

# Logistic Regression

En este proyecto, trabajaremos con un conjunto de datos publicitarios falsos, que indica si un usuario de Internet en particular hizo clic en un anuncio en el sitio web de una empresa. Intentaremos crear un modelo que predecirá si harán clic o no en un anuncio en función de las características de ese usuario.

Este conjunto de datos contiene las siguientes características:

- 'Daily Time Spent on Site': tiempo del consumidor en el sitio en minutos
- 'Age': edad cliente en años
- 'Area Income': promedio renta del área geográfica del consumidor
- 'Daily Internet Usage': promedio minutos al día que el consumidor está en Internet
- 'Ad Topic Line': título del anuncio
- 'City': ciudad del consumidor
- 'Male': si el consumidor era hombre o no
- 'Country': país del consumidor
- 'Timestamp': hora a la que el consumidor hizo clic en el anuncio o en la ventana cerrada
- 'Clicked on Ad': 0 o 1 indicado si el cliente realiza clic en el anuncio

## Import Libraries

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

## Get the Data

```
In [ ]:
```

Check the head of ad\_data

```
In [ ]:
```

Usar info() y describe() para obtener un análisis general de la información

```
In [ ]:
```

```
In [ ]:
```

## Exploratory Data Analysis

**Histograma Edad** Crear un histograma sobre la columna Age

In [ ]:

**Crear un jointplot que muestre la relación de la edad vs los ingresos del área** Utilizar la librería seaborn. Interprete la gráfica.

In [ ]:

**Crear un jointplot que muestre la distribución k de la relación de la edad vs el tiempo que diario empleado**

A la función jointplot agregue los argumentos color='red' y kind='kde'

In [ ]:

**Crear un jointplot del Daily Time Spent on Site vs. Daily Internet Usage**

In [ ]:

**Finalmente, crear un pairplot que separe las poblaciones de 'Clicked on Ad'** Utilice el argumento hue y puede colocar un palette='bwr'

In [ ]:

## Logistic Regression

¡Ahora es el momento de hacer una prueba de train/test split y entrenar nuestro modelo!

Aquí tendrás la libertad de elegir las columnas en las que quieras entrenar.

In [7]:

```
from sklearn.model_selection import train_test_split
```

In [9]:

```
X = ad_data[['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', 'Male']]
y = ad_data['Clicked on Ad']
```

Usar `train_test_split` para generar `X_train`, `X_test`, `y_train` y `y_test` con un 33% en el split para split

In [ ]:

Entrene y ajuste un modelo de regresión logística en el conjunto de entrenamiento.

In [11]: `from sklearn.linear_model import LogisticRegression`

In [ ]:

## Predictions and Evaluations

Ahora predecir los valores para los datos de prueba.

In [13]:

Crear un reporte de clasificación (`classification_report`) para el mdoelo.

In [14]: `from sklearn.metrics import classification_report`

In [ ]:

Imprima el confussion matrix de la clasificación

In [ ]: `from sklearn.metrics import confusion_matrix`

In [ ]:

## Extra (Random Forest)

Aplique la clasificación utilizando Random Forest y compare los resultados

In [ ]: `from sklearn.ensemble import RandomForestClassifier`

In [ ]: