HMR Institute of Technology & Management

(An ISO-2008 Certified Institute, Approved by AICTE & Affiliated to Guru Gobind Singh Indraprastha University)

Fostering Technical Excellence Through Education

Summer Training Project Report

On

MACHINE LEARNING

A PROJECT REPORT

Submitted in partial Fulfilment of the requirements
for
The award of the degree of

RITVIK SAPRA (00596502817)

In

Bachelor of Technology
ELECTRONICS & COMMUNICATION ENGINEERING

Guide By:
Mr. Chandan Kumar

DECLARATION

1. I/We , student(s) of B.Tech. (ECE 5th semester) hereby declare that summer training project entitled ***"50 STARTUPS"*** which is submitted to ***Department of ECE HMR Institute of Technology & Management***, Hamidpur, Delhi, affiliated to Guru Gobind Singh Indraprastha University, Dwarka(New Delhi) in partial fulfilment of requirement for the award of the degree of Bachelor of Technology in ECE has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

New Delhi	Trainer Name, Mr. CHANDAN Kr.
Date:	Sign.

Dr. A.K. SHARMA
(Head of Department)
(Professor of Electronics Department)

HMRITM Hamidpur, New Delhi-110036

CERTIFICATE

This is to certify that the project entitled
"_____" is

the bonafide work carried out by _____--student of B.Tech (HMR Institute of Technology and Management affiliated to Guru Gobind Singh Indraprastha University, Delhi) during the year_____, in

partial fulfillment of the requirements for the award of the Degree of Bachelor of Electronics and Communication Engineering and that the project has not formed the basis for the award previously of any degree, diploma, associateship, fellowship or any other similar title.

Signature of the Guide:

Place:

Date:

ACKNOWLEDGEMENTS

The Summer Training opportunity I had with Cetpa was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this training period.

I express my deepest thanks to Mr. Chandan Kumar, for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.He in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and allowing me to carry out my project at their esteemed organization and extending during the training.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives.

Sincerely,

Ritvik Sapra

Place:Delhi

Table of Contents

1. What is Data Analysis and Machine Learning?

Data analysis is a process of inspecting, cleansing, transforming and modelling data with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in decisions more scientific and helping businesses operate more effectively.

The data are necessary as inputs to the analysis, which is specified based upon the requirements of those directing the analysis or customers (who will use the finished product of the analysis). The general type of entity upon which the data will be collected is referred to as an experimental unit (e.g., a person or population of people). Specific variables regarding a population (e.g., age and income) may be specified and obtained. Data may be numerical or categorical (i.e., a text label for numbers).

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly

2. Project Overview

A. Abstract

In this project, we developed and evaluated the performance and the predictive power of a model trained and tested on data collected from 50 Start-ups .

- We used a dataset of 50 start-ups in New York, California and Florida. It gave us information about expenditure of start-ups on R&D, Administration and marketing and their respective profits

.

- We built a model to predict the Profit of a start-up based on the appropriate parameters.

- We followed the following steps:

  -first sorted the data and tried to visualise relationships between the data.

  -then built an appropriate ML model to predict the Profit.

  -In the end, the accuracy of the model was found out.

B. Requirements

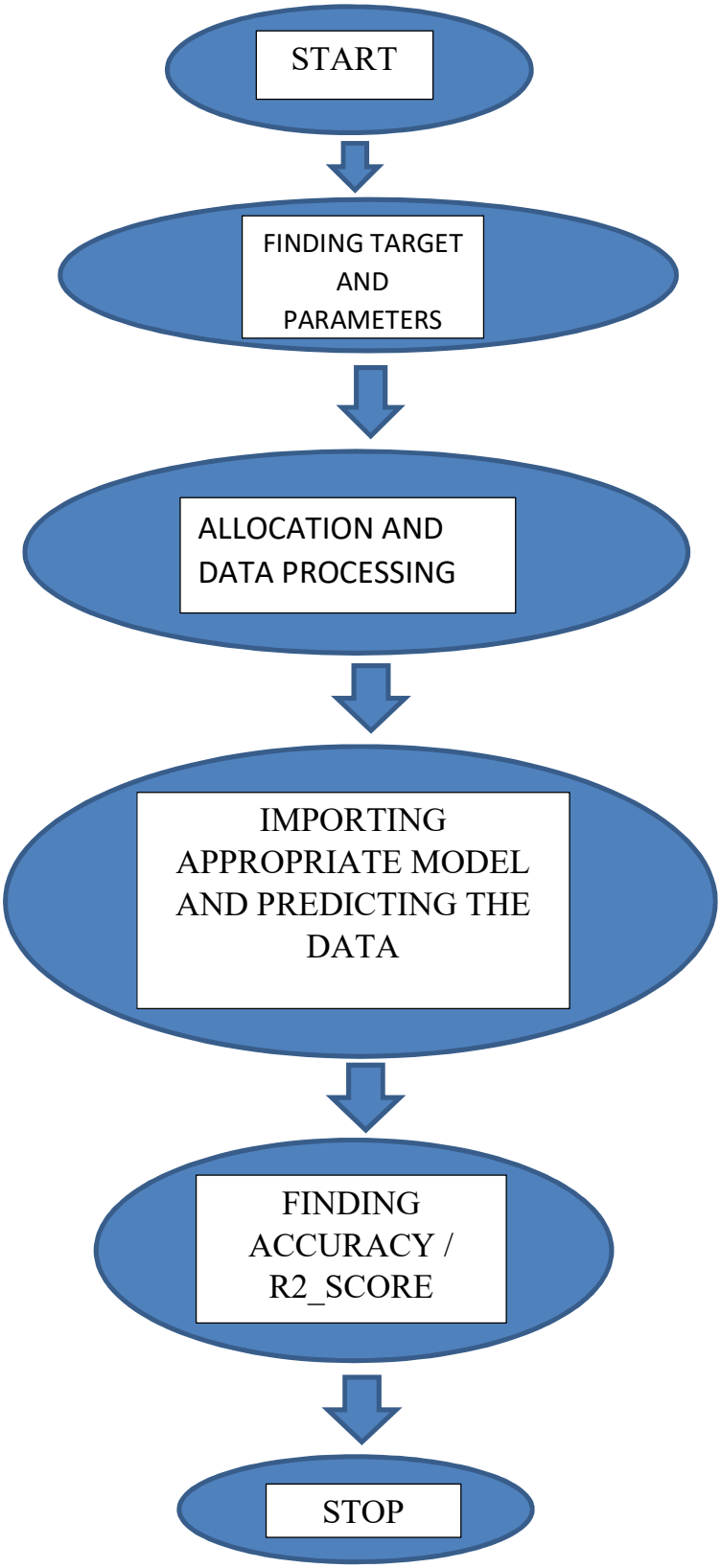
- Past Data of the start-ups of their expenditures and their Profit.
- Identifying the objectives of Start-ups.
- Appropriate Software.
- Knowledge of Python and Machine Learning.

	A	B	C	D	E
1	R&DSpend	Administra	Marketing$	State	Profit
2	165349.2	136897.8	471784.1	NewYork	192261.8
3	162597.7	151377.6	443898.5	California	191792.1
4	153441.5	101145.6	407934.5	Florida	191050.4
5	144372.4	118671.9	383199.6	NewYork	182902
6	142107.3	91391.77	366168.4	Florida	166187.9
7	131876.9	99814.71	362861.4	NewYork	156991.1
8	134615.5	147198.9	127716.8	California	156122.5
9	130298.1	145530.1	323876.7	Florida	155752.6
10	120542.5	148719	311613.3	NewYork	152211.8
11	123334.9	108679.2	304981.6	California	149760
12	101913.1	110594.1	229161	Florida	146122
13	100672	91790.61	249744.6	California	144259.4
14	93863.75	127320.4	249839.4	Florida	141585.5
15	91992.39	135495.1	252664.9	California	134307.4
16	119943.2	156547.4	256512.9	Florida	132602.7
17	114523.6	122616.8	261776.2	NewYork	129917
18	78013.11	121597.6	264346.1	California	126992.9
19	94657.16	145077.6	282574.3	NewYork	125370.4
20	91749.16	114175.8	294919.6	Florida	124266.9
21	86419.7	153514.1	0	NewYork	122776.9
22	76253.86	113867.3	298664.5	California	118474
23	78389.47	153773.4	299737.3	NewYork	111313
24	73994.56	122782.8	303319.3	Florida	110352.3
25	67532.53	105751	304768.7	Florida	108734
26	77044.01	99281.34	140574.8	NewYork	108552
27	64664.71	139553.2	137962.6	California	107404.3
28	75328.87	144136	134050.1	Florida	105733.5
29	72107.6	127864.6	353183.8	NewYork	105008.3
30	66051.52	182645.6	118148.2	Florida	103282.4
31	65605.48	153032.1	107138.4	NewYork	101004.6
32	61994.48	115641.3	91131.24	Florida	99937.59
33	61136.38	152701.9	88218.23	NewYork	97483.56
34	63408.86	129219.6	46085.25	California	97427.84
35	55493.95	103057.5	214634.8	Florida	96778.92

C. Analysis

- From the above data, we get the information about the Start-ups' expenditure.
- Since, Funds are a very important of a Start-up, it is very useful and wise to plan their expenditures and maximize their profits.
- Therefore, we build a model on Machine Learning to plan out their finances.
- Through this report, we lay out the steps to achieve the required target.
- We first apply Data Analysis and Visualisation techniques followed by an appropriate algorithm to build our model.
- We have used here different models and chosen the most accurate one .

D. Control Flow Diagram



3. LIBRARIES USED:

❖ **NumPy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

❖ **Pandas**

Loading and Saving Data with Pandas.When you want to use Pandas for data analysis, you'll usually use it in one of three different ways:

- Convert a Python's list, dictionary or NumPy array to a Pandas data frame.
- Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc.
- Open a remote file or database like a CSV or a JSONon a website through a URL or read from a SQL table/databaseThere are different commands to each of these options, but when you open a file

❖ **Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of matplotlib.

❖ Selection Data

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (df[col]) and return column with label col as Series or a few columns (df[[col1, col2]]) and returns columns as a new DataFrame. You can select by position (s.iloc[0]), or by index (s.loc['index_one']) . In order to select the first row you can use df.iloc[0,:] and in order to select the first element of the first column you would run df.iloc[0,0] . These can also be used in different combinations, so I hope it gives you an idea of the different selection and indexing you can perform in Pandas.

❖ Data Cleaning

Data cleaning is a very important step in data analysis. For example, we always check for missing values in the data by running pd.isnull() which checks for null Values, and returns a boolean array (an array of true for missing values and false for non-missing values).

❖ Seaborn

Seaborn is a library for making statistical graphics in Python. It is built on top of matplotlib and closely integrated with pandas data structures. ... Options for visualizing univariate or bivariate distributions and for comparing them between subsets of data.

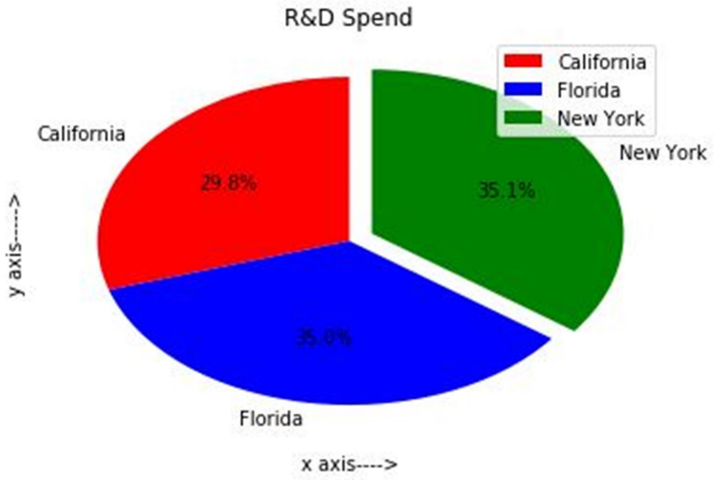
❖ Sklearn

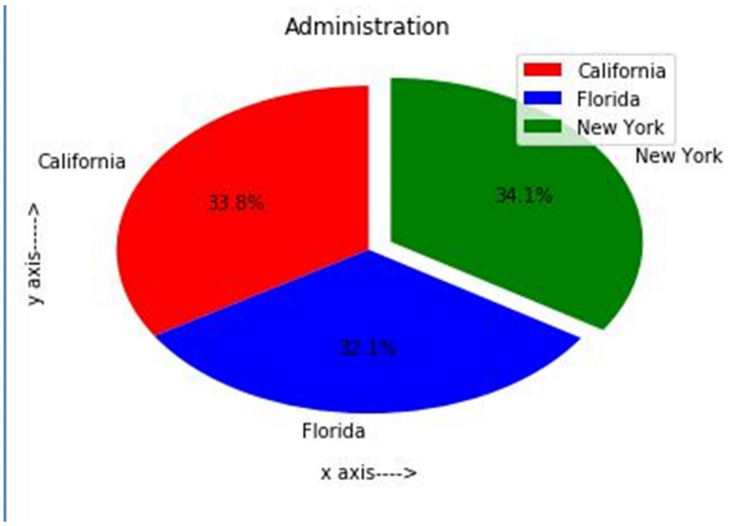
Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

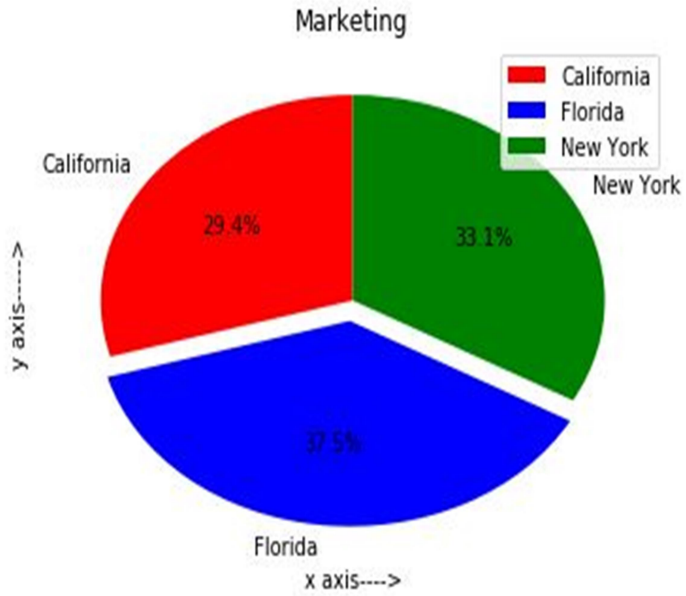
4. VISUALISATION

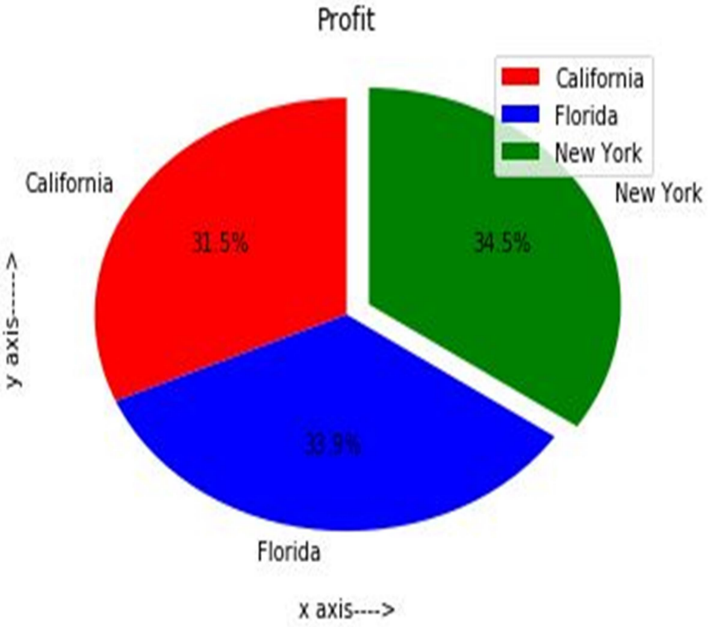
(Data Analysis)

Categorising Data:



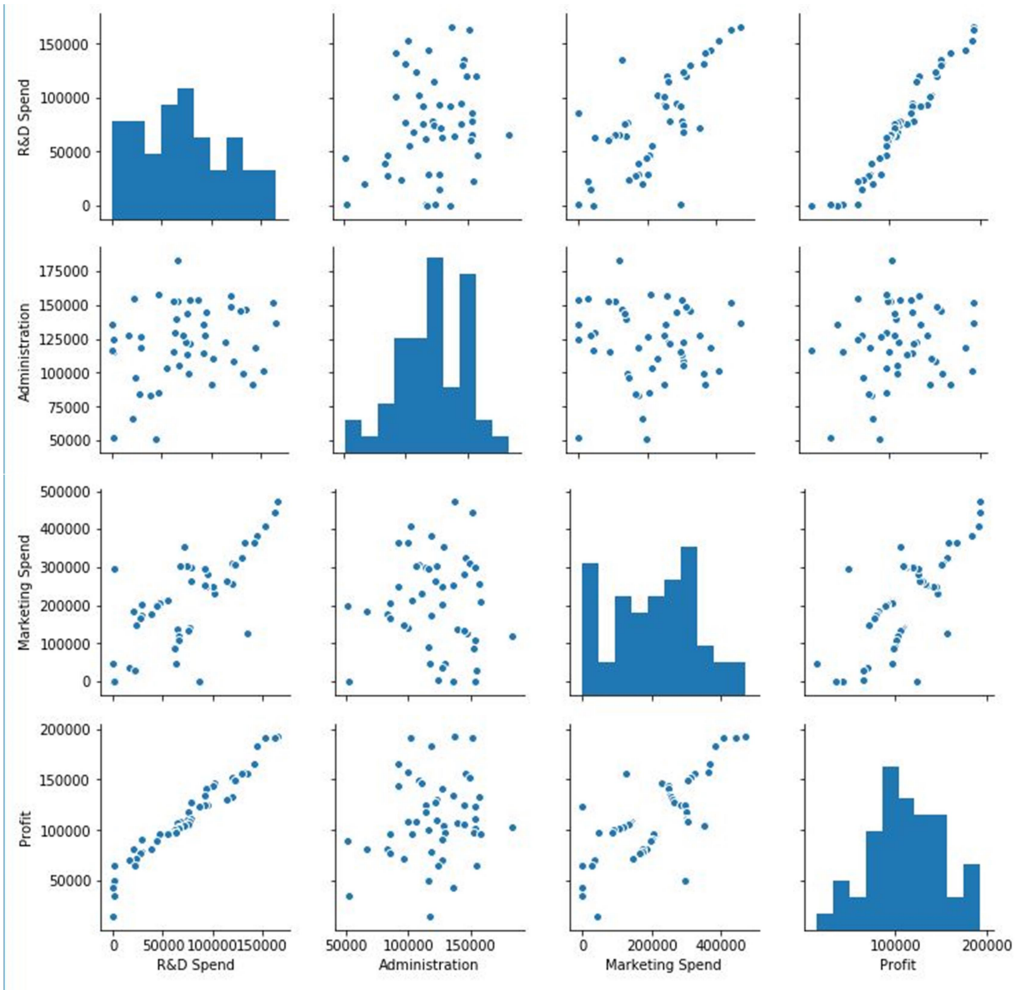






- Here we have categorised data according to the States.
- The pie charts show the individual expenditure and Profit percentage as per each State.

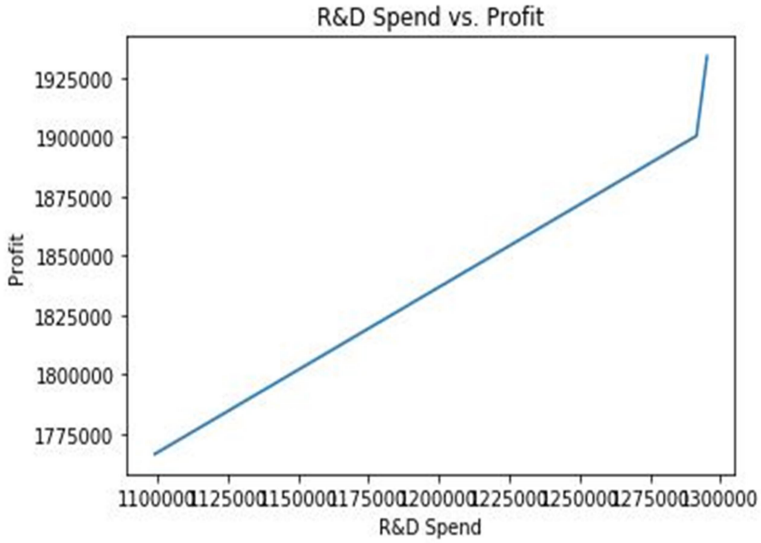
Analysing relations between all the columns:

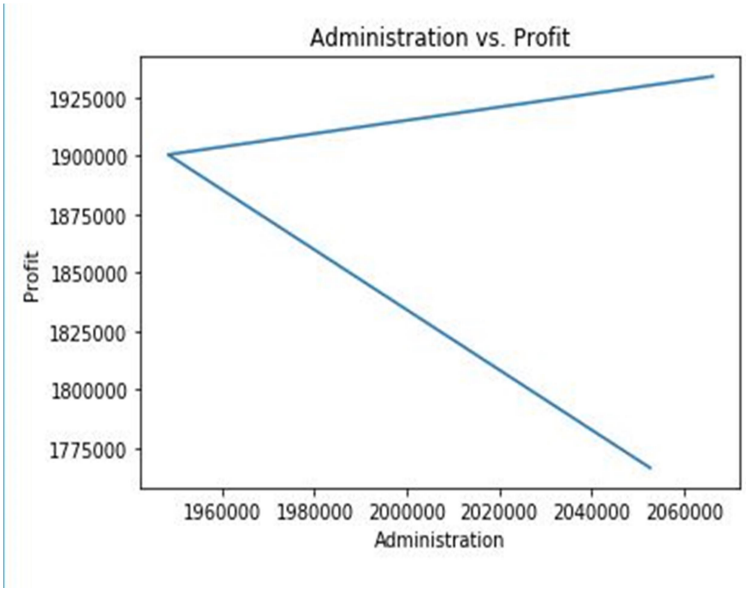


Analysis:

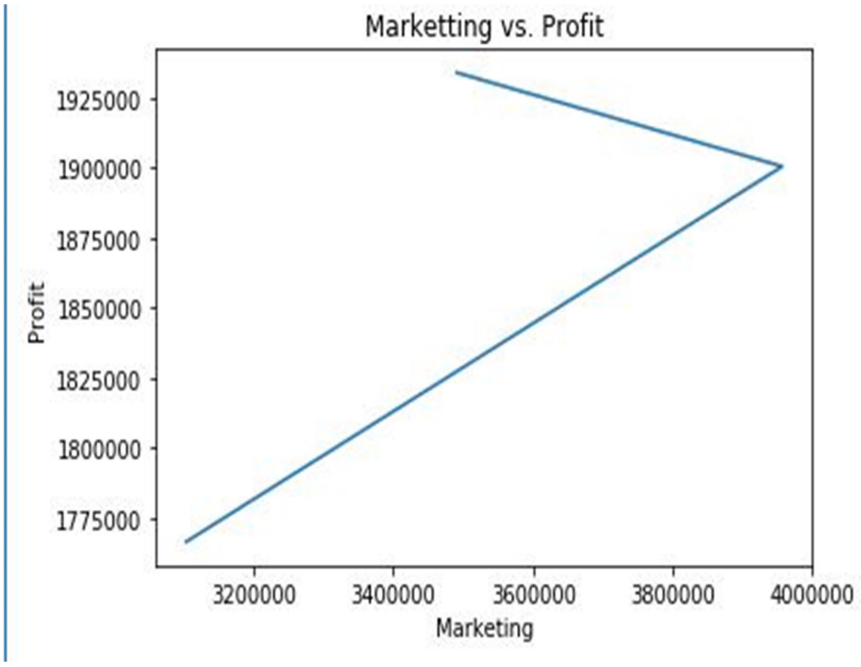
The above piecharts depict that startups in each state have spent and gained profit almost equally. Therefore, State is not related to the Profit of Startups.

Now, we have to select two parameters among the three which are best correlated with Profit. Hence, we plot each parameter with respect to Profit.





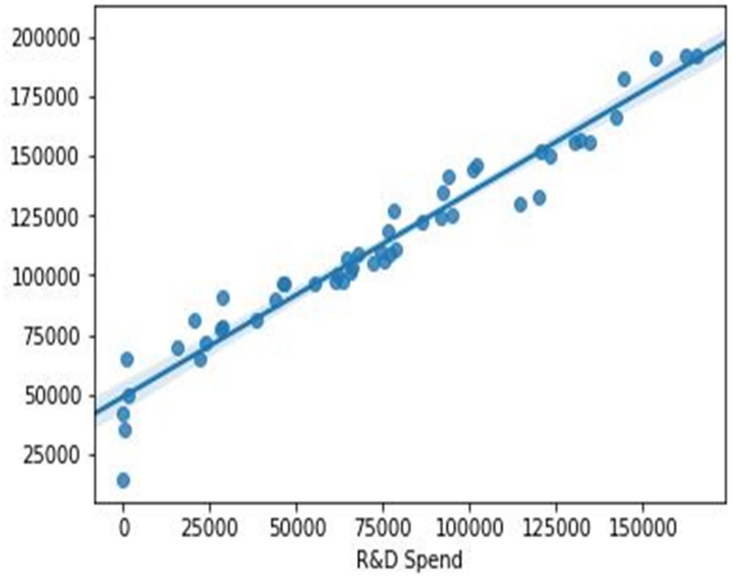
……

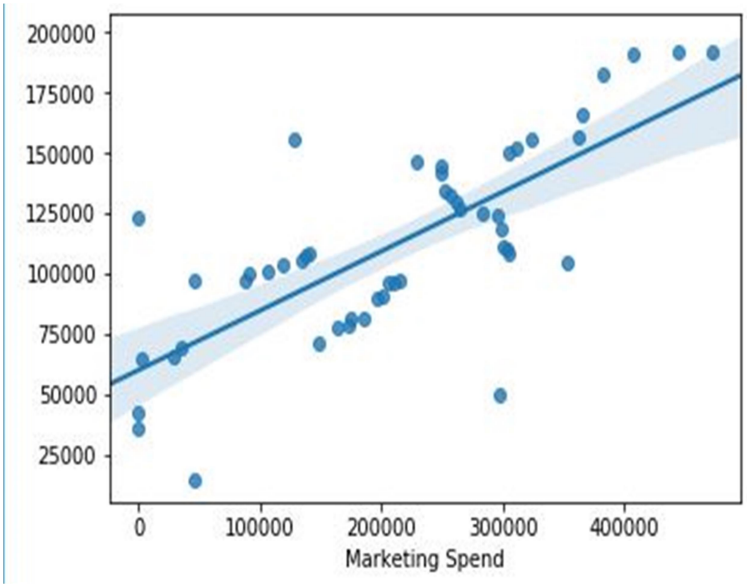


Conclusion:

- From the above plots, we conclude that R&D and Marketing Spend are the best linearlly related with Profit.
- Hence we choose R&D and Marketing Spend as our two parameters and Profit as our target dataset.

❖ To better understand the realtions between target and parameter datasets, we plot the Regplot().





5. Machine Learning Model

a. Model used:

i. Linear Regression:

Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

- If the goal is prediction, or forecasting, or error reduction,[*clarification needed*] linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.
-
- If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

ii. SVM:

In machine learning, **support-vector machines** (**SVMs**, also **support-vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces,

iii. Decision Tree Learning:

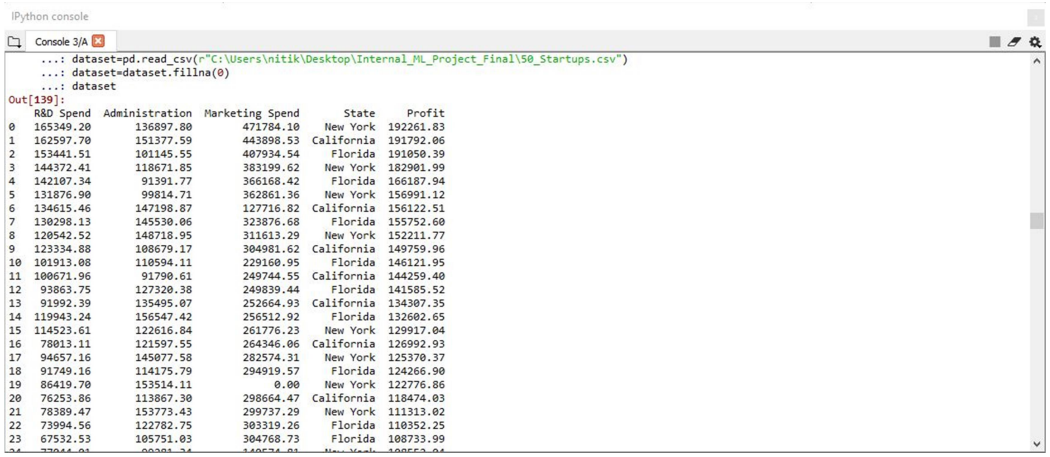
Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). This page deals with decision trees in data mining.

B. Model Implementation :

i. Developing the Model :

1. We load the dataset



```
IPython console
Console 3/A
    ...: dataset=pd.read_csv(r"C:\Users\nitik\Desktop\Internal_ML_Project_Final\50_Startups.csv")
    ...: dataset=dataset.fillna(0)
    ...: dataset
Out[139]:
    R&D Spend  Administration  Marketing Spend       State     Profit
0   165349.20       136897.80        471784.10    New York  192261.83
1   162597.70       151377.59        443898.53  California  191792.06
2   153441.51       101145.55        407934.54     Florida  191050.39
3   144372.41       118671.85        383199.62    New York  182901.99
4   142107.34        91391.77        366168.42     Florida  166187.94
5   131876.90        99814.71        362861.36    New York  156991.12
6   134615.46       147198.87        127716.82  California  156122.51
7   130298.13       145530.06        323876.68     Florida  155752.60
8   120542.52       148718.95        311613.29    New York  152211.77
9   123334.88       108679.17        304981.62  California  149759.96
10  101913.08       110594.11        229160.95     Florida  146121.95
11  100671.96        91790.61        249744.55  California  144259.40
12   93863.75       127320.38        249839.44     Florida  141585.52
13   91992.39       135495.07        252664.93  California  134307.35
14  119943.24       156547.42        256512.92     Florida  132602.65
15  114523.61       122616.84        261776.23    New York  129917.04
16   78013.11       121597.55        264346.06  California  126992.93
17   94657.16       145077.58        282574.31    New York  125370.37
18   91749.16       114175.79        294919.57     Florida  124266.90
19   86419.70       153514.11             0.00    New York  122776.86
20   76253.86       113867.30        298664.47  California  118474.03
21   78389.47       153773.43        299737.29    New York  111313.02
22   73994.56       122782.75        303319.26     Florida  110352.25
23   67532.53       105751.03        304768.73     Florida  108733.99
```

2. We then print the correlation table.

	R&D Spend	Administration	Marketing Spend	Profit
R&D Spend	1.000000	0.241955	0.724248	0.972900
Administration	0.241955	1.000000	-0.032154	0.200717
Marketing Spend	0.724248	-0.032154	1.000000	0.747766
Profit	0.972900	0.200717	0.747766	1.000000

3. Now we make R&D spend and Marketing Spend as our parameter dataset(x) and Profit as our target dataset(y).

ii. Choosing model:

We experimented with 3 ML Models.

a) Linear Regression
b) SVM
c) Decision

Firstly we try to implement all the models.

A) LINEAR REGRESSION:

Code:

```
#%%
''' We have properly visualised the data and found out the relation. Now we implement
the ML model, i.e., Linear Regresion (we found it most effective amongst other models) '''

#We train, test and split the 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=21)
x_train
#%%
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
pred1=lr.predict(x_test)
pred1
#%%
from sklearn.metrics import r2_score
r2=r2_score(y_test,pred1)
r2

```

Output:

```
In [82]: from sklearn.linear_model import LinearRegression
    ...: lr=LinearRegression()
    ...: lr.fit(x_train,y_train)
    ...: pred1=lr.predict(x_test)
    ...: pred1
Out[82]:
array([161169.90107216,  64484.27048906,  59488.56524987, 102229.4864024 ,
       150419.64541063, 182659.24206003, 110860.17486122,  97544.10378233,
       131439.85666316,  45884.10928803,  98333.10478669, 119244.4221274 ,
       114149.81329035])

In [83]: from sklearn.metrics import r2_score
    ...: r2=r2_score(y_test,pred1)
    ...: r2
Out[83]: 0.9587963239667265
```

Accuracy : 95.87% ~ 96%

B) SVM:

Code:

```
Created on Wed Sep  4 13:43:05 2019

@author: nitik
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
dataset=pd.read_csv(r"C:\Users\nitik\Desktop\Internal_ML_Project_Final\50_Startups.csv")
dataset=dataset.fillna(0)
dataset
#%%
x=dataset.iloc[:,[0,2]].values
y=dataset.iloc[:,-1].values
print(x)
print(y)
#%%
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5,random_state=21)
x_train
#%%
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x = sc.fit_transform(x_train)
y=sc.fit_transform(y_train.reshape(-1,1))
#%%
from sklearn.svm import SVR
regressor= SVR(kernel='rbf')
regressor.fit(x,y)

#%%
pred3=regressor.predict(x_test)
pred3
#%%
from sklearn.metrics import r2_score
r1=r2_score(y_test,pred3)
r1
```

Output:

```
       [ 0.95934858,  0.80679693]])

In [56]: from sklearn.preprocessing import StandardScaler
    ...: sc = StandardScaler()
    ...: x = sc.fit_transform(x_train)
    ...: y=sc.fit_transform(y_train.reshape(-1,1))

In [57]: from sklearn.svm import SVR
    ...: regressor= SVR(kernel='rbf')
    ...: regressor.fit(x,y)
C:\Users\nitik\Anaconda3\lib\site-packages\sklearn\utils\validation.py:578: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,
), for example using ravel().
  y = column_or_1d(y, warn=True)
Out[57]:
SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto',
  kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)

In [58]: pred3=regressor.predict(x_test)
    ...: pred3
Out[58]: array([-1.40581786, -0.36686459,  0.33060417, -1.46243845,  0.45614424])

In [59]: from sklearn.metrics import r2_score
    ...: r1=r2_score(y_test,pred3)
    ...: r1
Out[59]: 0.7880641652148291

In [60]:
```

Accuracy:78.80% ~ 79%

C)Decision Tree:

Code:

```
@author: nitik
"""

#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
dataset=pd.read_csv(r"C:\Users\nitik\Desktop\Internal_ML_Project_Final\50_Startups.csv")
dataset=dataset.fillna(0)
dataset
#%%
x=dataset.iloc[:,[0,2]].values
y=dataset.iloc[:,-1].values
print(x)
print(y)
#%%
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.9,random_state=21)
x_train
#%%
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x1_train = sc.fit_transform(x_train)
x1_test = sc.fit_transform(x_test)
y_train=sc.fit_transform(y_train.reshape(-1,1))
#%%
from sklearn.tree import DecisionTreeRegressor

regressor=DecisionTreeRegressor(random_state=0)
regressor.fit(x,y)
#%%
pred=regressor.predict(x_test)
pred
#%%
from sklearn.metrics import r2_score
r=r2_score(y_test,pred)
r
```

Output:

```
    ...:
    ...: regressor=DecisionTreeRegressor(random_state=0)
    ...: regressor.fit(x,y)
Out[64]:
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=0, splitter='best')

In [65]: pred=regressor.predict(x_test)
    ...: pred
Out[65]:
array([155752.6 ,  65200.33,  69758.98, 107404.34, 132602.65, 191050.39,
       108733.99,  97483.56, 125370.37,  42559.73,  99937.59, 111313.02,
       122776.86, 144259.4 , 110352.25,  96778.92, 192261.83, 134307.35,
       103282.38,  14681.4 ,  78239.91,  64926.08,  35673.41,  49490.75,
       101004.64,  89949.14, 191792.06, 105008.31, 108552.04,  97427.84,
       182901.99, 105733.54, 126992.93,  71498.49, 156122.51, 124266.9 ,
       156991.12, 141585.52,  81005.76, 118474.03,  81229.06,  90708.19,
        77798.83, 152211.77, 146121.95])

In [66]: from sklearn.metrics import r2_score
    ...: r=r2_score(y_test,pred)
    ...: r
Out[66]: 1.0
```

Accuracy: 100% (Impractical)

Now we observe that Linear Regression was most successful amongst the three since it is giving us ~ 96% r2_score.

Note : Decision Tree gives 100% r2_score hence we reject it due to it's impracticability.

6. SUMMARY :

- ✓ Past data of '50_Startups' was recorded and analysed.
- ✓ Relations between the data was understood.
- ✓ Accordingly, appropriate ML Model was built and its accuracy was calculated.
- ✓ Linear Regression was the best ML Model to find accuracy.
- ✓ **The high accuracy of the Model proves its success.**

Observing from the above outputs, we hence understand that "Linear regression" is most suited in this scenerio.

7. REFERENCES

- ✓ Machinelearningmastery.com
- ✓ Kaggle.com
- ✓ Wikipedia
- ✓ Towardsdatascience.com