**House Price Prediction Using Machine Learning Algorithms**

**Mini Project (CS705PC)**

**Submitted**

in partial fulfillment of the requirements for the award of the degree of

**Bachelor of Technology**

in

**Computer Science and Engineering**

by

**Mr. B. SAI SANKEERTH REDDY (17261A0504)**

Under the Guidance of

**Dr.C.R.K. Reddy**

**(Professor)**



**Department of Computer Science and Engineering**

**MAHATMA GANDHI INSTITUTE OF TECHNOLOGY**

**GANDIPET, HYDERABAD – 500 075, INDIA**

**2020-2021**

## MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

GANDIPET, HYDERABAD – 500 075. Telangana

**CERTIFICATE**



This is to certify that the thesis entitled **House Price Prediction Using Machine Learning Algorithms** is being submitted by **B. SAI SANKEERTH REDDY** in partial fulfillment for the award of **B. Tech** in **Computer Science and Engineering** to **Jawaharlal Nehru Technological University Hyderabad** is a record of bonafide work carried out by her under our guidance and supervision.

The results embodied in this project have not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |  |
| --- | --- | --- |
| Supervisor and coordinator |  | Principal |
| **Dr. C. R. K. Reddy** |  | **Dr .K. Jaya Sankar** |
| (Professor & HOD) |  | (Professor) |

**External Examiner**

**DECLARATION**

This is to certify that the work reported in this project titled “**House Price Prediction Using Machine Learning Algorithms”** is a record of work done by me in the **Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad.**

No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred in the text. The report is based on the work done entirely by me and not copied from any other source.

**B. SAI SANKEERTH REDDY(17261A0504)**

**ACKNOWLEDGEMENT**

I would like to express our sincere thanks to **Dr. K. Jaya Sankar, Principal, MGIT**, for providing the working facilities in college.

I wish to express our sincere thanks and gratitude to **Dr. C. R. K. Reddy, Professor and HOD**, Department of CSE, MGIT, for all the timely support and valuable suggestions during the period of project.

I am extremely thankful to **Dr. C. R. K. Reddy, Professor and B. Prashanthi, Associate Professor,** Department of CSE, MGIT, major project coordinators for their encouragement and support throughout the project.

I am extremely thankful and indebted to my internal guide **Dr. M. Rama Bai, Professor,** Department of CSE, for her constant guidance, encouragement and moral support throughout the project.

Finally, I would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, for completing this project.

**B. SAI SANKEERTH REDDY(17261A0504)**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Certificate | i |
| Declaration | ii |
| Acknowledgement | iii |
| List of Figures | vi |
| List of Tables | vii |
| Abstract | viii |
| **1. Introduction** | 1 |
| 1.1 Problem Definition | 1 |
| 1.2 Existing System | 2 |
| 1.3 Proposed System | 2 |
| 1.4 Requirements Specification | 3 |
| 1. 4. 1 Software Requirement | 3 |
| 1. 4. 2 Hardware Requirements | 3 |
| **2. Literature Survey** | 4 |
| **3. Design and Implementation of House Price Prediction** | 8 |
| 3.1 An approach for house price prediction  3.1.1 Type of Machine Learning | 8  9 |
| 3.2 Dataset | 9 |
| 3.3 Pre-processing Techniques | 11 |
| 3. 4 Training the Model | 17 |
| 3. 4.1 Multiple Linear Regression | 18 |
| 3. 4.2 Support Vector Regression | 19 |
| **4. Testing and Prediction**  4.1 Metrics Error  4. 1. 1 Metrics error used for MLR    4. 1. 2 Metrics error used for SVR | 21  21  22  22 |
| 4.2 Accuracy | 22 |
| 4. 2. 1 Accuracy for MLR | 23 |
| 4. 2. 2 Accuracy for SVR | 23 |
| **5.Conclusion and Future Scope**  **5. 1 Conclusion**    **5. 2 Future Scope**  **Bibliography**  **Appendix** | 24  24  24  25  26 |

**LIST OF FIGURES**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Figure 3.1 | System Architecture | 8 | | Figure 3.2 | Kaggle dataset used for house price prediction | 10 | | Figure 3.3 | Relation between independent attributes by pairing them before pre-processing | 11 | | Figure 3.4 | Code to import dataset in CSV file format | 13 | | Figure 3.5 | Handling missing data | 14 | | Figure 3.6 | Dividing the dataset into train and test data | 14 | | Figure 3.7 | Code for splitting train and test data | 15 | | Figure 3.8  Figure 3.9 | Formulas for Standardization and Normalization  Heatmap for the attributes by pairing them after pre-processing | 16  16 | | Figure 3.10  Figure 3.11 | Relation between independent attributes by pairing them before pre-processing  Multiple Linear Regression (MLR) formula | 17  18 | | Figure 3.12 | Train and test method | 18 | | Figure 3.13 | Importing SVR from sklearn | 19 | | Figure 3.14 | Equation of the hyperplane in SVR | 19 | | Figure 3.15 | Representation of Scatterplot for SVR | 20 | | Figure 4.1 | Metrics code | 21 | | Figure 4.2 | Metrics error for Multiple Linear Regression | 22 | |  |  |
| |  |  |  | | --- | --- | --- | | Figure 4.3 | Metrics error for Support vector Regression | 22 | | Figure 4.4  Figure 4.5 | Accuracy obtained for Multiple Linear Regression  Accuracy obtained for Support vector Regression | 23  23 | |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table 2.1 | Comparative Study of Existing Models | 7 |

**ABSTRACT**

This study uses two machine learning algorithms including, Multiple Linear Regression (MLR) and support vector machine (SVM) in the appraisal of house prices. It applies these methods to examine a data set and then compares the results of these algorithms. In terms of predictive power, MLR have achieved better performance when compared to SVM. The five performance regression metrics including explained variance, mean squared logarithmic error (MSLE), mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE) associated with these two algorithms also unambiguously outperform those of SVM.

However, our study has found that MLR is an effective machine learning algorithm and SVM is still a useful algorithm in data fitting because it can produce reasonably accurate predictions within a tight time constraint. Our conclusion is that machine learning offers a promising, alternative technique in property valuation and appraisal research especially in relation to house price prediction.

In this project I used the kaggle dataset which consists of about 12 columns and 1259 rows in which 6 of the features are numerical valued and rest are categorical, for training as well as for testing purpose.

**Keywords:** Multiple Linear regression (MLR), Support Vector Regression (SVR), explained variance, mean squared logarithmic error (MSLE), mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), regression, prediction, Machine Learning.

1. **INTRODUCTION**

House price prediction represents the summarized price changes of residential housing. While for a house price prediction, it needs more accurate methods based on location, house type, size, build year, local amenities, and some other factors which could affect house demand and supply. Prediction of house prices help people who plan to buy a house so they can know the house price in a certain area easily and quickly, then they can plan their finance well. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location.

This project uses Machine learning (ML) which is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Machine Learning contains many types of algorithms in which we implement Supervised learning where regression algorithms are used to predict. Regression is a machine learning tool that helps you make predictions by learning – from the existing statistical data – the relationships between your target parameter and a set of other parameters.

**1.1 Problem Definition**

The project “House Price Prediction Using Machine Learning Algorithms” is mainly based on predicting the house prices in different areas using regression technique in machine learning where various regression algorithms are implemented to predict the house prices. According to this definition, a house’s price depends on parameters such as the number of bedrooms, living area, location, etc. If we apply artificial learning to these parameters we can calculate house valuations in a given geographical area. The regression techniques used in the project are Multiple Linear Regression (MLR) and Support Vector Regression (SVR).

**1.2 Existing System**

The existing system used Logistic Regression, Support Vector Regression (SVR), Lasso Regression, Decision Tree [6] systems for predicting the house prices. Logistic Regression and Support Vector Regression (SVR) gives are baseline performance. Lasso Regression are powerful regression techniques, it works by penalizing the magnitude coefficients of features along with minimizing the error between predicted and actual observations. Lasso is called as L1

Regularization technique. Decision trees are considered to be the best and most widely used supervised learning algorithm. This model has the ability to predict the output with at most accuracy and stability. It is used to predict any kind of problems such as classification or regression.

**Disadvantages**

1. If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
2. Lasso has no way of distinguishing between a strong causal variable with predictive information and an associated high regression coefficient.
3. A small change in the data can result in a major change in the structure of the decision tree, which can convey a different result from what users will get in a normal event.
4. SVM algorithm is not suitable for large data sets.
5. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.

**1.3 Proposed System**

This research aims to create a house price prediction model that will help the users to check the house price in a certain area easily and quickly and gather information so that they can plan to buy a perfect house for them in a given budget, then they can plan their finance as well. In addition, house price predictions are also beneficial for property investors to know the trend of housing prices in a certain location. The project uses the regression techniques such as Multiple Linear Regression (MLR) and Support Vector Regression (SVR). to find different quality attributes such as accuracy, sensitivity, precision and specificity to know the performance.

**Advantages**

1. Multiple Linear Regression (MLR) helps us to understand the relationships among variables present in the dataset.
2. It helps us to understand the relationships among variables present in the dataset.
3. The advantages of support vector regression include that they can be used to avoid the difficulties of using linear functions in the high-dimensional feature space.
4. Support Vector Regression (SVR) is relatively memory efficient

**1.4 Requirements Specification**

**1.4.1 Software Requirements**

1. Operating System: Windows 10
2. Software: Jupyter Nootbook, PyCharm, Google Colaboratory

**1.4.2 Hardware Requirements**

1. RAM: 8 GB
2. Hard Disk: 120 GB
3. Memory: 64 GB
4. Input Devices: Keyboard, Mouse

**2. LITERATURE SURVEY**

In our literature survey we have investigated various researches on this particular domain some of them are as follows :-

**1) Housing Price Prediction via Improved Machine Learning Techniques (Quang Truong, Minh Nguyen, Hy Dang, Bo Mei)** [2], Commonly, House Price Index (HPI) is used to measure price changes of residential housing in many countries, such as the US Federal Housing Finance Agency HP, UK National Statistics HPI, UKL and Registry’s HPI and Singapore URA HPI. The HPI is a weighted, repeat sales index, meaning that it measures average price changes in repeat sales or refinancing on the same properties. With some analytical tools, it allows housing economists to estimate changes in the rates of mortgage defaults, prepayments, and housing affordability in specific geographic areas. Because HPI is a rough indicator calculated from all transactions, it is inefficient topredict the price of a specific house. Many features such as district, age, and the number of floors also need to beconsidered instead of just the repeat sales in previous decades. In recent years, due to the growing trend towards Big Data, machine learning has become a vital prediction approach because it can predict house prices more accurately based on their attributes, regardless of the data from previous years.

**2) Machine Learning: Implementing various regression algorithms to predict Boston house prices (Khushwant Rai)** [4], various regression algorithms are implemented to predict the Boston house prices. The Boston Housing dataset comprises data collected by the US consensus Service regarding various factors affecting the price of owner-occupied houses in the Boston area. The factors viz per capita crime rate, closeness to Charles river, nitric oxide concentration, number of rooms per house, accessibility to highways, tax, lower-status people percentage, etc. are considered to depict the prices and these prices are represented in $1000’s. The prices of the houses will be predicted, therefore, our target or dependent variable is the price of the house and relevant features or independent variables will be chosen from all factors affecting price, while the data exploration process. Clearly, it is a regression problem because continuous values are required to be predicted so, various regression algorithms are implemented to achieve the same.

**3) Predicting property prices with machine learning algorithms (Winky K.O. Ho, Bo-Sin Tang, Siu Wai Wong)** [1], Support vector machine (SVM), random forest (RF) and gradient boosting machine (GBM) in the appraisal of property prices. It applies these methods to examine a data sample of about 40,000 housing transactions in a period of over 18 years in Hong Kong, and then compares the results of these algorithms. In terms of predictive power, RF and GBM have achieved better performance when compared to SVM. The three performance metrics including mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) associated with these two algorithms also unambiguously outperform those of SVM. However, our study has found that SVM is still a useful algorithm in data fitting because it can produce reasonably accurate predictions within a tight time constraint.

**4)** **Utilization of Machine Learning Models in Real Estate House Price Prediction (Anurag Sinha)** [3], Machine learning participate a significant role in every single area of technology as per the today’s scenario. Even I can Say every phase of our lives is surrounded by the implementation of new era technologies such as Hospitality management, Railway, Transportation, Health care, Industry. Machine learning is used as an algorithm for building

model and thereby using that model for predicting new data set. The prominent difference of using conventional algorithm that result is oriented with the input data rather than focusing on a chain of different instructions set. Supervised learning is based on building a model based on

labeled data set whereas Unsupervised learning is totally oriented with unlabeled data set. There are several machine learning algorithms are regression, classification, clustering, SVM, neural network, deep learning and so on.

**5) House Price Prediction Using Machine Learning and Neural Networks (Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair)** [5], Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation. Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price. We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give most accurate results. The results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied. We also propose to use real-time neighborhood details using Google maps to get exact real-world valuations.

**6) Valuation of House Prices Using Predictive Techniques (Neelam Shinde, Kiran Gawande)** [6], Housing sales price are determined by numerous factors such as area of the property, location of the house, material used for construction, age of the property, number of bedrooms and garages and so on. This paper uses machine learning algorithms to build the prediction model for houses. Here, machine learning algorithms such as logistic regression and support vector regression, Lasso Regression technique and Decision Tree are employed to build a predictive model. We have considered housing data of 3000 properties. Logistic Regression, SVM, Lasso Regression and Decision Tree show the R-squared value of 0.98, 0.96,0.81 and 0.99 respectively. Further, we have compared these algorithms based on parameters such as MAE,

MSE, RMSE and accuracy. This paper also represents significance of our approach and the methodology.

**Table 2.1** Comparative Study of Existing Models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SNO | YEAR | AUTHOR | TITLE | ADVANTAGES | DISADVANTAGES | |
| 1 | 2020 | Quang Truong, Minh Nguyen,  Hy Dang,  Bo Mei | Housing Price Prediction via Improved Machine Learning Techniques | It reduces overfitting in decision trees and improves the accuracy | It also requires much time for training as it combines a lot of decision trees to determine the class. | |
| 2 | 2020 | Khushwant Rai | Machine Learning: Implementing various regression algorithms to predict Boston house prices | Using Many regression techniques helps to understand the problem better . | Need more memory to process the techniques separately and it’s time consuming. | |
| 3 | 2019 | Winky K.O. Ho, Bo-Sin Tang, Siu Wai Wong | Predicting property prices with machine learning algorithms | K-fold cross-validation is a technique used for making sure that our model is well trained, without using the test set. | They are unstable, meaning that a small change in the data can lead to a large change in the structure of the optimal decision tree. | |
| 4 | 2018 | Anurag Sinha | Utilization of Machine Learning Models in Real Estate House Price Prediction | It is easy to understand and calculations are very fast. | Cannot be used for complex mixture samples | |
| 5 | 2018 | Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair | House Price Prediction Using Machine Learning and Neural Networks | Used to simplify gradient image. | Reconstruction of the morphological gradient . | |
| 6 | 2018 | Neelam Shinde, Kiran Gawande | Valuation of House Prices Using Predictive Techniques | Some powerful and efficient regression algorithms are used for enhancing the accuracy. | | A small change in the data can result in a major change in the structure of the decision tree. |

**3. DESIGN AND IMPLEMENTATION OF HOUSE PRICE PREDICTION**

**3.1 An approach for house price prediction**

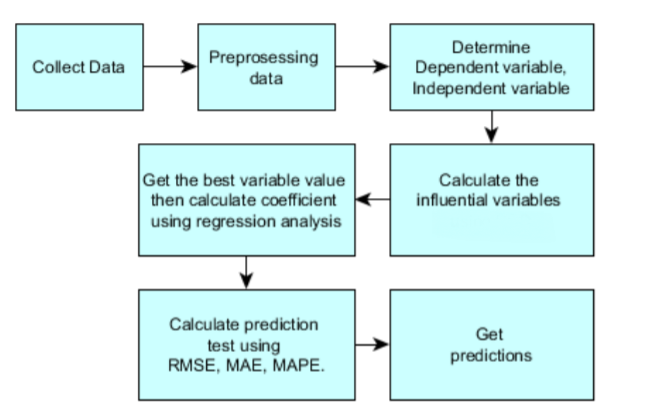


Figure 3.1: System Architecture

The above figure 3.1 represents the steps in regression techniques in Machine Learning (ML). Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning.

**3.1.1 Types of Machine Learning**

There are three types of Machine Learning algorithms, they are:

* Supervised Learning
* Unsupervised Learning
* Reinforcement Learning

In this project we use Supervise Learning. We apply supervised ML techniques when we have a piece of data that we want to predict or explain. We do so by using previous data of inputs and outputs to predict an output based on a new input. For example, you could use supervised ML techniques to help a service business that wants to predict.

Regression methods fall within the category of supervised ML. They help to predict or explain a particular numerical value based on a set of prior data, for example predicting the price of a property based on previous pricing data for similar properties. The simplest method is linear regression where we use the mathematical equation of the line (y = m \* x + b) to model a data set. We train a linear regression model with many data pairs (x, y) by calculating the position and slope of a line that minimizes the total distance between all of the data points and the line. In other words, we calculate the slope (m) and the y-intercept (b) for a line that best approximates the observations in the data.

In Supervise Learning there are many regression techniques in which the implemented methods are Multiple Linear Regression (MLR) and Support Vector Regression (SVR).

**3.2 DataSet**

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. kaggle dataset which consists of about 12 columns and 1259 rows in which 6 of the features are numerical valued and rest are categorical consisting parameters such as the number of bedrooms, living area, location, etc., for training as well as for testing purpose.

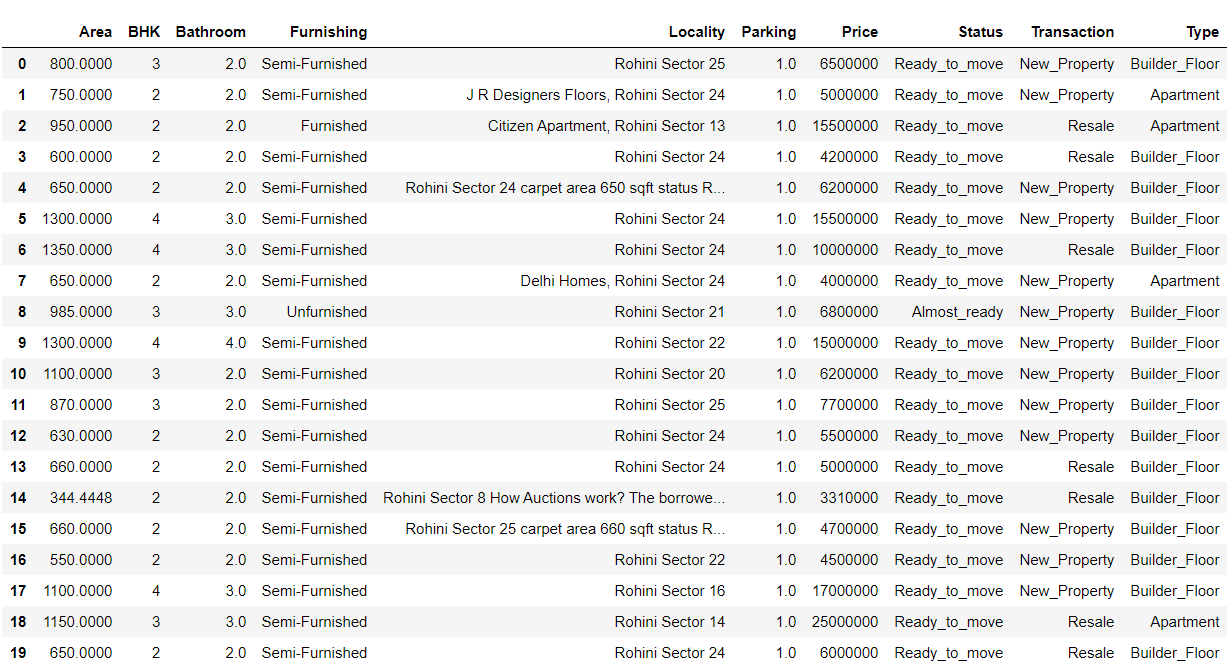


Figure 3.2: Kaggle dataset used for house price prediction

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with Pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them. Seaborn library is imported for displaying the paired relations between two independent variables.

The dataset relation between independent attributes by pairing them before pre-processing is shown below in figure 3.3.

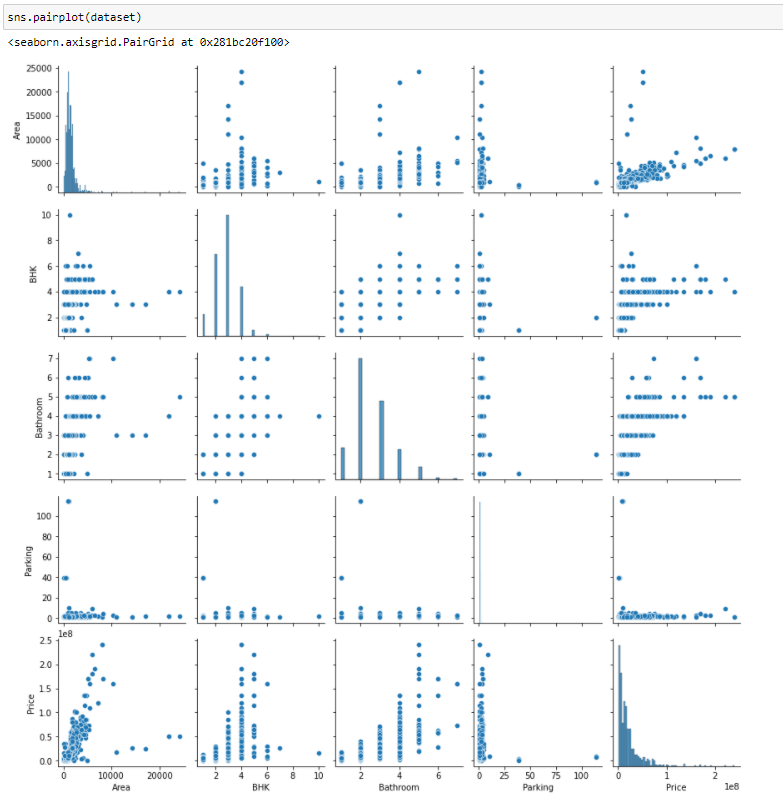


Figure 3.3: Relation between independent attributes by pairing them before pre-processing.

**3.3 Pre-processing**

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

**Getting the dataset**

Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV file. However, sometimes, we may also need to use an HTML or xlsx file.

**Importing libraries**

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

* Numpy
* Pandas
* Matplotlib
* Seaborn

To import the dataset, we will use read\_csv() function of pandas library, which is used to read a csv file and performs various operations on it. Using this function, we can read a csv file locally as well as through an URL or from a folder.

The figure 3.4 is the representation of the code which can be used to import the dataset in CSV file format using Numpy and Pandas libraries

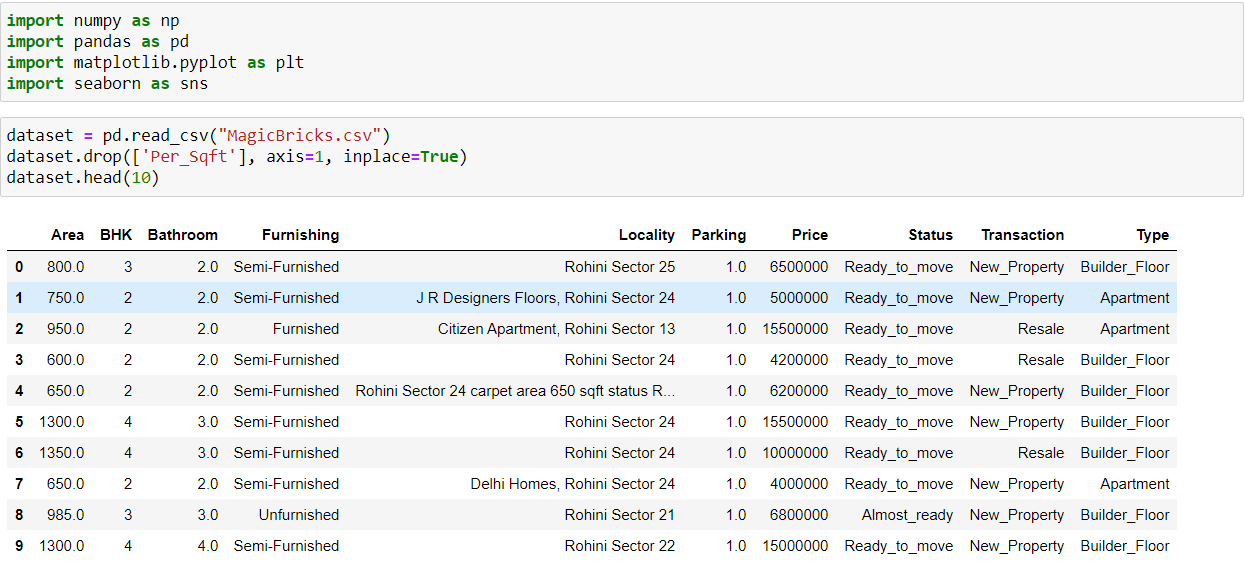


Figure 3.4: Code to import dataset in CSV file format

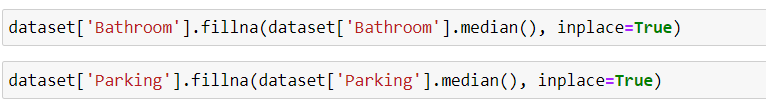
**Handling Missing Data and Noisy data**

If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

There are mainly two ways to handle missing data, which are:

* Deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.
* Calculating the mean or median: In this way, we will calculate the mean or median of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach as shown in figure 3.5.

Seaborn library is imported to use to plot graphs to find them and then Pandas library is used for handling the noisy data.

  
**Figure 3.5:** Handling missing data

**Encoding Categorical Data**

In our case, there are five country variables, and as we can see in the above output, these variables are encoded into 0, 1, and 2. By these values, the machine learning model may assume that there is some correlation between these variables which will produce the wrong output. So to remove this issue, we will use dummy encoding. Dummy variables are those variables which have values 0 or 1. The 1 value gives the presence of that variable in a particular column, and rest variables become 0. With dummy encoding, we will have a number of columns equal to the number of categories.

**Splitting dataset into training and test set**

In machine learning data preprocessing, dividing our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model, we divide the dataset train and test data to 8:2 ratio as shown in figure 3.6. Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

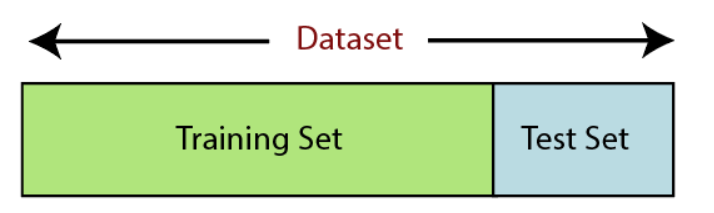


figure 3.6: Dividing the dataset train and test data to 8:2

Train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset as shown in figure 3.7. Here, we can define these datasets as:

* Training Set: A subset of dataset to train the machine learning model, and we already know the output.
* Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

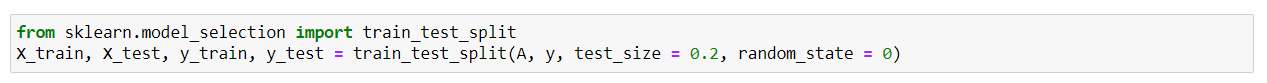


Figure 3.7: Code for splitting test and train data.

* x\_train: features for the training data
* x\_test: features for testing data
* y\_train: Dependent variables for training data
* y\_test: Independent variable for testing data

Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Some of the regression techniques and methods that can be imported using sci kit-learn are:

* StandardScaler
* LogisticRegression
* metrics
* train\_test\_split
* metrics

**Feature scaling**

Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. There are two methods in feature scaling Standardization and Normalization as shown in figure 3.7 in which we use Normalization.

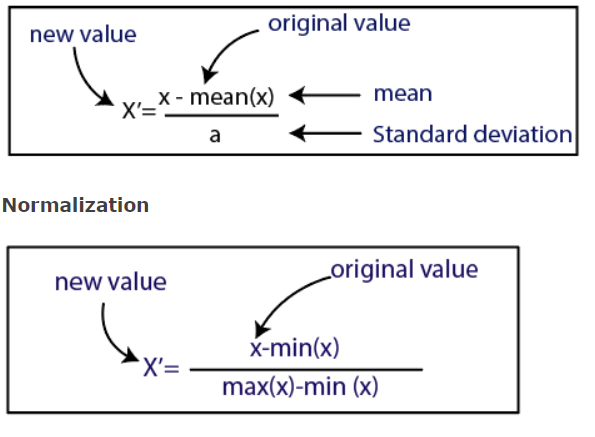
****

figure 3.8: formulas for Standardization and Normalization

The dataset relation between independent attributes by pairing them after pre-processing is shown below in figure 3.9 and figure 3.10.

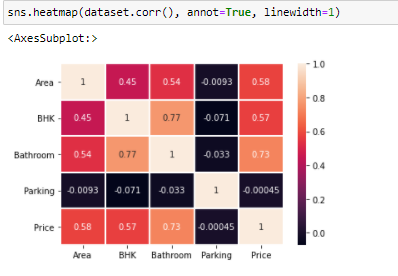


figure 3.9: Heatmap for the attributes by pairing them after pre-processing.



Figure 3.10: Relation between independent attributes by pairing them after pre-processing.

**3.4 Training the Model**

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

**3.4.1 Multiple Linear Regression**

It is also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables as represented in figure 3.11(also known as dependent and independent variables). The case of one explanatory variable is called simple linear regression; for more than one, the process is called multiple linear regression [4]. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

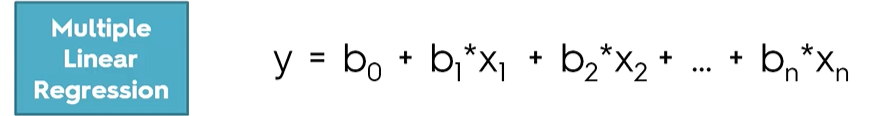
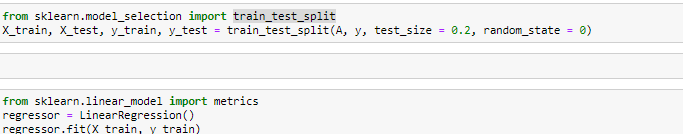


figure 3.11: Multiple Linear Regression (MLR) formula.

The above figure represents Multiple Linear Regression (MLR) formula where ‘x’ is an independent variable, ‘y’ is a dependent variable which is dependent on ‘x’ and ‘beta’ is the regression coefficients as shown in figure 3.12.





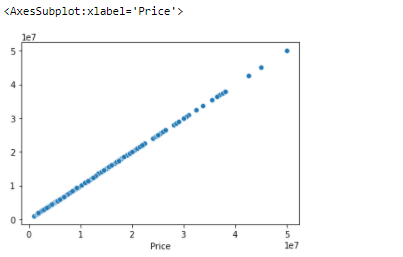


figure 3.12: Train and test method

**3.4.2 Support Vector Regression**

support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Support Vector Machine can also be used as a regression technique. It uses the same principles as that of support vector machine classifier. However, in Linear Support Vector Regression it estimates a function that maximizes the deviation from the actual target ‘Y’ within normalized margin strip, by keeping the function as flat as possible [6]. We use sci kit-learn library to import SVR method as shown in figure 3.13.

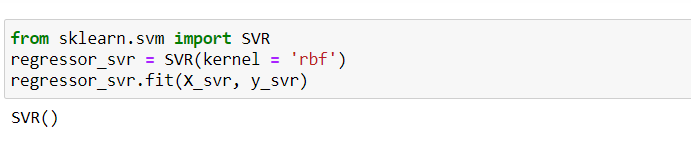


Figure 3.13: Importing SVR from sklearn

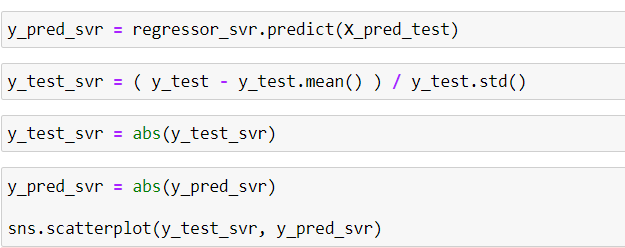
The support vector regression is a very powerful algorithm to solve linear and non-linear regression problems. In support vector regression, the decision boundary is formed within a range of hyperparameter ε which acts as threshold and SVR tries to include as many data points as possible inside this decision boundary. The SVR makes sure that error lies within a certain threshold. The vectors that are nearest to boundary lines are called support vectors. The kernels function in SVR help to work with non-linearly separable data without need of especially transforming data to new z-space. The Hyperplane or line inside the decision boundary makes the new predictions and equation of the hyperplane is shown in figure 3.14.



figure 3.14: Equation of the hyperplane in SVR

StandardScaler is imported from sklear library. The idea behind StandardScaler is that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1.

Scatterplot graph is plotted using Seaborn library to gather the information of the test set and predicted set in Support Vector Regression as shown below in figure 3.15.



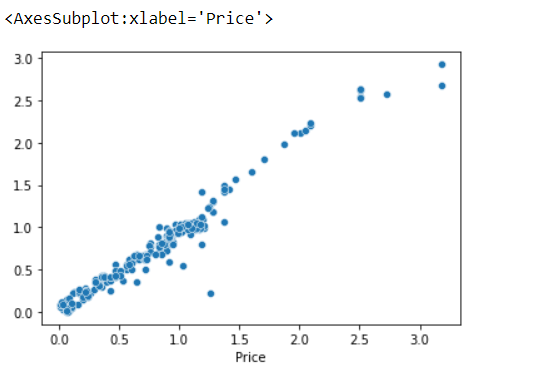


Figure 3.15: Representation of scatterplot for SVR

**4. TESTING AND PREDICTION**

Testing is a very important module in the software development to verify, validate and provide quality and service for different components of software. It is used to minimize the risks by efficient use of resources in the development life cycle. This module can be employed at any point of the development process. It is efficient for the testing phase to be implemented at initial level to get good accuracy.

**4.1 Metrics Error**

Different performance metrics are used to evaluate different Machine Learning Algorithms. For now, we will be focusing on the ones used for Classification problems. We can use classification performance metrics such as variance, mean squared logarithmic error (MSLE), mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE) which is represented in below figure 4.1.

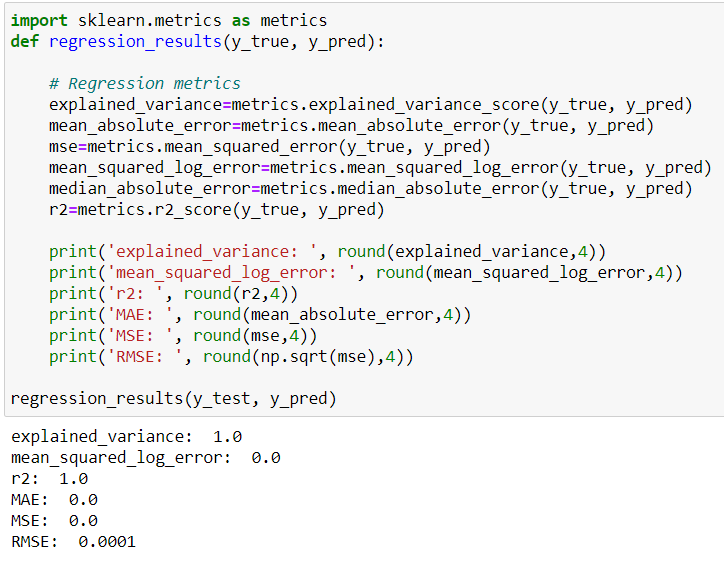


figure 4.1: Metrics code

* + 1. **Metrics error used for Multiple Linear Regression**

Metrics error for Multiple Linear Regression (MLR) is calculated is shown in figure 4.2.

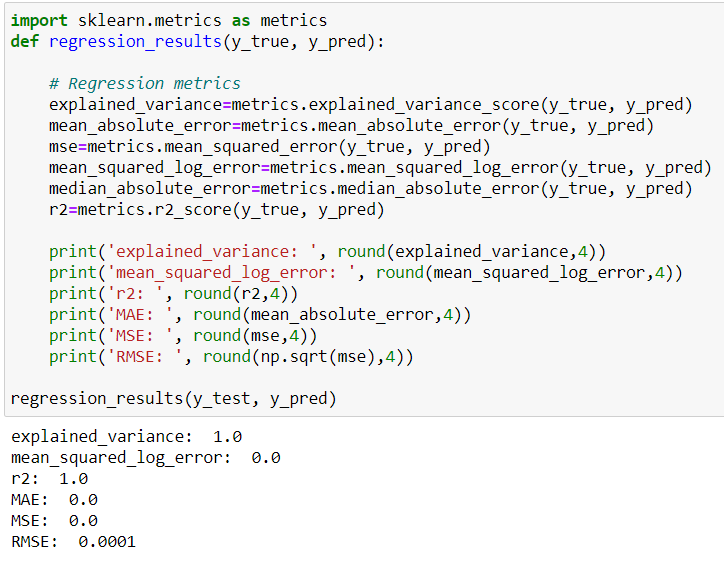


figure 4.2: Metrics error for Multiple Linear Regression

* + 1. **Metrics error used for Support Vector Regression**

Metrics error for Support Vector Regression (SVR) is calculated is shown in figure 4.3.

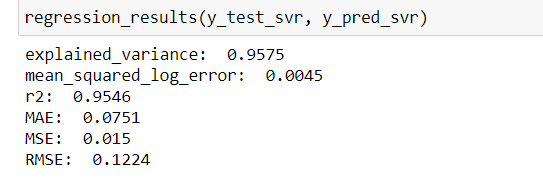


figure 4.3: Metrics error for Support Vector Regression

* 1. **Accuracy**

The accuracy is calculated using r2\_score by importing this method from metrics. The coefficient of determination, denoted R2 or r2 and pronounced “R squared”, is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

* + 1. **Accuracy for Multiple Linear Regression**

Accuracy obtained for Multiple Linear Regression (MLR) is (1.0) or 100% as represented in figure 4.4 for the given dataset after preprocessing and dividing it into test set and train set.

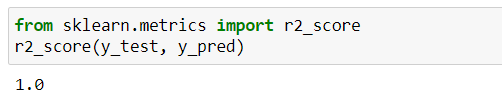


figure 4.4: Accuracy obtained for Multiple Linear Regression

* + 1. **Accuracy for Support Vector Regression**

Accuracy obtained for Support Vector Regression (SVR) is (0.6713) or 67.13% as represented in figure 4.5 for the given dataset after preprocessing and dividing it into test set and train set.

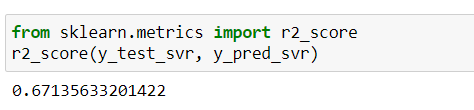


figure 4.5: Accuracy obtained for Support Vector Regression

**5. CONCLUSION AND FUTURE SCOPE**

**5.1 Conclusion**

The project “House Price Prediction using Machine Learning Algorithms” is used to we have used machine learning algorithms to predict the house prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the parameters. Thus we can select the parameters which are not correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a csv file was generated consisting of predicted house prices. Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. We found that Multiple Linear Regression overfits our dataset and gives the highest accuracy of 100%. Support Vector Regression giving an accuracy of 67.13% respectively Thus we conclude that we implemented classifiers to the problem of regression to check how well can classifier fit to regression problem.

**5.2 Future Scope**

In the near future, the system working on large dataset would yield a better and real picture about the model. We have undertaken only few Machine Learning algorithms that are actually classifiers but we need to train many other classifiers and understand their predicting behavior for continuous values too. By improving the error values this research work can be useful for development of applications for various respective cities.

**BIBLIOGRAPHY**

[1] Winky K.O. Ho, Bo-Sin Tang, Siu Wai Wong, “Housing Price Prediction via Improved Machine Learning Techniques”, 2020, Journal of machine learning research.

[2] Khushwant Rai, “Machine Learning: Implementing various regression algorithms to predict Boston house prices”, 2020.

[3] K.O. Ho, Bo-Sin Tang, Siu Wai Wong, “Predicting property prices with machine learning algorithms”, 2019, Amity Journal of Computational Sciences (AJCS) .

[4] Anurag Sinha, “Utilization of Machine Learning Models in Real Estate House Price Prediction”, 2018.

[5] Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair, “House Price Prediction Using Machine Learning and Neural Networks”, 2018, International Conference on Inventive Communication and Computational Technologies (ICICCT).

[6]Neelam Shinde, Kiran Gawande, “Valuation of House Prices Using Predictive Techniques”, 2018, International Journal of Advances in Electronics and Computer Science

**APPENDIX**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

dataset = pd.read\_csv("MagicBricks.csv")

dataset.drop(['Per\_Sqft'], axis=1, inplace=True)

dataset.head()

dataset.dtypes

dataset['Price'] = dataset['Price'].astype('float64')

dataset.describe()

sns.boxplot(dataset['Area'])

sns.boxplot(dataset['BHK'])

sns.boxplot(dataset['price'])

sns.boxplot(dataset['bathroom'])

sns.pairplot(dataset)

sns.heatmap(dataset.corr(), annot=True, linewidth=1)

sns.boxplot(dataset['BHK'], dataset['Area'])

sns.boxplot(dataset['Parking'], dataset['Area'])

dataset.head()

pd.crosstab(dataset['Furnishing'], dataset['Status'], normalize='index', margins=True)

pd.crosstab(dataset['Furnishing'], dataset['Status'], normalize='index', margins=True).plot.bar()

pd.crosstab(dataset['Bathroom'], dataset['Status'], normalize='index', margins=True).plot.bar()

dataset.isnull().sum()

dataset.head()

dataset['Bathroom'].fillna(dataset['Bathroom'].median(), inplace=True)

dataset['Parking'].fillna(dataset['Parking'].median(), inplace=True)

dataset.isnull().sum()

dataset[dataset['Furnishing'].isnull() | dataset['Type'].isnull()]null\_data

null\_data

dataset.dropna(axis=0, inplace=True)

dataset.shape

dataset.isnull().any()

dataset.describe()

sns.boxplot(dataset['Area'])

Q1 = dataset['Area'].quantile(0.25)

Q3 = dataset['Area'].quantile(0.75)

IQR = Q3 - Q1 #IQR is interquartile range.

filter = (dataset['Area'] >= Q1 - 1.5 \* IQR) & (dataset['Area'] <= Q3 + 1.5 \*IQR) & (dataset['Area'] <= 10000)

dataset= dataset.loc[filter]

new\_dataset = dataset.loc[filter]

new\_dataset

sns.boxplot(new\_dataset['Area'])

sns.scatterplot(new\_dataset['Area'], new\_dataset['Price'])

new\_dataset.describe()

sns.scatterplot(new\_dataset['Area'], new\_dataset['Price'])

sns.boxplot(new\_dataset['Price'])

dataset.isnull().sum()

sns.pairplot(new\_dataset)

dataset.head()

#X = new\_dataset.drop(['Price'], axis=1)

X = new\_dataset.drop(['Locality'], axis=1)

X.head()

y = new\_dataset['Price']

y.head()

A = pd.get\_dummies(new\_dataset, drop\_first=True, sparse=True)

A.head()

new\_dataset['Locality'].unique().shape

print(new\_dataset['Furnishing'].unique())

print(new\_dataset['Status'].unique())

print(new\_dataset['Transaction'].unique())

print(new\_dataset['Type'].unique())

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(A, y, test\_size = 0.2, random\_state = 0)

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(X\_train, y\_train)

y\_pred = regressor.predict(X\_test)

sns.scatterplot(y\_test, y\_pred)

regressor.intercept\_

regressor.coef\_

import sklearn.metrics as metrics

def regression\_results(y\_true, y\_pred):

# Regression metrics

explained\_variance=metrics.explained\_variance\_score(y\_true, y\_pred)

mean\_absolute\_error=metrics.mean\_absolute\_error(y\_true, y\_pred)

mse=metrics.mean\_squared\_error(y\_true, y\_pred)

mean\_squared\_log\_error=metrics.mean\_squared\_log\_error(y\_true, y\_pred)

median\_absolute\_error=metrics.median\_absolute\_error(y\_true, y\_pred)

r2=metrics.r2\_score(y\_true, y\_pred)

print('explained\_variance: ', round(explained\_variance,4))

print('mean\_squared\_log\_error: ', round(mean\_squared\_log\_error,4))

print('r2: ', round(r2,4))

print('MAE: ', round(mean\_absolute\_error,4))

print('MSE: ', round(mse,4))

print('RMSE: ', round(np.sqrt(mse),4))

regression\_results(y\_test, y\_pred)

from sklearn.metrics import r2\_score

r2\_score(y\_test, y\_pred)

X.head()

from sklearn.preprocessing import StandardScaler

sc\_X = StandardScaler()

sc\_y = StandardScaler()

X\_svr = sc\_X.fit\_transform(X\_train)

X\_svr

type(y)

y\_train.std()

y\_svr = ( y\_train - y\_train.mean() ) / y\_train.std()

y\_svr

from sklearn.svm import SVR

regressor\_svr = SVR(kernel = 'rbf')

regressor\_svr.fit(X\_svr, y\_svr)

sc\_X = StandardScaler()

X\_pred\_test = sc\_X.fit\_transform(X\_test)

y\_pred\_svr = regressor\_svr.predict(X\_pred\_test)

y\_test\_svr = ( y\_test - y\_test.mean() ) / y\_test.std()

y\_test\_svr = abs(y\_test\_svr)

y\_pred\_svr = abs(y\_pred\_svr)

sns.scatterplot(y\_test\_svr, y\_pred\_svr)

import sklearn.metrics as metrics

def regression\_results(y\_true, y\_pred):

# Regression metrics

explained\_variance=metrics.explained\_variance\_score(y\_true, y\_pred)

mean\_absolute\_error=metrics.mean\_absolute\_error(y\_true, y\_pred)

mse=metrics.mean\_squared\_error(y\_true, y\_pred)

mean\_squared\_log\_error=metrics.mean\_squared\_log\_error(y\_true, y\_pred)

median\_absolute\_error=metrics.median\_absolute\_error(y\_true, y\_pred)

r2=metrics.r2\_score(y\_true, y\_pred)

print('explained\_variance: ', round(explained\_variance,4))

print('mean\_squared\_log\_error: ', round(mean\_squared\_log\_error,4))

print('r2: ', round(r2,4))

print('MAE: ', round(mean\_absolute\_error,4))

print('MSE: ', round(mse,4))

print('RMSE: ', round(np.sqrt(mse),4))

regression\_results(y\_test\_svr, y\_pred\_svr)

from sklearn.metrics import r2\_score

r2\_s(y\_test\_svr, y\_pred\_svr)