PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

A.SRI MAHALAKSHMI

(Reg. no. 39110025)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119. TAMIL NADU.

APRIL, 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR,RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A.Sri Mahalakshmi (Reg. No: 39110025)** who carried out the project entitled **"Prediction of House rent using multiple linear regression"** under my supervision from January 2022 to April 2022.

Internal Guide

Mrs. J. Refonaa, M.E.,(Ph.D)

Head of the Department

Dr. S.Vigneshwari, M.E., Ph.D, and Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva voice Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I, **A.Sri Mahalakshmi** hereby declare that the Professional Training report entitled **"Prediction of House rent using Multiple linear regression"** done by me under the guidance of **Mrs. J. Refonaa** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science anf Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. J. Refonaa** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

TRAINING CERTIFICATE

ABSTRACT

Usually, House rent index represents the summarized price changes of residential housing. While for house price prediction, it needs a more accurate method based on location, house type, size, build year, latitude, longitude, distance from nearest MRT and some other factors which could affect house demand and supply.

The primary aim is to use these machine learning techniques and curate them into ML model which can then serve the users. So that a Buyer can easily find his dream house with all the amenities he needed. Additionally, there exists a possibility of under pricing the product. If the price is predicted for these users this might help them get estates for their deserving prices not more nor less.

The experiment is done using Python as a programming language and numpy, pandas, matplotlib, sklearn as the libraries used to preprocess, visualize, and train the models for prediction.

KEY WORDS: prediction, multiple linear regression, machine learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
1 MLR	MULTIPLE LINEAR REGRESSION
2 HTTPS	HYPERTEXT TRANSFER PROTOCOL
3 ORG	ORGANIZATION
4 URL	UNIVERSAL RESOURCE LOCATOR

INTRODUCTION

Machine Learning (ML) is a vital aspect of present day business and research. It progressively improves the performance of computer systems by using algorithms and neural network models. Machine Learning algorithms automatically build a mathematical model using sample data also referred to as training data which form decisions without being specifically programmed to make those decisions

Real Estate Evaluation falls under supervised learning algorithm. Supervised Learning is again of two types Regression and Classification. As data set consists of continuous numerical values, Prediction is done through regression.

Regression analysis is a statistical technique for estimating the relationship among variables which have reason and result relation. Main focus of uni-variate regression is analyze the relationship between a dependent variable and one independent variable and formulates the linear relation equation between dependent and independent variable. Regression models with one dependent variable and more than one independent variables are called Multiple linear regression.

1.1 OUTLINE OF THE PROJECT

We are going to predict the house rent of unit area(Y) using transaction date(X1), house age(X2), distance to the nearest MRT station(X3), Number of convenience stores(X4), latitude(X5), longitude(X6) with the use of Multiple Linear Regression Algorithm.

1.2 LITERATURE REVIEW

NEED FOR HOUSE RENT PREDICTION

- ✓ Buyer can easily find his dream house with all the amenities he needed.
- ✓ Securing investment property now will thorough due diligence and watch investment pay off over the next few years.

ADVANTAGES OF MACHINE LEARNING OVER HOUSE RENT PREDICTION

- ✓ Trends and patterns are identified with ease.
- ✓ Machine Learning improves over Time.
- ✓ Machine Learning Lets to adapt without human intervention.
- ✓ Enables Automation.

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM

To perform multiple linear regression analysis on real estate valuation data set (collected from Sindian dist.., Taiwan) to predict house rent(Y).

2.2 SCOPE

Predicting house rents is expected to **help people who plan to buy a house** so they can know the price range in the future, then they can plan their finances well. In addition, house rents predictions are also beneficial for property investors to know the trend of housing prices in a certain location.

ADVANTAGES:

- Linear Regression model is simple to implement and easier to interpret the output coefficients.
- It is a very good algorithm for linearly separable data set.
- Over-fitting can be reduced by regularization.
- Linear Regression gives relationships between dependent and independent variables.

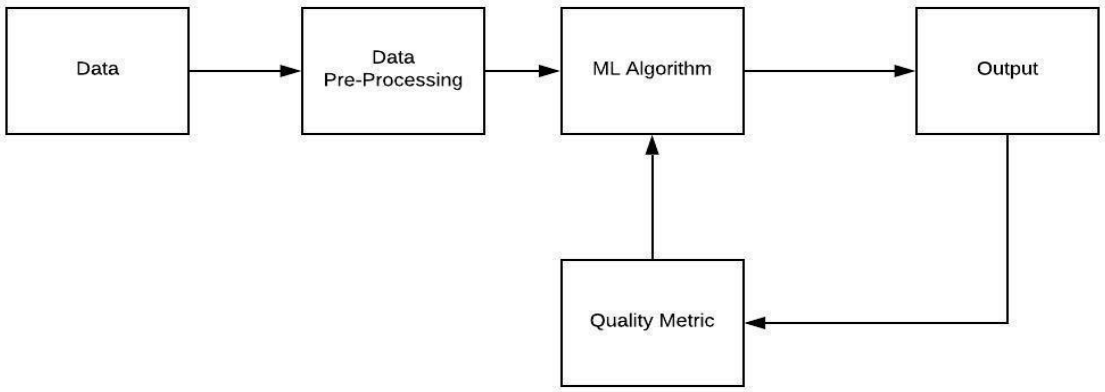
DISADVANTAGES:

- Prone to under-fitting.
- Sensitive to outliers.
- linear regression is not a complete description of relationships among variables.

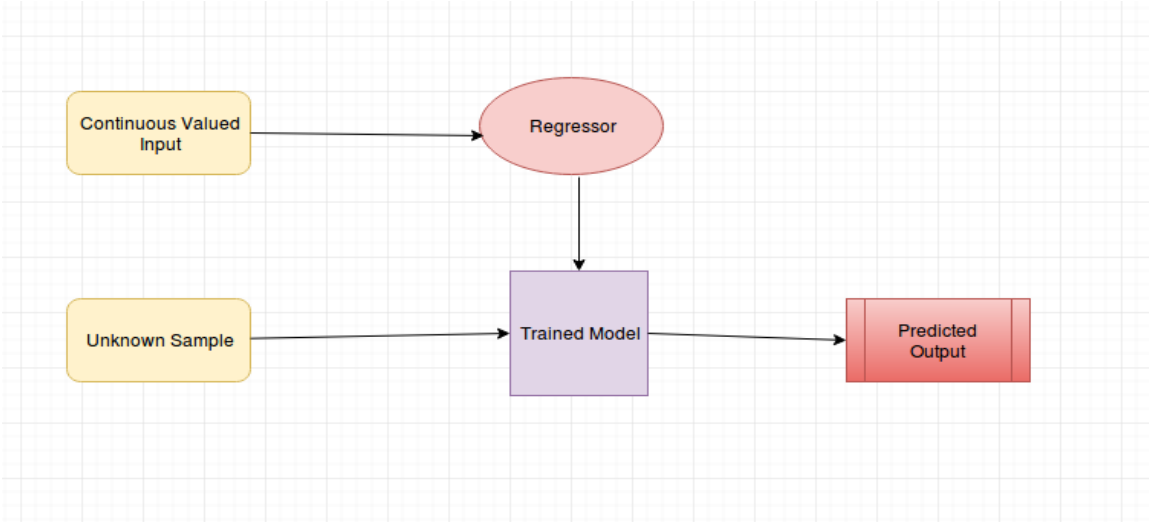
2.3 OBJECTIVE

The objective of multiple linear regression analysis is to use the independent variables(x) whose values are known to predict the value of the single dependent value(y).

2.4 SYSTEM ARCHITECTURE



DFD DIAGRAM:



EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

3.1 ALGORITHM

In Multiple Linear Regression, the target variable(Y) is a linear combination of multiple predictor variables x_1, x_2, x_3, ...,x_n. Since it is an enhancement of Simple Linear Regression, so the same is applied for the multiple linear regression equation, the equation becomes:

Multiple or multivariate linear regression is a case of linear regression with two or more independent variables.

x1,x2,...xp are independent variables

y is dependent variable

Formula and Calculation of Multiple Linear Regression:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

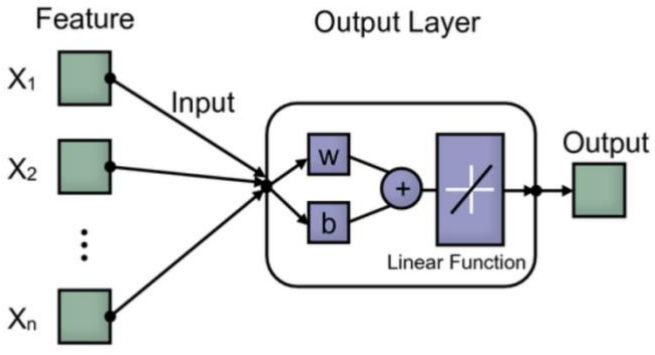
yi = dependent variable

xi = explanatory variables

β0 = y-intercept (constant term)

βp = slope coefficients for each explanatory variable

ε = The model's error term



3.2 EVALUATION METRICS OF LINEAR REGRESSION MODEL

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Lets look at MSE, MAE, R-squared, Adjusted R-squared, and RMSE.

Mean Squared Error (MSE)

The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

MSE penalizes large errors.

Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

MAE does not penalize large errors.

R-squared or Coefficient of Determination

This metric represents the part of the variance of the dependent variable explained by the independent variables of the model. It measures the strength of the relationship between the model and the dependent variable.
To understand what R-square really represents lets consider the following case where the error of the model with and without the knowledge of the independent variables.

- If R^2 is high (say 1), then the model represents the variance of the dependent variable.
- If R^2 is very low, then the model does not represent the variance of the dependent variable and regression is no better than taking the mean value, i.e. one is not using any information from the other variables.
- A Negative R^2 means one is doing worse than the mean value. It can have a negative value if the predictors do not explain the dependent variables at all such as RSS ~ TSS.

3.2 IMPLEMENTATION

To predict the house rent of unit area(Y) using transaction date(X1), house age(X2), distance to the nearest MRT station(X3), Number of convenience stores(X4), latitude(X5), longitude(X6) with the use of Multiple Linear Regression Algorithm.

3.2.1 METHODOLOGY

- **IMPORT LIBRARIES:**

Firstly import the library which will help in building the model

Pandas : Used for data wrangling and analysis.

Numpy: Stands for Numerical Python.

Matplotlib :Matplotlib is a plotting library for the Python programming language.

Sklearn :The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Linear Regressor : Performs the task to predict a dependent variable value (y) based on a given independent variable (x).

In [2]:
```
#import all requried libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from math import sqrt
```

In [3]:
```
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


from sklearn.metrics import mean_squared_error,r2_score,mean_absolute_error
```

❖ LOAD THE DATASET:

Import the data set using pandas library, which contains all the

variables

In [4]:
```
#load the dataset

df=pd.read_excel("./Real estate valuation data set copy.xlsx")
```

❖ CLEANING THE DATA:

Drop unnecessary columns from our data set.

In [10]:
```
#clean the dataset--drop unnecessary columns

df.drop('No',
  axis='columns', inplace=True)
```

In [11]:
```
df.head()
```

Out[11]:

	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	2012.916667	32.0	84.87882	10	24.98298	121.54024	37.9
1	2012.916667	19.5	306.59470	9	24.98034	121.53951	42.2
2	2013.583333	13.3	561.98450	5	24.98746	121.54391	47.3
3	2013.500000	13.3	561.98450	5	24.98746	121.54391	54.8
4	2012.833333	5.0	390.56840	5	24.97937	121.54245	43.1

❖ CORRELATION MATRIX:

Correlation matrix gives the co-variance of the given variables.

The correlation matrix shows the correlation of all predictors in pairs. The diagonal contains the variable names. The correlation of two variables are in the

intersection of the variables rows and columns. The color intensity of the boxes and numbers indicate the correlation degree: strong colors indicate high correlation, whereas light colors indicate low correlation.

cov(X, Y) = (sum (x - mean(X)) * (y - mean(Y))) * 1/(n-1)

Where,

X represents independent variables

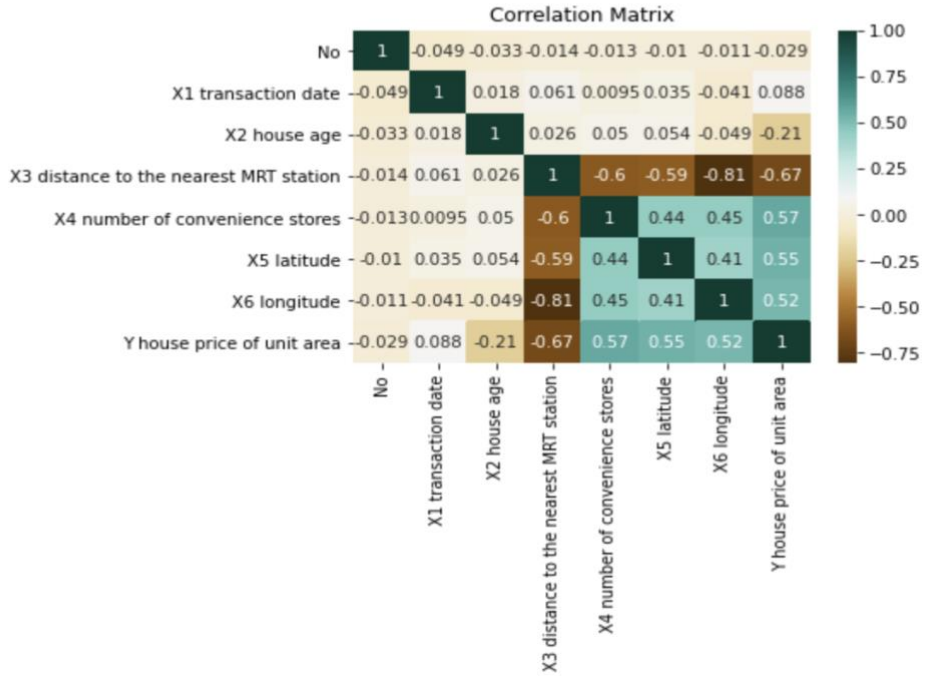
Y represents dependent variable

X-bar represents mean of x

Y-bar represents the meanof a dependent variable.

```
In [9]: #visualization of correlation matrix

c=df.corr()
sns.heatmap(c,cmap='BrBG',annot=True)
plt.title("Correlation Matrix")
plt.show()
```



Positive Correlation: both variables change in the same direction.

Neutral Correlation: No relationship in the change of the variables.

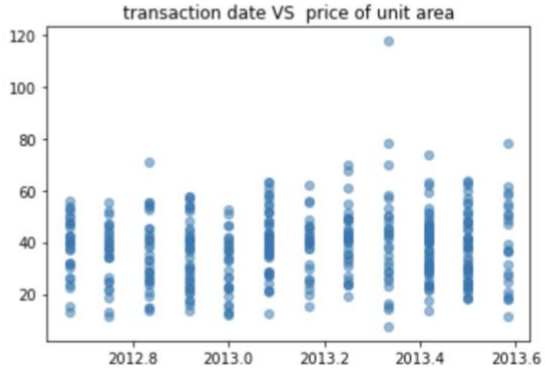
Negative Correlation: variables change in opposite directions.

if co-variance is '1' then they are highly correlated to each other and vice versa.

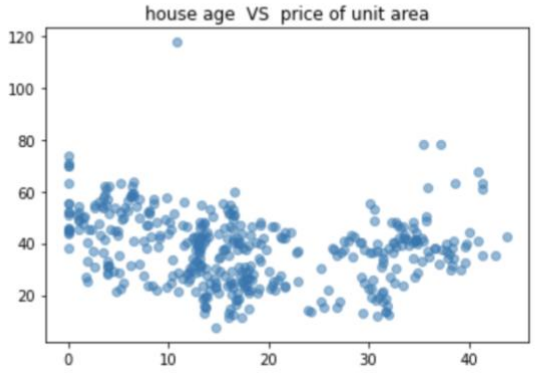
❖ DATA VISUALIZATION:

Visualizing the data using matplot library.Finding the relation between variables.

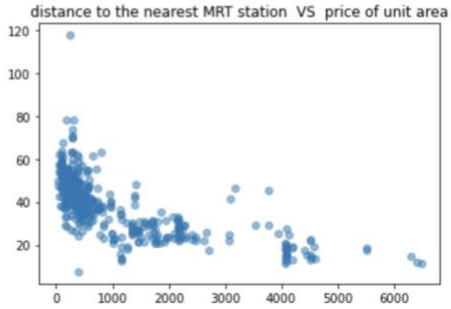
In [13]:
```
plt.scatter(df['X1 transaction date'],df['Y house price of unit area'],alpha=0.5)
plt.title('transaction date VS  price of unit area')
plt.show()
```



In [14]:
```
plt.scatter(df['X2 house age'],df['Y house price of unit area'],alpha=0.5)
plt.title('house age  VS  price of unit area')
plt.show()
```



In [15]:
```
plt.scatter(df['X3 distance to the nearest MRT station'],df['Y house price of unit area'],alpha=0.5)
plt.title('distance to the nearest MRT station  VS  price of unit area')
plt.show()
```



```
In [16]: plt.scatter(df['X4 number of convenience stores'],df['Y house price of unit area'],alpha=0.5)
         plt.title('number of convenience stores  VS  price of unit area')
         plt.show()
```

❖ Divide the dataset into independent and dependent variables:

Divide the dataset into two parts X and Y, X are multiple independent variables where Y is the one dependent variable

```
In [17]: #divide the dataset into independent and dependent variables
         #dependent variable is called target variable i.e 'Y'

In [18]: x = df.drop('Y house price of unit area',axis='columns').values
         y =df['Y house price of unit area'].values
```

❖ Splitting the Data into Training data and Testing data:

Split the dataset into two sets i.e. the Training set and the Testing set. Training set consists of 80% of the dataset and the testing set has 20% of the

dataset. The columns with only two distinct values and wanted to make sure that the splitting should split these values in equal proportions. Therefore, use a stratified shuffle split for train test splitting for better results.

```
In [21]: #using train test split divide x,y into training and testing data
         #they are divided in 80-20 rule

In [22]: X_train, X_test, y_train, y_test= train_test_split(x,y,test_size=0.3,random_state=0)

In [23]: reg=LinearRegression()

In [24]: #fit the linearModel

         reg.fit(X_train, y_train)

Out[24]: LinearRegression()
```

Using the train test split method, the dataset is divided into train sets and test sets. Train the model using a train set and for testing we use test data.

x_train,x_test,y_train,y_test = train_test_split (x , y , test_size=0.2 , random_state = 0)

Divide the total dataset into three subsets:

- Training data is used for learning the parameters of the model.

- Test data is used to evaluate the fit machine learning model

This splitting can prevent the model from over-fitting and to accurately evaluate our model

The random state parameter is used for initializing the internal random number generator,

which will decide the splitting of data into train and test.Setting random_state a fixed value

will guarantee that the same sequence of random numbers is generated each time.

❖ Fit the Linear Regression Model:

Using Linear regression module ,we fit our train dataset to our ML model

```
In [23]: reg=LinearRegression()

In [24]: #fit the linearModel

         reg.fit(X_train, y_train)

Out[24]: LinearRegression()
```

❖ Predict the output:

Finally we predict our y variable using LinearRegression.predict() method.

```
In [38]: #predict the values

         reg.predict([[2012.916667,32.0,84.87882,10,24.98298,121.54024]])

Out[38]: array([46.99125469])
```

❖ Get the Coefficients, Intercept:

In [30]:
```
# The coefficients
#y=ax1+b+e

print('Coefficients: \n', reg.coef_)

# The intercept
print('Intercept: \n', reg.intercept_)

# The mean squared error

print('Mean squared error: %.2f'
      % mean_squared_error(y_test, pred_y))

# mean absolue error

print('Mean absolute error: %.2f'
      % mean_absolute_error(y_test, pred_y))

# The coefficient of determination: 1 is perfect prediction

print('Coefficient of determination: %.2f'
      % r2_score(y_test, pred_y))
```

```
Coefficients:
 [ 5.21013721e+00 -2.75621110e-01 -4.59783392e-03  1.01886348e+00
  2.30116623e+02 -8.73223240e+00]
Intercept:
 -15129.240624837003
Mean squared error: 71.58
Mean absolute error: 6.06
Coefficient of determination: 0.58
```

❖ Apply metrics and get r2_score, mean absolute error and mean square error:

In [31]:
```
from sklearn import metrics
r_square=metrics.r2_score(y_test,pred_y)
r_square
```

Out[31]:
```
0.5800106026204446
```

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULT AND DISCUSSION:

Coefficient and intercept of our model.

1st coefficient β1 = 5.21013721e+00
2nd coefficient β2 = -2.75621110e-01
3rd coefficient β3 = -4.59783392e-03
4th coefficient β4 = 1.01886348e+00
5th coefficient β5 = 2.30116623e+02
6th coefficient β6 = -8.73223240e+00

Intercept of the model:

Intercept β0 = -15129.240624837003

Errors of the model:

Mean squared error = 71.58
Mean absolute error = 6.06
Coefficient of determination = R^2 = 0.58

In [30]:
```
# The coefficients
#y=ax1+b+e

print('Coefficients: \n', reg.coef_)

# The intercept
print('Intercept: \n', reg.intercept_)

# The mean squared error

print('Mean squared error: %.2f'
      % mean_squared_error(y_test, pred_y))

# mean absolue error

print('Mean absolute error: %.2f'
      % mean_absolute_error(y_test, pred_y))

# The coefficient of determination: 1 is perfect prediction

print('Coefficient of determination: %.2f'
      % r2_score(y_test, pred_y))
```

```
Coefficients:
 [ 5.21013721e+00 -2.75621110e-01 -4.59783392e-03  1.01886348e+00
  2.30116623e+02 -8.73223240e+00]
Intercept:
 -15129.240624837003
Mean squared error: 71.58
Mean absolute error: 6.06
Coefficient of determination: 0.58
```

4.2 PERFORMANCE ANALYSIS

By comparing the predicted value and provided value one can identify how exactly

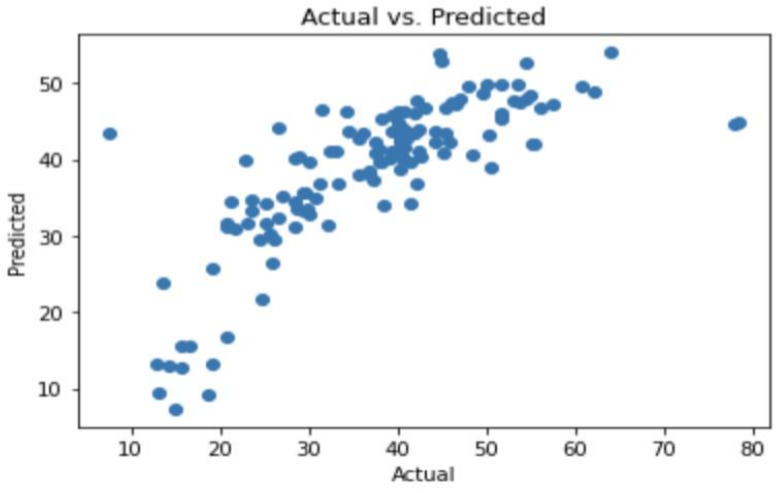
Is machinelearning working? and can also plot that difference between the test value and predicted value using distplot() in seaborn.

In statistics, the actual value is the value that is obtained by observation or by measuring the available data. It is also called the **observed value**. The predicted value is the value of the variable predicted based on the regression analysis.

Actual value=y_test

Predicted value= pred_y

```
In [29]: plt.scatter(y_test,pred_y)
         plt.xlabel('Actual')
         plt.ylabel('Predicted')
         plt.title('Actual vs. Predicted')
         plt.show()
```



Here actual value and predicted value were almost in the same linear line.So one say that this machine learning model is predicting accuratly.

```
In [40]: df=pd.DataFrame(data=y_test,columns=['y_test'])
         df['pred_y']=pred_y
         df['Difference']=y_test-pred_y
         df.head()
```

Out[40]:

	y_test	pred_y	Difference
0	45.3	40.848519	4.451481
1	14.4	12.949804	1.450196
2	46.0	42.167353	3.832647
3	15.6	12.744758	2.855242
4	50.2	43.273846	6.926154

Distplot()

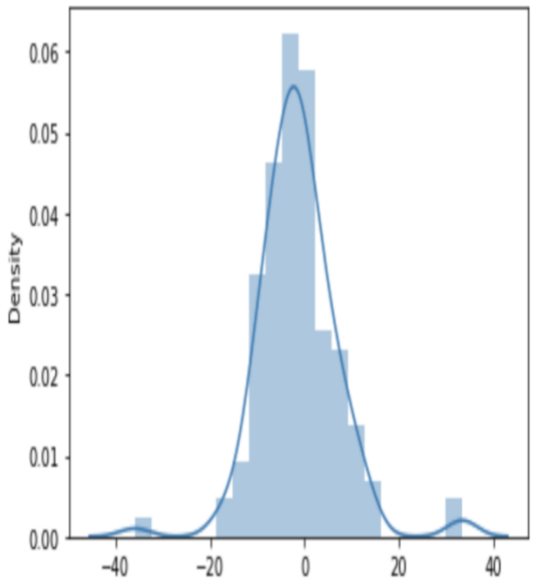
This function combines the matplotlib hist function (with automatic calculation of a good default bin size) with the seaborn kdeplot() and rug plot() functions. It can also

fit scipy.stats distributions and plot the estimated PDF over the data.

Distplot shows the variation between actual value and predicted value

```
[140]: import seaborn as sns
       a=y_test - pred_y
       sns.distplot(a,bins = 20)
```

```
t[140]: <AxesSubplot:ylabel='Density'>
```



Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are.

In other words, it tells how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

```
In [33]: def RMSE_score(y_actual, y_preds):
             return sqrt(mean_squared_error(y_actual, y_preds))
```

```
In [34]: print("Train score: ", RMSE_score(y_train, reg.predict(X_train)))
         print("Test score: ", RMSE_score(y_test, reg.predict(X_test)))
```

```
Train score:  8.937203354334068
Test score:  8.460281740972835
```

SUMMARY AND CONCLUSION

Finally the ML model is trained with the trained data and obtained the output

- Multiple linear regression is used to evaluate predictors for a continuously distributed outcome variable. The procedure calculates coefficients for each of the independent variables (predictors) that best agree with the observed data in the sample.
- The primary aim of this project is to use these machine learning techniques and curate them into ML model which can then serve the users.

The objective of a Buyer is to search for their dream house which has all the amenities they need.

- Finally build an ML model which can predict the price of a house with required parameters .And model got 58% of r2 score, which is a good model

The objective of predicting the last variable of the data set was achieved. The aim of predicting real estate house rent using machine learning algorithm is successfully completed.

REFERENCES

- Adichie, J. N. [1967], "Estimates of regression parameters based on rank tests," Ann. Math. Stat. 38, 894–904.
- Allen, D. M. [1971], "Mean square error of prediction as a criterion for selecting variables," Technometrics, 13, 469–475.
- Numpy reference;https://numpy.org/doc/1.20/user/index.html
- Matplotlib reference;https://matplotlib.org
- https://archive.ics.uci.edu/ml/datasets.php
- Python modules reference;https://pypi.org/
- https://docs.python.org/3/py-modindex.html
- Andrews, D. F. [1974], "A robust method for multiple linear regression," Technometrics, 16, 523–531.
- Andrews, D. F. [1979], "The robustness of residual displays," in R. L. Launer and G. N. Wilkinson (Eds.), Robustness in Statistics, Academic Press, New York, pp. 19–32.
- Andrews, D. F., P. J. Bickel, F. R. Hampel, P. J. Huber, W. H. Rogers, and J. W. Tukey [1972], Robust Estimates of Location, Princeton University Press,

  Princeton, N.J.
- Anscombe, F. J. [1961], "Examination of residuals," in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, University of California, Berkeley, pp. 1–34

APPENDIX

SCREENSHOT

```
%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from math import sqrt
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error,r2_score,mean_absolute_error

df=pd.read_excel(r"C:\Users\ramesh\Downloads\Real estate valuation data set.xlsx")
df.dtypes
df.info()
df.head()
df.describe()

c=df.corr()
sns.heatmap(c,cmap='BrBG',annot=True)
plt.title("Correlation Matrix")
plt.show()

df.drop('No',
  axis='columns', inplace=True)
df.head()

plt.scatter(df['X1 transaction date'],df['Y house price of unit area'],alpha=0.5)
plt.title('transaction date VS  price of unit area')
plt.show()

plt.scatter(df['X2 house age'],df['Y house price of unit area'],alpha=0.5)
plt.title('house age  VS  price of unit area')
plt.show()

plt.scatter(df['X3 distance to the nearest MRT station'],df['Y house price of unit area'],alpha=0.5)
plt.title('distance to the nearest MRT station  VS  price of unit area')
plt.show()

plt.scatter(df['X4 number of convenience stores'],df['Y house price of unit area'],alpha=0.5)
plt.title('number of convenience stores  VS  price of unit area')
plt.show()

x = df.drop('Y house price of unit area',axis='columns').values
y =df['Y house price of unit area'].values
x.shape
y.shape
X_train, X_test, y_train, y_test= train_test_split(x,y,test_size=0.3,random_state=0)

reg=LinearRegression()
reg.fit(X_train, y_train)

pred_y=reg.predict(X_test)
reg.predict([[2012.916667,32.0,84.87882,10,24.98298,121.54024]])

df=pd.DataFrame(data=y_test,columns=['y_test'])
df['pred_y']=pred_y
df.head()
plt.scatter(y_test,pred_y)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs. Predicted')
plt.show()

import seaborn as sns
sns.distplot((y_test - pred_y), bins = 20)

print('Coefficients: \n', reg.coef_)
print('Intercept: \n', reg.intercept_)
print('Mean squared error: %.2f'% mean_squared_error(y_test, pred_y))
print('Mean absolute error: %.2f'% mean_absolute_error(y_test, pred_y))
print('Coefficient of determination: %.2f'% r2_score(y_test, pred_y))

from sklearn import metrics
r_square=metrics.r2_score(y_test,pred_y)
r_square

def RMSE_score(y_actual, y_preds):
    return sqrt(mean_squared_error(y_actual, y_preds))

print("Train score: ", RMSE_score(y_train, reg.predict(X_train)))
print("Test score: ", RMSE_score(y_test, reg.predict(X_test)))
```

SOURCE CODE

%matplotlib inline

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from math import sqrt
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_
 absolute_error
df=pd.read_excel(".\Real estate valuation data set.xlsx")
df.dtypes
df.info()
df.head()
df.describe()
c=df.corr()
sns.heatmap(c,cmap='BrBG',annot=True)
plt.title("Correlation Matrix")
plt.show()
df.drop('No',axis='columns', inplace=True)
df.head()
```

```
plt.scatter(df['X1 transaction date'], df['Y house price of unit area'], alpha=0.5)

plt.title('transaction date VS  price of unit area')


plt.show()

plt.scatter['X1 transaction date'],df['Y house price of unit area'],alpha=0.5)

plt.title('transaction date VS  price of unit area')

plt.show()

plt.scatter['X1 transaction date'],df['Y house price of unit area'],alpha=0.5)

plt.title('transaction date VS  price of unit area')

plt.show()

plt.scatter(df['X4 number of convenience stores'],df['Y house price of unit area'],alpha=0.5)

plt.title('number of convenience stores  VS  price of unit area')

plt.show()

x=(df['X4 number of convenience stores'],df['Y house price of unit area'],alpha=0.5)


plt.title('number of convenience stores  VS  price of unit area')

plt.show()
```

```
x.shape

y.shape

X_train,X_test,y_train,y_test=train_test_split(x,y,test_size=0.3,rando
 m_state=0)


reg=LinearRegression()

reg.fit(X_train,y_train)

pred_y=reg.predict(X_test)

reg.predict([[2012.916667,32.0,84.87882,10,24.98298,121.54024]])

df=pd.DataFrame(data=y_test,columns=['y_test'])

df['pred_y']=pred_y

df.head()


plt.scatter(y_test,pred_y)

plt.xlabel('Actual')

plt.ylabel('Predicted')

plt.title('Actual vs. Predicted')

plt.show()

import seaborn as sns

sns.distplot((y_test - pred_y), bins = 20)

print('Coefficients: \n', reg.coef_)
```

```
print('Intercept: \n', reg.intercept_)

print('Mean squared error: %.2f% mean_squared_error(y_test, pred_y))

print('Mean absolute error: %.2f'% mean_absolute_error(y_test, pred_y))

print('Coefficient of determination: %.2f'% r2_score(y_test, pred_y))


from sklearn import metrics

r_square=metrics.r2_score(y_test,pred_y)

r_square

def RMSE_score(y_actual, y_preds):

    return sqrt(mean_squared_error(y_actual, y_preds))

print("Train score: ", RMSE_score(y_train, reg.predict(X_train)))

print("Test score: ", RMSE_score(y_test, reg.predict(X_test)))
```