Práctica ‘3’ {

[Regresión Lineal univariable]

< Profesora: Consuelo Varinia García Mendoza >
< Alumna: Vianey Maravilla Pérez

}

Código Fuente

```python
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np
```

```python
def F(w, X, y):
    return sum((w * x - y)**2 for x, y in zip(X, y))/len(y)
```

```python
def dF(w, X, y):
    return sum(2*(w * x - y) * x for x, y in zip(X, y))/len(y)
```

```python
# Carga del dataset proporcionado

def load_dataset(path):

    data = pd.read_csv(path)

    x = data['size']
    y = data['price']
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1, shuffle = True, random_state=0)
    return [X_train, X_test, y_train, y_test]
```

Código Fuente

```
X_train, X_test, y_train, y_test = load_dataset('./dataset/regresion.csv')
```

```
def lineas(points, w, iteration, line_color = None, line_style = 'dotted'):
    list_x = []
    list_y = []
    for index, tuple in enumerate(points):
        x = tuple[0]
        y = x * w
        list_x.append(x)
        list_y.append(y)
    x1.text(x,y, iteration, horizontalalignment='right')
    x1.plot(list_x, list_y, color = line_color, linestyle= line_style)
```

Código Fuente

```
errors = []
_w = []
iterations = 15

fig = plt.figure(figsize=(15, 5))
x1 = fig.add_subplot(1, 2, 1)
x1.set_title("Regresión Lineal")
x1.set(xlabel="size", ylabel="price")
x2 = fig.add_subplot(1, 2, 2)
x2.set_title("Función de perdida ")
x2.set(xlabel="weight", ylabel="error")

x1.scatter(np.array(X_train), y_train)

w= 0
alpha = 0.00001
```

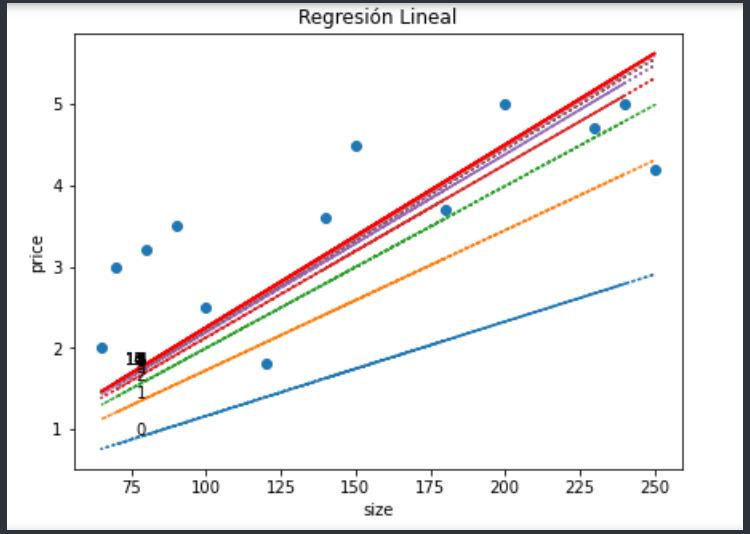
Código Fuente

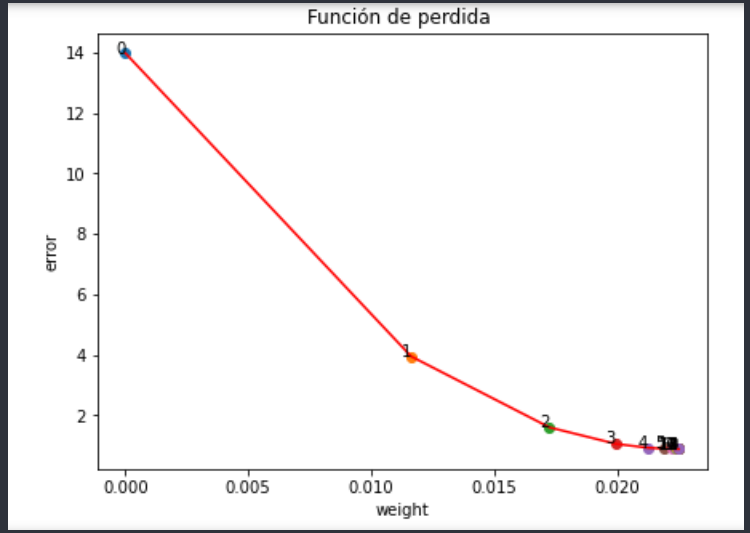
```
for t in range(iterations):
    error = F(w, np.array(X_train), y_train)
    gradient = dF(w, np.array(X_train), y_train)
    print ('gradient = {}'.format(gradient))
    x2.scatter(w, error)
    x2.text(w, error, t, horizontalalignment='right')
    _w.append(w)

    errors.append(error)
    w = w - alpha * gradient

    print ('iteration {}: w = {}, F(w) = {}'.format(t, w, error))
    lineas(zip(np.array(X_train), y_train), w, t)

lineas(zip(np.array(X_train), y_train), w, t, 'red', 'solid')
x2.plot(list_w, list_error, color = 'red', linestyle = 'solid')

plt.show()
```

Código Fuente

```
from tabulate import tabulate
resultado = []
pred = {'X_test': X_test,
        'X_test * w' : X_test * max(list_w),
        'Valor real' : y_test}
print(tabulate(pred,headers=['X_test', 'Valor predicho manualmente', 'Valor real de y_test'],tablefmt="grid", numalign="cent
```

Gráficas ‘Línea ajustada generada por el conjunto de entrenamiento’ y ‘Línea generada por el conjunto de prueba





M S E

X_test	Valor predicho manualmente	Valor real de y_test
150	3.37652	3.5
130	2.92632	2.8

Número de: ‘Iteraciones’ y ‘Alpha’

```
gradient = -1162.0
gradient = -562.1845384615385
gradient = -271.9892042041419
gradient = -131.5904692186115
gradient = -63.664481242342895
gradient = -30.801365751824296
gradient = -14.901937684315252
gradient = -7.2096720819615125
gradient = -3.4880948122688986
gradient = -1.6875671016742506
gradient = -0.8164579450754559
gradient = -0.3950086342731713
gradient = -0.19110821578935708
gradient = -0.09245962486214956
gradient = -0.04473267773613106
```

Número de: ‘Iteraciones’ y ‘Alpha’

```
gradient = -1162.0
iteration 0: w = 0.01162, F(w) = 13.97
gradient = -562.1845384615385
iteration 1: w = 0.017241845384615387, F(w) = 3.952487831538461
gradient = -271.9892042041419
iteration 2: w = 0.019961737426656807, F(w) = 1.6076899289522626
gradient = -131.5904692186115
iteration 3: w = 0.02127764211884292, F(w) = 1.0588433579161511
gradient = -63.664481242342895
iteration 4: w = 0.02191428693126635, F(w) = 0.9303749051740825
gradient = -30.801365751824296
iteration 5: w = 0.022222300588784594, F(w) = 0.9003043094540717
gradient = -14.901937684315252
iteration 6: w = 0.02237131996562775, F(w) = 0.893265688628056
gradient = -7.2096720819615125
iteration 7: w = 0.022443416686447365, F(w) = 0.8916181594738711
gradient = -3.4880948122688986
iteration 8: w = 0.022478297634570054, F(w) = 0.8912325225172876
gradient = -1.6875671016742506
```

Número de: ‘Iteraciones’ y ‘Alpha’

```
iteration 9: w = 0.02249517330558796, F(w) = 0.8911422565199272
gradient = -0.8164579450754559
iteration 10: w = 0.02250333788503752, F(w) = 0.8911211279684739
gradient = -0.3950086342731713
iteration 11: w = 0.02250728797138028 3, F(w) = 0.8911161824109044
gradient = -0.19110821578935708
iteration 12: w = 0.02250919905353817 4, F(w) = 0.8911150248048222
gradient = -0.09245962486214956
iteration 13: w = 0.02251012364978679 7, F(w) = 0.8911147538441019
gradient = -0.04473267773613106
iteration 14: w = 0.02251057097656416, F(w) = 0.8911146904203578
```

Contents Of ‘This Template’;

Here’s what you’ll find in this **Slidesgo** template:

- A slide structure based on a Workshop presentation, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.
- An assortment of graphic resources that are suitable for use in the presentation can be found in the **alternative resources slide**.
- A **thanks slide**, which you must keep so that proper credits for our design are given.
- A **resources slide**, where you’ll find links to all the elements used in the template.
- **Instructions for use**.
- Final slides with:
  - The **fonts and colors** used in the template.
  - A **selection of illustrations**. You can also customize and animate them as you wish with the online editor. Visit **Storyset** to find more.
  - More **infographic resources**, whose size and color can be edited.
  - Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you’re done editing the presentation.