A Project report on

# HOUSE PRICE PREDICTION

submitted in partial fulfilment of the requirement for the award of degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE & ENGINEERING**

## By

KEERTHI PRANEETH KUMAR (18JJ5A0505)

SALMAN KHAN (18JJ5A0511)

BANOTH SRIKANTH NAIK (18JJ5A0502)

Under the Guidance of

# Dr B. SATEESH KUMAR

# Professor, CSE Department



# 

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

COLLEGE OF ENGINEERING

NACHUPALLY(KONDAGATTU), JAGTIAL DIST-505501, TELANGANA

2020-2021



JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

COLLEGE OF ENGINEERING

NACHUPALLY(KONDAGATTU), JAGTIAL DIST-505501, TELANGANA

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

# CERTIFICATE

This is to certify that the Industry oriented mini project work entitled “**HOUSE PRICE PREDICTION**” that is being submitted by **Keerthi Praneeth Kumar (18JJ5A0505), Salman Khan (18JJ5A0511), Banoth Srikanth Naik (18JJ5A0502)** in partial fulfilment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2020-2021.

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

----------------- --------------------------

**Project Guide Head of the Department**

**Dr B. Sateesh Kumar Dr T. Venugopal**

Professor Professor & Head

Department of CSE Department of CSE

-----------------

**External Examiner**

# ACKNOWLEDGEMENT

“Task successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it a success. Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped us for the completion of our project work.

We consider ourselves lucky enough to get such a good project. This project would add as an asset to our academic profile.

We would like to express our thankfulness to our project guide, **Dr. B. SATEESH KUMAR**, Professor of CSE for his constant motivation and valuable help through the project work, and we express our gratitude to **Dr. T. VENUGOPAL**, Professor & Head, Department of Computer Science and Engineering for giving us the opportunity to carry out his project work. We are very much grateful to our project coordinator **Dr. T. VENUGOPAL**, Professor of CSE for his constant mentoring and cooperation during the project.

We show gratitude to our beloved principal **Dr. N.V. RAMANA** and Vice Principal **Dr. S. VISWANADHA RAJU** for providing necessary infrastructure and resources for theaccomplishment of our project report at JNTUHCEJ

Finally, we would like to thanks our friends for their co-operation to complete this project

# DECLARATION

We do declare that the project work entitled "HOUSE PRICE PREDICTION" submitted by us in the Department of Computer Science and Engineering. JNTUH College of Engineering Jagtial, Telangana State in partial fulfilment of degree for the award of BACHELOR OF TECHNOLOGY is a bonafide work, which was carried out under the supervision of Dr B. SATEESH KUMAR Garu, Professor of CSE department, JNTUH College of Engineering, Jagtial.

Also, we declare that the matter embedded in this thesis has not been submitted by us in full or partial thereof for the award of any degree/diploma of any other institution or University previously.

Place: K. Praneeth Kumar (18JJ5A0505)

Date: Salman Khan (18JJ5A0511)

B. Srikanth Naik (18JJ5A0502)

# ABSTRACT

# 

The valuation of real estate is the central tenet of all business. The term valuation is defined as the analytical process of influencing the current worth of an assert. However, there is a wide range of purpose for which valuations are required. But here valuations are done in an effective way to calculate the selling price of an entity. To develop a real estate valuation model which predicts the value of a property using the domain of Machine Learning. The algorithmic approach involves usage of linear regression approach (Supervised Learning). The selling price is estimates using by considering various parameters such as ground area, year built, modified year, overall condition, overall quality, garage area, garage cars. The dataset collection is taken from a standard source such that 81 parameters along with 1461 of test and training data are considered for property valuation. Users who are going to sell the property can get the accurate values based on this regression prediction.  Users require no intermediate person (broker) to sell in the entity.

**TABLE OF CONTENTS**

**1 TITLE 1**

**2 CERTIFICATE 2**

**3 ACKNOWLEDGEMET 3**

**4 DECLARATION 4**

**5 ABSTRACT 5**

**6 TABLE OF CONTENTS 6-7**

**7 LIST OF FIGURES 8**

**8 CHAPTER-1: INTRODUCTION 9**

1.1 PROJECT OVERVIEW 9

1.2 PROBLEM STATEMENT 9

1.3 OBJECTIVES 9

**9 CHAPTER-2: SYSTEM ANALYSYS 10-11**

2.1 REQUIREMENT ANALYSIS 10

2.2 REQUIREMENT SPECIFICATIONS 11

**10 CHAPTER-3: SYSTEM DESIGN 12-14**

3.1 ARCHITECTURE 12

3.2 SEQUENTIAL FLOW OF THE PROJECT 13

3.3 HARDWARE AND SOFTWARE REQUIREMNTS 14

**11 CHAPTER-4: FEASIBILITY STUDY 15-16**

4.1 TYPES OF FEASIBILITY 15

**12 CHAPTER-5: IMPLEMENTATION TOOLS 17-28**

5.1 PYTHON 17-19

5.2 DATASET DESCRIPTION 19-20

5.3 MACHINE LEARNING 20-24

5.4 FLASK 24

5.5 SCIKITLEARN 24-25

5.6 PANDAS 25-26

5.7 NUMPY 26-27

5.8 MATPLOTLIB 27

5.9 HTML & CSS 28

5.10 PICKLE 29

**13 CHAPTER-6: EXPERIMENTAL SETUP 30**

**14 CHAPTER-7: SCREENS AND REPORTS 31-36**

7.1 CODE SCREENS 31

7.2 EXECUTION PROCESS 32-34

**15 CHAPTER-8: TESTING 35-40**

8.1 TESTING STRATERGIES 35-38

8.2 TEST CASES 38-40

**16 CHAPTER-9: FUTURE SCOPE 41**

**18 CHAPTER-10: REFERENCES 42**

**LIST OF FIGURES**

Fig 3.1 Architecture 14

Fig 3.2 Sequential Flow Diagram 15

Fig 5.1 Dataset 20

Fig 5.2 Showing the different types of Machine Learning Algorithms 22

Fig 7.1 app.py code 31

Fig 7.2 Command line execution 32

Fig 7.3 Using the IP address in web Browser 32

Fig 7.4 Cover Page 33

Fig 7.5 First page 33

Fig 7.6 Provide Input 34

Fig 7.7 Predicted result 34

Fig 8.1 Levels of testing 36

Fig 8.2 Test Case Input 40

Fig 8.3 Test Case Result 40

**1.INTRODUCTION**

* 1. **PROJECT OVERVIEW**

The valuation of real estate is the central tenet of all business. The term valuation is defined as the analytical process of influencing the current worth of an assert. However, there is a wide range of purpose for which valuations are required. But here valuations are done in an effective way to calculate the selling price of an entity. To develop a real estate valuation model which predicts the value of a property using the domain of Machine Learning. The algorithmic approach involves usage of linear regression approach (Supervised Learning). The selling price is estimated using various parameters such as ground area, year built, modified year, overall condition, overall quality, garage area, garage cars. The dataset collection is taken from a standard source such that 81 parameters along with 1461 of test and training data are considered for property valuation. Users who are going to sell the property can get the accurate values based on this regression prediction. Users require no intermediate person (broker) to sell in the entity.

The Methodology of this project primarily covers four steps:

1.Importing all the required Python packages namely Numpy , Pandas , Sklearn , Pickle and Flask.

2.Using Sklearn package create a Linear Regression Model and using pandas library import the Dataset.

3.Using Pickle library save the created model and continue performing training and testing data using the linear regression model.

4.Peidict the sale price of using Linear regression Model.

**1.2 PROBLEM STATEMENT**

Users who are going to sell the property require an intermediate person (broker) to sell an entity. There's a chance of getting less price when compared with the actual price. And there are additional costs such as broker’s commissions.

**1.3 OBJECTIVES**

Users who are going to sell the property should get the accurate values based on this regression prediction. To make users independent from intermediate people (brokers) to sell the entity

**2.SYSTEM ANALYSIS**

## 2.1 REQUIREMENT ANALYSIS

2.1.1 PROJECT PERSPECTIVE

The ultimate perspective of any technology or innovation is to make the human life much easier to sustain either by providing a basic need or adding on a luxury to basic life. This project, House Price Prediction is of the latter type which adds on the luxury to human basic life, which makes him watch the highlights of cricket match directly without any discomfort or waiting for the fantastic events in the whole match.

2.1.2 PROJECT FEATURES

The features can be considered as the independency, Ease-to-use, Interactive & Responsive.

1. INDEPENDENCY
2. EASE-TO-USE
3. INTERACTIVE & RESPONSIVE

* Independency can be viewed as the extent to which the project runs on different devices i.e., on various configurations of devices without any support from third party helping tools or libraries and if support is must from third party, then minimising this support depends on the extent of independency of a project. However, this project is dependent to some extent on other libraries like scikitlearn, pickle, pandas but these are third party libraries instead they are official libraries to provide very important functionalities to the project.
* Ease-to-Use is the feature where the end user can feel comfortable in making use of this project. As this is a project not a product, implementation, functionality and efficiency are considered instead of user look and feel.
* As above said, House Price Prediction is a project not a product and hence the interactiveness and responsiveness are considered to a little extent. House Price Prediction is interacted through a web browser the responsiveness is depends on the connectivity of the user with the server.

**2.2 REQUIREMENT SPECIFICATIONS**

Requirement specifications are the necessities that should be satisfied for successful development and working the application. Such interface requirements are basically UI requirements, Hardware and software requirements

User Interfaces:

• Laptop/Desktop with a Web Browser

• CLI Support: Command Prompt, Terminal

Hardware Interfaces:

• Laptop, Desktop : Min i3 configuration,2.7GHZ

• RAM Capacity : 4GB

• Hard Disk : 100GB

• Programming Languages : Python, HTML, CSS

• Other Libraries Used : Pandas, Numpy, Flask, Sklearn, Pickle

• Operating Systems : Windows/Linux

**3.SYSTEM DESIGN**

**3.1 ARCHITECTURE**

Architecture describes the overall functionality of the project with all the modules specified in it. It gives a clear picture of the internal process of the projects. The architecture along with modules can be displayed as the following.

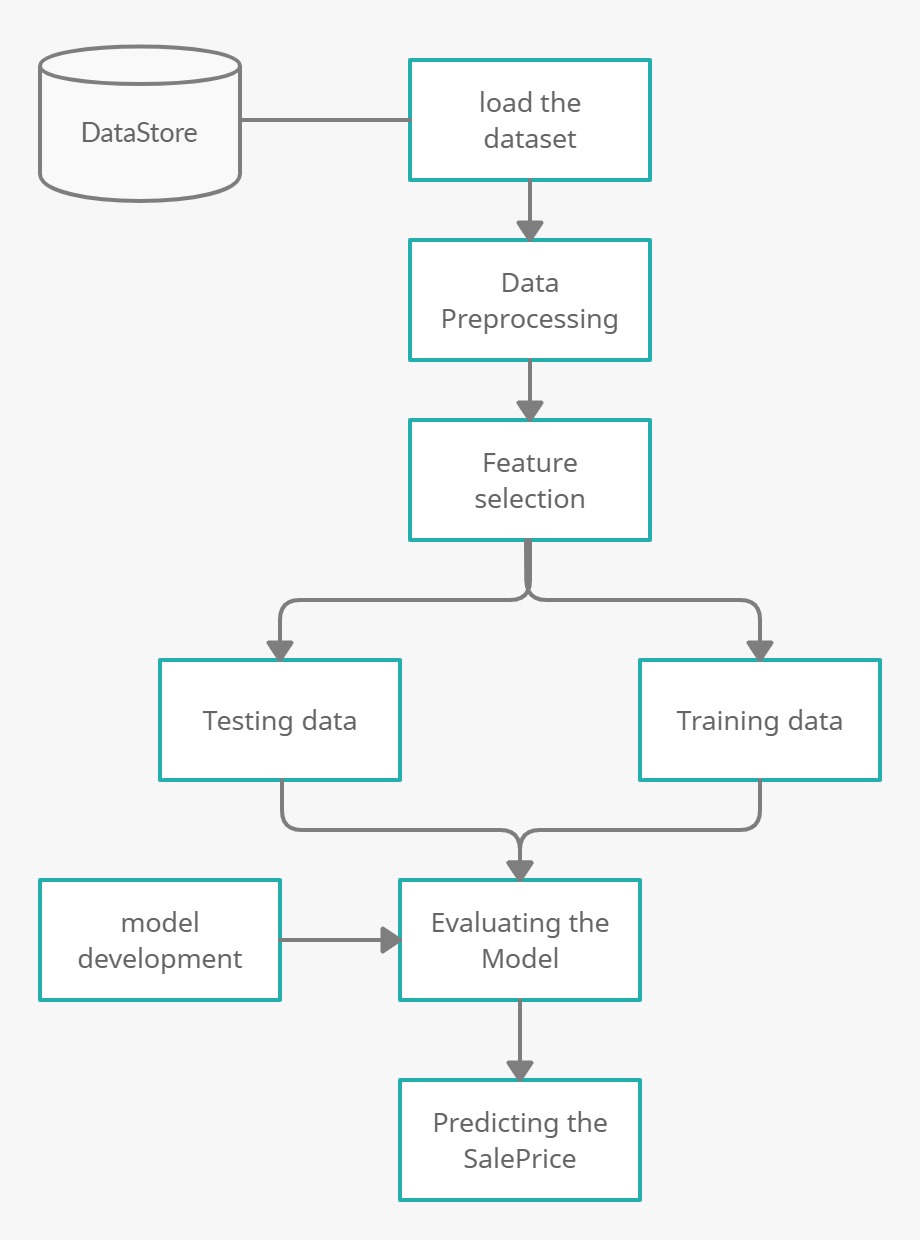


Fig 3.1 Architecture

**3.2 SEQUENTIAL FLOW OF THE PROJECT**

Sequential flow (or flow of control) is the order in which individual statements, instructions or function calls of an imperative program are executed or evaluated. The emphasis on explicit control flow distinguishes an imperative programming language from a declarative programming language.

The below figure shows the control flow and data flow of the project House Price Prediction

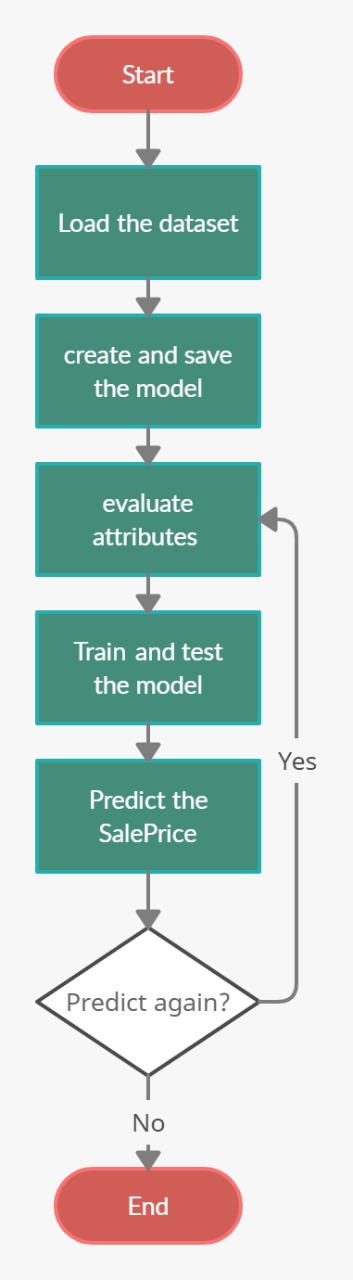


Fig 3.2 Sequential Flow Diagram

**3.3 HARDWARE AND SOFTWARE REQUIREMENTS**

HARDWARE

▪ Laptop, Desktop : Min i3 configuration,2.7GHZ

▪ RAM Capacity : 4GB

▪ Hard Disk : 100GB

SOFTWARE

▪ Programming Language : Python

▪ Operating Systems : Windows/Linux

▪ Platform : IDLE

▪ Web Browser : Google Chrome/Microsoft Edge

▪ Other Libraries Used : Scikit-learn, Numpy, Pandas, Pickle, Flask

**4. FEASIBILITY STUDY**

A feasibility study is an evaluation of a proposal designed to determine the difficulty in

carrying out a designated task. Generally, a feasibility study precedes technical development

and project implementation. In other words, a feasibility study is an evaluation or analysis of

the potential impact of a proposed project.

**4.1 TYPES OF FEASIBILITY:**

**➢ TECHNOLOGY AND SYSTEM FEASIBILITY:**

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

**➢ ECONOMIC FEASIBILITY:**

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

**➢ LEGAL FEASIBILITY:**

Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local data protection acts.

**➢ OPERATIONAL FEASIBILITY:**

Is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

**➢ SCHEDULE FEASIBILITY:**

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed it a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is.

OUTPUT: The feasibility study outputs the feasibility study report, a report detailing

the evaluation criteria, the study findings, and the recommendations.

# 5.IMPLEMENTATION TOOLS

## 5.1 PYTHON

Python was created in early 1990s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor of a language called ABC.

Guido remains python’s principle author, although it includes many contributions from others. Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use python and see almost immediate gains in productivity and lower maintenance costs.

**About python:**

Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains. Some of its key distinguish features include:

* Very clear, readable syntax
* Strong introspection capabilities
* Intuitive object orientation
* Natural expression of procedural code
* Full modularity, supporting hierarchical packages
* Exception-based error handling
* Very high-level dynamic data types
* Extensive standard libraries and third-party modules for virtually every task
* Extensions and modules easily written in C, C++ (or java for python, or .NET languages for python)
* Embeddable within applications as a scripting interface

**Python is Interpreted:**

Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

**Python is Interactive:**

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented:**

Python supports object-Oriented style or technique of programming that encapsulates code within objects.

**Python is a Beginner’s Language:**

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**Python runs everywhere:**

Python is available for all major operating systems: Windows, Linux/Unix, OS2, Mac, Amiga, among others. There are even versions that run on .NET and the java virtual machine. You’ll be pleased to know that the same source code will run unchanged across all implementations.

**Python is friendly and easy to learn:**

The python newsgroups known as one of the friendliest around. Python also comes with complete documentation, both integrated into the language and as separate webpages.

**Python is Open:**

The python implementation is under an open source licence that makes it freely usable and distributable, even for commercial use. The python licence is administered by the Python Software Foundation.

The “Python library” contains several different kinds of components. It contains data types that would normally be considered part of the “core” of a language, such as numbers and lists. For these types, the Python language core defines the form of literals and places some constraints on their semantics, but does not fully define the semantics. (On the other hand, the language core does define semantic properties like the spelling and priorities of operators.)

The library also contains built-in functions and exceptions-objects that can be used by all the python code without the need of an import statement. Some of these are defined by the core language, but many are not essential for the core semantics and are only described here.

The bulk of the library, however, consists of a collection of modules. There are many ways to dissect this collection. Some modules are written in C and built in to the Python interpreter; others are written in python and imported in source form. Some modules provide interfaces that are highly specific to Python, like printing a stack trace; some provide interfaces that are specific to particular operating systems, such as access to specific hardware; others provide interfaces that are specific to a particular application domain, like the World Wide Web. Some modules are available in all versions and ports of Python; others are only available when the underlying system supports or requires them; yet others are available only when a particular configuration option was chosen at the time when Python was compiled and installed.

**5.2 DATASET DESCRIPTION**

The dataset used in this project is Boston Housing Data, a preprocessed dataset containing 81 attributes and 1460 rows. The featured attributes used to train the model are as follows:

OverallQuall : Overall Quality of the house in rating upto 10

GrLiveArea : Ground Area of the house in square feets

GarageCars : Number of cars that can fit in the garage

GarageArea : Ground area of the garage in square feets

YearBuilt : Year in which the house is built

YearRemodAdd : Year in which the house has remodified or repaired

OverallCond : Overall Condition of the house in rating upto 10

SalePrice : Price for which the house has sold

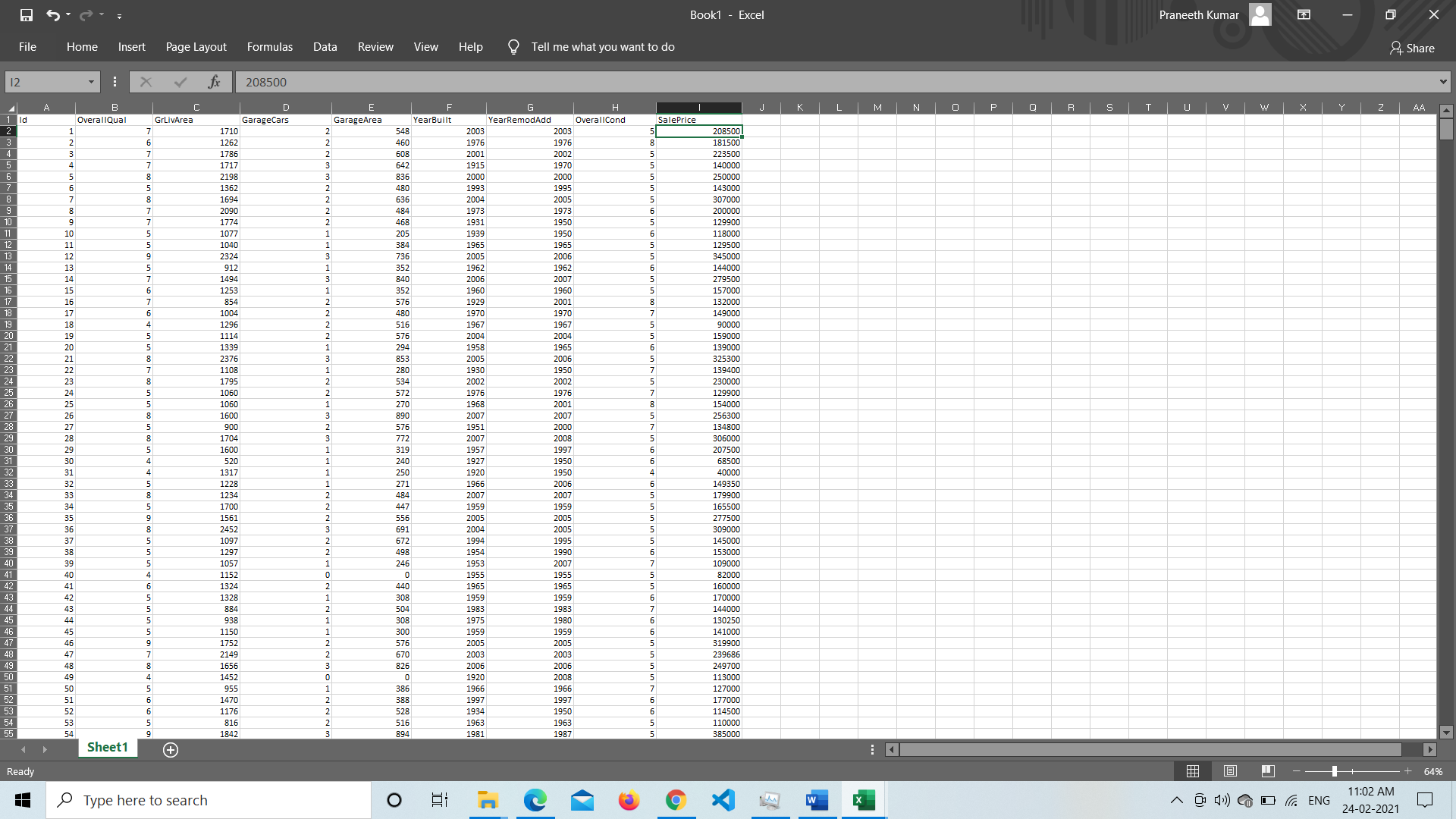


Fig 5.1 Dataset

## 5.3 MACHINE LEARNING

**Machine learning** (**ML**) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of [artificial intelligence.](https://en.wikipedia.org/wiki/Artificial_intelligence) Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data"](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision,](https://en.wikipedia.org/wiki/Computer_vision) where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

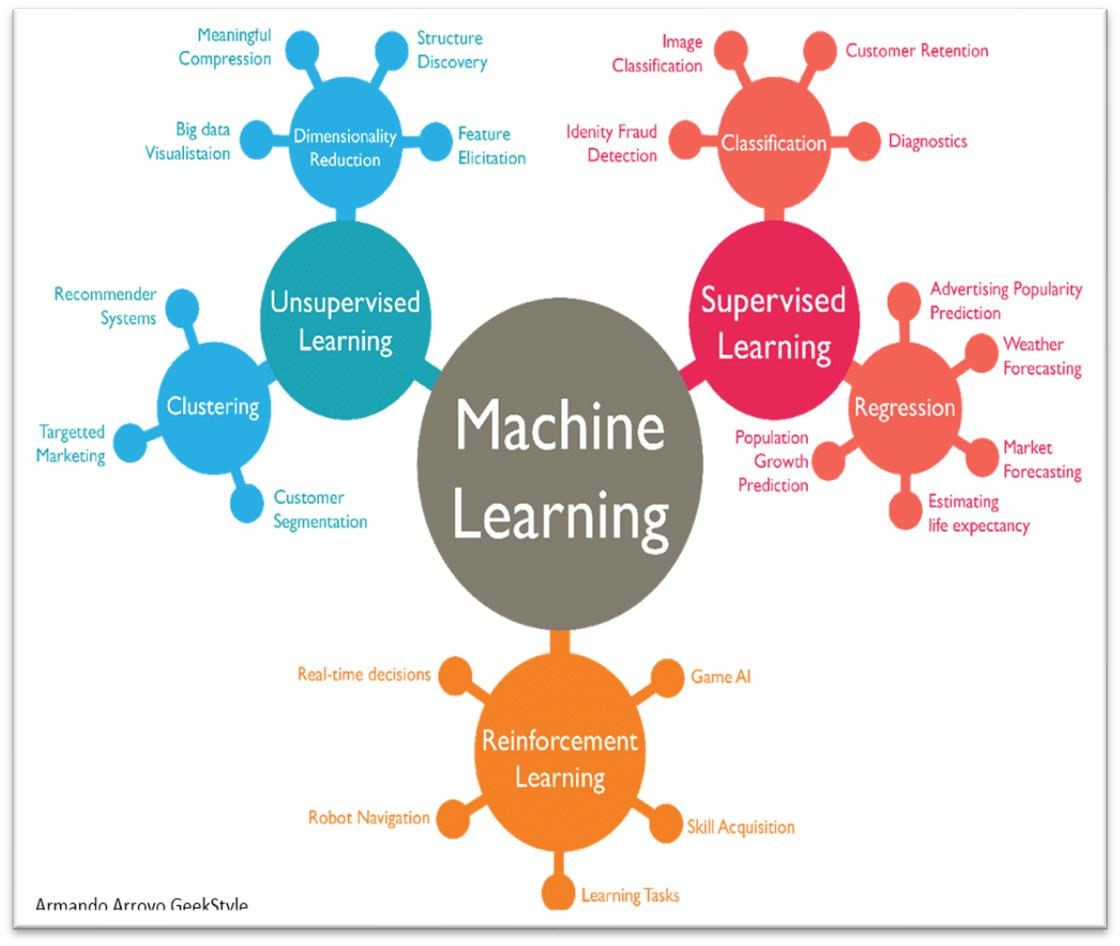
It is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

Machine learning algorithms are often categorized as supervised, unsupervised and reinforcement learning.

* **Supervised machine learning** algorithms can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
* **Unsupervised machine learning** algorithms are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.
* **Reinforcement machine learning** algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance.

Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Figure 5.2: Showing the different types of Machine Learning Algorithms



**Linear Regression**

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables (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. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

* If the goal is prediction, forecasting, or error reduction, [clarification needed] linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.
* If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

Linear regression models are often fitted using the least squares approach, but they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other norm (as with least absolute deviations regression), or by minimizing a penalized version of the least squares cost function as in ridge regression (L2-norm penalty) and lasso (L1-norm penalty). Conversely, the least squares approach can be used to fit models that are not linear models. Thus, although the terms "least squares" and "linear model" are closely linked, they are not synonymous.

## 5.4 FLASK

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

**Features**

* Development server and debugger
* Integrated support for unit testing
* RESTful request dispatching
* Uses [Jinja](https://en.wikipedia.org/wiki/Jinja_(template_engine)) templating
* Support for secure cookies (client side sessions)
* 100% [WSGI](https://en.wikipedia.org/wiki/WSGI) 1.0 compliant
* [Unicode](https://en.wikipedia.org/wiki/Unicode)-based
* Extensive documentation
* [Google App Engine](https://en.wikipedia.org/wiki/Google_App_Engine) compatibility
* Extensions available to enhance features desired

## 5.5 SCIKIT LEARN

Scikit-learn (formerly scikit.learn and also known as sklearn) 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.

Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas data frames, scipy, and many more.

**5.6 PANDAS**

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

Pandas is mainly used for data analysis. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features.

**Features**

* Data Frame object for data manipulation with integrated indexing.
* Tools for reading and writing data between in-memory data structures and different file formats.
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and subsetting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.
* Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
* Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
* Provides data filtration.

**5.7 NUMPY**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops, using NumPy. Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

**5.8 MATPLOTLIB**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since, then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

**5.9 HTML & CSS**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

**5.10 PICKLE**

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can converts the byte stream (generated through pickling) back into python objects by a process called as unpickling.

In real world scenario, the use pickling and unpickling are widespread as they allow us to easily transfer data from one server/system to another and then store it in a file or database

**6. EXPERIMENTAL SETUP**

Experimental Setup describes the entire process to implement this project as follows.

download and install Python https://www.python.org/downloads/, edit environmental variables

and open command prompt and install Numpy, Flask, Sklearn, Pickle, Pandas using PIP command .

To install the Numpy package, use the “pip install numpy” command.

To install the Flask package, use the “pip install flask:” command.

To install the sklearn package, use the “pip install scikit-learn” command.

To install the Pickle package, use the “pip install pickle-mixin” command.

To install the Pandas package, use the “ pip install pandas” command.

After completion of installation, now download and install Google Chrome https://www.google.com/intl/en\_in/chrome/ or any other web Browser.

**7. SCREENS AND REPORT**

**7.1 CODE SCREENS**

The below code shows the python code that loads the dataset and creates a regression model then train and tests the model using training and testing datasets respectively, After that it fits the model using training data and finally predict the SalePrice of the input it read earlier.



Fig 7.1 app.py code

**7.2 EXECUTION PROCESS**

Now, open command prompt and go to the folder containing the Project and execute app.py using “python app.py” command and wait for a while until you get an IP address.

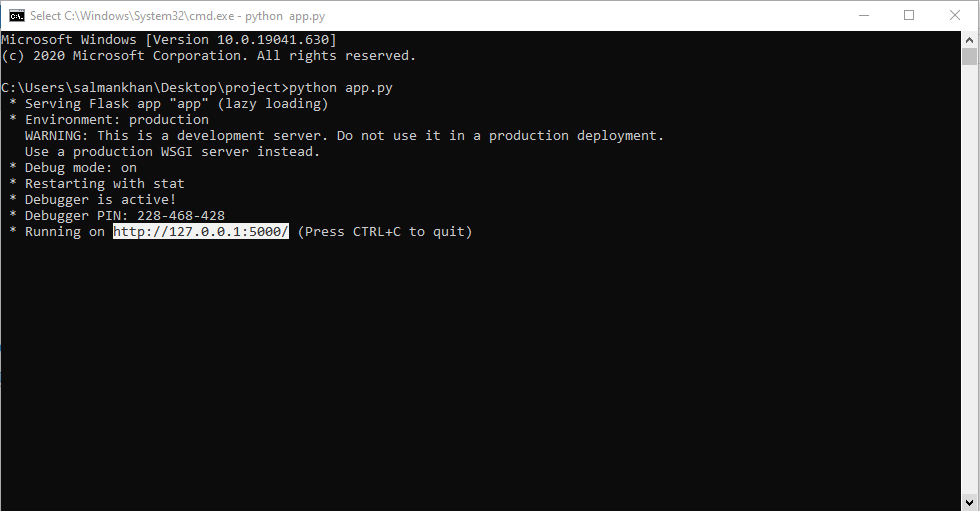


Fig 7.2: Command line execution

Now, copy and paste the IP address in Web Browser and type “/result” after IP address.

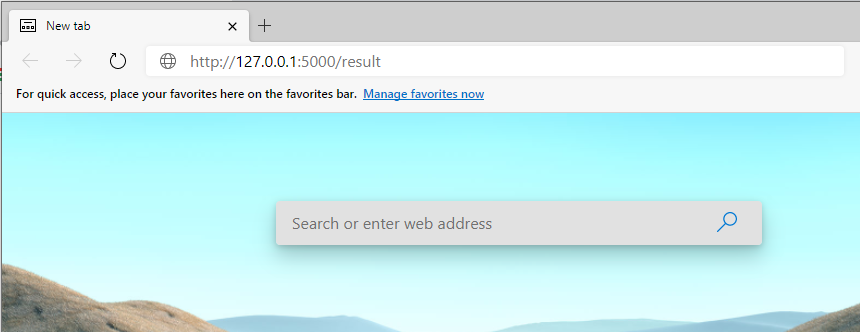


Fig 7.3: Using the IP address in web Browser

Now Click on “House Price Prediction” button.

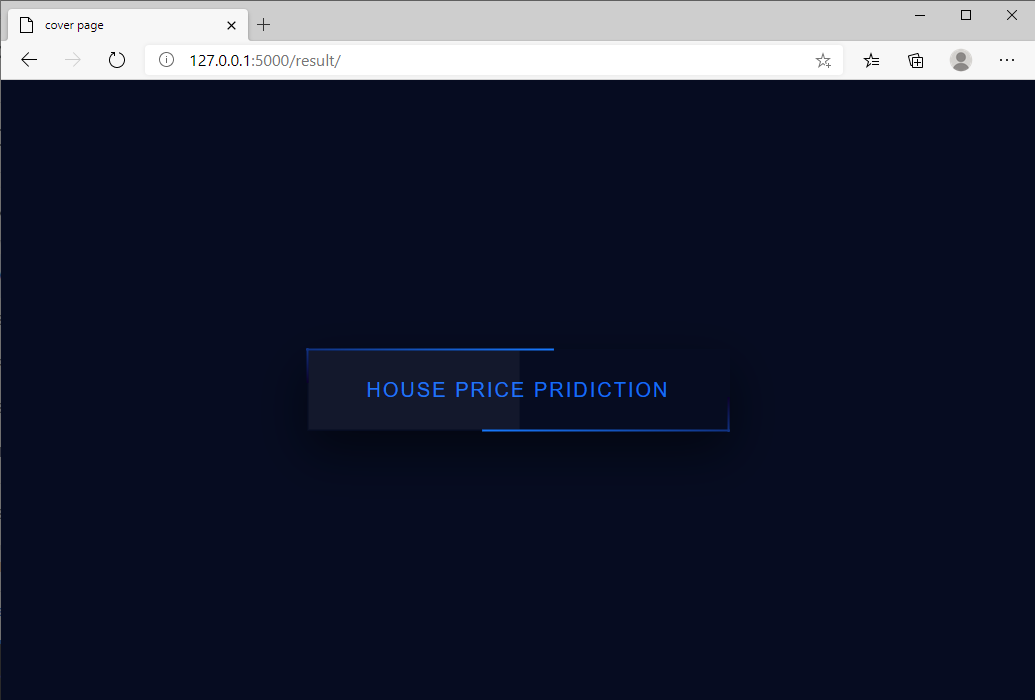


fig 7.4: Cover Page

Now click on Predict Button:

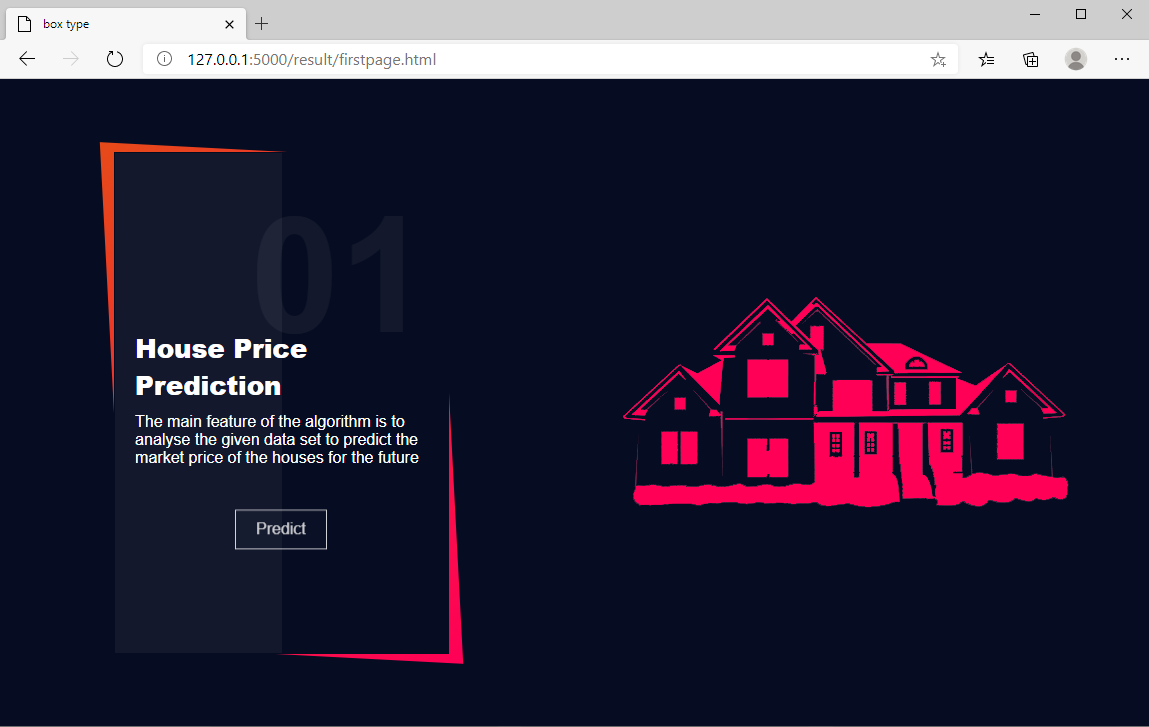


Fig 7.5 First page

Enter the required details which are required to have in a house:

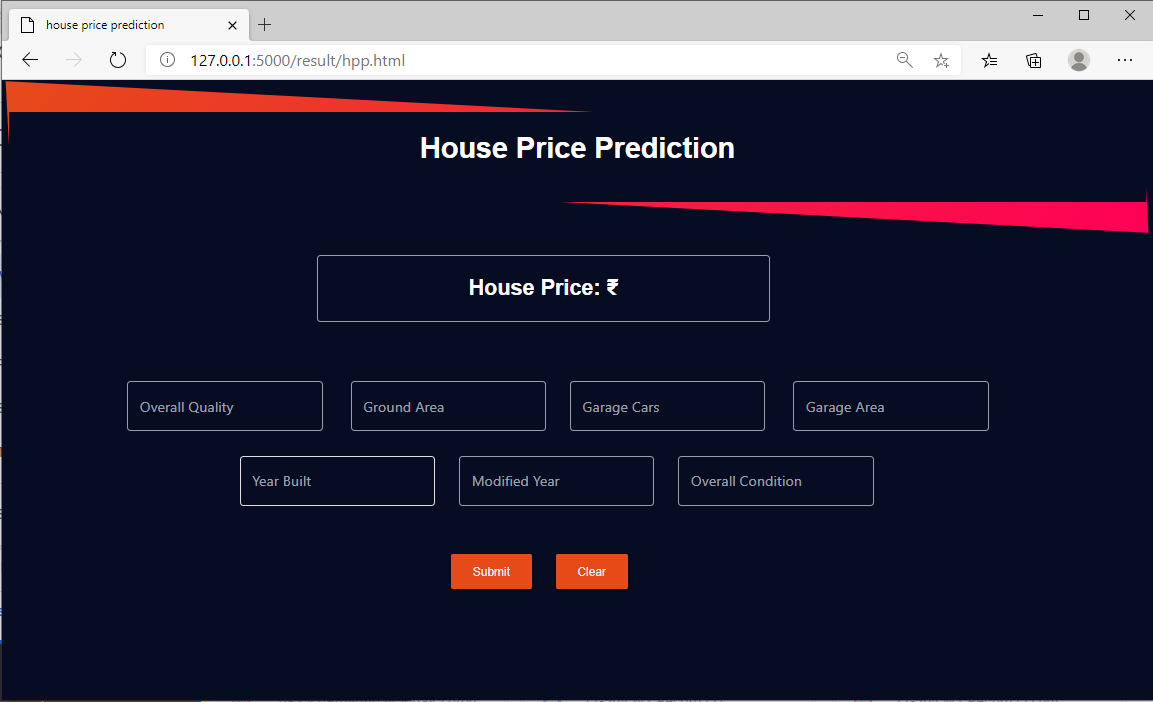


Fig 7.6: Provide Input

Now click on Submit button to view the predicted sale price:

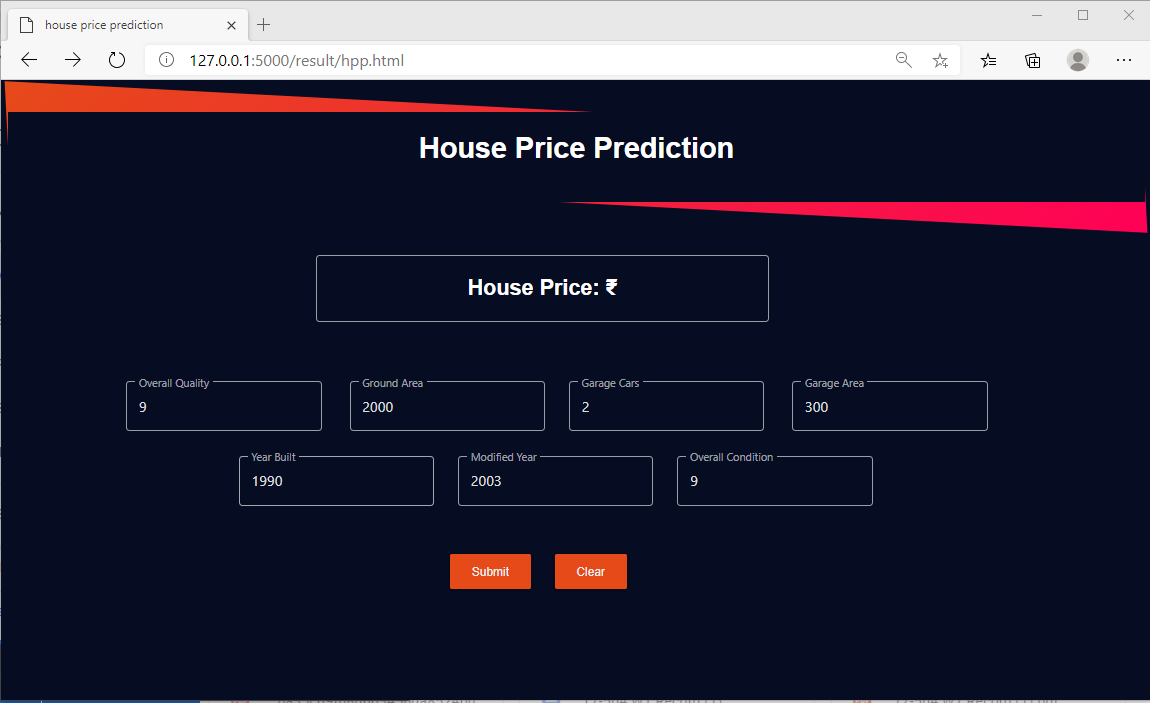


Fig 7.7 Predicted result

**8. TESTING**

**8.1 TESTING STRATEGIES**

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on even during maintenance also.

8.1.1 PSYCHOLOGY OF TESTING

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn’t work. Testing is the process of executing a program with the intent of finding errors.

8.1.2 TESTING OBJECTIVES

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

➢ Testing is a process of executing a program with the intent of finding an error.

➢ A successful test is one that uncovers an as yet undiscovered error.

➢ A good test case is one that has a high probability of finding error, if it exists.

➢ The tests are inadequate to detect possibly present errors.

➢ The software more or less confirms to the quality and reliable standards.

8.1.3 LEVELS OF TESTING

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below.

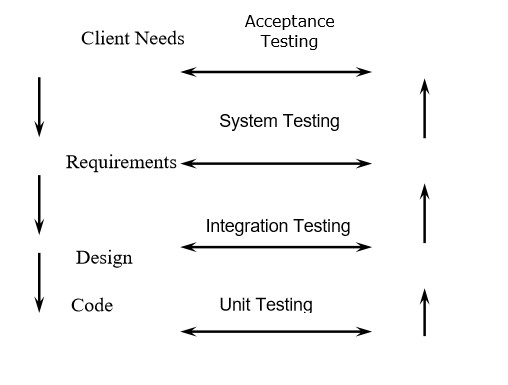


Fig 8.1: Levels of testing

8.1.4 SYSTEM TESTING

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

8.1.5 CODE TESTING

This strategy examines the logic of the program. To follow this method, we developed some test data that resulted in executing every instruction in the program and module i.e., every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect, two types of testing is performed on all systems.

8.1.6 TYPES OF TESTING

8.1.6.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Proficiency Checker, Readability Index Finder, Sentiment Analyzer. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error.

The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors. Through this error resulting from interaction between modules initially avoided.

8.1.6.2 LINK TESTING

Link testing does not test software but rather the integration of each module in system. The primary concern is the compatibility of each module. The Programmer tests where modules are designed with different parameters, length, type etc. 8.1.6.3 Integration Testing:

After the unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules, checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

8.1.6.4 SYSTEM TESTING

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire application has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

8.1.6.5 Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behaviour of the system; the internal logic of program is not emphasized.

In this project ‘Graphical User Interface handling input’ I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

8.1.6.6 WHITE BOX TESTING

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. Test has been done step wise for every piece of code, taking care that every statement in the code is executed at least once. The white box testing is also called Glass Box Testing.

A list of test cases, sample data have generated, which is used to check all possible combinations of execution paths through the code at every module level.

8.1.6.8 BLACK BOX TESTING

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block box that will take one input and generate output. Output for a given set of input combinations are forwarded to other modules

**8.2 TESTCASES**

As this doesn’t have any database or particular UI, there are no intended tests but the models which are built are tested by using validation strategies which has been covered already in Module description section and some test cases where input data handling are mentioned below.

**TEST REPORT 1:**

1. **Project Name:** HOUSE PRICE PREDICTION

1. **Module Name:** Prediction

1. **Unit Name:** Model Prediction

1. **Test Result**: SalePrice has been predicted for the given input

**TEST PLAN FOR HOUSE PRICE PREDICTION:**

**Unit id:** Model Performance

**Test case id:** 001

**Test type:** unit level testing

**PURPOSE:**

To check whether the created model is able to predict the house price for given input or not

Test Case:

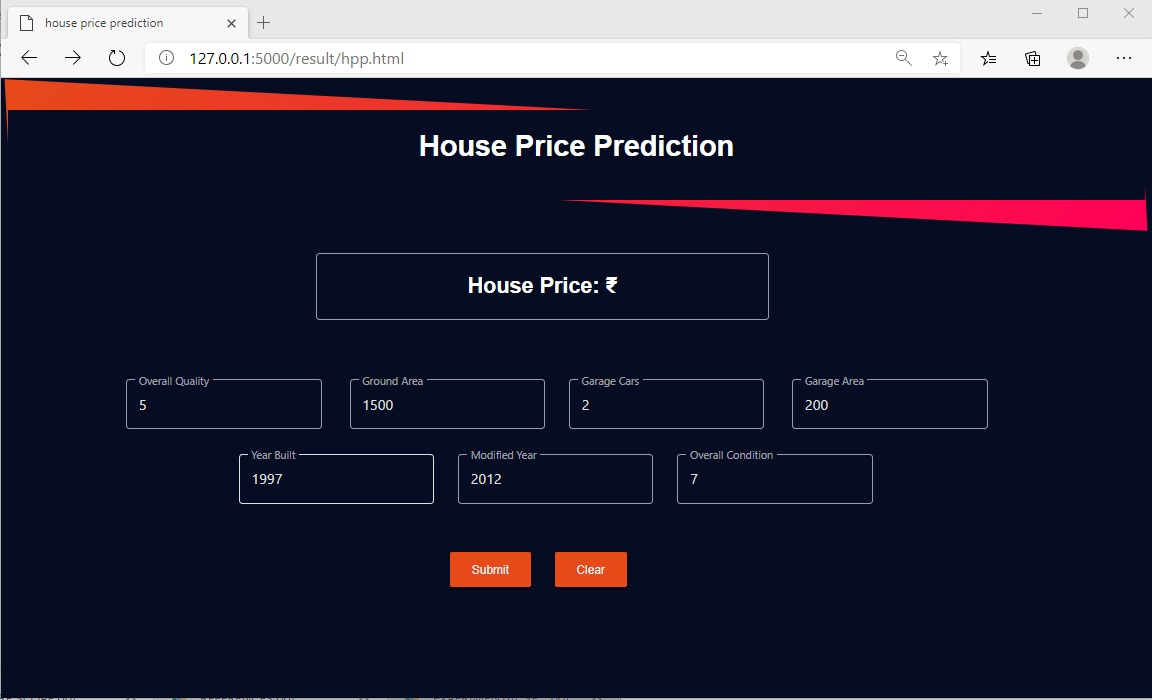


Fig 8.2: Test Case Input

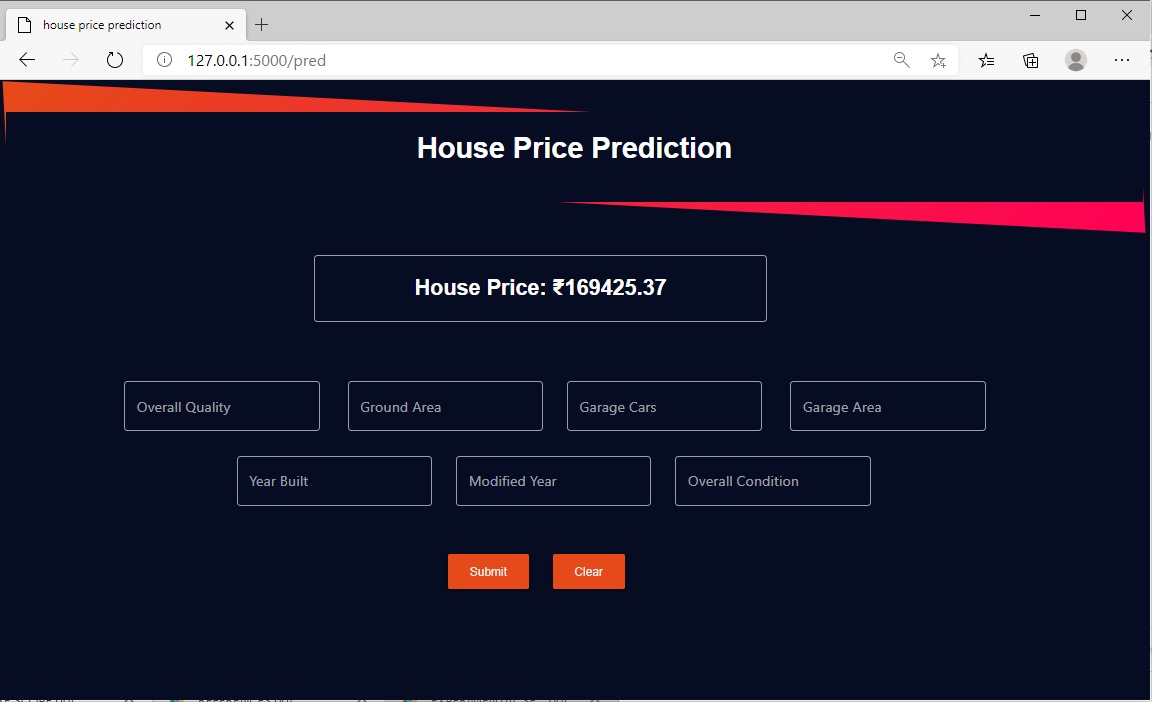


Fig 8.3: Test Case Result

**9.FUTURE SCOPE**

**SCOPE WITHIN THE SAME DOMAIN**

This project can have a huge impact on real estate. This project can make real estate

businesses much easier and helps users to have a better understanding of real estates and can

predict sales prices without contacting any intermediary person (broker).

**SCOPE IN DIFFERENT DOMAINS**

This project with fewer modifications can be used for other predictions like rainfall, heartbeat, entity failure, diseases predictions too. and can be also used in much more other domains.

**10.REFERENCES**

Python Reference

https://www.python.org/doc/

Scikit Learn reference

https://scikit-learn.org/stable/user\_guide.html

Pickle reference

https://docs.python.org/3/library/pickle.htm

Flask reference

https://flask.palletsprojects.com/en/1.1.x/

Numpy reference

https://numpy.org/doc/

Pandas reference

https://pandas.pydata.org/docs/

HTML reference

https://devdocs.io/html/

CSS reference

https://devdocs.io/css/

Linear Regression using Python

https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html