Table of Contents

* Simple linear regression vs multiple linear regression
* Dataset
* Reading dataset
* Independent and dependent variables
* Handling categorical variables
* Splitting data
* Applying model

Simple Linear Regression Vs Multiple Linear Regression

Now, before moving ahead let discuss the interaction behind the simple linear regression then we try to compare multiple and simple linear regression based on that intuition we actually doing our machine learning problem.

Simple Linear Regression

We considered a simple linear regression in any machine learning algorithm using example,

Now, suppose if we take a scenario of house price where our x-axis is the size of the house and the y-axis is basically the price of the house. In this basically, we have two features first one is**f1** and the second one is **f2**, where,

**f1** refers to the size of the house and,

**f2**refers to the price of the house

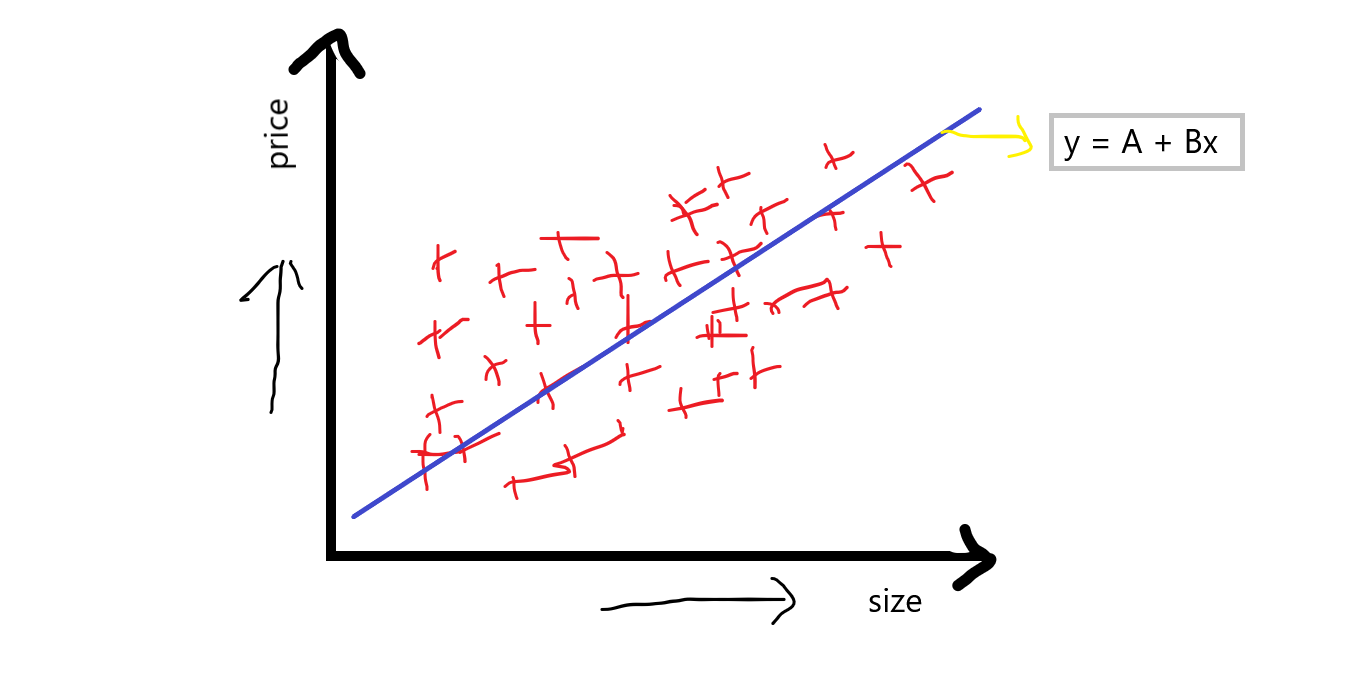
So, if **f1**becomes the independent feature and **f2** become the dependent feature, usually we know that whenever the size of house increases then price also increases, suppose we draw scatter points randomly, by this scatter point basically we try to find the best fit line and this best fit line is given by the equation:

**equation:   y = A + Bx**

|  |
| --- |
|  |

Suppose,**y**be the price of the house and**x** be the size of the house then this equation seems like this:

***equation:  price = A + B(size)*                             where,  
                                        A is an intercept and B is slop on that intercept**



When we discuss this equation, in which intercept basically, indicates the when the price of the house is 0 then what will be the base price of the house, and the slop or coefficient indicates that with the unit increases in size, then what will be the unit increases in slop.

Now, how is it different when compared to multiple linear regression?

Multiple Linear Regression

Multiple Linear Regression is basically indicating that we will be having many features Such as **f1**, **f2**,**f3**, **f4,** and our output feature **f5.**If we take the same example as above we discussed, suppose:

**f1**is the size of the house.

**f2** is bad rooms in the house.

**f3** is the locality of the house.

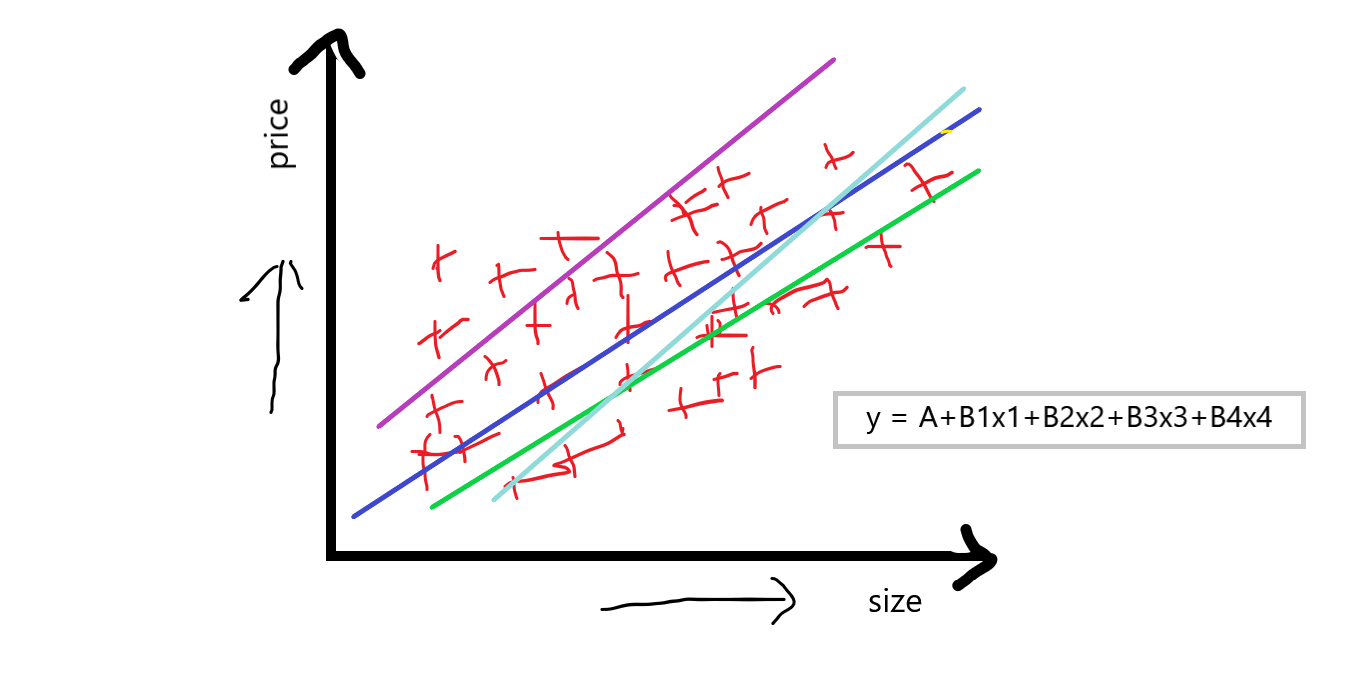
**f4** is the condition of the house and,

**f5**is our output feature which is the price of the house.

Now, you can see that multiple independent features also make a huge impact on the price of the house, price can vary from feature to feature. When we are discussing multiple linear regression then the equation of simple linear regression**y=A+Bx** is converted to something like:

**equation:  y = A+B1x1+B2x2+B3x3+B4x4**

“If we have one dependent feature and multiple independent features then basically call it a **multiple linear regression**.”



Now, our aim to using the multiple linear regression is that we have to compute **A**which is an intercept, and **B1B2B3B4**which are the slops or coefficient concerning this independent feature, that basically indicates that if we increase the value of**x1** by 1 unit then **B1** says that how much value it will affect int he price of the house, and this was similar concerning others **B2B3B4**

So, this is a small theoretical description of multiple linear regression now we will use the scikit learn linear regression library to solve the multiple linear regression problem.

Dataset

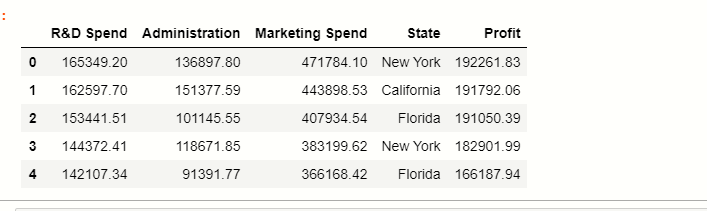
Now, we apply multiple linear regression on the **50\_startups** dataset, you can click [here](https://github.com/krishnaik06/Multiple-Linear-Regression) to download the dataset.

Reading dataset

Most of the dataset are in CSV file, for reading this file we use pandas library:

df = pd.read\_csv('50\_Startups.csv')

df



Here you can see that there are 5 columns in the dataset where the **state** stores the categorical data points, and the rest are numerical features.

Now, we have to classify independent and dependent features:

Independent and Dependent variables

There are total 5 features in the dataset, in which basically profit is our dependent feature, and the rest of them are our independent features:

#separate the other attributes from the predicting attribute

x = df.drop('Profit',axis=1)

#separte the predicting attribute into Y for model training

y = ['profit']

Handling categorical variables

In our dataset, there is one categorical column **State,**we have to handle this categorical values present inside this column for that we will use pandas**get\_dummies()** function:

# handle categorical variable

states=pd.get\_dummies(x,drop\_first=True)

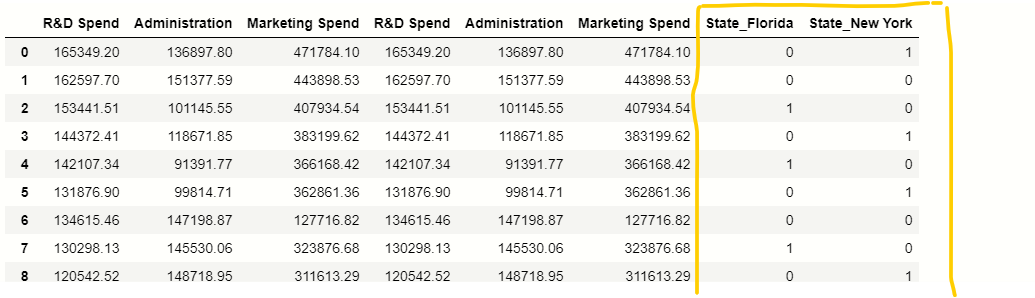
# dropping extra column

x= x.drop(‘State’,axis=1)

# concatation of independent variables and new cateorical variable.

x=pd.concat([x,states],axis=1)

x



Splitting Data

Now, we have to split the data into training and testing parts for that we use the scikit-learn **train\_test\_split()** function.

# importing train\_test\_split from sklearn

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

# splitting the data

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 42)

Applying model

Now, we apply the linear regression model to our training data, first of all, we have to import linear regression from the scikit-learn library, there is no other library to implement multiple linear regression we do it with linear regression only.

# importing module

from sklearn.linear\_model import LinearRegression

# creating an object of LinearRegression class

LR = LinearRegression()

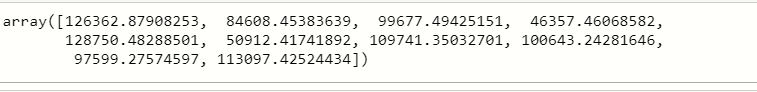
# fitting the training data

LR.fit(x\_train,y\_train)

finally, if we execute this then our model will be ready, now we have x\_test data we use this data for the prediction of **profit.**

y\_prediction =  LR.predict(x\_test)

y\_prediction



Now, we have to compare the y\_prediction values with the original values because we have to calculate the accuracy of our model, which was implemented by a concept called **r2\_score.**let’s discuss briefly on r2\_score:

**r2\_score:-**

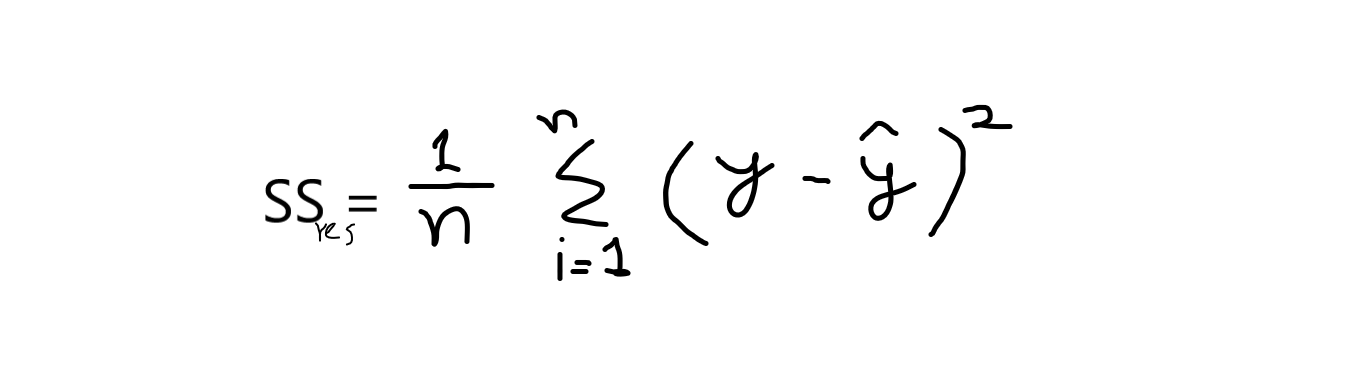
It is a function inside sklearn. metrics module, where the value of **r2\_score** varies between**0** and **100** percent,  we can say that it is closely related to MSE.

r2 is basically calculated by the formula given below:

**formula:  r2 = 1 – (SSres  /SSmean)**

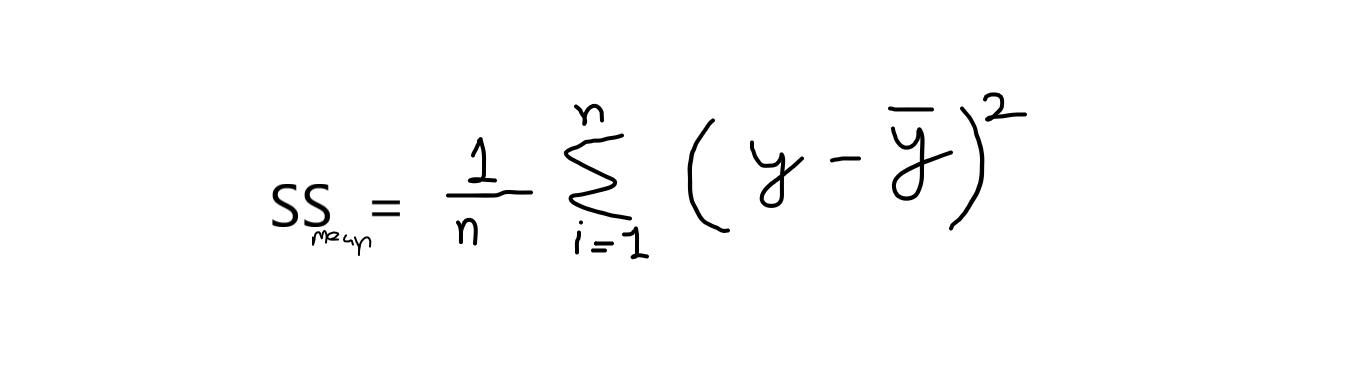
now, when I say **SSres**it means, it is the sum of residuals and **SSmean**refers to the sum of means.

where,



**y** = original values

**y^** = predicted values. and,



If we take calculation from this equation, then we have to know that the value of the sum of means is always greater than the sum of residuals. If this condition satisfies then our model is good for predictions. Its values range between 0.0 to 1.

*”The proportion of the variance in the dependent variable that is predictable from the independent variable(s).”*

The best possible score is 1.0 and it can be negative because the model can be arbitrarily worse. A constant model that always predicts the expected value of y, disregarding the input features, would get an R2 score of 0.0.

# importing r2\_score module

from sklearn.metrics import r2\_score

from sklearn.metrics import mean\_squared\_error

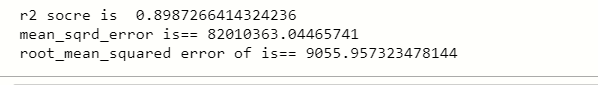
# predicting the accuracy score

score=r2\_score(y\_test,y\_prediction)

print(‘r2 socre is ‘,score)

print(‘mean\_sqrd\_error is==’,mean\_squared\_error(y\_test,y\_prediction))

print(‘root\_mean\_squared error of is==’,np.sqrt(mean\_squared\_error(y\_test,y\_prediction)))



You can see that the accuracy score is greater than 0.8 it means we can use this model to solve multiple linear regression, and also mean squared error rate is also low.

End Notes

hello, data scientists 😎 above we take a detailed discussion on multiple linear regression, and the example we used in it is the perfect example of multiple linear regression. I hope now you have a better understanding of multiple linear regression.

Hope you enjoyed this!

You can connect me on LinkedIn: [www.linkedin.com/in/mayur-badole-189221199](https://www.analyticsvidhya.com/blog/2021/05/multiple-linear-regression-using-python-and-scikit-learn/www.linkedin.com/in/mayur-badole-189221199)

Also, read my other articles:<https://www.analyticsvidhya.com/blog/author/mayurbadole2407/>

Thank You.