EXPOSYS DATA LABS

Bengaluru,Karnataka,560064

Internship report on

“Predicting Company Profit with Machine Learning: A Comparative Analysis of Regression Algorithms using R&D Spend, Administration Cost, and Marketing Spend Data.”

A Dissertation work submitted in partial fulfilment of the requirement for the

award of the degree of

Internship

By

Name- **Swapnil Jyot**

College-**Rajkiya Engineering college ,Kannauj**

Under Guidance of

EXPOSYS DATA LABS

Abstract

The objective of this project is to build a machine learning model that can predict the profit of a company based on its R&D Spend, Administration Cost, and Marketing Spend. The dataset consists of 50 companies with their corresponding profit and expenses. In this project, we have implemented different regression algorithms, divided the dataset into training and testing sets, and calculated various regression metrics to choose the best model.then we have to make graph between every independent and dependent variable .For analysing that how dependent variable has been changed with respect to independent variable .This process give close conclusion of data that helps in training our model . Now or data has been ready for training and testing. We have to split our dataset in train and test dataset the train dataset are used to trained our model and test dataset has been use to test predictions our model. On the basis of this we have to check the accuracy of all the machine learning regression models that which model have highest accuracy we have also plot it on the bar graph. The project was implemented using Python programming language and the dataset was obtained from the given link

TABLE OF CONTENTS

1.INTRODUCTION

In this modern era, businesses are constantly trying to maximize their profits by increasing their revenue and minimizing their expenses. One way to achieve this is to use machine learning techniques to analyze and predict profits based on various factors. In this project, we aim to build a machine learning model that can predict the profit of a company based on its expenses.

1.1 Background

Machine learning is a branch of artificial intelligence that deals with the development of algorithms and models that enable computers to learn from data and make predictions or decisions based on that learning. The origins of machine learning can be traced back to the 1940s and 1950s, when early researchers began exploring the concept of artificial neural networks, which are computational models loosely based on the structure and function of biological neurons in the human brain.

Over the next several decades, machine learning researchers developed a variety of algorithms and techniques for analyzing and modeling data, including linear regression, decision trees, Bayesian networks, and support vector machines. However, progress was slow due to limitations in computing power and data availability.

In the 2000s, the growth of big data and advances in computing technology, including the development of graphics processing units (GPUs) and cloud computing, led to a revolution in machine learning. Researchers were able to develop more sophisticated algorithms and train larger models using vast amounts of data, leading to breakthroughs in areas such as computer vision, natural language processing, and speech recognition.

Today, machine learning is used in a wide range of applications, from self-driving cars to personalized medicine to fraud detection. Its ability to analyze large and complex datasets and identify patterns and insights that would be difficult or impossible for humans to uncover has made it a valuable tool in many industries and fields.

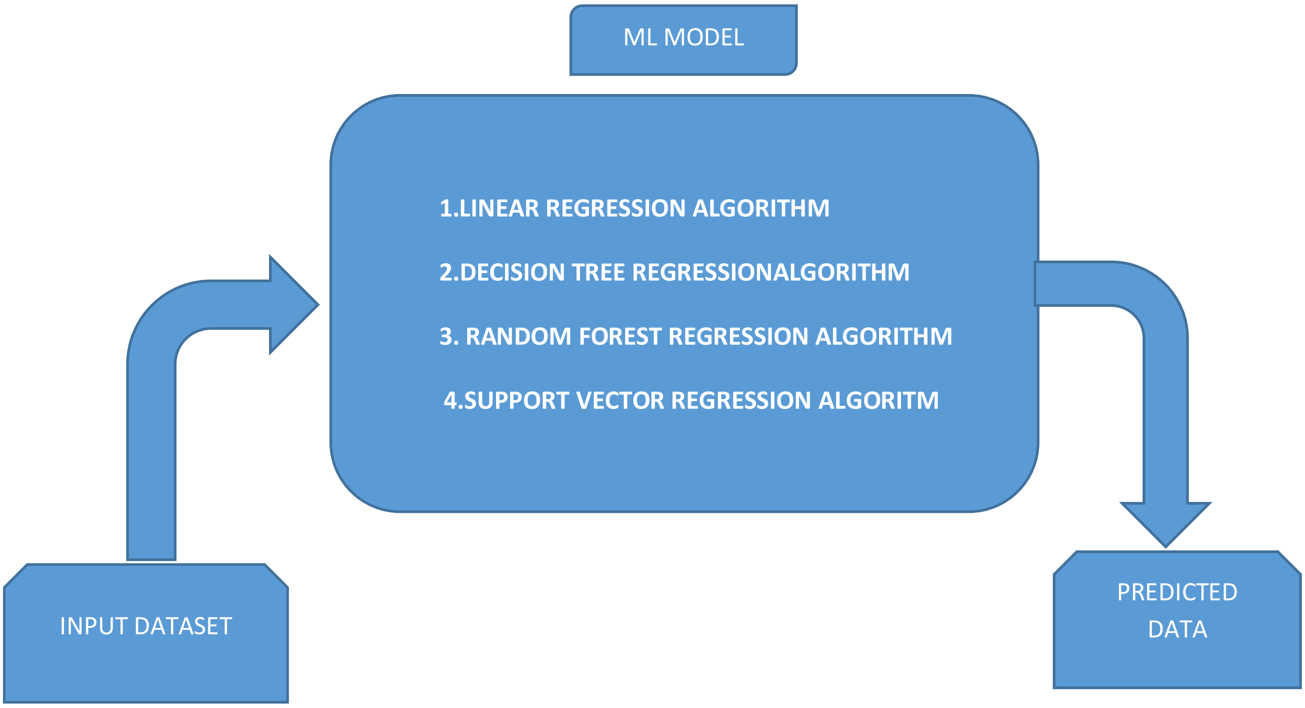
1.2 Goal

The goal of this project is that is to create a machine learning model using the different regression algorithms like linear regression, decision tree regression, random forest regression , support vector regression model with the use of all these model I have to create a machine learning environment that predict the future values of the given data . To reach the goals of the project, it is required to address the following questions:

- How should in analyse or examine the given datasets
- Making datasets compatible for the models
- How to create the models using the python libraries
- How to trained the models
- How to test the models

1.3 Setup

Under this setup I have top used four regression model and under training data set we have to trained the different model and estimate the output


ML MODEL
1.LINEAR REGRESSION ALGORITHM
2.DECISION TREE REGRESSIONALGORITHM
3. RANDOM FOREST REGRESSION ALGORITHM
4.SUPPORT VECTOR REGRESSION ALGORITM
INPUT DATASET
PREDICTED DATA

2.EXISTING METHODS

In the past, various regression algorithms have been used to predict the profit of a company. Some of the most commonly used algorithms are linear regression, decision tree regression, and support vector regression. However, the performance of these algorithms may vary depending on the dataset and the problem at hand.

2.1Machine learning basics

Types of machine learning

- **Supervised Learning**: In supervised learning, the machine learning model is trained on labeled data, meaning that the data is already classified or has known output values. The model is then able to predict the output values for new, unseen data. Examples of supervised learning include classification and regression problems.

Types of supervised learning model

i. <u>Classification</u>: In classification, the goal is to predict the categorical output variable or class label for new, unseen data. The model is trained on a dataset where each data point is labeled with a class label. The model learns to map input features to a class label, and can then predict the class label for new data. Examples of classification problems include image classification, spam detection, and sentiment analysis

ii. <u>Regression</u>: In regression, the goal is to predict a continuous output variable based on input features. The model is trained on a dataset where each data point has a known output value. The model learns to map input features to a numerical output value, and can then predict output values for new data. Examples of regression problems include predicting housing prices, stock prices, and customer lifetime value

2. There are also other subtypes of supervised learning, such as:
    i. Binary Classification: A specific case of classification where the output variable has only two possible values.
    ii. Multi-class Classification: A type of classification where the output variable can have more than two possible values.
    iii. Ordinal Regression: A type of regression where the output variable is ordered, such as rating a product on a scale from 1 to 5.
    iv. Time-series Prediction: A type of regression where the input data is a sequence of time-stamped data points, and the goal is to predict future values of the time series.

- **Unsupervised Learning:** In unsupervised learning, the machine learning model is trained on unlabeled data, meaning that the data has no known output values. The model is then able to discover patterns and relationships in the data on its own. Examples of unsupervised learning include clustering and anomaly detection.

Types of unsupervised learning model

i. Clustering: In clustering, the goal is to group similar data points together based on their similarities in input features. The model is trained on a dataset where each data point has a set of input features, and the model learns to group similar data points together based on these features. Examples of clustering problems include customer segmentation and image segmentation.

ii. Anomaly Detection: In anomaly detection, the goal is to identify unusual data points that do not conform to the expected patterns in the data. The model is trained on a dataset where the majority of data points are expected to be normal, and the model learns to identify unusual or anomalous data points that deviate from the norm. Examples of anomaly detection problems include fraud detection and network intrusion detection.

There are also other subtypes of unsupervised learning, such as:

iii. .Dimensionality Reduction: A type of unsupervised learning where the goal is to reduce the number of input features while preserving as much of the original information as possible. Examples of dimensionality reduction techniques include principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE).

iv. Association Rule Learning: A type of unsupervised learning where the goal is to discover relationships or associations between variables in a dataset. Examples of association rule learning problems include market basket analysis and recommendation systems.

v. Generative Models: A type of unsupervised learning where the goal is to generate new data that is similar to the original dataset. Examples of generative models include variational autoencoders (VAEs) and generative adversarial networks (GANs).

- **Reinforcement Learning:** In reinforcement learning, the machine learning model learns through trial and error by receiving feedback in the form of rewards or penalties for its actions. The model is then able to make decisions that optimize the rewards it receives over time. Examples of reinforcement learning include game playing and robotics.

i. Model-based Reinforcement Learning: In model-based reinforcement learning, the model learns the dynamics of the environment by constructing a model of the environment based on previous experience. The model then uses this information to make decisions that maximize the expected rewards. Examples of model-based reinforcement learning algorithms include dynamic programming and Monte Carlo methods.

ii. Model-free Reinforcement Learning: In model-free reinforcement learning, the model learns directly from experience, without constructing a model of the environment. The model learns to map the current state of the environment to an action that maximizes the expected rewards. Examples of model-free reinforcement learning algorithms include Q-learning and SARSA.

There are also other subtypes of reinforcement learning, such as:

iii. Policy-Based Reinforcement Learning: A type of reinforcement learning where the model learns a policy, which is a mapping from states to actions that maximizes the expected rewards. Examples of policy-based reinforcement learning algorithms include policy gradient methods and actor-critic methods.

iv. Value-Based Reinforcement Learning: A type of reinforcement learning where the model learns a value function, which is a mapping from states to expected rewards. The value function is used to determine the optimal action to take in each state. Examples of value-based reinforcement learning algorithms include Q-learning and SARSA.

v. Multi-Agent Reinforcement Learning: A type of reinforcement learning where multiple agents learn to cooperate or compete with each other to achieve a common goal. Examples of multi-agent reinforcement learning include game theory and swarm intelligence.

vi. <u>Deep Reinforcement Learning:</u> A type of reinforcement learning that combines reinforcement learning with deep neural networks to learn complex, high-dimensional representations of the environment. Examples of deep reinforcement learning algorithms include deep Q-networks (DQNs) and deep policy gradient methods.

Additionally, there are some hybrid methods that combine aspects of these main types, such as semi-supervised learning, where the model is trained on both labeled and unlabeled data, and transfer learning, where knowledge from a pre-trained model is used to improve performance on a related task

2.2Types of machine learning algorithm

There are several types of machine learning algorithm models. Here are some of the most common ones:

- **Linear models:** Linear models are used for regression and classification tasks. They work by fitting a linear function to the input features, with the goal of minimizing the difference between the predicted output and the actual output. Examples of linear models include linear regression, logistic regression, and support vector machines.

- **Decision trees:** Decision trees are a type of model that uses a tree-like structure to make decisions. The tree is constructed by splitting the data into smaller and smaller subsets based on the input features until a decision is reached. Decision trees are often used for classification tasks.

- **Neural networks:** Neural networks are a type of model inspired by the structure of the human brain. They consist of multiple layers of interconnected nodes that process the input data and generate an output. Neural networks are used for a variety of tasks, including image recognition, natural language processing, and speech recognition.

- **Random forests**: Random forests are a type of ensemble learning algorithm that combines multiple decision trees to make predictions. Each tree in the forest is trained on a different subset of the data, and the final prediction is made by combining the predictions of all the trees.

- **Support vector machines:** Support vector machines are a type of model used for classification tasks. They work by finding the hyperplane that maximally separates the different classes of data.

- **Clustering models:** Clustering models are used to group similar data points together. Examples of clustering models include k-means clustering and hierarchical clustering.

- **Association rule learning models:** Association rule learning models are used to discover patterns in data. They work by identifying frequent itemsets, which are sets of items that frequently occur together, and generating rules based on those itemsets. Examples of association rule learning models include Apriori and FP-Growth.

These are just a few examples of the many types of machine learning algorithm models that exist. The choice of model depends on the specific problem at hand and the type of data being used.

3.PROPOSED METHOD WITH ARCHITECTURE

In this project, we construct different regression algorithms such as linear regression, decision tree regression, random forest regression, and support vector regression. We use Python as our programming language and various libraries such as pandas, scikit-learn, and matplotlib for data preprocessing, modeling, and visualization. The architecture of the proposed method involves dividing the data into train and test sets, constructing different regression models, evaluating their performance using different metrics, and choosing the best model.

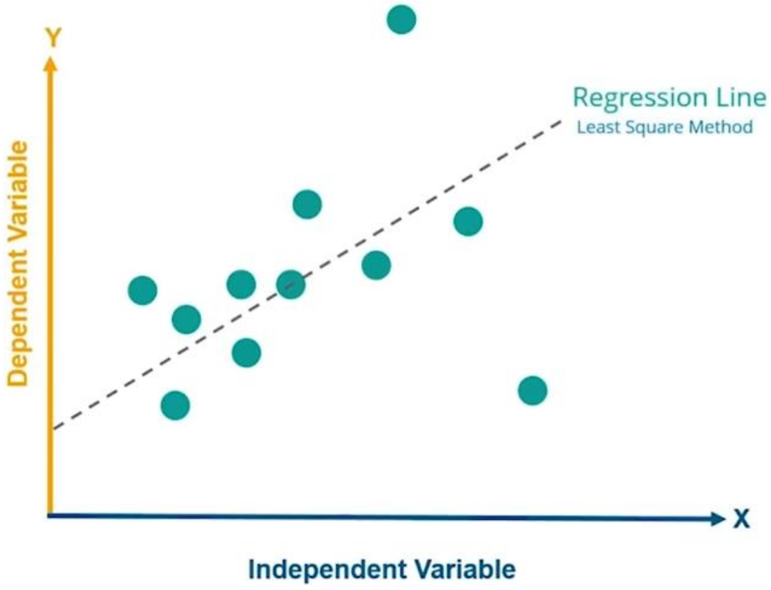
3.1 LINEAR REGRESSION ALGORITHM :-

3.1.1 INTRODUCTION OF ALGORITHM-

- Regression analysis is a form of predictive modeling technique which investigate the modelling technique which investigates the relationship between a dependend and independent variable
- Three major uses for regression analysis are
    - Determining the strength of predictors
    - Forecastingh an effect and
    - Trending Forecasting
- It is used for the continous variables

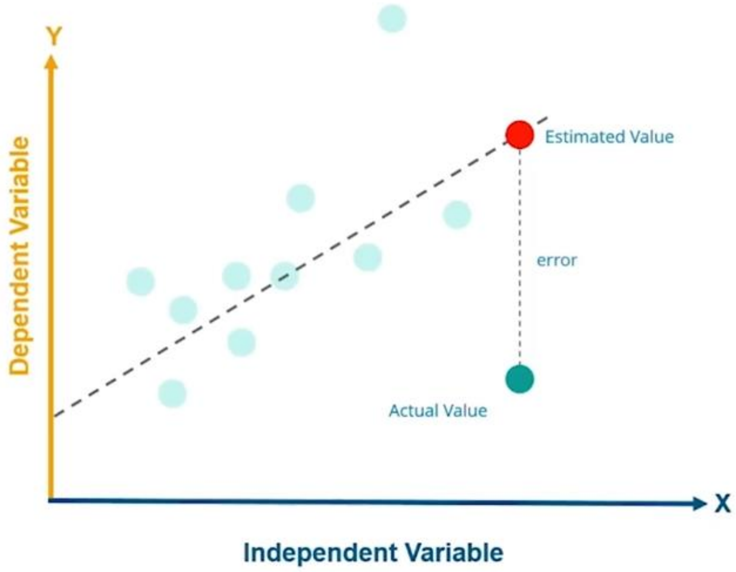
- Linear regression can be classified into two types: simple linear regression and multiple linear regression. Simple linear regression involves only one independent variable, whereas multiple linear regression involves more than one independent variable.

3.1.2 WORKING OF ALGORITHM-

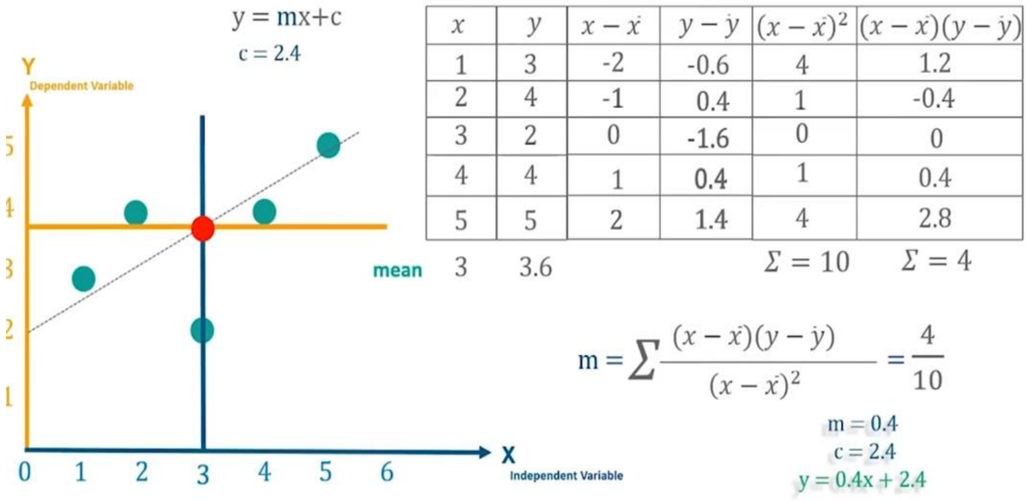


This is our best fit line that I have to predicted with our given data and this line is called regression line

Now our next task is that to reduce te error between the actual value and the predicted value and the line wich has the minimum error between the acual point and the predicted point is called the line of linear regression and best fit line .



Lets understand the how we have to predict the line of regression or lets understand the linear regression algorithm



	x	y	$x-\dot{x}$	$y-\dot{y}$	$(x-\dot{x})^2$	$(x-\dot{x})(y-\dot{y})$
	1	3	-2	-0.6	4	1.2
	2	4	-1	0.4	1	-0.4
	3	2	0	-1.6	0	0
	4	4	1	0.4	1	0.4
	5	5	2	1.4	4	2.8
mean	3	3.6			$\Sigma = 10$	$\Sigma = 4$

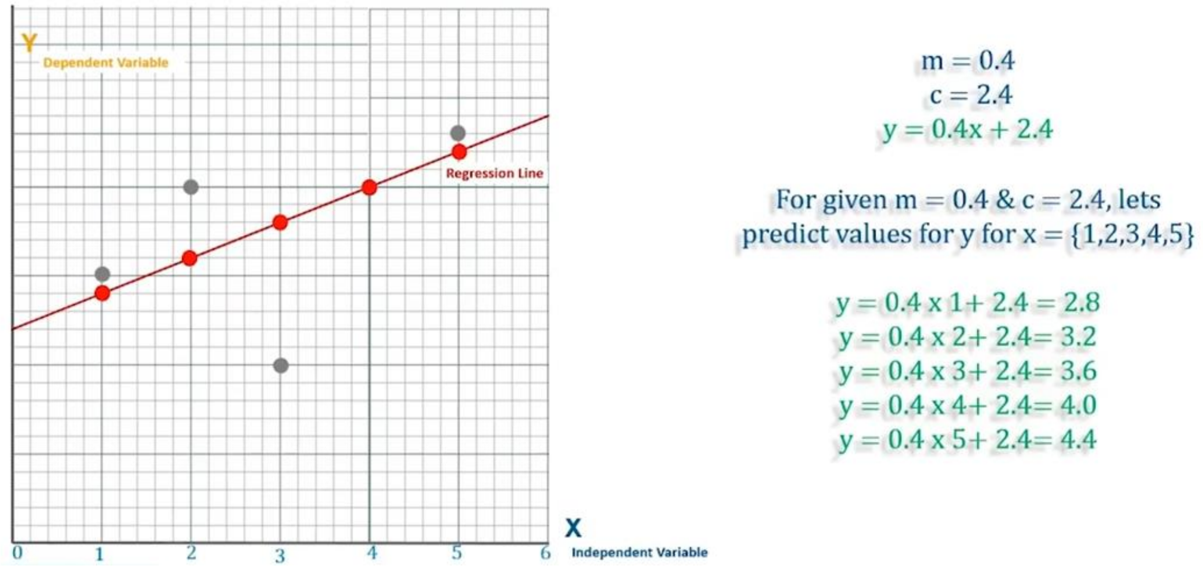
$$\mathrm{m} = \sum \frac{(x-\dot{x})(y-\dot{y})}{(x-\dot{x})^2} = \frac{4}{10}$$

$$\mathrm{m} = 0.4$$
$$\mathrm{c} = 2.4$$
$$\mathrm{y} = 0.4\mathrm{x} + 2.4$$

You can clearly see that the how we have to predict the slope and the equation of the regression line

Comparison betweeen the the line of regression line and the actual point

Or in other words distance between the actual poin t and the ptredicted values

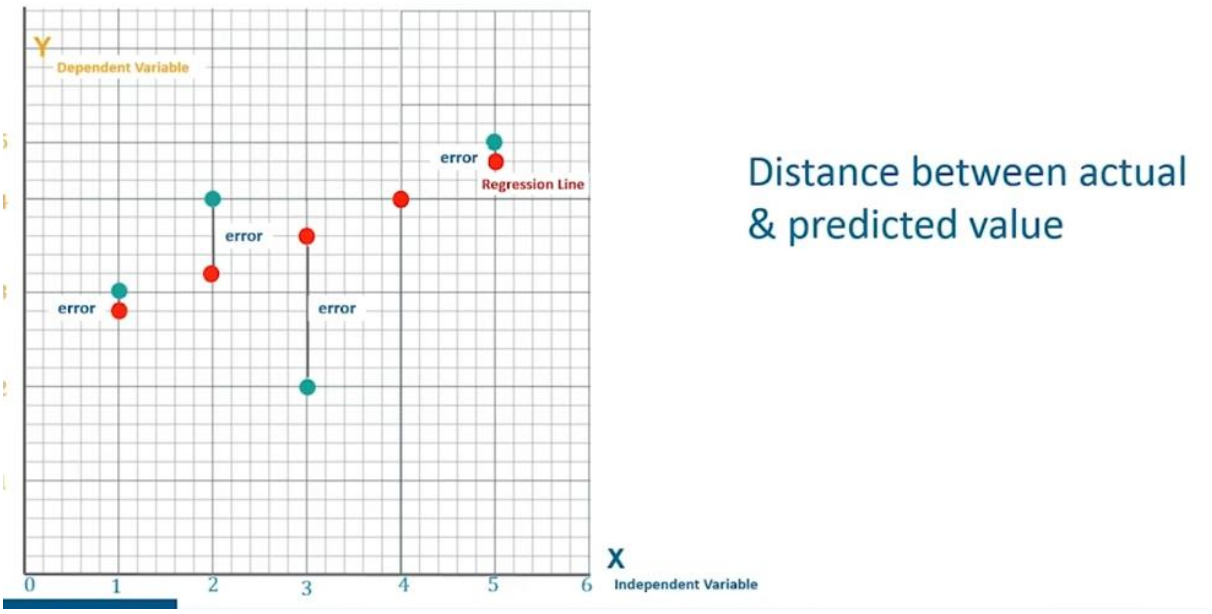


$m = 0.4$
$c = 2.4$
$y = 0.4x + 2.4$

For given $m = 0.4$ & $c = 2.4$, lets predict values for y for $x = \{1,2,3,4,5\}$

$y = 0.4\ x\ 1 + 2.4 = 2.8$
$y = 0.4\ x\ 2 + 2.4 = 3.2$
$y = 0.4\ x\ 3 + 2.4 = 3.6$
$y = 0.4\ x\ 4 + 2.4 = 4.0$
$y = 0.4\ x\ 5 + 2.4 = 4.4$

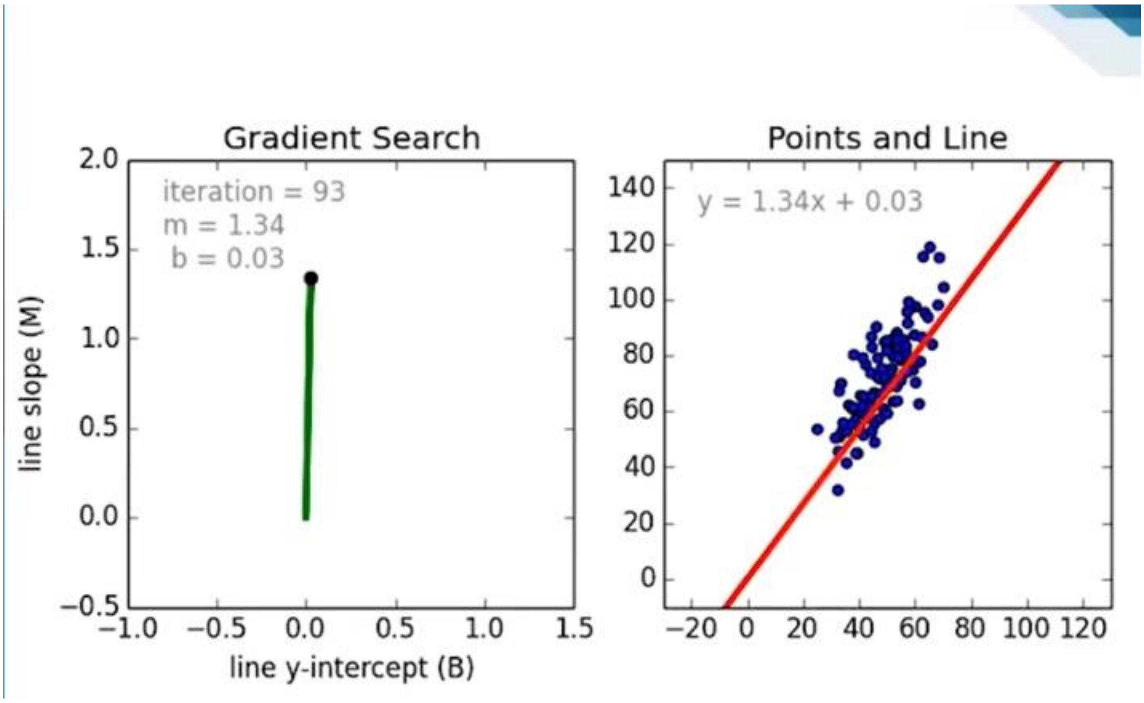
Error comparision between them



Distance between actual & predicted value

Hence computer used this technique to finding the best fit line by illterating the value of m from 0 to 1 and compares the distance between the actual value and the predicted value the value of m for which the distance between the actual value and the predicted value is minimum will be selected as the best fit line

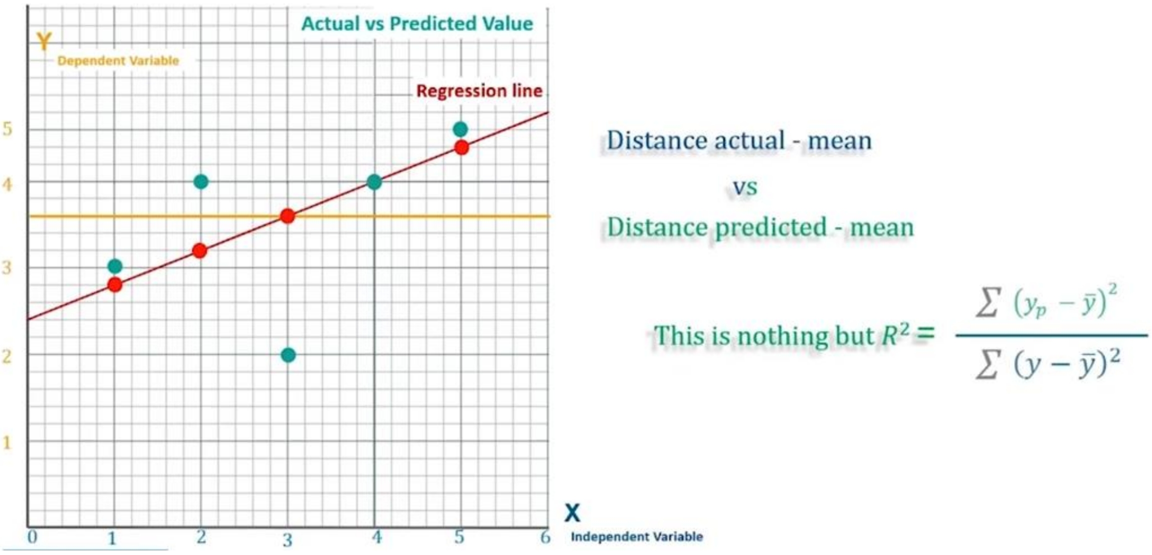
You can see in the following graph.



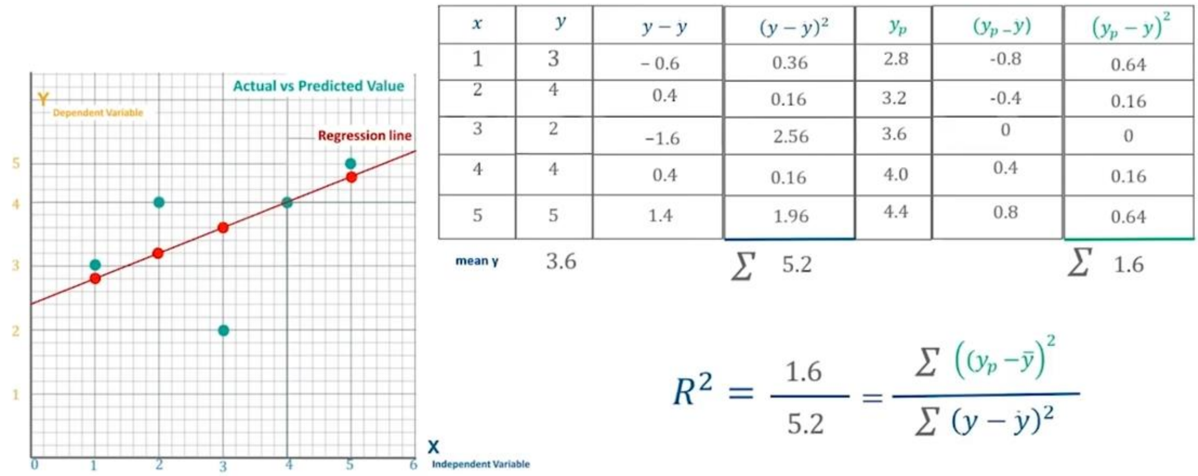
Now calculating the goodness of fit.

CALCULATION OF R^2 VALUE:-

- R-squared value is a statistical measure of how close the data are to the fitted regression line
- It is also known as **coefficient of determination ,** or the **coefficient of multiple determination**



Now calculation of r-squared from the given dataset



x	y	$y-\dot{y}$	$(y-\dot{y})^2$	y_p	$(y_p - \dot{y})$	$(y_p-\dot{y})^2$
1	3	− 0.6	0.36	2.8	-0.8	0.64
2	4	0.4	0.16	3.2	-0.4	0.16
3	2	−1.6	2.56	3.6	0	0
4	4	0.4	0.16	4.0	0.4	0.16
5	5	1.4	1.96	4.4	0.8	0.64
mean y	3.6		$\sum$ 5.2			$\sum$ 1.6

$$R^2 = \frac{1.6}{5.2} = \frac{\sum\left((y_p-\bar{y}\right)^2}{\sum(y-\dot{y})^2}$$

Now here r-squared value is 0.3 which is not very good as if we increase the r-squared value from 0.0 to 0.9 then our actual value come closer to the actual value when it is equal to 1 then our actual value comes on the regression line thus the r-squared value tells the accuracy of our model. If our r-square value is very less then our actual value is very far away from the data.

CALCULATION OF MEAN OBSOLUTE ERROR:-

- Mean Absolute Error (MAE) is a metric used to evaluate the performance of a machine learning model, particularly for regression problems like linear regression. It measures the average absolute difference between the actual and predicted values of the target variable.

- In the context of linear regression, MAE is calculated by taking the absolute difference between the predicted and actual target values for each data point, then taking the average of those differences. The formula for MAE is:

- MAE = (1/n) * sum(|y_actual - y_predicted|)
- y_actual is the actual target value

- y_predicted is the predicted target value
- n is the total number of data points in the dataset
- The MAE value indicates the average magnitude of the errors in the predictions made by the model. A lower MAE value indicates better accuracy and performance of the model, while a higher MAE value indicates poorer performance.
- Overall, the mean absolute error is a useful metric to assess the performance of a linear regression model and can be used to compare the performance of different models or to fine-tune model parameters to improve accuracy.

CALCULATION OF MEAN OBSOLUTE ERROR:-

- Mean Squared Error (MSE) is another commonly used metric for evaluating the performance of regression models like linear regression. It measures the average squared difference between the actual and predicted values of the target variable.
- In the context of linear regression, MSE is calculated by taking the squared difference between the predicted and actual target values for each data point, then taking the average of those squared differences. The formula for MSE is:
- MSE = (1/n) * sum((y_actual - y_predicted)^2)
- where:
- y_actual is the actual target value
- y_predicted is the predicted target value
- n is the total number of data points in the dataset
- The MSE value gives an idea of the average magnitude of the squared errors between the predicted and actual target values. A lower MSE value indicates better accuracy and performance of the model, while a higher MSE value indicates poorer performance.
- MSE has the advantage over MAE in that it gives more weight to large errors, which may be important in some applications. However, because MSE is calculated by taking the squared differences, it is sensitive to outliers and may not be as robust to outliers as MAE.

- In general, both MAE and MSE are useful metrics to evaluate the performance of linear regression models and can be used to compare the performance of different models or to fine-tune model parameters to improve accuracy.

3.2 DECISION TREE REGRESSION ALGORITHM :-

3.2.1 INTRODUCTION OF ALGORITHM-

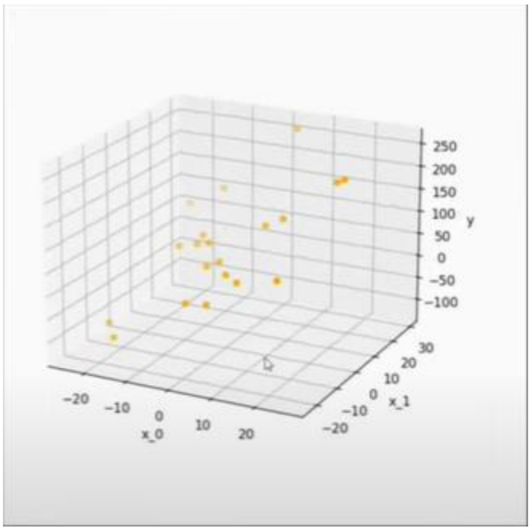
- Decision tree regression is a supervised machine learning algorithm used for predicting a continuous dependent variable (also called the target variable) based on one or more independent variables (also called features or predictors). The algorithm uses a tree-like model of decisions and their possible consequences to make the predictions.
- In decision tree regression, the independent variables are split into different subsets based on their values. The algorithm chooses the variable that best splits the data into subsets that are as homogeneous as possible in terms of the dependent variable. This process continues recursively until the subsets are sufficiently small, and a prediction for the dependent variable is made for each subset based on the mean value of the dependent variable in that subset.
- The decision tree is constructed by splitting the data into subsets based on the values of the independent variables, using a set of rules that are learned from the data. The rules are represented as a tree structure, where each node represents a test of the value of an independent variable, and each branch represents the outcome of that test.
- To make a prediction for a new data point, the algorithm follows the tree structure until it reaches a leaf node, which provides the prediction for the dependent variable based on the mean value of the dependent variable in the subset represented by that leaf node.
- The main advantage of decision tree regression is that it can handle non-linear relationships between the independent variables and the dependent variable, as well as interactions between them. It is also easy to interpret and visualize the decision tree, which can help in understanding the relationships between the variables.

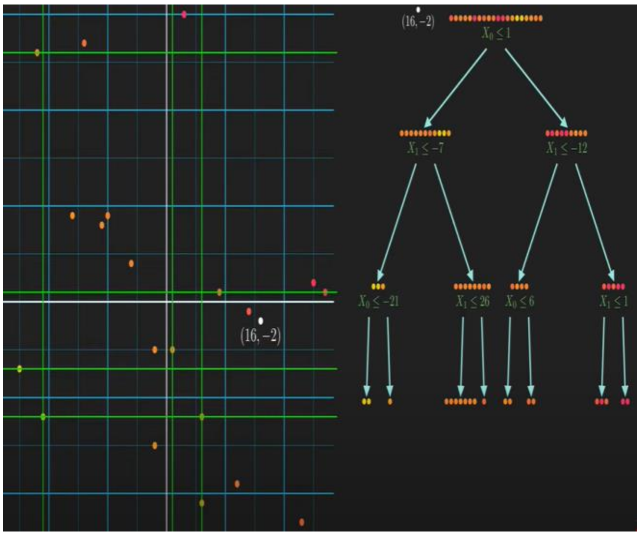
- However, decision tree regression can be prone to overfitting if the tree is too complex or if there is noise or outliers in the data. Therefore, it is important to use techniques such as pruning, regularization, or ensemble methods to reduce the complexity of the tree and improve its generalization performance.

- Some specific applications of decision tree regression include predicting the price of a house based on its features such as location, size, number of rooms, etc., forecasting sales based on marketing expenditure, and estimating the crop yield based on weather and soil conditions. The decision tree algorithm is also often used in ensemble methods such as random forests to improve the accuracy of the predictions.

3.2.2 LETS UNDERSTAND THE CONCEPT BEHIND THE DECISION TREE REGRESSION

Let we have two two feature x[0] and x[1] and y is target variable   now what should I have to do that the

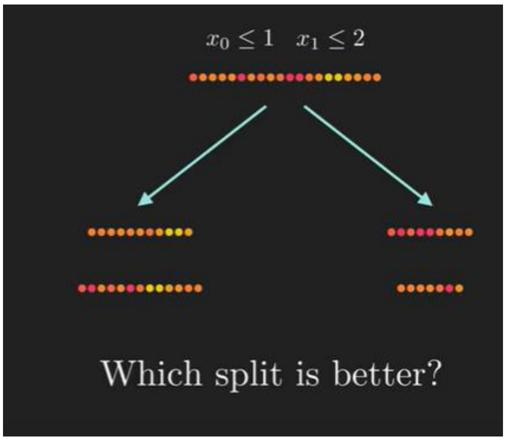


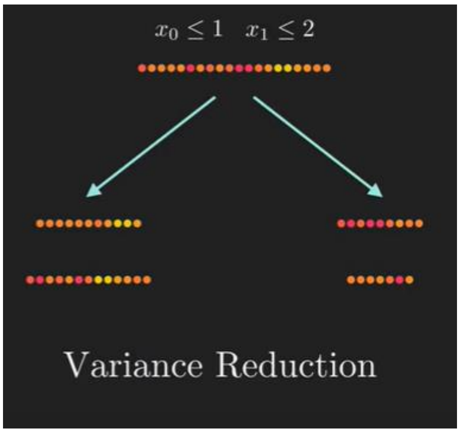


here I have basically make a plot where the horizontal axis represent the x[0]and vertical axis represent represent the x[1]

Points and darker point represent the heigher value of y and lighter point represent the lighter value of y

Now here in decision tree regresson algorithm we have to take one feature independent variable as a root node now one question comes that



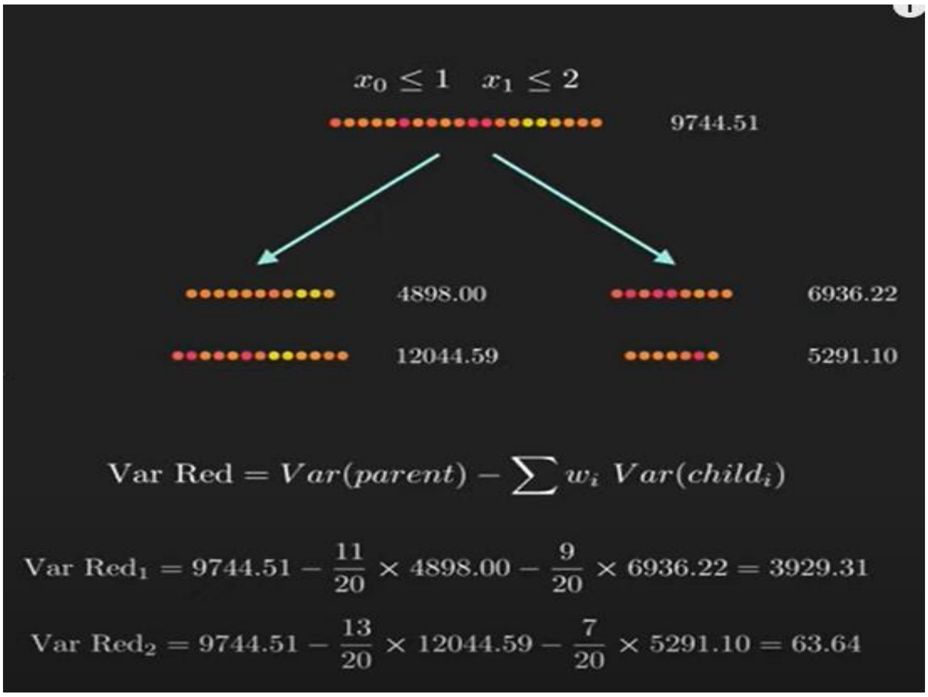


which is the best child node to find this we need to calculate this by which split will decreasing the impurity of the child node the most for this we have to used the varience reduction technique just like entropy or the gini index in the classification problem.

Note that all the yellow and the red bubbles are the y values .

Note that heigher value of varience is heigher value of impurity .

$$Var = \frac{1}{n}\sum(y_i - \bar{y})^2$$



$$\text{Var Red} = Var(parent) - \sum w_i \, Var(child_i)$$

$$\text{Var Red}_1 = 9744.51 - \frac{11}{20} \times 4898.00 - \frac{9}{20} \times 6936.22 = 3929.31$$

$$\text{Var Red}_2 = 9744.51 - \frac{13}{20} \times 12044.59 - \frac{7}{20} \times 5291.10 = 63.64$$

This will tells that the first one decrease the impurity much more than the second one hence we have to take first one as a child node than the second one

Now for any random value of x[0] and x[1] we have to take mean of the y when I have to reached that leaf node by passing all the conditions.

But there is one draw back of the decision tree which is over fitting which is that high training accuracy and low testing accuracy so for solving this problem we have to used the random forest algorithm.

3.3 RANDOM FOREST REGRESSION ALGORITHM:-

3.3.1 INTRODUCTION OF ALGORITHM-

- Random forest regression is a machine learning algorithm used for regression tasks, where the goal is to predict a continuous variable. It is an extension of the random forest algorithm, which is primarily used for classification tasks.

- Random forest regression works by building a large number of decision trees, each trained on a random subset of the input features and a random subset of the training data. The algorithm then combines the predictions of all the trees to obtain a final prediction.

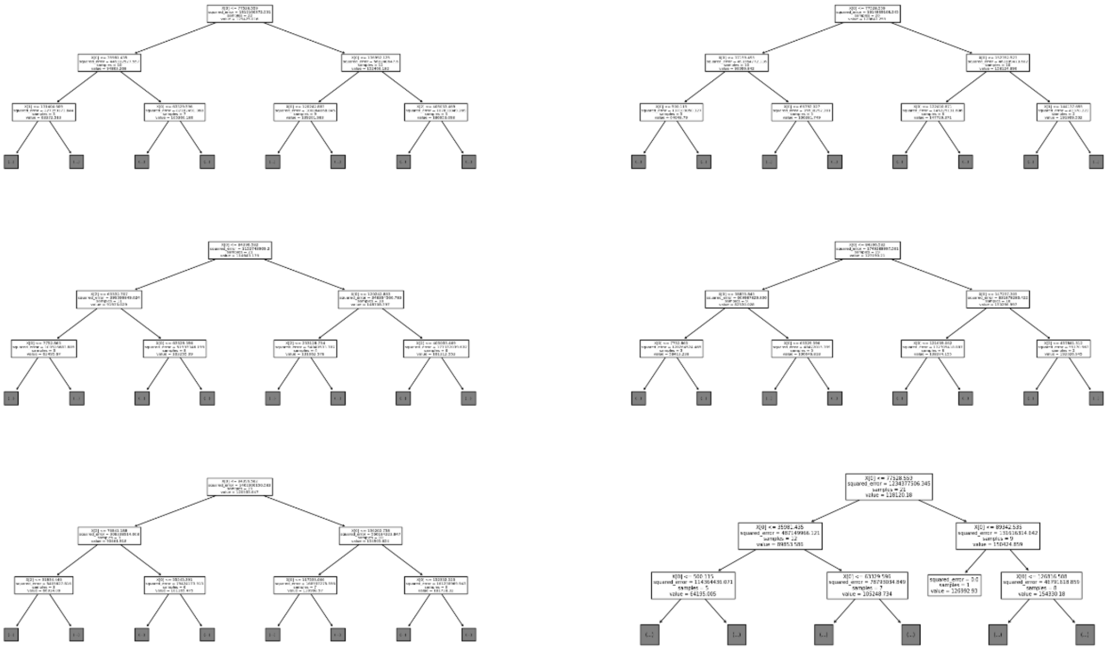
3.3.2 WORKING

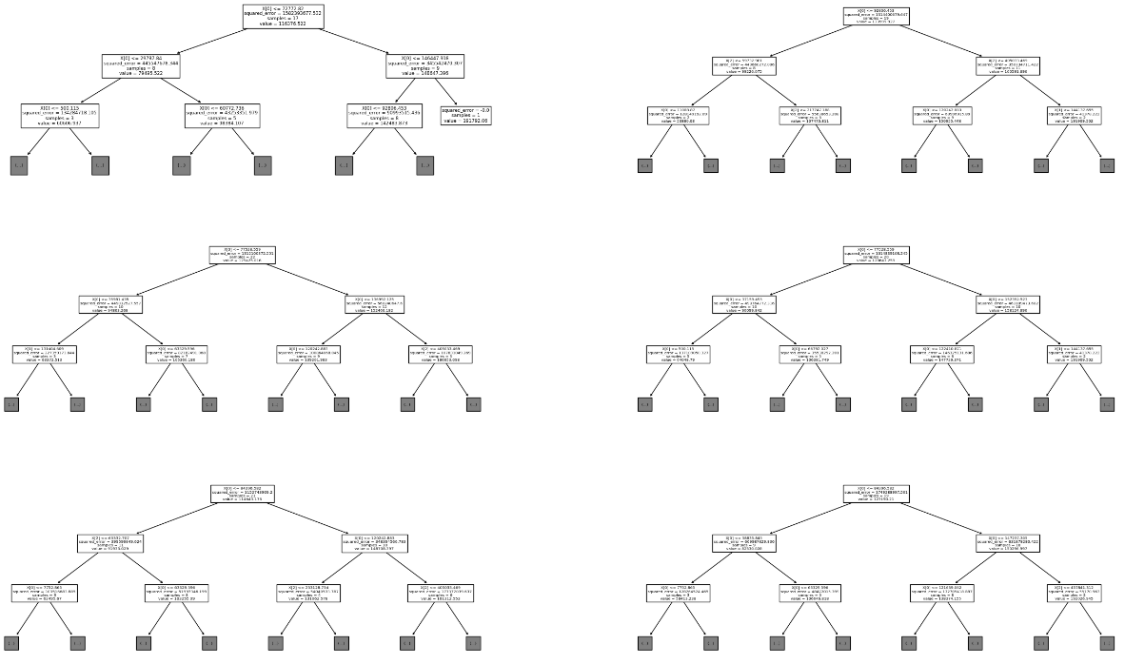
- The decision tress is made by taking the sampling of rows and features of the data with replacement .

- In a random forest regression model, the decision trees are constructed such that the variance between the trees is maximized, while the correlation between the trees is minimized. This helps to reduce overfitting and improve the generalization performance of the model.

- The random forest regression algorithm is popular because it is relatively easy to use, and can be very effective in a wide range of applications. It can handle large datasets with many input features, and is robust to noise and outliers in the data.

- And lastly in random forest regressor we have to take the mean of the output of the all decision tree and predict this value.





3.4 SUPPORT VECTOR REGRESSION ALGORITHM:-

3.4.1 INTRODUCTION OF ALGORITHM-

- Support Vector Regression (SVR) is a type of regression analysis based on Support Vector Machines (SVMs). In traditional regression analysis, the objective is to find a line or a curve that best fits the data, while in SVR the aim is to find a hyperplane that maximizes the margin between the predicted values and the actual values.

- SVR works by mapping the input data to a high-dimensional feature space, where a hyperplane is constructed to separate the predicted values from the actual values. The hyperplane is selected in such a way that it maximizes the margin, i.e., the distance between the hyperplane and the closest data points on either side. The data points that are closest to the hyperplane are called support vectors, and they play a crucial role in determining the shape of the hyperplane.

- SVR uses a kernel function to map the input data to a high-dimensional feature space. The most commonly used kernel functions are linear, polynomial, and radial basis function (RBF).

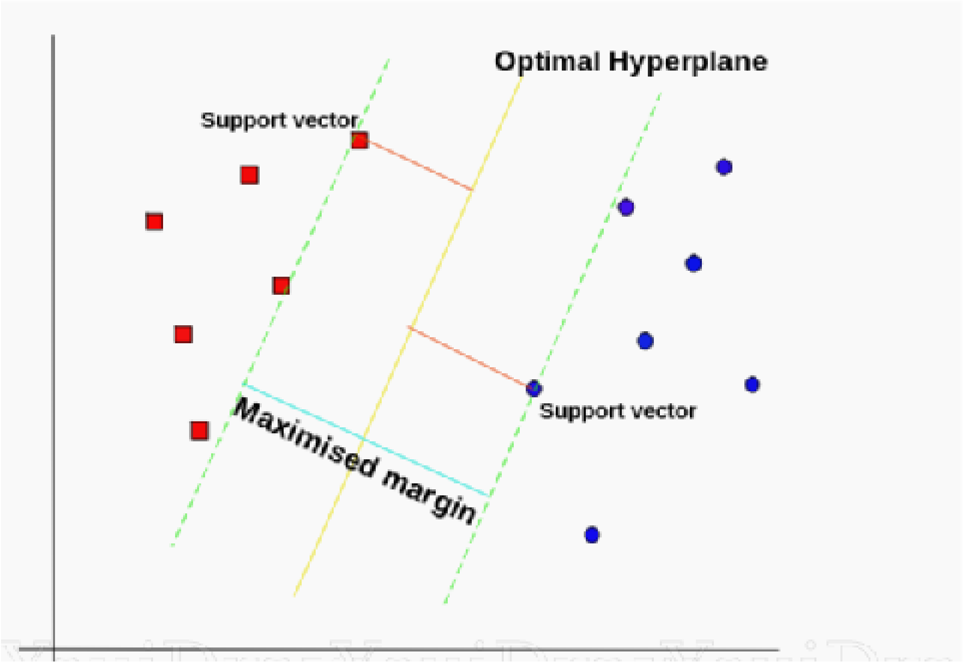
- SVR can be used for both linear and nonlinear regression problems. It is particularly useful when dealing with high-dimensional data with a small number of samples. SVR has been successfully applied in various fields, including finance, engineering, and bioinformatics.

3.4.2 LETS UNDERSTAND THE CONCEPT BEHIND THE SUPPORT VECTOR REGRESSION

ALGORITHM:-

Support Vector Machine (SVM) is a very popular Machine Learning algorithm that is used in both Regression and Classification. Support Vector Regression is similar to Linear Regression in that the equation of the line is y= wx+b In SVR, this straight line is referred to as hyperplane. The data points on either side of the hyperplane that are closest to the hyperplane are called Support Vectors which is used to plot the boundary line.

Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value (Distance between hyperplane and boundary line), a. Thus, we can say that SVR model tries satisfy the condition $-a < y-wx+b < a$. It used the points with this boundary to predict the value.



To understand the figure you need to know the following term:

1. **Hyperplane**: It is a separation line between two data classes in a higher dimension than the actual dimension. In SVR it is defined as the line that helps in predicting the target value.

2. **Kernel**: In SVR the regression is performed at a higher dimension. To do that we need a function that should map the data points into its higher dimension. This function is termed as the kernel. Type of kernel used in SVR is Sigmoidal Kernel, Polynomial Kernel, Gaussian Kernel, etc,

3. **Boundary Lines**: These are the two lines that are drawn around the hyperplane at a distance of ε (epsilon). It is used to create a margin between the data points.

4. **Support Vector**: It is the vector that is used to define the hyperplane or we can say that these are the extreme data points in the dataset which helps in defining the hyperplane. These data points lie close to the boundary.

The objective of SVR is to fit as many data points as possible without violating the margin. Note that the classification that is in SVM use of support vector was to define the hyperplane but in SVR they are used to define the linear regression.

2. Selection of Kernel

You can choose any kernel like Sigmoid Kernel, Polynomial Kernel, Gaussian Kernel, etc based upon the problem. All of these kernels have hyperparameters that need to be trained. In this article, I will be taking the Gaussian Kernel. Gaussian Kernel is defined as:

$$[K_G(\vec{x^i}, \vec{x^j}, \vec{\theta}) = \exp\left(\sum_{k}^{N_D} \theta_k \left|x_k^i - x_k^j\right|^2\right)]$$

where:

ND: is the dimension in every data point

x and θ: are the set of hyperparameters.

Selection of kernel is important because if we choose a kernel like the Gaussian, which starts giving zero as the distance between the argument grows then as we start to move away

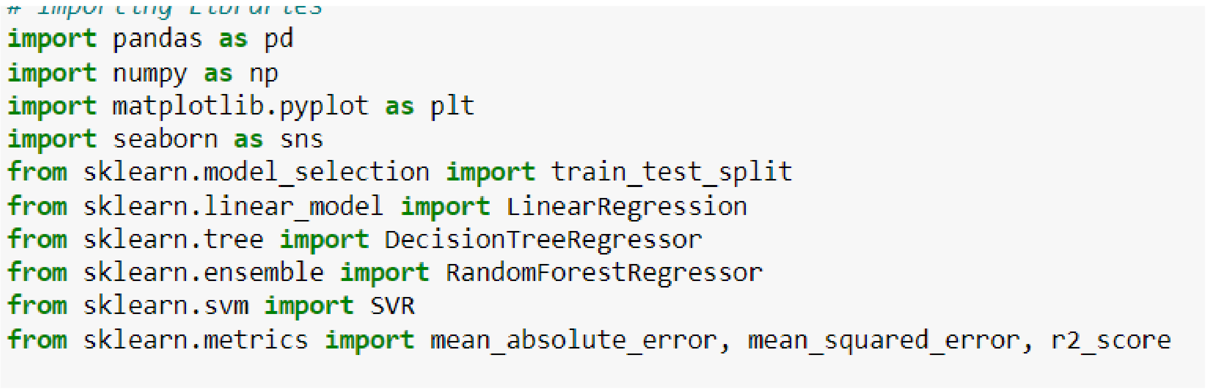
In SVR this training phase is the most expensive part, and lots of research are going on to develop a better way to do it. We can train it using the gradient-based optimization method like CG and minimizing the cost function.

4.METHODOLOGY:

We used the following methodology to create the machine learning model:

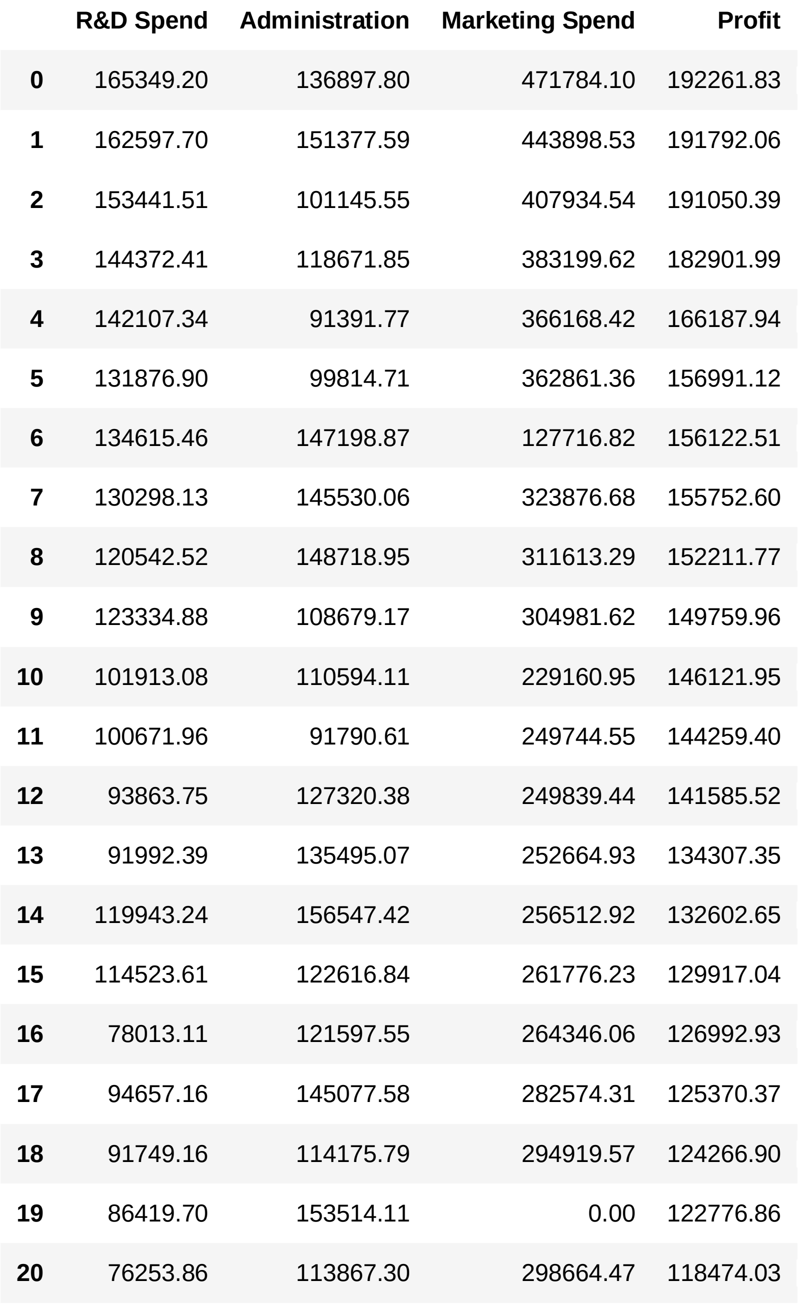
4.1**Data Collection**: We collected the dataset containing the R&D Spend, Administration Cost, Marketing Spend, and Profit of 50 companies.

Import all the libraries

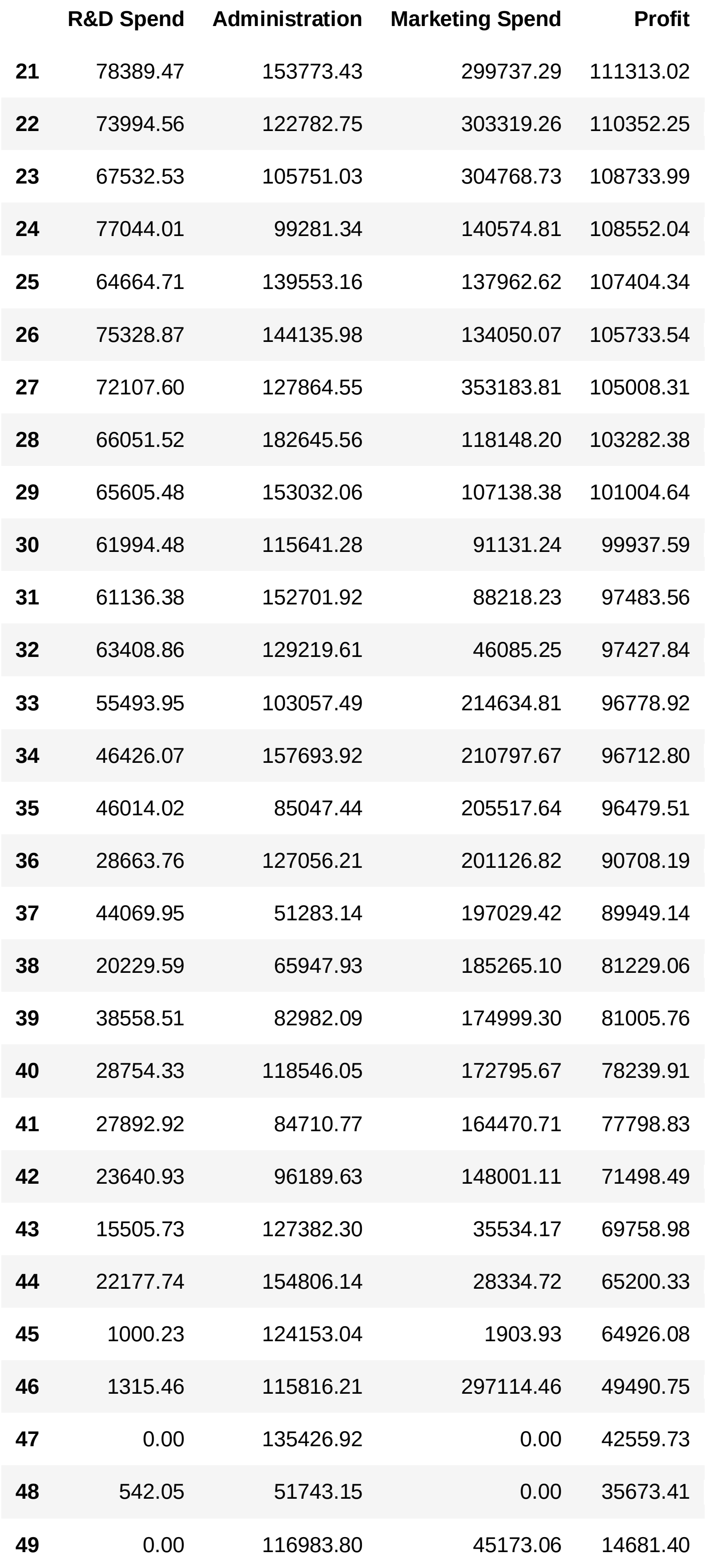


```
# importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

- Firstry I have to fetch the data from the computer with the help of this function) as **pd.read_csv( )** then we have to print our data



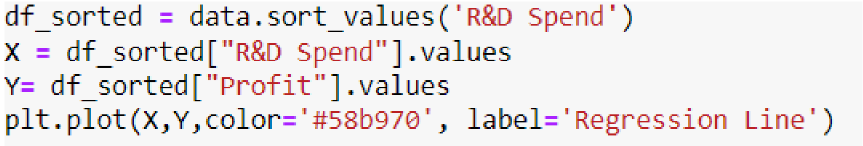
	R&D Spend	Administration	Marketing Spend	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94
5	131876.90	99814.71	362861.36	156991.12
6	134615.46	147198.87	127716.82	156122.51
7	130298.13	145530.06	323876.68	155752.60
8	120542.52	148718.95	311613.29	152211.77
9	123334.88	108679.17	304981.62	149759.96
10	101913.08	110594.11	229160.95	146121.95
11	100671.96	91790.61	249744.55	144259.40
12	93863.75	127320.38	249839.44	141585.52
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
19	86419.70	153514.11	0.00	122776.86
20	76253.86	113867.30	298664.47	118474.03



	R&D Spend	Administration	Marketing Spend	Profit
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49
43	15505.73	127382.30	35534.17	69758.98
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

4.2**Analysing the data**:- In this step we have to analyse our data that which features act as a dependent variable and which feature act as a dependent variable.and also see the relationship between dependent and independent variable.

- Here profit column is dependent variable and (R&D spend,administration ,marketing spend) act as a independent variable.
- Now lets see the graphs between dependent and independent variable

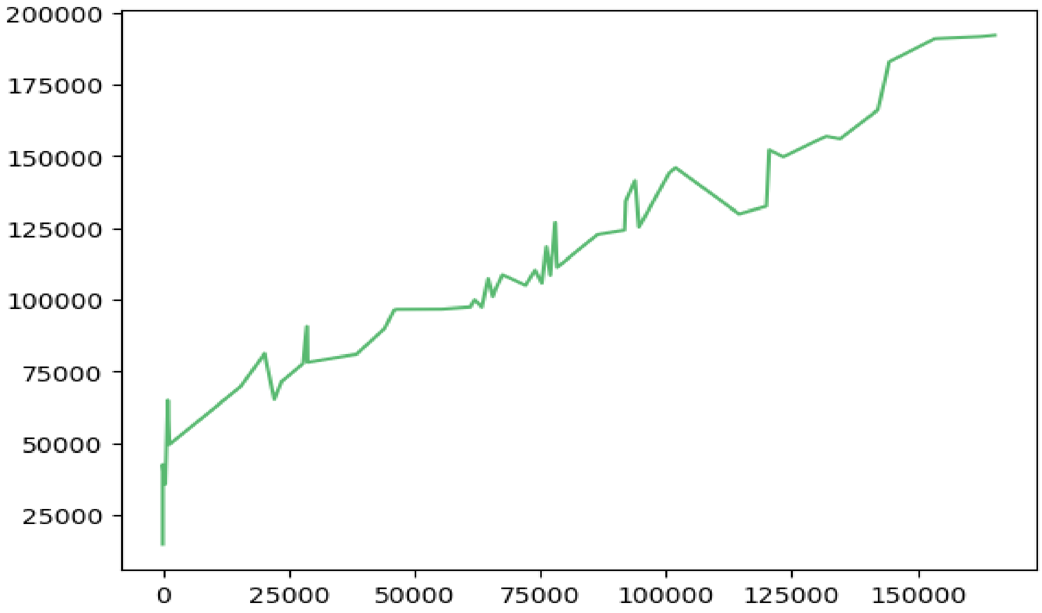


```
df_sorted = data.sort_values('R&D Spend')
X = df_sorted["R&D Spend"].values
Y= df_sorted["Profit"].values
plt.plot(X,Y,color='#58b970', label='Regression Line')
```

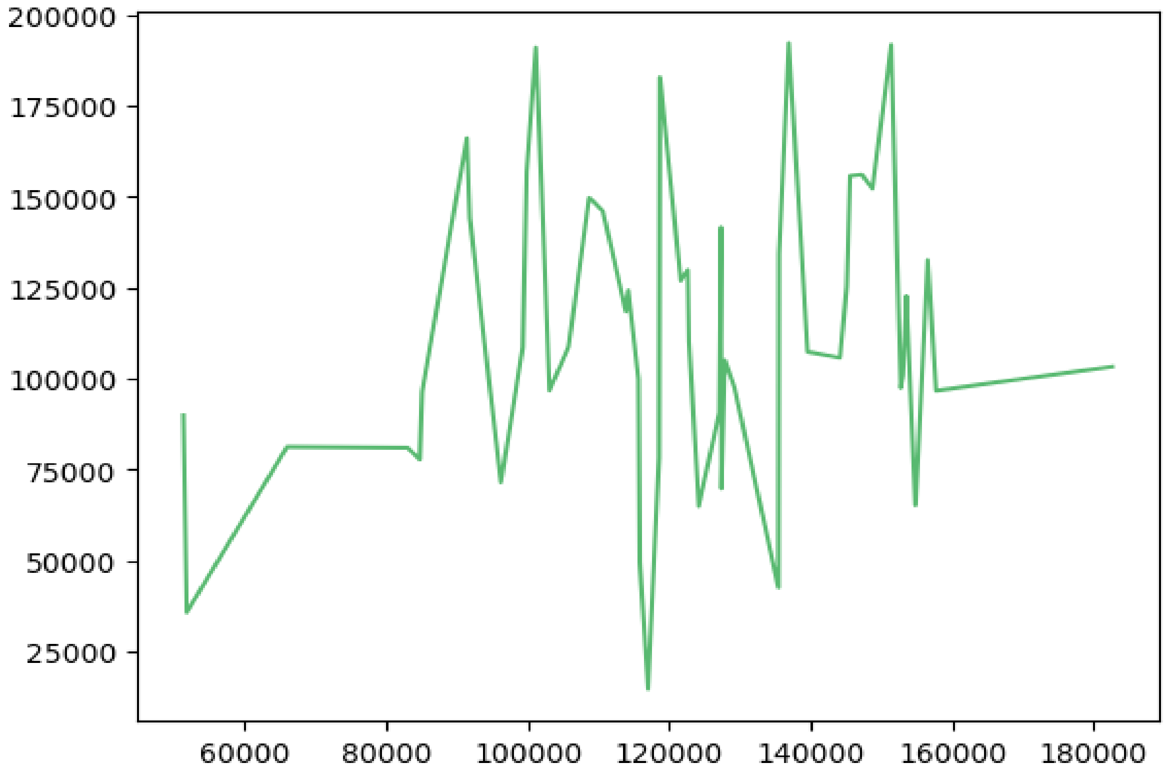
Here firstly to plot the graph we have to firstly sort the data that has been calibereated on the x axis henc for this we have to use the function **df_sorted = data.sort_values('R&D Spend')** which take column name as a argument.

And for plotting all these graphs we have to used the **plt.plot(X,Y,color='#58b970', label='Regression Line')** this function which take allthse values as argument and plot the graph between these two variables.

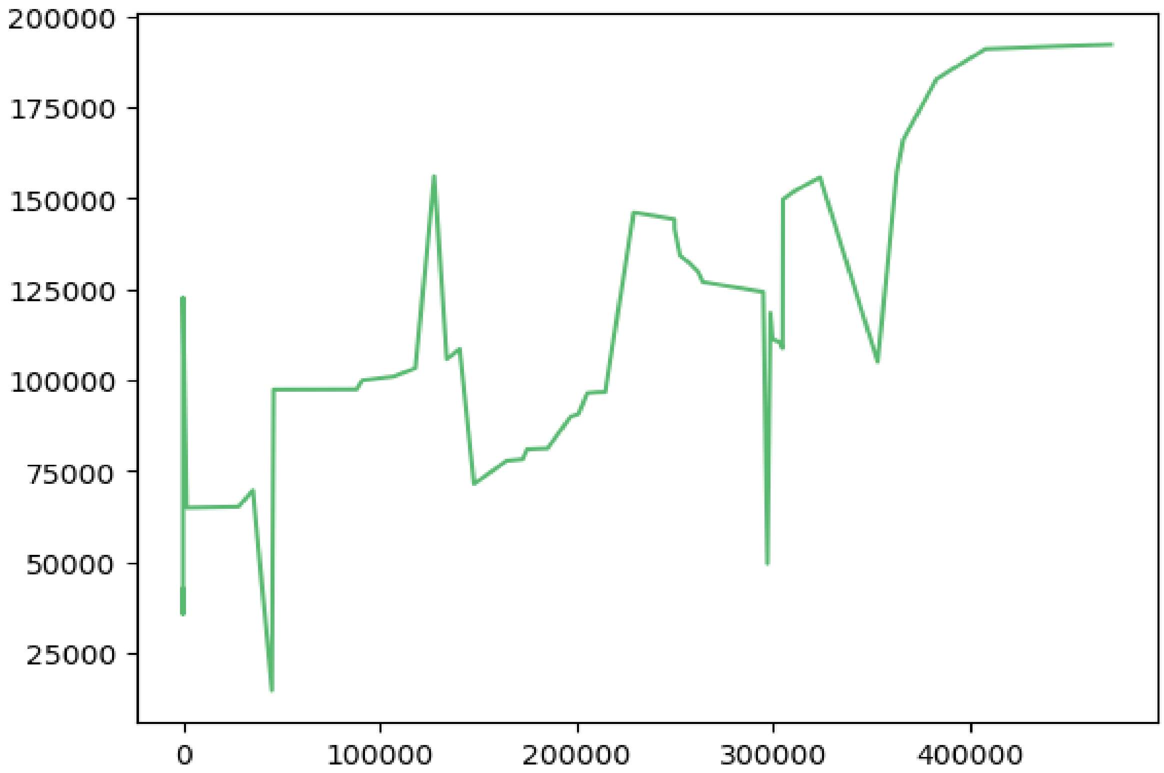
1.R&D Spend vs Profit



2.Administration vs Profit



3.MARKETING SPEND VS PROFIT

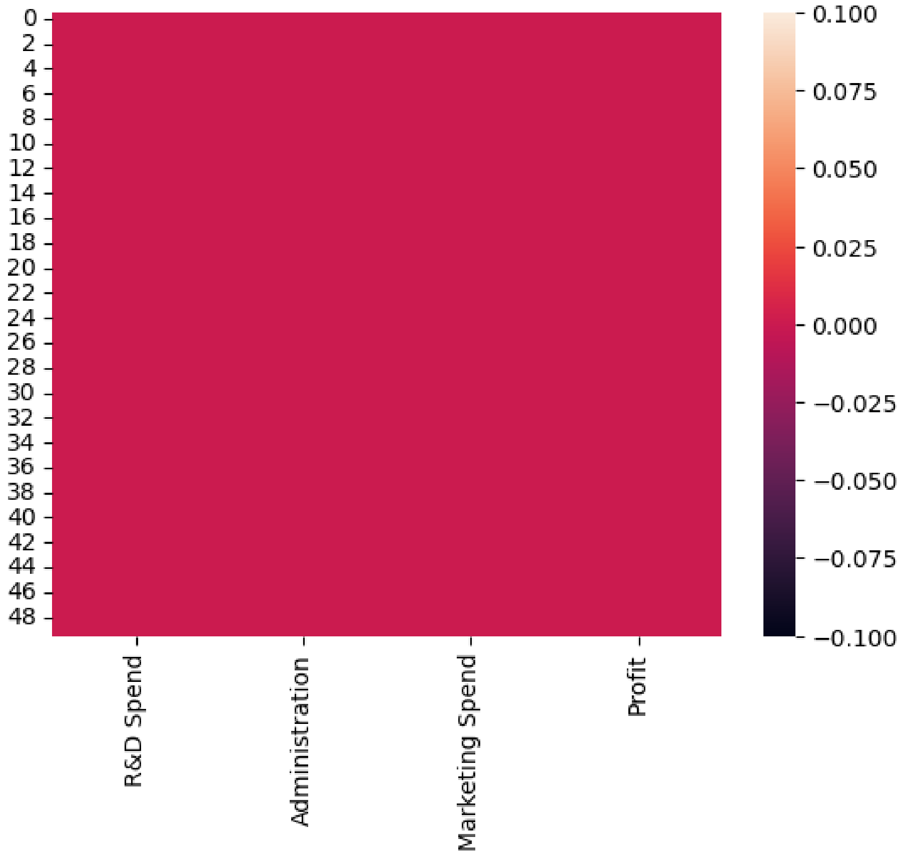


Conclusion from these graphs:-

- As profit change partial linearly with the R&D spend
- As profit depend partially with the Marketing Spend
- As profit is almost independent of the parameter Administration

4.3 **Data Preprocessing**: We removed any missing or irrelevant data from the dataset. We also performed data normalization to ensure that the data is in the same range.

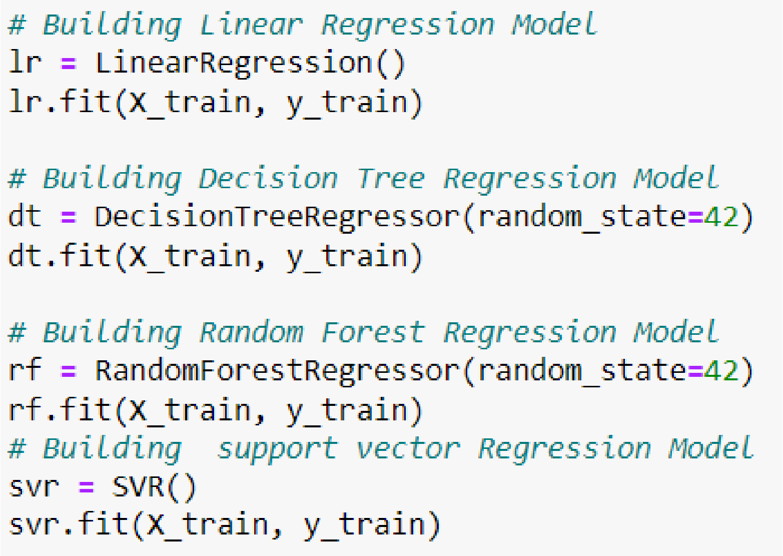
- Here we have to choose the **data.isnull()** to find any null values in the graph or not we have also plot a heat map with the use of this function **sns.heatmap(data.isnull())** and my output is



If there is any null values in the dta then it will be showm with the white mark.

4.4 **Algorithm Selection and Model Training**: We selected different regression algorithms such as linear regression, decision tree regressor , random forest regressor and support vector regressor.

- **x=data.drop("Profit",axis=1), y=data['Profit']** we have to used these function to stored the values in x and y in the form of data frame
- after that we have to split the the data into the testing and training dta set with the use of this function **X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=1)**

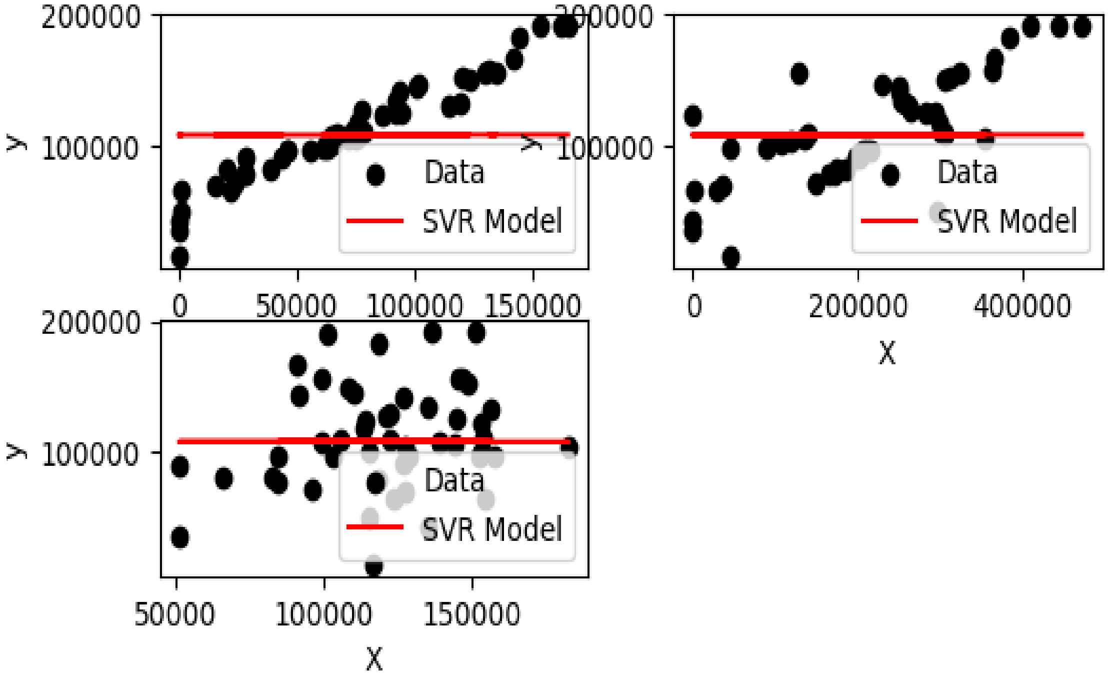


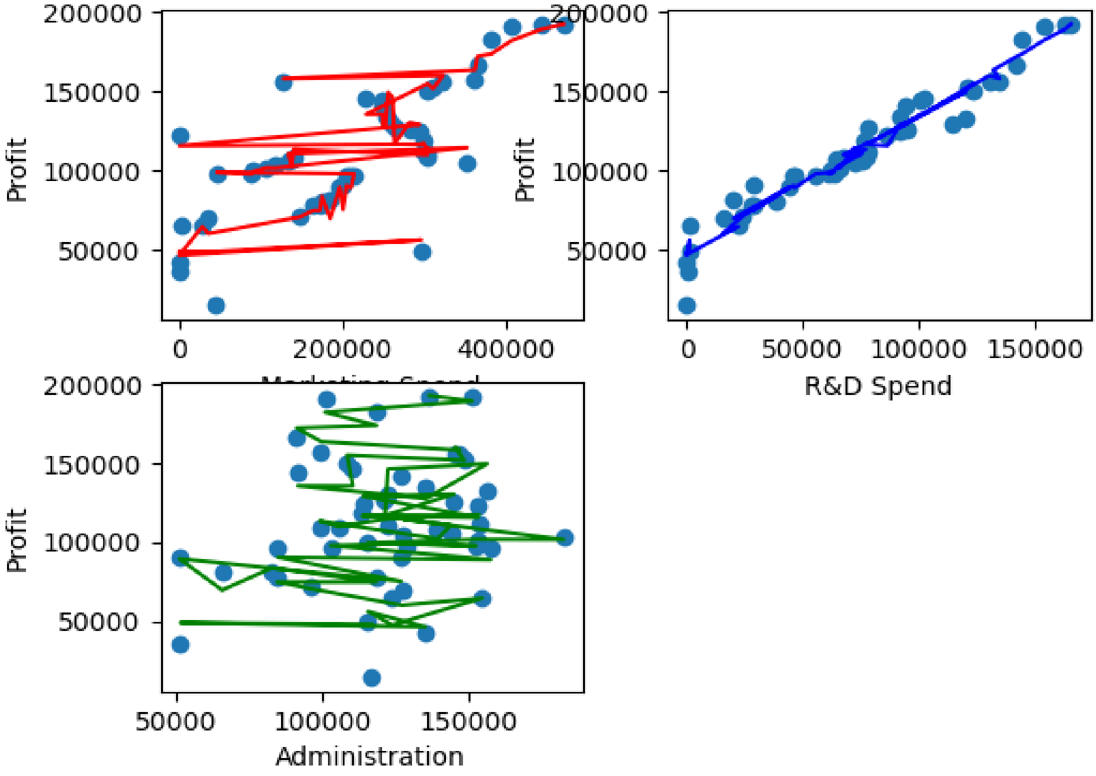
```
# Building Linear Regression Model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Building Decision Tree Regression Model
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)

# Building Random Forest Regression Model
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)
# Building  support vector Regression Model
svr = SVR()
svr.fit(X_train, y_train)
```

-
- The type of lr is a LinearRegression object. When the code lr = LinearRegression() is executed, an instance of the LinearRegression class is created and assigned to the variable lr. The LinearRegression() function in scikit-learn is used to create a linear regression model object. The object lr can then be used to train the model on the training data and make predictions on the test data.
- We have do similar thing with all the algorithms

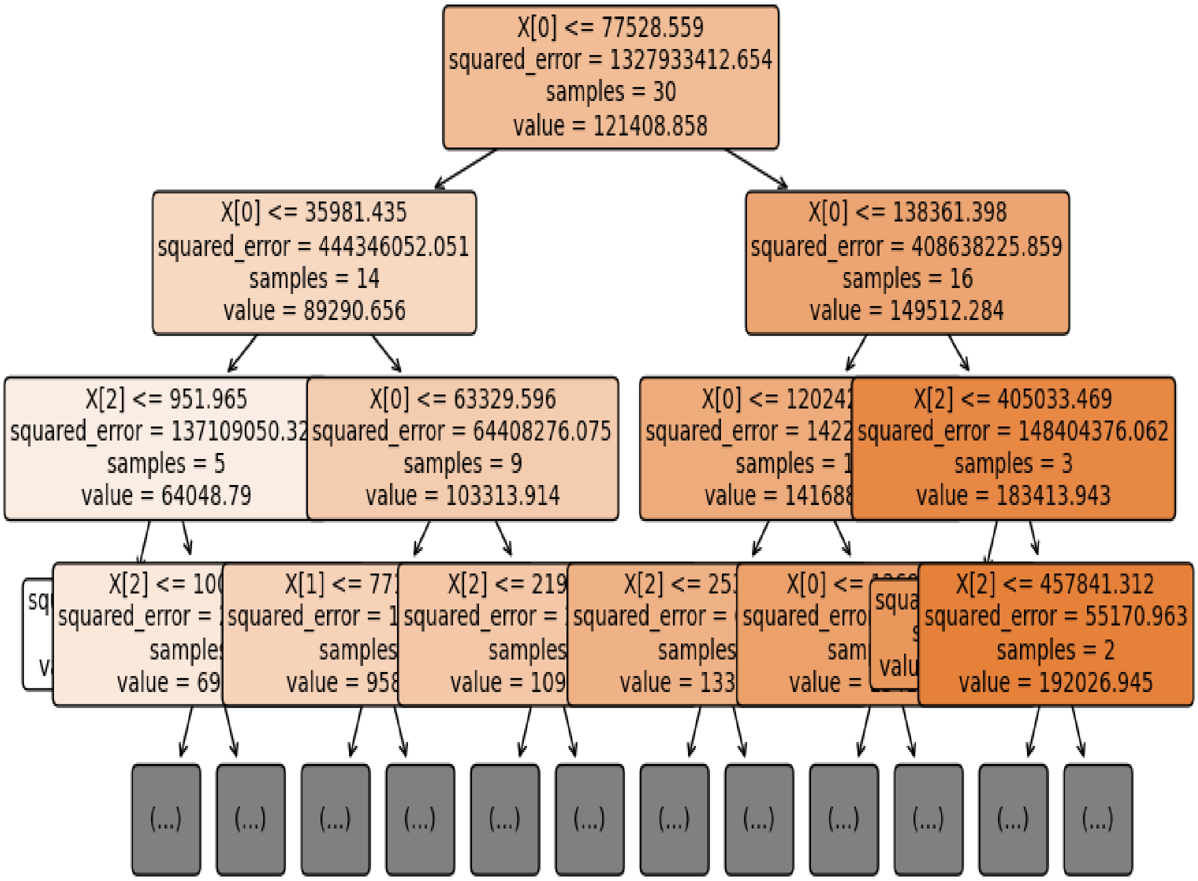
We have to see the plots of different algorithms:-

SUPPORT VECTROR REGRESSOR:-





LINEAR REGRESSOR PLOTS:-

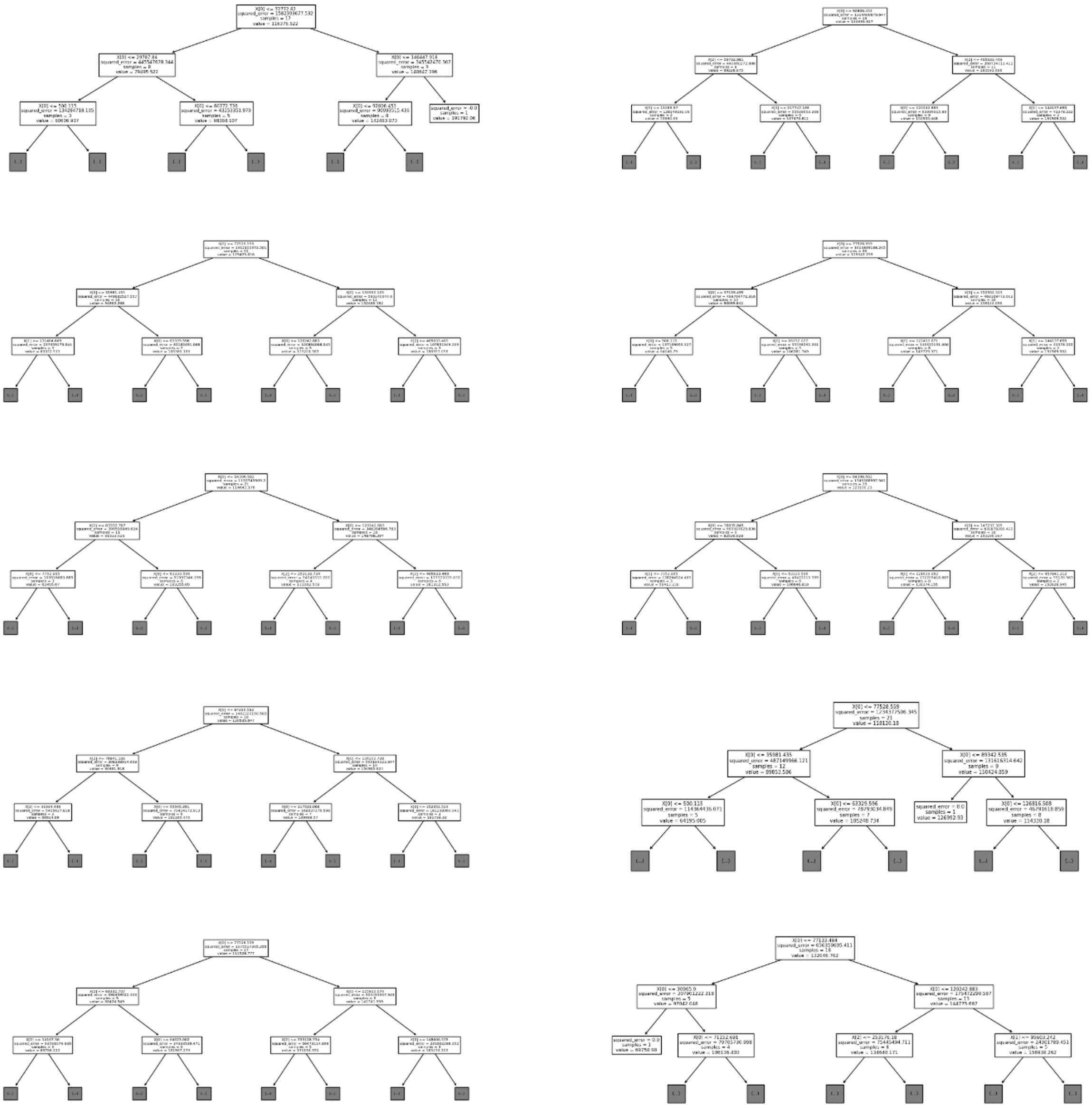


DECISION TREE PLOT:-

Now the last one is

RANDOM FOREST PLOT:-

In random forest youe can clearly see the many decision tress



4.5 **Model Evaluation**: We calculated different regression metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-Squared (R2) to evaluate the accuracy of the different models.

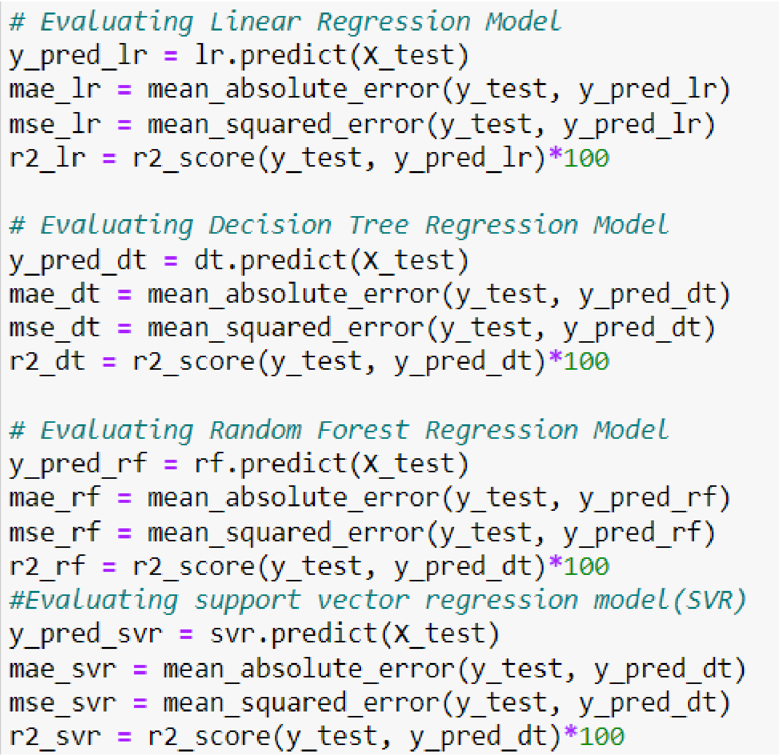
- Now in model evaluation we have to chech the r –squared ,mean squared error and root mean squared error of the all ythe models

- This will helpful to choose the best model

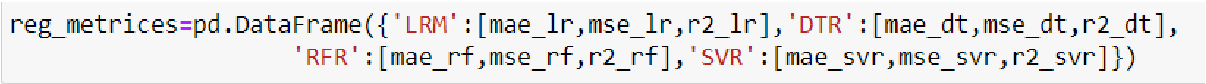
- Now firstly we have to predict the value **y_pred_lr** on giving the x_test values

- ➢ Then on the basis of the comparison between the **y_pred_lr** and **y_test**

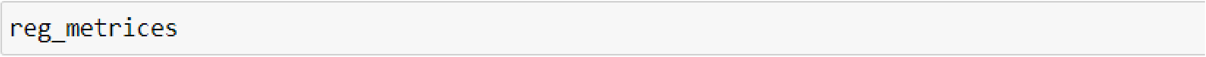
- ➢ We have to count the regression matrices
  With the help of following functions



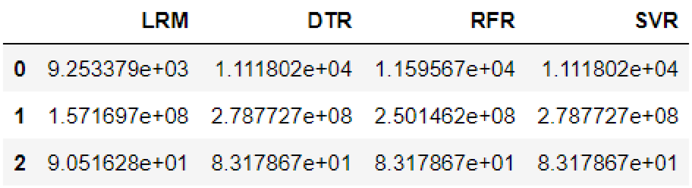
```
# Evaluating Linear Regression Model
y_pred_lr = lr.predict(X_test)
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)*100

# Evaluating Decision Tree Regression Model
y_pred_dt = dt.predict(X_test)
mae_dt = mean_absolute_error(y_test, y_pred_dt)
mse_dt = mean_squared_error(y_test, y_pred_dt)
r2_dt = r2_score(y_test, y_pred_dt)*100

# Evaluating Random Forest Regression Model
y_pred_rf = rf.predict(X_test)
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_dt)*100
#Evaluating support vector regression model(SVR)
y_pred_svr = svr.predict(X_test)
mae_svr = mean_absolute_error(y_test, y_pred_dt)
mse_svr = mean_squared_error(y_test, y_pred_dt)
r2_svr = r2_score(y_test, y_pred_dt)*100
```



```
reg_metrices=pd.DataFrame({'LRM':[mae_lr,mse_lr,r2_lr],'DTR':[mae_dt,mse_dt,r2_dt],
                          'RFR':[mae_rf,mse_rf,r2_rf],'SVR':[mae_svr,mse_svr,r2_svr]})
```

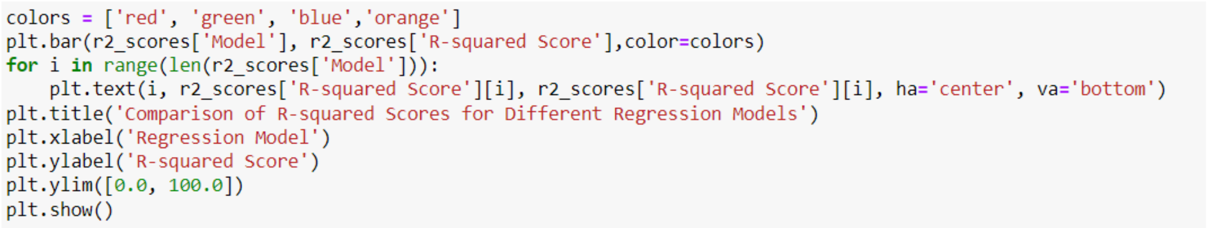


```
reg_metrices
```

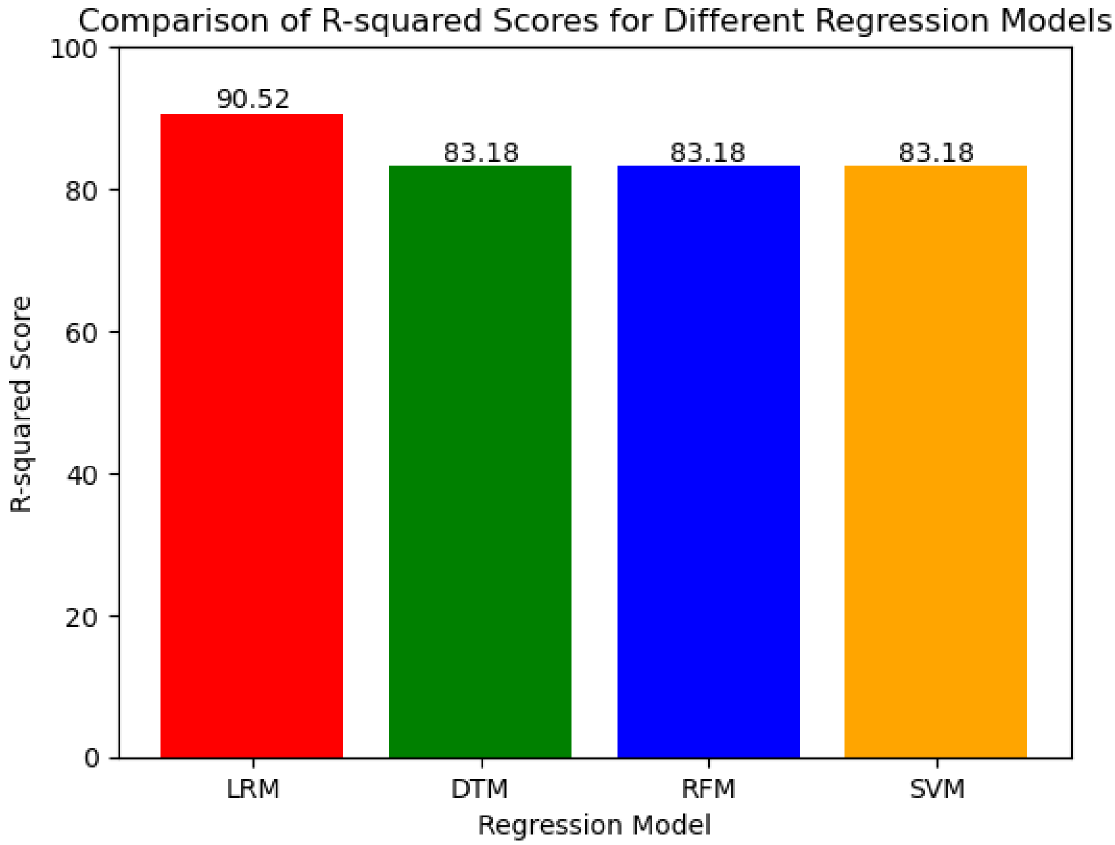


	LRM	DTR	RFR	SVR
0	9.253379e+03	1.111802e+04	1.159567e+04	1.111802e+04
1	1.571697e+08	2.787727e+08	2.501462e+08	2.787727e+08
2	9.051628e+01	8.317867e+01	8.317867e+01	8.317867e+01

4.6 **Model Selection**: Based on the regression metrics calculated, we selected the best model for the given dataset.



```
colors = ['red', 'green', 'blue','orange']
plt.bar(r2_scores['Model'], r2_scores['R-squared Score'],color=colors)
for i in range(len(r2_scores['Model'])):
    plt.text(i, r2_scores['R-squared Score'][i], r2_scores['R-squared Score'][i], ha='center', va='bottom')
plt.title('Comparison of R-squared Scores for Different Regression Models')
plt.xlabel('Regression Model')
plt.ylabel('R-squared Score')
plt.ylim([0.0, 100.0])
plt.show()
```



Now x axis represent the models and y axis represent the R-Squared scores

And our linear regression model have the heighest accuray according to our plot

5.IMPLEMENTATION:

We implement the methodology using Python as our programming language and various libraries such as pandas, scikit-learn, and matplotlib. We first load the dataset into a pandas dataframe and perform exploratory data analysis to understand the dataset's characteristics. We then split the dataset into train and test sets and construct different regression models. We evaluate the performance of each model using different regression metrics and choose the best model based on its performance on the test set.

SOURCE CODE:-

```
# Importing Libraries
import pandas as pd            #Version: 1.5.3
import numpy as np             #Version: 1.24.2
import matplotlib.pyplot as plt #Version: 3.7.1
import seaborn as sns               # Version: 0.12.2
from sklearn.model_selection import train_test_split #Version: 1.2.2
from sklearn.linear_model import LinearRegression #Version: 1.2.2
from sklearn.tree import DecisionTreeRegressor #Version: 1.2.2
from sklearn.ensemble import RandomForestRegressor #Version: 1.2.2
from sklearn.svm import SVR #Version: 1.2.2
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
#Version: 1.2.2

# Loading Dataset
data = pd.read_csv(r'C:\Users\Swapnil Jyot\Downloads\50_Startups.csv')
data
df_sorted = data.sort_values('R&D Spend')
df_sorted

len(data.index)

# # ANALYSING THE DATA....

plt.subplot(3, 2, 1)
data["Profit"].plot.hist()
plt.title('Histogram of Profit')

# #Hence we have to get maximum profit of near about 100000 for 10 times
acoording to the table
#
# lets see the how all three factors(R&D Spend,Administration,Marketing Spend)
affect the profit of the companies ...

# ### 1.R&D Spend vs Profit

df_sorted = data.sort_values('R&D Spend')
X = df_sorted["R&D Spend"].values
Y= df_sorted["Profit"].values
plt.subplot(3, 2, 4)
plt.plot(X,Y,color='#58b970', label='Regression Line')
plt.title('Line Graph of R&D Spend vs Profit')


# ### 2.Administration vs Profit

df_sorted = data.sort_values('Administration')
X = df_sorted["Administration"].values
```

```
Y= df_sorted["Profit"].values
plt.subplot(3, 2, 5)
plt.plot(X,Y,color='#58b970', label='Regression Line')
plt.title('Line Graph of Administration vs Profit')

# ### 3.Marketing Spend vs Profit

df_sorted = data.sort_values('Marketing Spend')
X = df_sorted["Marketing Spend"].values
Y= df_sorted["Profit"].values
plt.subplot(3, 2, 6)
plt.plot(X,Y,color='#58b970', label='Regression Line')
plt.title('Line Graph of Marketing Spend vs Profit')

# ### CONCLUSION

# #As profit change partial linearly with the R&D spend
# #As profit depend partially with the Marketing Spend
# #As profit is almost independent of the parameter Administration

# # DATA WRANGLING

# #finding the null values in the data before training and testing the data
data.isnull()

plt.subplot(3, 2, 2)
sns.heatmap(data.isnull())
plt.title('Heatmap of Correlation Matrix')

# ### Almost zero null values in the data now data is ready for testing and
training

# # Training And Testing the data

x=data.drop("Profit",axis=1)
y=data['Profit']

X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=
1)

print(X_train)

print(y_train)


# # BUILDING THE REGRESSION MODELS

# Building Linear Regression Model
lr = LinearRegression()
```

```
lr.fit(X_train, y_train)

# Building Decision Tree Regression Model
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)

# Building Random Forest Regression Model
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)

# Building  support vector Regression Model
svr = SVR()
svr.fit(X_train, y_train)

# Evaluating Linear Regression Model
y_pred_lr = lr.predict(X_test)
mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)*100

# Evaluating Decision Tree Regression Model
y_pred_dt = dt.predict(X_test)
mae_dt = mean_absolute_error(y_test, y_pred_dt)
mse_dt = mean_squared_error(y_test, y_pred_dt)
r2_dt = r2_score(y_test, y_pred_dt)*100

# Evaluating Random Forest Regression Model
y_pred_rf = rf.predict(X_test)
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_dt)*100
#Evaluating support vector regression model(SVR)
y_pred_svr = svr.predict(X_test)
mae_svr = mean_absolute_error(y_test, y_pred_dt)
mse_svr = mean_squared_error(y_test, y_pred_dt)
r2_svr = r2_score(y_test, y_pred_dt)*100

print(mae_lr,mse_lr,r2_lr)
print(mae_dt,mse_dt,r2_dt )
print(mae_rf,mse_rf,r2_rf)
print(mae_svr,mse_svr,r2_svr)

reg_metrices=pd.DataFrame({'LRM':[mae_lr,mse_lr,r2_lr],'DTR':[mae_dt,mse_dt,r2_dt],
                    'RFR':[mae_rf,mse_rf,r2_rf],'SVR':[mae_svr,mse_svr,r2_svr]
})

reg_metrices
```

```
r2_scores = pd.DataFrame({'Model': ['LRM', 'DTM', 'RFM','SVM'],
                          'R-squared Score': [r2_lr,r2_dt,r2_rf,r2_svr]})
r2_scores['R-squared Score'] = [round(score, 2) for score in r2_scores['R-
squared Score']]

plt.subplot(3, 2, 3)
colors = ['red', 'green', 'blue','orange']
plt.bar(r2_scores['Model'], r2_scores['R-squared Score'],color=colors)
for i in range(len(r2_scores['Model'])):
    plt.text(i, r2_scores['R-squared Score'][i], r2_scores['R-squared
Score'][i], ha='center', va='bottom')
plt.title('Comparison of R-squared Scores for Different Regression Models')
plt.xlabel('Regression Model')
plt.ylabel('R-squared Score')
plt.ylim([0.0, 100.0])
plt.tight_layout()
plt.show()

# ### hence you can clearly see that the heighest accuracy levels of different
regression models
```

6.CONCLUSION:

In this project, we constructed different regression models to predict the profit value of a company based on its R&D spend, administration cost, and marketing spend. We evaluated the performance of each model using different regression metrics and chose the best model based on its performance on the test set. Our results suggest that the linear regression regression algorithm outperforms other regression algorithms in predicting the profit value of a company. Our proposed method can be used in various applications such as business forecasting, investment analysis, and financial planning.

References:

1.Scikit-learn library: https://scikit-learn.org/stable/index.html

2.Pandas library: https://pandas.pydata.org/

3.Matplotlib library: https://matplotlib.org/

4. https://www.analyticsvidhya.com/blog/

5. https://www.kaggle.com/