**HOUSE SALES PRICE PREDICTION**

***TABLE OF CONTENTS***

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **INTRODUCTION** |  |
|  | * 1. General Introduction |  |
|  | * 1. Project Objectives |  |
|  | * 1. Problem Statement |  |
|  | **SYSTEM** **PROPOSAL** |  |
|  | * 1. Existing System |  |
|  | * + 1. Disadvantages |  |
|  | * 1. Proposed System |  |
|  | 2.2.1 Advantages |  |
|  | * 1. Literature Survey |  |
|  | **SYSTEM DIAGRAMS** |  |
|  | * 1. Architecture Diagram |  |
|  | * 1. Flow Diagram |  |
|  | * 1. UML Diagrams |  |
|  | **IMPLEMENTATION** |  |
|  | * 1. Modules |  |
|  | * 1. Modules Description |  |
|  | **SYSTEM** **REQUIREMENTS** |  |
|  | * 1. Hardware Requirements |  |
|  | * 1. Software Requirements |  |
|  | * 1. Software Description |  |
|  | * 1. Testing of Products |  |
|  | **CONCLUSION** |  |
| **7.** | **FUTURE** **ENHANCEMENT** |  |
| **8.** | **SAMPLE** **CODING** |  |
| **9.** | **SAMPLE** **SCREENSHOT** |  |
| **10.** | **REFERENCES** |  |

**ABSTRACT**

Predictive models for determining the sale price of houses in cities is still remaining as more challenging and tricky task. The sale price of properties in cities like Bengaluru depends on a number of interdependent factors. Key factors that might affect the price include area of the property, location of the property and its amenities. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine. We have made an attempt to build house price prediction regression model for data set that remains accessible to the public in Machine platform. Modelling uses machine learning algorithms, where machine learns from the data and uses them to predict a new data. The most frequently used model for predictive analysis is regression. Different groups of buyers may focus on different house attributes. The purpose is to create a model that can estimate housing prices. The input dataset was taken from the dataset repository. Then, we have to implement the pre-processing techniques.. We have to implement the regression technique to predict the housing price such as Ridge and lasso regression. Finally, the experimental results shows that the error values, estimate the house price in Lakhs.

**CHAPTER 1**

**INTRODUCTION**

* 1. **General Introduction:**

Viruses are microscopic organisms that replicate only inside the living cells of an organism. Coronaviruses are one of the most common families of viruses that cause various respiratory diseases in living organisms. Forecasting the spread of coronavirus is one of the challenges in recent times. Forecasting the pandemic with high accuracy will help different countries prepare a plan to fight a war against the virus spread. Machine learning techniques are extensively used for modeling real-world problems. Specifically, applications of machine learning techniques to predict diseases have been used extensively in recent times.

Some of the traditional methods used in forecasting of an epidemic include time series modeling as well as regression modeling. We have chosen some of the traditional as well as modern machine learning techniques. Support Vector Machine technique has been traditionally used for predictions while we have compared some of these traditional techniques with a modern technique. Since December 2019, an outburst of global pandemic Corona Virus disease (COVID-19) started in Wuhan (China) where 2873 deaths were initially reported in China only, and 104 deaths were stated outside the country. It caused the death rate to 3.6% and 1.5%, respectively, till February 2020.

Although it originated from China, its quick spread and mortality rate were observed in other areas globally, especially in the US, Italy, U.K., and Spain. By March 1, 2020, around 80 000 patients were reported in China, and 7200 patients were noticed in the rest of the world. In China, the coronavirus’s flare-up started, but, by March 15, 2020, the number of infected people surpassed Europe and the USA.

The number of mortalities in a few regions presently exceeds the mortality count in China. By the end of April 2020, positive carriers were more than three million worldwide, with more than 210 000 mortalities. COVID-19 early indications start with any one or combination of more than one symptoms, including temperature, cough, diarrhea, sore throat, high respiratory rate, low oxygen saturation, or shortness of breathing.

In later stages, this may lead to acute respiratory distress syndrome. Since December 2019, the virus has been recognized in millions of people around the world and killed thousands of people. According to research and World Health Organizations’ findings, COVID-19 overspread overwhelmingly by respiratory beads, but it can also spread from hard surfaces.

These asymptomatic cases are the primary cause of their outspread. Corona Virus has spread its roots well deep and could be very dangerous due to its increasing mortality rate. In a few cases, symptom onset might result in death due to intense alveolar damage and respiration failure.

* 1. **Objectives:**

The main objective of our project is,

* To predict the house price based on different locations.
* To implement the machine learning algorithm.
* To enhance the overall performance for classification algorithms.

**CHAPTER 2**

**SYSTEM PROPOSAL**

* 1. **EXISTING SYSTEM:**

In existing, Predictive models for determining the sale price of houses in cities like Bengaluru is still remaining as more challenging and tricky task. The sale price of properties in cities like Bengaluru depends on a number of interdependent factors. Key factors that might affect the price include area of the property, location of the property and its amenities. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine hackathon platform. The data set has nine features. In this study, an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price. Modelling explorations apply some regression techniques such as multiple linear regression (Least Squares), Lasso and Ridge regression models, support vector regression, and boosting algorithms such as Extreme Gradient Boost Regression (XG Boost). Such models are used to build a predictive model, and to pick the best performing model by performing a comparative analysis on the predictive errors obtained between these models. Here, the attempt is to construct a predictive model for evaluating the price based on factors that affects the price.

**2.1.1 DISADVANTAGES:**

* The results is low when compared with proposed
* It doesn’t efficient for large volume of data’s
* Theoretical limits.
  1. **PROPOSED SYSTEM:**

In this system, the house dataset was taken as input from the dataset repository. Then, we have to implement the data pre-processing step. In this step, we have to handle the missing values for avoid wrong prediction. Then, we have to split the data into test and train. In this step, test is used for predict the model and train is used for evaluate the model.we have to implement the machine learning regression algorithms such as Ridge and lasso regression .Finally, the experimental results shows that the performance metrics such as MAE and predict the house price based on input attributes.

**2.2.1 ADVANTAGES:**

* It is efficient for large number of datasets.
* The experimental result is high when compared with existing system.
* The prediction is efficient
* The process is implemented with removing unwanted data.

**2.3 LITERATURE SURVEY:**

# **2.3.1House Price Prediction Using Regression Techniques: A Comparative Study, 2019**

# ***Author:*** CH.Raga Madhuri, Anuradha G, M.Vani Pujitha

***Methodology*:**

People are careful when they are trying to buy a new house with their budgets and market strategies. The objective of the paper is to forecast the coherent house prices for non-house holders based on their financial provisions and their aspirations. By analysing the foregoing merchandise, fare ranges and also forewarns developments, speculated prices will be estimated. The paper involves predictions using different Regression techniques like Multiple linear, Ridge, LASSO, Elastic Net, Gradient boosting and Ada Boost Regression. House price prediction on a data set has been done by using all the above mentioned techniques to find out the best among them. The motive of this paper is to help the seller to estimate the selling cost of a house perfectly and to help people to predict the exact time slap to accumulate a house. Some of the related factors that impact the cost were also taken into considerations such as physical conditions, concept and location etc.

***Advantage:***

* The algorithm has high result value when compared to all the other algorithms regarding house price predictions.
* The best possible prediction of house prices by using appropriate algorithms and finding out which among them is best suitable for predicting the price with low error rate.

# **2.3.2House Price Prediction Using Machine Learning and Neural Networks, 2020**

# ***Author*:** Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair

# ***Methodology*:**

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Our dataset comprises of various essential parameters and data mining has been at the root of our system. We initially cleaned up our entire dataset and also truncated the outlier values. 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 neighbourhood details using Google maps to get exact real-world valuations.

**Advantage**:

* The data mining process in such an industry provides an advantage to the developers by processing those data, forecasting future trends and thus assisting them to make favorable knowledge-driven decisions.

# **2.3.3 Prediction of House Pricing Using Machine Learning with Python, 2020**

# ***Author*:** Mansi Jain, Himani Rajput, Neha Garg, Pronika Chawla

***Methodology*:**

This paper provides an overview about how to predict house costs utilizing different regression methods with the assistance of python libraries. The proposed technique considered the more refined aspects used for the calculation of house price and provide the more accurate prediction. It also provides a brief about various graphical and numerical techniques which will be required to predict the price of a house. This paper contains what and how the house pricing model works with the help of machine learning and which dataset is used in our proposed model. This project is proposed to predict house prices and to get better and accurate results. The stacking algorithm is applied on various regression algorithms to see which algorithm has the most accurate and precise results. This would be of great help to the people because the house pricing ids a topic that concerns a lot of citizens whether rich or middle class as one can never judge or estimate the pricing of a house on the basis of locality or facilities available. To accomplish this task, the python programming language is used. Python is a high level programming language for general purpose programming.

***Advantage:***

* K-Fold Cross Validation technique splits the dataset into n number of subsets. At that point, it has been attempted for preparing on the entirety of the subsets however leave one (k-1) subset for the assessment of the prepared model. This strategy emphasizes k times with an alternate subset turned around for the preparation reason each time.

# **2.3.4 Housing Price Prediction via Improved Machine Learning Techniques, 2019**

# ***Author***: Quang Truong, Minh Nguyen, Hy Dang, Bo Mei

# ***Methodology*:**

House Price Index (HPI) is commonly used to estimate the changes in housing price. Since housing price is strongly correlated to other factors such as location, area, population, it requires other information apart from HPI to predict individual housing price. There has been a considerably large number of papers adopting traditional machine learning approaches to predict housing prices accurately, but they rarely concern about the performance of individual models and neglect the less popular yet complex models. As a result, to explore various impacts of features on prediction methods, this paper will apply both traditional and advanced machine learning approaches to investigate the difference among several advanced models. This paper will also comprehensively validate multiple techniques in model implementation on regression and provide an optimistic result for housing price prediction.

***Advantage***:

* The Random Forest method has the lowest error on the training set but is prone to be over fitting. Its time complexity is high since the dataset has to be fit multiple times.
* The model performed with a high accuracy where the RMSLE of the training set is around 0.16118.

# **2.3.5 Deep Learning Model for House Price Prediction Using Heterogeneous Data Analysis Along With Joint Self-Attention Mechanism, 2021**

***Author*:** pei-ying wang1, chiao-ting chen 2, jain-wun su1, ting-yun wang1, and szu-hao huang

***Methodology***:

Our proposed model differs from self-attention models because it considers the interaction between two different features to learn the complicated relationship between features in order to increase prediction precision. We conduct experiments to demonstrate the performance of the model. Experimental data include actual selling prices in real estate transaction data for the period from 2017 to 2018, public facility data acquired from the Taipei and New Taipei governments, and satellite maps crawled using the Google Maps application programming interface. We utilize these datasets to train our proposed and compare its performance with that of other machine learning-based models such as Extreme Gradient Boosting and Light Gradient Boosted Machine, deep learning, and several attention models. The experimental results indicate that the proposed model achieves a low prediction error and outperforms the other models. To the best of our knowledge, we are the first research to incorporate attention mechanism and STN network to conduct house price prediction.

***Advantage***:

* We import heterogeneous data comprising Google satellite maps and public facilities to prove that data on house transactions alone are inadequate for achieving high prediction performance.
* The MAPEs derived for some regions, such as Ruifang, are high.

# **2.3.6 Machine Learning based Predicting House Prices using Regression Techniques, 2020**

***Author*:** Manasa J, Radha Gupta, Narahari N S

***Methodology***:

Predictive models for determining the sale price of houses in cities like Bengaluru is still remaining as more challenging and tricky task. The sale price of properties in cities like Bengaluru depends on a number of interdependent factors. Key factors that might affect the price include area of the property, location of the property and its amenities. In this research work, an analytical study has been carried out by considering the data set that remains open to the public by illustrating the available housing properties in machine hackathon platform. The data set has nine features. In this study, an attempt has been made to construct a predictive model for evaluating the price based on the factors that affect the price. Modelling explorations apply some regression techniques such as multiple linear regression (Least Squares), Lasso and Ridge regression models, support vector regression, and boosting algorithms such as Extreme Gradient Boost Regression (XG Boost). Such models are used to build a predictive model, and to pick the best performing model by performing a comparative analysis on the predictive errors obtained between these models. Here, the attempt is to construct a predictive model for evaluating the price based on factors that affects the price.

***Disadvantage:***

* But in case of multiple linear regressions, R-squared value may increase with increasing features even though the model is not actually improving. A related, Adjusted R-squared statistic can be used to address this disadvantage.

**2.3.7 House price forecasting using machine learning, 2020**

***Author*:** alisha kuvalekar, shivani manchewar, sidhika mahadik.

***Methodology***:

The real estate market is a standout amongst the most focused regarding pricing and keeps fluctuating. It is one of the prime fields to apply the ideas of machine learning on how to enhance and foresee the costs with high accuracy. The objective of the paper is the prediction of the market value of a real estate property. This system helps find a starting price for a property based on the geographical variables. By breaking down past market patterns and value ranges, and coming advancements future costs will be anticipated. This examination means to predict house prices in Mumbai city with Decision tree repressor. It will help clients to put resources into a bequest without moving towards a broker. The result of this research proved that the Decision tree repressor gives an accuracy of 89%.

***Advantage:***

* Predicting the house prices, we have included two other features - air quality and crime rate. These features provide a valuable contribution towards predicting property prices since the higher values of these features will lead to a reduction in house prices.
* The decision tree repressor provided the highest accuracy in terms of predicting the house prices.

**CHAPTER 3**

**SYSTEM DIAGRAMS**

**3.1 SYSTEM ARCHITECTURE:**

Input data

Preprocessing

Data Splitting

Classification

Performance Analysis

Data set (House dataset)

Handle missing value

Lasso

Ridge Regressor

Label Encoding

Test

Drop unwanted columns

Train

MAE

Predict the house price

***FIGURE 3.1: SYSTEM ARCHITECTURE***

**3.2 FLOW DIAGRAM**

Input Data

Preprocessing

Data splitting

Classification

Performance analysis

***FIGURE 3.2: FLOW DIAGRAM***

**3.3 UML DIAGRAMS:**

**3.3.1 USE CASE DIAGRAM:**

System

User

***FIGURE 3.3.1: USE CASE DIAGRAM***

**3.3.2 USE CASE DIAGRAM:**

Input Data

Preprocessing

Data Splitting

Performance Analysis

Classification

***FIGURE 3.3.2: ACTIVITY CASE DIAGRAM***

**3.3.3 SEQUENCE DIAGRAM:**

Input Data

Preprocessing

Classification

Performance Analysis

Select data

Missing value

Lasso

Load data

Data splitting

Ridge

***FIGURE 3.3.3: SEQUENCE DIAGRAM***

**3.3.4 ER DIAGRAM:**

Data selection

Preprocessing

Data Splitting

Classification

***FIGURE 3.3.4: ER DIAGRAM***

**3.3.6 CLASS DIAGRAM:**

Select data ()

Load data ()

View data ()

INPUT

Test ()

Ridge ()

Train ()

Classification

Prediction ()

Lasso ()

Performance analysis

Preprocessing

Missing values ()

Label encode ()

Data Splitting

MAE ()

***FIGURE 3.3.5: CLASS DIAGRAM***

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 MODULES:**

* Data selection
* Data preprocessing
* Data splitting
* Classification
* Result Generation

**4.2 MODULES DESCRIPTION:**

**4.2.1: DATA SELECTION:**

* The input data was collected from dataset repository like UCI repository.
* In our process, house dataset is used.
* The data selection is the process of predicting the house price.
* The dataset contains the information about the house like square root, bed room, bath square feet and so on.,

**4.2.2: DATA PREPROCESSING:**

* Data pre-processing is the process of removing the unwanted data from the dataset.
* Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
* This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
* Missing data removal
* Encoding Categorical data
* Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
* Missing and duplicate values were removed and data was cleaned of any abnormalities.
* Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
* That most machine learning algorithms require numerical input and output variables.

**4.2.3: DATA SPLITTING:**

* During the machine learning process, data are needed so that learning can take place.
* In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
* In our process, we considered 70% of the Bot-IoT dataset to be the training data and the remaining 30% to be the testing data.
* Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
* One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
* Separating data into training and testing sets is an important part of evaluating data mining models.
* Typically, when you separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

**4.2.4: CLASSIFICATION:**

* In our process, we have to implement the machine learning algorithms such as lasso regression and ridge regression.
* **Ridge regression** is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.
* Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters)

**4.2.5: RESULT GENERATION:**

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

* **MAE:** In statistics, the **mean absolute error** (MAE) is a way to measure the accuracy of a given model. It is calculated as:

**MAE = (1/n) \* Σ|yi – xi|**

Where:

* **Σ:** A Greek symbol that means “sum”
* **yi:** The observed value for the ith observation
* **xi:** The predicted value for the ith observation
* **n:** The total number of observations

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

**5.1 HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz
* Hard Disk : 200 GB
* Mouse : Logitech.
* Keyboard : 110 keys enhanced
* Ram : 4GB

**5.2 SOFTWARE REQUIREMENTS:**

* O/S : Windows 7.
* Language : Python
* Front End : Anaconda Navigator – Spyder

**5.3 SOFTWARE DESCRIPTION:**

**5.3.1 Python**

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. I will discuss most of these features in more detail in the next section.

## **5.3.2 Features of Python**

### **Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

### **Easy to Learn**

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

### **Free and Open Source**

Python is an example of a FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

### **High-level Language**

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

### **Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

You can use Python on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

You can even use a platform like [Kivy](http://kivy.org) to create games for your computer and for iPhone, iPad, and Android.

### **Interpreted**

This requires a bit of explanation.

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When you run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. You just run the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes your Python programs much more portable, since you can just copy your Python program onto another computer and it just works!

### **Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

### **Extensible**

If you need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, you can code that part of your program in C or C++ and then use it from your Python program.

### **Embeddable**

You can embed Python within your C/C++ programs to give scripting capabilities for your program's users.

### **Extensive Libraries**

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

**5.4 TESTING PRODUCTS:**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**5.4.1 UNIT TESTING:**

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’.

The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

**5.4.2 INTEGRATION TESTING:**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing. ii) Bottom-up integration testing.

**5.4.3 TESTING TECHNIQUES/STRATEGIES:**

* **WHITE BOX TESTING:**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we

Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

* **BLACK BOX TESTING:**

1. Black box testing is done to find incorrect or missing function
2. Interface error
3. Errors in external database access
4. Performance errors.
5. Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**5.4.4 SOFTWARE TESTING STRATEGIES**

**VALIDATION TESTING:**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many,

But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer

**USER ACCEPTANCE TESTING:**

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

**OUTPUT TESTING**:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**CHAPTER 6**

**CONCLUSION**

We conclude that, the house price dataset was taken as input. The input dataset was mentioned in our research paper. We are implemented the different machine algorithm such as lasso regression and ridge regression. Then, we are predicted the house price and performance metrics such as MAE.

**CHAPTER 7**

**FUTURE ENHANCEMENT**

In the future, we should like to hybrid the two different machine learning. In future, it is possible to provide extensions or modifications to the proposed clustering and classification algorithms to achieve further increased performance. Apart from the experimented combination of data mining techniques, further combinations and other clustering algorithms can be used to improve the detection accuracy.

**CHAPTER 8**

**SAMPLE CODING**

#======================= IMPORT PACKAGES ============================

import pandas as pd

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

import warnings

warnings.filterwarnings('ignore')

from sklearn import preprocessing

#======================= DATA SELECTION =========================

print("=======================================")

print("---------- Data Selection -------------")

print("=======================================")

data=pd.read\_csv('train.csv')

print(data.head(10))

print()

#==================== PREPROCESSING =======================================

#checking missing values

print("=====================================================")

print("--------- Before Checking missing values ------------")

print("=====================================================")

print(data.isnull().sum())

print()

print("=====================================================")

print("--------- After Checking missing values ------------")

print("=====================================================")

data=data.fillna(0)

print(data.isnull().sum())

print()

#==== LABEL ENCODING ====

label\_encoder = preprocessing.LabelEncoder()

print("------------------------------------------------------")

print(" Before label encoding ")

print("------------------------------------------------------")

print()

print(data['SaleCondition'].head(10))

print("------------------------------------------------------")

print("After label encoding ")

print("------------------------------------------------------")

print()

data= data.astype(str).apply(label\_encoder.fit\_transform)

print(data['SaleCondition'].head(10))

#========================= DATA SPLITTING ============================

#=== TEST AND TRAIN ===

x=data.drop('SalePrice',axis=1)

y=data['SalePrice']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=2)

print("-----------------------------------------------------------")

print("======================= Data splitting ====================")

print("-----------------------------------------------------------")

print()

print("Total No Of data :",data.shape[0])

print()

print("Total No of Training data :",X\_train.shape[0])

print()

print("Total No of Testing data :",X\_test.shape[0])

print()

#========================= CLASSIFICATION ============================

from sklearn.linear\_model import Ridge

from sklearn import metrics

#=== ridge regression ===

#initialize the model

ridgeR = Ridge(alpha = 1)

#fitting the model

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

#predict the model

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

print("-----------------------------------------------------------")

print("======================= RIDGE REGRESSION ==================")

print("-----------------------------------------------------------")

print()

mae\_ridge=metrics.mean\_absolute\_error(y\_test, y\_pred)

print("1.Mean Absolute Error : ",mae\_ridge)

print()

mse\_ridge=metrics.mean\_squared\_error(y\_test, y\_pred)/1000

print("2.Mean Squared Error : ",mae\_ridge)

print()

import numpy as np

rmse\_rid=np.sqrt(mse\_ridge)

print("3.Root Mean Squared Error : ",rmse\_rid)

#=== lasso regression ===

from sklearn.linear\_model import Lasso

#initialize the model

lasso = Lasso(alpha = 1)

#fitting the model

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

#predict the model

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

print("-----------------------------------------------------------")

print("======================= LASSO REGRESSION ==================")

print("-----------------------------------------------------------")

print()

mae\_lasso=metrics.mean\_absolute\_error(y\_test, y\_pred)

print("1.Mean Absolute Error : ",mae\_lasso)

print()

mse\_lasso=metrics.mean\_squared\_error(y\_test, y\_pred)/1000

print("2.Mean Squared Error : ",mse\_lasso)

print()

import numpy as np

rmse\_las=np.sqrt(mse\_lasso)

print("3.Root Mean Squared Error : ",rmse\_las)

#========================= PREDICTION ============================

print("-----------------------------------------------------------")

print("======================= PREDICTION ========================")

print("-----------------------------------------------------------")

print()

for i in range(0,10):

Results=y\_pred[i]

print("------------------------------------------")

print()

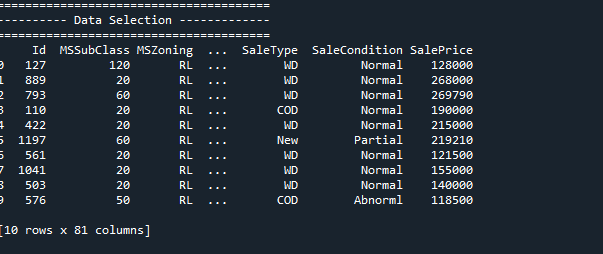
print([i],"The predicted house price is ", Results)

print()

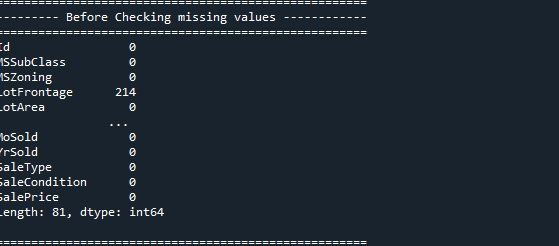
**CHAPTER 9**

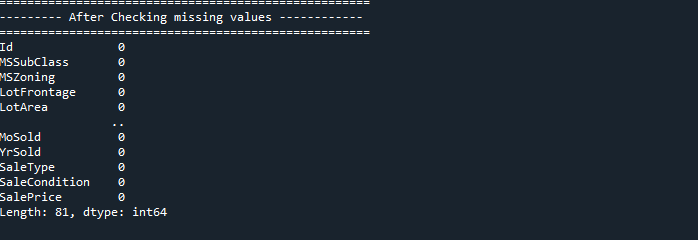
**SAMPLE SCREENSHOTS**

***1. Data Selection:***

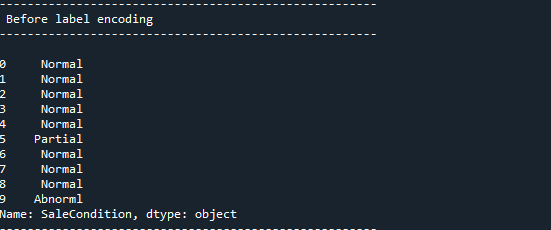


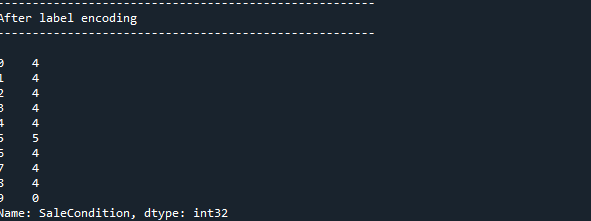
***2. Pre-processing-missing values:***

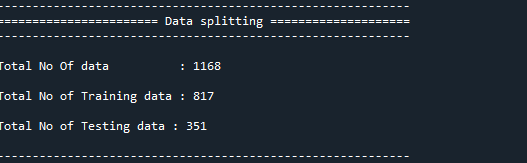




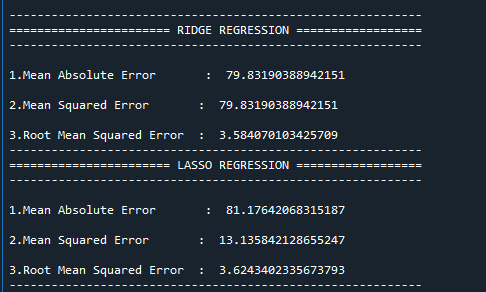
***3. Pre-processing-Label Encoding:***

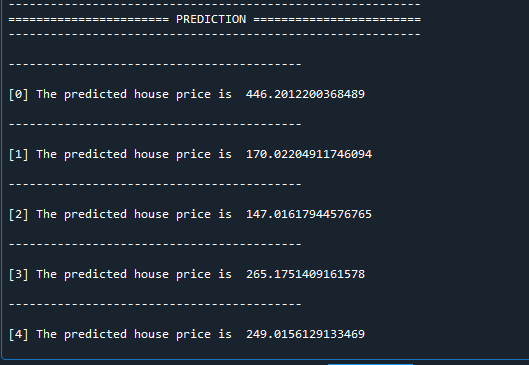






***4. ML ALGORITHM:***





**CHAPTER 10**

**REFERENCES**

[1] Jiawei Han, MichelineKamber, “Data Mining Concepts and Techniques”, pp. 279-328, 2001.

[2] J. R.Quinlan,” Simplifying decision trees”, Int. J. HumanComputer Studies.

[3] Maria-Luiza Antonie, et. al., “Application of Data Mining Techniques for Medical Image Classification”, Proceedings of the Second International Workshop on multimedia Data Mining(MDM/KDD’2001) in conjunction with ACM SIGKDD conference. San Francisco,USA, August 26,2001.

[4] Nikita Patel and Saurabh Upadhyay, “Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA”, International Journal of Computer Applications, Volume 60– No.12, December 2012, pp 20-25.

[5] J.R. Quinlan, “C4.5: programs for Machine Learning”, Morgan Kaufmann, New York, 1993.

[6] J.R. Quinlan, “Induction of Decision Trees”, Machine Learning 1, 1986, pp.81-106.

[7] SamDrazin and Matt Montag”, Decision Tree Analysis using Weka”, Machine Learning-Project II, University of Miami.

[8] Gang-Zhi Fan, Seow Eng Ong and Hian Chye Koh, “Determinants of House Price: A Decision Tree Approach”, Urban Studies, Vol. 43, No. 12, November 2006, PP.NO.2301- 2315.

[9] Ong, S. E., Ho, K. H. D. and Lim, C. H., “A constantquality price index for resale public housing flats in Singapore”, Urban Studies, 40(13), 2003, pp. 2705 –2729.

[10] Berry, J., McGreal, S., Stevenson, S., “Estimation of apartment submarkets in Dublin, Ireland”, Journal of Real Estate Research, 25(2), 2003, pp. 159–170.

[11] Neelam Shinde, Kiran Gawande, “Valuation of house prices using Predictive Techniques”, International Journal of Advances in Electronics and Computer Science, ISSN: 2393-2835, Volume-5, Issue-6, Jun.-2018 pp. 34 to 40.

[12] Adyan Nur Alfiyatin , Hilman Taufiq, Ruth Ema Febrita and Wayan Firdaus Mahmudy, “Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 10, 2017, pp. 323 to 326.

[13] Timothy C. Au, “Random Forests, Decision Trees, and Categorical Predictors: The Absent Levels Problem”, Journal of Machine Learning Research 19, 2018, pp. no.1- 30.

[14] K.C. Tan, E.J. Teoh, Q. Yu, K.C. Goh,” A hybrid evolutionary algorithm for attribute selection in data mining”, Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, Singapore.Rochester, Institute of Technology, USA.

[15] Liangxiao Jiang, Chaoqun LI, “An Empirical Study on Attribute Selection Measures in Decision Tree Learning”, Journal of Computational Information Systems6:1, 2010, pp. 105-112.