**Documentación proyecto IA**

**Integrantes:**

* Ralph Castellanos Couott
* Oscar Iván Vera Puerto

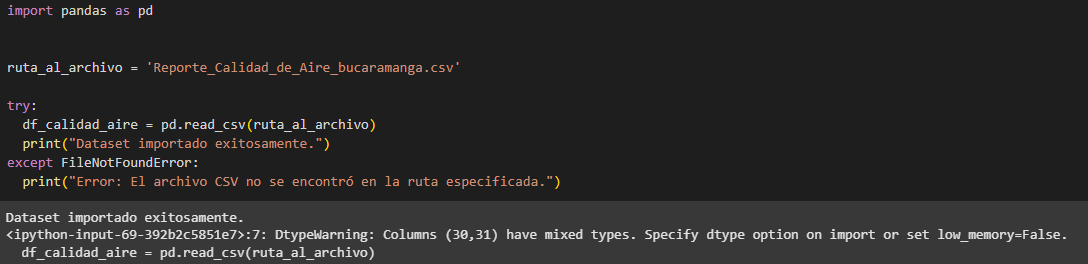
Para la realización del modelo y aplicación sobre la predicción acerca de la calidad del aire en Bucaramanga, se realizó una investigación previa que consistió en la búsqueda sobre cómo se puede calcular la calidad del aire en una zona y se buscaron los datos de Bucaramanga, provenientes de una fuente confiable, respecto a la calidad del aire en el lugar.

Para poder calcular la calidad del aire, se utilizó la información que la empresa *IQAir* proporciona en su página web sobre el tema: “El AQI se calcula en base a las mediciones de los principales contaminantes del aire, incluyendo la materia particulada (), ozono troposférico (), monóxido de carbono (), dióxido de azufre () y dióxido de nitrógeno ()” (*IQAir | First In Air Quality*, s. f.).

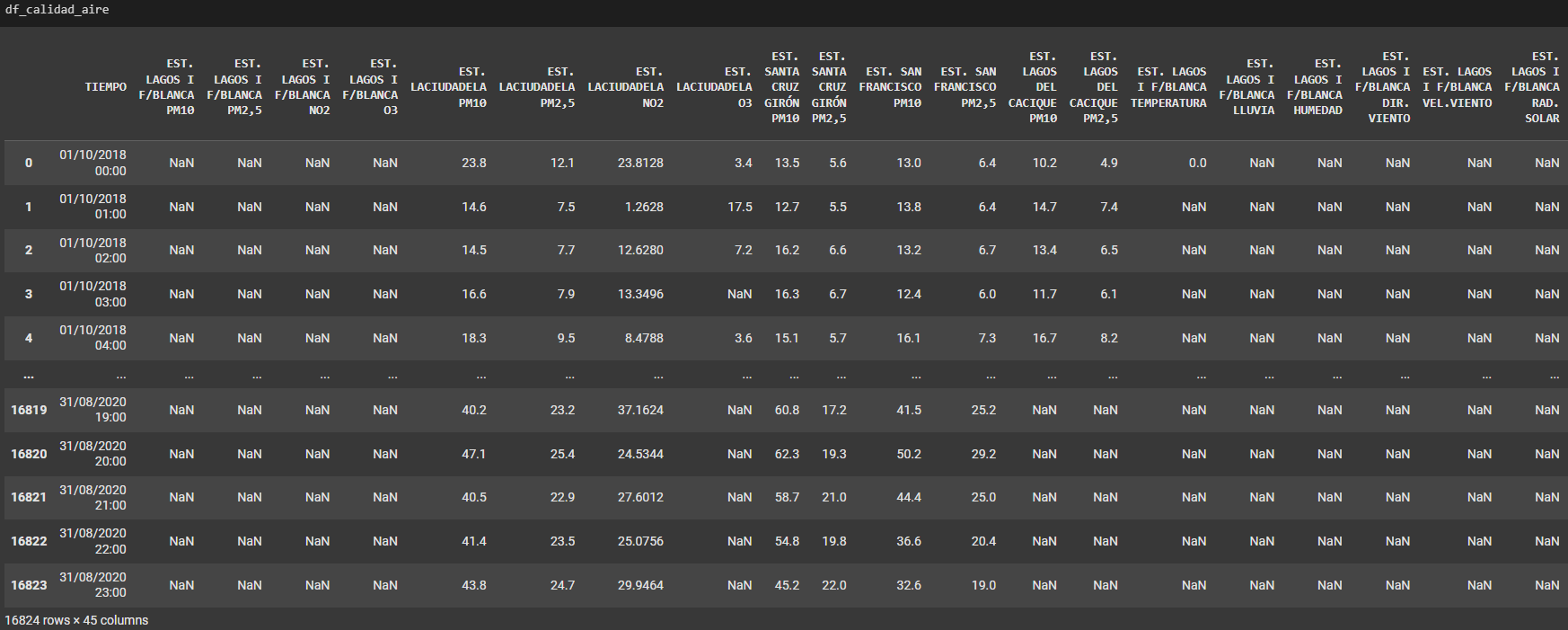
Una vez encontrada la información acerca del entendimiento sobre qué es y cómo se calcula el AQI, se buscaron los datos respectivos al tema en la zona de Bucaramanga. Dichos datos se encontraron en la página web *Datos Abiertos de Colombia*, sitio oficial del gobierno colombiano en el que se comparten datos abiertos de diversos temas, y el que nos interesaba para este proyecto, era el referente sobre la calidad del aire en la zona metropolitana de Bucaramanga.

Una vez encontrado los datos, se analizó la información contenida para ver si nos servía teniendo en cuenta lo investigado sobre el cálculo de la calidad del aire según IQAir, en esta ocasión, los datos suministrados concordaban con el contenido procedente del sitio web, por lo tanto, se descargó el archivo CSV que *Datos Abiertos de Colombia* nos proporcionaba para estar listos con la inicialización del preprocesamiento de nuestro modelo.

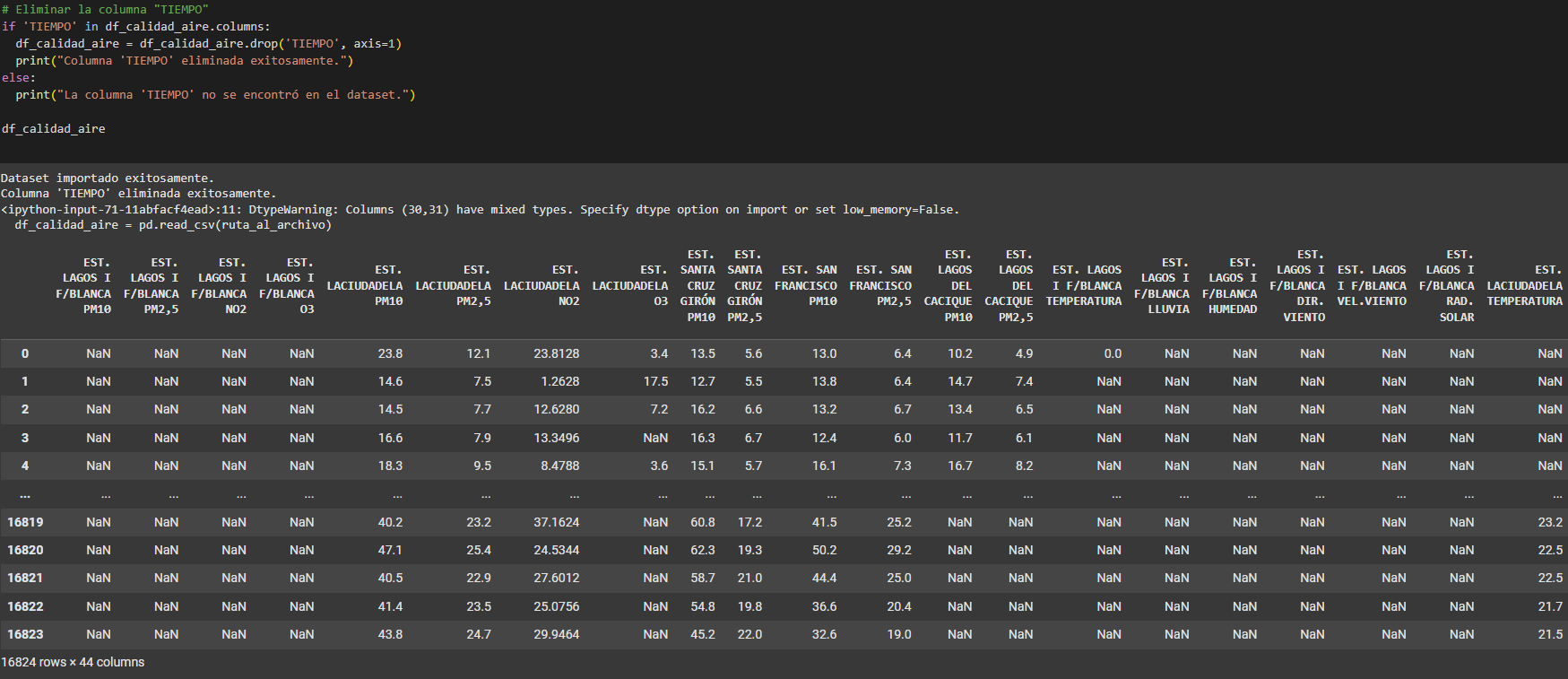
Se abrió un cuaderno en *Google Collab* para hacer el preprocesamiento del modelo. Primero, se exportó el archivo CSV que contenía los datos que utilizaríamos para entrenar al modelo. Se instalaron las librerías necesarias, en este caso se instaló la librería Pandas para poder realizar ciertas acciones, y se guarda la ubicación del archivo CSV en la variable ‘ruta\_al\_archivo’. Tras esto, se crea un DataFrame ‘df\_calidad\_aire’ que lee los datos suministrados y que muestra un mensaje dependiendo de si tiene éxito o no en la búsqueda del archivo CSV.



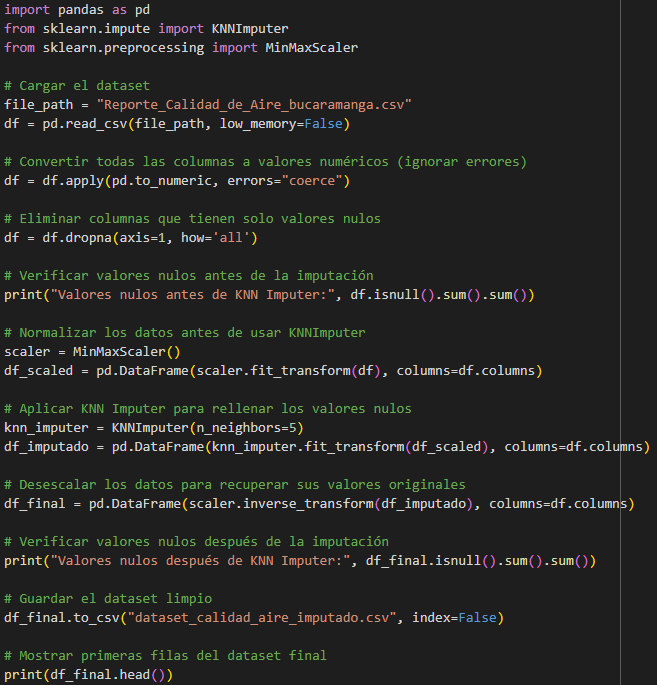
Se llama al DataFrame para corroborar que se vean los datos entregados.

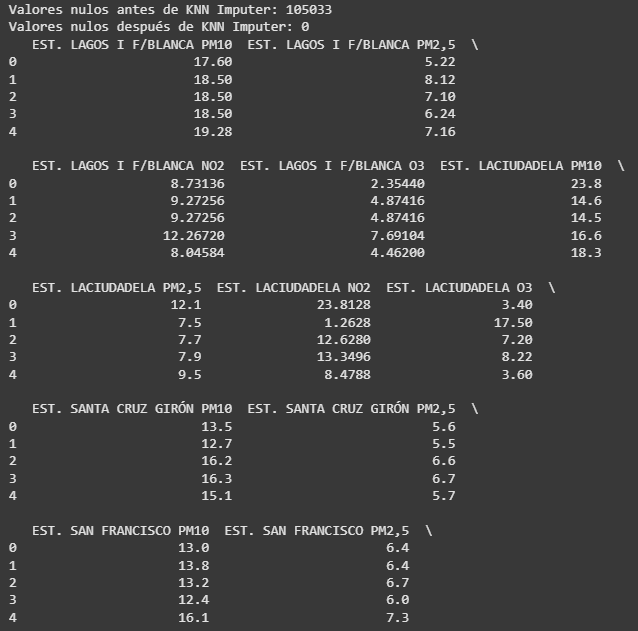


Se mira si es necesario eliminar alguna columna del DataFrame, en caso de que su información no sea relevante para calcular la calidad del aire, y en este caso, nos percatamos de que la columna ‘*TIEMPO*’ no es fundamental para el objetivo del modelo, por lo que se optó por desecharla. Se vuelve a llamar al DataFrame para confirmar la eliminación de dicha columna.

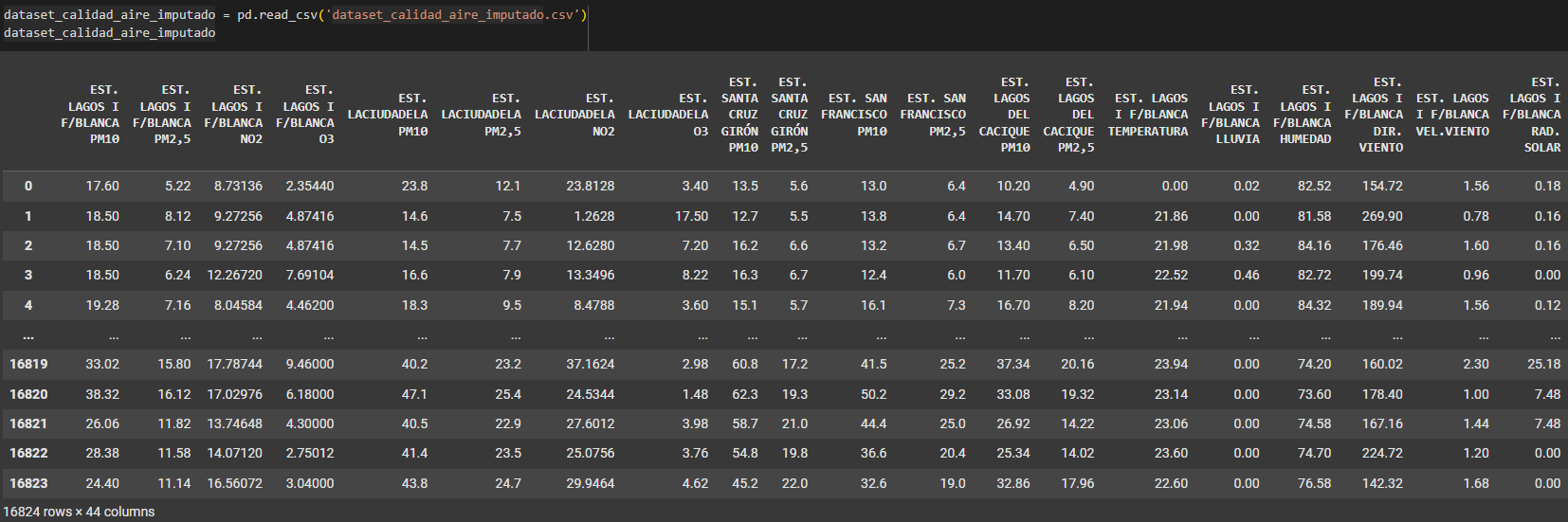


Se importan algunas funciones específicas de la librería *sklearn*. A continuación, se realizan los siguientes pasos: se carga el DataSet sin que tenga errores de memoria, se convierten todas las columnas a valores numéricos, se eliminan aquellas columnas que hayan quedado solo con valores nulos, se cuentan cuántos valores quedaron nulos antes de hacer la imputación, se normalizan o escalan los datos usando la función MinMaxScaler() que se descargó previamente gracias a la librería *sklearn.preprocessing*, se aplica la función *KNNImpute*r para rellenar los valores nulos usando la opción de vecino cercano (en este caso se usan a los 5 vecinos más cercanos), se deshace la normalización para así restaurar los valores originales, se verifican cuántos valores quedaron nulos tras la imputación, se guarda el DataSet limpio y finalmente se muestran las primeras filas del DataSet limpio para mirar cómo quedó tras la normalización de los valores.

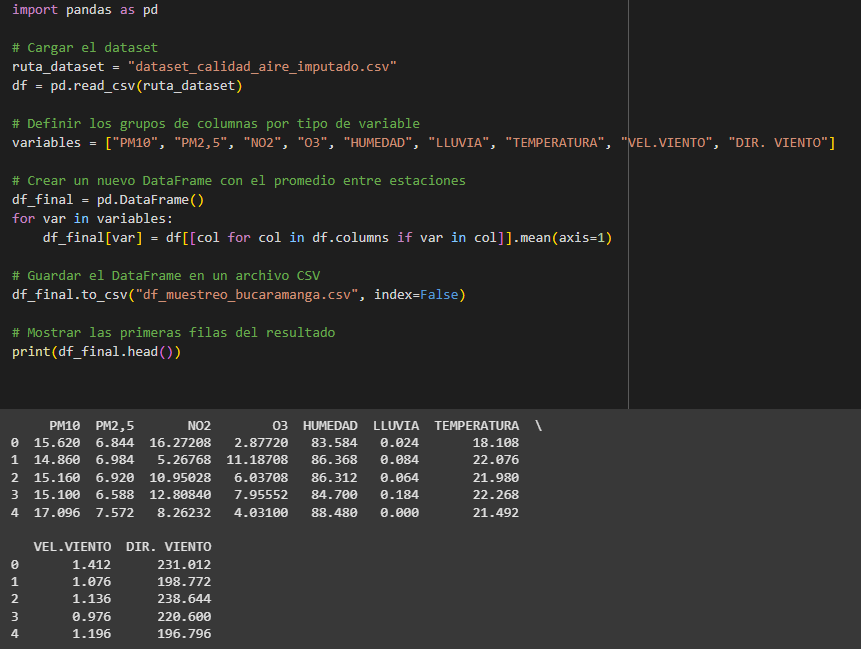




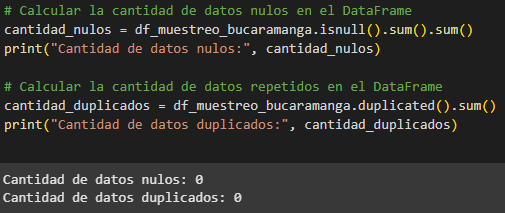
Se crea un DataFrame ‘dataset\_calidad\_aire\_imputado’ que lee al DataSet limpio guardado anteriormente y se lo llama para visualizarlo.



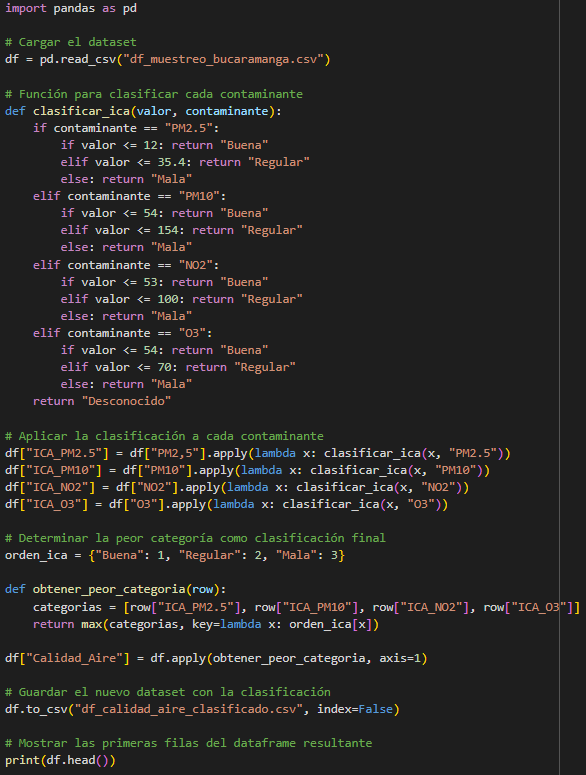
Se crea una lista con los nombres de las variables que queremos analizar (PM10, PM2.5, NO2, etc.) Inicializamos un nuevo DataFrame ‘df\_final’ que contendrá el promedio entre las estaciones, se recorre la lista filtrando todas las columnas del DataFrame original que contengan el nombre de cada variable, se calcula el promedio de las columnas en cada fila y se guarda el DataFrame en un archivo CSV.

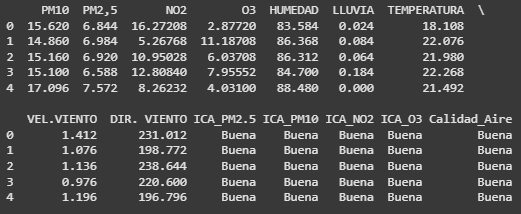


Se calculan la cantidad de datos nulos y repetidos en el DataFrame.

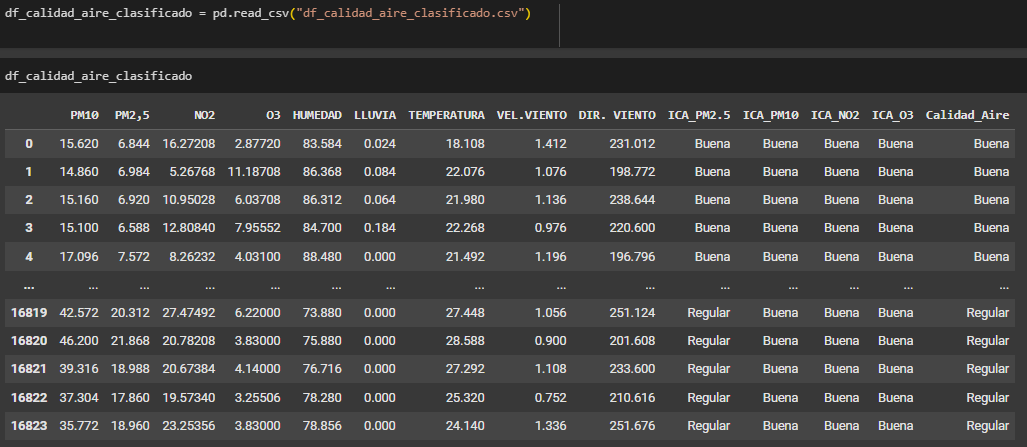


A continuación, se crea una función ‘def clasificar\_ica’ que tendrá como finalidad clasificar cada contaminante, es decir, si es PM10 o PM2.5 y darle una categoría dependiendo del valor que contenga. Se crean cuatro nuevas columnas, una por cada tipo de clasificación, y se le aplica la función mencionada anteriormente para asignarle un valor. Creamos un diccionario ‘orden\_ica’ para otorgar el valor que escojamos a las clasificaciones. Se crea una nueva función ‘def obtener\_peor\_categoria’ cuyo objetivo es obtener las categorías de los contaminantes en una fila y que devuelva la peor clasificación presente entre los contaminantes. Aplicamos la función anterior fila por fila y esto crea una nueva columna en el DataFrame. Finalmente, se guarda el DataFrame con clasificación en un archivo CSV.

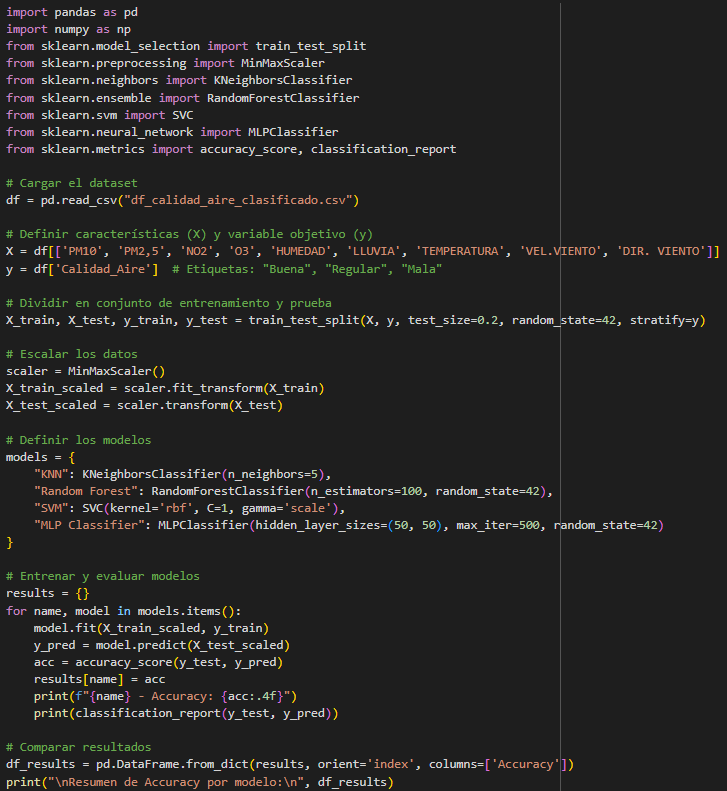


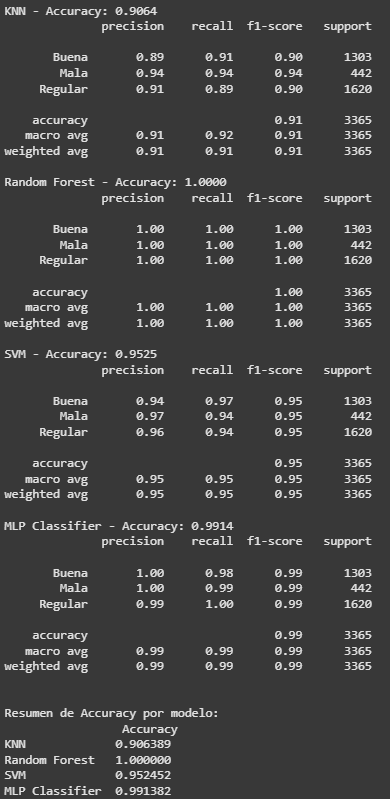


Creamos un nuevo DataFrame ‘df\_calidad\_aire\_clasificado’ que carga el archivo CSV previo y lo llamamos para previsualizarlo.

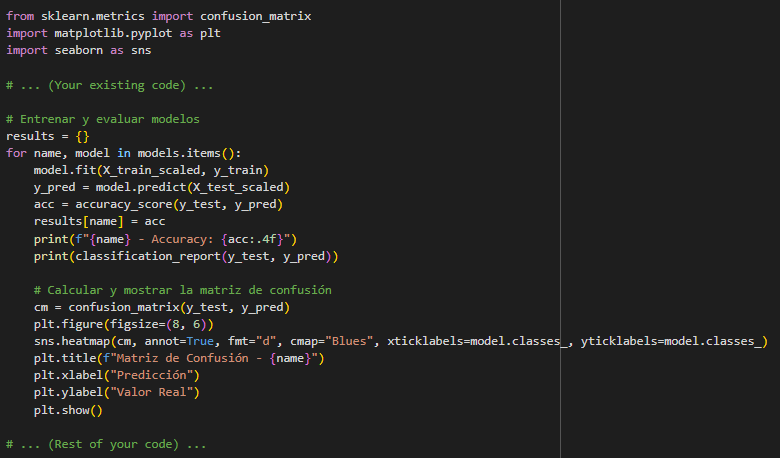


A continuación, vamos a hacer los modelos de entrenamiento. Cargamos el archivo CSV que contiene la clasificación del AQI en un DataFrame ‘df’. Definimos los Labels (X) y la etiqueta (y). Dividimos el conjunto de los datos en dos: uno de entrenamiento y uno de prueba, siendo el 80% para entrenamiento y el porcentaje restante para la prueba. Escalamos los datos de los Labels con la función de escalado mínimo y máximo. Definimos los modelos. Entrenamos y evaluamos los modelos, para al final mirar la exactitud de cada uno de ellos.





Calculamos y mostramos la matriz de confusión de todos los modelos, para así tener un apoyo visual sobre la exactitud de cada uno de ellos.

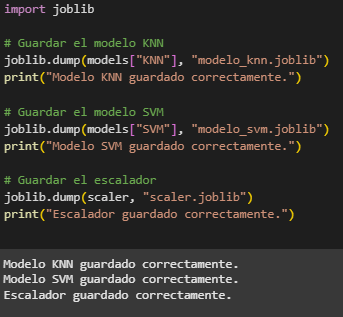






Se miró la exactitud de cada uno de ellos y nos fijamos que el modelo del RandomForest fue del 100%, lo que significa que quedó con mucha exactitud, lo que podría indicar que estuviera sobre ajustado, por lo tanto, nos decidimos por los modelos KNN y SVM, debido a que sus valores fueron más realistas y son menos propensos al sobre ajuste.

Se guardaron los modelos seleccionados (KNN y SVM) en archivos joblib, y también el escalador.



**Referencias bibliográficas**

* *IQAir | First in Air Quality*. (s. f.). <https://www.iqair.com/es/support/knowledge-base/KA-5110-ES>
* *Reporte Calidad de aire AMB | Datos abiertos Colombia*. (2020, 29 septiembre). <https://www.datos.gov.co/Ambiente-y-Desarrollo-Sostenible/Reporte-Calidad-de-Aire-AMB/58ct-h586/about_data>