ITTI | High Tech Institute

TRABAJO FIN DE MÁSTER

MÁSTER DATA SCIENCE EXECUTIVE 22-23

DASHBOARD CON SRTEAMLIT FRAMEWORK

DE LA PREDICCIÓN DE CLASE DE IRIS

AUTOR: Alberto Rodríguez Gómez

TUTORES: José Manuel Peña (Responsable del TFM)

Isabel Maniega (Colaboradora en el TFM)

**RESUMEN DEL PROYECTO**

Este proyecto tiene como objetivo llevar a cabo una aplicación que permita al usuario, en función de unos parámetros que defina relacionados con las dimensiones del pétalo y el sépalo de una planta, hallar la probabilidad de que sea de un tipo u otro de iris. La aplicación se ha realizado como es requisitado con el framework de Streamlit y se ha desarrollado en el lenguaje de Python. Se ha utilizado como editor del código fuente la plataforma de Visual Studio Code, que además de ofrecer la posibilidad de integrar el lenguaje de desarrollo utilizado, también nos ofrece soporte para integrar otras herramientas que se han utilizado como los cuadernos de Jupyter o la integración con Github.

**ESTRUCTURA DEL DOCUMENTO Y CÓDIGO**

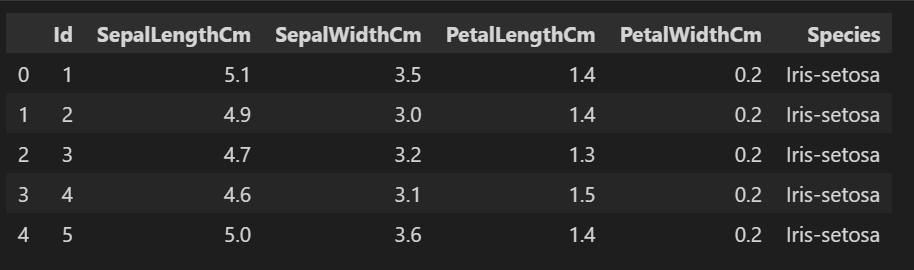
Para describir el funcionamiento de esta aplicación, se va a realizar en dos partes claramente diferenciadas. Por un lado, se va a explicar la generación del modelo que nos permitirá hallar la probabilidad de que, según los valores recogidos posteriormente, pueda ser de un tipo de iris u otro. Y en la segunda parte, se explicará la generación del código necesario sobre el framework de Streamlit para generar la aplicación visual con la que el usuario interactuará.

En la siguiente parte del documento, se realizará una pequeña guía/ejemplo de cómo un usuario podría ingresar los valores en los que se basará el modelo para hallar la probabilidad de pertenencia a un tipo y otro de iris.

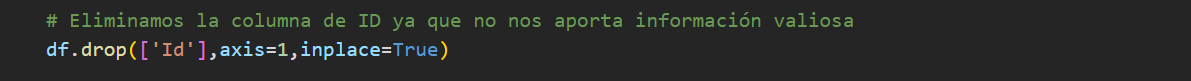
**Generación del modelo de predicción**

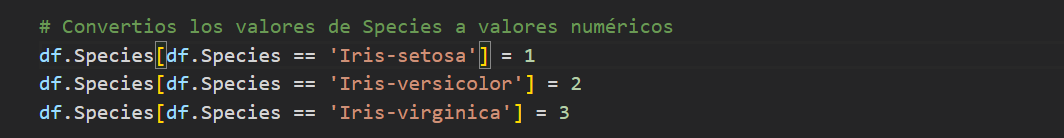
Para hacer la generación del modelo de predicción se ha utilizado el siguiente código sobre un cuaderno de Jupyter. El nombre del archivo dentro del código de la aplicación es: Iris Classifier Model.ipynb. A continuación se procede con la explicación del mismo.

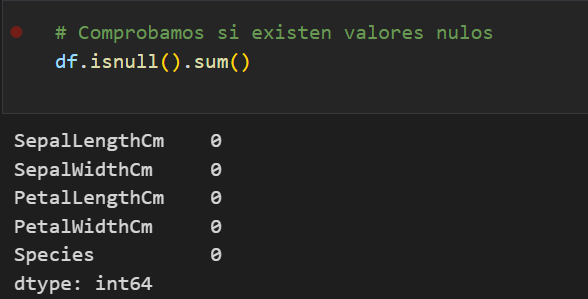
En la primera parte, como es habitual, importamos las librerías necesarias para poder correr el código. Además importamos el dataset Iris.csv y hacemos una visulización de los primeros registros del dataset para tener una idea más clara de la estructura del mismo.



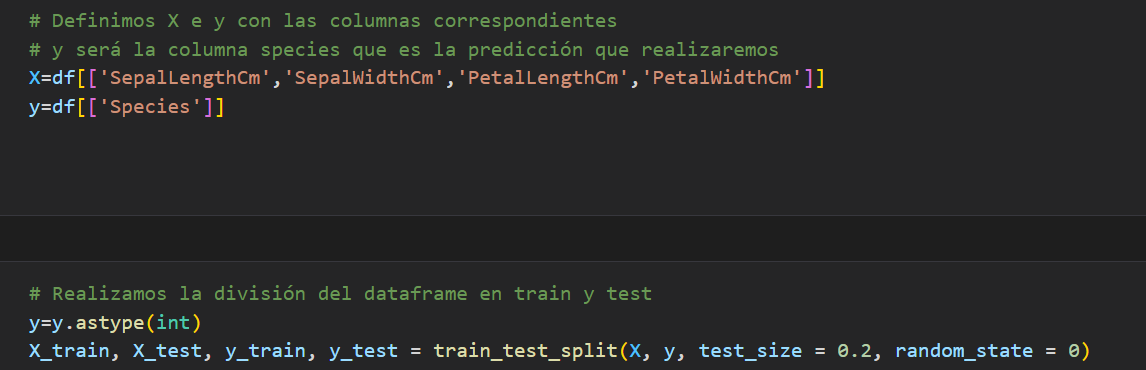
En la siguiente parte, lo que se hace es analizar, modificar y limpiar el dataset, de modo que el contenido que quede sea lo más valioso posible para poder hacer el entrenamiento y la creación del modelo posterior. En este caso, los pasos fundamentales realizados han sido, eliminar la columna de Id, convertir los tipos de especies a valores númericos (setosa=1, versicolor=2 y virginica=3) y comprobar que no existen valores nulos en el dataset.



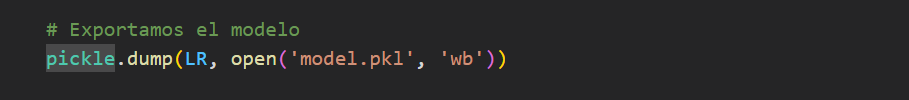




La siguiente parte, es la encargada de dividir el dataset en X e y para hacer el entrenamiento y el test del modelo.



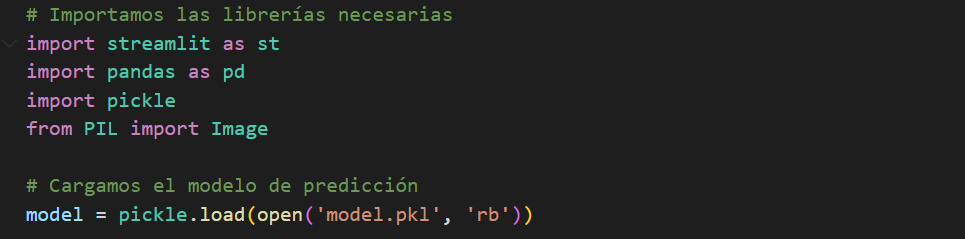
En la última parte, vamos a definir el algoritmo a utilizar, entrenar y calcular la precisión del modelo y finalmente se exporta el modelo que se utilizará en la segunda parte de la aplicación.



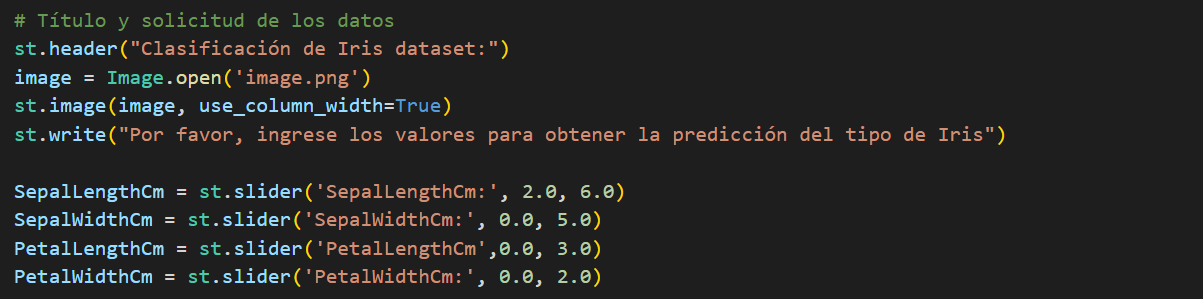
**Creación de la aplicación sobre el framework de Streamlit**

El nombre del archivo sobre el que se ha creado la aplicación en sí, es app.py. Está desarrollado sobre Python, y a continuación se explica brevemente su contenido.

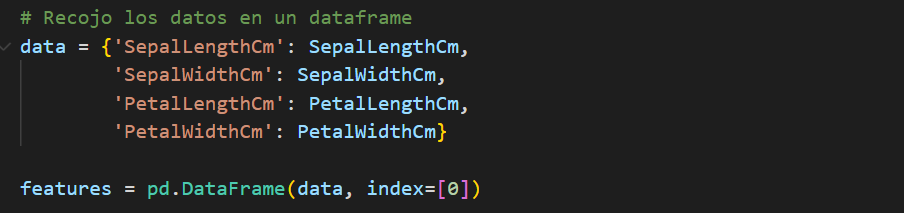
Como es habitual, las primeras líneas continenen el código necesario para la importación de las librerías necesarias. A continuación se hace la importación del modelo creado en la anterior parte que será, parte fundamental para el calculo de las probabilidades de que sea de un tipo u otro de iris.



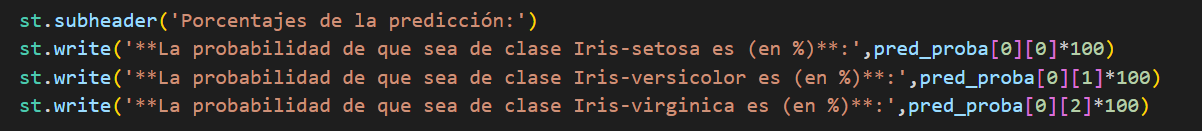
La siguiente parte, es la parte gráfica que se pintará al usuario. En concreto, se define la cabecera de la misma, se carga una imagen decorativa, y se le solicitan al usuario los valores con barras de desplazamiento con valores de inicio y fin predefinidos.



El código que viene a continuación se utiliza para recoger los valores que ha definido el usuario en un dataframe y utilizar este dataframe como argumento para el cálculo de la predicción utilizando el model anteriormente definido.

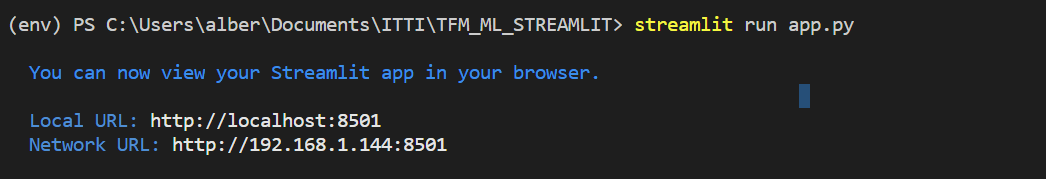


Por último, se devuelve el cálculo realizado al usuario, mostrando en pantalla, la probabilidad que hay de que en función de los parámetros seleccionados, el iris sea de un tipo o de otro.

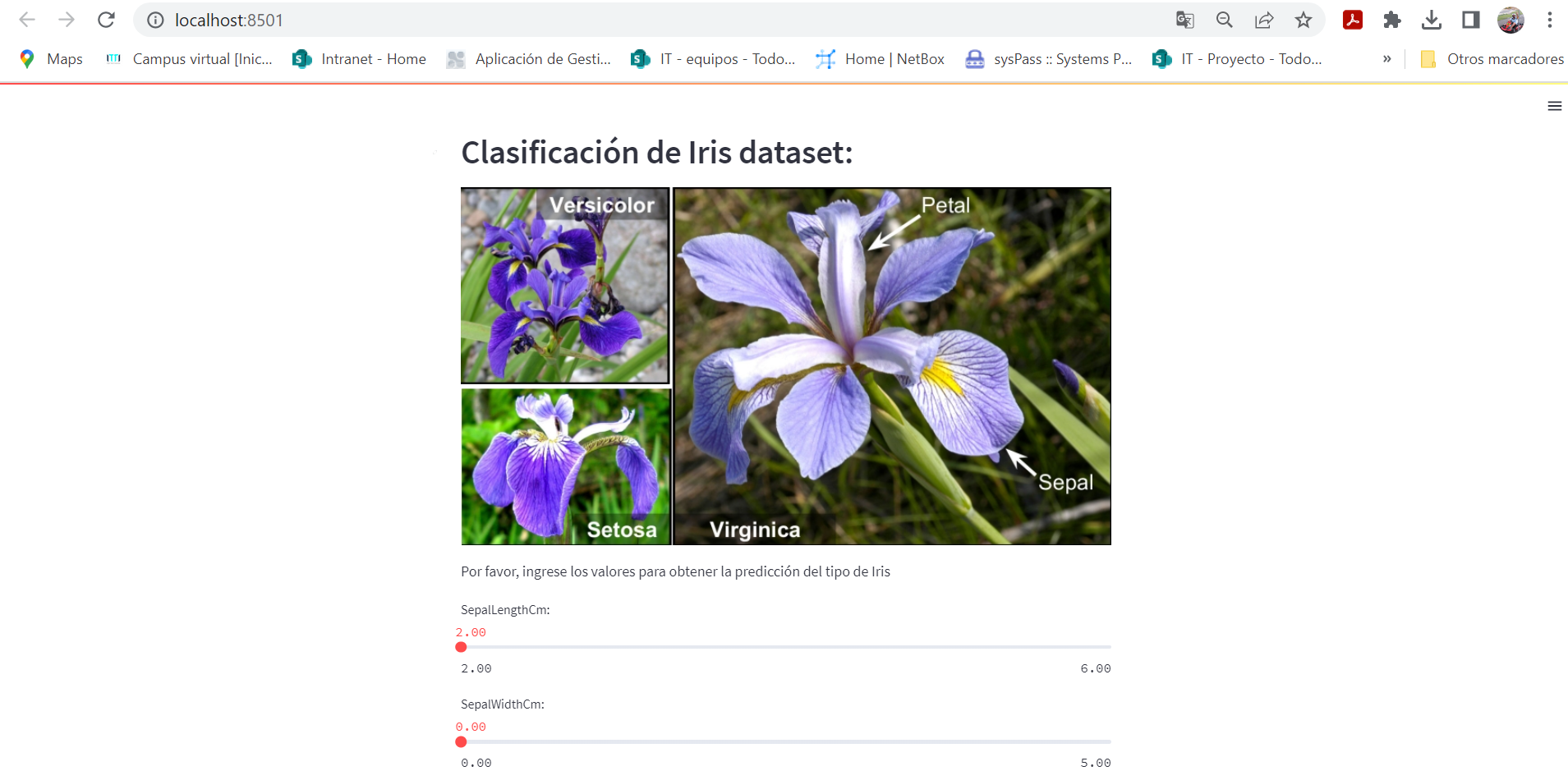


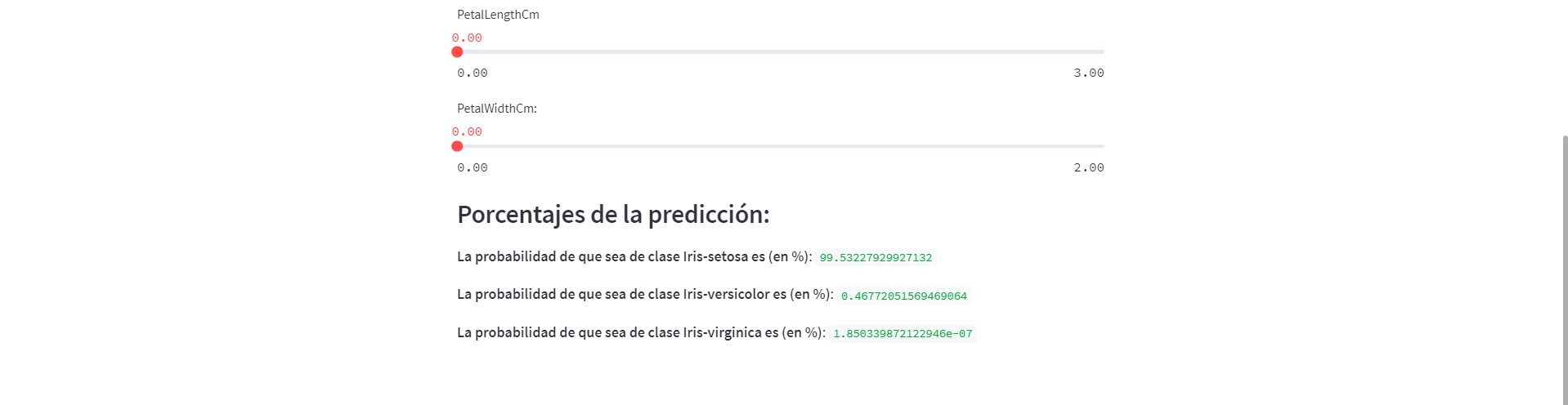
**EJECUCIÓN DE LA APLICACIÓN Y EJEMPLO DE CÁLCULO**

Para la ejecución de la aplicación, haremos una simple llamada al código anterior desde la terminal. En este caso, se ha utilizado la terminal integrada en el programa de Visual Studio Code. La llamada se realiza del siguiente modo (ejecución desde el directorio que aloja la apliacación en sí):



A continuación, accederemos mediante un navegador WEB a la URL local que nos ha mostrado la llamada de la aplicación. De este modo, visualizamos la aplicación que se ha definido. En la siguiente imagen, se muestra el contenido de la misma, en la que como hemos dicho vemos claramente diferenciados, el título, la imagen, la solicitud de los datos, y el calculo de la probabilidad en función de los datos recogidos.





Mediante el uso de los sliders, se puede definir los parámetros que le llegarán a la aplicación para hallar la probabilidad de que sea un tipo de iris u otro. En la anterior imagen, se ven los valores que se mostrarán por defecto. En la siguiente, se muestran los porcentajes de predicción para los siguientes valores de ejemplo: longitud del sépalo 2,8cm, anchura del sépalo 1,5cm, longitud del pétalo 1,5cm y anchura del pétalo 0,3cm.



Como se puede ver, la predicción nos muestra que con los valores mencionados, la probabilidad de que sea del tipo iris setosa, es del 95,61%, de que sea del tipo versicolor es de 4,38% y la probabilidad de que sea virginica es casi nula.

**ALOJAMIENTO DEL CÓDIGO DE LA APLICACIÓN EN GITHUB**

Como se ha mencionado ya en el documento, desde la aplicación de Visual Studio Code, podemos integrar el desarrollo realizado con la herramienta de Github. Para concluir el documento, se expone la URL en la que se ha dejado la aplicación subida: https://github.com/arodgom/TFM\_ML\_STREAMLIT/