**Ex No: 3**

**LINEAR REGRESSION**

**AIM**:

To perform prediction with Linear regression using random linear regression dataset.

**Dataset Description:**

The dataset contains information on over 2,000 mobile phones from different brands. It includes details such as the storage capacity, RAM, screen size, camera specifications, battery capacity, and price of each device.

The dataset is structured as a CSV file with 7 columns:

* Brand: The brand name of the mobile phone.
* Model: The model name of the mobile phone.
* Storage: The amount of storage space available on the mobile phone in GB.
* RAM: The amount of random access memory available on the mobile phone in GB.
* Screen Size: The size of the mobile phone's screen in inches.
* Camera: The quality of the mobile phone's cameras, measured in megapixels.
* Battery Capacity: The amount of battery life the mobile phone has in mAh.
* Price: The price of the mobile phone in USD.

**Problem Statement**

The mobile phone price prediction problem is to develop a model that can predict the price of a mobile phone given a set of features. The target variable is the price of the mobile phone in USD. The goal of the problem is to develop a model that can accurately predict the price of a mobile phone given its features. This model can be used by a variety of stakeholders, including:

Mobile phone manufacturers: Manufacturers can use the model to develop a pricing strategy for their products. They can also use the model to identify the features that are most important to consumers and to determine how much they should charge for their phones based on those features.

Retailers: Retailers can use the model to set prices for mobile phones in their store. They can also use the model to compare the prices of different phones from different manufacturers and to ensure that they are charging a competitive price.

Consumers: Consumers can use the model to make informed decisions about which mobile phone to buy. They can use the model to compare the prices of different phones with different features and to find the best value for their money.

**PROGRAMS WITH OUTPUT:**

Linear Regression Using numpy

*def mean(values):*

*return sum(values)/float(len(values))*

*def variance(values,mean):*

*return sum([(x-mean)\*\*2 for x in values])*

*def covariance(x,mean\_x,y,mean\_y):*

*covar=0.0*

*for i in range(len(x)):*

*covar+=(x[i]-mean\_x) \* (y[i]-mean\_y)*

*return covar*

*def coefficients(dataset):*

*b1=covariance(x,mean\_x,y,mean\_y)/variance(x,mean\_x)*

*b0=mean\_y-b1\*mean\_x*

*return[b0,b1]*

*def simple\_linear\_regression(train,test):*

*for row in test:*

*ytest = b0 + b1 \* row[0]*

*return ytest*

*dataset=[[50,28],[60,40],[48,45],[70,50],[55,50],[60,38],[45,20]]*

*x=[row[0] for row in dataset]*

*y=[row[1] for row in dataset]*

*mean\_x=mean(x)*

*mean\_y=mean(y)*

*variance\_x=variance(x,mean\_x)*

*variance\_y=variance(y,mean\_y)*

*print('x stats:mean=%.3f variance=%.3f' % (mean\_x,variance\_x))*

*print('y stats:mean=%.3f variance=%.3f' % (mean\_y,variance\_y))*

*covar = covariance(x,mean\_x,y,mean\_y)*

*print('covariance: %.3f' % (covar))*

*b0,b1 = coefficients(dataset)*

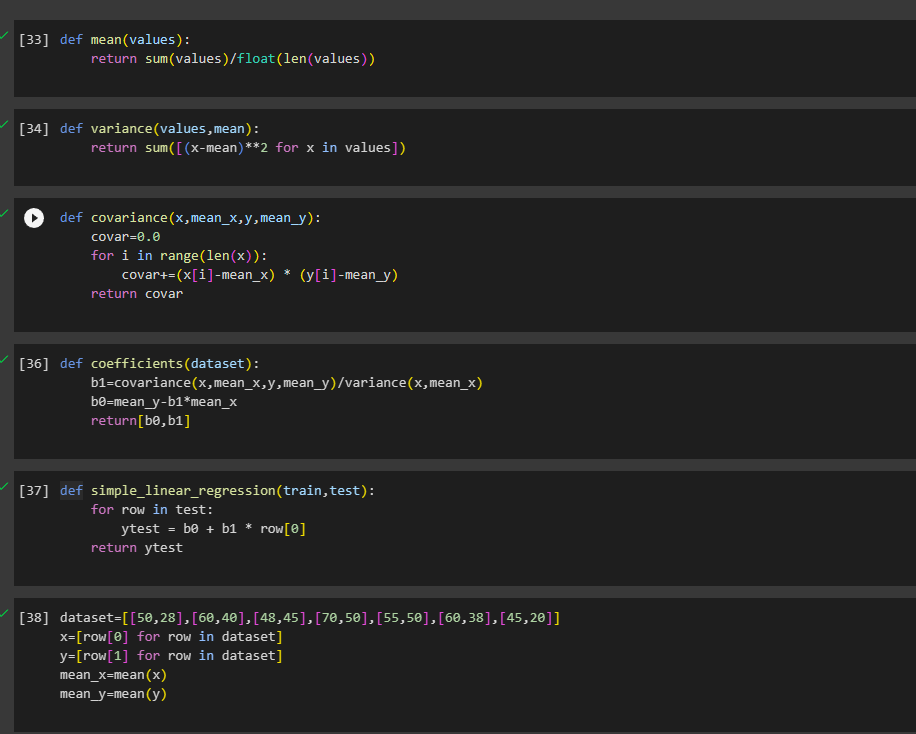
*print('coefficients:b0=%.3f,b1=%.3f' % (b0,b1))*

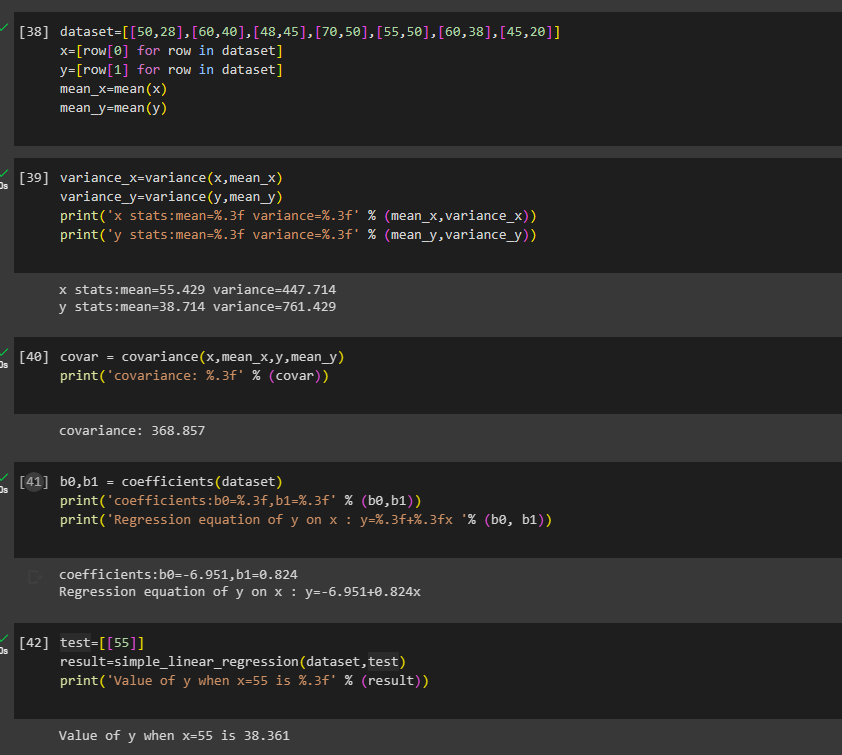
*print('Regression equation of y on x : y=%.3f+%.3fx '% (b0, b1))*

*test=[[55]]*

*result=simple\_linear\_regression(dataset,test)*

*print('Value of y when x=55 is %.3f' % (result))*





Linear Regression Using Sklearn

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

*Xtrain,Xtest,ytrain,ytest=train\_test\_split(X,y,test\_size=0.3,random\_state=1)*

*from sklearn.preprocessing import StandardScaler*

*scaler=StandardScaler()*

*Xtrain=scaler.fit\_transform(Xtrain)*

*Xtest=scaler.transform(Xtest)*

*from sklearn.linear\_model import LinearRegression*

*model = LinearRegression()*

*model.fit(Xtrain, ytrain)*

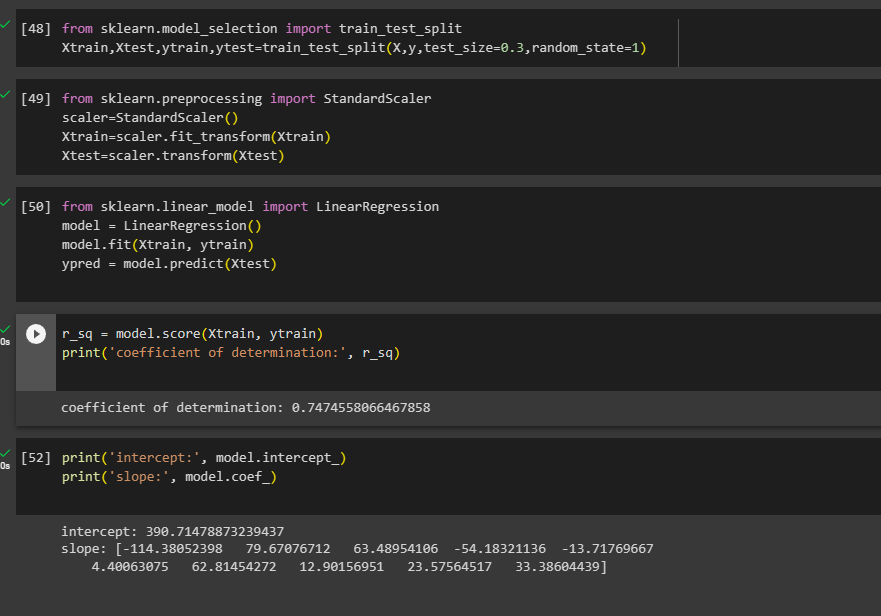
*ypred = model.predict(Xtest)*

*r\_sq = model.score(Xtrain, ytrain)*

*print('coefficient of determination:', r\_sq)*

*print('intercept:', model.intercept\_)*

*print('slope:', model.coef\_)*



**CONCLUSION:**

The predicted output is displayed using the linear regression model trained with the given dataset and results are verified.