# prog_datasci_6_preproc_entrega_cast

May 2, 2021

# 1 Fundamentos de Programación

## 1.1 PEC 6 - Enunciado

En este Notebook se 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.

---

### 1.2 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.

## 2 Ejercicio 1

Una empresa especializada en Big Data y Data Science quiere contratar analistas de datos entre personas que superen con éxito algunos cursos que imparte. La empresa quiere saber cuáles de estos candidatos realmente quieren trabajar para la empresa después de formarse, ya que tener esta información les ayuda a reducir el coste y el tiempo, así como la calidad de la formación o la planificación de los cursos y la clasificación de los candidatos.

En esta PAC trabajaremos con el *dataset* `DS_HR.csv`, que contiene información relacionada con la demografía, la educación y la experiencia de los candidatos. El conjunto original puede encontrarse en el repositorio de datos de Kaggle HR Analytics: Job Change of Data Scientists, pero el conjunto de datos que utilizaremos tiene algunas modificaciones.

a) Importa el archivo `DS_HR.csv` de la carpeta de datos en un dataframe. Muestra por pantalla el número total de medidas, el nombre de las variables, y las 5 primeras entradas.

**(0.5 puntos)** NM

[1]:
```
# Respuesta
```

b) Hay duplicados en el *dataset*?

**(0.5 puntos)** NM

[2]:
```
# Respuesta
```

c) Qué variables son numéricas? Y categóricas?

**(0.5 puntos)** NM

[3]:
```
# Respuesta
```

d) Cuáles son los estadísticos (e.g. máximo, mínimo, desviación estándar, etc) de las variables numéricas? ¿Cuantos elementos hay en las variables categóricas (e.g. Gender tiene X Female e Y Male)?

**(0.5 puntos)** NM

[4]:
```
# Respuesta
```

### 2.1 Ejercicio 2

En el primer ejercicio hemos hecho una exploración del *dataset* y hemos podido identificar ciertos aspectos a corregir:

a) Observa las variables categóricas y unifica el formato de los elementos. Comprueba que los valores son correctos después de la modificación.

**(1 punto)** NM

[5]:
```
# Respuesta
```

b) Observa las variables numéricas y sustituye por NaNs los valores anómalos que puedas identificar. Razona por qué los consideras valores anómalos. Asegúrate de que las variables numéricas ya no contienen estos valores anómalos después de la modificación.

**(1 punto)** EI

[6]:
```
# Respuesta
```

c) Como se ha explicado en teoría, uno de los puntos clave del procesamiento de datos es el tratamiento de valores perdidos:

- Detecta si hay valores perdidos en el *dataset*. **(0.5 puntos)**
- Crea una **función** que, dado un *dataset*, muestre por pantalla el número de valores perdidos por cada columna y devuelva una lista con aquellas columnas que contienen valores perdidos. **(1 punto)**

**(1.5 puntos)** NM

[7]:
```
# Respuesta
```

d) Explica el código siguiente. Para qué se utiliza?

**(0.5 puntos)** EG

[ ]:
```
from sklearn.impute import SimpleImputer

imp = SimpleImputer(strategy='mean')

data_transf = data.copy()

data_transf["experience"] = imp.fit_transform(data_transf[["experience"]]).
 ↪ravel()
```

[8]:
```
# Respuesta
```

e) Como podemos tratar los valores perdidos en las variables categóricas? Da 2 posibles soluciones. **(1 punto)** EI

[9]:
```
# Respuesta
```

f) Comprueba que ya no hay valores perdidos en el *dataset*. **(0.5 puntos)** NM

[10]:
```
# Respuesta
```

### 2.2 Ejercicio 3

Queremos representar la variable `training_hours` como una variable discreta, en lugar de continua, ya que nos interesa separar esta información en 3 categorías: pocas horas, horas estándar y muchas horas.

Buscando por StackOverFlow hemos encontrado un artículo que explica 2 posibles maneras de hacerlo, con la función cut (), ya explicada en teoría, y con la función qcut (). Qué diferencia hay entre ellas? Aplica las dos funciones en el *dataset* y muestra los diferentes resultados.

**(1.5 puntos)** EG

[11]:
```
# Respuesta
```

### 2.3 Ejercicio 4

Cuando trabajamos con un dataset que posteriormente queremos introducir en algún modelo de machine learning es importante estandarizar las variables numéricas, ya que la mayoría de algoritmos no funcionan bien si las variables numéricas tienen diferentes escalas. Las dos formas más comunes de transformar los atributos a la misma escala son la *min-max scaling* y la *Standarization*.

Consulta la información de estos dos métodos en la página sklearn y, a continuación, aplica las transformaciones a las variables numéricas del *dataset*.

**(1 punto)** EG

[12]:
```
# Respuesta
```

### 2.4 Ejercicio opcional

Para poder trabajar con información en forma de texto, la mayoría de modelos de machine learning necesitan que estos valores sean transformados en atributos numéricos. EI

- Explica la diferencia entre la función `LabelEncoder` y`OneHotEncoder` de la librería de sklearn.
- Aplica la transformación que creas oportuna a las variables **Gender**, **major_discipline** y **education_level**. La variable major_discipline se simplificará como STEM y no STEM (todas las disciplinas que no son STEM englobarán en una única categoría no_STEM).

[13]:
```
# Respuesta
```