**House Price Prediction using Python and Machine Learning**

Submitted in partial fulfilment of the requirements for the award of the degree of

**Bachelor of Technology**

**in**

**INFORMATION TECHNOLOGY**

Guided Under: - Submitted By: -

Mr. Devanshu Shukla Vaibhav Khera

(Chief Mentor) (35651203117)



**Information Technology Department**

**Bharati Vidyapeeth’s College of Engineering** **A-4, Paschim Vihar, New Delhi-110063**

**August, 2019**

# CERTIFICATE

# CANDIDATE’S DECLARATION

I hereby declare that the work presented in this report entitled “**House Price Prediction**”, in partial fulfilment of the requirements for the award of the degree **Bachelor of Technology** and submitted in **Information Technology Department of Bharati Vidyapeeth’s College of Engineering**,(affiliated to Guru Gobind Singh Indraprastha University, New Delhi) is an authentic record of my own work carried out during the period from June – July 2019 under the guidance of **Mr. Devanshu Shukla, Chief Mentor**.

The work reported in this has not been submitted by me for the award of any other degree or diploma of this or any other institute.

**Vaibhav Khera**

**(35651203117)**

# ACKNOWLEDGEMENT

I express my deep gratitude to Mr. Devanshu Shukla, Chief Mentor, Hackveda Pvt. Ltd., for his valuable guidance and suggestion throughout my project work. We are thankful to Mr. Arun Dubey and Mr. Pranav Das, Class Advisor for their valuable guidance.

I am thankful to every member of the project for making this period of time knowledgeable and inspirational experience.

**Vaibhav Khera**

**Enrollment No. (35651203117)**

# TABLE OF CONTENTS

[CERTIFICATE 2](#_Toc26446771)

[CANDIDATE’S DECLARATION 3](#_Toc26446772)

[ACKNOWLEDGEMENT 4](#_Toc26446773)

[TABLE OF CONTENTS 5](#_Toc26446774)

[APPENDIX 7](#_Toc26446775)

[ORGANIZATION PROFILE 8](#_Toc26446776)

[ABSTRACT 9](#_Toc26446777)

[Chapter – 1 Introduction to Project 10](#_Toc21081981)

1.1 Problem Statement

1.2 Explanation of Problem

1.3 Approach

[Chapter – 2 System Requirements 11](#_Toc26446780)

[[Chapter – 3 System Design 12](#_Toc26446782)](#_Toc21081985)

[3.1 Physical Design](#_Toc26446782)

[3.2 System Flow Diagram](#_Toc26446782)

[[Chapter – 4 Environment 13](#_Toc26446784)](#_Toc21081987)

[[Chapter – 5 Libraries 14](#_Toc26446786)](#_Toc21081989)

[5.1 Pandas](#_Toc26446786)

[5.2 NumPy](#_Toc26446786)

[5.3 Matplotlib](#_Toc26446786)

[5.4 Scikit-learn](#_Toc26446786)

[[Chapter – 6 Dataset 15](#_Toc26446788)](#_Toc21081991)

[6.1 About Dataset](#_Toc26446788)

[6.2 Dependent and Independent variables](#_Toc26446788)

[6.3 Data Description](#_Toc26446788)

[[Chapter – 7 Working 18](#_Toc26446790)](#_Toc21081993)

[7.1 What is the KDD Process?](#_Toc26446790)

[7.2 Procedure:](#_Toc26446790)

[[Chapter – 8 Testing 26](#_Toc26446792)](#_Toc21081995)

[8.1 Kinds of Testing:](#_Toc26446792)

[Chapter - 9](#_Toc26446794) [Google Cloud Platform 27](#_Toc26446795)

[Chapter - 10](#_Toc26446796) [Interface Design 29](#_Toc26446797)

**10.1 Input Design:**

**10.2 Output Design:**

[Chapter –11](#_Toc26446798) [Challenges Faced 31](#_Toc26446799)

[Chapter –12](#_Toc26446800) [Future Development 32](#_Toc26446801)

[Limitations 33](#_Toc26446802)

[CONCLUSION 34](#_Toc26446803)

[REFERENCES 35](#_Toc26446804)

# APPENDIX

* **List of Figures:**

|  |  |
| --- | --- |
| **Figures Sr. No.** | **Description** |
| Figure No-3.1 | Flow Diagram |
| Figure No.- 4.1 | Layout of Google Colab |
| Figure No. - 6.1 | Example of Dataset |
| Figure No. – 7.1 | An Outline of the Steps of the KDD Process |
| Figure No. – 7.2 | Visualization of data before pre-processing. |
| Figure No. – 7.3 | Visualization of data after pre-processing. |
| Figure No. – 7.4 | One Hot Encoding |
| Figure No. – 7.5 | Heatmap-Correlation Matrix Graph |
| Figure No. – 7.6 | OLS Model Summary |
| Figure No. – 7.7 | Predicted Price |
| Figure No. – 7.8 | Model Accuracy |
| Figure No. – 7.9 | Overview of GCP |
| Figure No. – 7.10 | Layout of VM Instances. |
| Figure No. – 7.11 | Layout of Input Design |
| Figure No. – 7.12 | Layout of Output Design |

* **List of Tables:**

|  |  |
| --- | --- |
| Table No-2.1 | Hardware Requirements |
| Table No-2.2 | Software Requirements |
| Table No. - 6.1 | Dependent and Independent Variable |
| Table No. - 6.2 | Numerical and Binary Data |
| Table No. – 8.1 | Input Fields |

# ORGANIZATION PROFILE

* The name of the organization where the project is intended is Hackveda, VMDD Technologies. Hackveda, VMDD Technologies is an ISO 9001:2008, approved organization for Software Development. VMDD Technologies was established in 2008 by Mr. V.K.Shukla, RETD. Indian Air Force Officer. During this period VMDD Technologies has successfully delivered product and services to Defense Research Development Organization, Ministry Of Defense (India), Indian Institute of Technology, National Physical Laboratory and various other Govt. and private organization.
* VMDD Technologies is providing software product & services since 2011. VMDD started its cyber security & Training Service in 2011. Later Hackveda, VMDD Technologies Introduced android application development & training Services, commercial android applications development etc. Hackveda is also a Creative Learning, Certification, Development, Publishing and Research Center.

# ABSTRACT

* The House Price Prediction is designed to provide user efficient working environment. This system provides user friendly interface resulting in knowing each and every usability features of the system.
* This system completes the work in a very less time resulting in less time consumption and high level of efficiency.
* This system is developed in such a way that even a naïve user can also operate the system easily. They can depend on the dataset.
* The developed system helps in predicting the house price.
* The main objectives of our project are: -
  + Management of house prices faster and easier at one place: By using of this software one can easily get the price of house which he/she wants to buys or sells at the suitable requirements.
  + Customers satisfaction: One can directly go through to our website for checking the estimate of house price.
  + Unbiased Results: As the users won’t be cheated by the buyer or seller.

**Need of the New System**

Following are the user’s requirements which old system don’t hold and therefore there is a need to develop a new efficient system:-

(a) Removes overhead of third Party

(b) Search house according to their need

(c) It provides all needed information related to house like lotsize, does your house has Garage

(d) Secrecy of information

(e) Fast Processing

(f) Faster and accurate generation of reports

(g) Decision making is quite easy in this system.

**Proposed system advantages**

Following are the characteristics of proposed system: -

(a) Efficiency: Provides appropriate interactions with the resources, data accessing, data management, memory management, coding practices, compliance with Python and MySQL programming.

(b) Effectiveness: The effectiveness of proposed system is measured in producing a desired result.

The scope of this system is to provide user efficient house prediction through each user search houses along with all needed things. This system completes the work in a very less time resulting in less time consumption and high level of efficiency. Also, this system provides high level of security.

# Chapter – 1

# Introduction to Project

**1.1 Problem Statement**

To develop a House Price Prediction System, using important factors of house on which price is dependent and predict the price of the house given these factors.

**1.2 Explanation of Problem**

For many people, buying a property is one of the most important decision and purchase in life. Besides the affordability of a house, other factors such as the desirability of the location also affect the decision-making process.

Researching how much the house you are interested in is worth on your own can be difﬁcult for multiple reasons. One particular reason is that there many factors that inﬂuence the potential price of a house, making it more complicated for an individual to decide how much a house is worth on their own without external help. This can lead to people making

Poorly informed decisions whether to buy houses and which price is reasonable. Because houses are long term investments, it is imperative that people make their decisions with the most accurate information possible.

When a buyer looks to buy a house, he/she is often made to pay higher prices than what it really costs, and unknowingly they fall for it. A good housing price prediction would better prepare them for what to expect before they make one of the most important ﬁnancial decisions in their lives.

Thus, we need a system which analysis the factors on which the price of a house depends and which based on such factors predicts the accurate price of the house according to buyer’s configuration.

**1.3 Approach**

To Extract knowledge from dataset available, we followed the KDD (Knowledge Discovery Process) approach

This process includes the following stages:  
1. Data collection

2. Data selection (Dependent, Independent Variables)

3. Data pre-processing (Remove missing values, convert text into numbers)

4. Data transformation (Scale all the factors/variables on a common scale between 0 - 1)  
  
5. Data interpretation and Evaluation (generate graphs of the values and identify possible reasons for the facts)

Knowledge (Logic/Mathematical equation/formula) is acquired. Using this knowledge machine can perform prediction/forecasting/machine learning.

# Chapter – 2

# System Requirements

|  |  |
| --- | --- |
| Minimum Requirements | |
| Computer (CPU) | 1GHz Intel or AMD processor |
| Memory | At least 2 GB RAM |
| Hard Disk | 1 GB of Free Space |
| Display | Super VGA (800 × 600) or higher-resolution monitor with 256 colors |
| Input Device | Keyboard, Mouse |

Table No-2.1: Hardware Requirements

|  |  |
| --- | --- |
| Minimum Requirements | |
| Programming Language | Python |
| Operating System | 64-Bit Required. Windows 7(SP1), Windows 8,  Windows 8.1 or Windows 10 |
| Development Tool | Anaconda Navigator/Google Colab |
| Internet Connectivity | Wi-Fi, Ethernet (Minimum 2 Mbps) |
| Web Browser | Internet Explorer (5.0 or Above), Firefox, Google  Chrome |

Table No-2.2: Software Requirements

# Chapter – 3

# System Design

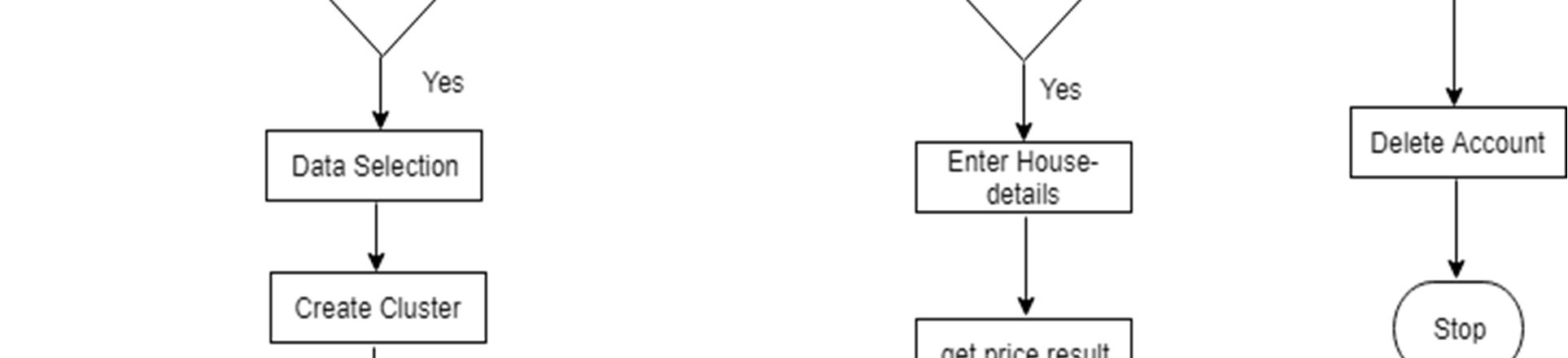
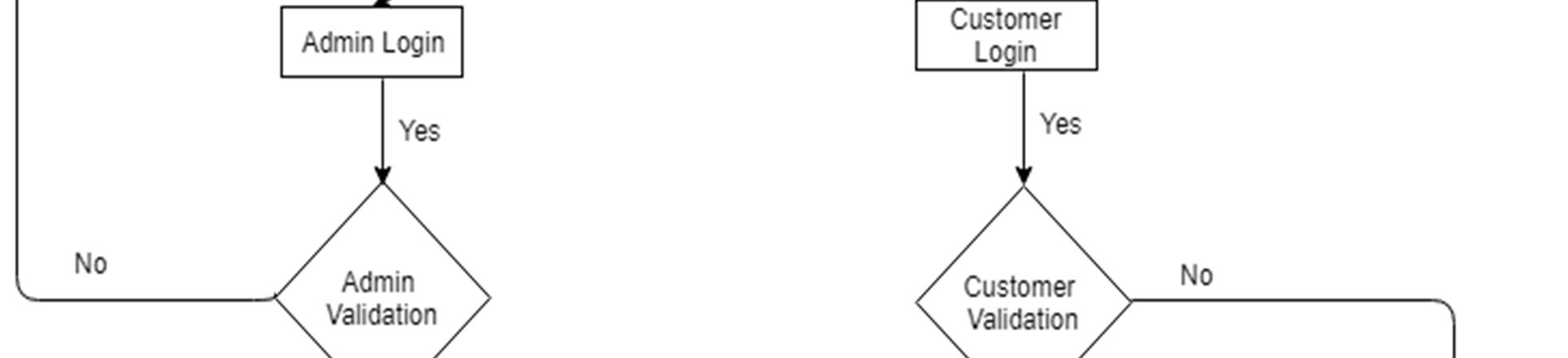
**3.1 Physical Design**

In the following section the relationship between various components (processes, input, output and entities) of the system is represented using various diagrams. During the physical design process, you convert the data gathered during the logical design phase into a description of the physical database, including tables and constraints. Physical design decisions, such as the type of index or partitioning have a large impact on query performance.

**3.2** **System Flow Diagram**

It is a collective term for a diagram representing a flow or set of dynamic relationships in a system. The term flow diagram is also used as a synonym for flowchart and sometimes as a counterpart of the flowchart. The term flow diagram is used in theory and practice in different meanings.

Figure No-3.1: Flow Diagram



# Chapter – 4

# Environment

**Google Colaboratory**

* Colaboratory (also known as Colab) is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. Colab notebooks are Jupyter notebooks that are hosted by Colab.
* With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.
* When you create your own Colab notebooks, they are stored in your Google Drive account. You can easily share your Colab notebooks with co-workers or friends, allowing them to comment on your notebooks or even edit them.
* Colaboratory is a free Jupyter notebook environment provided by Google where you can use free GPUs and TPUs. It’s a Jupyter notebook environment that requires no setup to use.

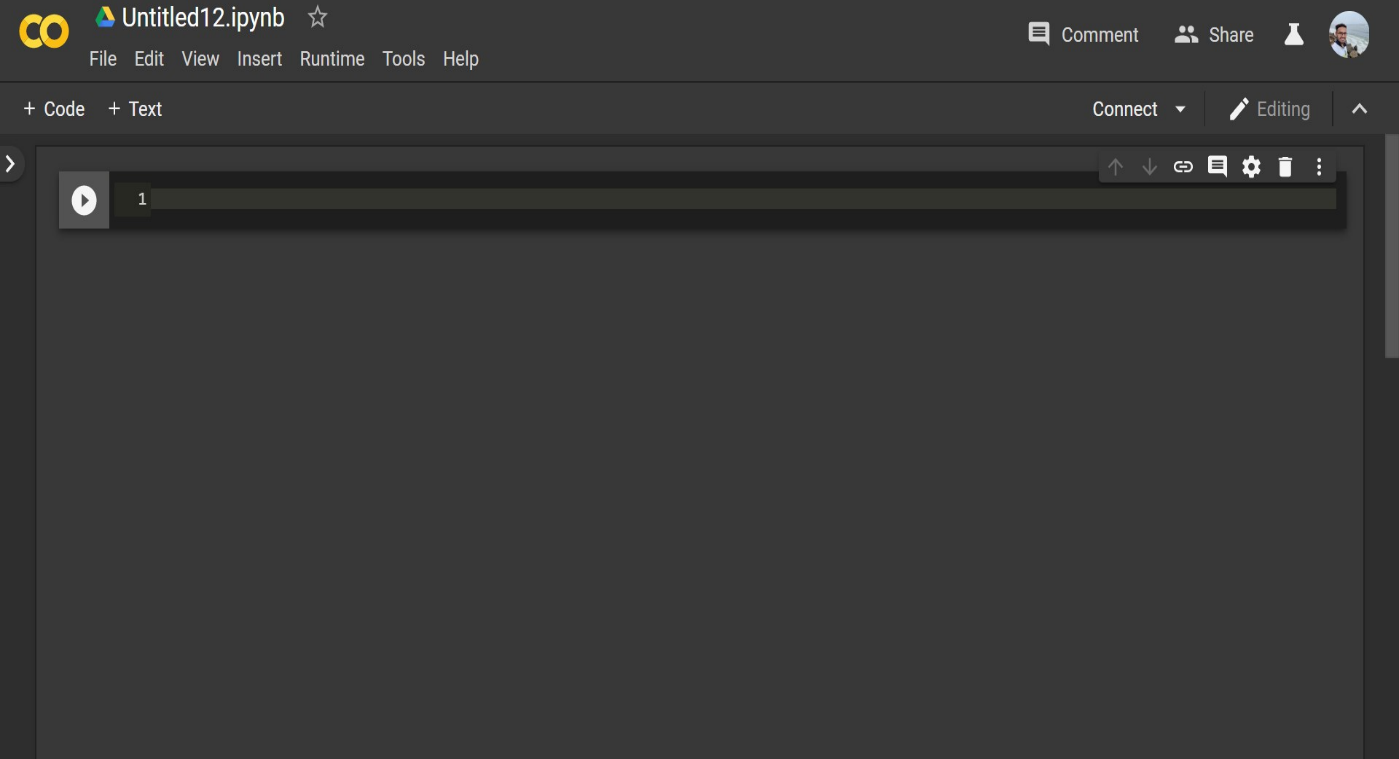
****

Figure No.- 4.1: Layout of Google Colab

# Chapter – 5

# Libraries

**5.1 Pandas**

Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the python programming language.

Pandas is mainly used for Machine Learning in form of data frames. Pandas allow importing data of various file formats such as csv, excel etc. Pandas allows various data manipulation operations such as groupby, join, merge, melt, concatenation as well as data cleaning features such as filling, replacing or imputing null values.

**5.2 NumPy**

Numpy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays.

The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are strided views on memory.

**5.3 Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

**5.4 Scikit-learn**

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language.

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

* Simple and efficient tools for data mining and data analysis.
* Accessible to everybody, and reusable in various contexts
* Built on NumPy, SciPy, and matplotlib
* Open source, commercially usable - BSD license
* Classification - Identifying to which category an object belongs to.
* Regression - Predicting a continuous-valued attribute associated with an object.
* Clustering- Automatic grouping of similar objects into sets.

# Chapter – 6

# Dataset

**6.1 About Dataset**

Dataset is the most important step for solving any Machine Learning problem. The dataset we used is “Housing\_Modified.csv”, which consists of Price of house and the various factors that affect the price.

The dataset is maintained in a CSV File(Comma Separated Values) and we use this file to perform analysis of the data.

Columns of the dataset are:

* price (in USD)
* lotsize (in Sq. Metres)
* bedrooms (No. of bedrooms -1, 2,3..)
* bathrms (No. of bathrooms -1, 2,..)
* stories (Categorial values like One,Two,Three..)
* driveway (Binary Categorical values like yes/no)
* recroom (Binary Categorical values like yes/no)
* fullbase (Binary Categorical values like yes/no)
* gashw (Binary Categorical values like yes/no)
* airco (Binary Categorical values like yes/no)
* garagepl (No. of garage place -0,1, 2)
* prefarea (Binary Categorical values like yes/no)

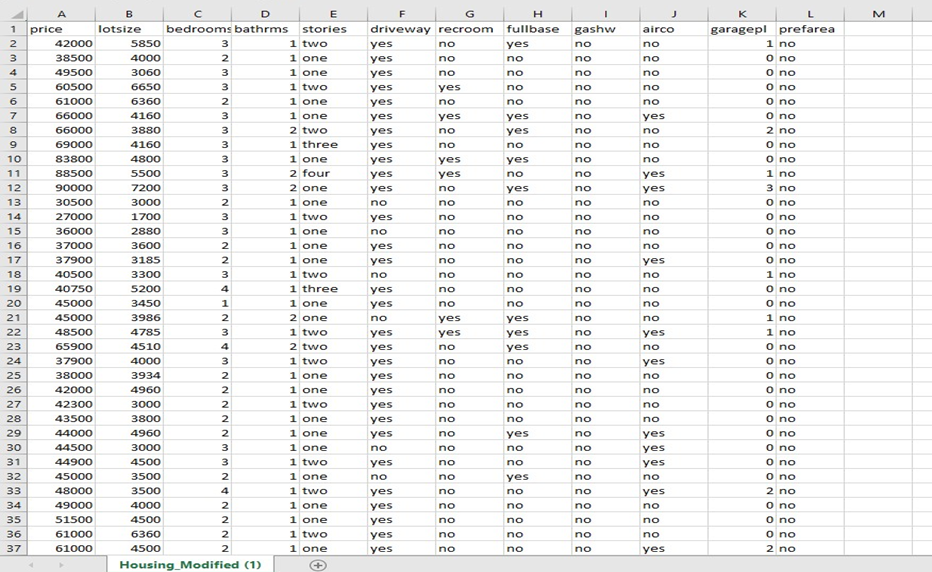


Figure No. - 6.1: Example of Dataset

**6.2 Dependent and Independent variables**

Out of all the information available to us, we have identiﬁed relevant features to be used in our prediction model, as we believe these variables contribute to the price of a house.

|  |  |
| --- | --- |
| Independent Variables | Dependent Variable |
| lotsize (in Sq. Metres) | price (in USD) |
| bedrooms (No. of bedrooms -1, 2,3..) |  |
| bathrms (No. of bathrooms -1, 2,..) |  |
| stories (Categorial values like One,Two,Three..) |  |
| driveway (Binary Categorical values like yes/no) |  |
| recroom (Binary Categorical values like yes/no) |  |
| fullbase (Binary Categorical values like yes/no) |  |
| gashw (Binary Categorical values like yes/no) |  |
| airco (Binary Categorical values like yes/no) |  |
| garagepl (No. of garage place -0,1, 2) |  |
| prefarea (Binary Categorical values like yes/no) |  |

Table No. - 6.1: Dependent And Independent Variable

**6.3 Data Description**

In this dataset, there are 546 rows and 12 columns. The columns contain the factors on which the price of the house depends.

Some columns contain numeric values while others contain binary categorical values and thus, needs to be converted into numerical characters to evaluate the dataset, as the regression techniques only work on numerical data.

Columns with Numerical and Binary Data

Table No. - 6.2: Numerical and Binary Data

|  |  |
| --- | --- |
| Numerical Data | Binary Data |
| price (in USD) | driveway (Binary Categorical values like yes/no) |
| lotsize (in Sq. Metres) | recroom (Binary Categorical values like yes/no) |
| bedrooms (No. of bedrooms -1, 2,3..) | fullbase (Binary Categorical values like yes/no) |
| bathrms (No. of bathrooms -1, 2,..) | gashw (Binary Categorical values like yes/no) |
| garagepl (No. of garage place -0,1, 2) | airco (Binary Categorical values like yes/no) |
|  | prefarea (Binary Categorical values like yes/no) |

# Chapter – 7

# Working

**7.1 What is the KDD Process?**

The term Knowledge Discovery in Databases, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.

The unifying goal of the KDD process is to extract knowledge from data in the context of large databases.

It does this by using data mining methods (algorithms) to extract (identify) what is deemed knowledge, according to the specifications of measures and thresholds, using a database along with any required pre-processing, subsampling, and transformations of that database.

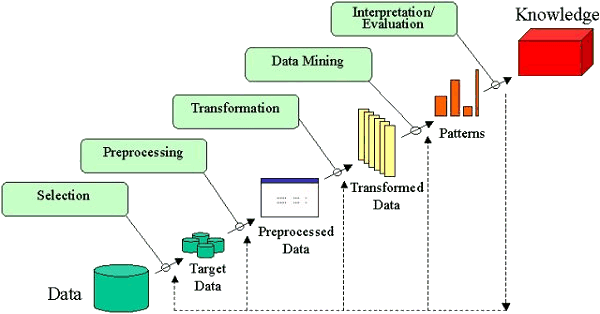


Figure No. – 7.1: An Outline of the Steps of the KDD Process

**7.1.1 Steps**

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Developing an understanding of

* the application domain
* the relevant prior knowledge
* the goals of the end-user

2. Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed.

3. Data cleaning and pre-processing

* Removal of noise or outliers.
* Collecting necessary information to model or account for noise.
* Strategies for handling missing data fields.
* Accounting for time sequence information and known changes.

4. Data reduction and projection

* Finding useful features to represent the data depending on the goal of the task.
* Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.

5. Choosing the data mining task

* Deciding whether the goal of the KDD process is classification, regression, clustering, etc.

6. Choosing the data mining algorithm(s)

* Selecting method(s) to be used for searching for patterns in the data.
* Deciding which models and parameters may be appropriate.
* Matching a particular data mining method with the overall criteria of the KDD process.

7. Data mining

* Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.

8. Interpreting mined patterns.

9. Consolidating discovered knowledge.

**7.2 Procedure:**

**7.2.1 Data Visualization**

Data Visualization and Data Analysis is the most crucial part before proceeding further in manipulating data in a number of different ways, such as plotting it out and finding correlations or by creating a pivot table in Excel or Tableau.

A pivot table lets you sort and filter data by different variables and lets you calculate the mean, maximum, minimum and standard deviation of your data.

**7.2.2 Pre-processing**

Pre-Processing of data is also the essential part, in which the data is pre-processed on the selection of features which are useful and delete features which are not useful by analysing though plotting graphs and histograms. The pre-processing of data like finding mean, median, mode etc and generate new useful feature columns depending on given feature columns. Also, here is handling of missing Data

The pre-processing of data will help in making feature columns useful which can be used in Deep learning and Machine Learning models. The main objective is to find or arrange data in such a manner that there is some pattern observed in data

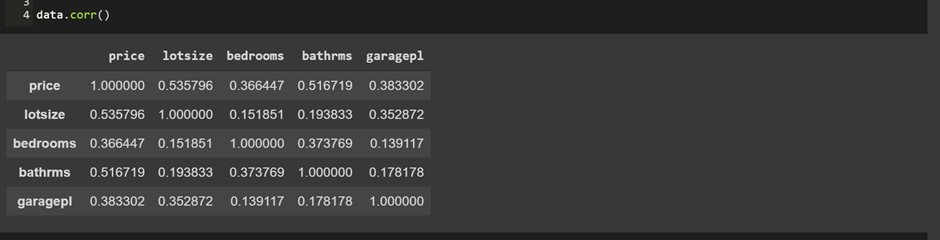


Figure No. – 7.2: Visualization of data by determining the correlation between the attributes before pre-processing.

Here we can see that it only contains 5 columns out of 12 as the other columns contains binary categorical values, which were ignored.

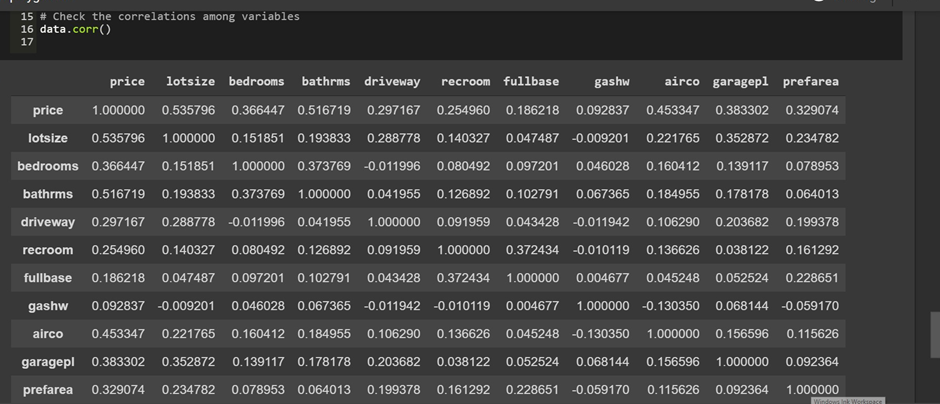


Figure No. – 7.3: Visualization of data after pre-processing.

Now when we converted the binary categorical into numerical values and then find the correlation among the variables, we get all the 12 columns instead of just 5.

**7.2.3 One Hot Encoding**

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

It refers to splitting the column which contains numerical categorical data to many columns depending on the number of categories present in that column. Each column contains “0” or “1” corresponding to which column it has been placed.

****

Figure No. – 7.4: One Hot Encoding

**7.2.4 Generation** **of** **Heatmap**

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

The seaborn python package allows the creation of annotated heatmaps which can be tweaked using Matplotlib tools as per the creator’s requirement.

* Create a correlation matrix or a heatmap visual correlation computation.
* Thumb rule is, for a value greater than 0.5 there is a correlation among independent variables.

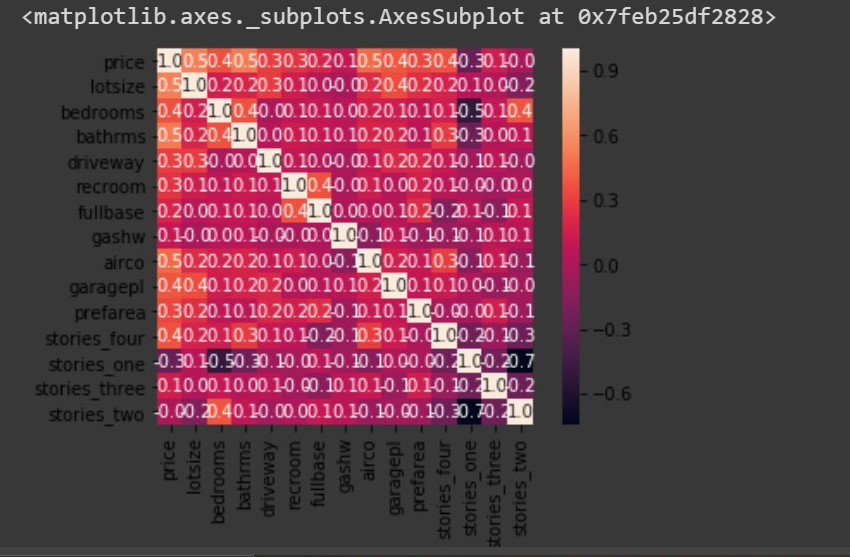


Figure No. – 7.5: Heatmap-Correlation Matrix Graph

**7.2.5 Multivariate** **Regression**

Multivariate Regression is a method used to measure the degree at which more than one independent variable (predictors) and more than one dependent variable (responses), are linearly related. The method is broadly used to predict the behavior of the response variables associated to changes in the predictor variables, once a desired degree of relation has been established.

In most of the real-world cases there will be more than one independent variables, this concept is called Multivariate Regression.

**7.2.6 Multicollinearity**

Multicollinearity exists whenever an independent variable is highly correlated with one or more of the other independent variables in a multiple regression equation. Multicollinearity is a problem because it undermines the statistical significance of an independent variable.

The best regression models are those in which the predictor variables each correlate highly with the dependent (outcome) variable but correlate at most only minimally with each other. Such a model is often called "low noise" and will be statistically robust.

An issue with multicollinearity is that small changes to the input data can lead to large changes in the model. A principal danger of such data redundancy is that of overfitting in regression analysis models.

**7.2.6.1 VIF**

variance inflation factor(VIF) detects multicollinearity in regression analysis. The VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model.

In general, a VIF above 10 indicates high correlation and is cause for concern.

A rule of thumb for interpreting the variance inflation factor:

* 1 = not correlated.
* Between 1 and 5 = moderately correlated.
* Greater than 5 = highly correlated.

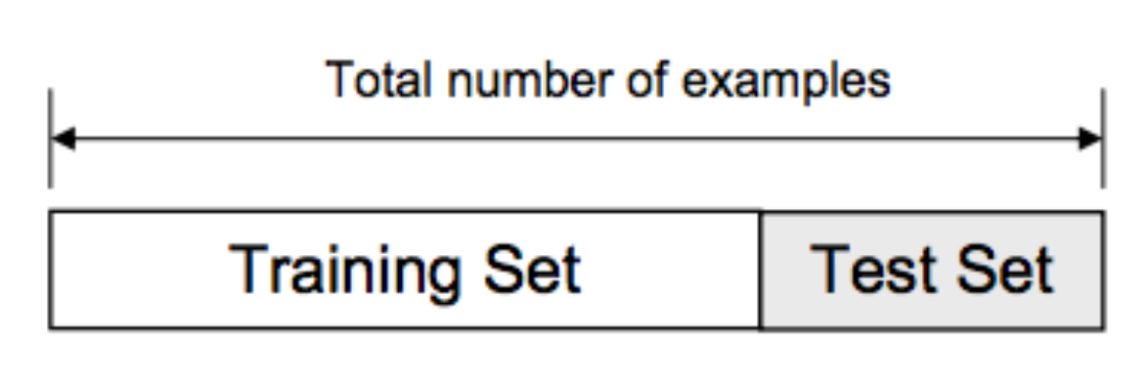
**7.2.7 Train\_Test\_Split**

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on.

We have the test dataset (or subset) in order to test our model’s prediction on this subset.

Split arrays or matrices into random train and test subsets

From Sklearn, sub-library model\_selection, we imported the train\_test\_split so we can, well, split to training and test sets.

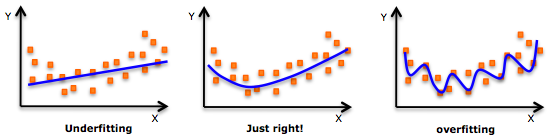


**7.2.7.1 Underfitting**:

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data.

**7.2.7.2 Overfitting**:

A statistical model is said to be overfitted, when we train it with a lot of data (just like fitting ourselves in an oversized pants!). When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set. A solution to avoid overfitting is using a linear algorithm if we have linear data.



**7.2.8 OLS Model**

Ordinary least squares (OLS) regression is a statistical method of analysis that estimates the relationship between one or more independent variables and a dependent variable; the method estimates the relationship by minimizing the sum of the squares in the difference between the observed and predicted values of the dependent variable configured as a straight line.

OLS Model acts as the same way as VIF and helps to determine whether correlation exists between independent variables.

* Thumb rule is that : if the probability value is greater than 0.05 then we can say that the variable is highly correlated.

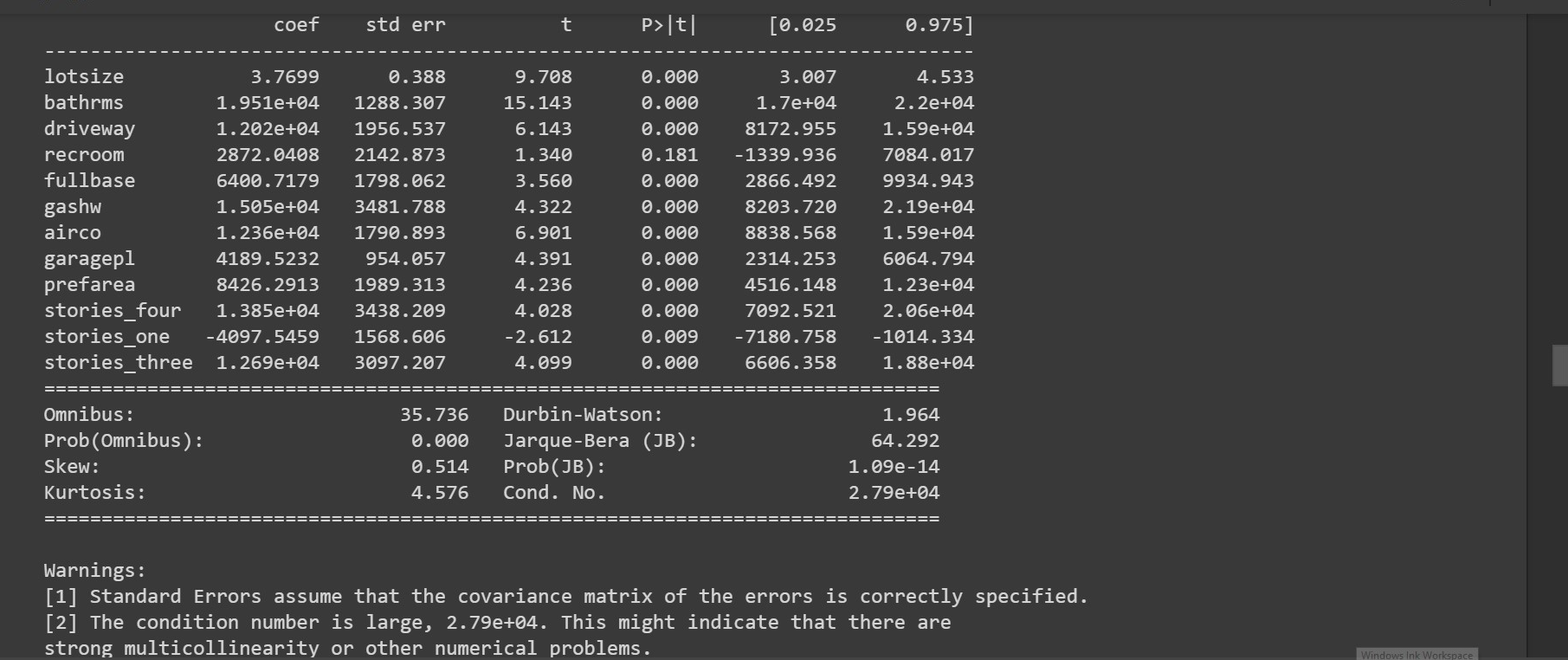


Figure No. – 7.6: OLS Model Summary

**7.2.9 Prediction**

model.predict() : given a trained model, predict the label of a new set of data. This method accepts one argument, the new data X\_new (e.g. model.predict(X\_new)), and returns the learned label for each object in the array.

Basically, when we pass the inputs in the form of an matrix into the predict method, it return linear predicted values from a design matrix.

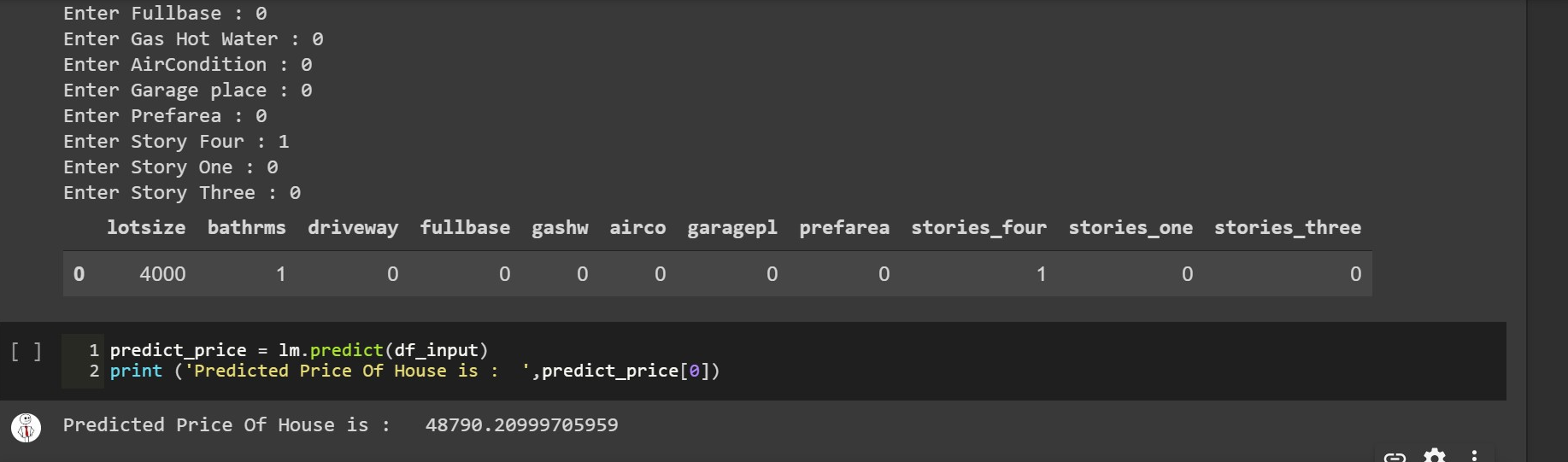


Figure No. – 7.7: Predicted Price

**7.2.10 Accuracy**

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:



At this step we compare the original values with the machine generated predicted values, to analyse the accuracy of our model.

We used r2\_score to find the accuracy. It is present in sklearn.metrics package

* Best possible score is 1.0 and it can be negative. A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

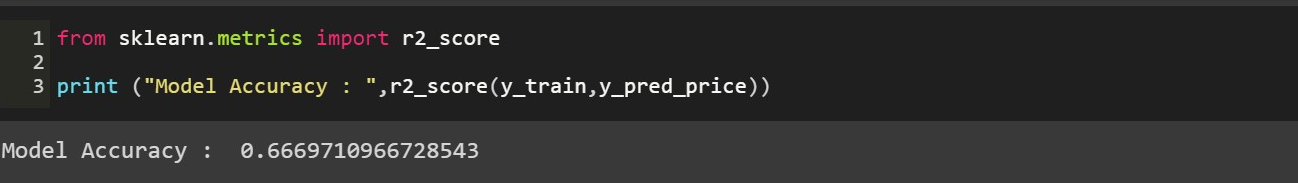


Figure No. – 7.8: Model Accuracy

# Chapter – 8

# Testing

**8.1 Kinds of Testing**:

**8.1.1** **Black Box Testing**: Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on the software requirements and specifications.

**8.1.2 White Box Testing**: White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

**8.1.3** **Unit Testing**: Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyse and fix the defects.

**8.1.4** **Integration Testing**: Testing of combined parts of an application to determine if they function together correctly. The ‘parts’ can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/ server and distributed systems.

**8.1.5 Acceptance Testing**: Final testing based on the specifications of the end user or customer or based on use by end-users/ customers over some limited period of time.

**8.1.6 User Acceptance Testing**: Determining if software is satisfactory to an end user customer.

**8.1.7 Module Testing**: A module is composed of various programs related to that module.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case | Expected Result | Actual Result |
| 1. | Leave the Lotsize and all other fields empty and press Predict to button. | Warning message displayed to prompt Lotsize and other fields. | As Expected |
| 2. | Fill all the fields as required the system to evaluate the price | Price reflected as an output on the screen | As Expected |

Table No. – 8.1: Input Fields

# Chapter - 9

# Google Cloud Platform

Google Cloud Platform (GCP) offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.

Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

Google provides a **12-month free trial** period worth **$300** of credit and on top of that it also offers a **free trial option** which has no time limit.

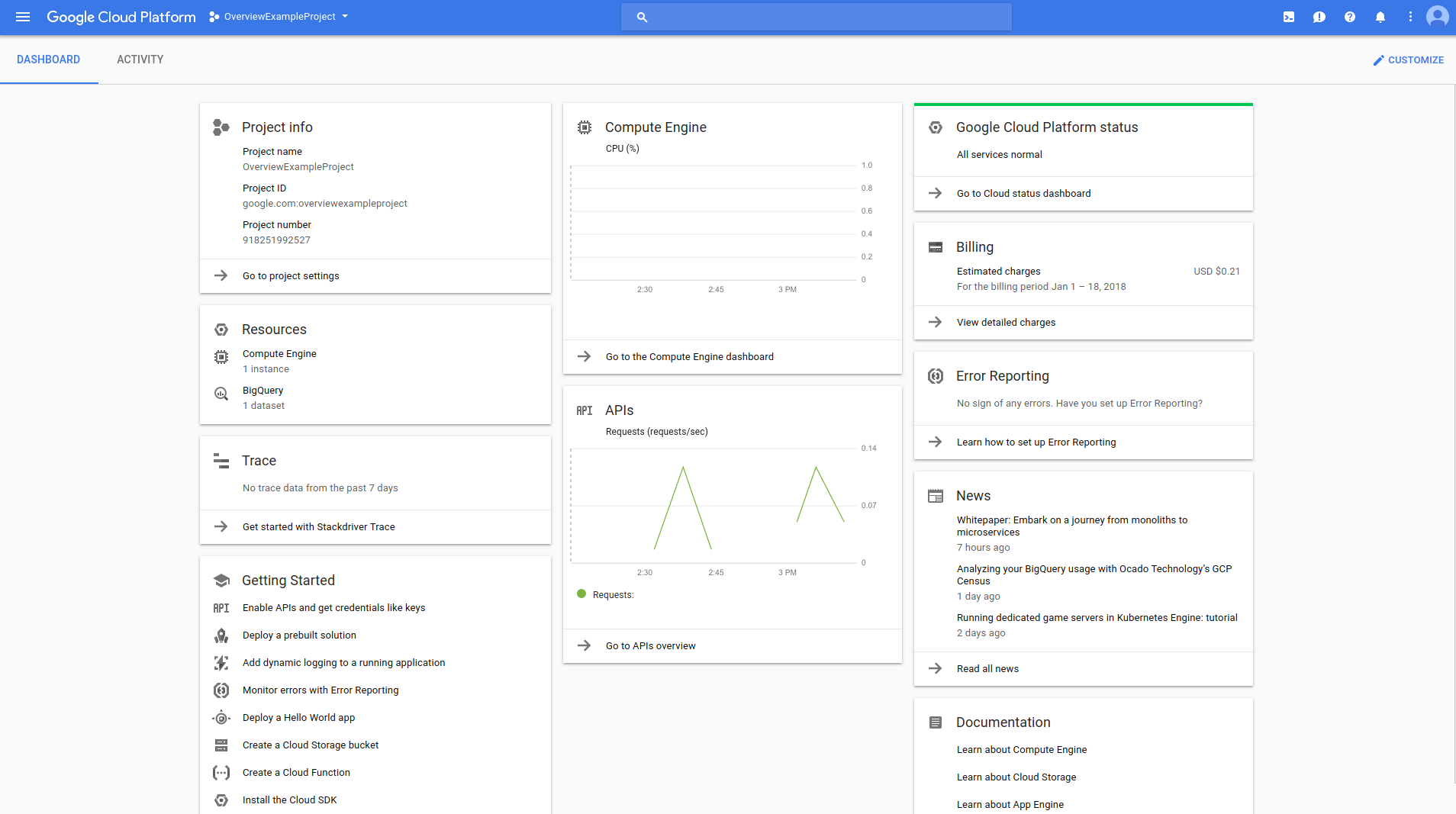


Figure No. – 7.9: Overview of GCP

**GCP also provides :-**

* A temporary Compute Engine virtual machine instance.
* Command-line access to the instance from a web browser.
* A built-in code editor.
* 5 GB of persistent disk storage.
* Pre-installed Google Cloud SDK and other tools.
* Language support for Java, Go, Python, Node.js, PHP, Ruby and .NET.
* Web preview functionality.
* Built-in authorization for access to GCP Console projects and resources.

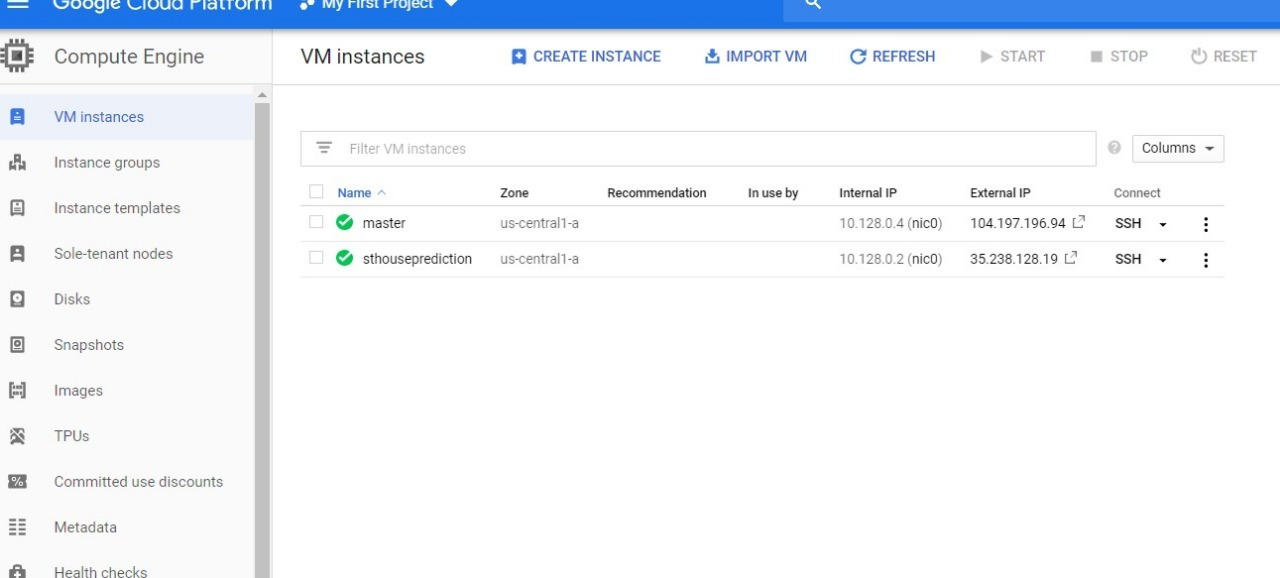


Fig. No. – 7.10: Layout of VM Instances.

# Chapter - 10

# Interface Design

**10.1 Input Design:**

This is interactive layout on web browser of house piece prediction. Here user puts the requirements on the basis of their need like Lot size, Driveway, Bathrooms, Recroom etc. After fills the all values, user click in predict button for predict the value of House.

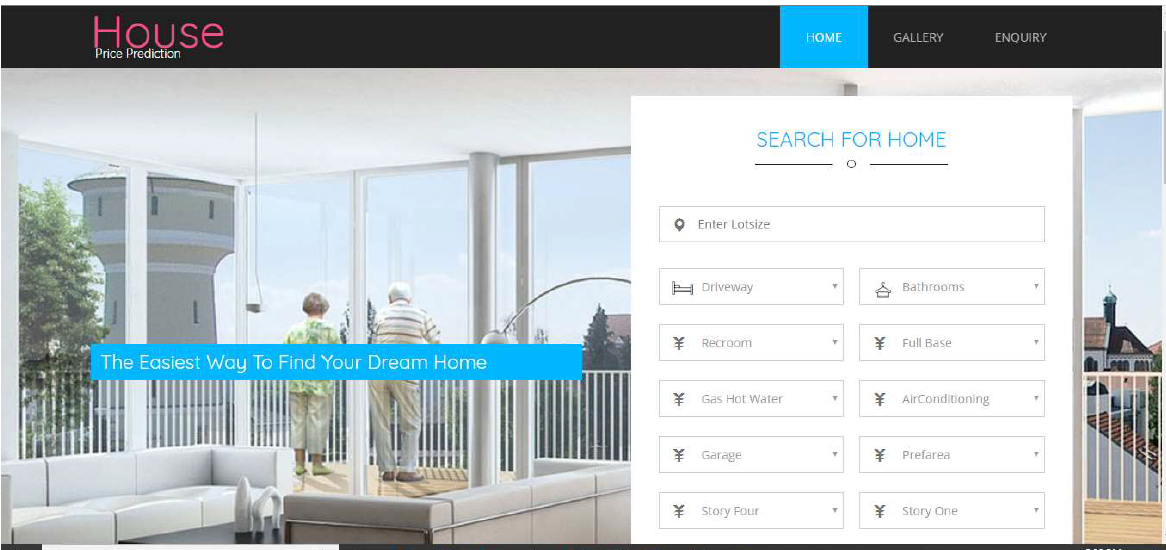
****

Fig. No. – 7.11: Layout of Input Design

**10.2 Output Design:**

Here user entered the values of all parameters on the basis of their requirement. As a result user got the house price by single click. So user a idea of their budget and pricing of house in their area or elsewhere.

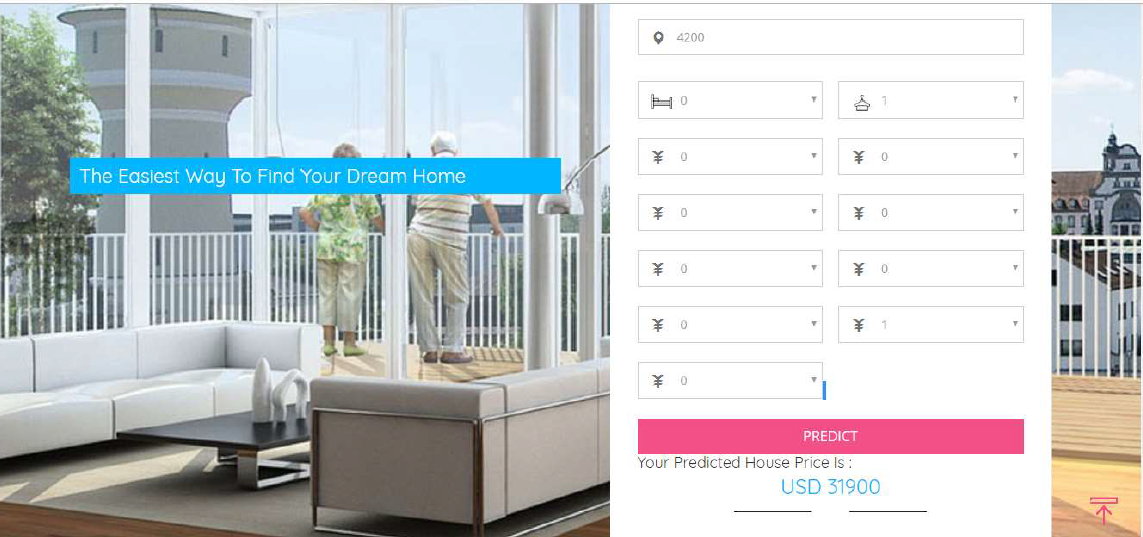
****

Fig. No. – 7.12 : Layout Of Output Design

# Chapter – 11

# Challenges Faced

Some challenges that we faced during this project are as follows:

* The dataset that we used for analysis had columns containing binary categorical values, null values and n- category values that needs to be converted into numerical values, for which we had to use various techniques during the pre-processing phase.
* The data in the dataset was highly correlated and had multiple independent variables that were correlated with each other, this needs to be corrected to acquire accurate results.
* VIF technique used to determine the correlated variables couldn’t identify all the correlated variables and we had to use another technique called OLS model to identify the remaining correlated variables correctly.

# Chapter – 12

# Future Development

* For future work, we were 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 can be useful for development of applications for various respective cities as well. The existing system involves calculation of house prices without the necessary prediction about future market trends and price increase.
* To predict the efficient house pricing for real estate customers with respect to their budgets and priorities.

# Limitations

As we know manual system are quite tedious, time consuming and less efficient and

inaccurate in comparison to the computerized system. So following are some

limitations:

1. Time consuming
2. Less accurate
3. Less efficient
4. Lot of paper work
5. Slow data processing
6. Not user friendly environment
7. Third Party Involvement
8. The system can be handle by Admin only.
9. Some keywords in system are difficult to understand so the admin or

operator person should understand them thoroughly to use the system

accurately.

# CONCLUSION

We have used machine learning algorithms to predict the house prices. We have mentioned the step by step procedure to analyse 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.

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper System.

While making the system, an eye has been kept on making it as user-friendly, as cost effective and as flexible as possible. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs.

# REFERENCES

**Websites Reference**

1. www.w3schools.com
2. http://stackoverflow.com/
3. http://www.tutorialspoint.com/asp.net/
4. https://console.cloud.google.com
5. https://Hackveda.in
6. https://Wikipedia.com
7. https://Python.org
8. http://www2.cs.uregina.ca
9. https://GeeksforGeeks