Trabajo Práctico Integrador

El trabajo final de la materia tiene cómo objetivo realizar un EDA. Para este deberán:

● Identificar un problema a analizar o desarrollar un caso de negocio y definir los datos a utilizar para el análisis.

● Consumir información desde un archivo o desde una API, limpiar el dataset, transformar los datos necesarios y realizar el EDA que les permita extraer conclusiones.

● Detallar conclusiones y discoveries en el README del repositorio.

● Por último, proveer una respuesta final o recomendación al problema / cuestionamiento inicial planteado.

Formato de la entrega:

● Link del repositorio Git donde se encuentre el trabajo realizado. El mismo deberá contener todo el proyecto con sus respectivas conclusiones, incluyendo el código utilizado y el dataset elegido.

Rúbrica de Evaluación:

● Código

○ Estructura y organización del código

○ Parametrización/reusabilidad de componentes

○ Dificultad en el proceso de curado (cleaning)

● Dataset

○ Dificultad del dataset

○ Claridad de los datos

● Analytics

○ Planteo del análisis y caso de negocio

○ Delimitación del scope

○ Calidad y cantidad de conclusiones elegidas

○ Variedad de recursos utilizados ○ Nuevos recursos

○ Próximos pasos

● Bonus:

○ Utilización de Git (commits, PRs, estructura, README)

○ Utilización y exploración de modelos predictivos

Obtener data de una API o cargar el archivo si es un excel

Limpieza

Transformación de los datos

Categorizarlo bien

Poder hacer el EDA

entregables: codigo y detalle de las conclusiones a las que se llego

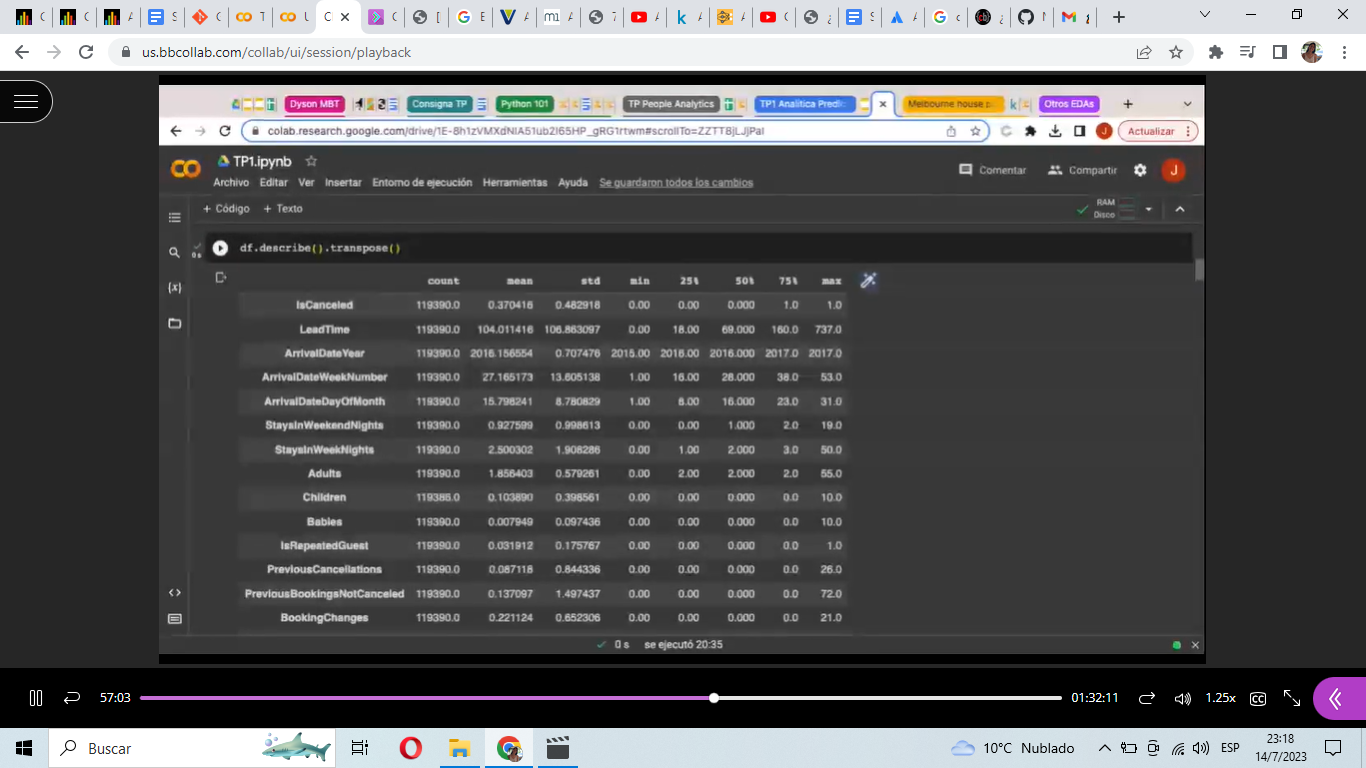
Repositorio en GIt

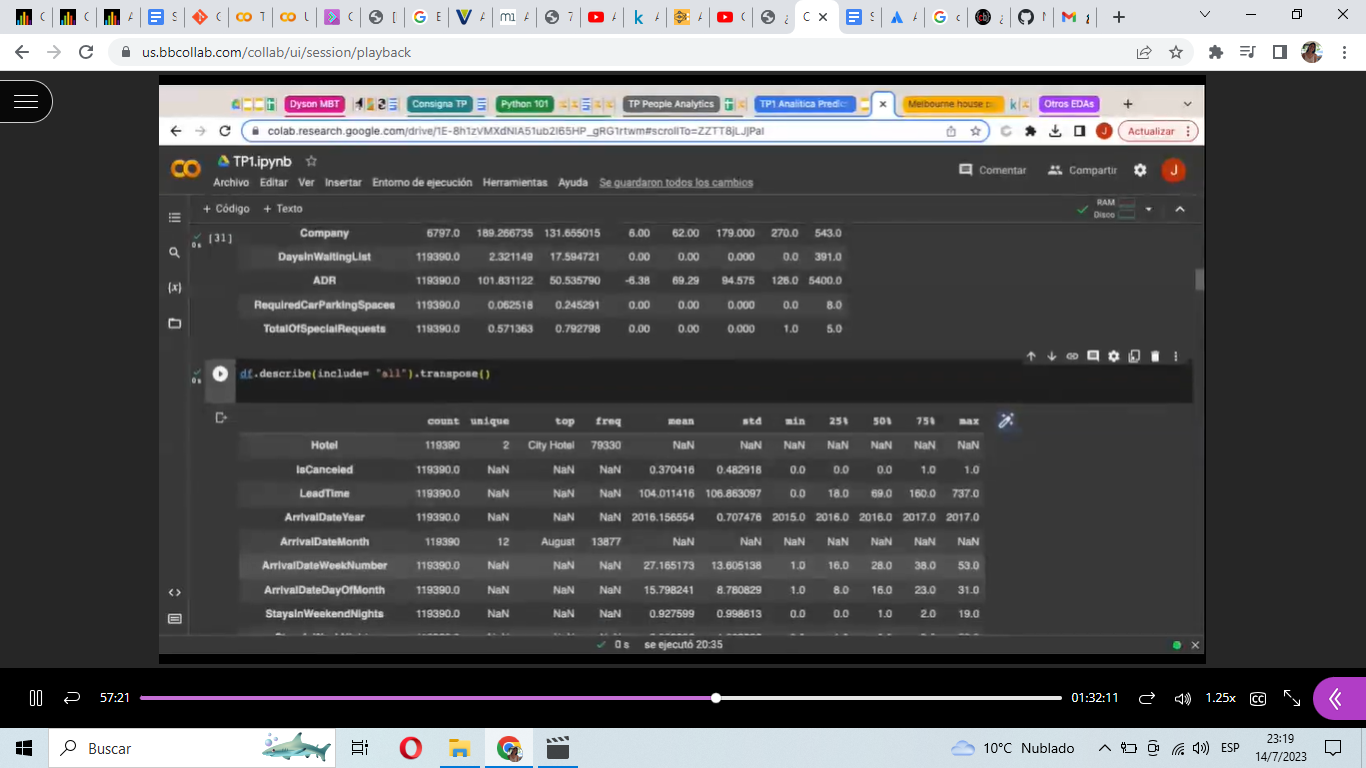
codigo con la limpieza la visualización etc.

En el GIT

una carpeta de diga DATA y ahi meto los datos

Limpiarlos—- sacar espacios de los nombres

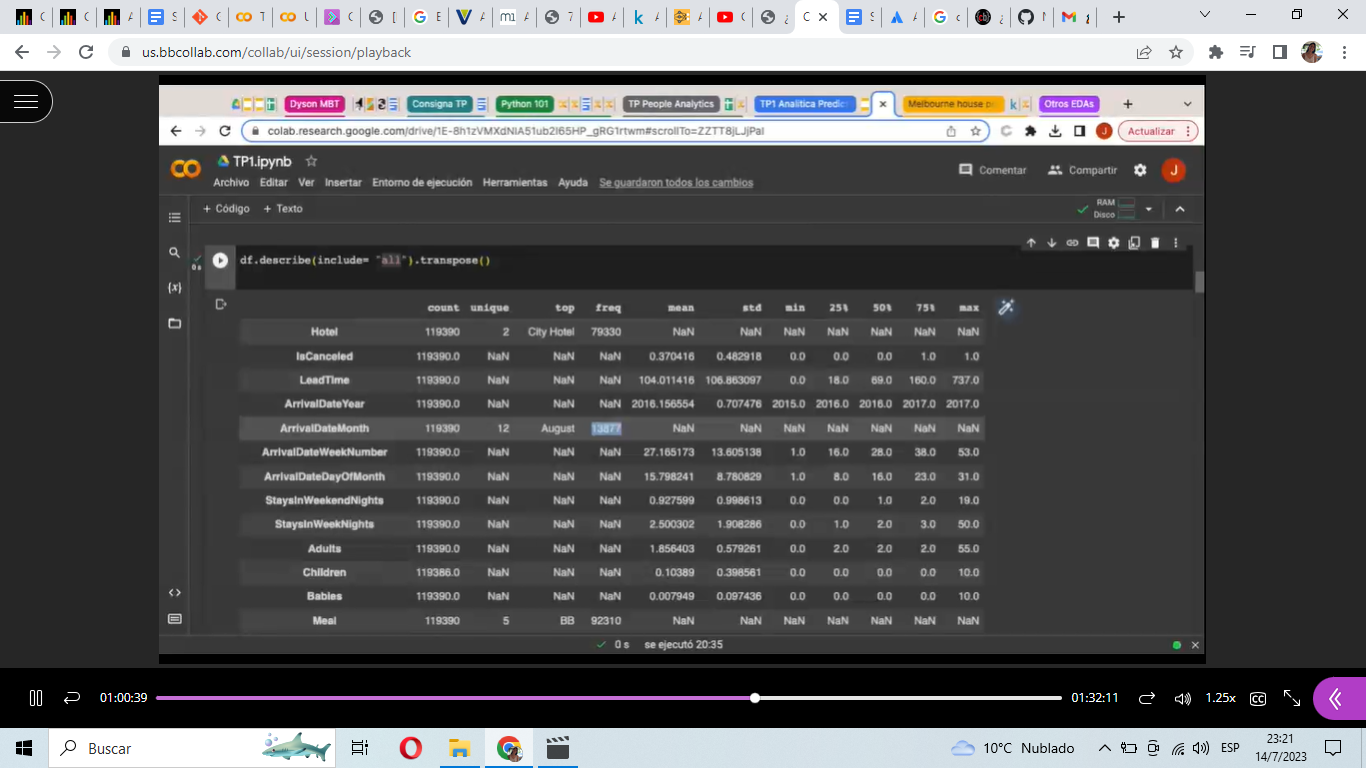


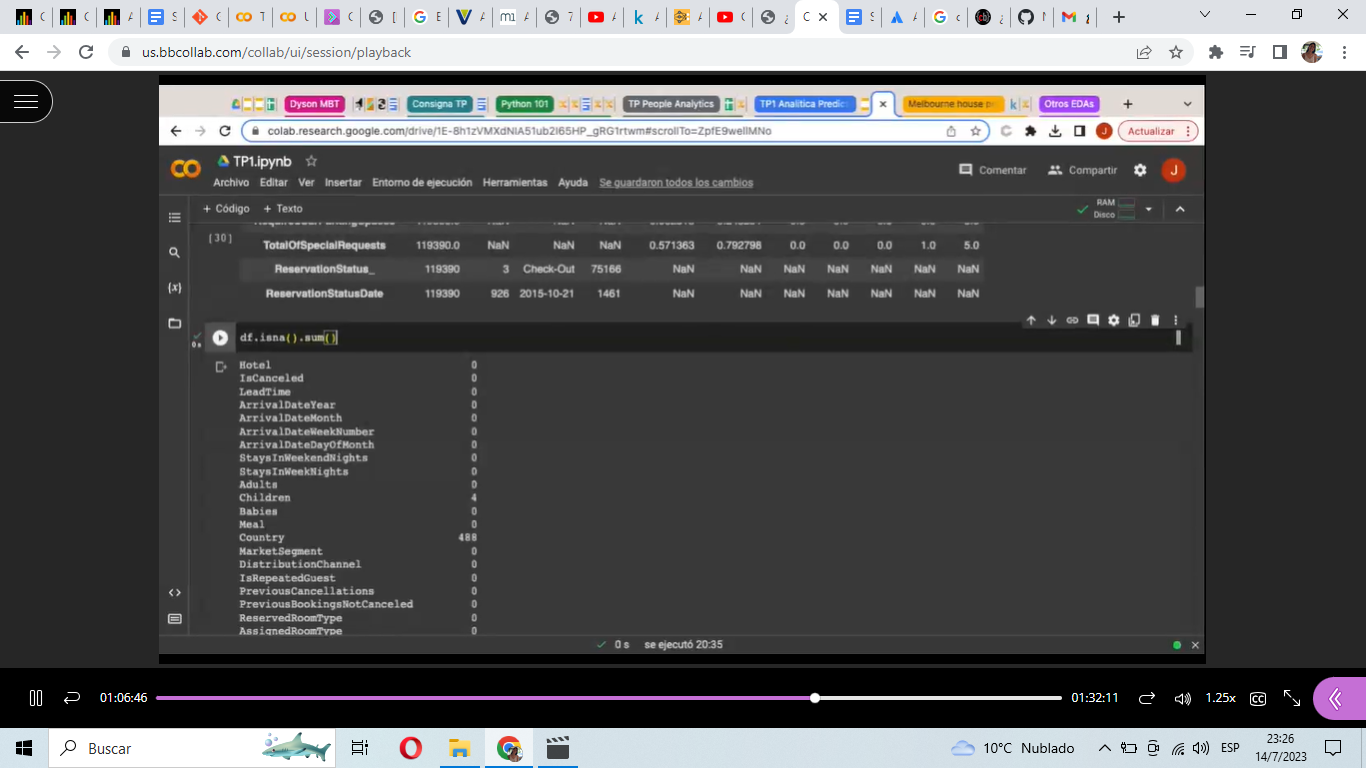


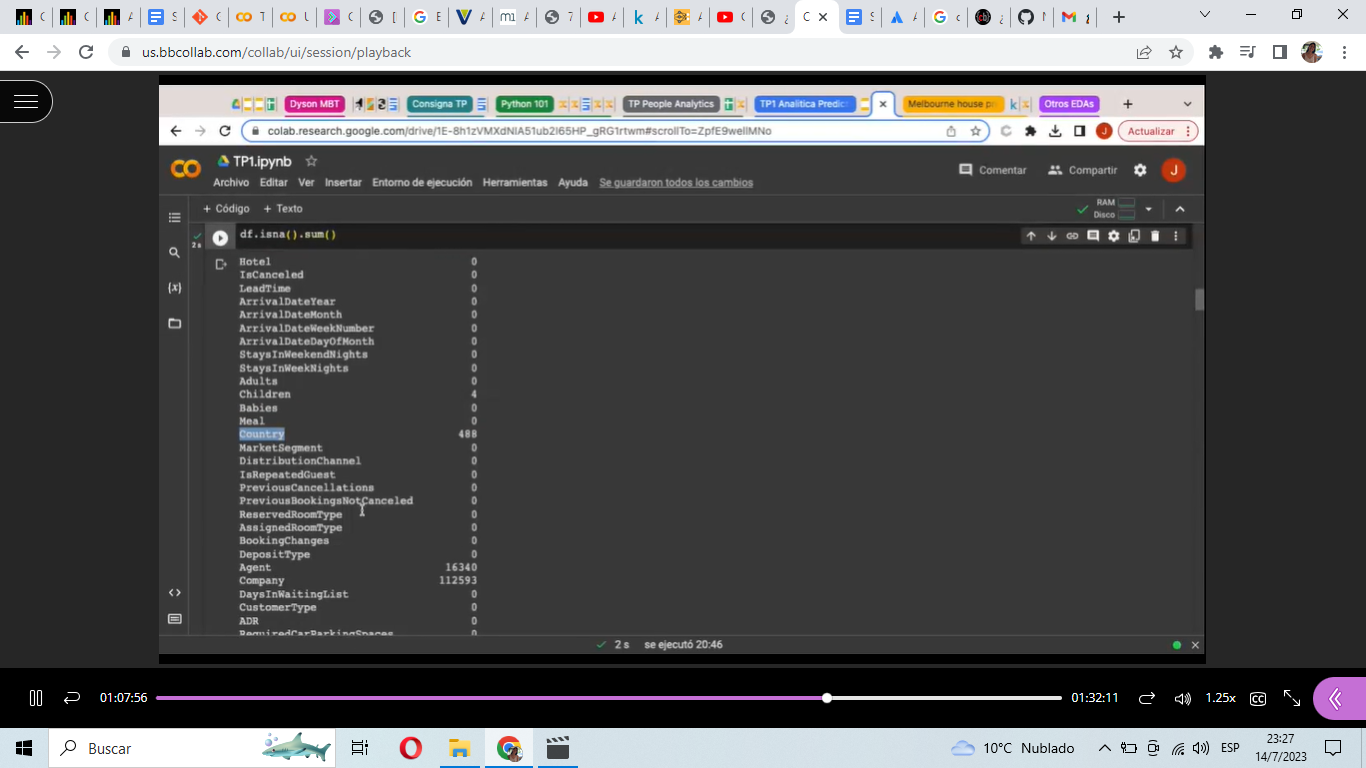
Menos columnas ahora. Te las da ordenadas las columnas.

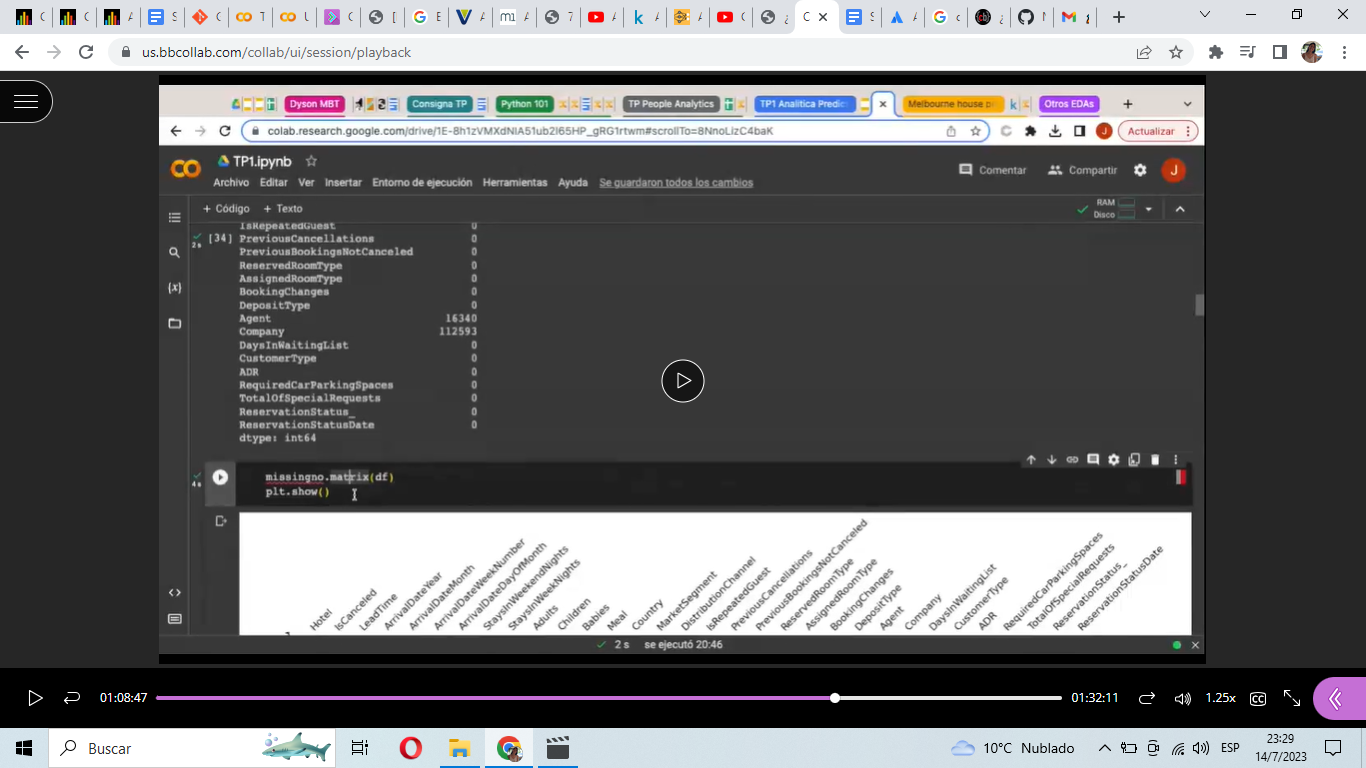
describe me da solamente las numericas.

inlude all trae las numericas como las categoricas



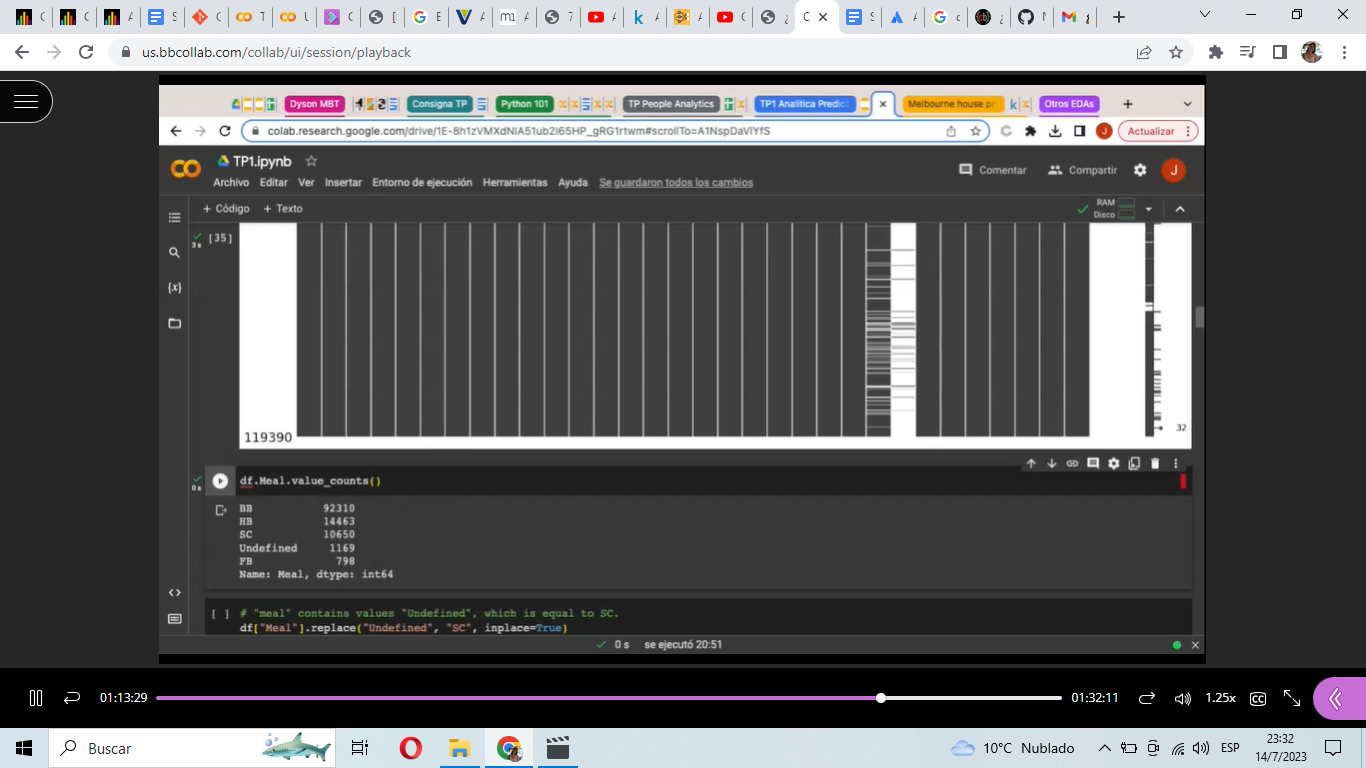


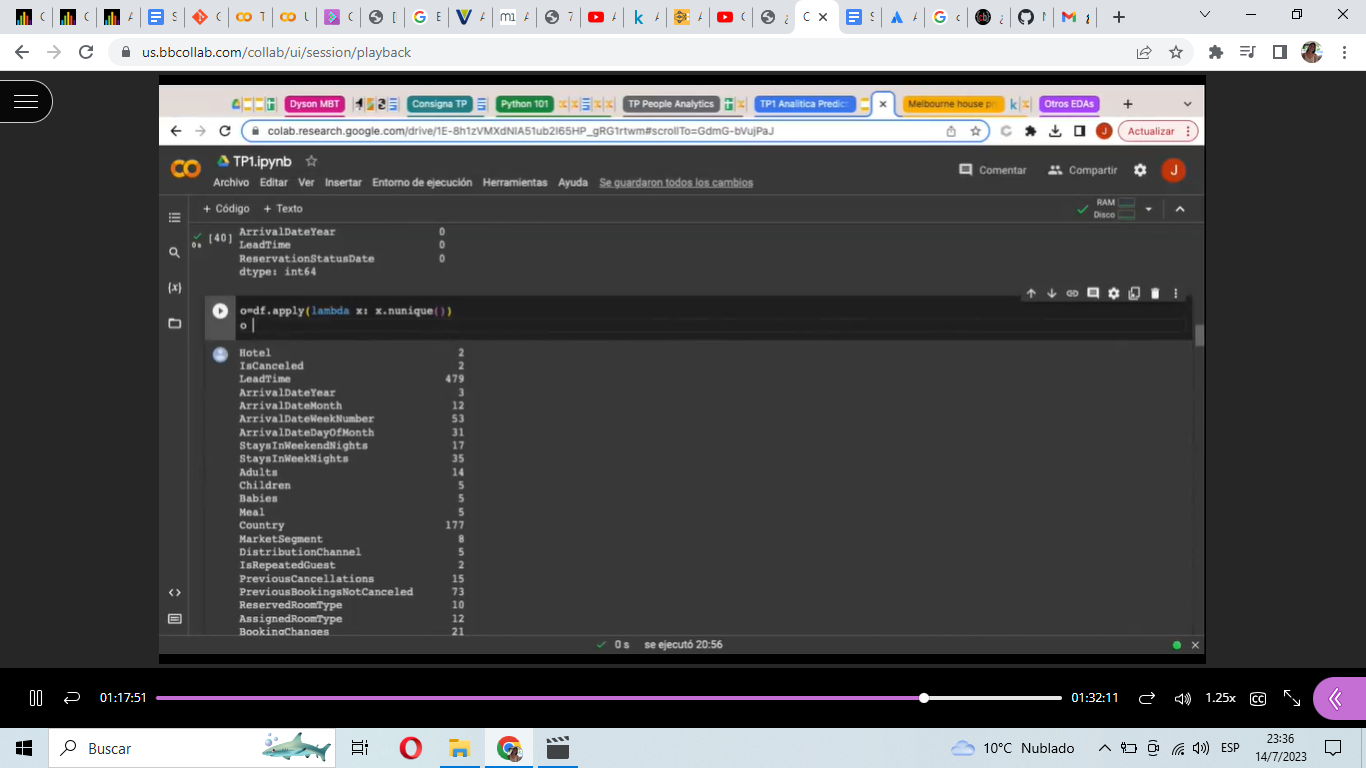


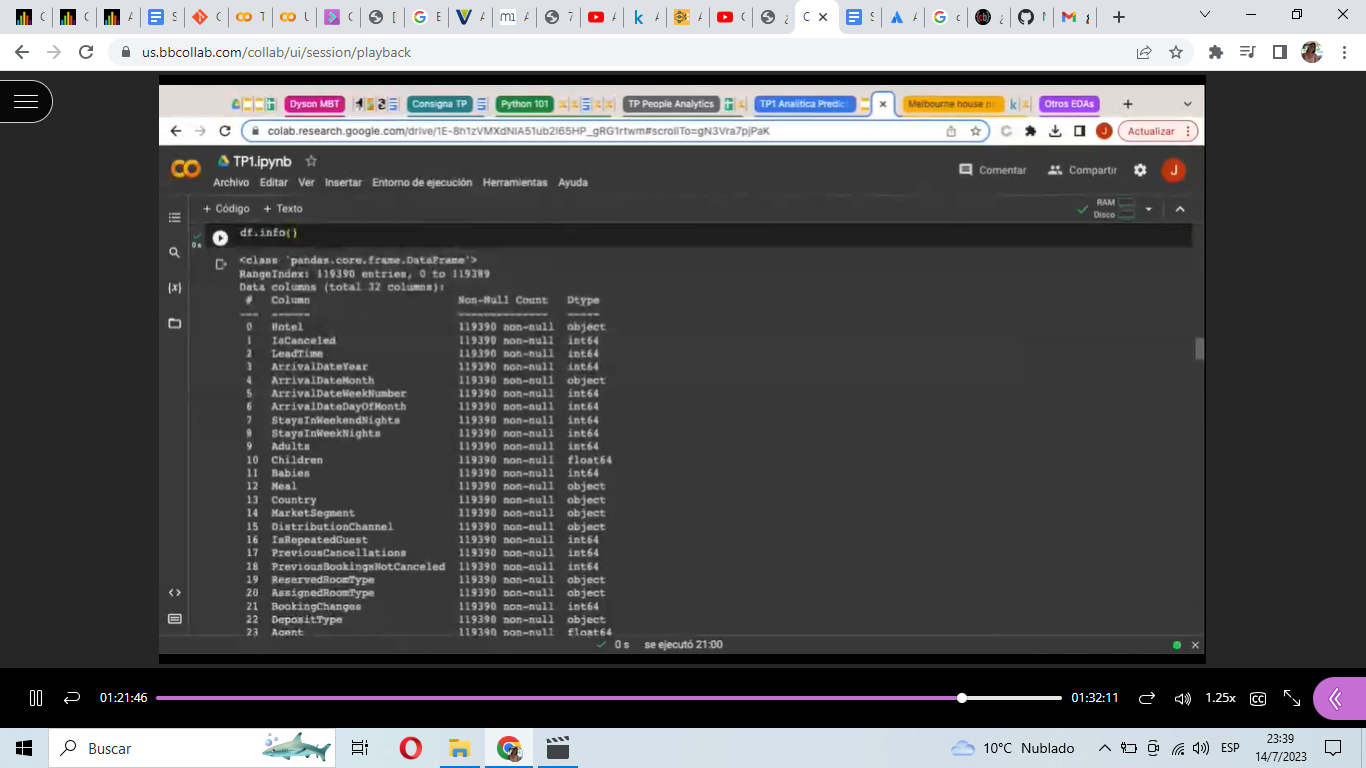


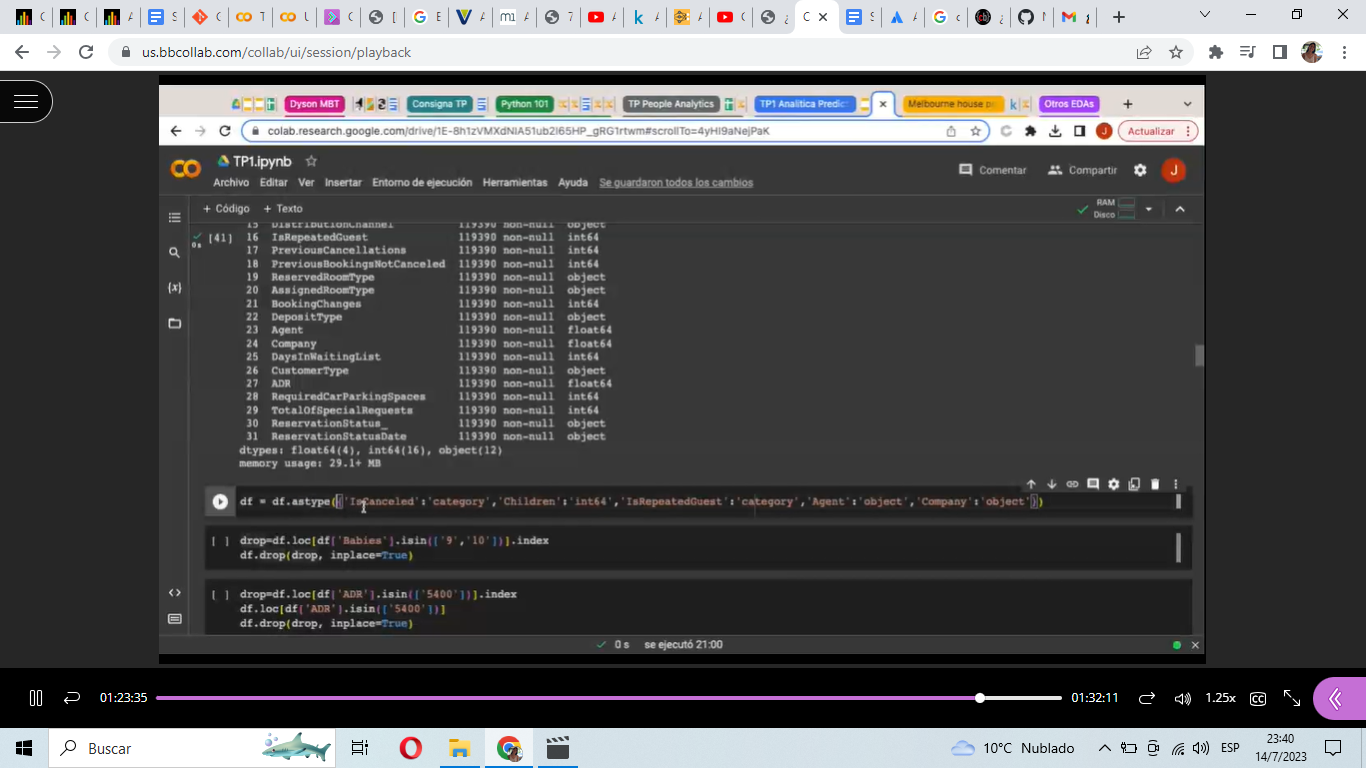
completamente vacio

Otro el dato esta completo pero con vaciopor ejemplo

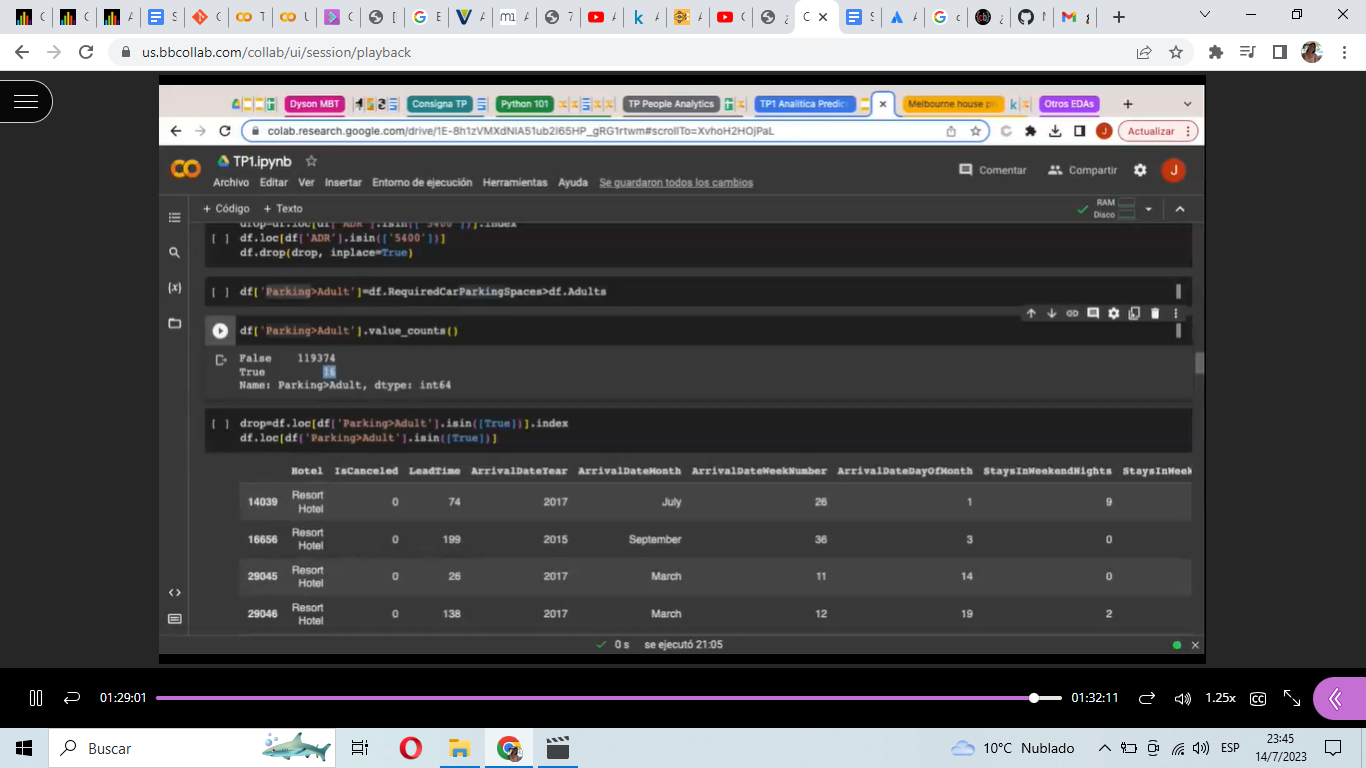


cuantos valores unicos tengo para cada columna.

Podria averlo puesto antes para ver los nulls.



paso iscancelled a categoria.



TP FINAL

Análisis de la tasa de abandono de clientes:

El abandono de clientes es un fenómeno bien conocido en muchas empresas. De hecho, en prácticamente todos y cada una de los que se pueda imaginar, ya que cada negocio tiene clientes y les vende productos y servicios que éstos pueden dejar de comprar o consumir. Lo que se vende es irrelevante para el hecho de que el cliente siempre está presente. Incluso si el negocio es B2B, hay clientes, sólo que esta vez son otros negocios. Por lo tanto, el cliente es siempre la piedra angular de cada negocio y perderlos es terrible sin importar lo que vendamos.

Vamos a ver cómo podemos reducir la tasa de abandono de clientes gracias a Machine Learning.

El aumento de las tasas de retención de clientes en un 5% aumenta los beneficios entre un 25% y un 95%.

Cuesta cinco veces más atraer a un nuevo cliente que mantener uno existente.

¿Quién?

Sabiendo qué clientes son más propensos a irse podemos diseñar campañas de retención y evitar su abandono.

¿Por qué?

Estas campañas de retención deben centrarse en áreas clave de nuestro negocio. Conocer las razones por las que los clientes que probablemente abandonan nuestro negocio nos ayudará a dirigirnos a los perfiles adecuados y diseñar acciones.

(en este caso, un clasificador binario ya que tenemos 2 clases –‘abandona’ y ‘no abandona’) para aprender de los datos ya que se muestran menos casos de la clase ‘abandona’.

Tener demasiadas características cuando podemos llegar a un modelo lo suficientemente bueno (recuerda, nunca tendremos un modelo perfecto) con menos características es una pérdida de tiempo y potencia de cómputo.

Esto es especialmente crítico si pensamos en una de las preguntas clave de nuestro enfoque de abandono: “¿por qué mis clientes me abandonan?“. Si la respuesta se basa en cientos de factores, será extremadamente difícil hacerla comprensible para nuestros colegas o clientes al diseñar campañas de retención

Tener modelos más simples (siempre y cuando funcionen lo suficientemente bien) suele ser mejor que los modelos demasiado complicados, ya que normalmente generalizan mejor con nuevos datos (funcionarán mejor contra los datos no vistos).

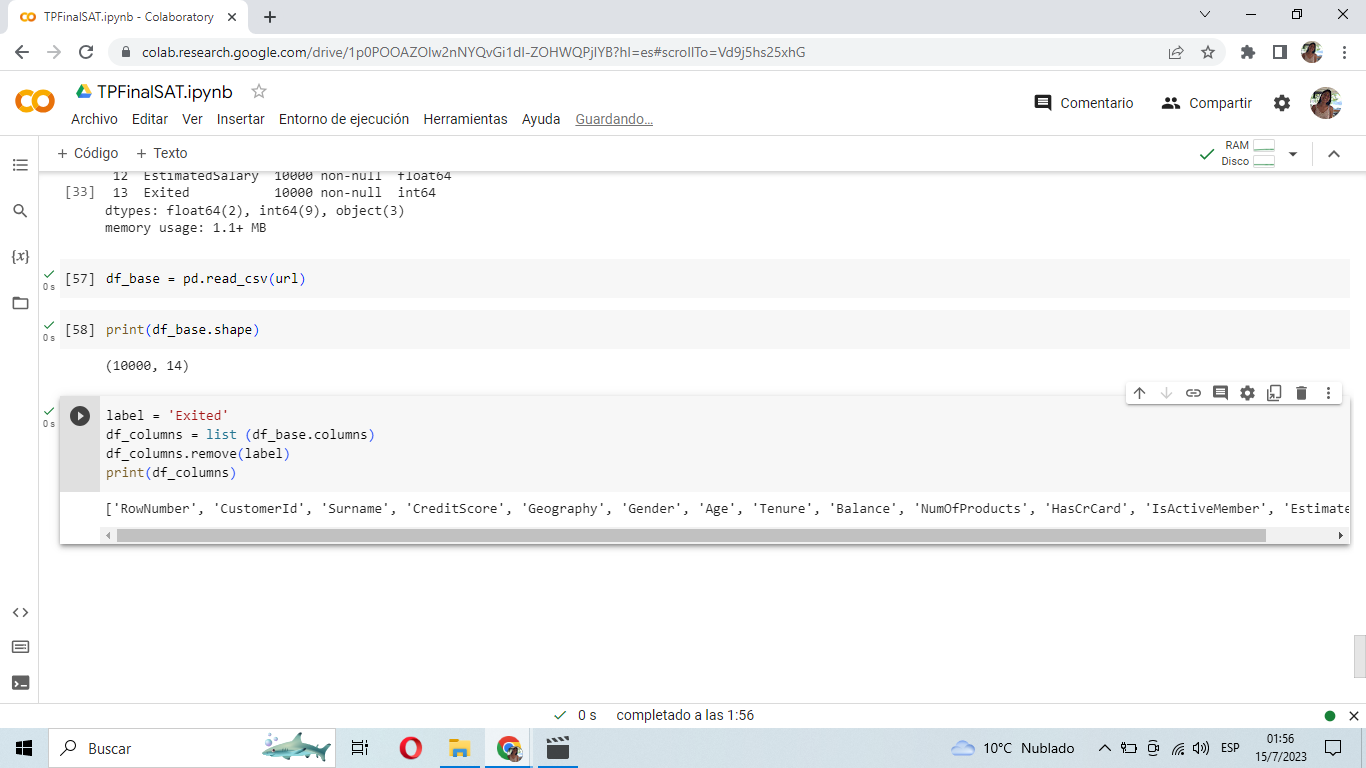
Lo primero que haremos será obtener nuestro dataset de ejemplo. Hay muchos datasets sobre abandono de clientes

este caso lo tendremos en formato CSV. Se trata de un dataset sobre dato de abandono de clientes en una entidad bancaria.

## **Nuestro dataset, nuestro tesoro**

Una vez descargado y puesto en la carpeta adecuada (en nuestro caso, la carpeta “data” pero puede ser cualquiera a vuestra elección), procederemos a cargarlo y, como hacemos habitualmente en procesos de Machine Learning, dar un primer vistazo a los datos con los que trabajamos. Además, vamos a eliminar algunas variables que ya sabemos que tendremos y que no nos van a interesar por ser irrelevantes (como el apellido del cliente) o que no aportarán al modelo predictor de abandono por ser identificadores. Esto es un caso habitual en datasets de todo tipo, donde tenemos datos MUY detallados (teléfonos, IDs, etc.) que sabemos automáticamente que no van a aportar.

Tenemos varios atributos como la edad de los clientes, la antiguedad, el número de productos, etc. Todos potencialmente interesantes para analizar y predecir el abandono de los clientes. Además de una previsualización de los datos, también es importante saber la forma de los mismos: cuántas filas y columnas tenemos disponibles. Además vamos a crear unas cuantas variables que nos serán útiles a lo largo de la exploración.



No es un dataset muy masivo, 10.000 filas y 11 columnas. Habitualmente los datasets de sistemas reales pueden ser bastante más ricos en atributos, desde las decenas de atributos hasta los miles (especialmente si generamos atributos automáticamente). Aún siendo relativamente pocas filas y columnas es imposible analizar 10.000 filas manualmente.

## **Exploración visual del dato**

Para hacernos una idea breve de cómo se relacionan nuestros datos con nuestro label (la columna “Exited”, que marca si ese cliente se fue del banco o no) vamos a explorar visualmente los datos con algunos gráficos básicos. Así podremos empezar a tener una idea de qué atributos pueden ser importantes a la hora de predecir el abandono. Esta exploración de variables puede ser mucho más compleja, pero es importante hacerla de manera al menos de un modo básico porque nos puede dar pistas sobre varios puntos fundamentales de aquí en adelante:

* Qué atributos pueden ser importantes por sí mismos (si ayudan a diferenciar claramente los clientes que abandonan de los que no)
* Qué atributos pueden ser útiles para generar atributos compuestos para incrementar el rendimiento del algoritmo
* Potenciales problemas de calidad de datos
* Presencia de nulos
* Presencia de categorías muy dominantes en atributos de tipo texto
* Presencia de valores extremos y qué tipo de distribuciones tenemos: ¿son muy compactas? ¿muy dispersas?

Una de las columnas que puede ser importante en el análisis puede ser Edad. Vemos que al segmentarla por el label, diferencia los casos 0 (no abandona) y 1 (abandona) de manera significativa. Sin embargo, el hecho de que por sí misma diferencie el label no es una garantía de que al entrenar nuestro modelo de predicción de abandono tenga una gran importancia, ya que la mayoría de los modelos de Machine Learning actuales se aprovechan de las relaciones entre atributos además de su importancia individual.

Para las variables categóricas (aquellas no-numéricas que representan categorías, como “Género” o “País”) podemos crear histogramas en vez de boxplots para representar la distribución de los valores.