Telco Customer Churn

Los casos de deserción de clientes son comunes comercialmente, las empresas de telecomunicaciones son una de las más afectadas por este fenómeno. La deserción de clientes es un problema que afecta a las empresas de telecomunicaciones, ya que es más costoso mantener un cliente que adquirir uno nuevo. Por lo tanto, es importante para las empresas de telecomunicaciones predecir quién se irá para que puedan tomar medidas para retener a los clientes. En este proyecto, se utilizará la ciencia de datos para predecir la deserción de clientes y encontrar los clientes que tienen mayor probabilidad de deserción para que las empresas de telecomunicaciones puedan tomar medidas para retenerlos.

Elegí este dataset por ser bastante interesante y por la cantidad de información que contiene. Además, es un dataset que se puede utilizar para realizar un análisis exploratorio de datos, un análisis de datos descriptivos y un análisis predictivo.

Objetivos

- Predecir la deserción de clientes para que la empresa Telco pueda tomar medidas para retenerlos.
- Mostrar el aprendizaje de los conceptos de ciencia de datos y la aplicación de los mismos en un proyecto de ciencia de datos.
- Ilustrar el uso de las herramientas de ciencia de datos para resolver un problema de negocio.
- Generar un demo de un modelo de machine learning.
- Establecer las bases para usar MLOPS en un proyecto de ciencia de datos.

Dataset

El dataset elegido para este proyecto es el dataset de deserción de clientes de la empresa Telco. El dataset contiene información sobre 7043 clientes de la empresa Telco. Cada fila representa un cliente, cada columna contiene los atributos del cliente descritos a continuación.

Los datos fueron tomados de Kaggle Aquí

- **CustomerID**: la identificación del cliente
- **Gender**: Masculino Femenino
- **SeniorCitizen**: si el cliente es una persona mayor (0/1)
- **Partner**: si vive con pareja (sí/no)
- **Dependants**: si tienen dependientes (sí/no)
- **Tenure**: número de meses desde el inicio del contrato
- **PhoneService**: si tienen servicio telefónico (sí/no)
- **MultipleLines**: si tienen varias líneas telefónicas (sí/no/no servicio telefónico)
- **InternetService**: el tipo de servicio de internet (no/fibra/óptica)
- **OnlineSecurity**: si la seguridad en línea está habilitada (si/no/no internet)
- **OnlineBackup**: si el servicio de copia de seguridad en línea está habilitado (si/no/no internet)
- **DeviceProtection**: si el servicio de protección de dispositivos está habilitado (si/no/no internet)
- **TechSupport**: si el cliente tiene soporte técnico (si/no/no internet)

- **StreamingTV**: si el servicio de streaming de TV está habilitado (si/no/no internet)
- **StreamingMovies**: si el servicio de streaming de películas está habilitado (si/no/no internet)
- **Contract**: el tipo de contrato (mensual/anual/bianual)
- **PaperlessBilling**: si la facturación es sin papel (sí/no)
- **PaymentMethod**: método de pago (cheque electrónico, cheque enviado por correo, transferencia bancaria, tarjeta de crédito)
- **MonthlyCharges**: el monto cobrado mensualmente (numérico)
- **TotalCharges**: el monto total cobrado (numérico)
- **Churn**: si el cliente ha cancelado el contrato (sí/no)

Análisis exploratorio de datos

El análisis exploratorio de datos (EDA) es un proceso de análisis de datos que se utiliza para explorar, describir, resumir y visualizar los datos.

Para analizar los datos, se utilizaron las siguientes herramientas:

- **Pandas**: para la manipulación de datos.
- **Matplotlib**: para la visualización de datos.
- **Scikit-learn**: para la implementación de modelos de machine learning.
- **Explorator**: Clase personalizada para el análisis de tipos, frecuencias, variables categoricas y númericas.
- **Pandas Profiling**: para el análisis exploratorio de datos.

Análisis descriptivo de datos

El análisis descriptivo de datos es un proceso de resumen de los datos que se utiliza para describir los datos de una manera concisa.

Se utilizo Explorator para encontrar los detalles en los datos, asi como pandas profiling para revisar las relaciones entre las variables.

Se genero el indice de deserción de clientes, el cual es el porcentaje de clientes que se fueron de la empresa por variable.

Tambien se verifico la información mutua con respecto a la variable objetivo, para ver que variables son las que mas influyen en la deserción de clientes.

Proceso de limpieza de datos

El proceso de limpieza de datos es un proceso de preparación de datos que se utiliza para limpiar los datos para que puedan ser utilizados para el análisis.

Transformaciones realizadas:

- Se eliminaron las filas con valores nulos.
- Se convirtió la columna **Churn** a valores booleanos.
- Se convirtieron las columnas **gender**, **Partner**, **Dependents**, **PhoneService**, **MultipleLines**, **OnlineSecurity**, **OnlineBackup**, **DeviceProtection**, **TechSupport**, **StreamingTV**, **StreamingMovies**, **PaperlessBilling** a valores booleanos.
- Se realizo Hot Encoding a las columnas categoricas.
- Se normalizaron los nombres de las columnas y se convirtieron a minúsculas.
- Se normalizaron los valores de las columnas y se convirtieron a minúsculas.

Modelos de machine learning

Para predecir la deserción de clientes, se utilizaron los siguientes modelos de machine learning:

- **Regresión logística**: es un modelo de clasificación binaria que se utiliza para predecir la probabilidad de que un cliente se vaya.

También se genero una exploración de modelos de machine learning para encontrar el mejor modelo para este problema.

Evaluación de modelos

Para evaluar los modelos de machine learning, se utilizaron las siguientes métricas:

- **Accuracy**: es la proporción de predicciones correctas.

FastAPI

FastAPI es un framework de Python moderno y rápido para construir APIs web con Python 3.6+ basado en estándares abiertos para APIs, con documentación automática interactiva y edición de código en tiempo real.

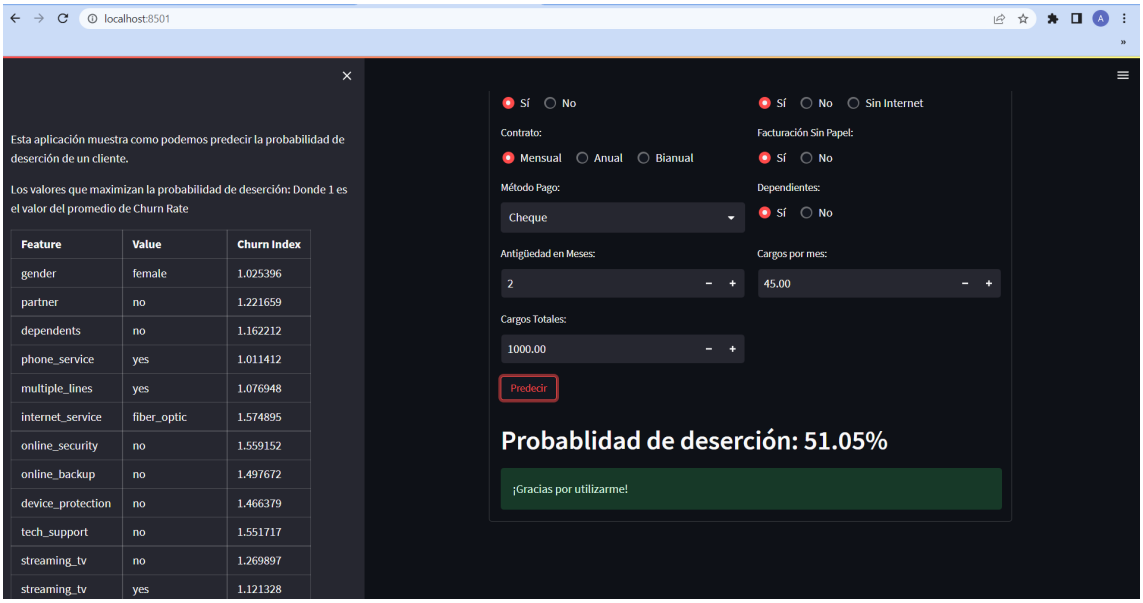
Utilice FAsAPI para crear una API web que reciba los datos de un cliente y prediga si el cliente se va o no.

Demo

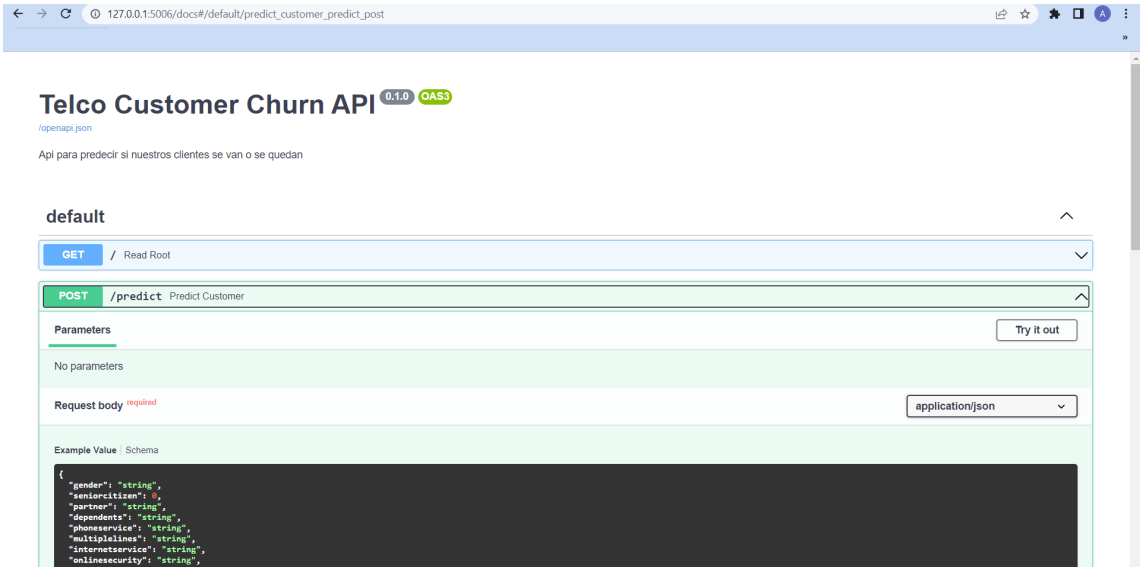
Utilice Streamlit para crear una aplicación web que reciba los datos de un cliente y prediga si el cliente se va o no. Es necesario tener instalado Python 3.10.9+ y las siguientes librerías:

- **Streamlit**: para crear la aplicación web.
- **Pandas**: para la manipulación de datos.
- **Scikit-learn**: para la implementación de modelos de machine learning.
- **FastAPI**: para crear la API web.
- **Uvicorn**: para ejecutar la API web.

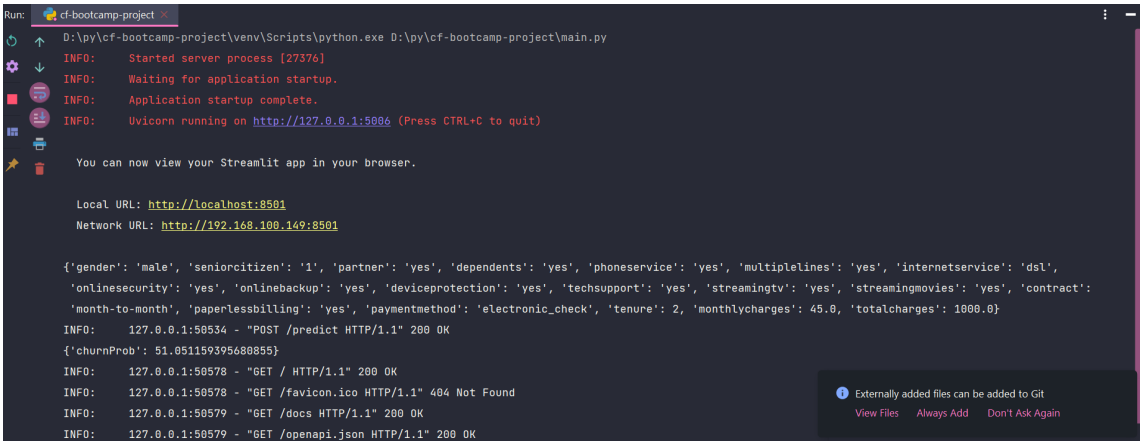
Aplicación Streamlit



Api FastApi



Resultado de Terminal



Instalación

Es necesario ejecutar el siguiente comando sobre un ambiente virtual de preferencia sobre la versión de Python 3.10.9+:

```
pip install -r requirements.txt
```

Considerar que si es la primera vez que ejecutas streamlit en tu equipo te pedirá un correo.

Aconsejo modificar la función main y ejecutar para llenar el email y permitir que streamlit se ejecute por primera vez.

```
def main():
    #api = multiprocessing.Process(target=server_api)
    #st_app = multiprocessing.Process(target=streamlit_app)
    #st_app.start()
    #api.start()
    streamlit_app() # DEJAR SOLO ESTA LINEA
```