**JUNAID GIRKAR**

**60004190057**

**TE COMPS A4**

**DWM**

**EXPERIMENT 3**

**LINEAR REGRESSION**

AIM: Implementation of Linear Regression

1. Single Variate

2. Multi Variate

**THEORY:**

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression :



While training the model we are given :

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.

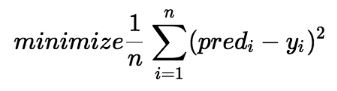
θ1: intercept

θ2: coefficient of x

Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

**Cost Function (J):**

By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).





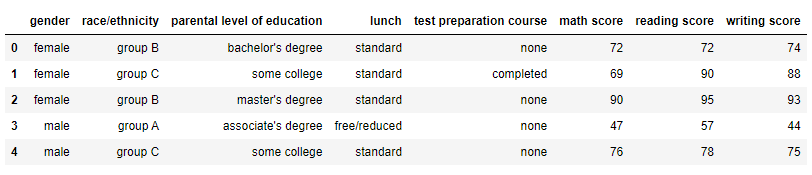
Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

**CODE:**

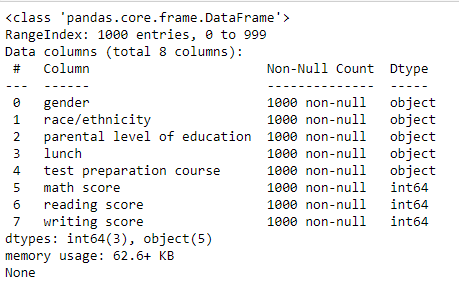
| import warnings import numpy as np import pandas as pd import seaborn as sns import matplotlib.pyplot as plt  from sklearn.metrics import mean\_squared\_error from sklearn.linear\_model import LinearRegression from sklearn.preprocessing import LabelEncoder from sklearn.metrics import accuracy\_score import matplotlib.pyplot as plb from sklearn.model\_selection import train\_test\_split  sns.set() warnings.simplefilter("ignore")  df = pd.read\_csv("StudentsPerformance.csv") df.head()  print(df.info())  df['final score'] = df.apply(lambda x : (x['math score'] + x['reading score'] + x['writing score']) / 3, axis=1)  df.head() data2 = df.drop('final score', axis=1) plt.figure(figsize=(16, 6)) sns.boxplot(data=data2)  df = df.apply(LabelEncoder().fit\_transform)  # MULTIVARIATE  X = df.drop('final score', axis=1) y = df['final score'] X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 0.2) lr = LinearRegression() lr.fit(X\_train, y\_train)  pred = lr.predict(X\_test)  lr.score(X\_test, y\_test) accuracy = mean\_squared\_error(y\_test, pred) print('Mean Squared Error: ', accuracy)  # UNIVARIATE sns.scatterplot(df["writing score"],df["final score"]) plt.savefig('scp-1', dpi=500) m, b = np.polyfit(df["writing score"], df["final score"], 1)  plt.plot(df["writing score"], m\*df["writing score"] + b)  X\_uni = df['writing score'] y\_uni = df['final score'] X\_uni\_train, X\_uni\_test, y\_uni\_train, y\_uni\_test = train\_test\_split(X\_uni,y\_uni,test\_size = 0.2)  lr2 = LinearRegression() X\_uni\_train = X\_uni\_train.reshape(-1,1) X\_uni\_test = X\_uni\_test.values.reshape(-1,1)  lr2.fit(X\_uni\_train, y\_uni\_train) pred\_uni = lr2.predict(X\_uni\_test) lr2.score(X\_uni\_test, y\_uni\_test)  accuracy\_uni = mean\_squared\_error(y\_uni\_test, pred\_uni) print('Mean Squared Error: ', accuracy\_uni) |
| --- |

OUTPUT:

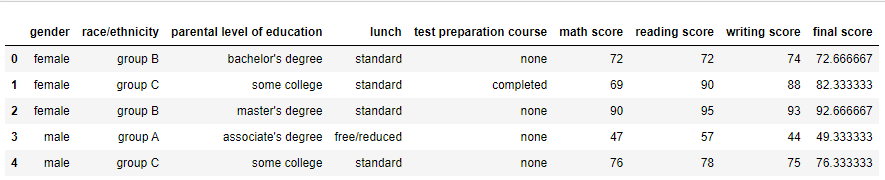
| head() of the database: |
| --- |



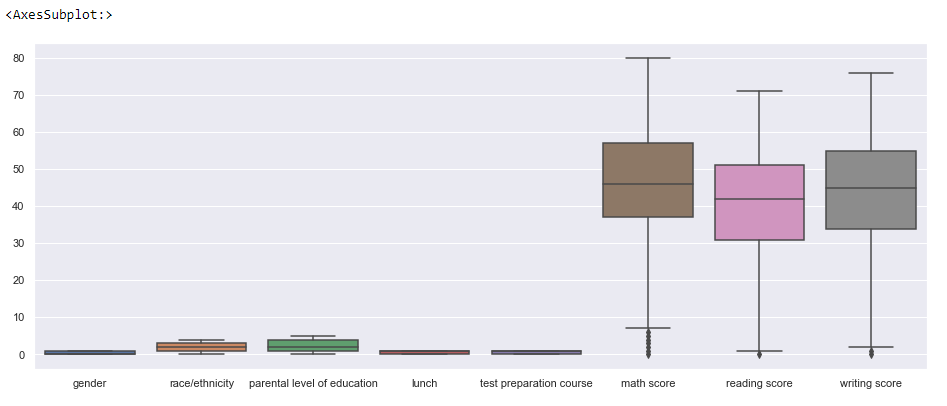
| After running df.info() |
| --- |



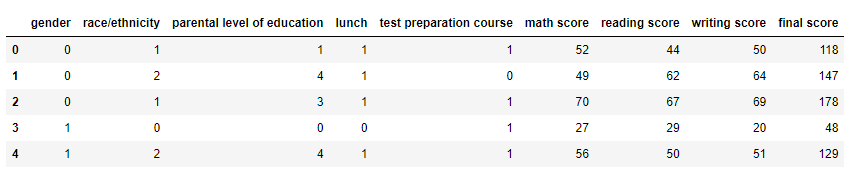
| df.head() after adding a final score column |
| --- |



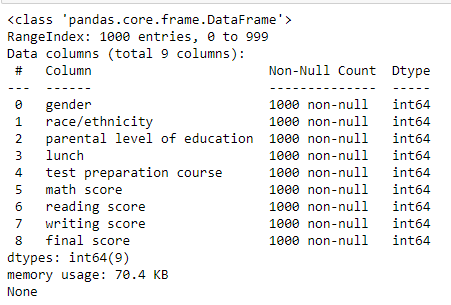
| Boxplot of the features |
| --- |



| df.head() after applying LabelEncoder to the dataset |
| --- |

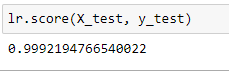


| df.info() after applying LabelEncoder to the dataset |
| --- |

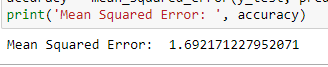


**Considering Multivariate Linear Regression**

| Prediction Score of MultiVariate Linear Regression |
| --- |

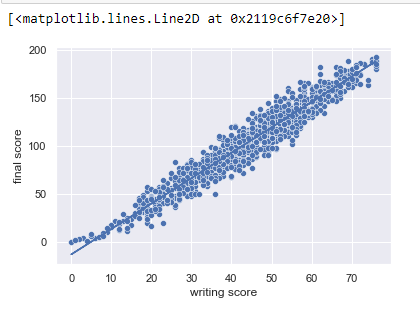


| Mean Square Error of MultiVariate Linear Regression |
| --- |

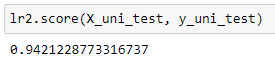


**Now considering Univariate Linear Regression with Writing Score as the feature**

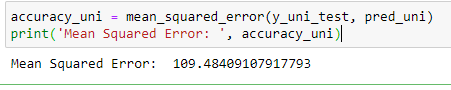
| Scatter Plot of the dataset |
| --- |



| Prediction Score of Univariate LR |
| --- |



| Mean Square Error of Univariate LR |
| --- |



**CONCLUSION**: We have implemented Multivariate and Univariate Linear Regression on a dataset and have observed the differences in their Accuracy Score and Mean Squared Errors. We observe 99.92% accuracy in the case of Multivariate with a Mean Squared Error of 1.62 whereas in the case of Univariate, the accuracy score is 94.21% and the Mean Squared Error is 109.48. Therefore we can conclude that using Multivariate Linear Regression is better than using Univariate but nevertheless the efficiency of Univariate is still great.