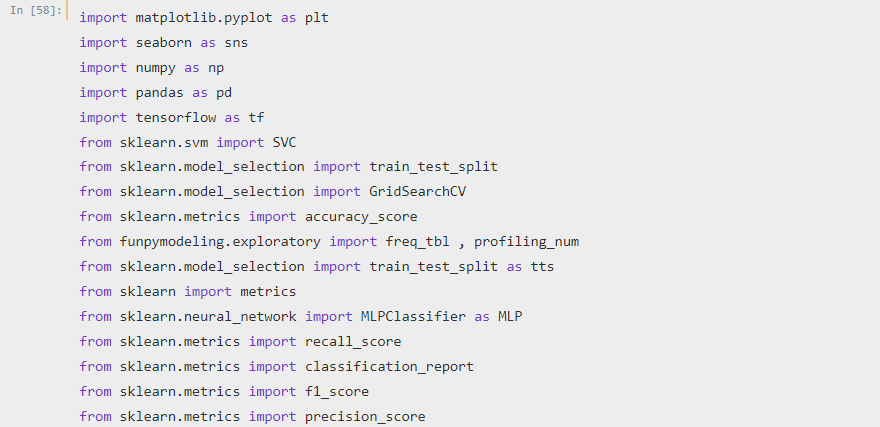
Actividad: Clasificación con máquina de vectores de soporte y redes de neuronas

Objetivos

Se analizará el archivo DataSet mobile-price-classification#train.csv para aplicar Redes Neuronales y Máquinas de vectores con la finalidad de predecir la columna “price\_range” , es decir, el precio de los distintos móviles que se listan en el archivo.

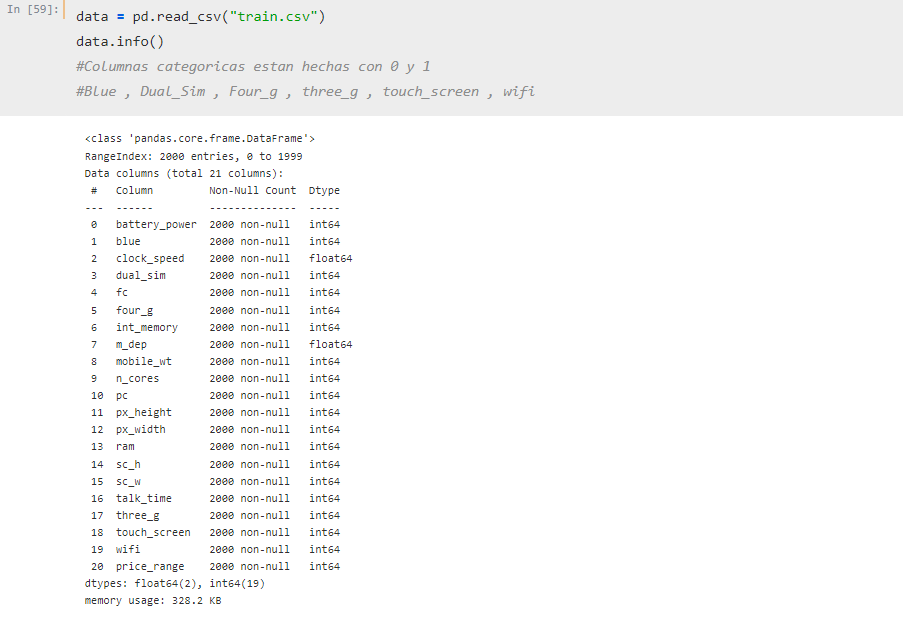
Librerías a usar

A continuación, se listan las librerías a usar en el trabajo.

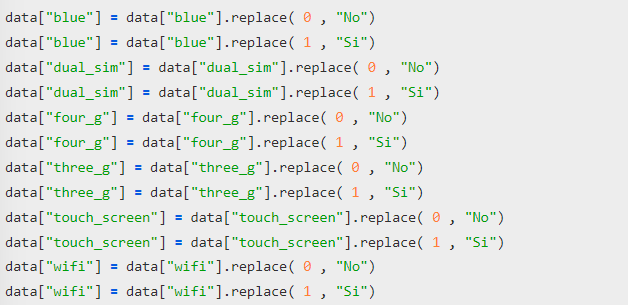


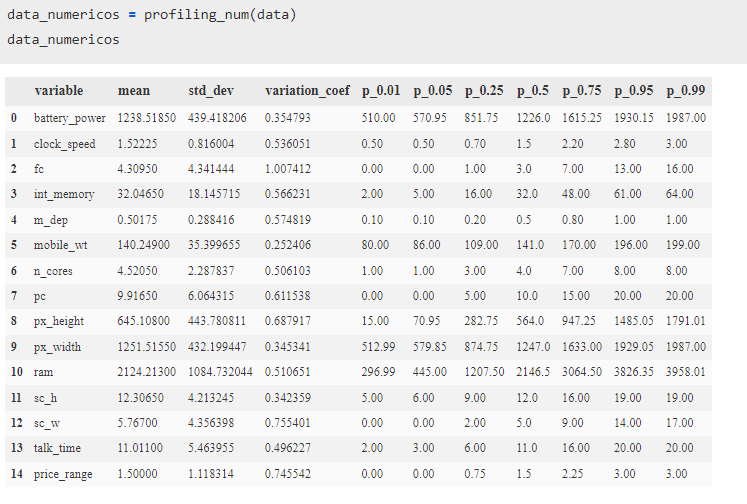
Análisis Descriptivo de los datos

Empezaremos el análisis cargando el archivo “train.csv”, asimismo, daremos un rápido vistazo al tipo de datos que contiene cada columna de la Base de Datos.

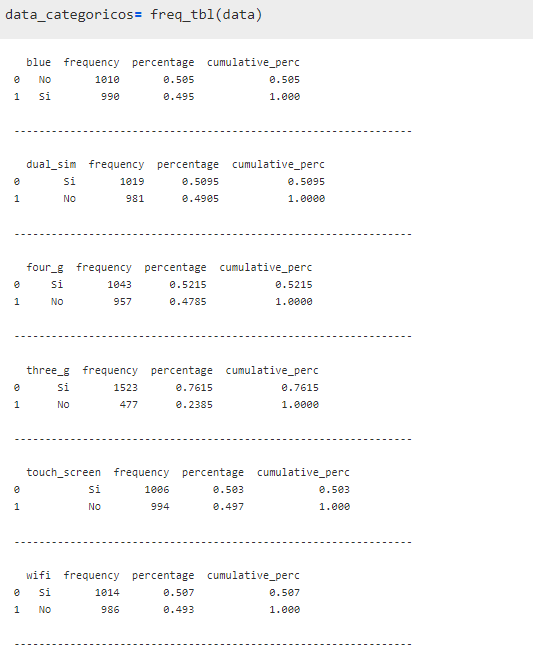


Notaremos inmediatamente que todas las columnas son tipo Float64 o Int64, esto significa que son datos numéricos. Por otra parte, al abrir la base de datos nos percatamos de que en realidad existen 6 columnas, y no obstante que sean números, en realidad son datos categóricos, toda vez que contienen un 0 (false) y 1 (true). Para poder entender y hacer el análisis de la información se decidió cambiar algunos datos para facilitar su lectura. **Este cambio de datos solamente se realizará para el análisis, por lo que al momento de efectuar las operaciones pertinentes se mantendrá el archivo original sin cambio alguno.**



A continuación, se muestran aspectos estadísticos de los datos numéricos.

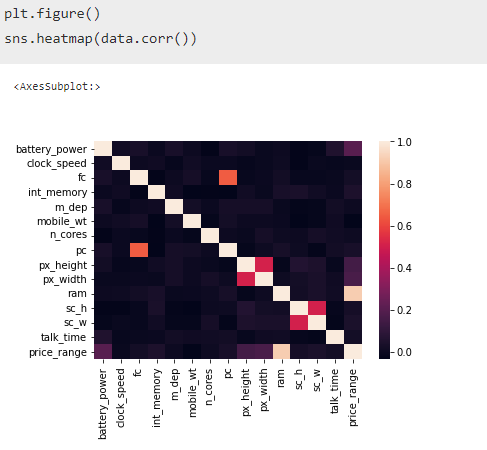
En seguida, se muestran aspectos estadísticos de los datos categóricos.



**Matriz de Correlaciones**

Una matriz de correlación es una tabla que contiene coeficientes de correlación entre variables. Cada celda de la tabla representa la correlación entre dos variables, el valor se encuentra entre -1 y 1. Se utiliza una matriz de correlación para resumir los datos,

como diagnóstico para análisis avanzados y como entrada para un análisis más avanzado.



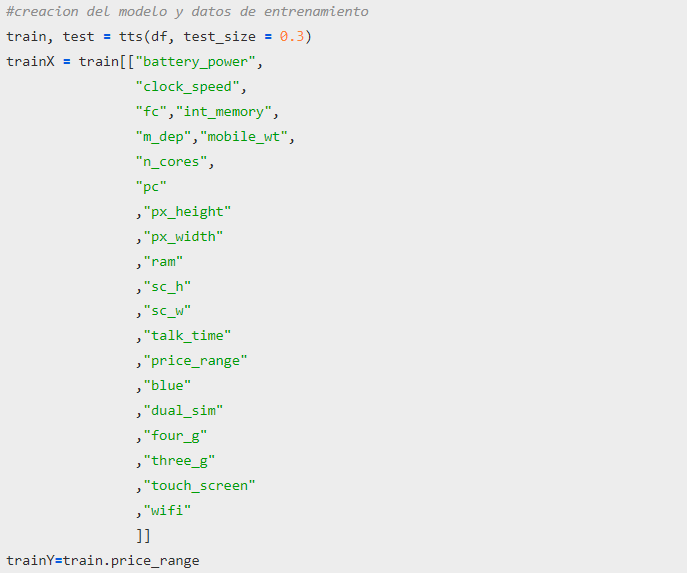
En la Matriz de Correlaciones podemos observar como las columnas sc\_h (screen height) y sc\_w (screen weight), tienen una mayor relevancia en los datos, lo cual en un análisis en la vida real, tanto **el ancho como el alto del móvil tienen una gran importancia para el precio de este.**

**Redes Neuronales**

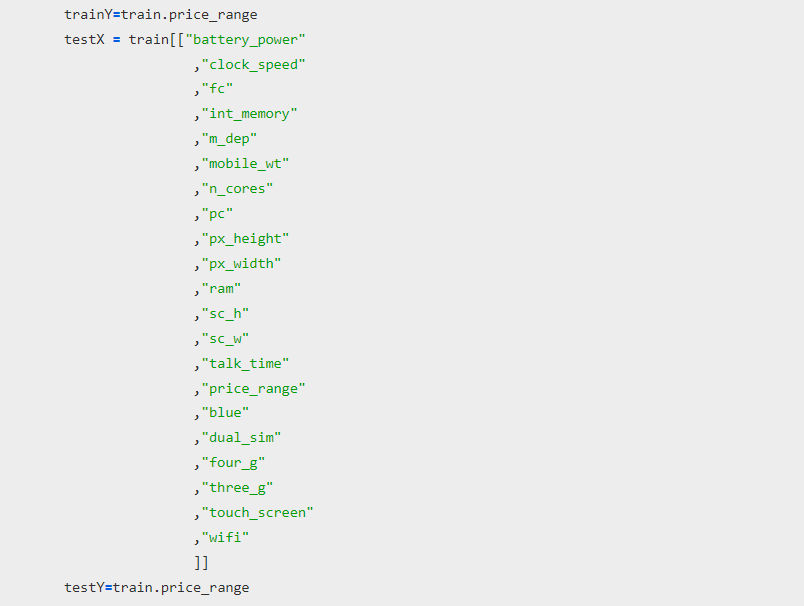
A continuación se aplican las técnicas de Redes Neuronales para poder predecir price\_range . Como se mencionó al inicio del presente trabajo, se volverá a cargar la Base de Datos para poder aplicar las operaciones pertinentes.



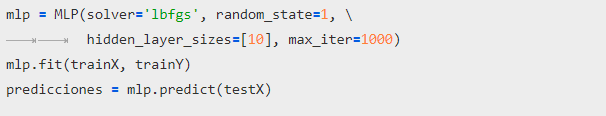
Posteriormente, ya con la Base de Datos cargada en el DataFrame , se empieza a crear el modelo y los datos de entrenamiento. Empezamos con los datos de entrenamiento para X y Y:



Continuamos con los datos de práctica o test para X y Y:



Creamos el modelo y entrenamos a la Red Neuronal para su predicción.



En seguida, realizaremos un análisis del algoritmo de clasificación creado (Red Neuronal) para medir su rendimiento. **Se entiende que es un problema de clasificación multiclase, ya que se tienen varias opciones para definir el precio del móvil: 0,1,2 y 3**

**Evaluación del modelo de Red Neuronal**

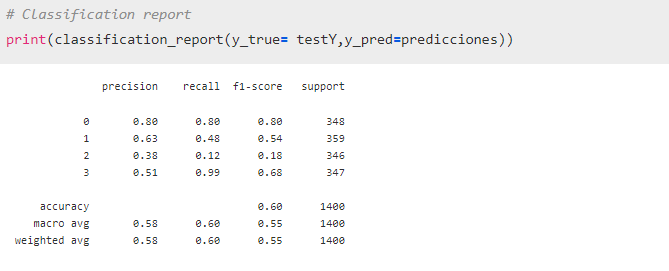
A continuación, se muestran varias métricas para la evaluación del algoritmo, usaremos los siguientes:

Accuracy. - Representa la proporción del número de predicciones correctas entre el número total de predicciones.

Precisión. - Compromisos hechos en la clasificación, indican lo interesantes y relevantes que son los resultados de un modelo. Indica la proporción de ejemplos que son **verdaderamente positivos** (PPV)

Recall. - Indica que tan completos son los resultados

F1 score. - Es una métrica que combina precisión y recall utilizando la media.

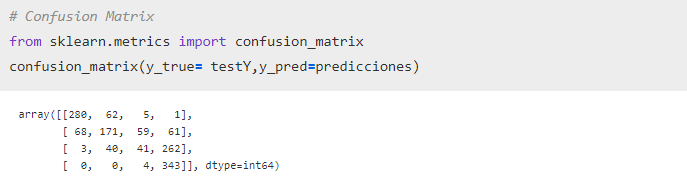
****

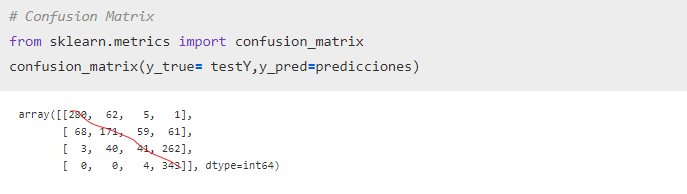
Se puede observar que se tiene una precisión baja, siendo la más alta de .80 y de ahí desciende considerablemente a .38, teniendo un promedio de .58. Es decir, cuando el modelo predice la clase positiva ¿Cuántas veces está en lo cierto? nos indica que **los verdaderos positivos del modelo son bajos**.

Una de las métricas más importantes para poder leer el rendimiento de un modelo de clasificación es F1, ya que como se había explicado anteriormente, es la media entre la precisión y el recall, simplificando el rendimiento a una única métrica. Al igual que Precisión, podemos observar la tendencia a la baja mientras más se hace un Score.

**Matriz de confusión del modelo Red Neuronal**

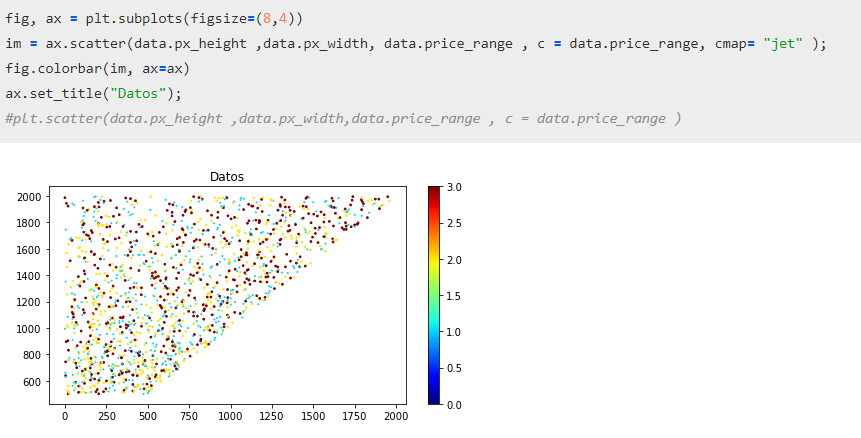
Es una tabla que organiza las predicciones en función de los valores reales de los datos. La categoría de interés se conoce como **clase positiva**, mientras las otras son **clase negativa**. Estas instancias caen en la diagonal de la matriz. Los valores fuera de la diagonal indican las instancias clasificadas incorrectamente, la lectura de esta matriz se basa en cuantas instancias caen dentro y fuera de la diagonal de **True Positive.** A continuación, se muestran 2 matrices de confusión, la primera es la que el programa nos muestra al momento de correr el código, y la segunda muestra la lectura que se hace como **True Positive** de las instancias mostradas.



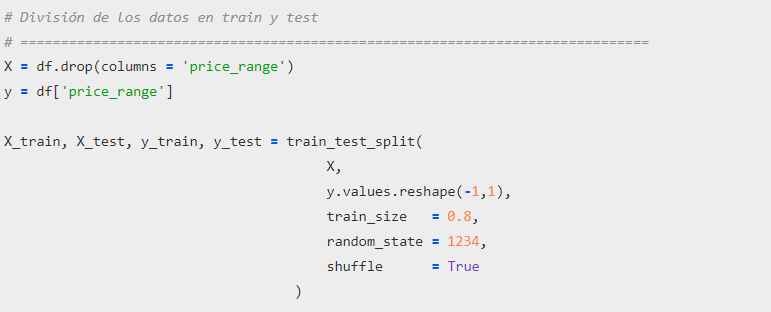


**Máquina de Vectores**

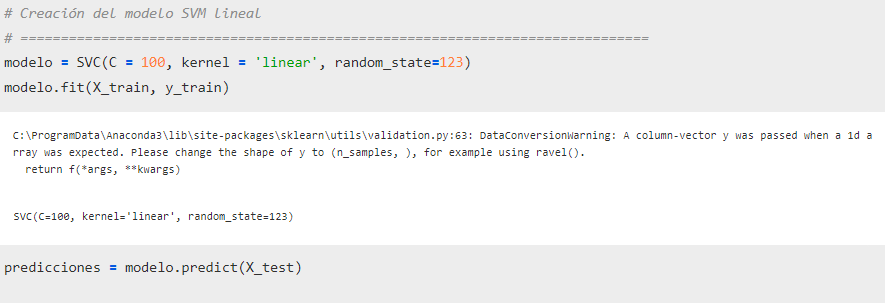
Ahora se aplican las técnicas de Máquina de Vectores para poder predecir price\_range. En la siguiente gráfica, se muestra prince\_range con respecto a las columnas **px\_height y px\_width**, se decidió usar estas 2 columnas ya que, de acuerdo con la matriz de correlaciones, su importancia es relevante para determinar prince\_range.



En seguida, ya con la Base de Datos cargada en el DataFrame, se empieza a crear el modelo y los datos de entrenamiento. Empezamos con los datos de entrenamiento para X y Y.



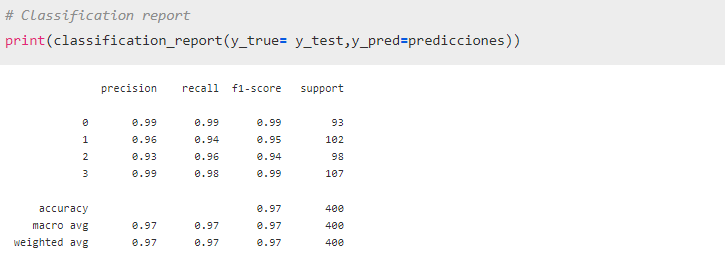
Ahora se empezará a entrenar el modelo con los datos de X y Y , además guardamos las predicciones del modelo en una variable para poder hacer su análisis.



**Evaluación del modelo de Máquina de Vectores.**

A continuación, se muestran varias métricas para la evaluación del algoritmo, usaremos los siguientes métodos, previamente definidos en el documento.

Accuracy, Precisión, Recall y F1 score

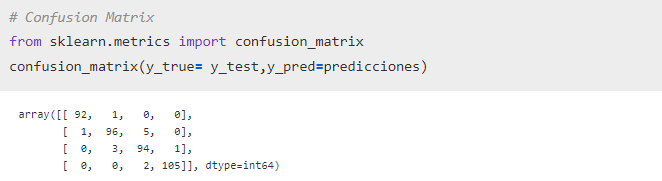


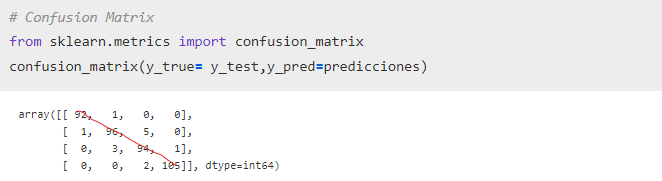
Se puede observar que se tiene una precisión muy alta a diferencia de la Red Neuronal, la más alta fue de .99 teniendo un promedio de **.97**. Es decir, cuando el modelo predice la clase positiva ¿Cuántas veces está en lo cierto? lo cual indica que **los verdaderos positivos del modelo son altos**.

Como se dijo anteriormente en el análisis de la Red Neuronal, F1 es la **media entre la precisión y el recall,** simplifica el rendimiento a una única métrica, podemos observar la tendencia al alta mientras más se hace un Score.

**Matriz de confusión del modelo Máquina de Vectores**

Como se indicó anteriormente en el presente documento, la categoría de interés se conoce como **clase positiva**. A continuación, se muestran 2 matrices de confusión, la primera es la que el programa nos muestra al momento de correr el código, y la segunda muestra la lectura que se hace como **True Positive** de las instancias expuestas. Un detalle importante es que se muestran valores más pequeños que la Red Neuronal, esto es debido a que las métricas de rendimiento son tan altas que la matriz de confusión mide y **detalla los pocos datos a negociar como True Positive y False Negative.**

****



Conclusiones

Se utilizaron 2 modelos diferentes para poder obtener el price\_range de la Base de Datos, Redes Neuronales y Máquina de Vectores. Al momento de realizar la medición de métricas para ambos modelos, podemos observar que la **Máquina de Vectores nos da una alta fidelidad para los resultados**. **Cabe resaltar que para el modelo de Red Neuronal solamente se uso 1 capa,** es posible que usando diferentes Layers y neuronas podamos obtener un resultado mucho más preciso.

**Bibliografía**

\*https://www.datasource.ai/es/data-science-articles/comprension-de-la-matriz-de-confusion-y-como-implementarla-en-python

\*https://empresas.blogthinkbig.com/como-interpretar-la-matriz-de-confusion-ejemplo-practico/