REPORT

**Introduction –** *The target is to prepare an ML Model which can predict the profit value of a company if the value of R&D spend, Administration cost, Marketing spend are given.*

**Body –**

We are given a dataset that contains information about 50 startups, with features like R&D Spend**,** Administration, Marketing Spend, and Profit. Here we have to build the machine learning model to predict the profit of the start-ups.

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable.

**Formula and Calculation of Multiple Linear Regression**

​

*Yi* ​= *β*0​ + *β*1​*xi1* + *β*2​*xi2* ​+...+ *βp*​*xip* ​+ *ϵ*

*where, for**i*=*n observations***:**

*yi*​ = dependent variable

*xi*​ = explanatory variables

*β*0​ = y-intercept (constant term)

*βp* = slope coefficients for each explanatory variable

*ϵ* = the model’s error term (also known as the residuals)​

**From our given dataset, ( Defining X and Y )**

X = Independent Variable. CODE: ‘ x=company.iloc[:,:-1].values ‘

Y = Dependent Variable. CODE: ‘ y=company.iloc[:,3].values ‘

**Data pre-processing**

* *Importing libraries* –
* Import train\_test\_split to split the dataset into training and testing datasets.

CODE:

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

Here sklearn is a Python module integrating classical machine learning algorithms in the highly-knit world of scientific Python Packages. ( numpy, scipy, matplotlib )

* Import LinearRegression, is the model on which we have to work.

CODE:

‘ from sklearn.linear\_model import LinearRegression ‘

* Import  r2\_score , which is to find the accuracy of the model.

CODE:

‘ from sklearn.metrics import r2\_score ‘

* Import  Matplotlib , used for visualizations.

CODE:

‘ import matplotlib.pyplot as plt ‘

* Import seaborn, used for visualizations.

CODE:

‘ import seaborn as sns

* Import warnings and we set it to ignore so that it will ignore all the warnings that we will come throughout.

CODE:

‘ import warnings

warnings.filterwarnings("ignore") ‘

* *Analysing the relationship between Dependent variable with other independent variable-*
* **R&D Spend** (Research and development) - is the process by which a company works to obtain new knowledge that it might use to create new technology, products, services, or systems that it will either use or sell. So, this column tells you how much you spend on R&D.

CODE:

‘x1 = company.iloc[:, 0].values

y1 = company.iloc[:, -1].values

plt.scatter(x1,y1,color='Red',s=50)

plt.xlabel('R&D')

plt.ylabel('Profit')

plt.title('R&D vs Profit')

plt.show() ‘



R&D has Positive Correlation.

* **Administration** - It is the process of arrangements and tasks needed to control the operation of a plan or organization.

CODE:

‘x2 = company.iloc[:, 1].values

y2 = company.iloc[:, -1].values

plt.scatter(x2,y2,color='Blue',s=50)

plt.xlabel('Administration')

plt.ylabel('Profit')

plt.title('Administration vs Profit')

plt.show() ‘



Zero correlation between Administration and Profit.

* **Marketing Spend** refers to activities undertaken by a company to promote the buying or selling of a product or service. Marketing includes advertising, selling, and delivering products to consumers or other businesses.

CODE:

‘x3 = company.iloc[:, 2].values

y3 = company.iloc[:, -1].values

plt.scatter(x3,y3,color='Brown',s=50)

plt.xlabel('Marketing Spend')

plt.ylabel('Profit')

plt.title('Marketing Spend vs Profit')

plt.show( ) ‘



High correlation between Marketing Spend and Profit.

**Building a Model**

* *Splitting the Dataset into train-set and test-set –*

We divide the train-set and test-set into two parts with 80% of the Dataset into training-set and 20% of the Dataset into testing-set.

CODE: ‘x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=0)’

* *Fitting the Multiple Linear Regression into training set* ***(Training the Model)****–*

CODE:

‘ regressor=LinearRegression()

regressor.fit(x\_train,y\_train) ‘

* *Predicting the Test Results (Profit) –*

We predict the results, then pass the independent variables into it and view the results. It will give the array with all the values in it.

CODE:

‘ y\_pred=regressor.predict(x\_test)

print(y\_pred)

data = pd.DataFrame(data={"Predicted Profit": y\_pred.flatten()})

print(data.head()) ‘

OUTPUT :

[103901.8969696 132763.05993126 133567.90370044 72911.78976736

179627.92567224 115166.64864795 67113.5769057 98154.80686776

114756.11555221 169064.01408795]

Predicted Profit

0 103901.896970

1 132763.059931

2 133567.903700

3 72911.789767

4 179627.925672

* *Predicted Values –*

A new data frame that contains actual values, predicted values, and differences between them so that we will understand how near the model predicts its actual value.

CODE:

‘pred\_df=pd.DataFrame({'Actual Value':y\_test,'Predicted

Value':y\_pred,'Difference':y\_test-y\_pred})

print(pred\_df) ‘

OUTPUT :

.................PREDICTED VALUE..................

Actual Value Predicted Value Difference

0 103282.38 103901.896970 -619.516970

1 144259.40 132763.059931 11496.340069

2 146121.95 133567.903700 12554.046300

3 77798.83 72911.789767 4887.040233

4 191050.39 179627.925672 11422.464328

5 105008.31 115166.648648 -10158.338648

6 81229.06 67113.576906 14115.483094

7 97483.56 98154.806868 -671.246868

8 110352.25 114756.115552 -4403.865552

9 166187.94 169064.014088 -2876.074088

Here, we can see the difference between Actual values and predicted values which are not very high. When values are in the range of lakhs, then the difference in thousands is not much.  
We can see during in the Evaluation of the Model that the accuracy of this model is about 94 percent.

* *Plottitng the Results –*
* ***Scatter Plot*** - We will plot the scatter plot between actual values and predicted values.

CODE:

‘plt.scatter(y\_test,y\_pred)

plt.xlabel('Actual')

plt.ylabel('Predicted')

plt.show()’

**Scatter Plot**



* ***Regression Plot*** - A regression plot is useful to understand the linear relationship between two parameters. It creates a regression line in-between those parameters and then plots a scatter plot of those data points.

CODE:

‘sns.regplot(x=y\_test,y=y\_pred,ci=None,color ='Green')’

**Regression Plot of our Model**



* ***Heatmap*** - A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colours.

CODE:

‘sns.heatmap(company.corr()) ‘

**Heatmap**



* *Evaluating the model (****R2 Value****)–*

We have different metrics to find the accuracy score of the model, and here we use r2\_score to evaluate our model and find its accuracy.

CODE:

‘ print(r2\_score(y\_test,y\_pred))

Accuracy=r2\_score(y\_test,y\_pred)\*100

print(" Accuracy of the model is %.2f" %Accuracy) ‘

OUTPUT :

................CALCULATING THE R SQUARED VALUE...................

0.9393955917820571

…........PREDICTING THE ACCURACY OF THE MODEL..........

Accuracy of the model is 93.94

**Conclusion**

(0.9393955917820571 is the R squared value)

**To be noted** : R squared value of 0.91 proves the model is a good model.

4% ͌ 94% proves that our model is a good model, with the predicted profit as :

Predicted Profit

0 103901.896970

1 132763.059931

2 133567.903700

3 72911.789767

4 179627.925672