**Multi Linear Regression**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**comp\_data = pd.read\_csv("https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/Datasets/master/Regression/1000\_Companies.csv")**

**comp\_data.head()**

A screenshot of a graph

Description automatically generated

comp\_data.shape

(1000, 5)

comp\_data.info

<bound method DataFrame.info of R&D Spend Administration Marketing Spend State Profit

0 165349.20 136897.800 471784.1000 New York 192261.83000

1 162597.70 151377.590 443898.5300 California 191792.06000

2 153441.51 101145.550 407934.5400 Florida 191050.39000

3 144372.41 118671.850 383199.6200 New York 182901.99000

4 142107.34 91391.770 366168.4200 Florida 166187.94000

.. ... ... ... ... ...

995 54135.00 118451.999 173232.6695 California 95279.96251

996 134970.00 130390.080 329204.0228 California 164336.60550

997 100275.47 241926.310 227142.8200 California 413956.48000

998 128456.23 321652.140 281692.3200 California 333962.19000

999 161181.72 270939.860 295442.1700 New York 476485.43000

[1000 rows x 5 columns]>

# preprocess the data

comp\_data.isnull().sum()

R&D Spend 0

Administration 0

Marketing Spend 0

State 0

Profit 0

dtype: int64

import seaborn as sns

sns.pairplot(comp\_data)

<seaborn.axisgrid.PairGrid at 0x1e48eacf010>

A collage of graphs

Description automatically generated

import matplotlib.pyplot as plt

plt.scatter(comp\_data['Marketing Spend'],comp\_data['Profit'])

plt.show()

A blue dotted line graph

Description automatically generated

# seperate input and output data

x = comp\_data[['R&D Spend','Administration','Marketing Spend']]

x.head()

A screenshot of a graph

Description automatically generated

y = comp\_data['Profit']

y.head()

A screenshot of a computer

Description automatically generated

# Seperate Train and test data

from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.25,

random\_state=1)

from sklearn.linear\_model import LinearRegression

linear = LinearRegression()

# train the model fit

linear.fit(x\_train,y\_train)

LinearRegression

LinearRegression()

# test the model using predict

answers = linear.predict(x\_test)

linear.score(x\_train,y\_train)

**0.9593139672748138**

linear.intercept\_

-50443.433146575946

linear.coef\_

array([0.65583466, 0.87546548, 0.03849168])

linear.predict([[150000,40000,320000]])

array([95267.72112696])