# Machine Learning Project

In this project, I have used **Linear Regression** Machine Learning model for the PC parts prices Dataset from kaggle website.

## Problem Statement

Compare CPU prices with raising demand for memory chips in smart phone industry and older parts tend to stop being made. Prices don't really go down. Prices stay static. High Prices of certain components may hikes the actual price of the PCs parts. By Checking the Product id, Merchant Id and Region Id we can predict the Original Price of the CPU.

## Machine Learning (Methodology)

The methodology used for training and testing the dataset is **Linear Regression**.

**Linear Regression** is a method to explain the relationship between a dependent variable and one or more explanatory variables using a straight line. It is a special case of regression analysis. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Models depend linearly on their unknown parameters are easier to fit the models which are non-linearly related to their parameters.

## Dataset Description

### Some relevant columns in the dataset

- 1664-CPU Products
- ProdId -Id of a CPU Product
- TimeId -Id of a Time Dimension
- RegionId -Id of a Region Dimension
- Merchant -Id of a Merchant Dimension
- Price_USD -Price in USD
- Price_Original -Price in Regional Currency

# Pre-Processing:

Pre-processing involves transforming raw data into an understandable format. For example: Extracting data from a larger dataset.

1. %matplotlib inline

   Import matplotlib.pyplot as plt

   Import numpy as np

   import pandas as pd

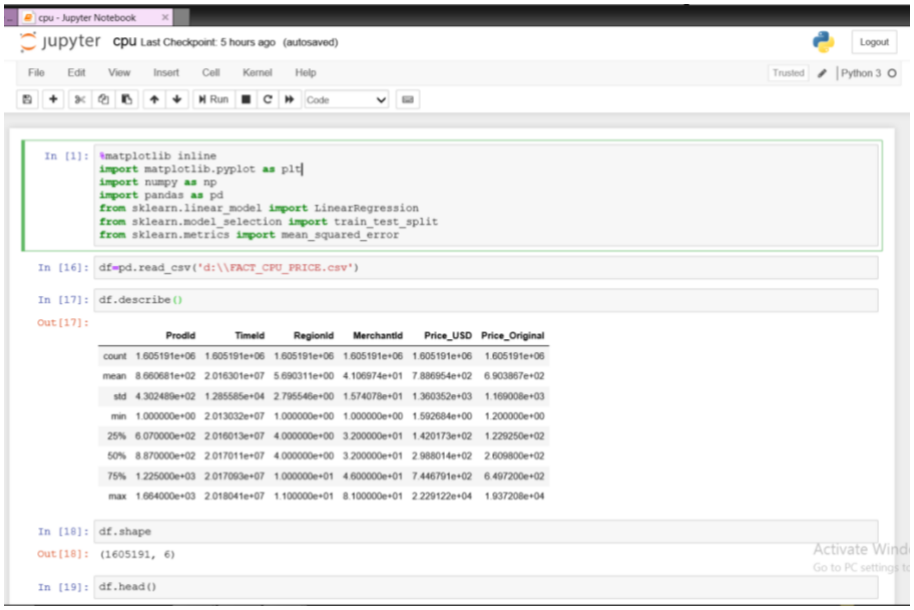
   from sklearn.linear_model import LinearRegression

   from sklearn.model_selection import train_test_split

   from sklearn.metrics import mean_squared_error

2. df=pd.read_csv('d:\\FACT_CPU_PRICE.csv')

   df.describe()



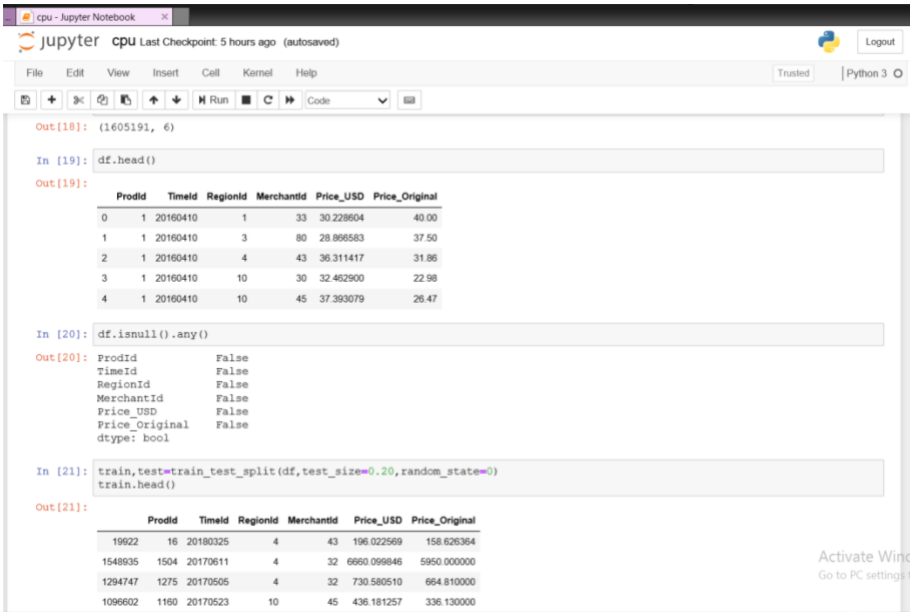

**In[3]:**df.shape

**In[4]:**df.head()

**In[5]:**df.isnull().any()

**In[6]:**train,test=train_test_split(df,test_size=0.20,random_state=0)

train.head()




**In[7]:**X_train=train[['RegionId','MerchantId','Price_USD']]

**In[8]:**y_train=train.Price_Original

**In[9]:**X_test=test[['RegionId','MerchantId','Price_USD']]

**In[10]:**y_test=test.Price_Original

**In[11]:**print(X_train.shape)

**In[12]:**print(X_test.shape)

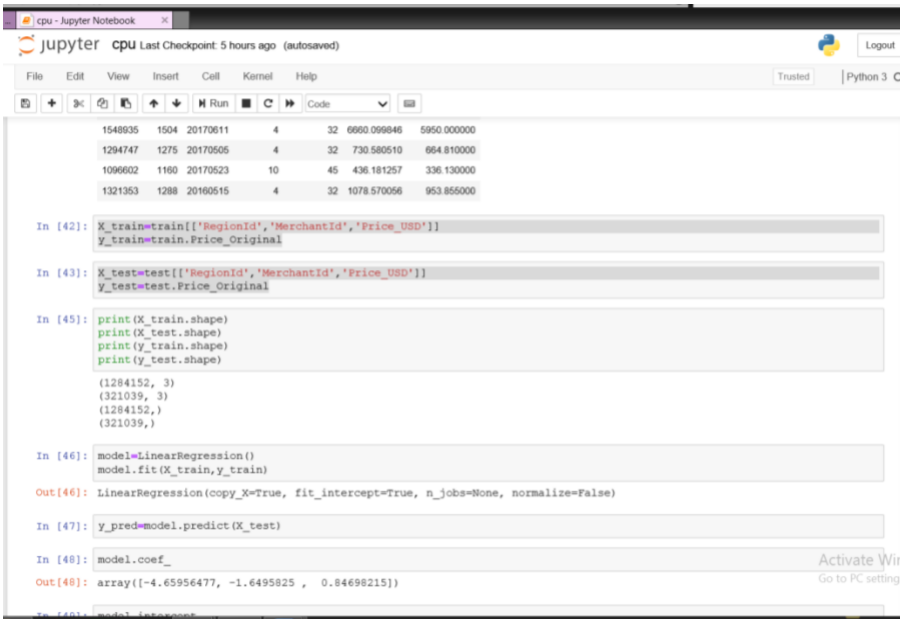
**In[13]:**print(y_train.shape)

**In[14]:**print(y_test.shape)

**In[15]:**10.model=LinearRegression()

**In[16]:**model.fit(X_train,y_train)

**In[17]:**y_pred=model.predict(X_test)




**In[18]:**model.coef_

**In[19]:**model.intercept_

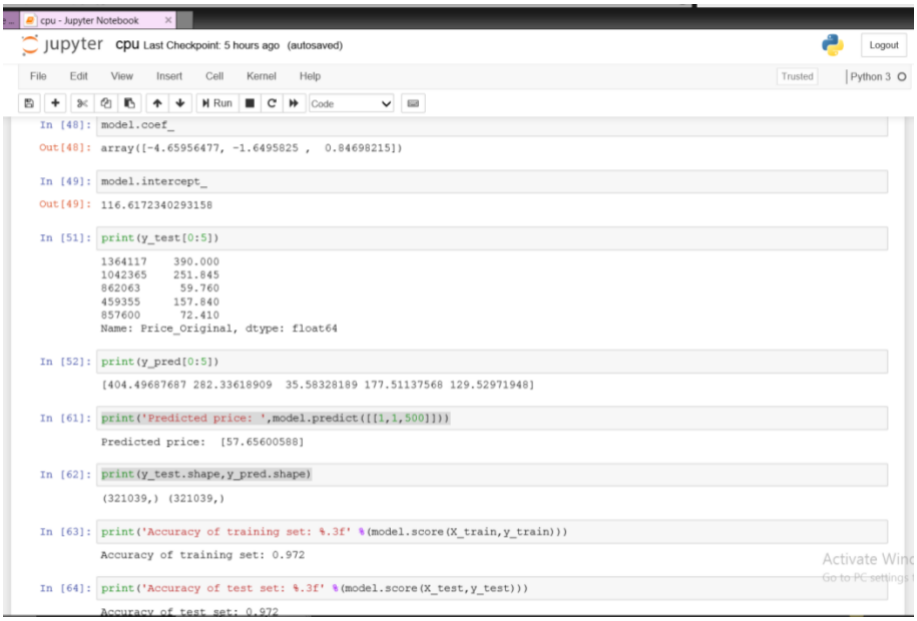
**In[20]:**print(y_test[0:5])

**In[21]:**print(y_pred[0:5])

**In[22]:**print('Predicted price: ',model.predict([[1,1,500]]))

**In[23]:**print(y_test.shape,y_pred.shape)

**In[24]:**print('Accuracy of training set: %.3f' %(model.score(X_train,y_train)))




**In[25]:**print('Accuracy of test set: %.3f' %(model.score(X_test,y_test)))

**In[26]:**mse=mean_squared_error(y_test,y_pred)

**In[27]:**print('Mean Squared Error: %.3f' %mse)

**In[28]:**y_pred=y_pred.round()

**In[29]:**y_test=y_test.astype(float).round()

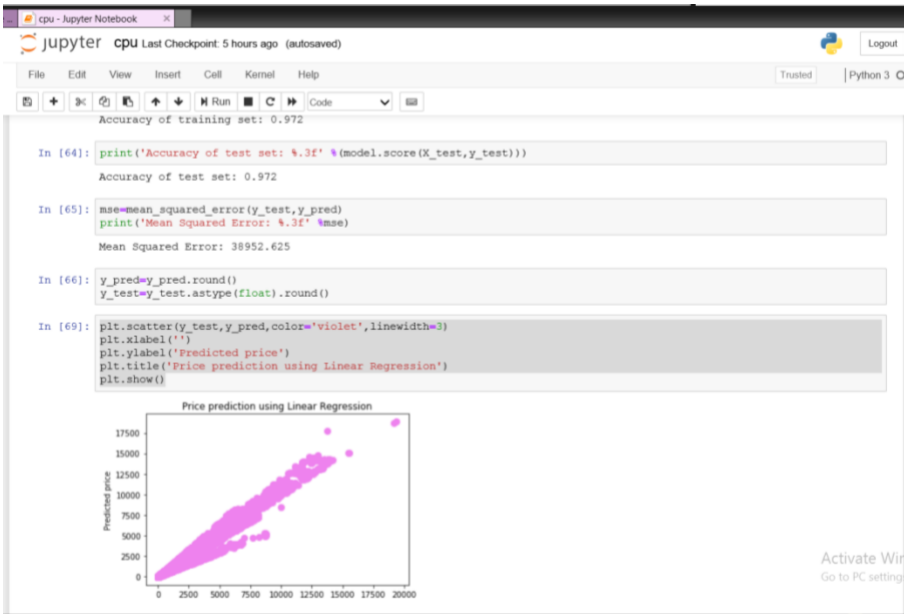
**In[30]:**plt.scatter(y_test,y_pred,color='violet',linewidth=3)

**In[31]:**plt.xlabel('')

**In[32]:**plt.ylabel('Predicted price')

**In[33]:**plt.title('Price prediction using Linear Regression')

**In[34]:**plt.show()




## Conclusion:

Here the CPU price is predicted with merchant Id, Region Id and Price_USD using Linear Regression. So that we could find the cheapest PC’s parts .