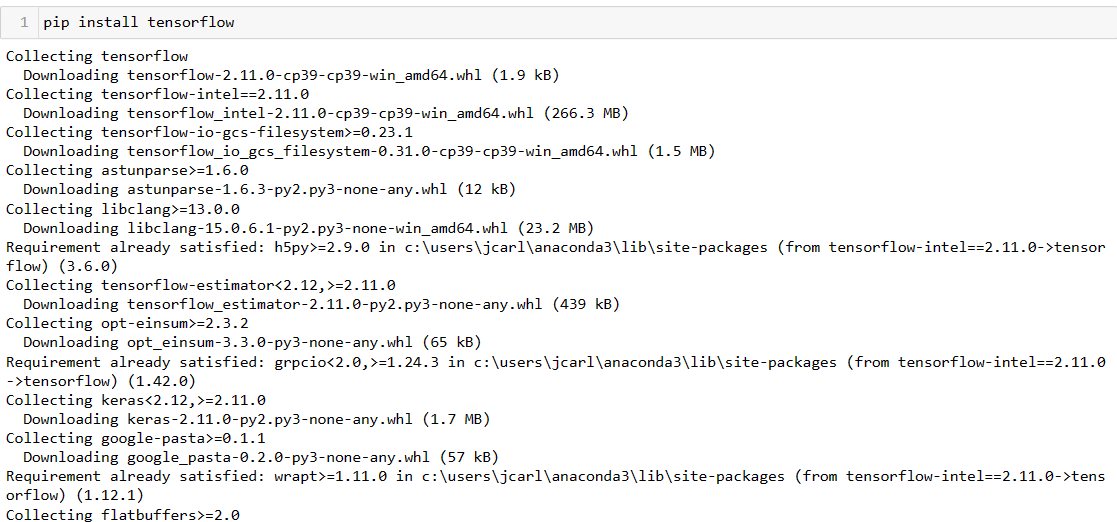


Con las variables de salida, se realiza un procedimiento para cambiar los valores categóricos en binarios, ya que las redes neuronales trabajan mejor con valores de 0 y 1.

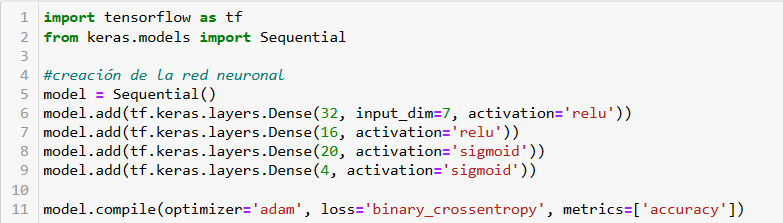


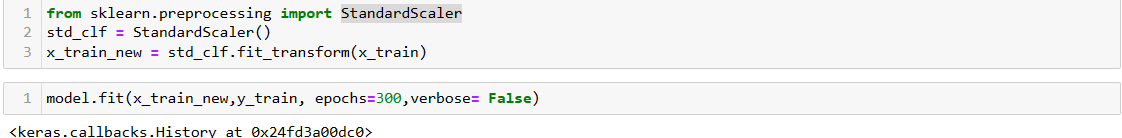
Una vez realizado el proceso de tratamiento de datos, se realiza la división para el set de entrenamiento y el set de test.



****Para realizar la red neuronal, se instala tensorFlow en caso de no estar instalado, mediante la siguiente instrucción se puede realizar la instalación en jupyter notebook.

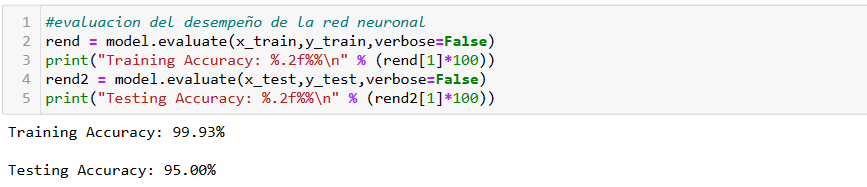
Una vez instalada la librería tensorFlow se procede a importar la librería y a crear la red neuronal con la siguiente instrucción.



Una vez creada la red neuronal se realiza un escalamiento de los datos para que sean valores mejor procesados por nuestra red, mediante la librería StandardScaler y una vez escalado nuestros datos se realiza un entrenamiento con los datos ya escalados.

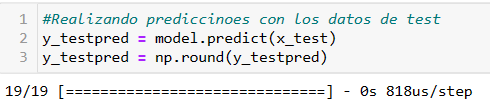
*Rendimiento del modelo Red neuronal*

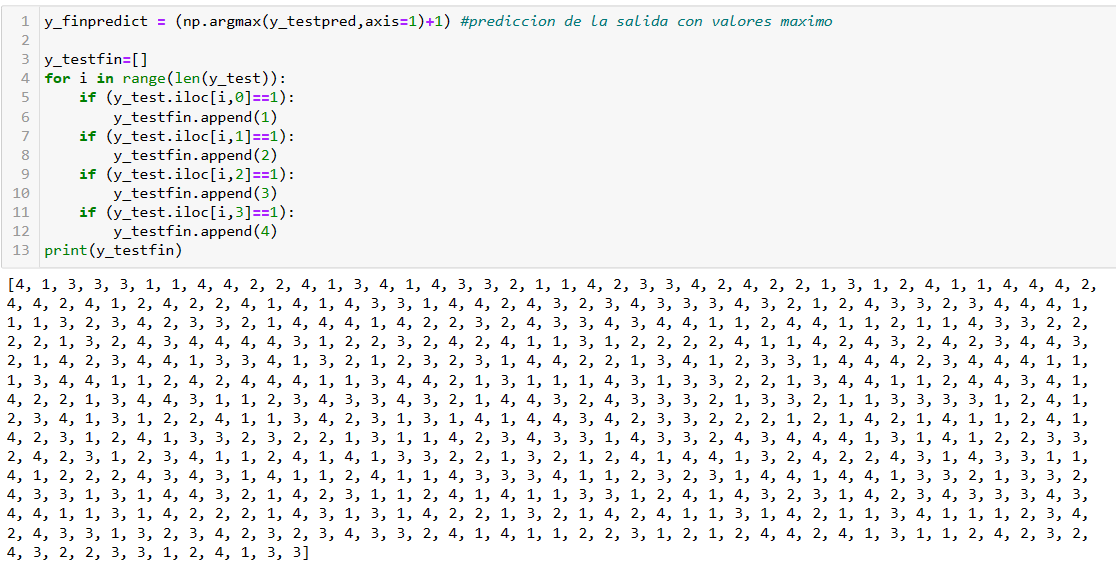
Para evaluar el modelo creado anteriormente a partir de una red neuronal, se realiza un calculo de accuracy tal y como se muestra en la siguiente imagen.

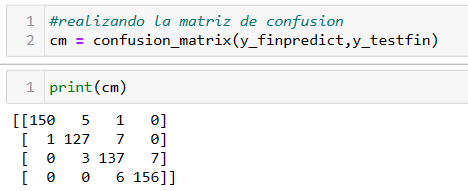


Como se observa en la imagen, el modelo esta bien entrenado tanto con los datos de training como los de testing, por lo cual se puede decir que realiza las predicciones en su mayoría bien y en este caso no presenta sobre entrenamiento por lo cual no hay necesidad de realizar algún ajuste extra.

Una vez revisada el accuracy se realizan las predicciones con los datos de test.

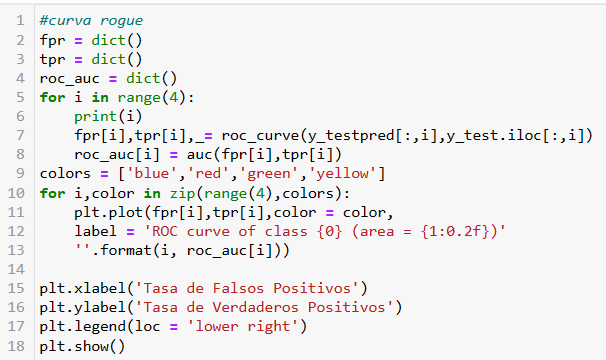


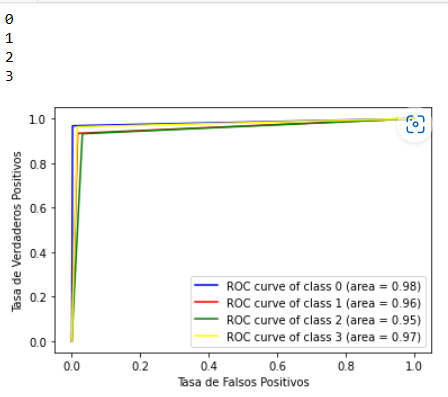
Para observar mejor el desempeño del modelo, se realiza una matriz de confusión y para realizarla primero se predicciones de salida con los valores máximos.

Una vez se obtienen estos datos se crea la matriz de confusión y se grafica.

Como se observa el modelo, clasifica correctamente los datos, salvo en unas ocasiones donde se equivoca y los coloca en otra fila o columna que no debería ser, esto es normal ya que nuestro modelo no presenta un 100% de accuracy por lo cual quiere decir que no es perfecto, pero viendo la matriz y observando los datos se llega a la conclusión de que realiza un trabajo bien hecho a la hora de clasificar los datos.

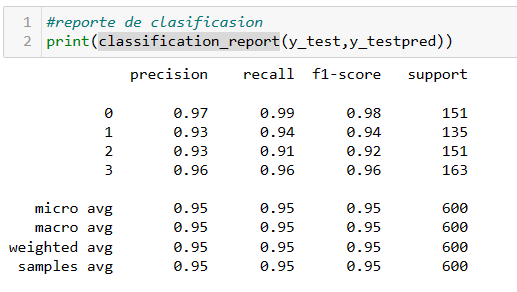
Otra forma de evaluar el modelo, es mediante la curva ROC, la cual es un procedimiento que se usa ampliamente en la ciencia de datos e IA para evaluar modelos de clasificación, el sitio oficial nos da la siguiente definición para entender mejor de que van estas curvas.

“El Análisis ROC (Receiver operating characteristic o característica operativa del receptor) es una forma útil de evaluar la precisión de las predicciones de modelo al trazar la sensibilidad frente a la especificidad de una prueba de clasificación (ya que el umbral varía en todo un rango de resultados de pruebas de diagnostico). El Análisis ROC admite la inferencia en relación con una sola AUC, curvas de precisión/exhaustividad (PR), y proporciona opciones para comparar dos curvas ROC que se generan de grupos independientes o sujetos emparejados.”



Con estas curvas Roc se puede observar que nuestro modelo tiene pocos errores de falsos positivos.

Por último, para finalizar el análisis de desempeño del modelo, se realiza un reporte completo de rendimiento.



Conclusiones

Las técnicas observadas en esta práctica, tanto redes neuronales como maquinas de soporte vectorial, me parece de una gran utilidad para temas de clasificación, ambos modelos presentaron resultados excelentes y su desempeño fue bastante bueno para este set de datos analizado. Sin embargo, en gustos personales preferiría las redes neuronales ya que permiten una mejor personalización y ajuste, por lo cual creo que se pueden adaptar mejor a otros tipos de set de datos y problemas, sin embargo, para el caso de las maquinas de soporte vectorial me parecieron bastante fáciles de implementar, además de dar resultados bastantes óptimos.

Sin duda ambas estrategias funcionan muy bien para realizar problemas de clasificación, en un futuro se pondrán a prueba con algún problema de regresión para ver que tal se comportan ambos modelos, ya que según leí en varias paginas estas dos estrategias funcionan tanto para regresión como para clasificación.

Bibliografía

TensorFlow (S.F). Clasificacion Basica: Predecir una imagen de moda Recuperado el 27 de 02 de 2022, de

https://www.tensorflow.org/tutorials/keras/classification?hl=es-419

Máquinas de Vector Soporte (SVM) con Python by Joaquín Amat Rodrigo, available under a Attribution 4.0 International (CC BY 4.0) at https://www.cienciadedatos.net/documentos/py24-svm-python.html

IBM (2021). *IBM*-Análisis ROC. Recuperado el 01 de 03 de 2022, de https://www.ibm.com/docs/es/spss-statistics/beta?topic=features-roc-analysis.