Fundamentos de Programación

PEC 6 - Enunciado

En este Notebook encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PAC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- `NM` **Sólo materiales**: las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- `EG` **Consulta externa guiada**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- `EI` **Consulta externa independiente**: la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, puedéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

Ejercicios para la PEC

A continuación encontraréis los ejercicios que se deben completar en esta PEC y que forman parte de la evaluación de esta unidad.

Ejercicio 1

En esta PEC trabajaremos con el *dataset* `netflix_dataset`, que contiene información relacionada con las características de las películas y series que ofrece la plataforma Netflix. El conjunto original puede encontrarse en el repositorio de datos de Kaggle Netflix Movies and TV Shows, pero el conjunto de datos que utilizaremos tiene algunas modificaciones.

a) Importa el archivo `netflix_dataset.csv` de la carpeta de datos en un dataframe. Muestra por pantalla el número total de entradas, el número de variables, el nombre de las variables, y las 3 primeras entradas.

(0.5 puntos) `NM`

In [1]:

```
# Respuesta
```

b) ¿Hay entradas duplicadas? ¿Cuántas? Crea una lista con el número identificador (*show_id*) de todas las entradas duplicadas y elimina estas entradas del dataset. ¿Cuál es el tamaño del nuevo *dataset*?

(1 punto) NM

In [2]:

```
# Respuesta
```

c) Para cada columna o variable, muestra el número de datos que faltan y sustitúyelos por NaN usando el valor especial NaN de NumPy. **(1 punto)** EG

In [3]:

```
# Respuesta
```

Ejercicio 2

En el primer ejercicio hemos hecho una exploración preliminar del *dataset*, hemos identificado duplicados y entradas con datos incompletos. Sin embargo, todavía no hemos explorado los datos en profundidad.

a) Haz un listado de los tipos de variables que contiene el *dataset*. **(0.5 puntos)** NM

In [4]:

```
# Respuesta
```

b) Para la variable categórica *type* unifica el formato de los elementos. **(1 punto)** NM

In [5]:

```
# Respuesta
```

Ejercicio 3

¿Cuántos países aparecen representados en el catálogo de Netflix? Lístalos en orden alfabético. **(1 punto)** EI

In [6]:

```
# Respuesta
```

Ejercicio 4

Explora la variable numérica *release_year* y detecta los valores anómalos que puedas identificar.

(1 punto) NM

In [7]:

```
# Respuesta
```

Ejercicio 5

Comenta y explica la función del código siguiente.

(0.5 puntos) EI

In [8]:

```
from sklearn.preprocessing import FunctionTransformer

X = np.linspace(0, 1, num=10).reshape((5, 2))
```

```
X = np.linspace(0, 1, num=10).reshape((5, 2))

F = FunctionTransformer(np.around, kw_args=dict(decimals=3))

print(F.transform(X))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-8-73b3b1d5dffe> in <module>
      1 from sklearn.preprocessing import FunctionTransformer
      2
----> 3 X = np.linspace(0, 1, num=10).reshape((5, 2))
      4
      5 F = FunctionTransformer(np.around, kw_args=dict(decimals=3))

NameError: name 'np' is not defined
```

In [ ]:

```
# Respuesta
```

Ejercicio 6

Un grupo de antropólogos se pregunta si la producción de películas y series disminuye en épocas de crisis económica y queremos ayudarles a recoger evidencia.

Tendremos que representar la variable `release_year` como una variable discreta, en lugar de continua, ya que nos interesa separar esta información en 3 categorías: *BeforeCrisis*, *DuringCrisis*, *AfterCrisis*.

BeforeCrisis recogerá el conjunto de películas anteriores a 2008, *DuringCrisis* aquellas de 2008 a 2013 (ambos años incluidos), y *AfterCrisis* incluirá todas las posteriores a enero de 2014.

a) Crea un nuevo dataset derivado del anterior que incluya también la columna *time_frame* con este etiquetado. **(1 punto)** NM

In [ ]:

```
# Respuesta
```

b) Compara gráficamente el número de películas en cada una de las tres categorías. **(0.5 puntos)** NM

In [ ]:

```
# Respuesta
```

Ejercicio 7

El análisis de sentimientos es el proceso de detección de sentimientos positivos o negativos en el texto. Las empresas lo utilizan a menudo para detectar la opinión en los datos sociales, medir la reputación de la marca y comprender a los clientes. Para procesar texto Python ofrece un paquete integrado llamado re, que se puede usar para trabajar con expresiones regulares. Explora la documentación del paquete re.

a) En este ejercicio prepararemos los datos de la columna *description* del dataset para un posterior análisis de sentimientos. Para ello implementa una función que retorne, para cada película o serie, el texto que aparece en la variable *description*, excluyendo cualquier carácter o conjunto de caracteres no alfabéticos (ej. signos de puntuación, números...). Asigna el resultado a una nueva variable del dataset llamada *clean_description*. **(1 punto)** EG

In [ ]:

```
# Respuesta
```

b) Añade al *dataset* la variable booleana *gender_gap* y asígnale el valor True sólo en las películas y series cuya descripción sugiere que participan personajes femeninos. Aplica el método que consideres oportuno e indica el porcentaje de entradas resultantes que cumplen esta condición. **(1 punto)** EG

In [ ]:

```
# Respuesta
```

Ejercicio opcional

Comenta el siguiente código y explica para qué sirve.

EI

In [ ]:

```
import pandas as pd
from sklearn import preprocessing

aux = df.values
min_max_scaler = preprocessing.MinMaxScaler()
aux_scaled = min_max_scaler.fit_transform(aux)
d1 = pd.DataFrame(aux_scaled)
```

In [ ]:

```
# Respuesta
```

Respuesta