1.1 PROBLEM STATEMENT:

We have given with a data set that contains the price of diamond according to their carat; we need a machine learning model which can tell us the price of diamond if we give him the carat, with highest possible accuracy.

1.2 GIVEN DATA

Data is the fact or statistics collected together for Analysis.

Here we have given a data set of diamond that contains the price of diamond according to their carat. We will feed the data to our model and our model will be prepared for new analysis and will give us the price of diamond we want based on the dataset.

The given data-set :

	Carat	Price
0	0.23	326
1	0.21	326
2	0.23	327
3	0.29	334
4	0.31	335

A dataset contains two types of variables i.e. independent and dependent variable. Here the price is depend upon the carat that means carat is independent variable and price is the dependent variable. By assigning independent variable to x and dependent variable to y, so the x-axis = carat and y-axis = price.

1.3 MODEL USED:

1.3.1 SIMPLE LINEAR REGRESSION:

In simple linear regression, we predict scores on one variable from the scores on a second variable. The variable we are predicting is called the dependent variable (Y). The variable we are basing our predictions on is called the independent variable (X). When there is only one dependent variable, then the method is called **simple regression.** In simple linear regression, the predictions of Y when plotted as a function of X form a straight line.

The black line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best. The line can be modeled based on the linear equation shown below.

y = a_0 + a_1 * x ### Linear Equation

The motive of the linear regression algorithm is to find the best values for a_0 and a_1

1.3.2 POLYNOMIAL REGRESSION:

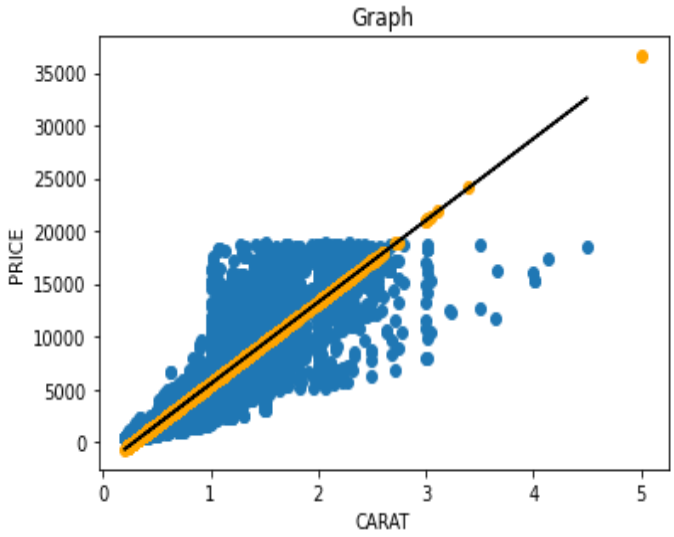
Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression. It is also called the special case of Multiple Linear Regression in ML.The polynomial regression to fit a polynomial line so that we can achieve a minimum error or minimum cost function. The equation of the polynomial regression for the above graph data would

This is the general equation of a polynomial regression is

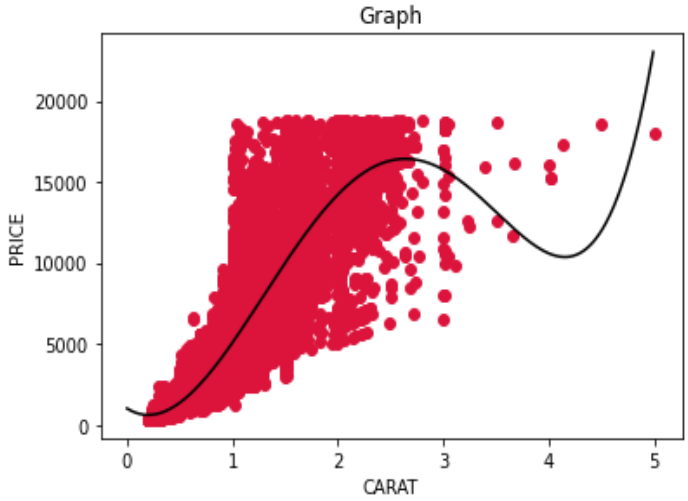
$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+..... b_nx_1^n$$

1.4 GRAPHS:

1.4.1 SIMPLE LINEAR REGRESSION MODEL:



1.4.2 POLYNOMIAL REGRESSION MODEL:



2.1 SIMPLE LINEAR REGRESSON:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

data = pd.read_csv('Assignment-1_Diamond_price.csv')

data.head()

x = data[['carat']]

y = data['price']

from sklearn.linear_model import LinearRegression

lr=LinearRegression()

lr.fit(x_train,y_train)

predictions=lr.predict(x_test)

Y_pred=lr.predict(x_train)

from sklearn.metrics import r2_score

y_pred=lr.predict(x_test)

r2_score(y_pred, y_test)

```
plt.scatter(x_train, y_train)
plt.plot(x_train, Y_pred, color='black')
plt.scatter(x_test, predictions, color='orange')
plt.title('Simple Linear Regression')
plt.xlabel('carat')
plt.ylabel('price')
plt.show()
```

2.2 POLYNOMIAL REGRESSION:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('Assignment-1_Diamond_price.csv')
data.head(10)

x = data[['carat']]
y = data['price']

from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(x, y)
lin_preds= lin_reg.predict(x)

from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 5)
x_poly = poly_reg.fit_transform(x)
print(x_poly)
```

```
from sklearn.model_selection import train_test_split

xtr, xts, ytr, yts=train_test_split(x_poly,y, test_size=0.25, random_state=0)


lin_reg_2 = LinearRegression()

model=lin_reg_2.fit( xtr , ytr )

lin_reg_2.fit(xtr,ytr)


y_pred_poly = lin_reg_2.predict(xts)

print(model.intercept_)
print(model.coef_)

from sklearn.metrics import r2_score

r2_score(y_pred_poly,yts)*100

plt.scatter(x, y, color = 'crimson')

x_grid=np.arange(0,5, 0.01)

x_grid=x_grid.reshape(len(x_grid),1)

plt.plot(x_grid,lin_reg_2.predict(poly_reg.fit_transform(x_grid)), color='black')


plt.title('Polynomial Regression')

plt.xlabel('carat')

plt.ylabel('price')

plt.show()
```