**EXPOSYS DATA LABS – INTERNSHIP**

**Name :** Gurrala Madhu Mohan Vamsi

**Domain of Project :** Data Science

**Duration of Project :** 1Month

**Instamoji Payment ID :** MOJO3429C05D77301717

**ABSTRACT**

The association among x, the independent variable, and y, the dependent variable treated as an nth degree polynomial in polynomial regression, a type of linear regression. A nonlinear association between the value of x and the corresponding conditional mean of y, represented by the representation of E(y |x), is fit through polynomial regression.

Some correlations may be curvilinear, according to a researcher's hypothesis. Such scenarios will undoubtedly have a polynomial term.

A scatter plot of residuals (Y-axis) on the predictor (X-axis) will show patches of many positive residuals in the middle if we attempt to fit a linear model to curved data. As a result, it is inappropriate in this circumstance.

Common multiple linear regression analysis makes the premise that each independent variable is independent of the others. This presumption is not true for polynomial regression models.

A polynomial is a mathematical statement composed of variables and coefficients and uses only the operations of that include addition, subtraction, multiplication, and a positive integer exponents of the variables. The term "polynomial" simply means "many terms" in mathematics.

While linear functions adhere to the mathematical criteria of a polynomial, we might think of them as two distinct kinds of regression analysis in the context of machine learning.

Actually, a subset of linear regression is known as polynomial regression. Despite fitting a nonlinear model to the data, polynomial regression is a linear statistical estimation problem because the regression function E(y|x) is linear in the unknown parameters that are inferred from the data. As a result, Polynomial Regression is thought of as a specific case of multiple linear regression.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **List of Figures** | **Page No** |
|  | Introduction | 4 – 13 |
|  | Existing Method | 14 – 24 |
|  | Proposed method with architecture | 25 – 32 |
|  | Methodology | 33 – 39 |
|  | Implementation | 40 – 43 |
|  | Conclusion | 44 – 47 |

**CHAPTER – 1**

**INTRODUCTION**

**Introduction**

**Machine learning algorithms**

Programmes utilising machine learning algorithms are able to identify hidden patterns in data, forecast results, and enhance performance based on past performance. In machine learning, several algorithms can be employed for various tasks, such as basic linear regression for prediction issues like stock market forecasting and the KNN algorithm for categorization issues.

Let's look more closely at the various machine learning algorithms.

**Fig: Types of machine learning algorithms**

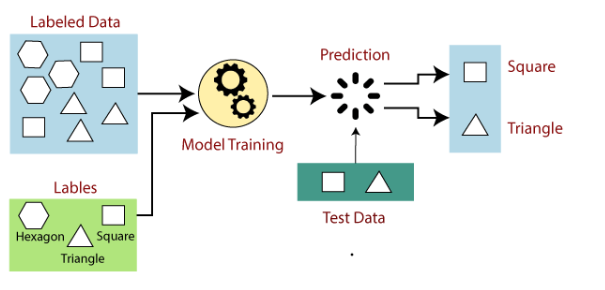
1. **Supervised machine learning**

Supervised learning is a sort of machine learning in which the output is predicted by the machines using well-labelled training data that has been used to train the machines. The term "labelled data" refers to input data which has already been allocated an appropriate outcome.

In supervised learning, the training data that is given to the computers serves as the supervisor, instructing them on how to correctly predict the output. It employs the same idea that a pupil would learn under a teacher's guidance.

The method of supervised learning involves giving the machine learning model the right input data as well as the output data. Finding a mapping function to link the input variable (x) with the output variable (y) is the goal of a supervised learning algorithm.

Supervised learning has applications in the real world such as risk assessment, image categorization, fraud detection, spam filtering, etc.



**Fig: Working of supervised machine learning**

Models are trained using labelled datasets in supervised learning, where the model learns about various types of input. Following the completion of the training phase, the model is evaluated using test data (a subset of the training set), and it then makes output predictions.

Let's say we have a dataset of various forms, such as squares, rectangles, triangles, and polygons. The model must now be trained for each shape, which is the first stage.

* If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square.
* If the given shape has three sides, then it will be labelled as a triangle.
* If the given shape has six equal sides then it will be labelled as hexagon.

Now, after training, we test our model using the test set, and the task of the model is to identify the shape.

The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output.

1. **Classification**

On the basis of training data, new observations are organised using the Classification algorithm, a Supervised Learning technique. In classification, a programme makes use of the dataset or observations that are provided to learn how to categorise fresh observations into various classes or groupings. For instance, cat or dog, yes or no, 0 or 1, spam or not spam, etc. Targets, labels, or categories can all be used to describe classes.

In contrast to regression, classification's output variable is a category rather than a value, such as "Green or Blue," "fruit or animal," etc. The Classification algorithm uses labelled input data because it is a supervised learning technique, therefore it comprises input and output information.

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

y=f(x),

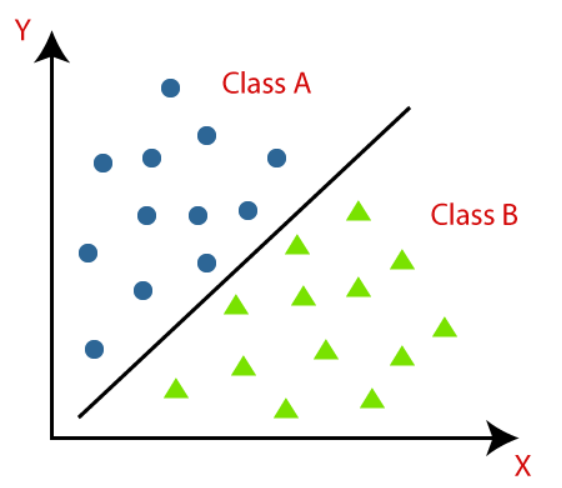
where,

y = categorical output

The best example of an ML classification algorithm is Email Spam Detector.

The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data.

Classification algorithms can be better understood using the below diagram. In the below diagram, there are two classes, class A and class B. These classes have features that are similar each other and dissimilar to other classes to.



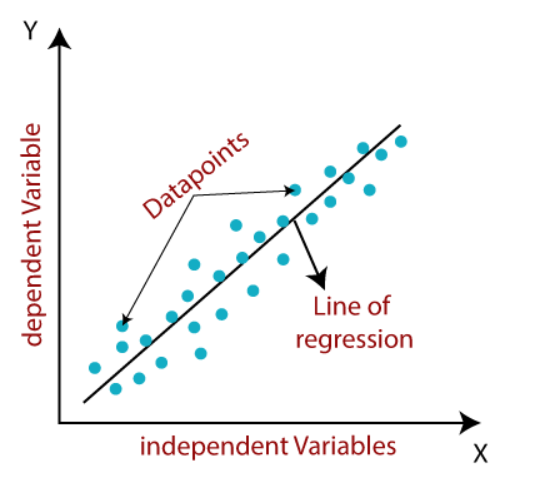
**Fig: Classification graph**

1. **Regression**

Regression analysis uses one or more independent variables to describe the relationship between a dependent (target) and independent (predictor) variables. More specifically, regression analysis enables us to comprehend how, while other independent variables are held constant, the value of the dependent variable changes in relation to an independent variable. It forecasts real, continuous values like temperature, age, salary, and cost, among others.

A continuous variable can be predicted with the aid of regression analysis. In the real world, there are many situations where we need to make predictions about the future, including those involving the weather, sales, marketing trends, and other factors. In these situations, we need technology that can make forecasts more precisely. Regression analysis, a statistical technique utilised in machine learning and data science, is therefore necessary in this situation. Additional justifications for adopting regression analysis are listed below:

* Regression estimates the relationship between the target and the independent variable.
* It is used to find the trends in data.
* It helps to predict real/continuous values.
* By performing the regression, we can confidently determine the most important factor, the least important factor, and how each factor is affecting the other factors.



**Fig: Regression graph**

1. **Simple Linear Regression**

One of the simplest and most widely used Machine Learning techniques is linear regression. It is a statistical technique for performing predictive analysis. For continuous/real/numeric variables like sales, salary, age, and product price, among others, linear regression makes predictions.

The linear regression algorithm, often known as linear regression, demonstrates a linear relationship between a dependent (y) and one or more independent (y) variables. Given that linear regression demonstrates a linear relationship, it may be used to determine how the dependent variable's value changes as a function of the independent variable's value.

1. **Multiple Linear regression**

By fitting a linear equation to the observed data, multiple linear regression makes an attempt to describe the link between two or more features and a response. The procedures for performing multiple linear regression resemble those for performing simple linear regression almost exactly. The appraisal makes a difference. It can be used to determine which factor has the most influence on the expected result and how several factors interact with one another.

1. **Polynomial Regression**

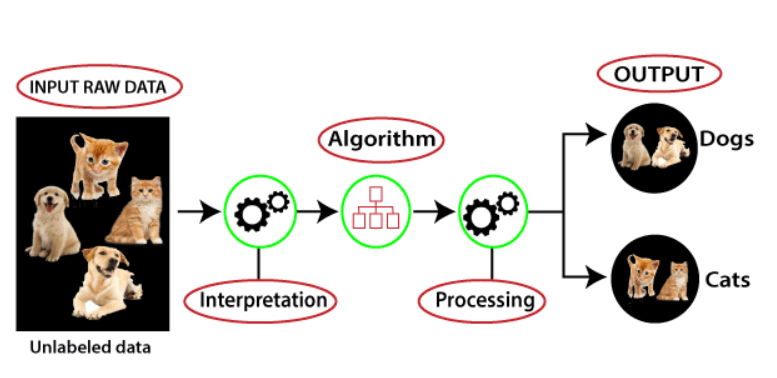
The sole reason we add some polynomial terms to linear regression to make it into polynomial regression is because the relationship between the dependent and independent variables is not linear.

The link between the dependent and independent variables is modelled as an nth-degree polynomial function in polynomial regression. A quadratic model is used when the degree of the polynomial is 2, a cubic model is used when the degree is 3, and so on.

1. **Unsupervised machine learning**

Unsupervised learning is a type of machine learning in which models are not supervised using training datasets, as the name implies. Instead, models themselves decipher the provided data to reveal hidden patterns and insights. It is comparable to the learning process that occurs in the human brain while learning something new. It is characterised as:

Unsupervised learning is a subcategory of machine learning in which models are trained using unlabeled datasets and are free to operate on the data without being checked by a human observer.



**Fig: Working of unsupervised machine learning**

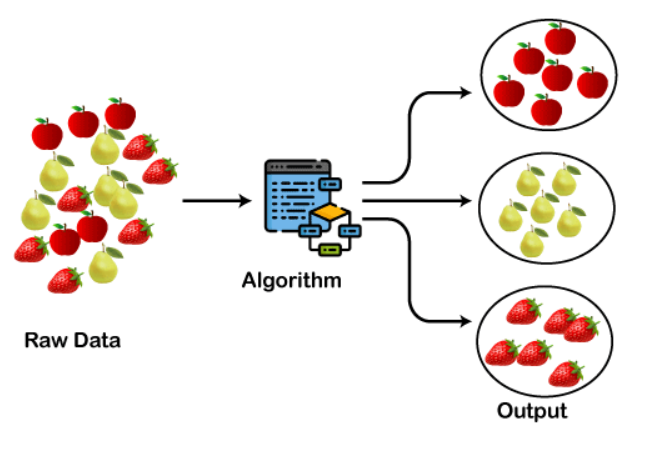
Because unlike supervised learning, we have the input data but no corresponding output data, unsupervised learning cannot be used to solve a regression or classification problem directly. Finding the underlying structure of a dataset, classifying the data into groups based on similarities, and representing the dataset in a compressed format are the objectives of unsupervised learning.

In this case, we have used unlabeled input data, which means that neither its category nor any associated outputs are provided. Now, the machine learning model is being trained using the unlabeled input data. It will first analyse the raw data to identify any hidden patterns in the data before using the appropriate algorithms, such as k-means clustering, decision trees, etc. Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

1. **Clustering**

A machine learning approach called clustering or cluster analysis groups the unlabeled dataset. It is described as "A method of clustering the data points into different groups, each grouping similar data points." The items with potential resemblances continue to be in a group that shares little to no characteristics with another group.

It accomplishes this by identifying comparable patterns in the unlabeled dataset, such as shape, size, colour, behaviour, etc., then classifying the data according to the presence or absence of these patterns.



**Fig: Working of clustering**

It uses an unsupervised learning approach, which means the algorithm receives no supervision, and it works with an unlabeled dataset.

Each cluster or group is given a cluster-ID after using this clustering technique, which ML systems can employ to streamline the processing of big and complicated datasets.

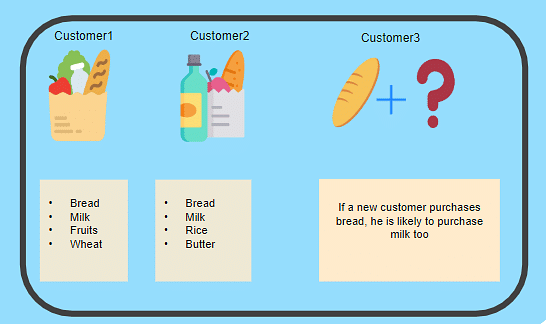
The clustering method is frequently employed for analysing statistical data.

1. **Association**

Across a huge collection of things or activities, association rule algorithms count the frequency of complementary occurrences, or associations. Finding relationships that occur together far more frequently than you would in a random sample of possibilities is the aim. A quick and effective method for mining categorised, non-numeric databases is the rule-based technique.

Finding significant relationships between variables or features in a data collection is done using the rule-based machine learning and data mining technique known as association learning. By using some metric of interestingness to construct an association rule for new searches, association rule learning, in contrast to standard association algorithms evaluating degrees of similarity, identifies hidden relationships in databases.

.



**Fig: Example of association**

Across a huge collection of things or activities, association rule algorithms count the frequency of complementary occurrences, or associations. Finding relationships that occur together far more frequently than you would in a random sample of possibilities is the aim. A quick and effective method for mining categorised, non-numeric databases is the rule-based technique.

**CHAPTER – 2**

**EXISTING METHOD**

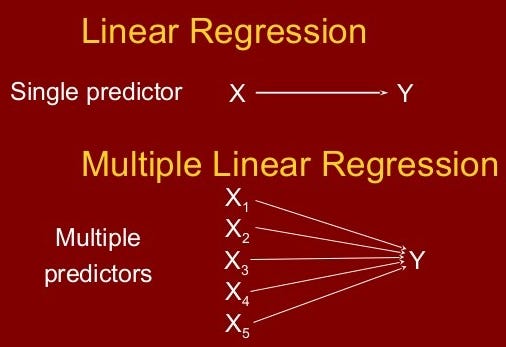
**EXISTING METHOD**

**Linear Regression**

The statistical method used to determine the relationship between variables is regression. As a result, the Linear Regression assumes that variables have a linear relationship. Depending on how many input variables there are, the regression problem is divided into

1) Simple Linear Regression (SLR)

2) Multiple Linear Regression (MLR)



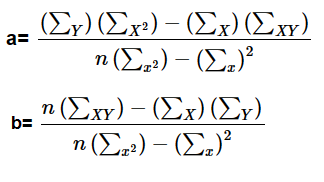
**Fig: Difference between SLR and MLR**

By applying a linear equation to the observed data, linear regression can be used to determine the relationship between two variables. There are two different kinds of variables; the first is referred to as an independent variable, and the second is a dependent variable. Predictive analysis often uses linear regression.

As we know, linear regression shows the linear relationship between two variables. The equation of linear regression is similar to that of the slope formula. Linear Regression formula is given by the equation

Y= a + bX

We will find the value of a and b by using the below formula



Let us try to understand with an solved example.

**Example -**

**Problem:**

Find a linear regression equation for the following two sets of data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | 2 | 4 | 6 | 8 |
| **Y** | 3 | 7 | 5 | 10 |

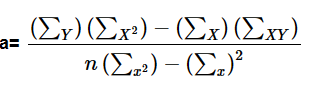
**Solution:**

To find the linear regression equation we need to find the value ΣX, ΣY, ΣX2, ΣXY

Construct the table and find the values.

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **X2** | **XY** |
| 2 | 3 | 4 | 6 |
| 4 | 7 | 16 | 28 |
| 6 | 5 | 36 | 30 |
| 8 | 10 | 64 | 80 |
| **ΣX = 20** | **ΣY = 25** | **ΣX2 = 120** | **ΣXY = 144** |

Now,

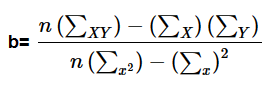


a =

a =

a = 1.5

and



b =

b =

b = 0.95

we got the values a=1.5 and b=0.95

put a, b values in linear equation. Then we get

y = 1.5 + 0.95x which is a required linear equation

1. **Simple linear regression**

A form of regression method called simple linear regression simulates the relationship between a single independent variable and a dependent variable. A Simple Linear Regression model displays a linear or sloping straight line relationship, therefore the name of the model.

The dependant variable for simple linear regression must have a continuous or real value. The independent variable, however, can be quantified using either continuous or categorical values.

Simple Linear regression algorithm has mainly two objectives:

* Model how the two variables are related. For instance, the connection between income and spending, experience and pay, etc.
* Forecasting new observations. For instance temperature-based weather forecasting, Revenue of a corporation based on annual investments, etc.

y= b0+b1x

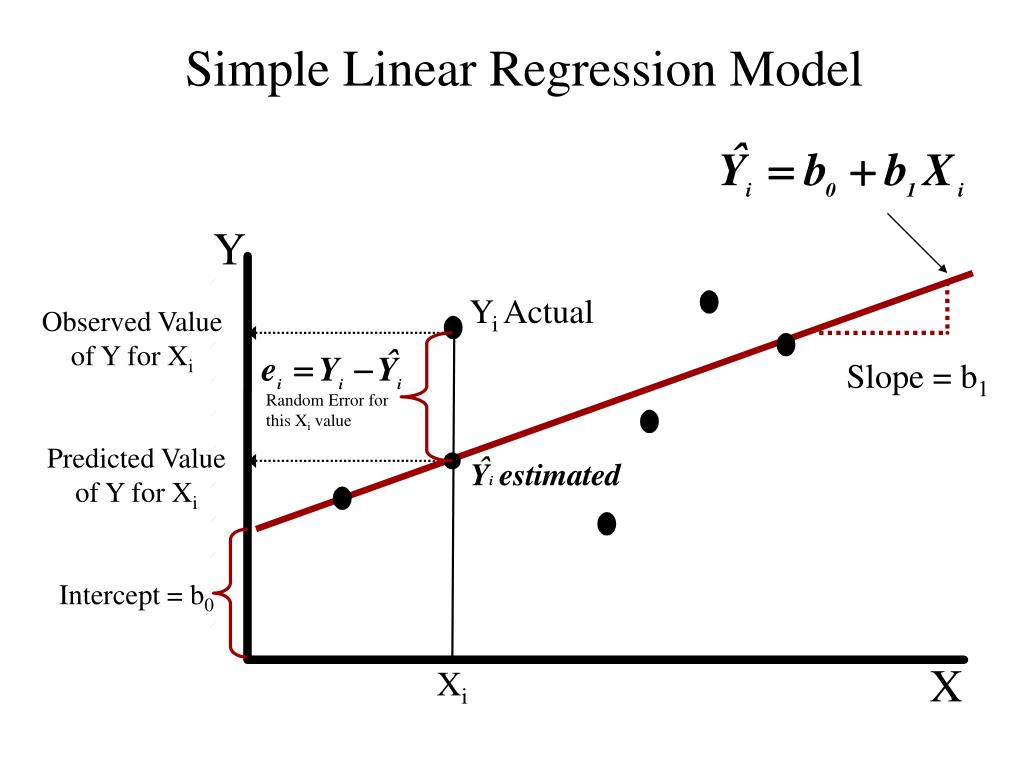
where,

b0 = intercept

b1 = coefficient or slope

x = independent variable

y = dependent variable



**Fig: Simple Linear Regression graph**

The goals of this problem is –

* We want to find out if there is any correlation between these two variables
* We will find the best fit line for the dataset.
* How the dependent variable is changing by changing the independent variable.

Now let us implement simple linear regression in python.

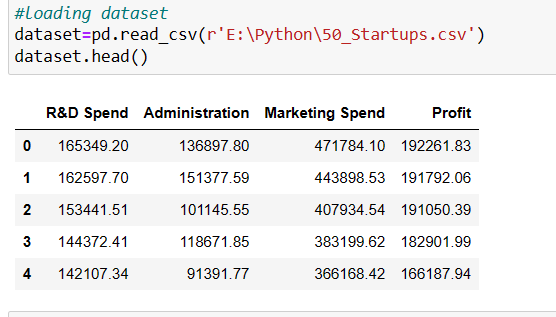
**Step – 1:**

Import the libraries.



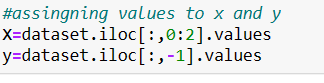
**Step – 2:**

Load the dataset.



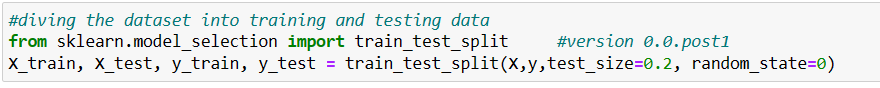
**Step – 3:**

Divide the dataset into X and y. X will contain the Column between 0 and 2 i.e., R&D Spend, Administration, Marketing Spend. And these are all independent variables. y will contain the last column. i.e., Profit which is a dependent variable.



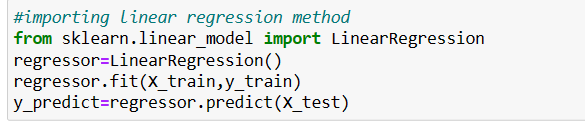
**Step – 4:**

Divide the data set into training and testing sets such that training set is 40 columns (80%) and testing set is 10 columns (20%). So the test size is 0.2



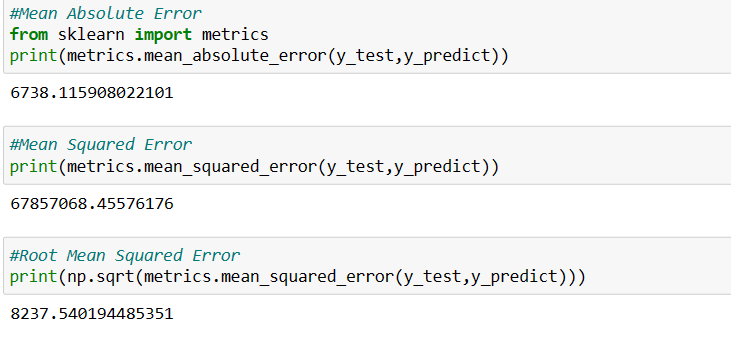
**Step – 5:**

Import linear regression method.



**Step – 6:**

Apply regression metrics like Mean Absolute Error, Mean Squared Error, Root Mean Squared Error.



1. **Multiple Linear regression**

A single Independent/Predictor(X) variable is utilised to model the response variable (Y), as we have taught about Simple Linear Regression. The Multiple Linear Regression (MLR) approach, however, is utilised in circumstances where more than one predictor variable has an impact on the response variable.

Moreover, Multiple Linear Regression is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable. We can define it as:

y = b0 + b1x1 + b2x2 + ..... + bnxn

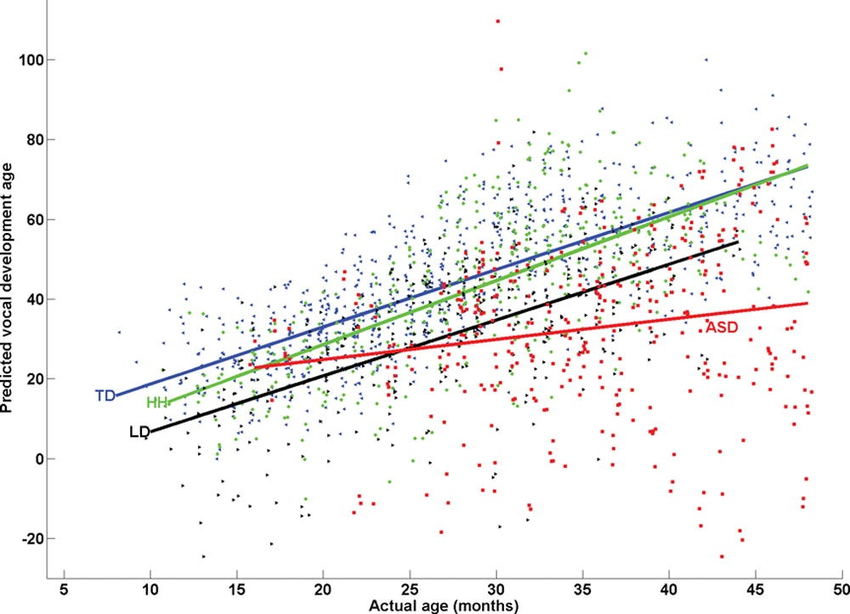
where,

b0 = intercept

b1,b2,b3,b4....bn = coefficient or slope

x1,x2,x3,x4,....xn = independent variable

y = dependent variable



**Fig: Multiple Linear Regression graph**

* For MLR, the dependent or target variable(Y) must be the continuous/real, but the predictor or independent variable may be of continuous or categorical form.
* Each feature variable must model the linear relationship with the dependent variable.
* MLR tries to fit a regression line through a multidimensional space of data-points.

Now let us implement multiple regression in python.

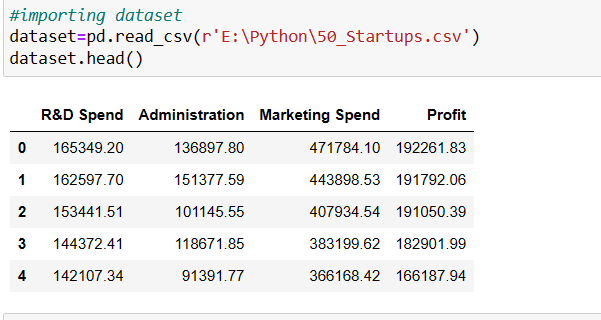
**Step – 1:**

Import the libraries.



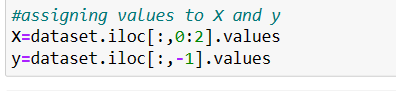
**Step – 2:**

Load the dataset.



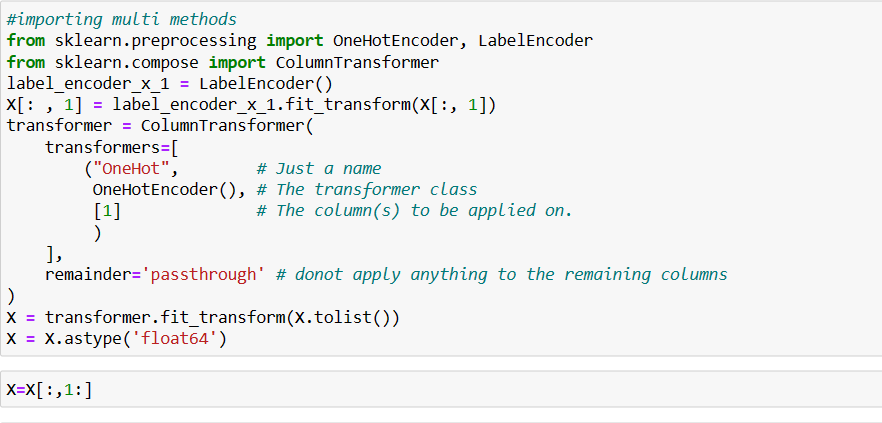
**Step – 3:**

Divide the dataset into X and y. X will contain the Column between 0 and 2 i.e., R&D Spend, Administration, Marketing Spend. And these are all independent variables. y will contain the last column. i.e., Profit which is a dependent variable.



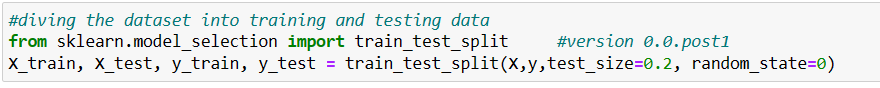
**Step – 4:**

Import multi methods i.e., OneHotEncoder, LabelEncoder



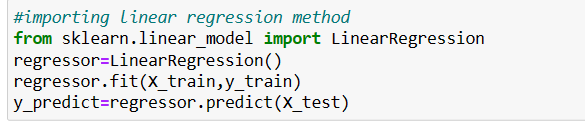
**Step – 5:**

Divide the data set into training and testing sets such that training set is 40 columns (80%) and testing set is 10 columns (20%). So the test size is 0.2



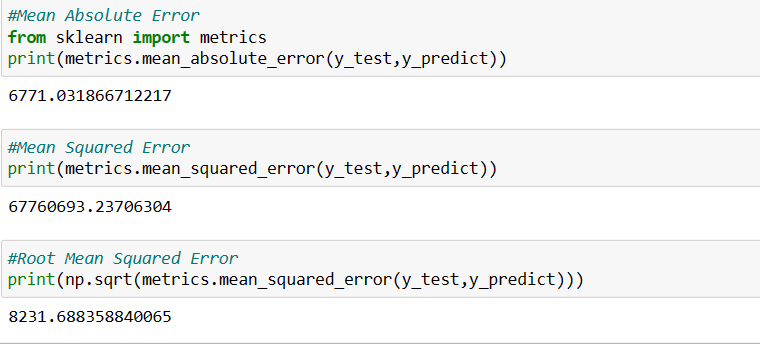
**Step – 5:**

Import linear regression method.



**Step – 6:**

Apply regression metrics like Mean Absolute Error, Mean Squared Error, Root Mean Squared Error.



**CHAPTER – 3**

**PROPOSED METHOD WITHE ARCHITECTURE**

**PROPOSED METHOD WITH ARCHITECTURE**

When we have a dataset with one predictor variable and one response variable, we often use simple linear regression to quantify the relationship between the two variables.

However, simple linear regression (SLR) assumes that the relationship between the predictor and response variable is linear. Written in mathematical notation, SLR assumes that the relationship takes the form:

**y= b0+b1x**

But in practice the relationship between the two variables can actually be nonlinear and attempting to use linear regression can result in a poorly fit model.

Regression analysis is a c onceptually simple method for investigating functional relationships between variables. The relationship is expressed in the form of an equation or a model connecting the response or dependent variable and one or more explanatory or predictor variables.

We denote the response variable by y, and the set of predictor variables by x1,x2,x3,...xn, where n denotes the number of predictor variables.

The general regression model is specified as:

**y = f(x1,x2,x3,...xn)**

One way to account for a nonlinear relationship between the predictor and response variable is to use polynomial regression, which takes the form:

**y = b0 + b1x + b2x2 + … + bnxn**

There are six steps to represent the architecture of regression analysis. They are namely -

**Step – 1:**

**Statement of the problem**

Formulation of the problem includes determining the questions to be addressed.

**Step – 2:**

**Selection of Potentially Relevant Variables**

We select a set of variables that are thought by the experts in the area of study to explain or predict the response variable.

**Step – 3:**

**Model specification**

The form of the model that is thought to relate the response variable to the set of predictor variables can be specified initially by experts in the area of study based on their knowledge or their objective and/or subjective judgments.

**Step – 4:**

**Method of Fitting**

We want to perform parameter estimation or model fitting after defining the model and collecting the data. the most commonly used method of estimation is called the least squares method. Other estimation methods we consider are the maximum likelihood method, ridge regression and the principal components method.

**Step – 5:**

**Model Fitting**

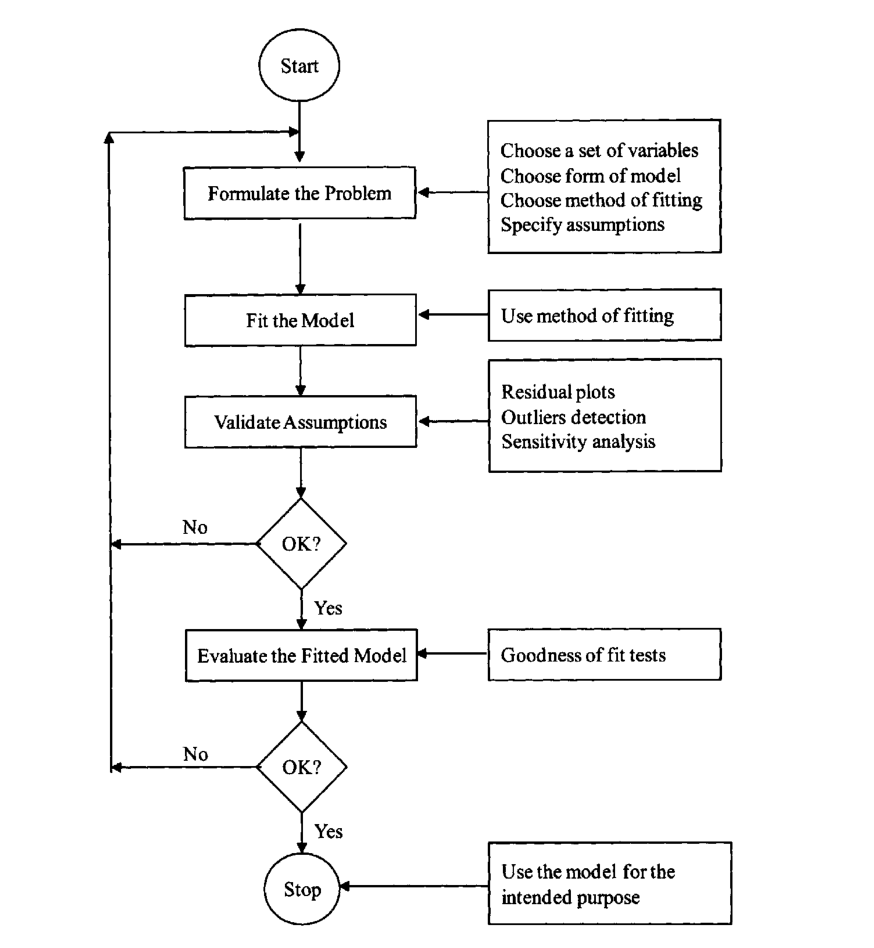
The estimates of the regression parameters b0,b1,b2,…bn are denoted by ~~b~~~~0~~~~,b~~~~1~~~~,b~~~~2~~,…~~b~~~~n~~ . The obtained ~~y~~ denotes the predicted value.

**Step – 6:**

**Model Criticism and Selection**

The validity of statistical methods depend on certain assumptions, about the data and the model.

Now let us see it’s architecture in detail.



**Fig: Regression Analysis Architecture**

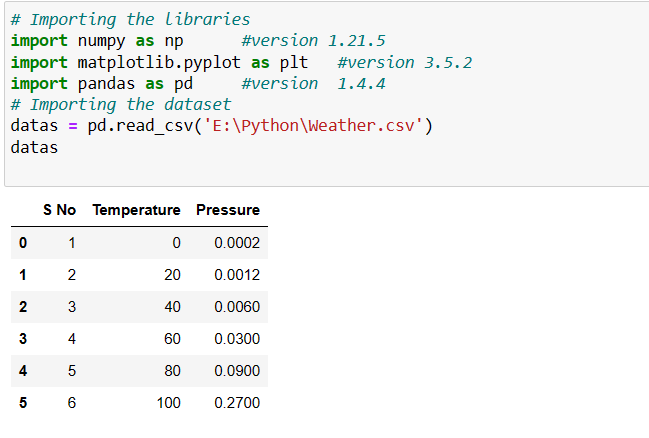
Let us try to understand polynomial regression architecture with an example.

Here there are 6 steps to implement polynomial regression graph. Along with that linear regression graph is also implemented to make comparison.

Now let us create a graph for polynomial regression with an example in python.

**Step – 1:**

Import the libraries and dataset.



**Step – 2:**

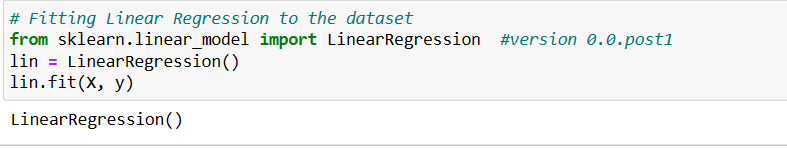
Dividing the dataset into 2 components

Divide dataset into two components that is X and y.X will contain the Column between 1 and 2. y will contain the 2 columns.



**Step – 3:**

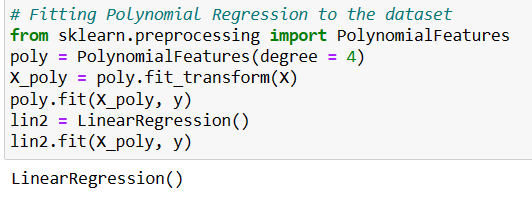
Fitting Linear Regression to the dataset. Fitting the linear Regression model On two components.



**Step – 4:**

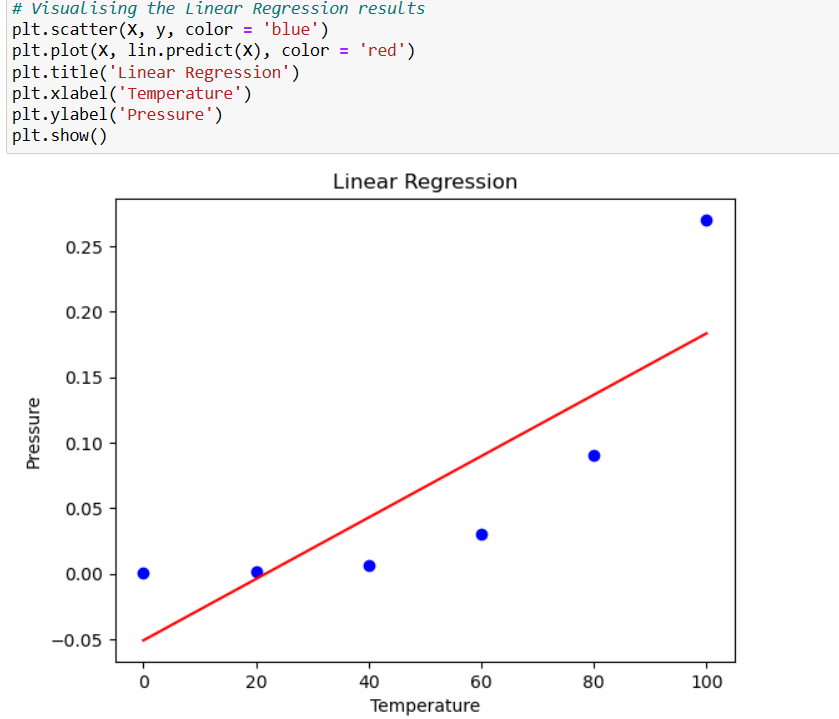
Fitting Polynomial Regression to the dataset

Fitting the Polynomial Regression model on two components X and y.



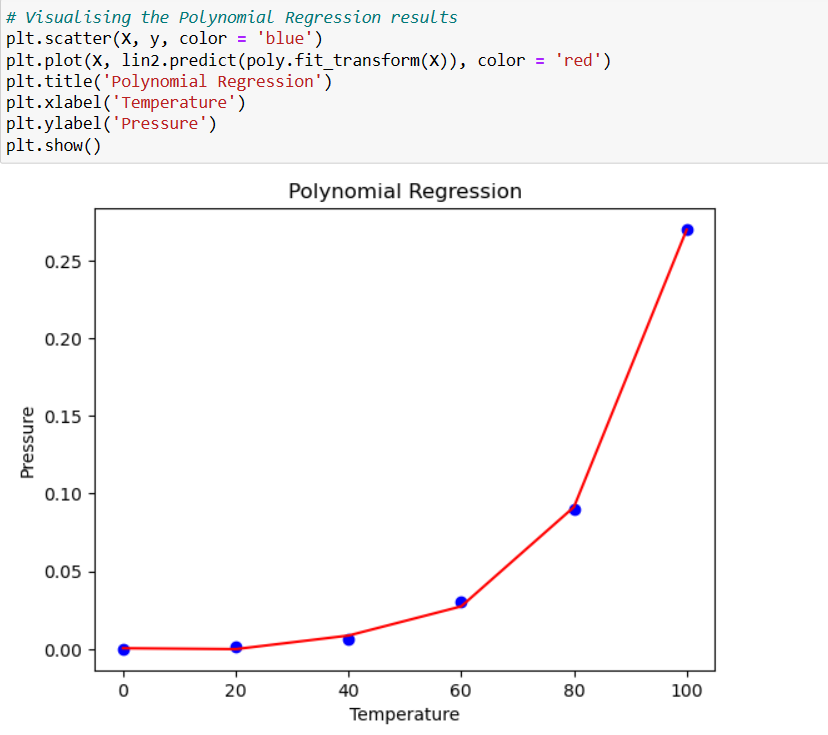
**Step – 5:**

In this step, we are Visualising the Linear Regression results using a scatter plot.



**Step – 6:**

Visualising the Polynomial Regression results using a scatter plot.



**CHAPTER – 4**

**METHODOLOGY**

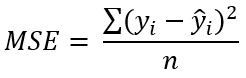
**METHODOLOGY**

**Evaluation of regression model**

There are mainly three error metrics that are commonly used for evaluating and reporting the performance of a regression model. They are –

1. Mean Squared Error (MSE)
2. Root Mean Squared Error (RMSE)
3. Mean Absolute Error (MAE)
4. **MSE**

MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.



Where,

n = total number of terms for which the error is to be calculated

yi = observed value of variable

ȳi = predicted value of variable

MSE = 1 (High) model is worst

MSE = 0 (Low) model is best

**Example -**

**Problem:**

Consider the given data.

|  |  |  |
| --- | --- | --- |
| **Predicted/Expected value** | **Observed/Actual value** | **Error** |
| 5 | 10 | 5 |
| 11 | 19 | 8 |
| 37 | 32 | -5 |
| 9 | 9 | 0 |
| 21 | 30 | 9 |
| 48 | 43 | -5 |
| 33 | 21 | -12 |
| 25 | 22 | -3 |
| 12 | 15 | 3 |

**Solution:**

**Step – 1:**

Square the errors.

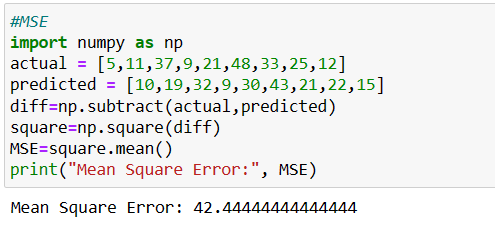
|  |  |
| --- | --- |
| **Error** | **Squared error** |
| 5 | 25 |
| 8 | 64 |
| -5 | 25 |
| 0 | 0 |
| 9 | 81 |
| -5 | 25 |
| -12 | 144 |
| -3 | 9 |
| 3 | 9 |

**Step – 2:**

Sum the squared errors and divide the result by the number of examples(calculate the average)

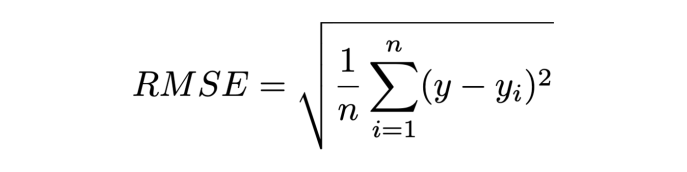
MSE = (25+64+25+0+81+25+144+9+9)9 = 42.44

Now, let us implement in python.



1. **RMSE**

* Root Mean Square Error (RMSE) is the root squared mean of the difference between actual and predicted values.
* RMSE can be used in situations where we want to penalize high errors but not as much as MSE does.
* RMSE range can be 0 to ∞ but lower value is better.



Where,

yi = actual value of variable

ȳi = predicted value of variable

n = number of data points

**Example –**

**Problem:**

Consider the given data.

|  |  |  |
| --- | --- | --- |
| **Predicted/Expected value** | **Observed/Actual value** | **Error** |
| 5 | 10 | 5 |
| 11 | 19 | 8 |
| 37 | 32 | -5 |
| 9 | 9 | 0 |
| 21 | 30 | 9 |
| 48 | 43 | -5 |
| 33 | 21 | -12 |
| 25 | 22 | -3 |
| 12 | 15 | 3 |

**Solution:**

**Step – 1:**

Square the errors.

|  |  |
| --- | --- |
| **Error** | **Squared error** |
| 5 | 25 |
| 8 | 64 |
| -5 | 25 |
| 0 | 0 |
| 9 | 81 |
| -5 | 25 |
| -12 | 144 |
| -3 | 9 |
| 3 | 9 |

**Step – 2:**

Sum the squared errors and divide the result by the number of examples(calculate the average)

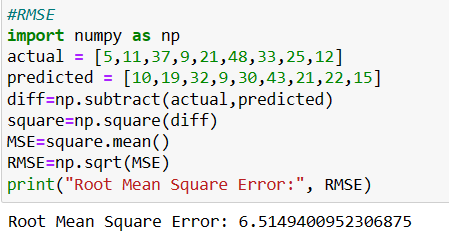
MSE = (25+64+25+0+81+25+144+9+9)9 = 42.44

**Step – 3:**

Calculate the square root of the average

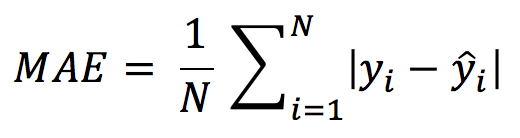
RMSE = √(42.44) = 6.51

Now, let us implement in python.



1. **MAE**

* The Mean Absolute Error(MAE) represents the average of the absolute difference between the actual and predicted values in the dataset.
* It measures the average of the residuals in the dataset.
* Used: difference between 30 & 0 is thrice difference between 10 & 0.
* Range 0 to ∞. Lower value is better.



Where,

yi = actual value of variable

ȳi = predicted value of variable

n = number of data points

**Example –**

**Problem:**

Consider the given data.

|  |  |  |
| --- | --- | --- |
| **Predicted/Expected value** | **Observed/Actual value** | **Error** |
| 5 | 10 | 5 |
| 11 | 19 | 8 |
| 37 | 32 | -5 |
| 9 | 9 | 0 |
| 21 | 30 | 9 |
| 48 | 43 | -5 |
| 33 | 21 | -12 |
| 25 | 22 | -3 |
| 12 | 15 | 3 |

**Solution:**

**Step – 1:**

Sum the errors..

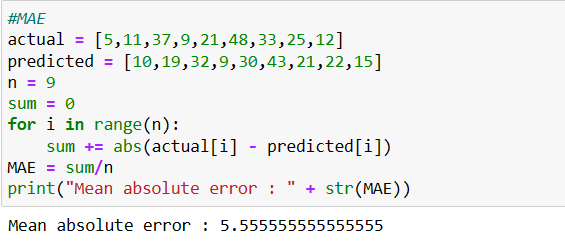
sum = |5+8+(-5)+0+9+(-5)+(-12)+(-3)+3| = |25| + |-25| = 25 + 25 = 50

**Step – 2:**

Divide the result by the number of examples.

MAE = 50/9 = 5.55

Now, let us implement in python.



**CHAPTER – 5**

**IMPLEMENTATION**

**Implementation**

Now let us implement polynomial regression in python.

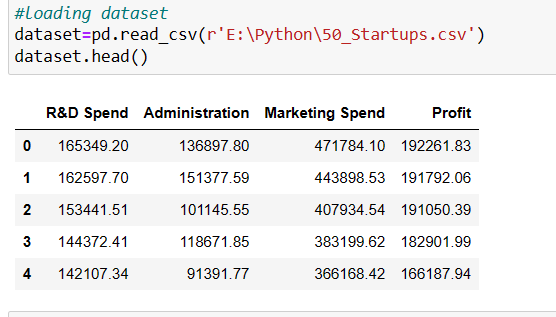
**Step – 1:**

Import the libraries.



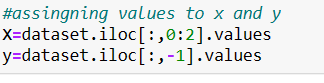
**Step – 2:**

Load the dataset.



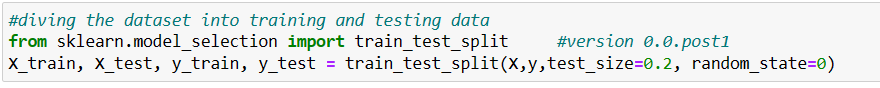
**Step – 3:**

Divide the dataset into X and y. X will contain the Column between 0 and 2 i.e., R&D Spend, Administration, Marketing Spend. And these are all independent variables. y will contain the last column. i.e., Profit which is a dependent variable.



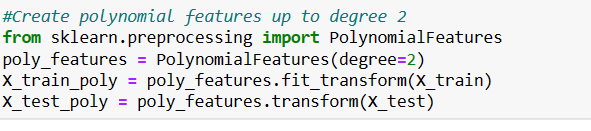
**Step – 4:**

Divide the data set into training and testing sets such that training set is 40 columns (80%) and testing set is 10 columns (20%). So the test size is 0.2



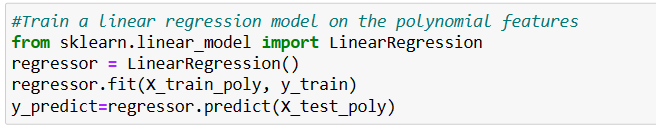
**Step – 5:**

Import polynomial regression method.



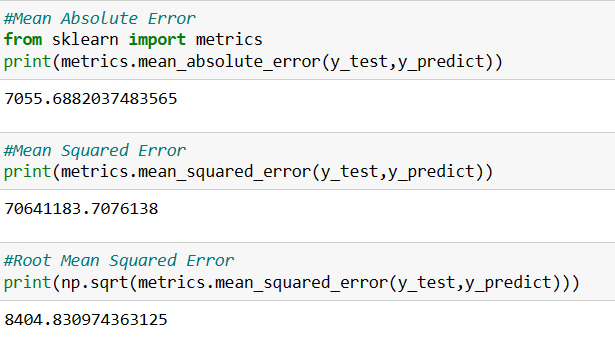
**Step – 6:**

Import linear regression method.



**Step – 7:**

Apply regression metrics like Mean Absolute Error, Mean Squared Error, Root Mean Squared Error.



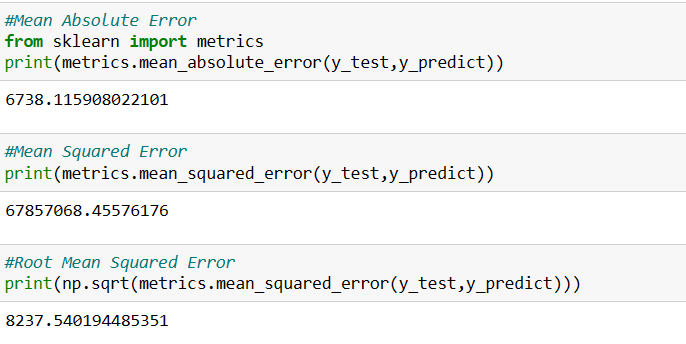
**CHAPTER – 6**

**CONCLUSION**

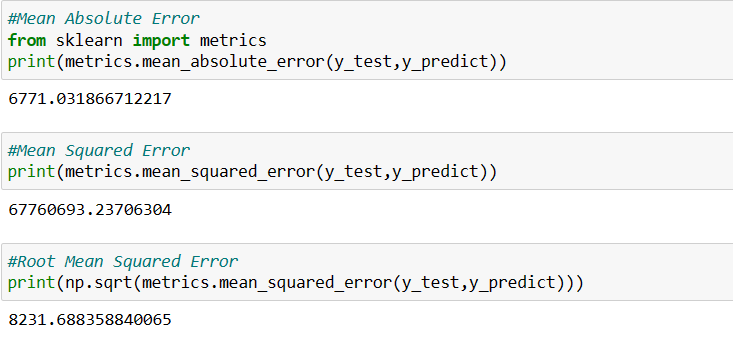
**Conclusion**

Let us conclude by comparing the regression metrics obtained from simple linear regression, multiple linear regression and polynomial regression.

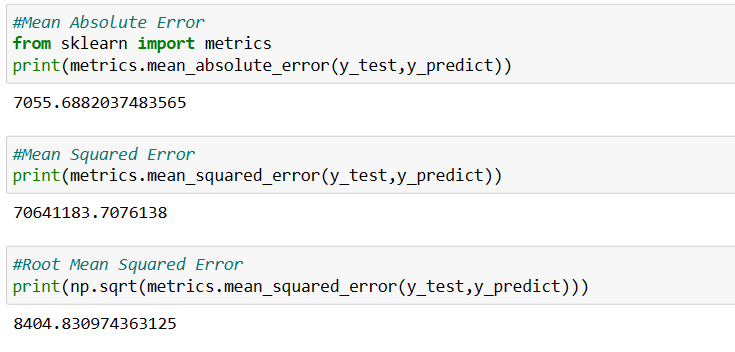
1. **Simple linear regression**



1. **Multiple linear regression**



1. **Polynomial regression**



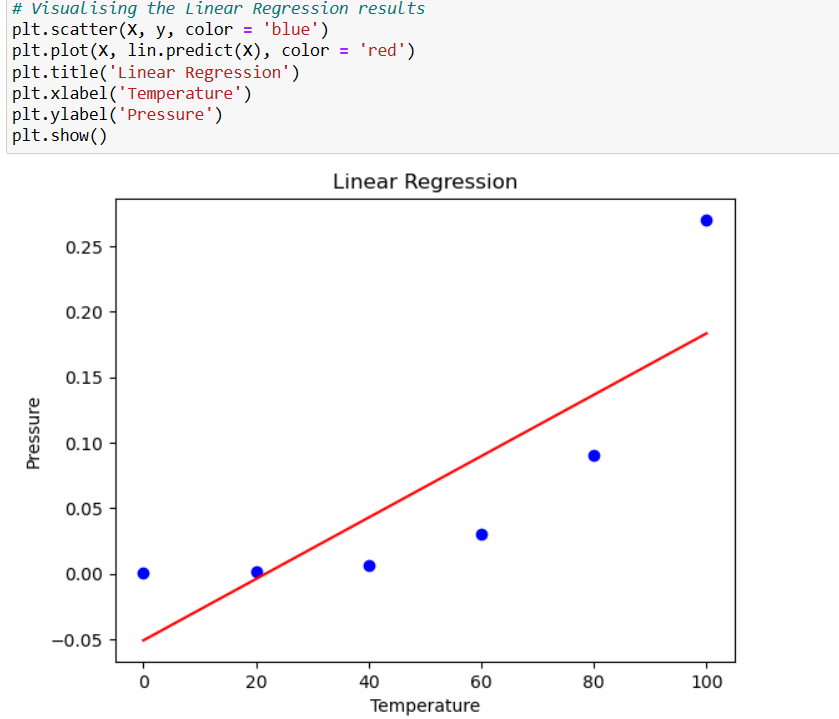
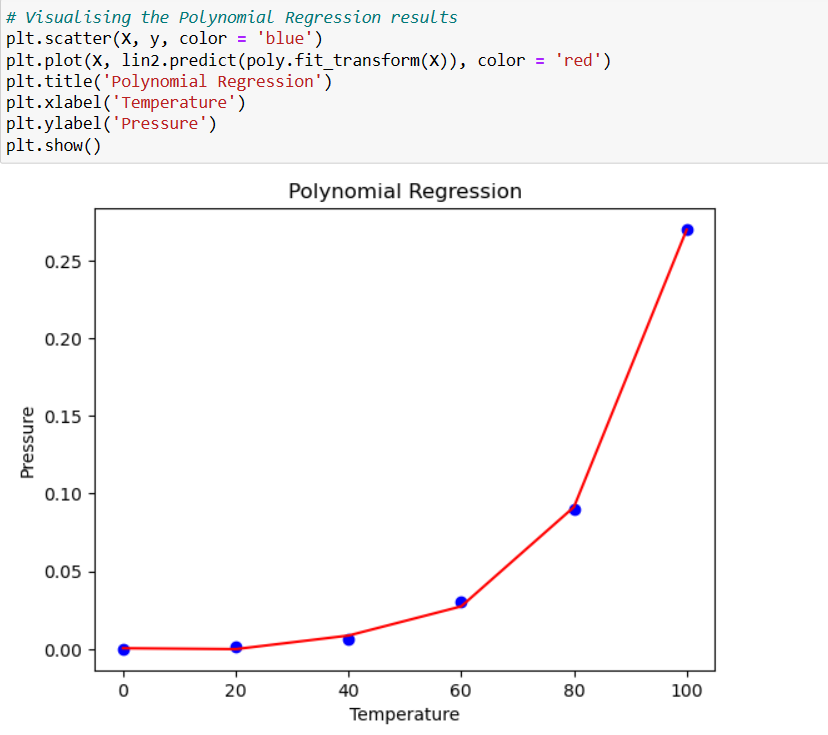
These can be written as -

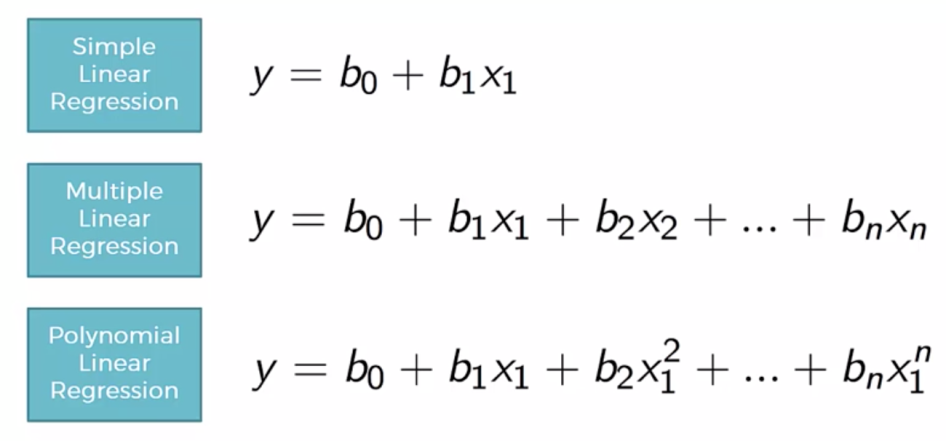
|  |  |  |  |
| --- | --- | --- | --- |
| **Regression Types/Regression metrics** | **Mean Absolute Error** | **Mean Squared Error** | **Root Mean Squared Error** |
| **Linear Regression** | 6738.115908022101 | 67857068.45576176 | 8237.540194485351 |
| **Multiple Regression** | 6771.031866712217 | 67760693.23706304 | 8231.688358840065 |
| **Polynomial Regression** | 7055.6882037483565 | 70641183.7076138 | 8404.830974363125 |

The above table clearly shows the difference between regression types and regression metrics. By using polynomial regression will provide accurate and exact results than other regressions.

**Need for polynomial regression**

The need of Polynomial Regression in ML can be understood in the below points:

* If we apply a linear model on a linear dataset, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a non-linear dataset, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.
* So for such cases, where data points are arranged in a non-linear fashion, we need the Polynomial Regression model. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.
* In the above image, we have taken a dataset which is arranged non-linearly. So if we try to cover it with a linear model, then we can clearly see that it hardly covers any data point. On the other hand, a curve is suitable to cover most of the data points, which is of the Polynomial model.
* Hence, if the datasets are arranged in a non-linear fashion, then we should use the Polynomial Regression model instead of Simple Linear Regression.



The Simple and Multiple Linear Equations are also Polynomial equations with a single degree, and the Polynomial regression equation is a Linear equation with the nth degree. Therefore, if we add a degree to our linear equations, then it will be converted into Polynomial Linear Equations..