Informe PC-AGUACATE-202310

Apresa Rubens · Gómez Laura · Espinel Luis

Asignatura: Estructuras Discretas
Ingeniería de Sistemas
Universidad del Norte

25/04/2023

Implementación de librerías Pandas y NumPy

El ejercicio del Coleccionista requiere leer un archivo .csv con la información de jugadores de fútbol, para poder simular la compra de láminas para el álbum del mundial. Los objetivos planteados consisten en: a partir de un nuevo archivo .csv generado por el código, mostrar gráficas de frecuencia dada la compra de láminas y mostrar múltiples dataframes de una compra finita de 4000 hasta cumplir ciertas condiciones.

Respecto al código, está diseñado para que cualquier archivo .csv que cumpla con las columnas requeridas, pueda ser modificado para obtener la información requerida, esto se hace con un simple input que lee la dirección del archivo, el string es pasado al siguiente método de Pandas: `pd.read_csv(csv_path)`, que verifica él path y procesa los datos asignando el dataframe a una variable para que sea más sencillo de manipular.

Cuando se quiere completar un álbum, no todas las láminas tienen la misma probabilidad de salir, es por esto que es una de las columnas más importantes para la simulación; si el archivo ingresado por el usuario no registra en todas las filas este dato, se llamará a la función `df['Probability'].fillna(0, inplace=True)` para esta columna y cambiará **NaN** por 0. Siguiendo esta misma idea, es necesario normalizar las probabilidades para segurarnos que sumen 1, esto se hace mediante la operación:

`df['Probabilidad'] / df['Probabilidad'].sum()` $= \frac{\text{Probabilidad de jugador}}{\text{La probabilidad acumulada}}$.

Una vez obtenidas las nuevas probabilidades, y ya que estamos simulando la vida real, utilizamos la función `np.random.choice()`, de esta forma:
`np.random.choice(df['nombre'], p=probs_norm)`, donde el parámetro p permite realizar una selección ponderada, lo que significa que las láminas más probables tienen más posibilidades de ser seleccionadas que las menos probables.

Uno de los requerimientos de este trabajo es cambiar el nombre de las columnas, Pandas lo permite con un comando bastante explícito que solo requiere saber cuál es el datafrane, como se llama la columna originalmente y por cuál se quiere cambiar, por ejemplo: `df.rename(columns={'Player': 'nombre'})`.

Se puede presentar la ocasión de que el archivo subido por el usuario presente más de las columnas requeridas, es por esto que se implementa el método *loc*, ya que permite escoger solo las columnas con las que se requiere trabajar, por ejemplo:
`df.loc[:,['nombre', 'Probabilidad', 'Seleccion']]`.

Existen distintas formas de obtener la media, la moda y la mediana, en el código nos destacamos por la optimización de tareas, es por esto que se usa métodos de la librería NumPy para calcular esas medidas como: la media `np.mean()`, la moda `np.argmax()`, la mediana `np.median()`, también permite calcular la suma de los datos numéricos de la a la que se esté refiriendo con por `.sum`, por ejemplo: `df['Probabilidad'].sum()`

A modo de copia de seguridad, el nuevo dataframe generado se descargará y se guardará en la misma carpeta en el que el código se encuentre, por ejemplo en Google Colab es **/content**, esto es posible gracias al comando `to_csv()`; `df_new.to_csv('df_new.csv', index=False)`, que como se puede notar, necesita saber al dataframe al que se le está aplicando la función y dentro de los paréntesis se pone como primer parámetro el nuevo nombre del dataframe y de forma opcional si se desea con índice o no.

Los dataframes pueden ser extensos, es por esto que si se quiere verificar si funciona el nuevo df, para no mostrar todas las columnas, comando `head()` permite mostrar cierta cantidad de filas, por ejemplo `df_new.head(10)` aquí solo se muestran las primeras 10 filas.

Concluimos que tanto NumPy como Pandas son de vital importancia para el análisis de datos a partir de un archivo .csv, ya que permiten leer, corregir, cambiar, obtener las medidas de tendenia, la suma de datos numéricos de las columnas, escoger una fila al azar de acuerdo a un valor (probabilidad), entre otras cosas.

Si se desea comprender más a fondo como funciona el código, puede ingresar a este link para poder leerlo con más detenimiento, ya que se encuentra comentado Cuaderno de Google Colab