

TFM

PYTHON



JosÉ cabezas

Agradecimientos

Contenido

[Abstract 1](#_Toc138151959)

[Introducción 2](#_Toc138151960)

[Objetivos generales del trabajo 2](#_Toc138151961)

[Justifica la elección del tema. 2](#_Toc138151962)

[Resumen metodología 3](#_Toc138151963)

[Estructura 4](#_Toc138151964)

[Planteamiento del problema 4](#_Toc138151965)

[Objetivos del trabajo 5](#_Toc138151966)

[Metodología 6](#_Toc138151967)

[Evaluación de los resultados (¿o quiere decir que evaluemos que variables hacen que existan mayor riesgo de accidentes? , se podría usar la matriz de coorelación usada cuando se generó el modelo) 6](#_Toc138151968)

[Conclusiones 7](#_Toc138151969)

[Referencias 8](#_Toc138151970)

[Anexos TFM I](#_Toc138151971)

# Abstract

Our Master's thesis focuses on developing a Python-based system for predicting airplane accidents. To achieve this, we utilized the Streamlit library and worked with a preprocessed dataset of airplane accidents. Additionally, we employed the PyCaret library for the prediction process.

Firstly, we dedicated time to preprocess the airplane accident dataset. This involved tasks such as data cleaning, outlier removal, and variable transformation. Through this process, we obtained a refined dataset ready for use in our prediction model.

Next, we utilized the PyCaret library to develop a machine learning model capable of predicting the occurrence of airplane accidents. PyCaret allowed us to explore various classification algorithms and identify the most suitable one for our dataset. Furthermore, it greatly simplified the model training and evaluation process.

Once the prediction model was generated, we utilized the Streamlit library to create an interactive interface. We designed a form where users could input relevant data for making predictions, including the date, airplane type, and airline, among others. Streamlit facilitated the creation of this user interface by enabling quick design and deployment of interactive web applications.

In summary, our group's Master's thesis focused on developing a Python-based airplane accident prediction system. We employed Streamlit to create a user input form and PyCaret for training and evaluating the prediction model. Our work involved preprocessing the airplane accident dataset, selecting the most suitable classification algorithm, and creating an interactive interface for users. With this system, we aim to contribute to enhancing safety in the aviation industry and preventing potential accidents.

# Introducción

## Objetivos generales del trabajo

## El objetivo general de nuestro proyecto de Fin de Máster (TFM) es desarrollar un sistema de predicción de accidentes aéreos utilizando programación Python. Para lograr esto, utilizamos las bibliotecas Streamlit y PyCaret junto con un conjunto de datos de accidentes aéreos preprocesados.

## Nuestro objetivo principal es ayudar a mejorar la seguridad en la industria de la aviación proporcionando herramientas para predecir accidentes de aeronaves. La capacidad de predecir posibles accidentes puede facilitar la toma de decisiones y la implementación de medidas preventivas por parte de las autoridades, las aerolíneas y otros actores de la industria. Además, nos hemos fijado el objetivo de desarrollar un sistema eficiente y fácil de usar que utilice la programación Python y las bibliotecas Streamlit y PyCaret. Queremos proporcionar una interfaz intuitiva que permita a los usuarios ingresar datos relevantes y predecir con precisión la probabilidad de un accidente aéreo. En resumen, nuestro trabajo se centra en el desarrollo de sistemas de predicción de accidentes de aviación para mejorar la seguridad en la industria aeronáutica. Utilizando la programación de Python, las bibliotecas Streamlit y PyCaret y los conjuntos de datos preprocesados, nuestro objetivo era proporcionar una herramienta eficiente y fácil de usar para predecir la probabilidad de accidentes aéreos.

## Justifica la elección del tema.

La elección del tema para nuestro trabajo de Fin de Máster (TFM) se basa en diversas justificaciones fundamentales. La seguridad en la industria de la aviación es un tema de vital importancia, tanto para la protección de vidas humanas como para la economía global. Los accidentes de aviones tienen consecuencias devastadoras en términos de pérdida de vidas, daños materiales y repercusiones legales y financieras. Por lo tanto, desarrollar un sistema de predicción de accidentes de aviones se presenta como una medida crucial para prevenir y reducir los riesgos asociados, mejorando así la seguridad en este sector crítico.

Además, los avances tecnológicos en el procesamiento de datos y las técnicas de aprendizaje automático han brindado nuevas oportunidades para abordar problemas complejos como la predicción de accidentes de aviones. Al aprovechar el potencial de la programación en Python y las bibliotecas especializadas como Streamlit y PyCaret, podemos desarrollar un sistema eficiente y preciso.

La elección de utilizar un conjunto de datos de accidentes de aviones previamente recopilados también se justifica por la disponibilidad de información valiosa y la reducción de costos y tiempo asociados con la recopilación de nuevos datos. Al analizar patrones y factores de riesgo en estos datos históricos, podemos generar modelos predictivos confiables que contribuyan a una mejor comprensión de los accidentes de aviones y a la implementación de medidas preventivas más efectivas.

Por último, la creación de un sistema de predicción de accidentes de aviones ofrece una herramienta valiosa para los responsables de la toma de decisiones en la industria de la aviación, como las autoridades de regulación, las compañías aéreas y los investigadores de seguridad. Al proporcionar información precisa sobre la probabilidad de ocurrencia de accidentes, se pueden implementar estrategias de seguridad más efectivas y tomar medidas preventivas proactivas.

En resumen, la elección del tema de nuestro TFM se fundamenta en la importancia de mejorar la seguridad en la industria de la aviación, los avances tecnológicos en el aprendizaje automático, la utilización de datos existentes y la contribución a la toma de decisiones en este sector crítico.

## Resumen metodología

En nuestro trabajo de Fin de Máster (TFM), realizado en equipo por tres personas, hemos utilizado una metodología colaborativa para desarrollar un sistema de predicción de accidentes de aviones. Cada miembro del equipo se ha enfocado en construir un modelo de predicción utilizando diferentes agrupaciones de columnas de datos. Uno de nosotros se ha encargado de los datos de la tripulación, otro de los datos de la aeronave y el tercero de los datos climatológicos.

En primer lugar, cada miembro del equipo ha trabajado de manera individual en la construcción de su propio modelo de predicción. Utilizando las agrupaciones de columnas asignadas, hemos aplicado algoritmos y técnicas de aprendizaje automático para entrenar modelos capaces de predecir la probabilidad de que ocurra un accidente de avión.

A continuación, hemos utilizado la biblioteca Streamlit para crear páginas individuales donde cada miembro del equipo ha desarrollado un formulario para introducir los datos específicos de su modelo. Estas páginas permiten a los usuarios ingresar información relacionada con la tripulación, la aeronave y los datos climatológicos, respectivamente.

Una vez que se han ingresado los datos en cada una de las páginas, hemos utilizado una página final donde hemos combinado y analizado los resultados obtenidos por cada modelo de forma conjunta. En esta página final, interpretamos los resultados de cada algoritmo y los presentamos de manera global. Si varios modelos indican la posibilidad de un accidente, consideramos que existe una mayor probabilidad de que dicho accidente ocurra.

Esta metodología colaborativa nos ha permitido aprovechar la experiencia y conocimientos individuales de cada miembro del equipo en diferentes aspectos del problema de predicción de accidentes de aviones. Al combinar nuestros esfuerzos, hemos logrado desarrollar un sistema más completo y robusto, que abarca aspectos cruciales como la tripulación, la aeronave y los datos climatológicos.

## Estructura

La estructura utilizada en nuestro trabajo de Fin de Máster (TFM) se basó en una metodología colaborativa entre tres personas. A continuación, describiré brevemente la estructura adoptada:

-División de responsabilidades: Cada miembro del equipo asumió la responsabilidad de trabajar en una parte específica del proyecto. Uno se enfocó en los datos de la tripulación, otro en los datos de la aeronave y el tercero en los datos climatológicos. Esta división permitió un enfoque más especializado y eficiente en cada área.

-Construcción de modelos individuales: Cada miembro del equipo trabajó de manera individual en la construcción de un modelo de predicción utilizando las agrupaciones de columnas asignadas. Utilizamos algoritmos y técnicas de aprendizaje automático para entrenar y ajustar estos modelos, con el objetivo de predecir la probabilidad de accidentes de aviones.

- Desarrollo de páginas individuales en Streamlit: Utilizamos la biblioteca Streamlit para crear páginas individuales en las que cada miembro del equipo desarrolló un formulario para introducir los datos específicos de su modelo. Estas páginas permitieron a los usuarios ingresar información relacionada con la tripulación, la aeronave y los datos climatológicos, respectivamente.

-Página final de análisis global: Además de las páginas individuales, creamos una página final en Streamlit donde combinamos y analizamos los resultados obtenidos por cada modelo de forma conjunta. En esta página, interpretamos los resultados de cada algoritmo y presentamos una visión global de la probabilidad de ocurrencia de accidentes de aviones, considerando la información proporcionada por todos los modelos.

Esta estructura nos permitió aprovechar la experiencia y conocimientos individuales de cada miembro del equipo, mientras trabajábamos de manera colaborativa para desarrollar un sistema completo de predicción de accidentes de aviones.

# Planteamiento del problema

En el planteamiento del problema de nuestro Trabajo de Fin de Máster (TFM), nos enfrentamos al desafío de mejorar la seguridad en la industria de la aviación mediante la predicción de accidentes de aviones. Los accidentes de aviones tienen un impacto significativo en términos de pérdida de vidas humanas, daños materiales y consecuencias económicas. Por lo tanto, es crucial desarrollar herramientas que permitan predecir la ocurrencia de estos eventos y tomar medidas preventivas para evitarlos.

Aunque se han implementado múltiples medidas de seguridad y protocolos en la industria de la aviación, los accidentes todavía ocurren y es necesario abordar este problema de manera proactiva. Actualmente, la toma de decisiones en la gestión de la seguridad aérea se basa en datos históricos y en análisis manuales, lo que puede ser limitante y no proporcionar una visión holística de los riesgos.

El objetivo principal de nuestro TFM ha sido desarrollar un sistema de predicción de accidentes de aviones utilizando técnicas de aprendizaje automático y datos relevantes. La idea es aprovechar la disponibilidad de conjuntos de datos de accidentes de aviones previamente recopilados y aplicar algoritmos de aprendizaje automático para construir un modelo predictivo.

Este modelo nos permitirá evaluar la probabilidad de que ocurra un accidente de avión en función de diversos factores, como información de la tripulación, datos de la aeronave y condiciones climatológicas. Al predecir la probabilidad de accidentes, se podrían tomar medidas proactivas para mejorar la seguridad, como ajustar procedimientos, implementar cambios en el diseño de las aeronaves o fortalecer la formación y capacitación de la tripulación.

En resumen, el planteamiento del problema de nuestro TFM se centra en la necesidad de mejorar la seguridad en la industria de la aviación mediante la predicción de accidentes de aviones. Al utilizar técnicas de aprendizaje automático y datos relevantes, buscamos desarrollar un sistema predictivo que proporcione información valiosa para la toma de decisiones y la implementación de medidas preventivas, contribuyendo así a reducir los riesgos y garantizar la seguridad en el ámbito aeronáutico.

# Objetivos del trabajo

El objetivo general de nuestro trabajo consiste en desarrollar un sistema de predicción de accidentes de aviones utilizando técnicas de aprendizaje automático y datos relevantes. Esta iniciativa tiene como finalidad mejorar la seguridad en la industria de la aviación, abordando de manera proactiva la prevención de accidentes y sus consecuencias.

Para lograr este objetivo general, hemos establecido varios objetivos secundarios que se derivan del principal y que justifican y permiten su desarrollo. En primer lugar, nos planteamos realizar un análisis exploratorio exhaustivo de los datos de accidentes de aviones disponibles. A través de este análisis, buscamos comprender la estructura de los datos, identificar patrones significativos, detectar valores atípicos y establecer posibles correlaciones entre las variables.

El segundo objetivo se centra en el preprocesamiento de los datos. Aquí, nos enfocamos en llevar a cabo tareas de limpieza, manejo de valores faltantes, normalización de variables y transformaciones necesarias para asegurar la calidad y coherencia de los datos utilizados en el modelo de predicción.

Como tercer objetivo, cada miembro del equipo se encargó de construir un modelo de predicción individual utilizando agrupaciones específicas de columnas de datos. Estas agrupaciones abarcaron la información de la tripulación, los datos de la aeronave y los datos climatológicos. Para ello, aplicamos algoritmos de aprendizaje automático con el fin de entrenar y ajustar los modelos, buscando predecir la probabilidad de accidentes de aviones de manera individual.

El siguiente objetivo fue implementar una interfaz de usuario utilizando la biblioteca Streamlit. Esta interfaz fue diseñada de manera intuitiva y amigable, permitiendo a los usuarios introducir los datos relevantes en cada uno de los modelos de predicción construidos por los miembros del equipo. Esta implementación facilitó la interacción con el sistema y mejoró su usabilidad.

Finalmente, el último objetivo consistió en integrar los resultados de los modelos individuales y realizar un análisis global. Este análisis nos permitió interpretar la probabilidad de ocurrencia de accidentes de aviones de manera conjunta, considerando la combinación de los resultados individuales. Esta aproximación nos proporcionó una visión más precisa y confiable de los posibles accidentes.

En resumen, nuestro trabajo se basó en el desarrollo de un sistema de predicción de accidentes de aviones, donde los objetivos secundarios de realizar un análisis exploratorio, preprocesar los datos, construir modelos individuales, implementar una interfaz de usuario y realizar un análisis global de resultados se complementaron para alcanzar el objetivo general de mejorar la seguridad en la industria de la aviación.

# Metodología

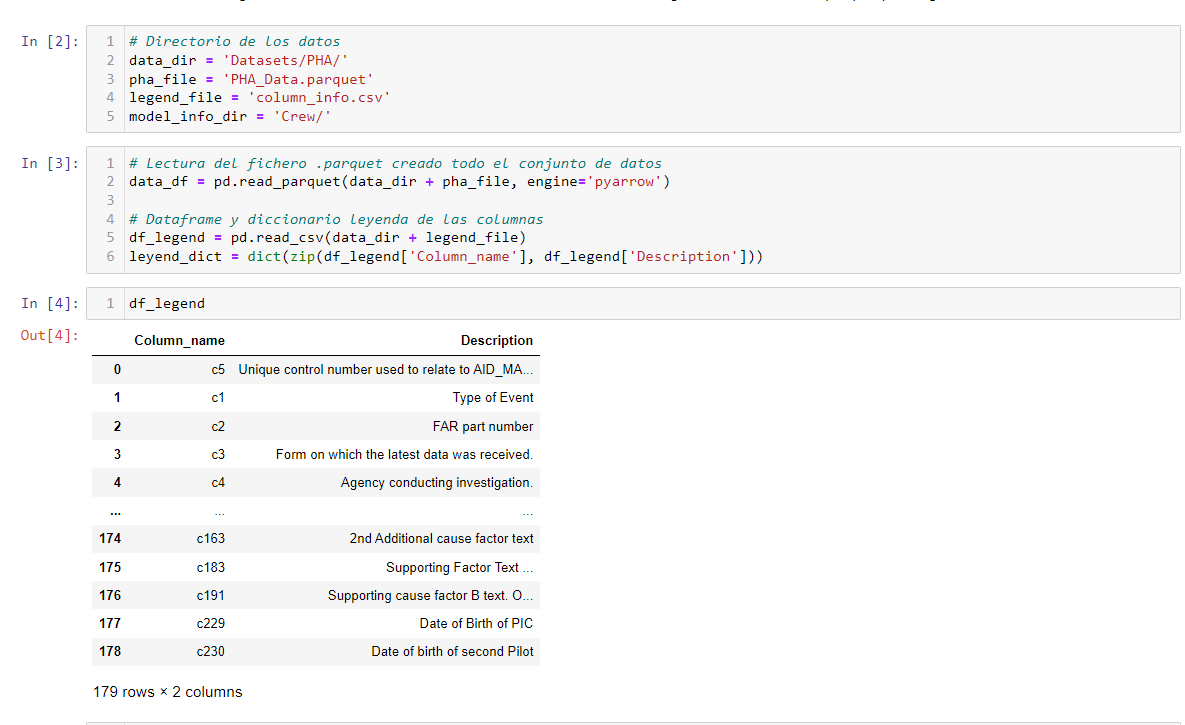
## 1.-Preprocesamiento del dataset

### 1.1.-Forecast(VICTOR)

### 1.2.-Crew

#### 1.2.1.- Obtención de datos

En primer lugar se obtienen los datos y se carga el modelo



#### 1.2.2.-Exploratory data analysis (EDA)

**Reducción de dimensionalidad y *multicolinealidad:***

Dado que el número de columnas es elevado y alguna de estas no aporta información a nuestro caso de estudio, se realiza primero la selección de los atributos de estudio dejando así un dataframe que sea más reducido en cuanto a sus atributos de partida para poder trabajar el análisis de datos con mayor facilidad.

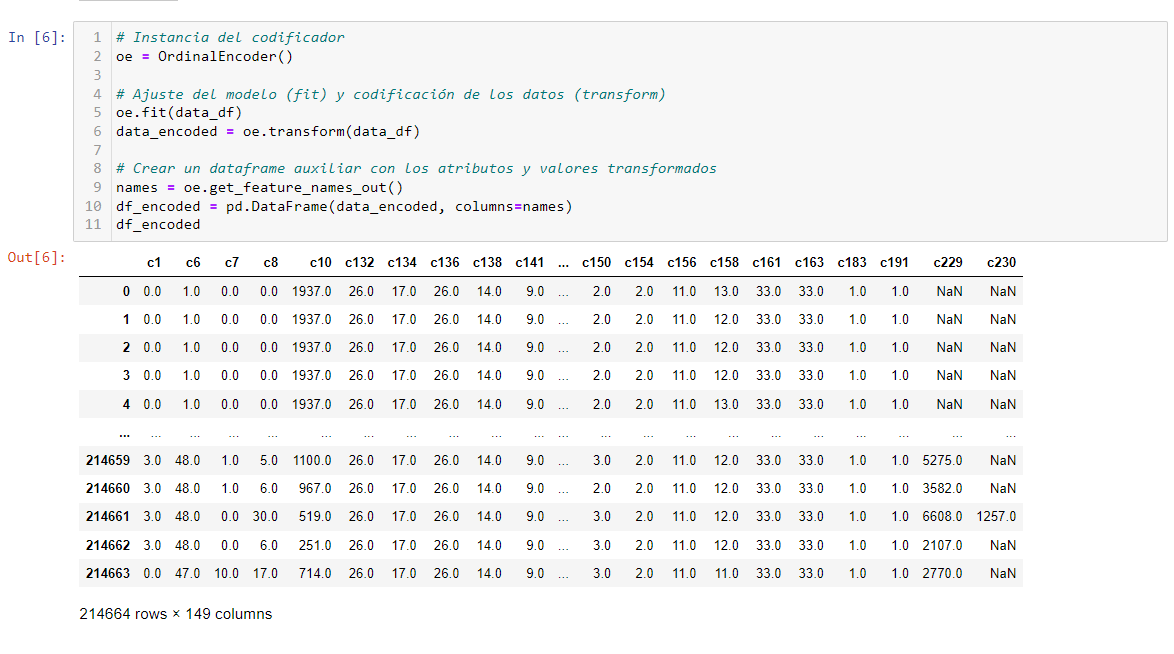
Si las variables predictoras tienen una alta correlación entre sí, puede haber problemas de multicolinealidad. La multicolinealidad ocurre cuando hay una fuerte relación lineal entre las variables independientes, lo que puede dificultar la interpretación de los coeficientes del modelo y conducir a estimaciones inestables. En tales casos, puede ser deseable eliminar una de las variables correlacionadas o combinarlas para crear una nueva variable

También es importante la interpretación y simplicidad; una alta correlación entre las variables predictoras puede dificultar la interpretación de los resultados. En ese caso, puede ser preferible trabajar con variables menos correlacionadas para facilitar la comprensión de cómo cada una afecta a la variable objetivo.

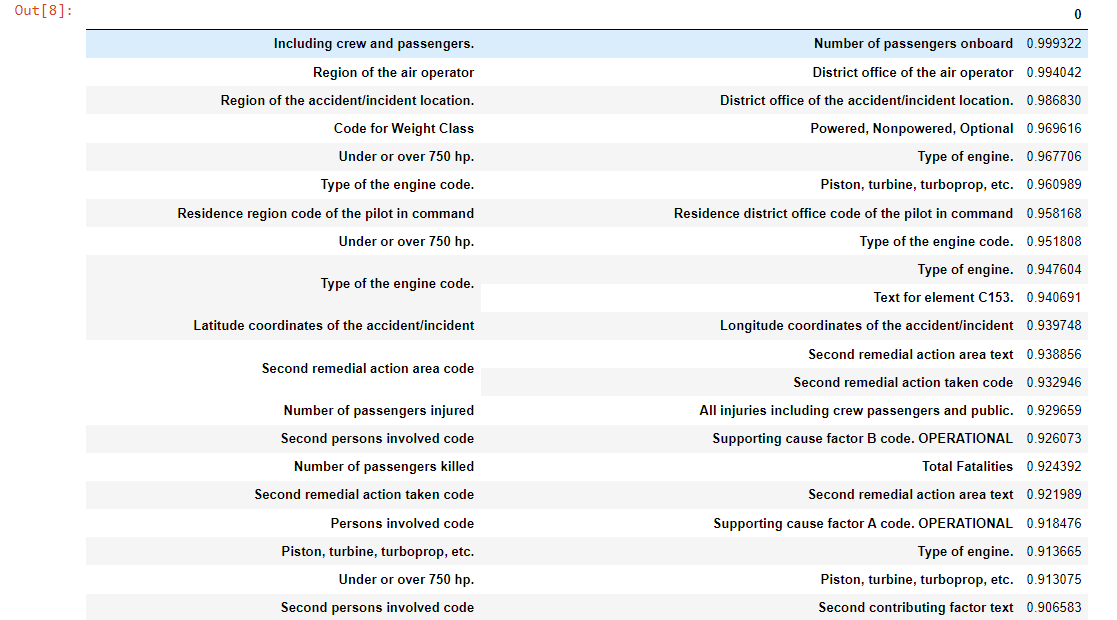
Como primer estudio para ver la relación de los datos y ayudarnos a tomar la decisión de los atributos a seleccionar, se observa la relación de las columnas de forma gráfica. Esto es, la dependencia que tiene una columna con respecto a las otras si estas varían.

Para este estudio, se emplea el método corr de Pandas. Pero, previamente se requiere codificar las variables no numéricas ya que esta función, únicamente actúa sobre las columnas numéricas, y es este momento, queremos ver la relación de todas las columnas. Para ello, se hace uso del módulo preprocessing de scikit-learn. Con este módulo se puede hacer uso de métodos que nos permitirá codificar los atributos categóricas a numéricos.

Se utiliza el encoder OrdinalEncoder que transforma el valor en el ordinal para la columna, dando como resultado valores numéricos de 0 a n-1 característica.



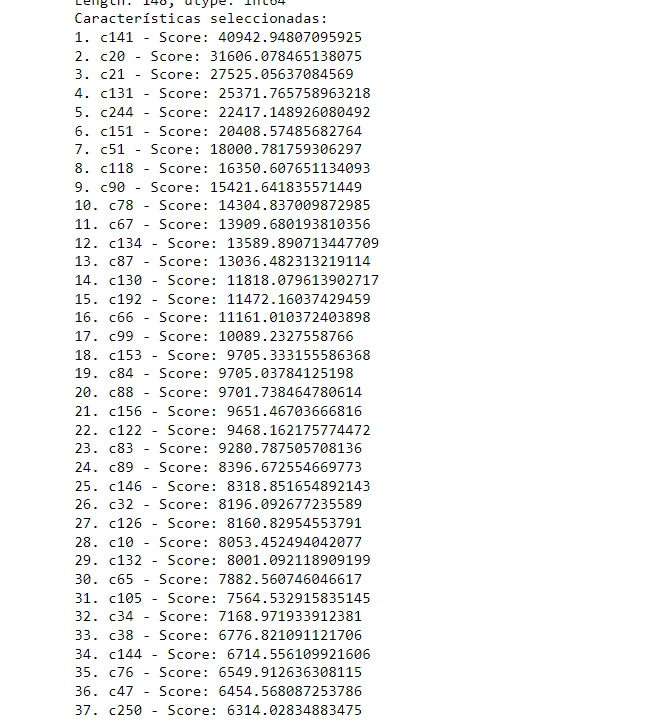




Si todas las variables predictoras están altamente correlacionadas entre sí pero también tienen una fuerte correlación con la variable objetivo, esto puede ser beneficioso. Significaría que todas las variables están capturando información relevante para predecir la variable objetivo. Sin embargo, si algunas variables no están correlacionadas con la variable objetivo pero sí entre sí, es posible que esas variables no estén aportando información adicional al modelo y podrían eliminarse.



Como se puede apreciar hay variables con alta coorelación con la variable objetivo por lo que podemos tener una idea de que variables son potencialmente interesantes.



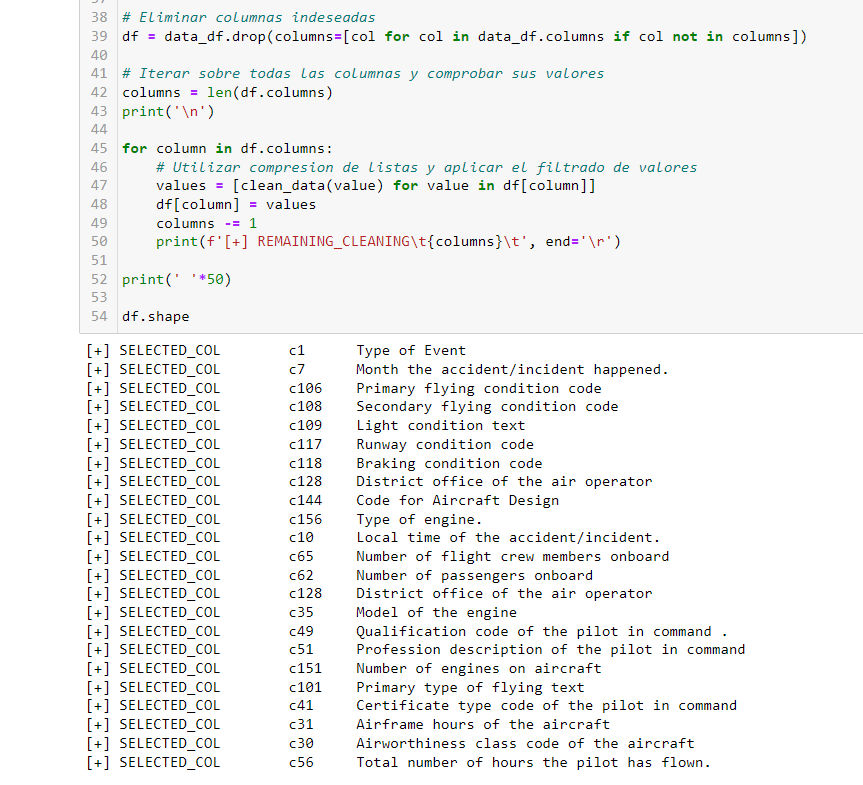
**Selección de atributos**

Si las variables están altamente correlacionadas y una de ellas falta o se modifica, es más probable que otras variables correlacionadas puedan proporcionar información similar. Esto puede hacer que tu modelo sea más robusto y estable en presencia de cambios o ruido en los datos.

De estas 50 variables con alta relación con la variable objetivo escogeremos teniendo en cuenta que queremos realizar una predicción en relación con los datos de la tripulación, pasajeros y algunas características adicionales relacionadas.

Esta lista de variables con gran correlación sobre la variable objetivo debemos filtrar las variables que tienen una alta correlación entre sí pero no con la variable objetivo ya que como hemos mencionado anteriormente pueden ser redundantes y no aporta ningún beneficio extra.

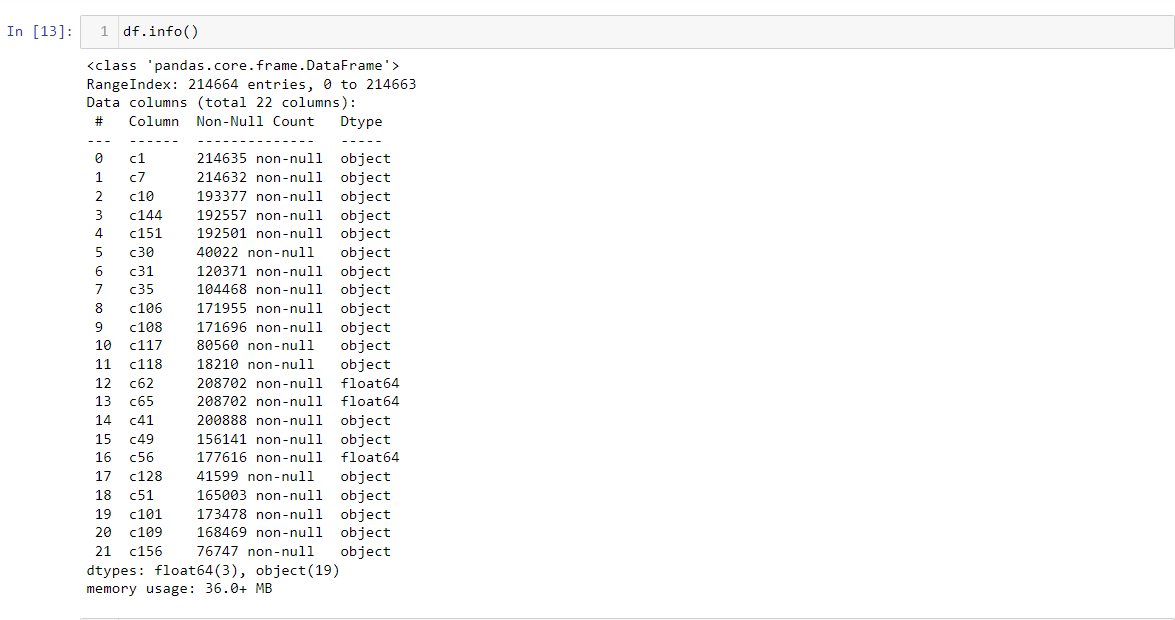




**Análisis conjunto de datos**

Para tener un contexto de los datos que se tiene, se realiza una serie de visualización de datos.

La primera de estas acciones consiste en describir los datos que tenemos, en su tipo y cuántos de estos datos son válidos, o al menos, no desconocidos (NaN).

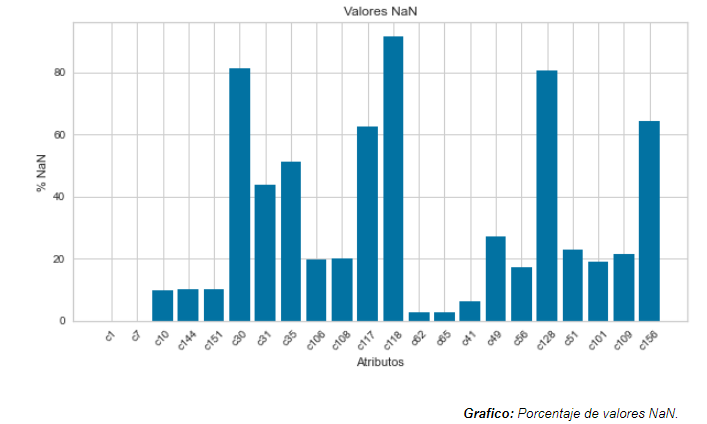




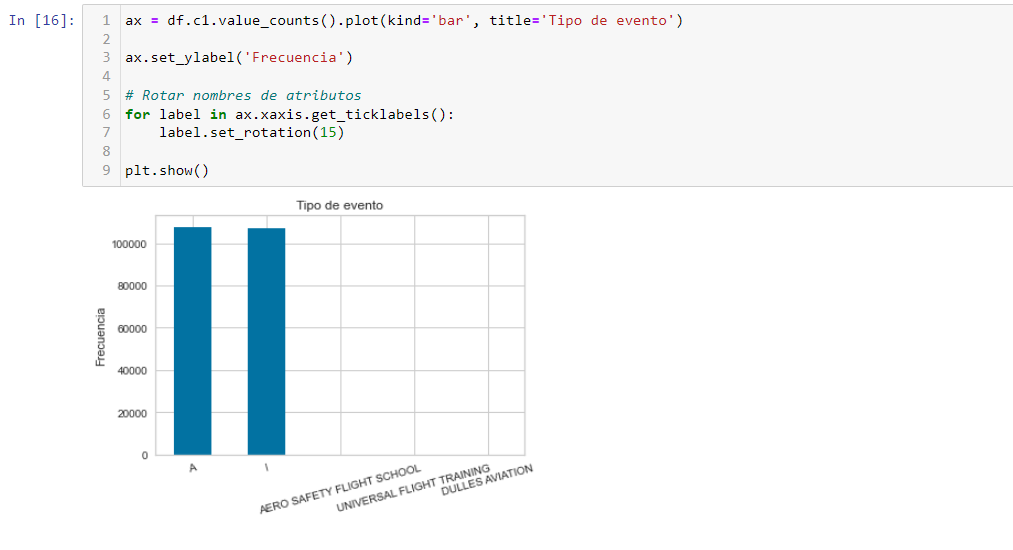
De forma gráfica, y desde el concepto de porcentaje de valores NaN en las columnas, se muestra el siguiente gráfico.



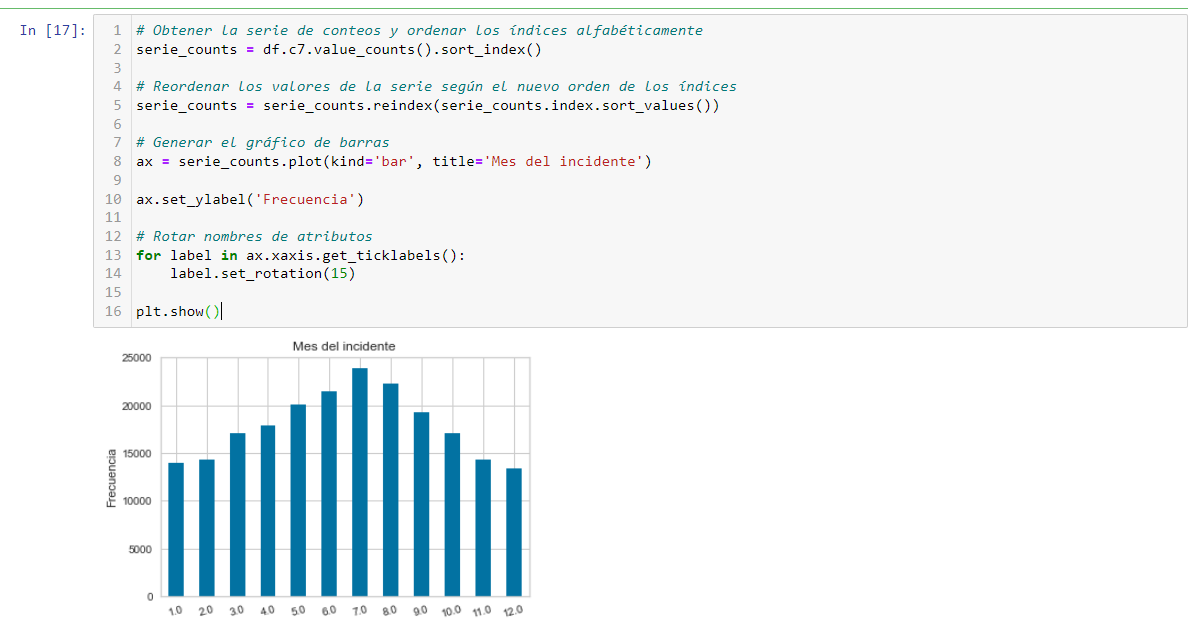
Se puede observar que en algunas columnas hay bastantes datos faltantes.



El modelo que se desea realizar, debe de ser capaz de predecir el tipo de incidente que ocurrirá. Para ver si tenemos unos datos balanceados, a continuación se muestra una gráfica de la predicción a realizar, donde podemos observar que si existe un balance de datos.

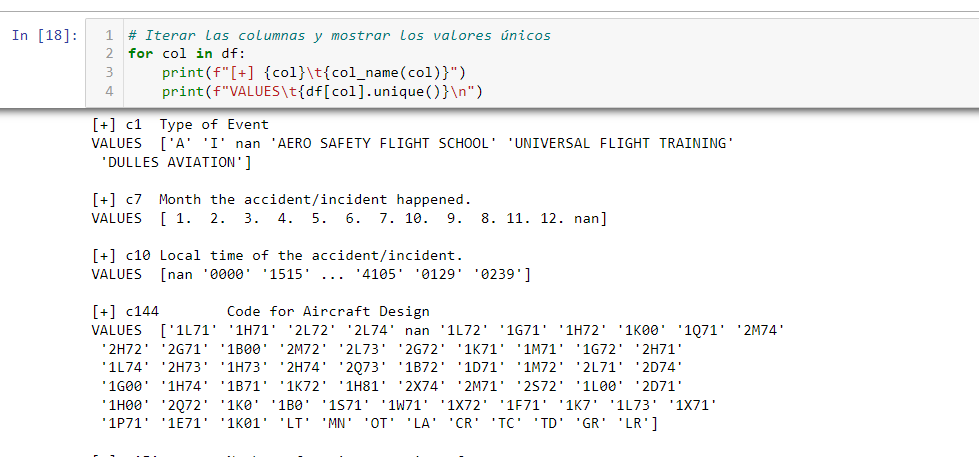


A continuación se muestra el balanceo de datos para la columna del mes del incidente. Teniendo en cuenta el tipo de análisis a realizar, se realiza una previa visualización de la distribución de los sucesos por esta característica.



**Descripción de valores**

Dado el elevado número de datos faltantes y sabiendo que existe valores que están en columnas diferentes por el estudio realizado para obtener el dataset de trabajo. A continuación se detalla los valores que existe para cada uno de los atributos del modelo de datos obtenido.



Como se puede observar, existe la mitad de columnas que contiene más del 50% de datos nulos o vacíos que habrá que tener en cuenta a la hora de la selección de caracteristicas para nuestro estudio. Además de valores atípicos para su columna por encontrarse desplazados.

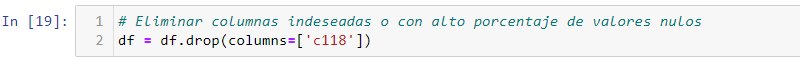
Se observa que las clases de estudio (Accidente o Incidente) están balanceadas

Por último, analizando el tipo de datos para el dataframe, se observa que la gran mayoría de atributos hace referencia a tipo objeto. Se debe a que determinados de estos atributos, tiene datos categóricos que mezcla números y letras en sus datos. No obstante, se analiza más adelante cuál de estas son realmente este tipo de columnas y cuáles son fruto de un error en los datos.

#### 1.2.3.-Data preprocessing(DP)

Con las conclusiones extraídas en el apartado anterior, en este apartado se realiza las correcciones necesarias para corregir estos errores.

**Eliminar características**

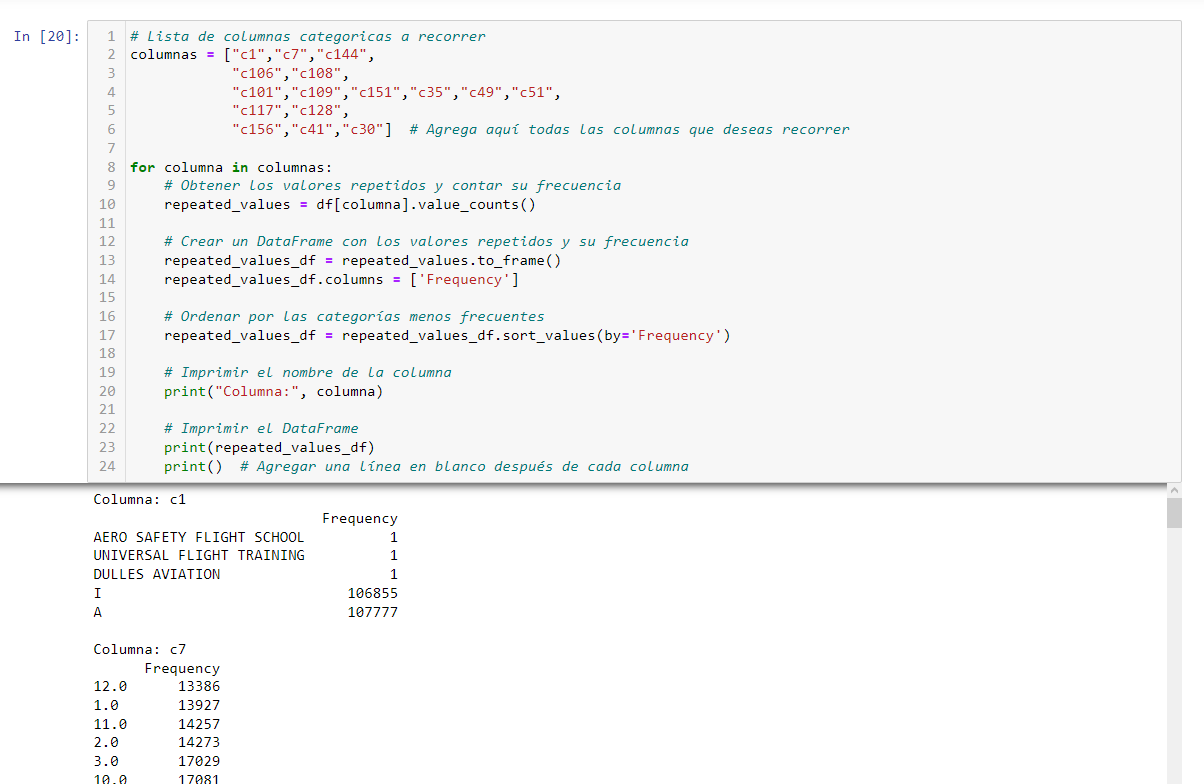
La columna c118 tiene demasiados valores nulos, por lo que una imputación de valores podría afectar negativamente a la predicción. 

**Valores desplazados o atipicos**

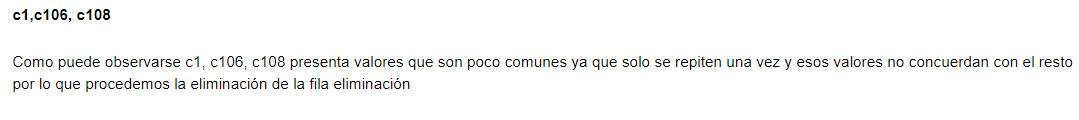
A continuación, se define los valores que ha de tener cada una de las columnas categóricas para encontrar en que otras columnas puede existir y asi poder relocalizarlas. Esta busqueda se realiza sobre todo el dataset, de este modo, se podrá importar esos valores a las columnas del dataset de estudio.

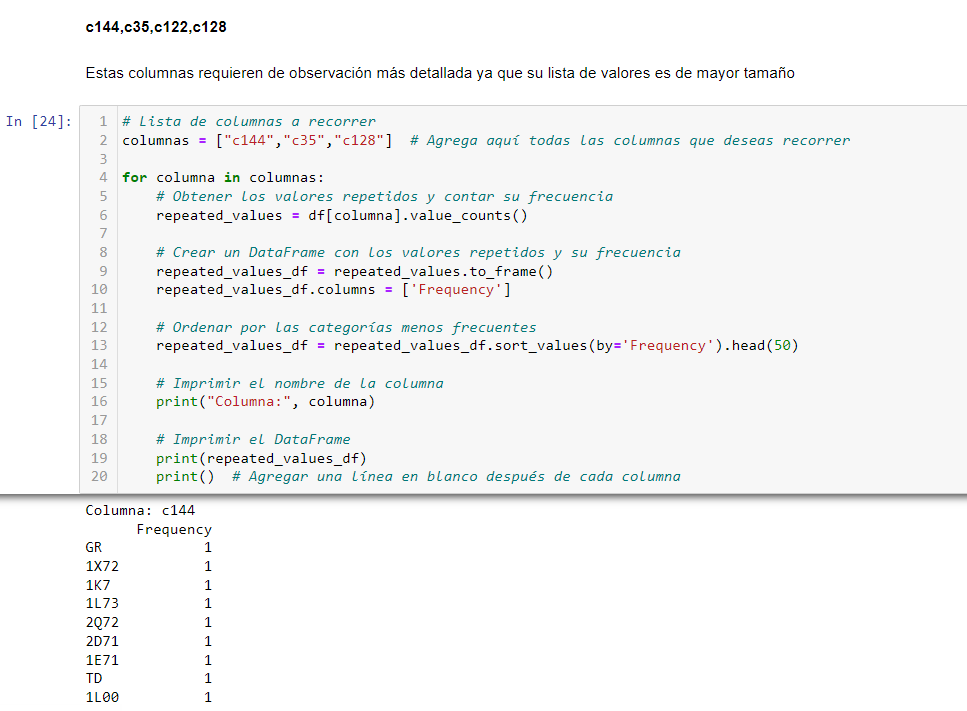
**Análisis**

Vamos a contar los valores de las columnas que a priori parecen categóricas para poder identificar algunos valores que no deberían estar en determinadas columnas debido a un desplazamiento, generando un valor atípico.



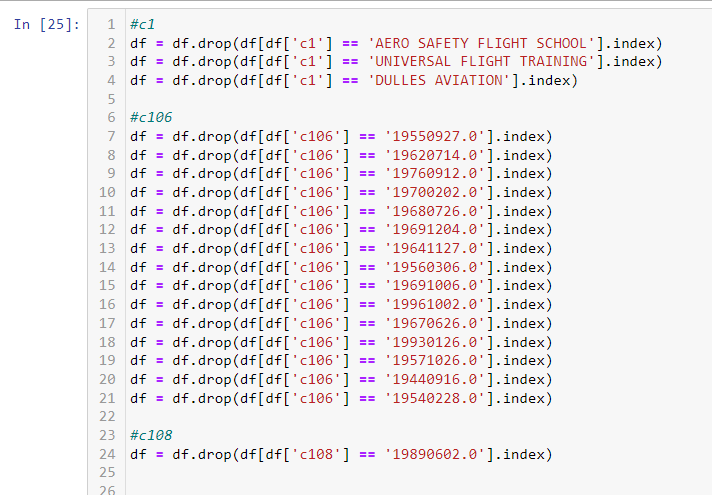






Tras ver los resultados, se puede apreciar que estas columnas no presentan ninguna alteración.

**Acciones**

Tras poderse comprobar que c1, c106 y c108 presentan valores atípicos se procede a borrar las filas que alteran la variable ya que son pocos registros los afectados y no alteraría la predicción. 

**Variables numéricas.**

Tras la comprobación no aparenta tener datos atípicos o registros desplazados



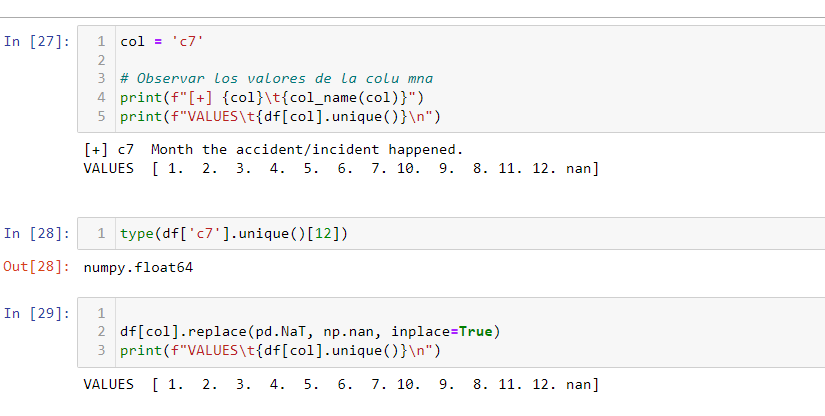
**Formateo de datos**

Ya tenemos comprobado el desplazamiento y valores atipicos, por lo que el siguiente paso sería la verificación y adecuación de los datos para que pueda ser interpretado correctamente por el algoritmo de predicción.

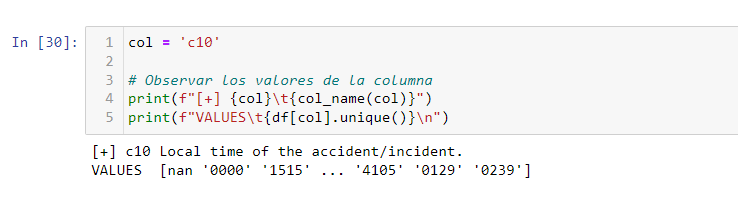
**\*NOTA**

Los datos aquí indicados como categóricos o numéricos, no son necesariamente de dicha índole para el estudio del modelo, esta agrupación se ha realizado por tipo de datos contenido pero no por significado para el modelo final

*c7 Month the accident/incident happened*

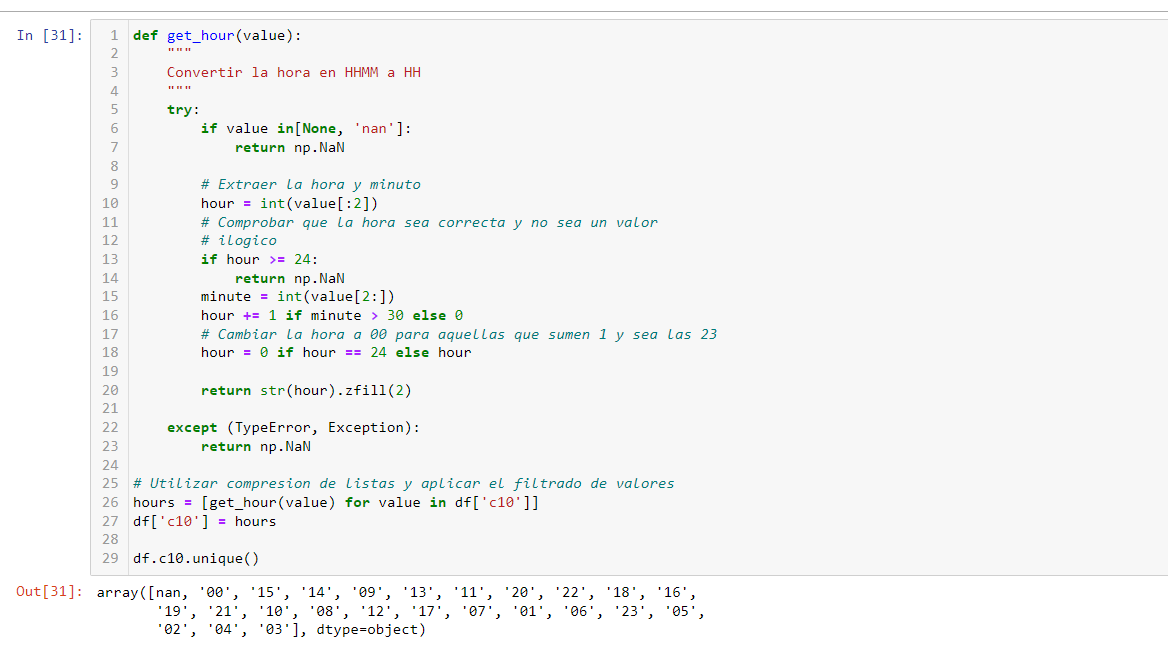


*c10 Local time of the accident/incident*

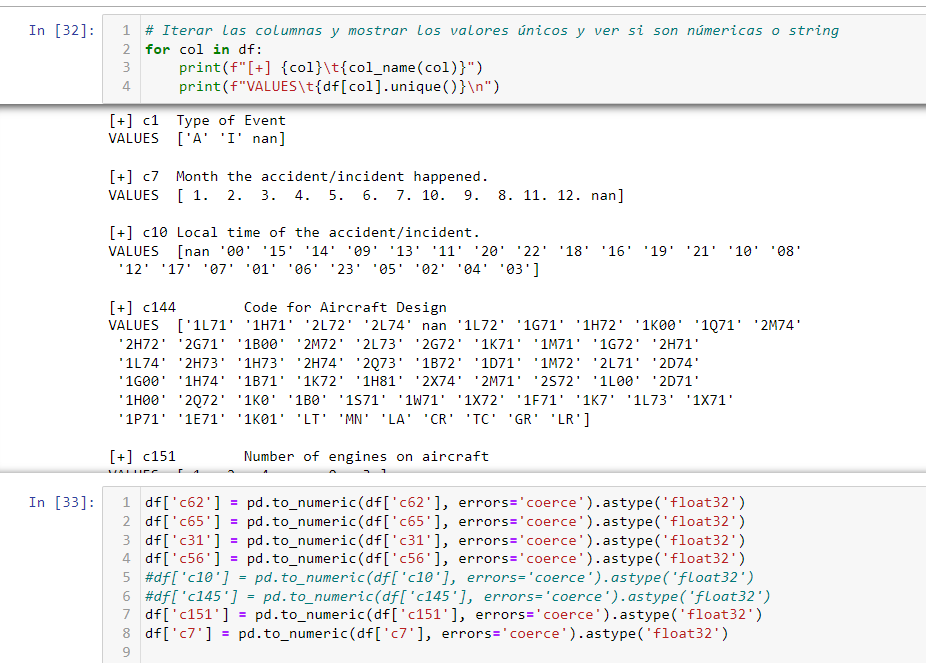


La columna c10 contiene los datos de la hora del incidente, pero para poder lidiar con todos los datos, se convierte los valores dados en formato HHMM en HH. Para ello, se recorre todos los valores y se modifica su valor, además, si los minutos son más de 30, se sumará una hora al resultado.

Se toma esta decisión para no sobrepasar al modelo de características para este atributo.

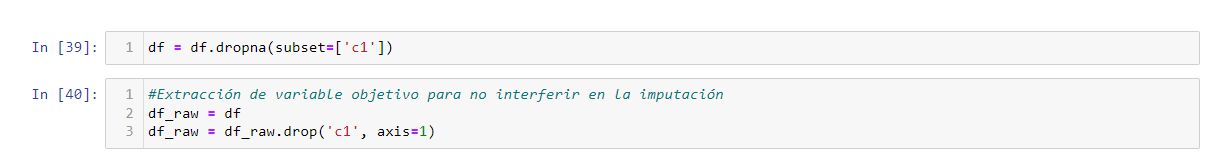


**Conversión de columnas mal identificadas como string a numéricas.**

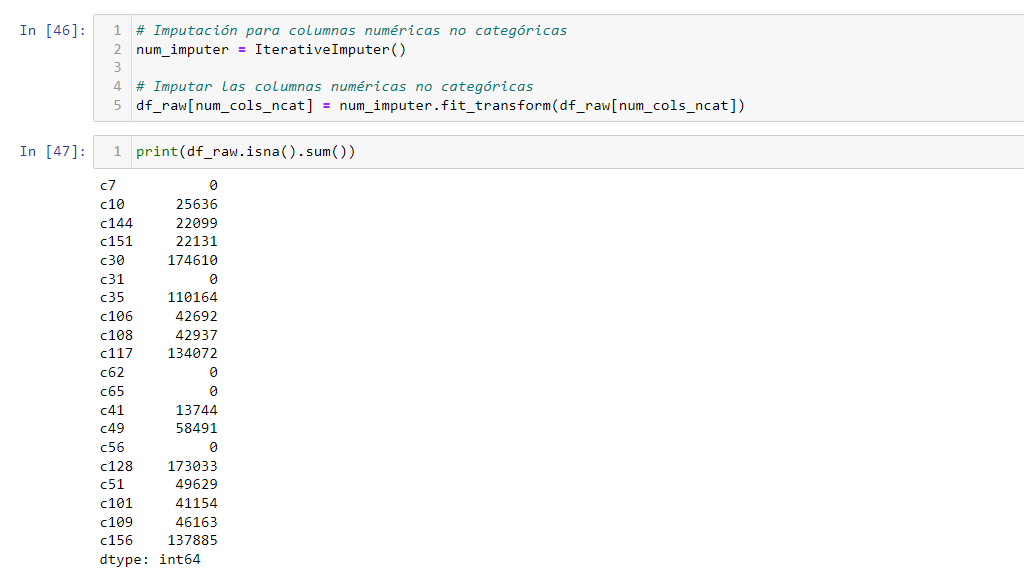


**Imputación de valores**

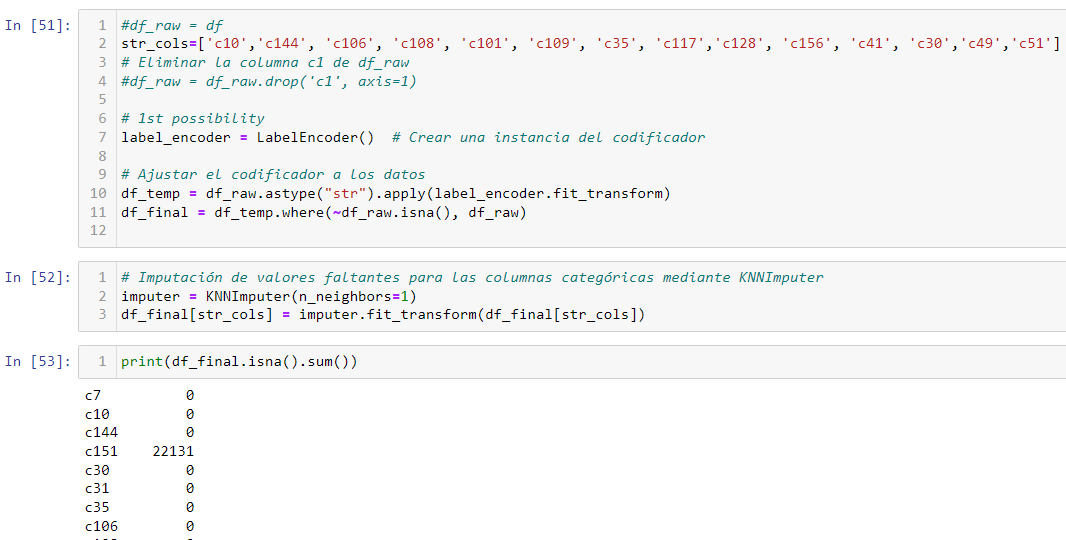
Primeramente debe de aislarse la variable objetivo de los datos a imputar ya que puede perjudicar gravemente a la predicción ya que los datos imputados pueden generar una relación muy fuerte con la variable generando ruido que el algoritmo de predicción puede mal interpretar.



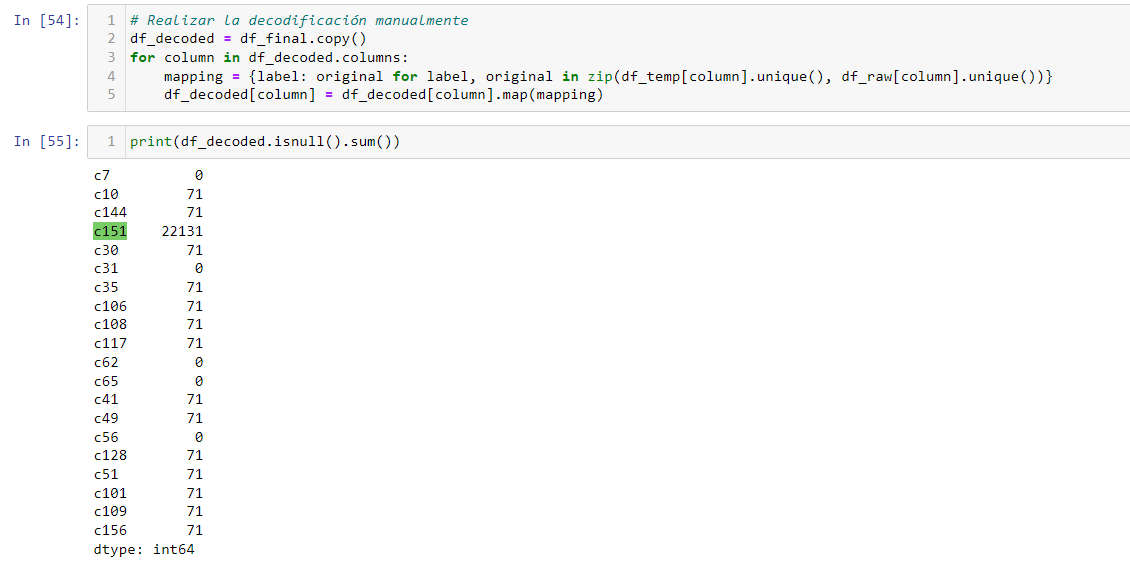
Imputamos con IterativeImputer de valores de columnas numéricas



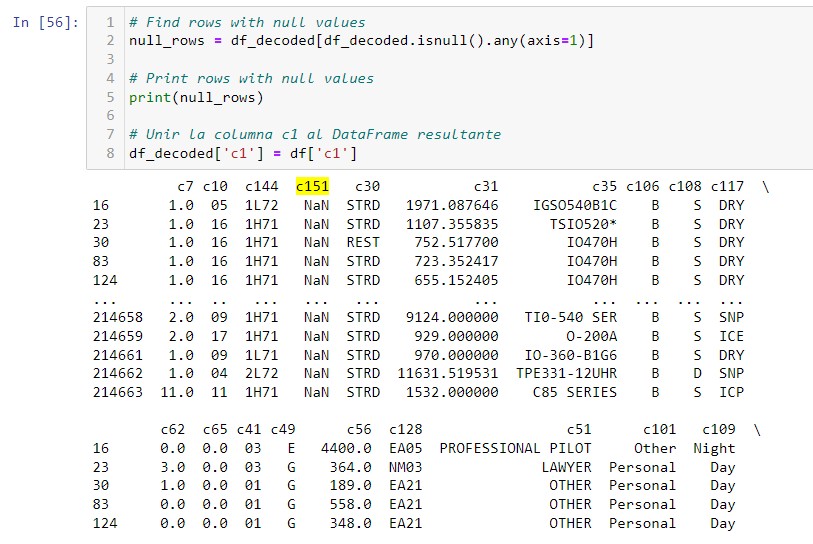
Ahora vamos a imputar las columnas tipo string para ello utilizaremos KnnImputer y codificaremos los datos mediante LabelEncoder, almacenando los valores originales para poder decodificarlos tras la imputación.



Ahora tras la imputación, realizamos la decodificación

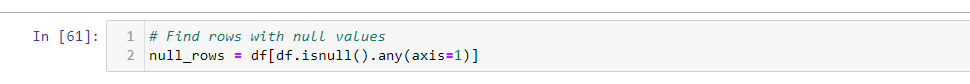


Los valores que no han podido imputarse se eliminarán dada la baja repercusión que tendrían sobre el dataset. Además al dataset imputado se le debe de volver a unir la variable objetivo.



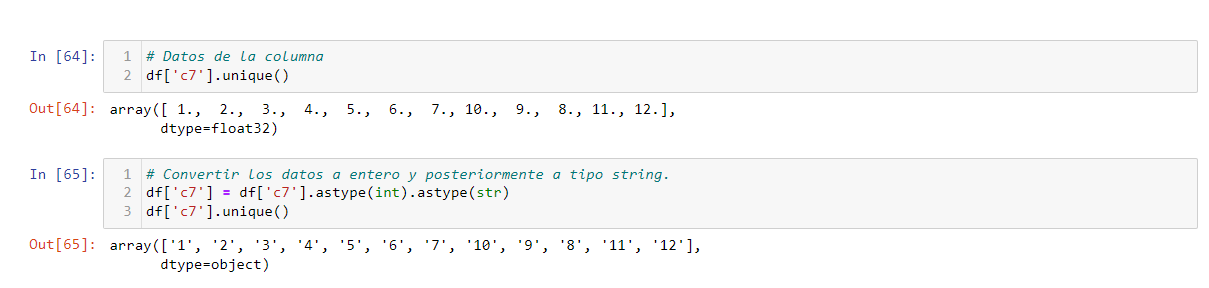


Tras la unión c1 tendrá registros que no se han borrado y que deben borrarse para que no tenga valores NaN.



**Conversión de datos**

Antes de continuar después de la imputación debemos de formatear la columna c7 como string.



#### 1.2.4.-Feature engineering (FE)

A continuación, se realiza una serie de modificaciones en los datos para que estos puedan ser interpretados por el modelo.

**Codificación de datos categóricos**

Existe una serie de columnas con datos categóricos que se deberá de codificar para poder tratar sus datos y realizar nuestra predicción del modelo.

Hasta ahora se ha trabajado con las columnas categóricas y numéricas en base a sus datos, pero, para realizar la codificación necesaria, se va a realizar sobre todas las columnas que necesariamente serán categóricas para realizar el modelo.

Existen tres técnicas de Data Encoding dependiendo de las características de los atributos:

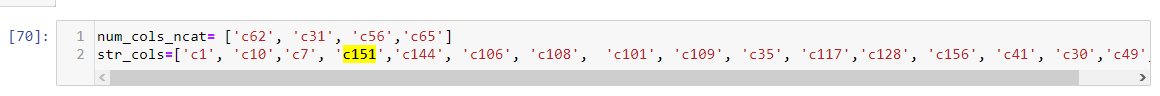
-OrdinalEncoder: Se utiliza para codificar variables categóricas que no tienen una relación de orden natural. Devuelve una única columna que va de 1 a n\_caracteristicas-1.

-LabelEncoder: Se utiliza para codificar variables categóricas que tienen una relación de orden natural. Devuelve una columna de 1 a n\_caracteristicas-1. Su uso se realiza principalmente para transformar la variable dependiente.

-OneHotEncoder: Se utiliza para codificar variables categóricas que no tienen una relación de orden natural. Devuelve una columna para cada atributo codificado, indicando 0 o 1 según la presencia de la característica para cada registro.

A continuación se muestra los tipos de datos categóricos del dataset para seleccionar que técnica emplear con cada atributo.

Para la variable dependiente u objetivo, c1, así como para los atributos de entrada cuyos datos tienen un orden de relación, se emplea LabelEncoder, col\_le. Para los atributos cuyas clases no están relacionadas, se emplea OrdinalEncoder. Para el resto de columnas se ha realizado un OrdinalEncoder, col\_oe.





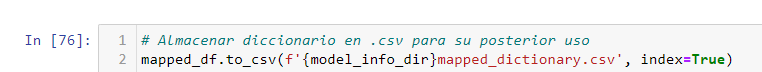
**LabelEncoder**

Para la transformación de los datos, se guarda el diccionario de los datos que se va a mapear para poder emplearlo en la entrada de nuevos datos.



**OrdinalEncoder**





**Normalización atributos numéricos**

En base a la desviación estandar de determinadas columnas, que son elevadas, se realiza el normalizado de los datos y el escalado entre 0 y 1.



#### 1.2.5.-Data modeling

Crear el modelo de predicción con el conjunto de datos obtenido.

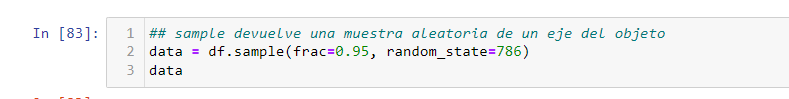
**División de los datos**

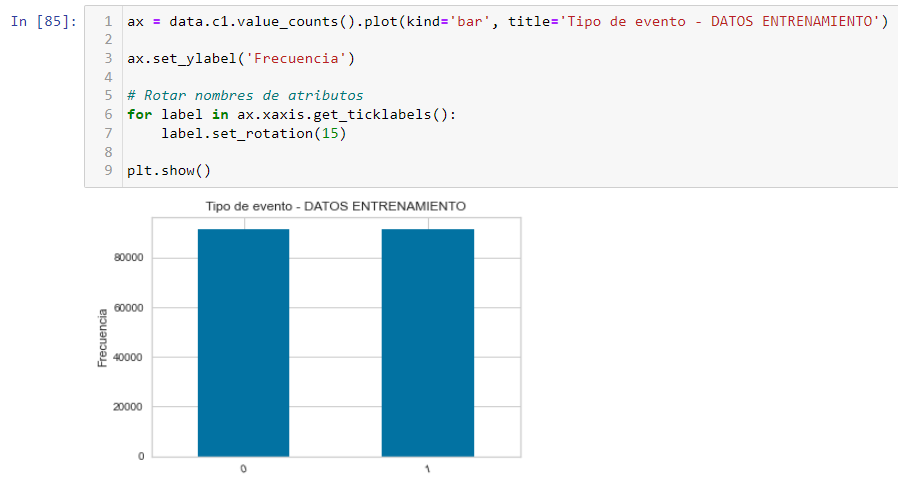
Para el modelado, se emplea unos datos de entrenamiento (train) y unos de prueba (test). Una vez el modelo está realizado, existe los datos de validación, estos no son vistos por el modelo durante el train o test.

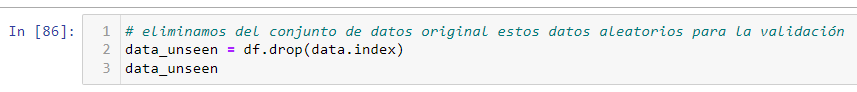
A continuación, se parte los datos en:

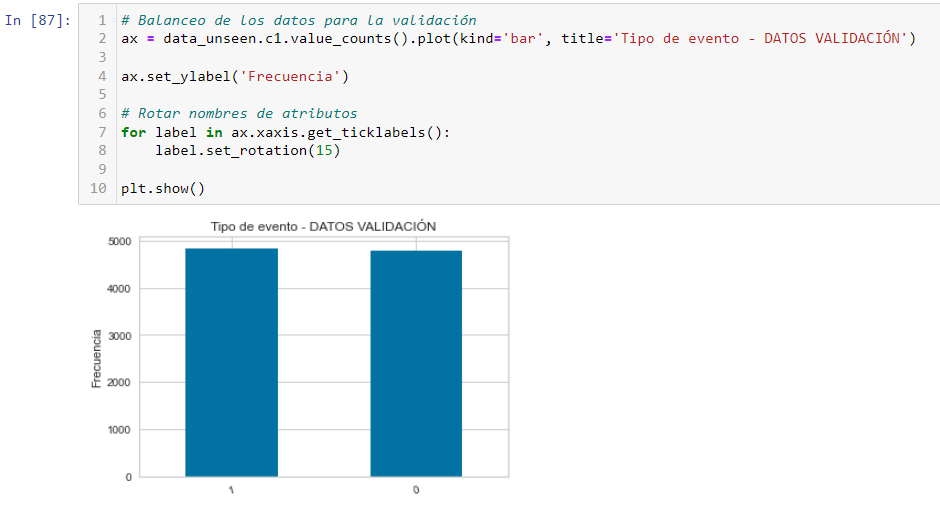
**data**: datos con los que se entrenará y testeará el modelo.

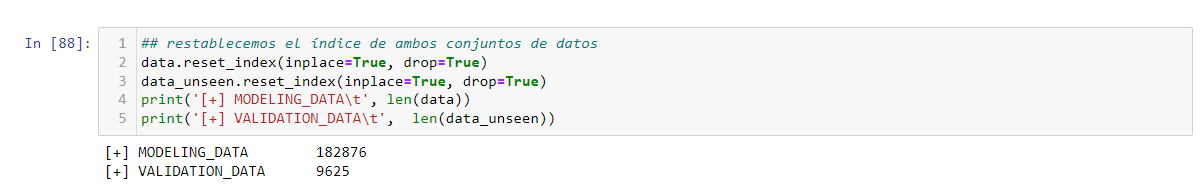
**data\_unseen**: datos que se emplearán para la validación (5% de los datos).











### 1.3.-Airplane(PABLO)

## 2.-Generar el modelo con pycaret

### 2.1.-Forecast(VICTOR)

### 2.2.-Crew

#### 2.2.1.-PyCaret SETUP

Para la configuraciones de nuestro clasificador, se emplea el metodo setup() de PyCaret, en el cual se le indica aquellos parámetros necesarios para realizar los modelos:

**fold\_strategy**: Permite seleccionar una técnica de validación cruzada.

**preprocess**: Existe la posibilidad de introducir datos ya preprocesados y por lo tanto indicar unicamente el pipeline de transformaciones que se desea (valor a False). O por el contrario, que PyCaret preprocese los datos para realizar imputacioens, codificacion de caracteristicas... (valor a True).

**pca**: Se permite el uso de reducción de dimensionalidad con el uso de Principal Component Analysis.

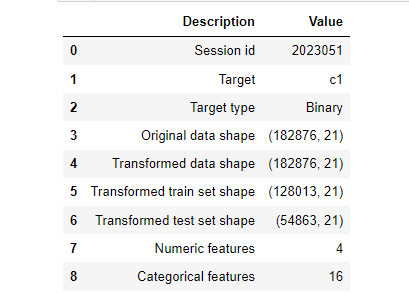
**low\_variance\_threshold:** Ignorar varianzas bajas en aquellos casos en los que existe una característica dominante sobre otras relacionadas y no existe mucha variación en la información de estas características.

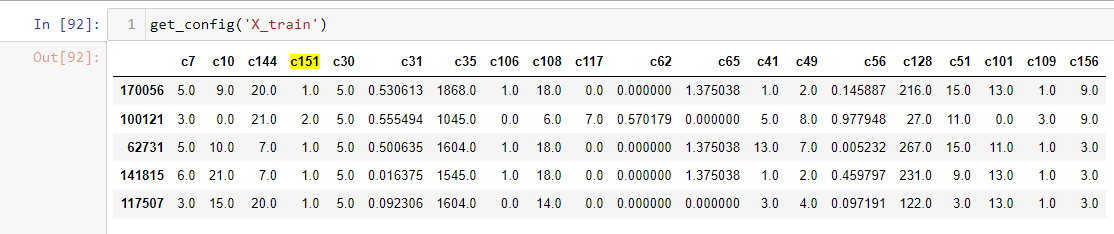
**remove\_multicollinearity**: Permite lidiar con las características que están fuertemente ligadas con otras del dataset.

**fix\_imbalance**: Cuando existe un conjunto de datos desbalanceado, se permite balancear este mediante diferentes técnicas.

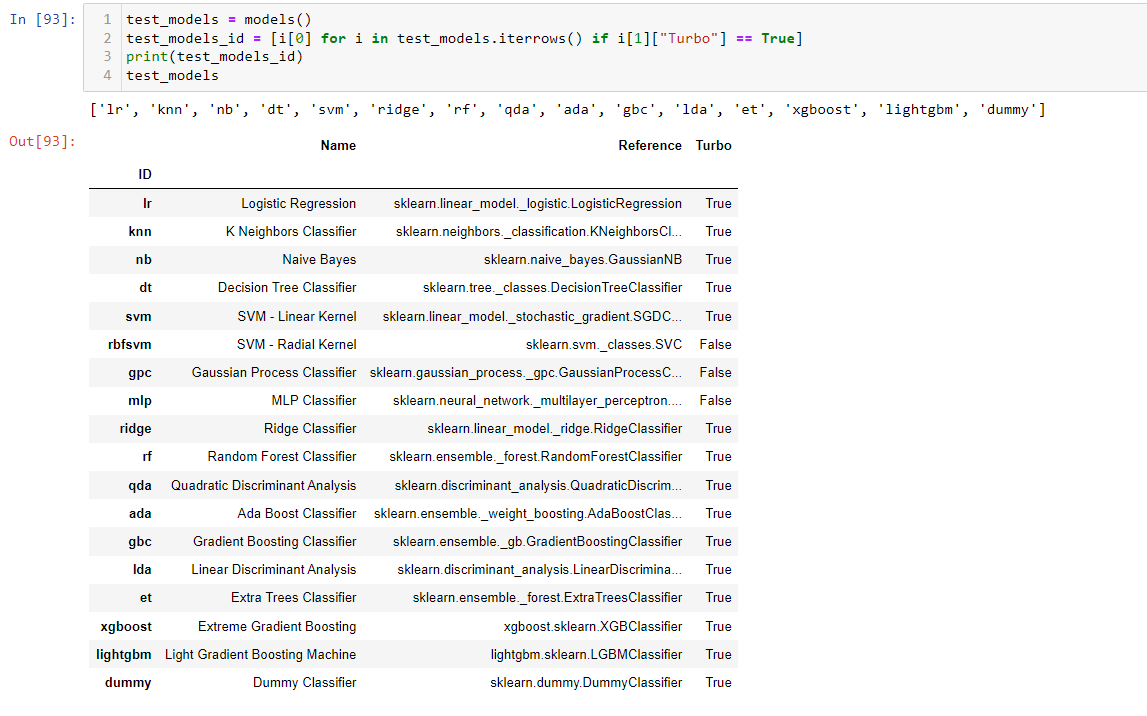
**remove\_outliers**: Eliminación de outliers en el conjunto de los datos.







#### 2.2.2.-Comparación de modelos



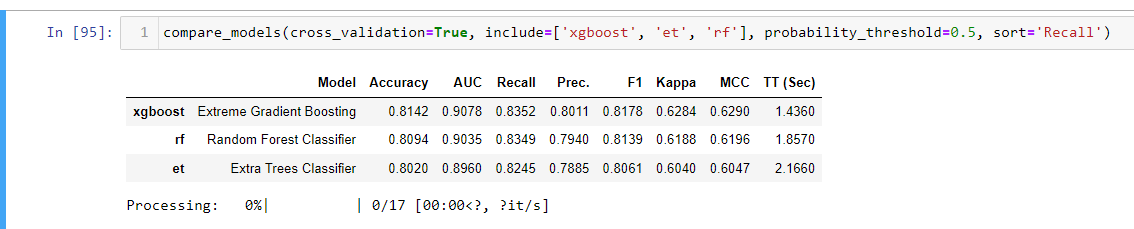
De los modelos disposnibles, se realiza una comprobación de sus predicciones ordenados por la Sensibilidad (Recall) para poder compararlos y escoger el mejor de ellos.

Dada la importancia de la predicción, es importante escoger un modelo cuya proporción de verdaderos positivos en relación con el total de muestras positivas en el conjunto de datos sea lo más elevada posible.



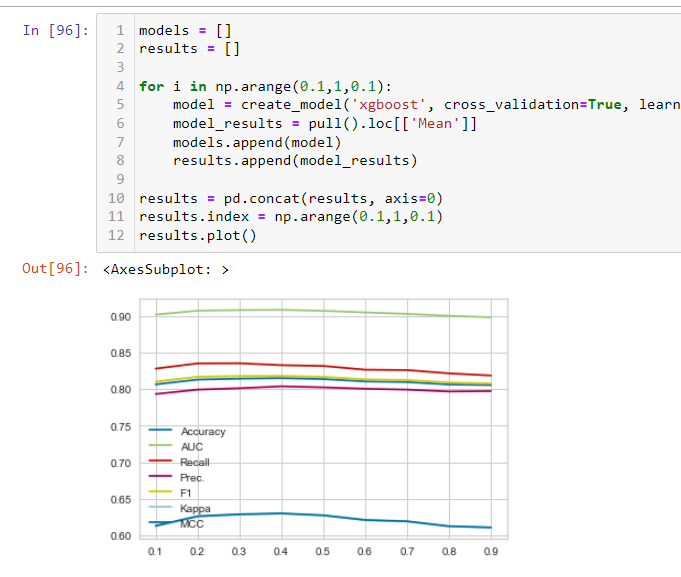
Teniendo en cuenta los resultados obtenidos, en cuanto a métricas y tiempos de ejecución, se decide escoger un top 3 para crear los modelos, poder ajustarlos y finalmente realizar el modelo con el mejor de ellos.

A continuación, se muestra la comparación de los modelos, esta vez, empleando la validación cruzada con 10 folds como se ha indicado en los ajustes.

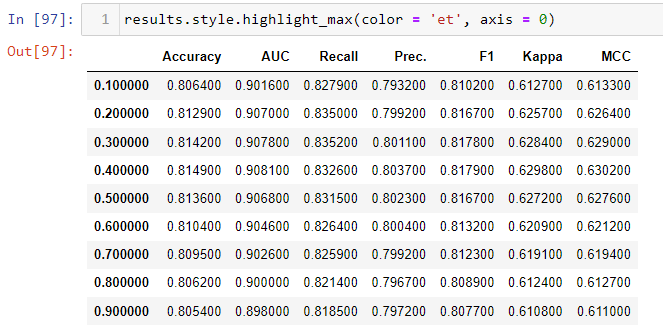


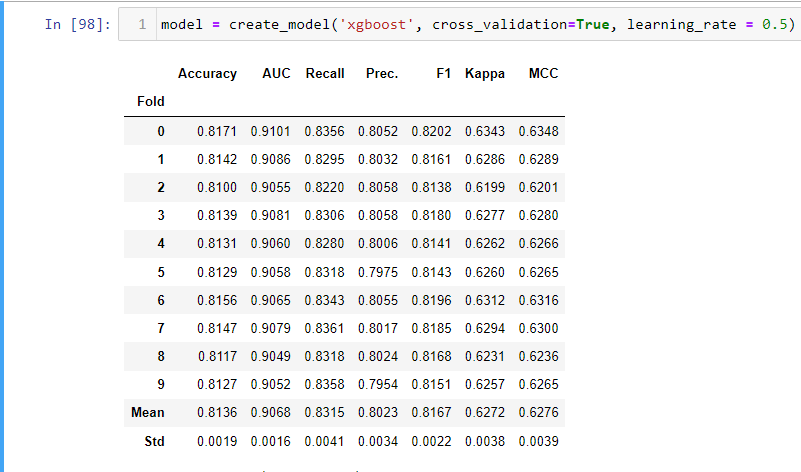
#### 2.2.3.-Creación del modelo

En base a los resultados anteriores, se escoge el modelo Extreme Gradient Boosting para realizar las predicciones.



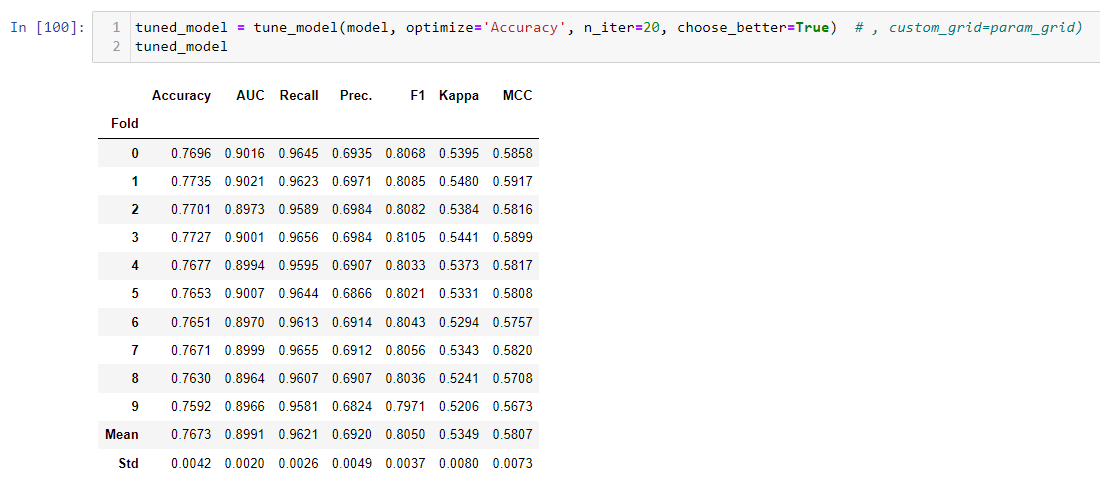
Tras la creación del modelo teniendo en cuenta su tasa de aprendizaje, se establece esta en 0.5.



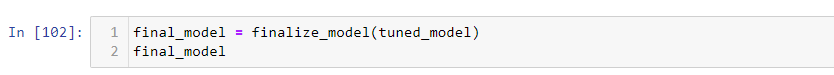


#### 2.2.4.-Ajustar el modelo

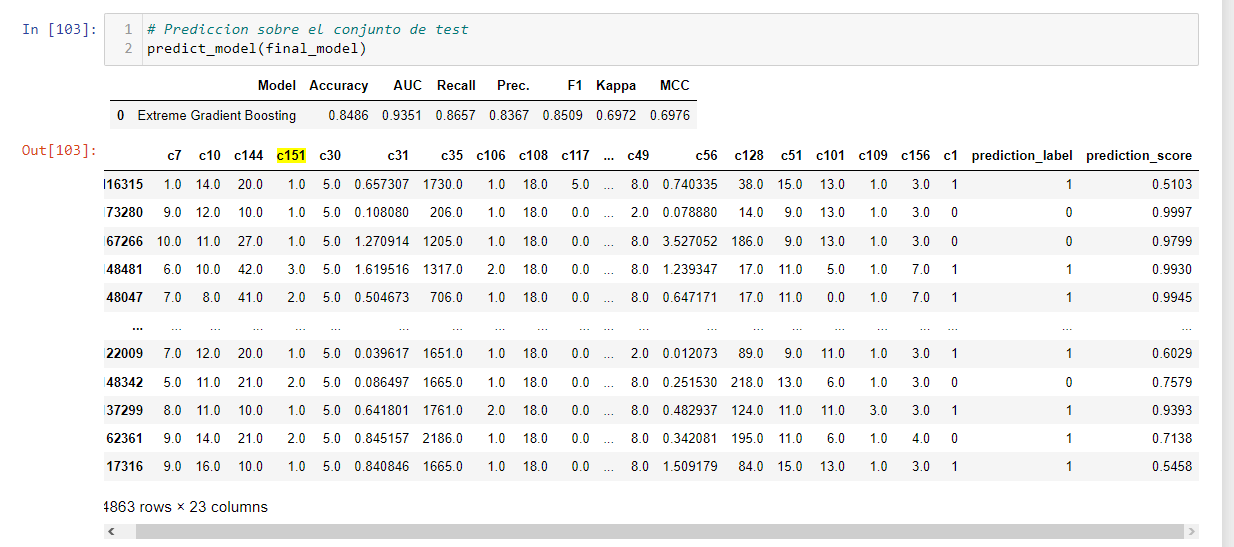


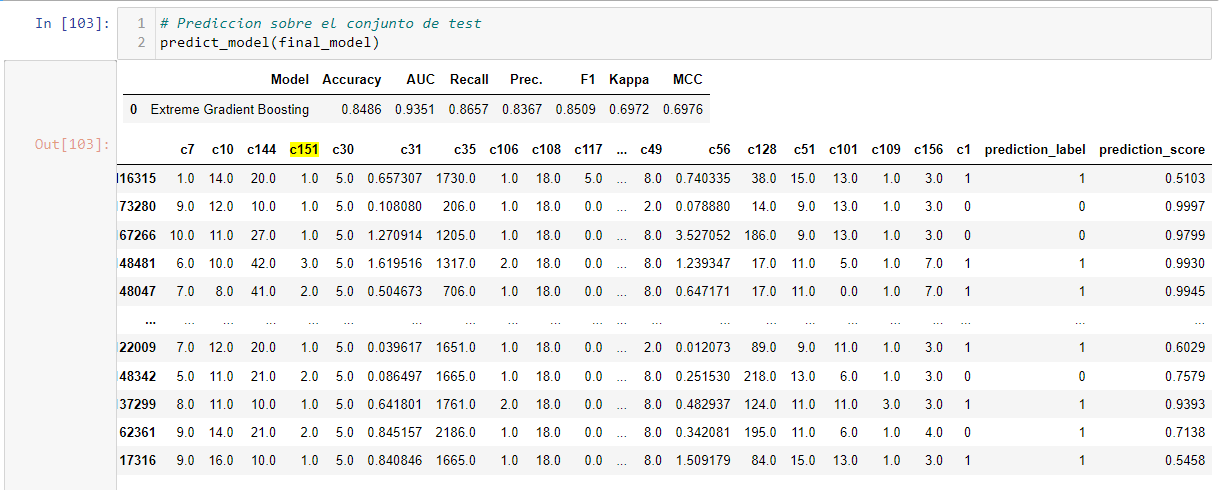


Con el mejor de los modelos seleccionado y ajustado, se ajusta el modelo al conjunto de todos los datos, incluyendo test y train. Realizando finalmente una predicción sobre el conjunto de datos que se extrajo en un principio y no se han usado durante todo el proceso de creación del modelo.



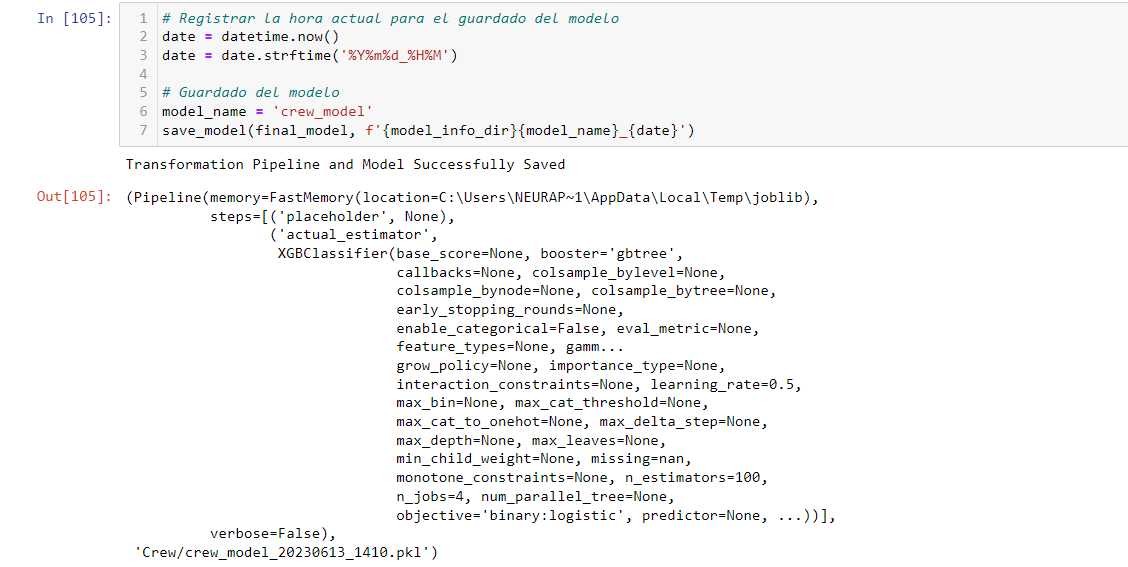
Ya con el modelo creado, pasamos a realizar las predicciones con los datos que se ha reservado para ello.



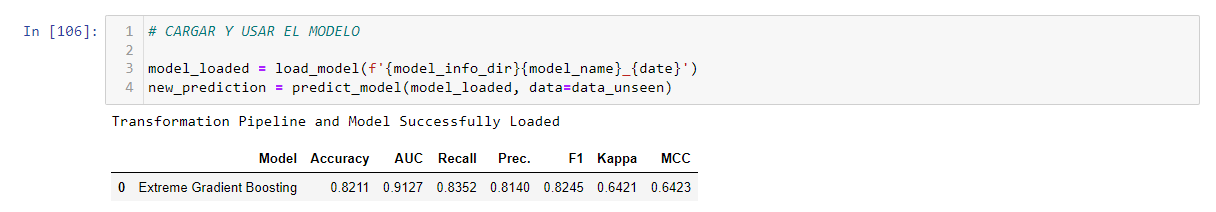


#### 2.2.5.-Guardar el modelo

Tras todo el proceso de modelado y ajuste del mismo, para poder utilizar el modelo en un entorno de producción, es necesario guardar el mismo. Para ello, se emplea las propias herramientas de PyCaret.



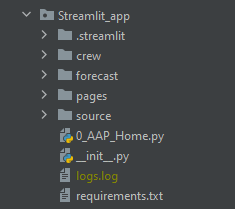
Ya estaría preparado para cargar y usar nuestro modelo.



### 2.3.-Airplane(PABLO)

## 3.-App de Streamlit

Para la app de Streamlit se ha generado esta estructura:

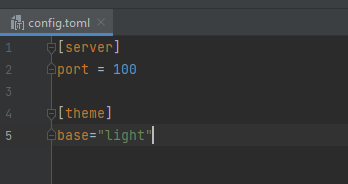


### 3.1.-Archivo .Streamlit

Como puede verse, el primer directorio “.streamlit” contiene el archivo de configuración.



En este archivo de configuración de la app aparece el puerto en el que correrá la aplicación y el tema a utilizar.

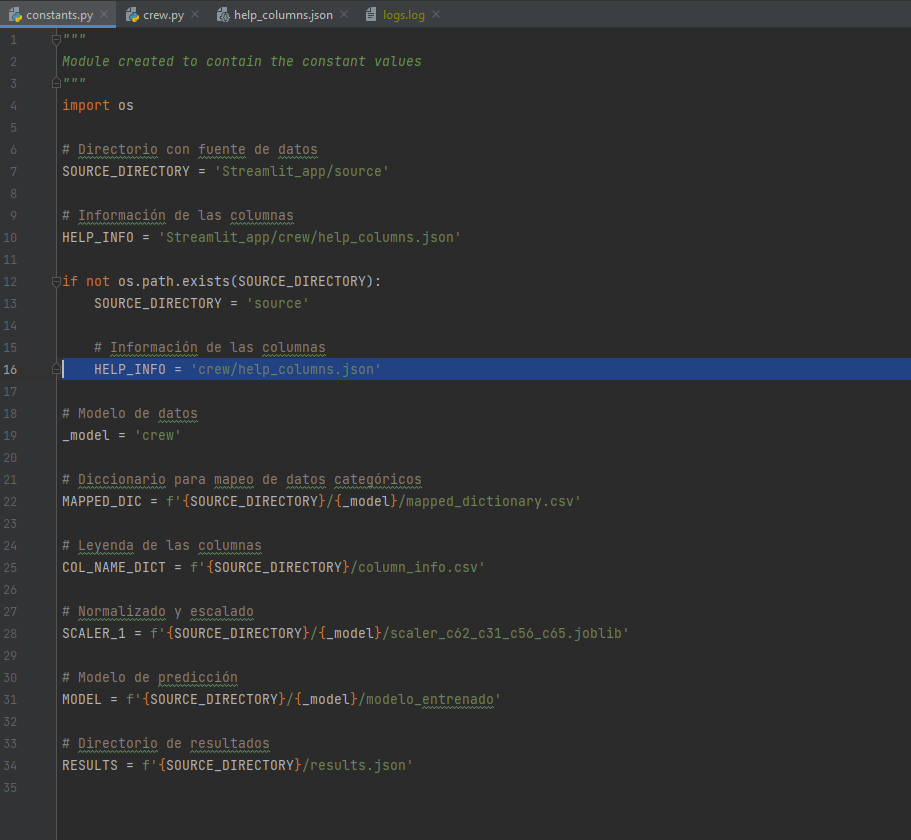


### 3.2.-Directorio Crew

#### 3.2.1.-Constants.py



En el directorio crew tenemos el archivo constants.py el cual contiene las rutas de archivos que necesitaremos para cada una de las funciones para la sección de “crew” de la aplicación.

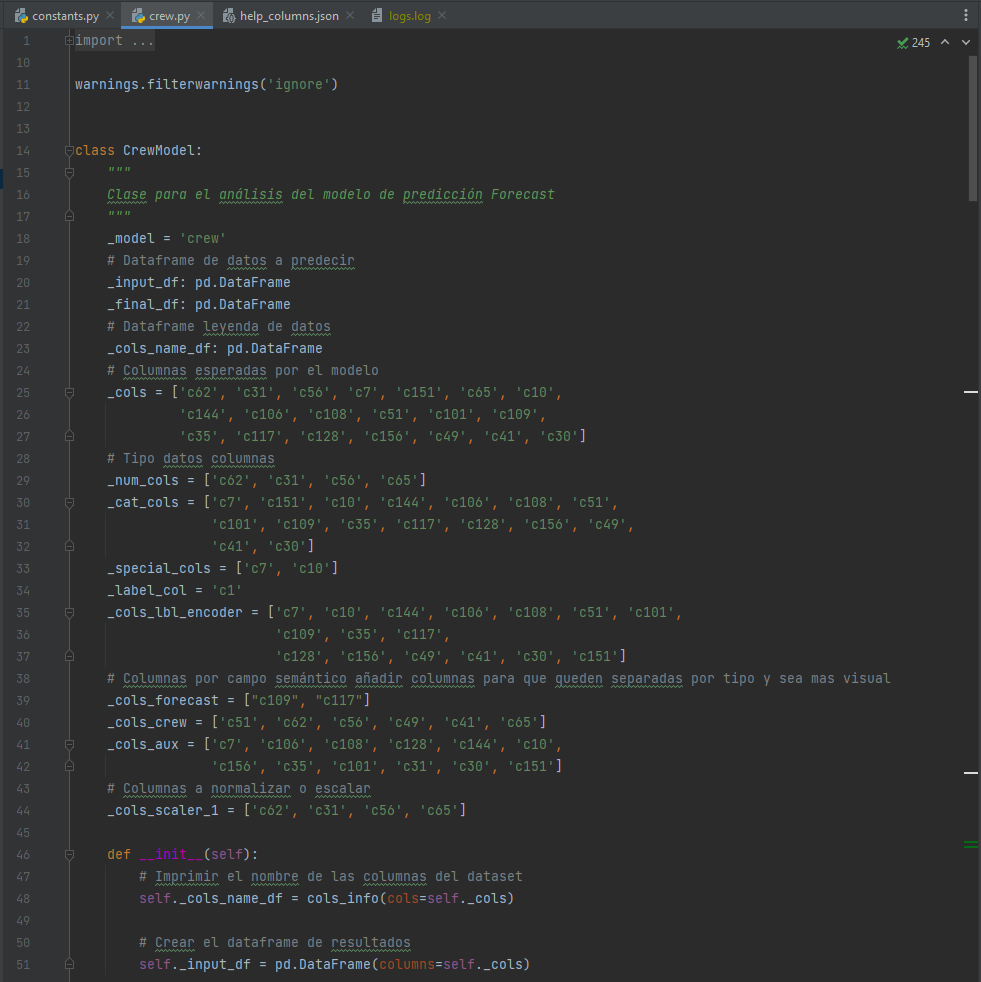


#### 3.2.2.-Crew.py

En este directorio también tenemos al archivo crew.py el cual contiene todas las funciones que se utilizarán en la sección “crew”.

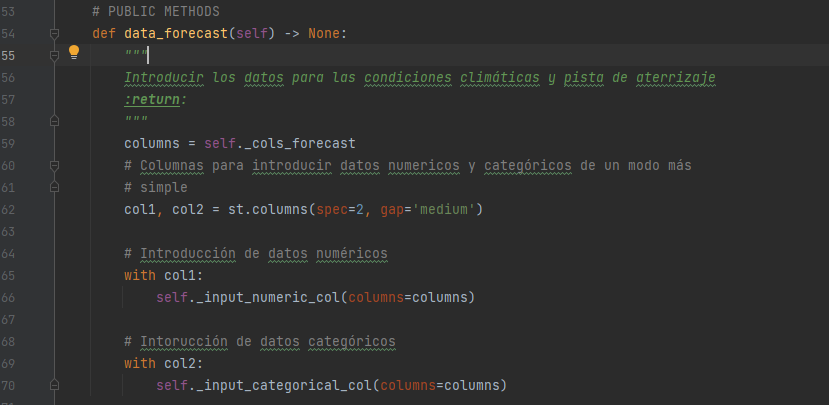
Primeramente se definen las columnas que se utilizarán dividiéndolas según sea numéricas o categóricas, segmentando además por tema de contenido para mostrar cada campo en las pestañas correspondientes. Además también se detallan las columnas que necesitan escalado y las que necesitan codificación.

Por ultimo en el método \_\_ini\_\_ se imprime las columnas del dataset y se crea el dataset de los resultados.

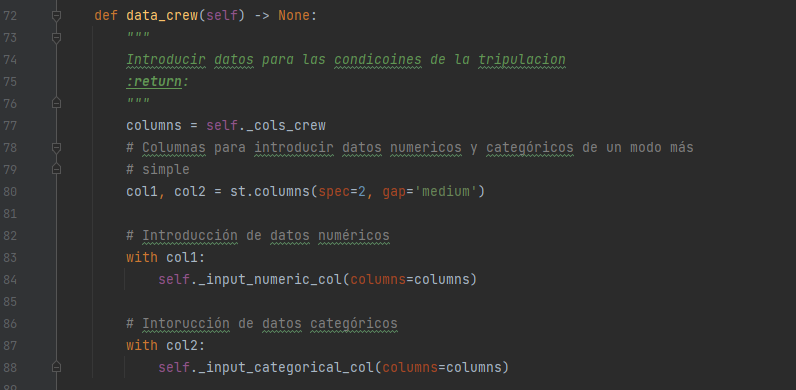


##### Métodos Publicos

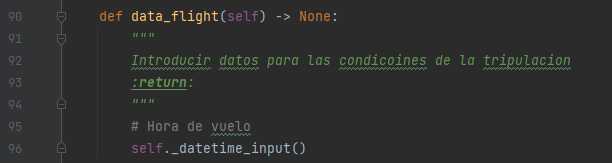
En data\_forecast se establece las columnas y disposión que aparecerá en la pestaña “forecast” de la sección “crew”.



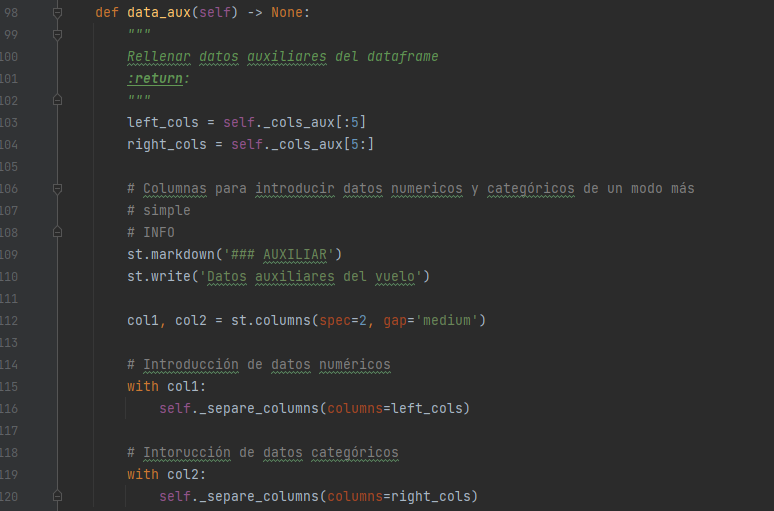
En data\_crew ocurre exactamente lo mismo que en el apartado anterior pero para la pestaña “crew”



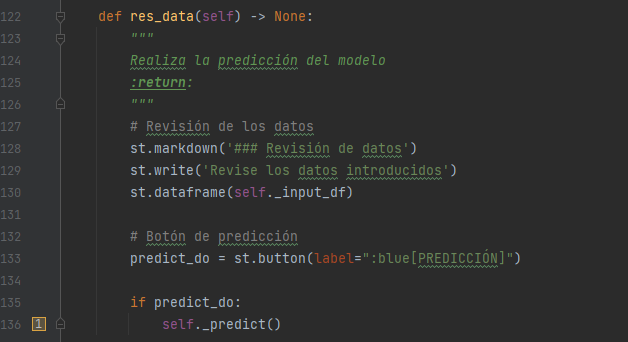
En data\_flight ocurre lo mismo pero para la pestaña “flight”



Y por último en data\_aux también se definen las posiciones de las columnas pero de la pestaña “auxiliar”

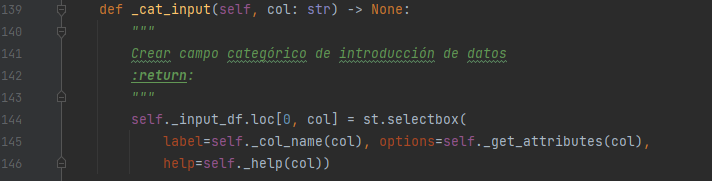


Con res\_data realizamos la predicción del modelo

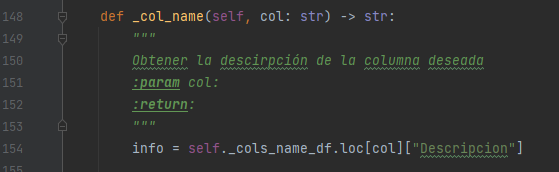


##### Métodos privados

Con \_cat\_input se crea un campo categorico para la introducción de datos



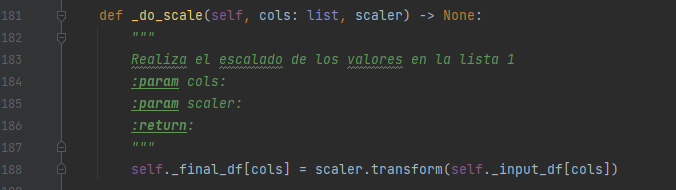
Con \_col\_name se obtiene la descripción de la columna deseada



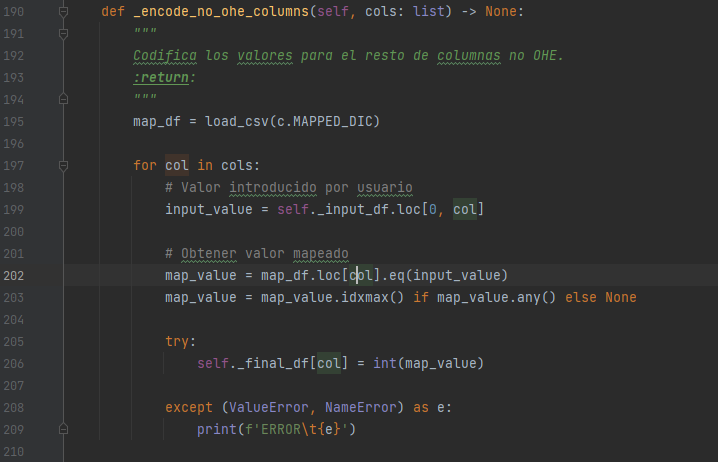
Con \_datetime\_input se establece los inputs de fecha y hora de vuelo.



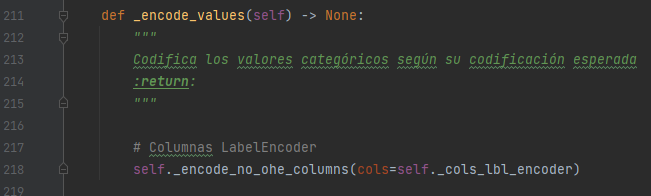
Con \_do\_scale realizamos el escalado de las columnas que lo requieren.



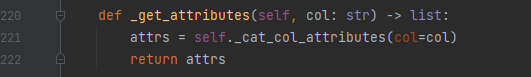
Con \_encode\_no\_ohe\_columns codificamos los valores para el resto de columnas en las que no se ha utilizado OneHotEncoder.



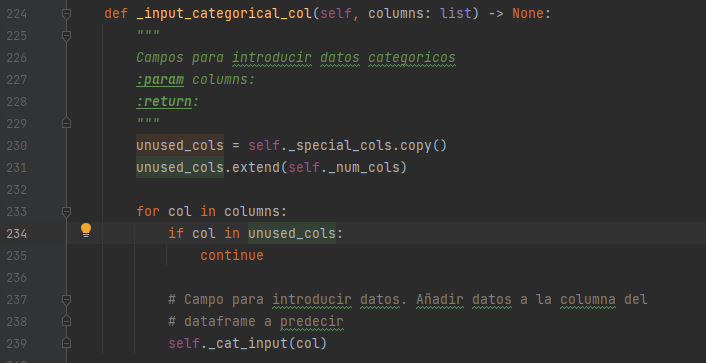
En \_encode\_values se codifican los valores categóricos según su codificación esperada.



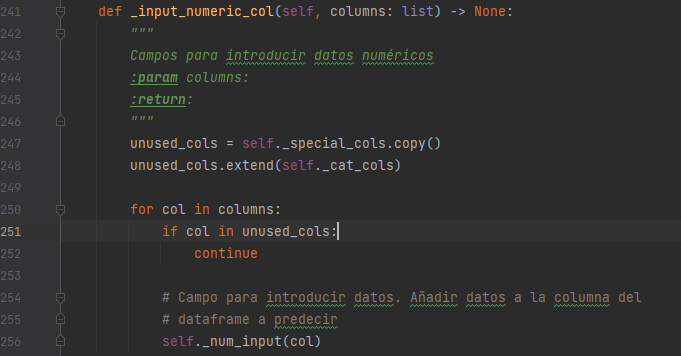
En \_get\_atributes se obtienen los atributos de las columnas categóricas.



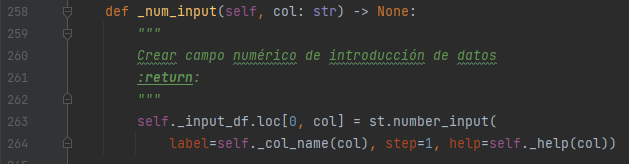
En \_input\_categorical\_col se establecen los campos para introducir datos categóricos.



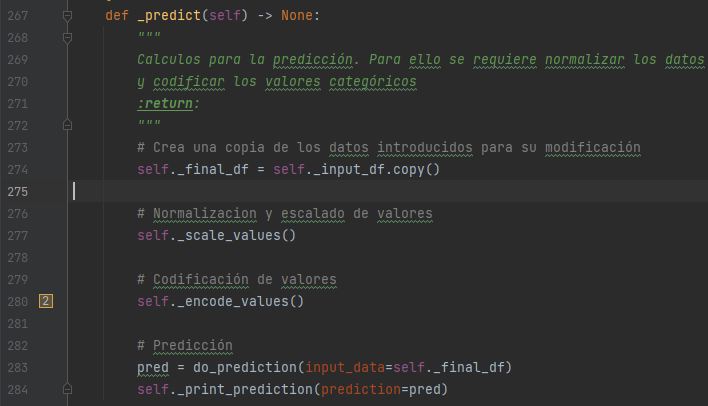
En \_input\_numeric\_col se establecen los campos para introducir datos numéricos.



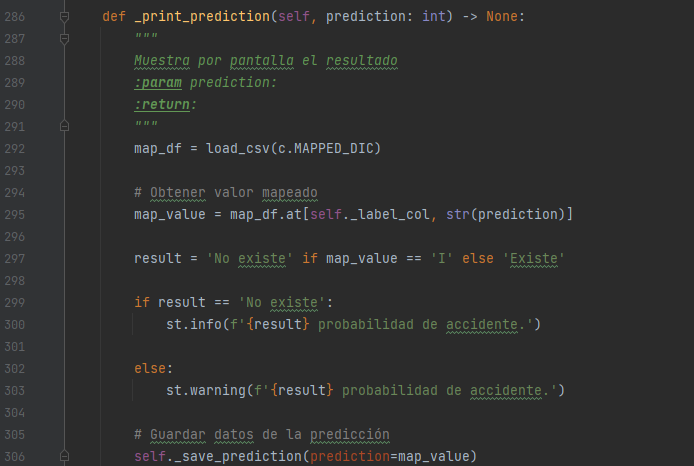
En \_num\_input se establecen los campos para introducir datos numéricos.



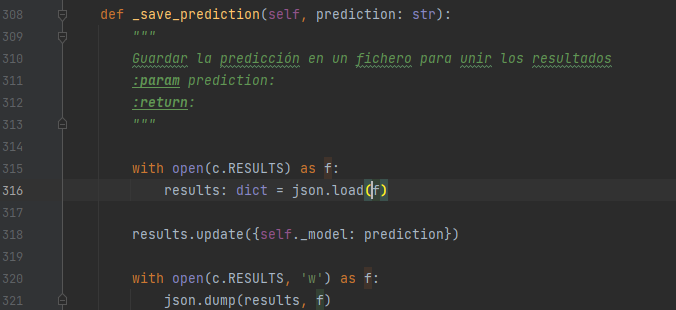
En \_predict se realizan los cálculos para la predicción. Para ello se requiere normalizar los datos y codificar los valores categóricos



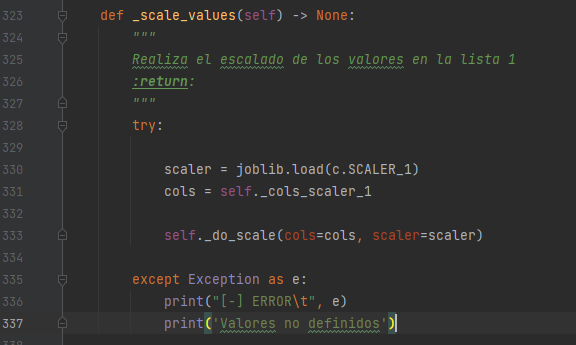
En \_print\_prediction se muestra por pantalla el resultado.



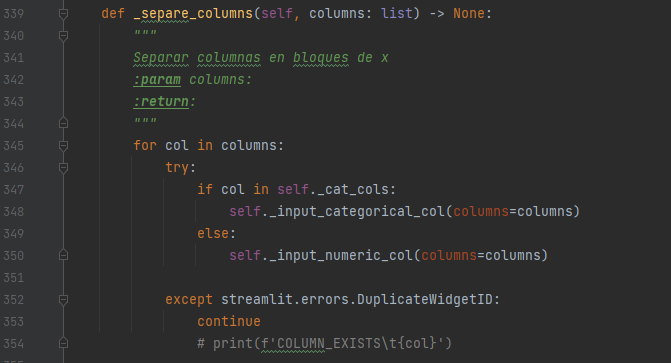
Con \_save\_prediction se guarda la predicción en un fichero para unir los resultados.



En \_scale\_values se realiza el escalado de los valores.

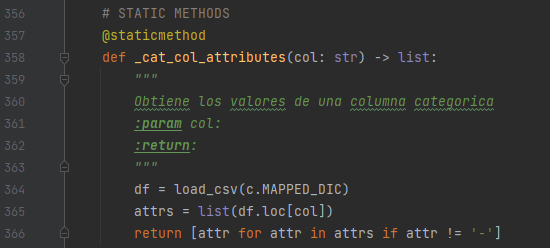


En \_separe\_columns se separan las columnas en bloques.

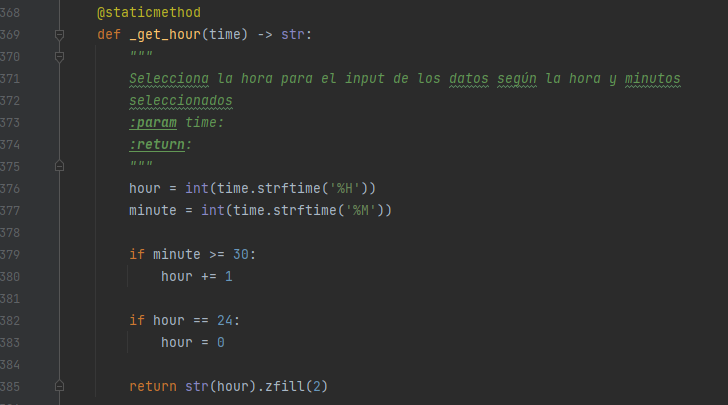


##### Métodos estáticos.

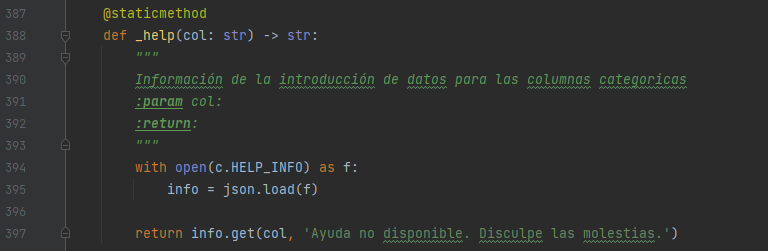
En \_cat\_col\_atributes se obtienen los valores de una columna catregórica.



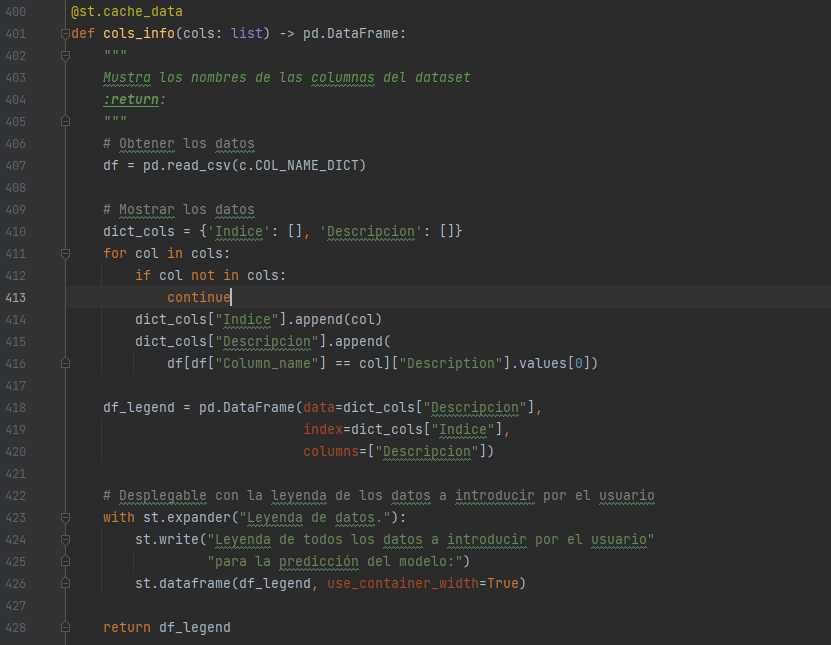
En \_get\_hour se selecciona la hora para el input de los datos según la hora y minutos seleccionados.



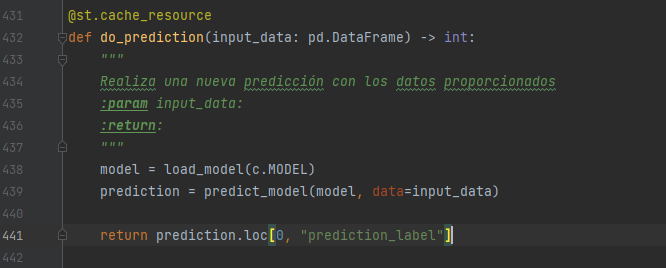
En \_help muestra la información de la introducción de datos para las columnas categóricas.



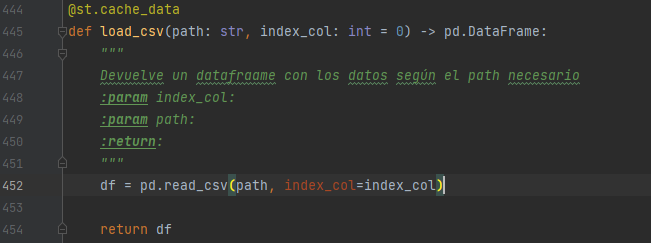
En cols\_info se muestra los nombres de las columnas del dataset



En do\_prediction se realiza una nueva predicción con los datos proporcionados.

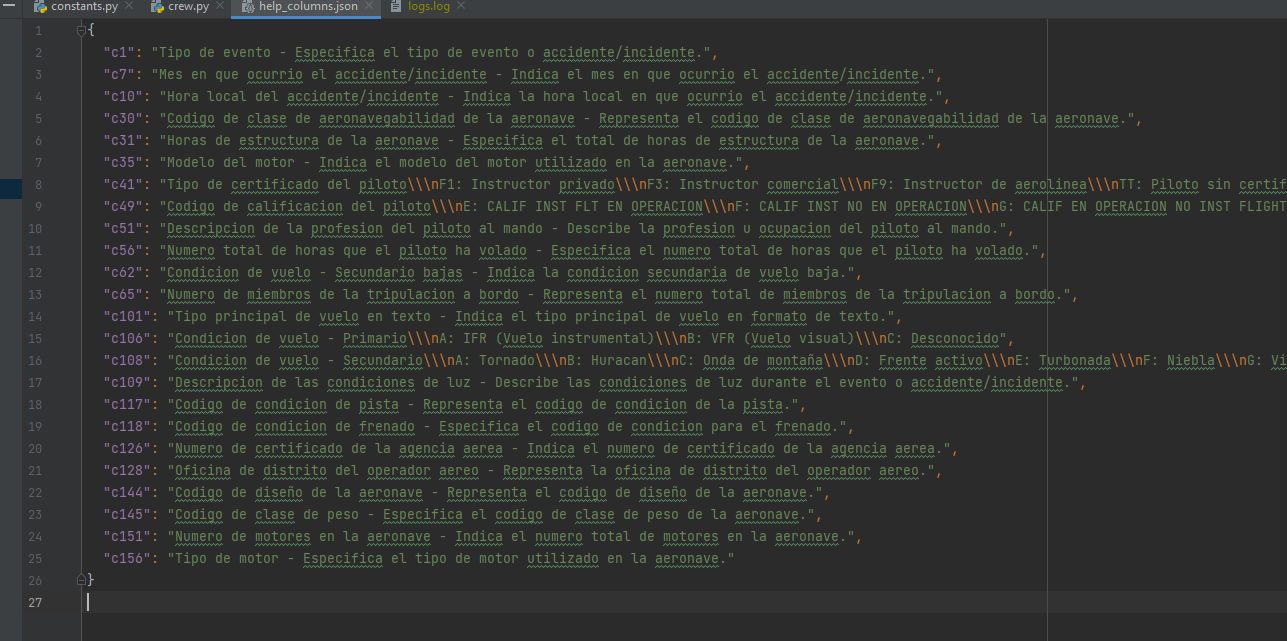


Por último en load\_csv devuelve un dataframe con los datos según el path necesario.



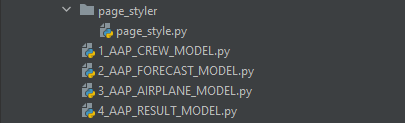
#### 3.2.3.-Help\_columns

Es un json donde se establece los valores de las columnas para el apartado de ayuda.



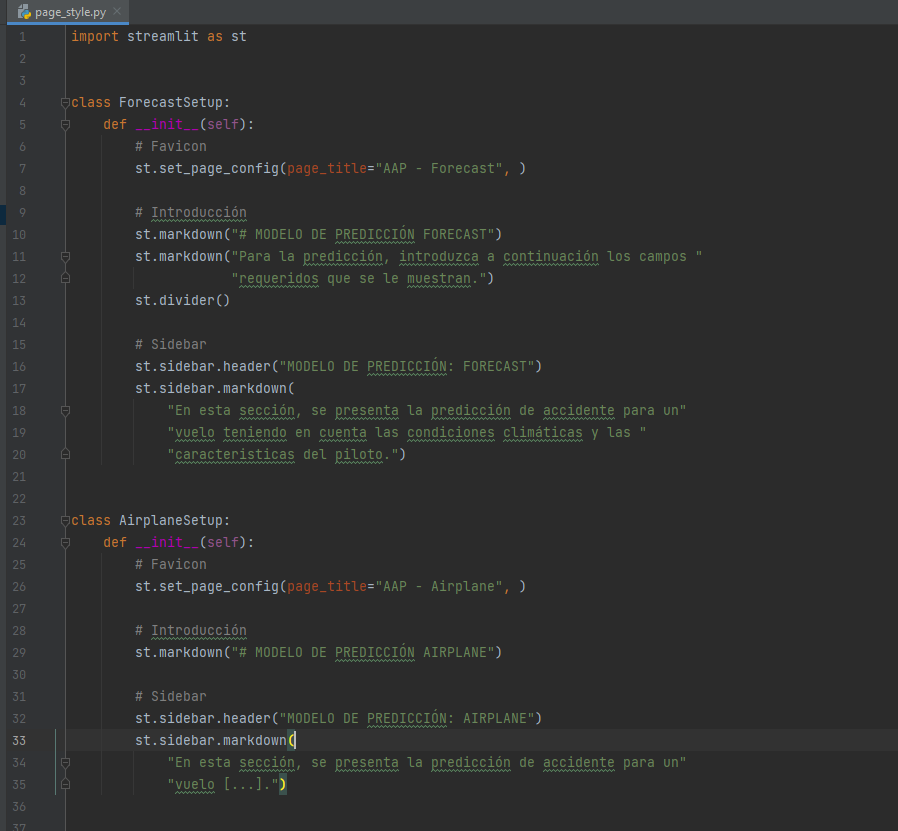
### 3.3 Directorio pages.

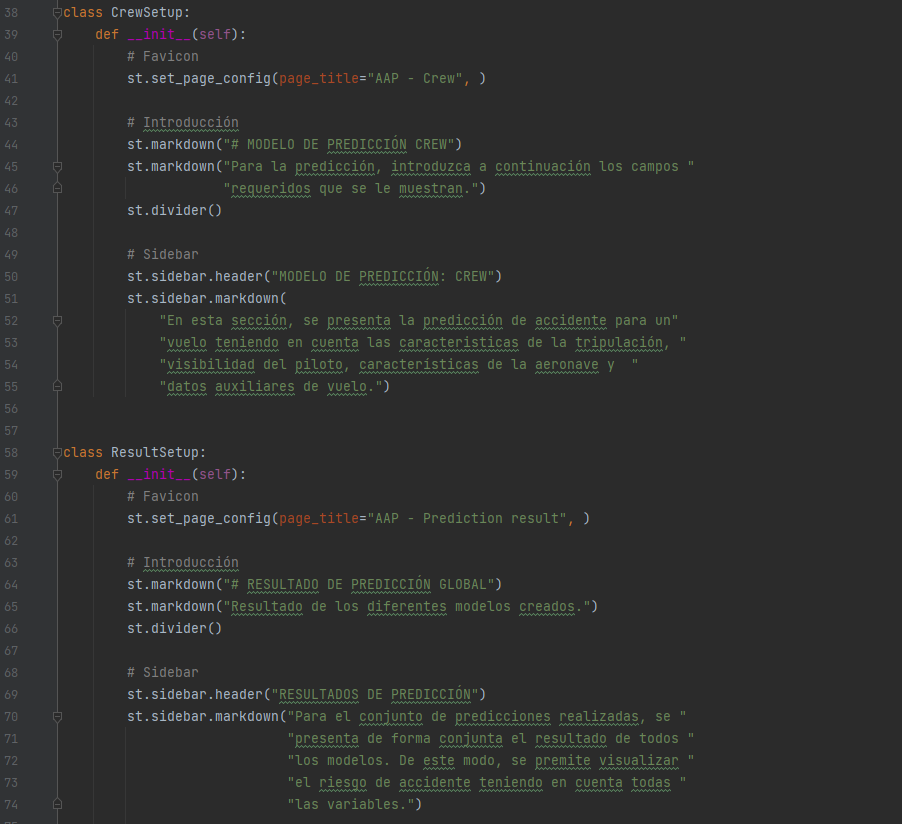
Contiene las páginas de cada sección.



##### Page\_styler.py

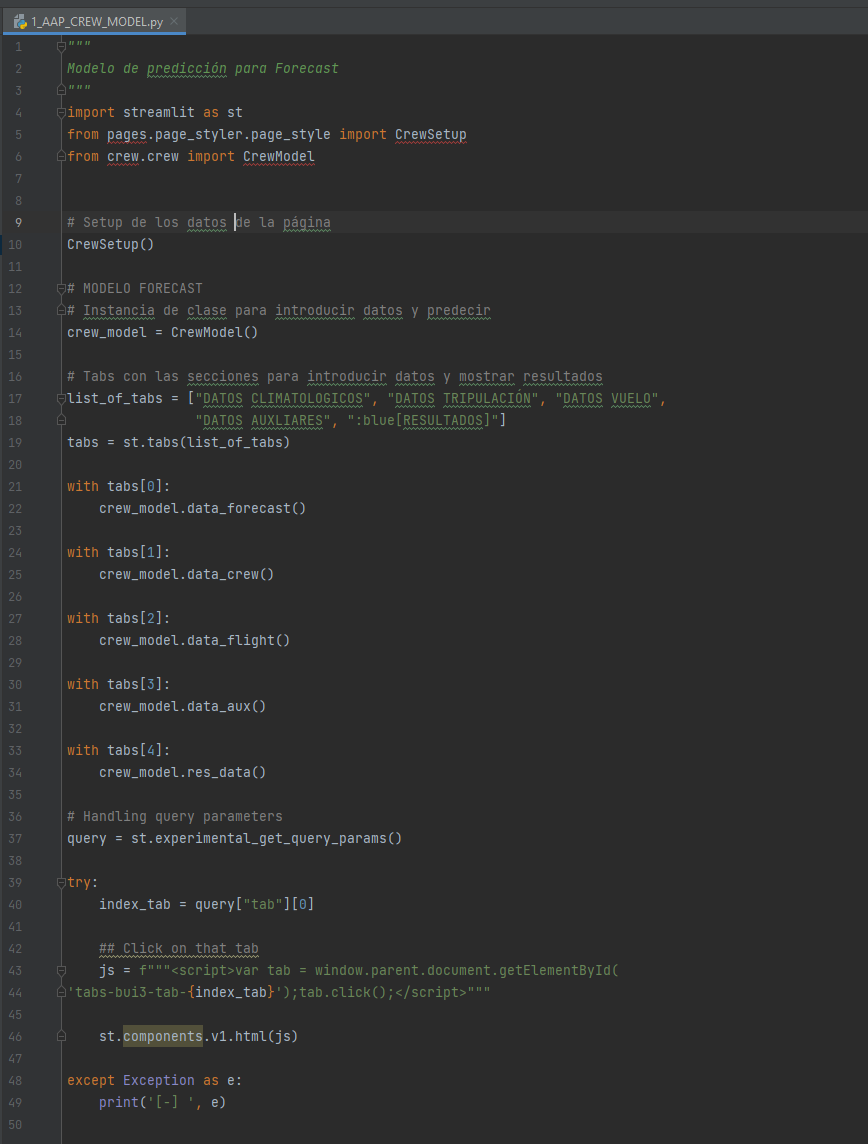
En este archivo se estable la descripción de cada sección para el menú izquierdo de navegación





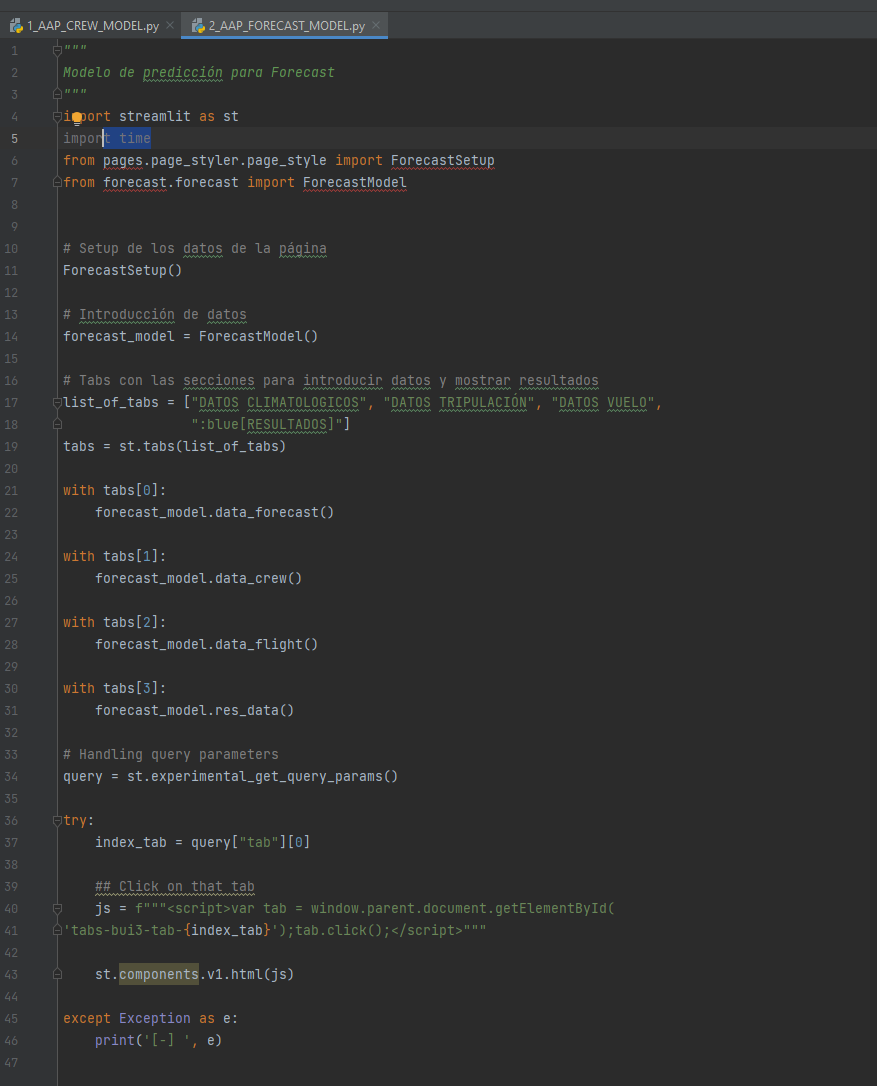
##### 1\_AAP\_CREW\_MODEL.py

En este archivo se configura la página de crew generando 5 tabs.



##### 2\_AAP\_FORECAST\_MODEL.py

En este archivo se configura la página de forecast generando 4 tabs.



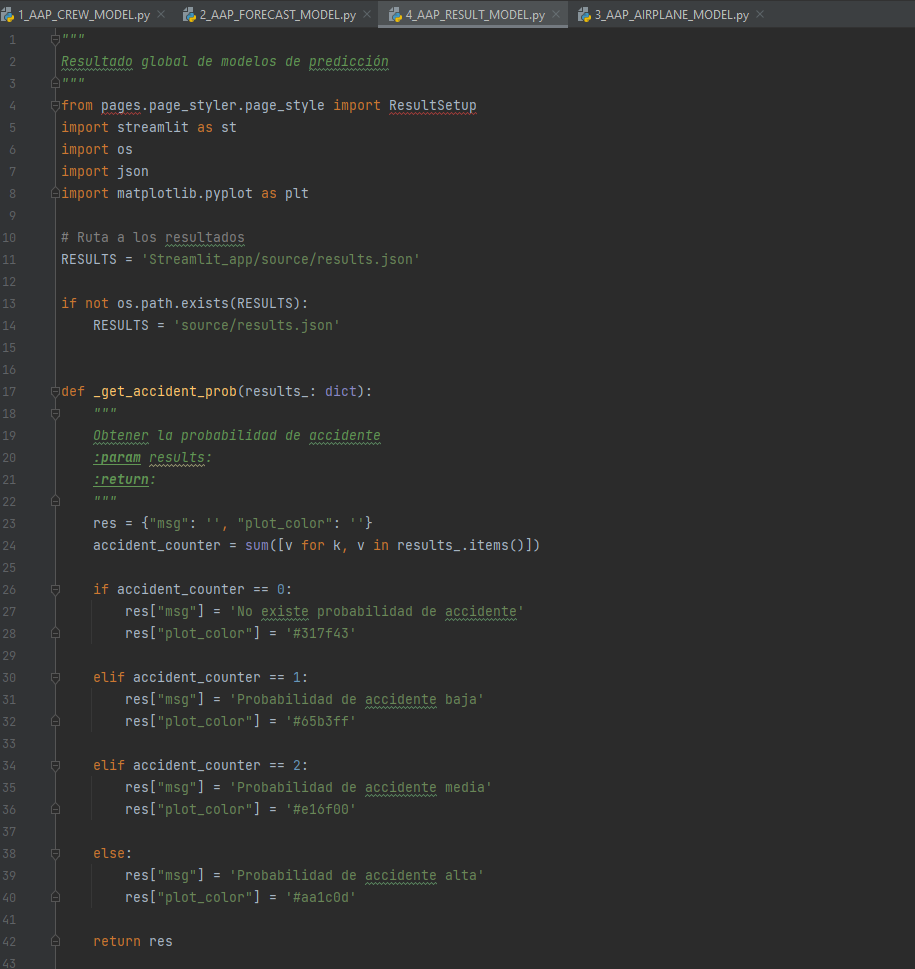
##### 3\_AAP\_AIRPLANE\_MODEL.py(PABLO)

##### 4\_AAP\_RESULT\_MODEL.py

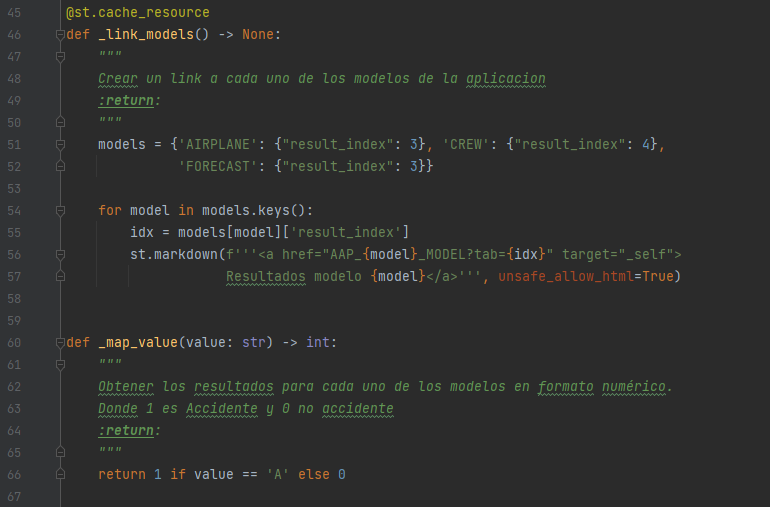
En este archivo se procesan los resultados obtenidos por cada modelo y se visualizan.

Primeramente se carga “result.json”el cual tiene almacenado los resultados de cada modelo.

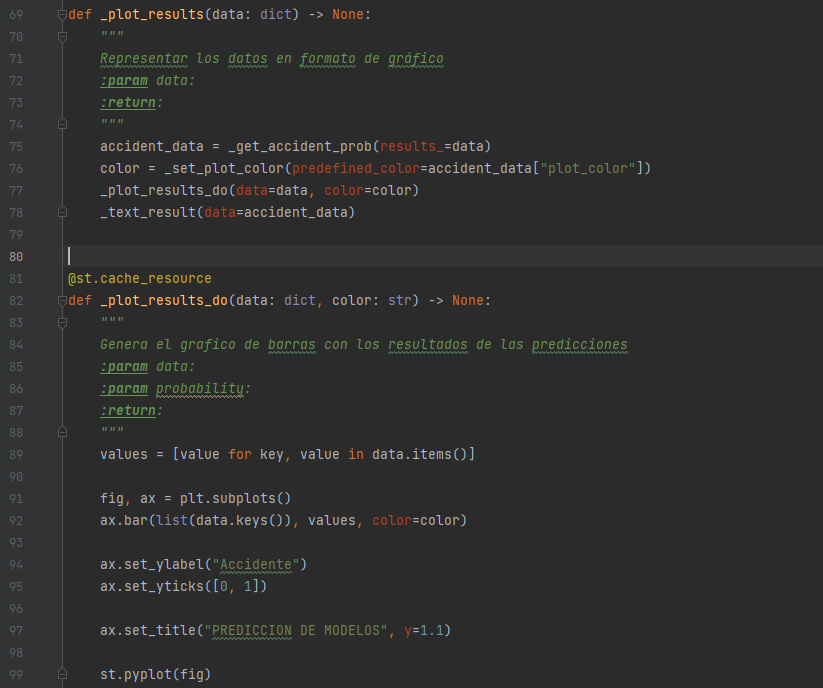
Posteriormente se establece las condiciones de color y los mensajes para cada una de las posibilidades en el método \_get\_accident\_prob.



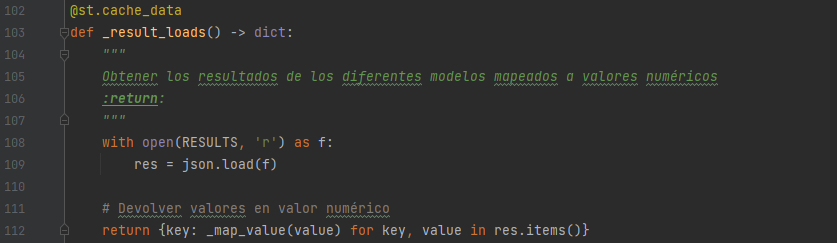
Seguidamente se crean estos métodos para crear links a cada uno de los modelos para mayor comodidad y se mapean los resultados.



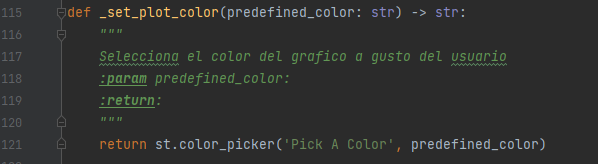
Con estos métodos se generan los gráficos según los resultados obtenidos.



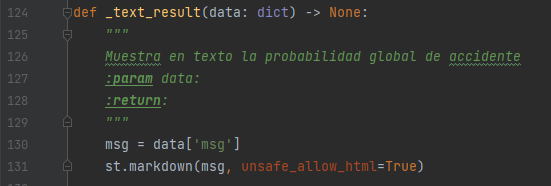
En este método se cargan los resultados



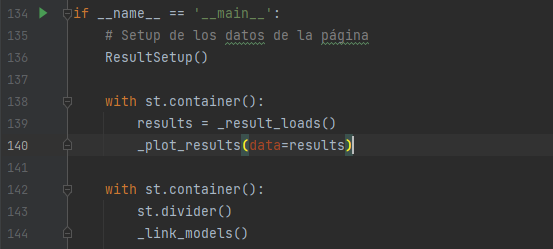
En este método se da la posibilidad de que el usuario selecciona el color del gráfico a su gusto



En este método se muestra la probabilidad global de accidente

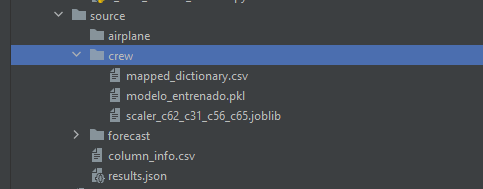


Por ultimo establecemos el Setup de los datos de la página



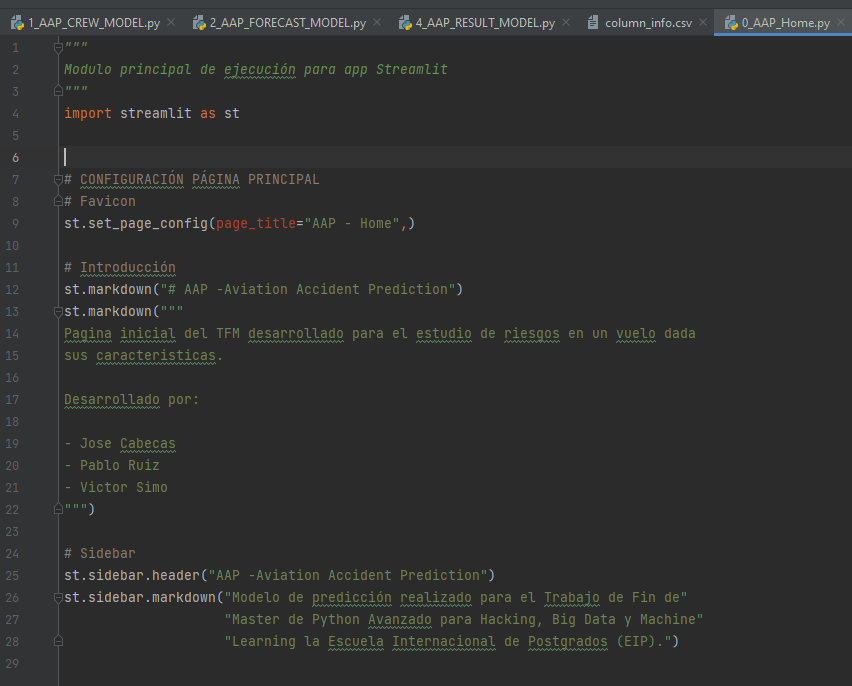
### 3.4.-Source(INCOMPLETO)

En el directorio Source tenemos almacenado los archivos que contienen los escalados, las codificaciones de label\_encoder y OneHotEncoder, los modelos ,el json de results.json mencionado anteniormente y column\_info.csv el cual contiene los nombres de las columnas.



### 3.5.-0\_AAP\_Home.

En este archivo se configura la página principal de la app donde se muestra un resumen del proyecto.



## 4.-Union de las partes y prueba en local

## 5.-Subida a la SteamCloud

# Evaluación de los resultados (¿o quiere decir que evaluemos que variables hacen que existan mayor riesgo de accidentes? , se podría usar la matriz de coorelación usada cuando se generó el modelo)

En primer lugar, se realizó un análisis exploratorio detallado de los datos de accidentes de aviones. A través de este análisis, se identificaron patrones y tendencias significativas en los datos, lo que permitió comprender mejor la naturaleza de los accidentes y los factores que podrían contribuir a su ocurrencia.

Posteriormente, se procedió al preprocesamiento de los datos, abordando cuestiones como la limpieza de datos, el manejo de valores faltantes y la normalización de variables. Este proceso fue fundamental para garantizar la calidad y consistencia de los datos utilizados en los modelos de predicción.

Se construyeron modelos de predicción individuales para cada una de las agrupaciones de columnas de datos (tripulación, aeronave y datos climatológicos). Se aplicaron diversos algoritmos de aprendizaje automático para entrenar y ajustar estos modelos, con el objetivo de predecir la probabilidad de accidentes de aviones.

A través de la implementación de una interfaz de usuario utilizando Streamlit, se permitió a los usuarios introducir los datos relevantes en cada uno de los modelos de predicción. Esto facilitó la interacción con el sistema y brindó una experiencia amigable al usuario.

Al integrar los resultados de los modelos individuales y realizar un análisis global, se obtuvieron conclusiones significativas sobre la probabilidad de ocurrencia de accidentes de aviones. Se pudo establecer una visión más precisa y confiable al considerar la combinación de los resultados individuales. Esto permitió identificar patrones comunes y factores de riesgo que influyen en la probabilidad de accidentes.

En resumen, los resultados obtenidos de nuestro TFM demostraron la eficacia de los modelos de predicción construidos utilizando técnicas de aprendizaje automático y datos relevantes. La integración de estos modelos y el análisis global de los resultados proporcionaron una comprensión más profunda de la probabilidad de ocurrencia de accidentes de aviones, lo que contribuye a la mejora de la seguridad en la industria de la aviación. Los aspectos destacados de esta evaluación incluyen la identificación de patrones, la validación de los modelos y la identificación de factores clave para la prevención de accidentes. Estos hallazgos son fundamentales para tomar medidas preventivas y mejorar la seguridad en el sector aeronáutico.

# Conclusiones

A lo largo de nuestro estudio, hemos logrado desarrollar un sistema de predicción de accidentes de aviones utilizando técnicas de aprendizaje automático y datos relevantes. Este sistema representa una herramienta efectiva para evaluar la probabilidad de ocurrencia de accidentes y contribuir a la mejora de la seguridad en la industria de la aviación.

Nuestro análisis exploratorio detallado de los datos de accidentes de aviones nos ha permitido identificar patrones y tendencias significativas. Esto ha brindado una comprensión más profunda de los factores que pueden influir en la ocurrencia de accidentes y, por lo tanto, ha proporcionado una base sólida para tomar medidas preventivas más efectivas.

El preprocesamiento exhaustivo de los datos ha sido esencial para asegurar la calidad y consistencia de los mismos. Mediante tareas de limpieza, manejo de valores faltantes y normalización de variables, hemos mejorado la precisión de los modelos de predicción y hemos aumentado la confiabilidad de los resultados obtenidos.

Nuestros modelos de predicción, construidos para cada agrupación de columnas de datos (tripulación, aeronave y datos climatológicos), han demostrado ser eficaces para predecir la probabilidad de accidentes de aviones de manera individual. Esta aproximación nos ha permitido obtener una visión detallada de los factores específicos que pueden influir en la ocurrencia de accidentes, lo que resulta fundamental para la prevención y toma de decisiones informadas.

La implementación de una interfaz de usuario intuitiva utilizando Streamlit ha mejorado la interacción con el sistema y ha facilitado el uso por parte de los usuarios. Esto ha permitido introducir los datos relevantes en cada modelo de predicción de manera sencilla y ha mejorado la experiencia global del usuario al utilizar nuestro sistema.

Al integrar los resultados individuales de los modelos de predicción y realizar un análisis global, hemos obtenido una visión más precisa y confiable de la probabilidad de ocurrencia de accidentes de aviones. Esta evaluación conjunta, considerando la combinación de los resultados individuales, nos ha permitido identificar la importancia de factores múltiples y obtener una evaluación más sólida de los riesgos asociados.

En resumen, a través de nuestro TFM hemos logrado desarrollar un sistema de predicción de accidentes de aviones que contribuye a mejorar la seguridad en la industria de la aviación. Nuestros resultados respaldan la eficacia de los modelos de predicción, la relevancia de los datos utilizados y la importancia de aplicar técnicas de aprendizaje automático para tomar decisiones informadas en el ámbito de la seguridad aérea.

# Referencias

La bibliografía irá al final del trabajo, ordenada alfabéticamente.

# Anexos TFM

Se trata de una sección que se utiliza para adjuntar la información en detalle.

En este caso, el fichero que contenga el código de su trabajo. Suele añadirse cualquier documento que por tamaño no se haya podido exponer en el trabajo y que aporte algún conocimiento o recurso adicional al resto del proyecto.