# Ejercicio1_Clustering_012

**- Carla Mayela De la Garza Fernandez**

**- Alan Zamarron Medrano**

## Ejercicio Regresion Lineal

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
%matplotlib inline
from sklearn.linear_model import LinearRegression
```

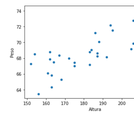
In [2]:
```
data = pd.read_csv("RegresionLineal.csv")
data
```

Out[2]:

| | Peso | Altura |
|---|---|---|
| 0 | 68.78 | 162 |
| 1 | 74.11 | 212 |
| 2 | 71.73 | 220 |
| 3 | 69.88 | 206 |
| 4 | 67.25 | 152 |
| 5 | 68.78 | 183 |
| 6 | 68.34 | 167 |
| 7 | 67.01 | 175 |
| 8 | 63.45 | 156 |
| 9 | 71.19 | 186 |
| 10 | 67.19 | 183 |
| 11 | 65.80 | 163 |
| 12 | 64.30 | 163 |
| 13 | 67.97 | 172 |
| 14 | 72.18 | 194 |
| 15 | 65.27 | 168 |
| 16 | 66.09 | 161 |
| 17 | 67.51 | 164 |
| 18 | 70.10 | 188 |
| 19 | 68.25 | 187 |
| 20 | 67.89 | 162 |
| 21 | 68.14 | 192 |
| 22 | 69.08 | 184 |
| 23 | 72.80 | 206 |
| 24 | 67.42 | 175 |
| 25 | 68.49 | 154 |
| 26 | 68.61 | 187 |
| 27 | 74.03 | 212 |
| 28 | 71.52 | 195 |
| 29 | 69.18 | 205 |

In [11]:
```
data.plot(kind="scatter", x="Altura", y="Peso")
```

Out[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x174b4cc2880>
```




In [4]:
```
lm = smf.ols(formula="Peso~Altura",data=data).fit()
lm.params
```

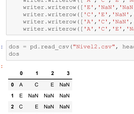
Out[4]:
```
Intercept    49.071634
Altura        0.108611
dtype: float64
```

In [9]:
```
altura_pred = lm.predict(pd.DataFrame(data["Altura"]))
altura_pred
```

Out[9]:
```
0     66.666580
1     72.097119
2     72.966006
3     71.445455
4     65.580473
5     68.947407
6     67.209634
7     68.078521
8     66.014916
9     69.273239
10    68.947407
11    66.775191
12    66.775191
13    67.752488
14    70.142125
15    67.318245
16    66.557970
17    66.883802
18    69.490461
19    69.381850
20    66.666580
21    69.924904
22    69.056018
23    71.445455
24    68.078521
25    65.797694
26    69.381850
27    72.097119
28    70.250736
29    71.336844
dtype: float64
```

In [10]:
```
data.plot(kind="scatter", x="Altura", y="Peso")
plt.plot(pd.DataFrame(data["Altura"]),altura_pred,c="green",linewidth=2)
```

Out[10]:
```
[<matplotlib.lines.Line2D at 0x174b4cac6d0>]
```




In [ ]:
```

```

## Ejercicio Algoritmo Apriori

### 1era parte: ejercicio resuelto sin libreria mlxtend

### 2da parte: ejercicio resuelto con libreria mlxtend

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:
```
data = pd.read_csv("Asociacion.csv", header=None)
data
```

Out[2]:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | A | B | C | E |
| 1 | B | E | NaN | NaN |
| 2 | C | D | E | NaN |
| 3 | A | C | D | NaN |
| 4 | A | C | E | NaN |

In [3]:
```
A=0
B=0
C=0
D=0
E=0
print("Nivel k=1")
for i in range(0, 5):
    for j in range(0, 4):
        if (data.values[i,j] == 'A'):
            A=A+1
        if (data.values[i,j] == 'B'):
            B=B+1
        if (data.values[i,j] == 'C'):
            C=C+1
        if (data.values[i,j] == 'D'):
            D=D+1
        if (data.values[i,j] == 'E'):
            E=E+1

SA=A/5
SB=B/5
SC=C/5
SD=D/5
SE=E/5
print("Soporte A=",str(SA))
print("Soporte B=",str(SB))
print("Soporte C=",str(SC))
print("Soporte D=",str(SD))
print("Soporte E=",str(SE))
print("Se eliminan B y D, debido a que tienen soporte menor del umbral de .5")
```

```
Nivel k=1
Soporte A=0.6
Soporte B=0.4
Soporte C=0.8
Soporte D=0.4
Soporte E=0.8
Se eliminan B y D, debido a que tienen soporte menor del umbral de .5
```

In [4]:
```
dns = ['A','C','E'],['B'],['C','E'],['A','C'],['A','C','E']
import csv
with open('Nivel2.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['A','C','E','NaN'])
    writer.writerow(['B','NaN','NaN','NaN'])
    writer.writerow(['C','E','NaN','NaN'])
    writer.writerow(['A','C','NaN','NaN'])
    writer.writerow(['A','C','E','NaN'])
```

In [5]:
```
dos = pd.read_csv("Nivel2.csv", header=None)
dos
```

Out[5]:

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | A | C | E | NaN |
| 1 | B | NaN | NaN | NaN |
| 2 | C | E | NaN | NaN |
| 3 | A | C | NaN | NaN |
| 4 | A | C | E | NaN |

In [6]:
```
print("Nivel k=2")
AC=0
AE=0
CE=0
a=0
y=0
for i in range(0, 5):
        y=1
        j=0
        for j in range(0, 4):
            if(dos.values[i,j]=='A' and dos.values[i,y]=='C'):
                AC=AC+1
                y=y+1
            if(dos.values[i,j]=='A' and dos.values[i,y]=='E'):
                y=y+1
                AE=AE+1
            if(dos.values[i,j]=='C' and dos.values[i,y]=='E'):
                CE=CE+1
                y=y+1
SAC=AC/5
SAE=AE/5
SCE=CE/5
print("SoporteAC= "+ str(SAC))
print("SoporteAE= "+ str(SAE))
print("SoporteCE= "+ str(SCE))
print("Se eliminan AE, debido a que tiene soporte menor del umbral de .5")

Nivel k=2
SoporteAC= 0.6
SoporteAE= 0.4
SoporteCE= 0.6
Se eliminan AE, debido a que tiene soporte menor del umbral de .5
```

In [ ]:
```

```

## Otra forma de solucionar el ejercicio utilizando mlxtend

In [11]:
```
pip install1 mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\carla\anaconda3\lib\site-packages (0.17.3)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (3.2.2)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (0.23.1)
Requirement already satisfied: setuptools in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (49.2.0.post20200714)
Requirement already satisfied: scipy>=1.2.1 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (1.5.0)
Requirement already satisfied: pandas>=0.24.2 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (1.0.5)
Requirement already satisfied: numpy>=1.16.2 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (1.18.5)
Requirement already satisfied: joblib>=0.13.2 in c:\users\carla\anaconda3\lib\site-packages (from mlxtend) (0.16.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\carla\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\carla\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\carla\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.7)
Requirement already satisfied: cycler>=0.10 in c:\users\carla\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\carla\anaconda3\lib\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\carla\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2020.1)
Requirement already satisfied: six>=1.5 in c:\users\carla\anaconda3\lib\site-packages (from python-dateutil>=2.1->matplotlib>=3.0.0->mlxtend) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [13]:
```
import pandas as pd
from mlxtend.frequent_patterns import apriori
file = pd.read_excel("ejercicio1_parte2.xlsx")
file.head()
```

Out[13]:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 | 1 |

In [18]:
```
frequent_itemsets = apriori(file, min_support=0.5, use_colnames=True)
frequent_itemsets
```

Out[18]:

| | support | itemsets |
|---|---|---|
| 0 | 0.6 | (A) |
| 1 | 0.8 | (C) |
| 2 | 0.8 | (E) |
| 3 | 0.6 | (A, C) |
| 4 | 0.6 | (C, E) |

In [ ]:
```

```