

Proyecto: Análisis de Datos con Streamlit

Descripción General

En este proyecto, se desarrollará una aplicación web interactiva para análisis de datos utilizando **Streamlit** y **Plotly**. El objetivo es permitir a los usuarios cargar archivos CSV o TSV, explorar los datos y visualizarlos mediante gráficos interactivos. Los estudiantes deberán completar funciones en el archivo 'utils.py' para que el proyecto funcione correctamente.

Para ejecutar el archivo main se debe ejecutar el comando **streamlit run main.py**

Archivos del Proyecto

1. **Archivo main.py:** Este archivo contiene la estructura principal de la aplicación de Streamlit. Las funcionalidades de la aplicación incluyen:
 - **Carga de archivos:** Los usuarios pueden subir archivos en formato CSV o TSV, que serán procesados y cargados en un DataFrame de Pandas.
 - **Visualización de datos:** Se presenta una vista previa de los primeros datos del archivo cargado, así como un selector para elegir columnas para el análisis.
 - **Gráficos interactivos:** Dependiendo de las columnas seleccionadas, se muestran gráficos de dispersión o histogramas generados con Plotly.
 - **Estadísticas básicas:** Se muestran estadísticas descriptivas de la columna seleccionada.
 - **Detección de valores atípicos:** Utiliza una función para identificar películas con valores atípicos en el número de votos.
 - **Análisis de películas por género:** Un gráfico de barras muestra la cantidad de películas por cada género.
 - **Top 5 películas con peor rating:** Muestra un DataFrame con las cinco películas con calificación más baja.
 - **Filtro dinámico:** Los usuarios pueden filtrar los datos por un valor mínimo en una columna numérica.
2. **Archivo utils.py:** Este archivo contiene las funciones que se utilizan en main2.py para realizar operaciones específicas con los datos. El reto para los estudiantes es completar estas funciones para que la aplicación funcione de forma correcta.

Funciones Por Completar

Las siguientes funciones se encuentran en 'utils.py' y deben ser completadas:

1. `cargar_csv(filename):`
 - Debe leer un archivo CSV y retornar un DataFrame de Pandas.
2. `cargar_tsv(filename):`

- Similar a la función anterior, pero debe leer un archivo TSV (separado por tabulaciones) y retornar un DataFrame de Pandas.
- 3. `cargar_cabecera(df)`:
 - Debe recibir un DataFrame y devolver las primeras filas del mismo, proporcionando una vista previa de los datos.
- 4. `obtener_columnas(df)`:
 - Debe recibir un DataFrame y devolver una lista con los nombres de las columnas.
- 5. `obtener_estadistica_basica(df, columna)`:
 - Debe recibir un DataFrame y el nombre de una columna, y devolver las estadísticas descriptivas básicas (como media, mediana, desviación estándar, etc.) de esa columna.
- 6. `encontrar_peliculas_con_valores_atipicos()`:
 - Debe leer los archivos necesarios para identificar las películas con valores atípicos en la columna de número de votos y devolver un DataFrame con los nombres de estas películas.
- 7. `obtener_cantidad_peliculas_por_genero()`:
 - Debe generar un DataFrame que contenga la cantidad de películas por cada género. El DataFrame debe tener dos columnas: 'genero' (nombre del género) y 'cantidad' (número de películas en ese género).
- 8. `obtener_top_5_peliculas_con_mas_bajo_raiting()`:
 - Debe generar un DataFrame que contenga las 5 películas con las calificaciones más bajas, mostrando el nombre de la película y su calificación.

Retos para los Estudiantes

Los estudiantes deben completar las funciones en 'utils.py' para que la aplicación de Streamlit funcione correctamente. A continuación, se detallan los retos específicos:

1. **Implementar la carga de archivos:**
 - Completar las funciones `cargar_csv()` y `cargar_tsv()` para que la aplicación pueda leer los archivos cargados por el usuario y convertirlos en DataFrames.
2. **Visualización de datos:**
 - Completar `cargar_cabecera()` para mostrar las primeras filas del DataFrame en la aplicación, proporcionando así una vista previa a los usuarios.
3. **Exploración y análisis de datos:**

- Implementar `obtener_columnas()` para permitir la selección de columnas en los gráficos.
- Completar `obtener_estadistica_basica()` para que la aplicación muestre las estadísticas descriptivas de las columnas seleccionadas.

4. **Análisis avanzado:**

- Implementar las funciones `encontrar_peliculas_con_valores_atipicos()`, `obtener_cantidad_peliculas_por_genero()`, y `obtener_top_5_peliculas_con_mas_bajo_raiting()` para que la aplicación pueda analizar y visualizar información específica sobre las películas, como la cantidad de películas por género, las películas con calificaciones más bajas, y aquellas con valores atípicos en los votos.

Entregables

- Archivo `main.py`
- Archivo `utils.py` con las implementaciones de las funciones
- Archivo de `requirements.txt` con las librerías utilizadas