# PEC4 - INFORME

Programación para la Ciencia de Datos

Jesús Sánchez Rodríguez [*]

14 de enero de 2024

Este informe corresponde a la PEC4 de la asignatura "Programación para la Ciencia de Datos", con la tutorización de Carlos Giner Baixauli[1]. Aquí se presentan las conclusiones derivadas de la implementación y ejecución de los ejercicios 1-4 de la prueba de evaluación y de los *tests* realizados.

[*]Estudiante de Grado en Ingeniería Informática (jessanrod3@uoc.edu)
[1]Profesor colaborador (cginerb@uoc.edu)

# Índice

# Ejercicio 1

## Descompresión de Ficheros

La función de descompresión se implementó y ejecutó con éxito, sin enfrentar problemas particulares.

## Integración de Datos con `Pandas` y `CSV`

Ambas funciones para integrar datos en un `DataFrame` de `pandas` y en un diccionario usando la librería `csv` se implementaron y ejecutaron con éxito, aunque el tiempo de procesamiento fue menor con `pandas`, reduciendo una media de 65 % el tiempo con respecto al módulo nativo.

## Diferencias en la Lectura

Si los archivos son muy grandes, el uso del módulo `csv` podría ser más eficiente que `Pandas`. Esto se debe a que `csv` trabaja con secuencias, permitiendo la lectura incremental sin cargar todo el archivo en memoria, a diferencia de `Pandas`, que carga los datos en su totalidad y, en caso de limitaciones de memoria, requiere de operaciones intermedias.

# Ejercicio 2

## Días en emisión

Se añadió la variable *air_days* al conjunto de datos, representando el número de días que una serie ha estado en emisión. Al analizar los datos, se observa que hay series que han estado en emisión durante períodos prolongados, lideradas por *CBS Evening News* con 30,043 días y seguida por eventos especiales como *Neujahrskonzert der Wiener Philharmoniker* y *Golden Globe Awards*. Vemos que se trata de emisiones históricas, que se transmiten desde hace muchos años.

### Diccionario de Posters

Se logró crear el diccionario ordenado con los posters de las series. Se encontraron casos donde las direcciones *web* o la ruta a la imagen no estaban disponibles, por lo que se manejaron etiquetándolos como `NOT AVAILABLE`. Esto asegura una representación uniforme y clara de los datos, incluso cuando no hay información disponible para esos campos, evitando la ausencia de valores.

## Ejercicio 3

### Filtrado por Idioma y Resumen

Se obtuvieron y mostraron las series cuyo idioma original es inglés y que contienen las palabras *mystery* o *crime* en el resumen. Los resultados fueron numerosos, entre los que se incluyen los ejemplos siguientes:

- *Stranger Things*
- *Breaking Bad*
- *The Umbrella Academy*
- *Wednesday*

### Lista de Series Canceladas en 2023

Se logró obtener la lista de series que comenzaron en 2023 y fueron canceladas, mostrando los primeros 20 elementos. Ejemplos de estas series son:

- *Lockwood & Co.*
- *The Idol*
- *Gotham Knights*
- *True Lies*

Este punto es muy interesante para analizar las características de las series canceladas desde un punto de vista estadístico, en primer lugar, para sacar posteriormente conclusiones.

### DataFrame con Series en Japonés

Se obtuvo exitosamente un DataFrame con información de las series en japonés, mostrando los primeros 20 registros. Ejemplos de estas series incluyen:

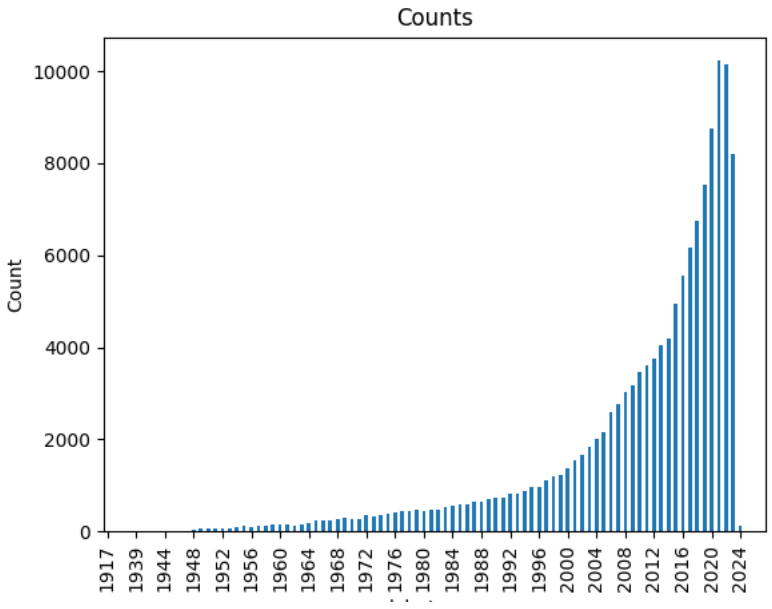
- *Naruto Shippūden*
- *Attack on Titan*
- *Naruto*
- *Dragon Ball Super*

Se consideraron tanto las que presentaban exclusivamente el idioma japonés como las que tenían otros idiomas disponibles junto a este.

## Ejercicio 4

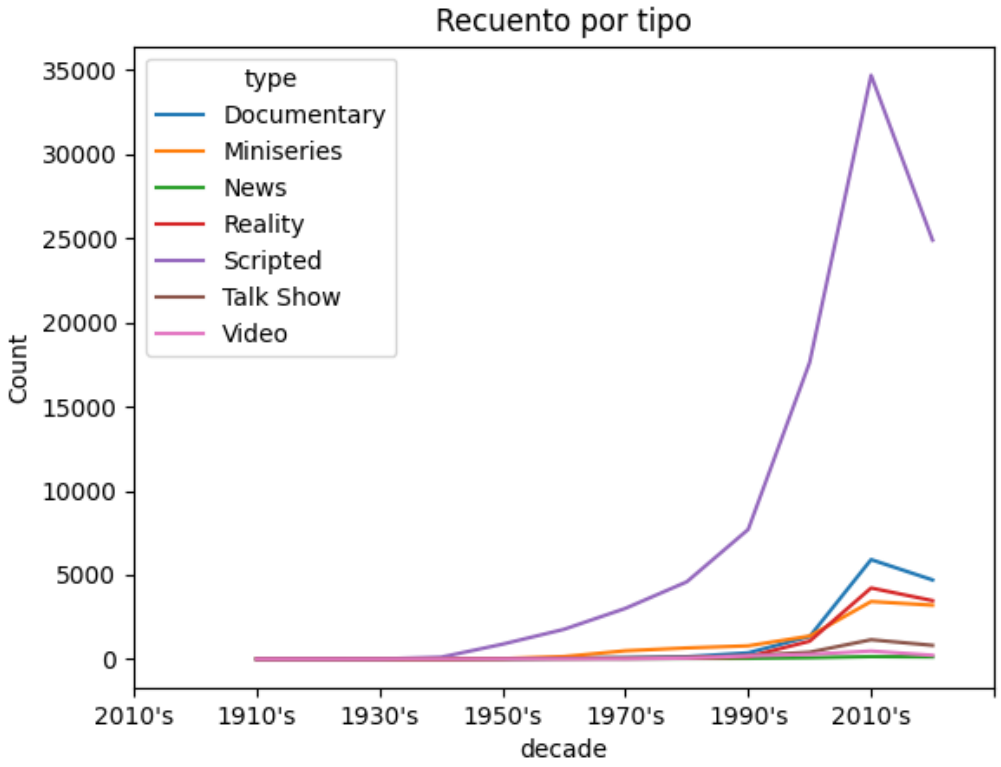
### Número de Series por Año de Inicio

Se mostró un gráfico de barras que ilustra el número de series por año de inicio. El gráfico revela un aumento más o menos lineal hasta los años 90, donde comienza un crecimiento exponencial hasta la actualidad. Este patrón puede indicar un incremento significativo en la producción de series en las últimas décadas.



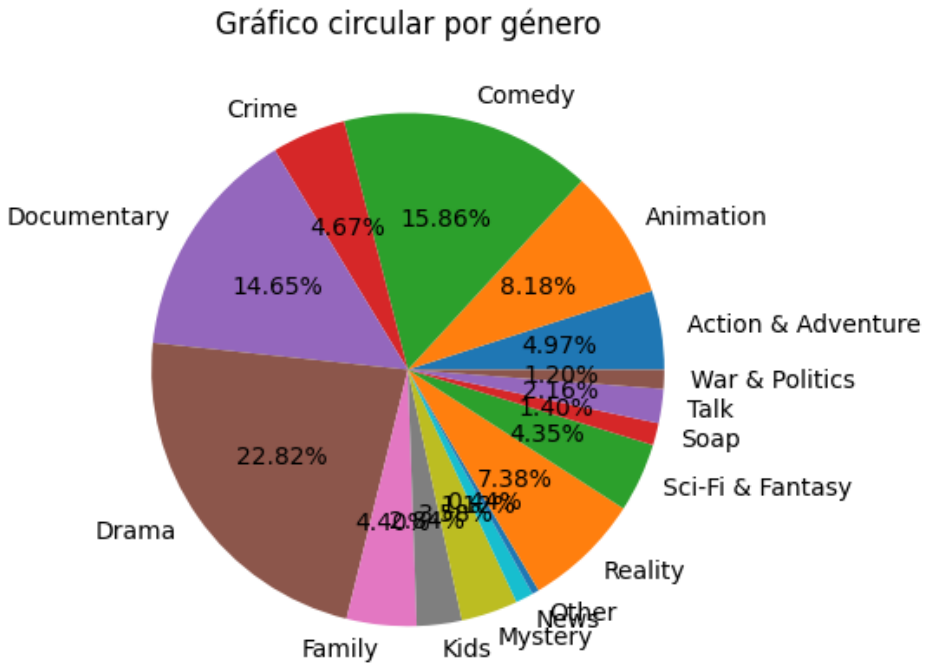
## Gráfico de Líneas por Categoría y Década

Se construyó un gráfico de líneas que muestra el número de series de cada categoría de la variable *type* por década. Se observa un incremento importante en los años 90, seguido por un crecimiento exponencial en la década de los 2010 en casi todos los tipos de serie. Sin embargo, destaca el tipo *scripted*, que experimenta un crecimiento exponencial ya en la década de los 50-60.



## Gráfico Circular por Género

Se obtuvo el número de series por género y se mostró el porcentaje respecto al total en un gráfico circular. Los géneros que representan menos del 1 % se etiquetaron como *Other*. Destaca la distribución de géneros como *Drama*, que ocupa el 22.82 % del total, *Documentary* el 14.65 %, *Comedy* el 15.86 %, *Animation* el 8.18 %, y *Reality* el 7.38 %.




En este análisis se consideraron los géneros únicos de cada serie. Esto implica que, en el caso de que una serie abarque varios géneros, se contabilizó cada uno de ellos de manera individual en el recuento.

# *Testing*

El análisis de cobertura revela la eficacia de los tests implementados en el desarrollo de la asignatura. La cobertura total alcanzada es del 84 %. A continuación se presentan los resultados:

**Coverage report: 84%**

*coverage.py v7.4.0, created at 2024-01-14 10:06 +0100*

| Module | statements | missing | excluded | coverage |
|---|---|---|---|---|
| exercises/base_exercise.py | 61 | 36 | 0 | 41% |
| exercises/ejercicio_1.py | 29 | 6 | 0 | 79% |
| exercises/ejercicio_2.py | 23 | 0 | 0 | 100% |
| exercises/ejercicio_3.py | 33 | 0 | 0 | 100% |
| exercises/ejercicio_4.py | 32 | 12 | 0 | 62% |
| tests/base_test.py | 18 | 12 | 0 | 33% |
| tests/test_1.py | 70 | 0 | 0 | 100% |
| tests/test_2.py | 30 | 0 | 0 | 100% |
| tests/test_3.py | 24 | 0 | 0 | 100% |
| tests/test_4.py | 14 | 0 | 0 | 100% |
| utils/utils_1.py | 44 | 0 | 0 | 100% |
| utils/utils_2.py | 22 | 0 | 0 | 100% |
| utils/utils_3.py | 11 | 0 | 0 | 100% |
| utils/utils_4.py | 18 | 1 | 0 | 94% |
| **Total** | **429** | **67** | **0** | **84%** |

*coverage.py v7.4.0, created at 2024-01-14 10:06 +0100*

Estos resultados ofrecen una visión inicial sobre la eficacia y fiabilidad del código en el escenario específico de la prueba de evaluación. Sin embargo, es importante señalar que cualquier cambio en los datos o los requisitos podría implicar ajustes en las funciones implementadas. Los *tests* están diseñados principalmente para los ejercicios y podrían no cubrir exhaustivamente los diversos casos que las funciones puedan enfrentar en otras situaciones.

# Referencias

[1] *Scaling to large datasets*. URL: https://pandas.pydata.org/docs/user_guide/scale.html. (accessed: 14.01.2024).

[2] *CSV File Reading and Writing*. URL: https://docs.python.org/3/library/csv.html. (accessed: 14.01.2024).

[3] *Python dictionaries*. URL: https://www.w3schools.com/python/python_dictionaries.asp.

[4] *Pandas documentation*. URL: https://pandas.pydata.org/docs/.

[5] *Using Matplotlib*. URL: https://matplotlib.org/stable/users/index.html.

[6] *Unittest - Unit testing framework*. URL: https://docs.python.org/3/library/unittest.html.

[7] *Coverage.py*. URL: https://coverage.readthedocs.io/en/7.4.0/.

[8] *How to run unittest tests using coverage API*. URL: https://stackoverflow.com/questions/71525147/how-to-run-unittest-tests-using-coverage-api.