



17 de Octubre de 2024

Tarea 3

Tarea 3

Pandas y Excepciones

Fecha de Entrega: 22 de Octubre de 2024 a las 23:59 hrs.

Introducción

En la era de la información, el manejo adecuado y eficiente de grandes conjuntos de datos es esencial. Esto no sólo implica acumular información, sino también poder procesarla, limpiarla y obtener perspectivas valiosas de ella. En este contexto, la biblioteca de Python Pandas emerge como una herramienta poderosa y versátil para el análisis de datos.

Sin embargo, con grandes cantidades de datos, también vienen grandes responsabilidades. Los datos pueden estar incompletos, ser inconsistentes o, simplemente, contener errores. Por ello, es vital no sólo saber cómo trabajar con ellos, sino también cómo manejar posibles problemas que puedan surgir durante su procesamiento. El manejo de excepciones en Python nos proporciona herramientas para enfrentar estos desafíos de manera elegante y robusta.

Objetivos

El objetivo de esta tarea es evaluar el uso de Pandas y Excepciones mediante el manejo de un dataset y los posibles errores que pueden surgir.

Archivos entregados

Para este trabajo se hará entrega de un archivo:

- `developers_info.csv`: Este archivo contiene información de desarrolladores alrededor del mundo, tal como su nivel educacional, sus años de experiencia, su estatus laboral, los lenguajes en los han programado, los distintos tipos de tecnologías con las que han trabajado, etc. Este corresponde al dataset con el que deberás trabajar en la tarea para extraer la información solicitada.

Trabajo a realizar

Esta tarea se divide en dos partes: La primera parte consiste en completar funciones que trabajan con el dataset entregado mediante el uso de Pandas, y la segunda parte consiste en crear un menú a prueba de errores utilizando excepciones. A continuación se detalla en que consisten ambas secciones.

1. Pandas

En esta parte deberás extraer información del archivo `developers_info.csv` usando Pandas para completar las funciones solicitadas. A continuación se describen las funciones a implementar:

- `def cargar_dataset(ruta_dataset: str) -> df`: Esta función recibe como parámetro un `str` con la ruta del archivo csv que contiene el dataset. Deberás cargar este dataset usando pandas, mostrar las primeras 5 filas para entender su contenido y finalmente retornar el dataset cargado.
- `def limpiar_dataset(dataframe: df, cantidad_null: int) -> df`: En esta función deberás encargarte de limpiar el dataframe cargado anteriormente. En específico, debes eliminar aquellas columnas que posean un porcentaje mayor a `cantidad_null` % de celdas nulas y luego reemplazar las celdas nulas restantes con el `str` 'Sin información'. Finalmente, debes retornar el dataframe limpio.
- `def calcular_ingresos_por_pais(dataframe: df) -> df`: Esta función recibe como argumento el dataframe retornado por la función `limpiar_dataset`. Debes calcular el ingreso promedio de 5 países a tu elección y retornar este nuevo dataframe ordenado por ingreso promedio de mayor a menor. A continuación se presenta un ejemplo del retorno de esta función:

País	Ingreso promedio
USA	120.000
India	100.000
Chile	70.000
Brasil	45.000
Argentina	30.000

- `def calcular_ingresos_por_experiencia(dataframe: df) -> df`: Esta función recibe como argumento el dataframe retornado por la función `limpiar_dataset`. Debes calcular el rango de ingresos que poseen los desarrolladores que tienen un rango de años de experiencia. Para esto, el dataframe debe contener 3 columnas:
 - Años de experiencia: Esta columna debe indicar el rango de años de experiencia de los desarrolladores, los cuales deben ir en intervalos de 5 años, es decir, 0-5, 5-10, 10-15, etc. Para cada rango se debe considerar el límite inferior del intervalo, pero no el límite superior, es decir, el intervalo de 0-5 considera a los desarrolladores que tienen 0, 1, 2, 3 y 4 años de experiencia.
 - Ingreso menor: Corresponde al menor ingreso que poseen los desarrolladores que se encuentran en cada rango.
 - Ingreso mayor: Corresponde al mayor ingreso que poseen los desarrolladores que se encuentran en cada rango.

Finalmente, debes retornar el dataframe generado. A continuación se presenta un ejemplo del retorno de esta función:

Años de experiencia	Intervalo inferior (USD/Año)	Intervalo superior (USD/Año)
0-5	1	10
5-10	20	30
10-15	40	050

- `def calcular_empleabilidad(dataframe: df) -> df`: Esta función recibe como argumento el dataframe retornado por la función `limpiar_dataset`. Deberás encontrar el porcentaje de empleabilidad basado en la raza y el género de los desarrolladores. Para esto, debes considerar todas las combinaciones posibles de raza y género existentes en el dataset, es decir, todos los posibles pares (raza, genero) sin repeticiones. Debes asegurarte de manejar posibles errores o entradas inesperadas de manera adecuada. Para facilitar considere solamente las siguientes razas/generos,

```

1  [
2  "Black or of African descent",
3  "Hispanic or Latino/a/x",
4  "East Asian",
5  "I don't know",
6  "Indigenous",
7  "Middle Eastern",
8  "White or of European descent"
9  ]
10 [
11  'Man',
12  'Non-binary',
13  'genderqueer',
14  'gender non-conforming',
15  'Woman'
16 ]
17

```

A continuación se presenta un ejemplo del trabajo esperado:

Suponga que se tiene un dataset representado por la siguiente tabla:

Ethnicity	Gender	Employment
White	Female	Employed full-time
White	Male	Employed full-time
Black	Male	Not employed, but looking for work
White	Male	Employed full-time
White	Male	Not employed, but looking for work
White	Male	Employed part-time
Black	Female	0

En ese caso, el resultado esperado sería:

Ethnicity	Gender	Employability (%)
White	Female	100
White	Male	75
Black	Male	0
Black	Female	0

NOTA: Los valores en los ejemplos de las funciones presentadas anteriormente son sólo ilustrativos. No basen la correctitud de su algoritmo en estos.

- (Bonus) `def graficar_ingresos_paises(dataframe: df)`: En esta función deberás graficar (en formato a tu elección) el ingreso promedio calculado en la función `calcular_ingresos_por_pais` versus los 5 países elegidos. Esta función otorga 1 pto extra al puntaje total.

2. Excepciones

Para esa parte de la tarea deberás implementar una consola a prueba de errores utilizando los conocimientos de excepciones aprendidos en el curso. En esta parte se evaluará que el programa no se caiga independiente del input que se entregue y además el uso correcto de excepciones para cada tipo de error.

La consola se encargará de ejecutar las consultas definidas previamente de acuerdo a las instrucciones del usuario. Para esto, es necesario guardar las consultas en forma de un `dict`, donde cada llave corresponda al número de consulta y su valor corresponda a la función a ejecutar. A continuación se presenta un ejemplo de lo que podría mostrar la consola:

```
1      '¡Hola, bienvenido/a!'
2      'Para continuar ingrese el número equivalente a la consulta:'
3      '[1] Limpieza de datos'
4      '[2] Calcular ingresos por país'
5      '[3] Calcular ingresos por experiencia'
6      '[4] Calcular empleabilidad'
7      '[5] Graficar ingresos paises'
8      '[6] Salir'
9      # El número al lado izquierdo corresponde a la llave.
10     # Esta debe estar asociada a la consulta correspondiente.
11     # Por ejemplo {1: limpiar_dataset}
```

IMPORTANTE: El uso de `if/else statements` para el manejo de errores implica 0 puntos en esta sección. Además, no se considerará correcto el uso de `except` sin argumentos o de `except Exception`, sino que debes identificar los posibles tipos de errores y trabajar con estos.

Requisitos

- Para esta tarea sólo se puede usar la librería **Pandas**, cualquier librería externa debe ser consultada con el equipo docente mediante el foro designado para la tarea.
- En la sección de manejo de excepciones no será evaluado el manejo de errores con `if/else statements`, ni será evaluado el uso de estructuras de datos no vistas en el curso.
- Cabe recalcar que la tarea es de carácter individual, por lo que no está permitido compartir código con sus compañeros.
- Está estrictamente prohibido el uso de ChatGPT u otras herramientas generadoras de código para resolver esta tarea. Sí está permitido utilizar código extraído de internet, en cuyo caso debe ser referenciado en un comentario en el código.

Archivos a entregar

Deben entregar un archivo `.zip` (carpeta comprimida) con los archivos a trabajar: `consultas.py` para la primera parte y `consola.py` para la segunda parte.

Puntaje

- (4.0 pts) Parte 1: Pandas.
 - (0.5 pt) Función `cargar_dataset`.
 - (0.5 pt) Función `limpiar_dataset`.
 - (1.0 pt) Función `calcular_ingresos_por_pais`.
 - (1.0 pt) Función `calcular_ingresos_por_experiencia`.
 - (1.0 pt) Función `calcular_empleabilidad`.
- (2.0 pts) Parte 2: Excepciones.