**WEEK 1**

**AIM:**

Implement a Regression algorithm in order to fit data points. Select appropriate data set for the experiment

**DESCRIPTION:**

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Mathematically, we can represent a linear regression as:

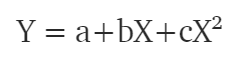
y= a0+a1x+ ε

**Here,**

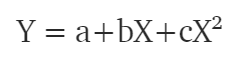
Y= Dependent Variable(Target Variable)  
X= Independent Variable (predictor Variable)  
a0= intercept of the line (Gives an additional degree of freedom)  
a1 = Linear regression coefficient (scale factor to each input value).  
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

**Simple Linear Regression**: This is one of the most common and interesting type of Regression technique. Here we predict a target variable Y based on the input variable X. A linear relationship should exist between target variable and predictor and so comes the name Linear Regression.



**Polynomial Regression**: In polynomial regression, we transform the original features into polynomial features of a given degree and then apply Linear Regression on it. Consider the above linear model Y = a+bX is transformed into something like



**CODE:**

import numpy as np

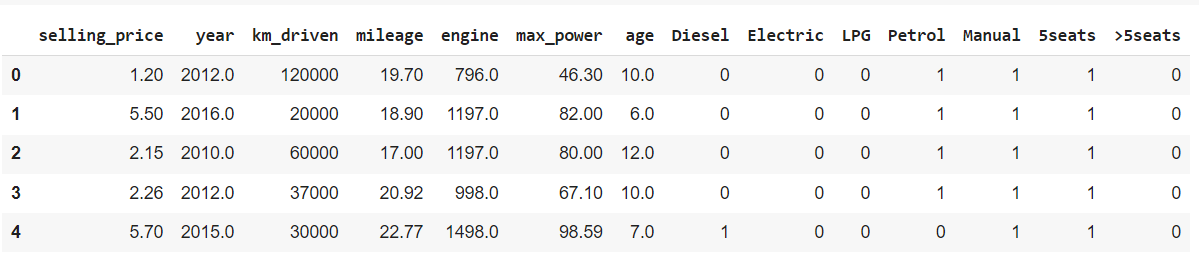
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('/content/cars24-price.csv')

df.head()



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

X = df.drop('selling\_price',axis=1)

Y = df['selling\_price']

X\_train , X\_test , Y\_train ,Y\_test = train\_test\_split( X ,Y , train\_size =0.3 , random\_state=1)

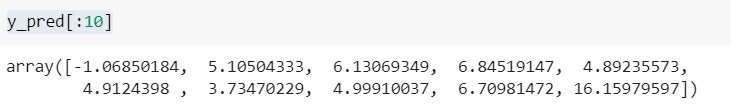
from sklearn.linear\_model import LinearRegression

model = LinearRegression()

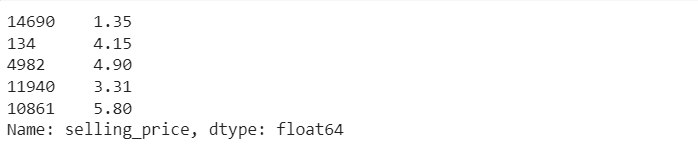
model.fit(X\_train , Y\_train)

y\_pred =model.predict(X\_test)

y\_pred[:10]



Y\_test[:10]



model.score(X\_test,Y\_test)



model.intercept\_

