**CONTENTS**

**Candidate’s Declaration** i

**Acknowledgement** ii

**Abstract** iii

**Contents**

**List of Figures**

**List of Tables**

**List of Screens**

**List of Symbols**

**List of Abbreviations**

**1. INTRODUCTION**

1.1 Motivation 1

1.2 Problem Definition 2

1.3 Objective of Project 3

1.4 Limitations of Project 3

1.5 Organization of Documentation 3

**2. LITERATURE SURVEY**

2.1 Introduction 4

2.2 Existing System 5

2.3 Disadvantages of Existing system 5

2.4 Proposed System 6

**3. ANALYSIS**

3.1 Introduction 8

3.2 Software Requirement Specification 13

3.2.1 User requirements 13

3.2.2 Software requirements 14

3.2.3 Hardware requirements 25

3.3 Content Diagrams of Project 30

3.4 Algorithms and Flowcharts 31

**4. DESIGN**

4.1 Introduction 34

4.2 DFD / ER / UML diagram (any other project diagrams) 34

4.3 Module design and organization 40

**5. IMPLEMENTATION & RESULTS**

5.1 Introduction 47

5.2 Explanation of Key functions 47

5.3 Method of Implementation 67

5.3.1 Forms 69

5.3.2 Output Screens 74

5.3.3 Result Analysis 79

**6. TESTING & VALIDATION**

6.1 Introduction 80

6.2 Design of test cases and scenarios 80

6.3 Validation 83

**7. CONCLUSION & FUTURE WORK** 84

**8. REFERENCES**

**List of Figures**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Figure No.** | **Figure Name** | **Page No.** |
| 1 | 3.3.1 | Flowchart of prediction model | 6 |
| 2 | 4.2.1 | Sequence diagram for house price prediction | 9 |
| 3 | 4.2.2 | DFD leve0 for house price prediction | 10 |
| 4 | 4.2.3 | DFD leve1 for house price prediction | 10 |
| 5 | 4.2.4 | DFD for house price prediction | 11 |
| 6 | 4.2.5 | Abstract view of house price prediction model | 11 |
| 7 | 5.3.1.1 | Training data overview | 14 |
| 8 | 5.3.1.2 | Stream lit code for frontend | 14 |
| 9 | 5.3.1.3 | Deployment code for model | 15 |
| 10 | 6.2.1 | Code for splitting the training and testing  data | 18 |

**List of Tables**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Table No.** | **Table Name** | **Page No.** |
| 1 | 6.3.1 | Expected output vs resultant output | 20 |

**List of Screens**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Screen No.** | **Name** | **Page No.** |
| 1 | 5.3.2.1 | Web application screen | 16 |
| 2 | 5.3.2.1 | Web application screen2 | 16 |
| 3 | 6.3.1 | Output displaying the predicted house price | 19 |

**List of Abbreviations**

Fig – Figure

## Chapter 1 INTRODUCTION

## Motivation

The exponential growth of online platforms, social media, and e-commerce has led to an overwhelming amount of user-generated content, including product reviews, social media posts, blog comments, and news articles. Manual analysis of such vast volumes of textual data is time consuming and often impractical. Sentiment analysis provides an automated solution to this challenge by employing machine learning and NLP techniques to classify and analyze sentiments expressed in the text.

The main motive of this model are as follows:

* + - To categorize text into different sentiment classes, typically positive, negative, and neutral.
    - By automatically determining the sentiment of textual data, businesses and organizations can gain valuable insights into customer opinions, preferences, and satisfaction levels.

## Problem Definition

Twitter, being a popular microblogging platform, provides a vast amount of user-generated content, including opinions, feedback, and sentiments on a wide range of topics. Analyzing sentiments in real-time Twitter data can provide valuable insights into public opinion, customer feedback, social trends, and brand perception. Sentiment analysis, also known as opinion mining, on real-time Twitter data is a powerful technique that involves the automatic extraction and classification of sentiments or emotions expressed in tweets. Its applications span across industries, including marketing, customer service, reputation management, and academic research. By leveraging advanced machine learning techniques, sentiment analysis provides organizations with invaluable insights into customer opinions, preferences, and market trends, facilitating data-driven decision-making and enhancing overall user experiences.

## Objective of Project

The objective of sentiment analysis on real-time Twitter data is to automatically process and interpret the sentiments expressed in tweets and categorize them as positive, negative, or neutral. This enables businesses, organizations, and researchers to gain a deeper understanding of how people feel about specific products, services, events, or topics in real-time.

The unique characteristics of Twitter data present both challenges and opportunities for sentiment analysis. Tweets are typically short, often limited to 280 characters, which requires specialized techniques to handle the brevity and informality of the language. Additionally, Twitter data is characterized by the use of hashtags, abbreviations, slang, emoticons, and user mentions, making the sentiment analysis task more complex.

## Limitations of Project

Sentiment analysis on real-time Twitter data comes with its own set of challenges Twitter data is characterized by its brevity, which often leads to informal language, abbreviations, slang, and misspellings. Dealing with such noisy and informal text poses challenges in accurately extracting sentiments and understanding the intended meaning. Tweets are often context-dependent, and the sentiment expressed can be subjective or ambiguous. Understanding the context and disambiguating sentiments based on limited text can be challenging, especially when sarcasm, irony, or figurative language is used. Emoticons and emojis are commonly used in tweets to express emotions. Incorporating these non-textual elements into sentiment analysis poses challenges, as they carry sentiment information that needs to be properly interpreted and considered during sentiment classification.

## Organization of Documentation

The organization of Documentation is as follows: Section 1 introduces the area of study. Section 2 gives an overview of the Literature. Section 3 gives an overview of the algorithms. shows the followed methods in this study, Section 4 presents the methods that are being used in the experiment in addition to the theoretical findings in addition to, the design of the experiment. Section 5 shows the experimental implementation process and the experiment results followed by a discussion in section 6. Finally, Section 7 concludes with remarks and hints about future work.

## Chapter 2 LITERATURE SURVEY

## Introduction

The following section summarizes the significant factor regarding the literacy of existing approaches.

G.Kavitha, B.Saveen, Nomaan Imtiaz suggested Discovering Public Opinions by Performing Sentimental Analysis on Real Time Twitter Data. The objective of this work is to generate a sentimental report regarding the hashtag specific data. They have used Hadoop tool to do this, by using the map reduce function for the sentiment analysis report was generated on the hashtag based data. “IPL” based hashtag data was used and collected using flume and hive for data ingestion and querying They have obtained a result labelled as positive 88%, negative 4%, and neutral 8%.Disadvantages of this approach is that only hashtag type data was collected whereas the replies to, from a user is missing.

Andleeb Aslam and his team suggested doing Opinion Mining Using Live Twitter Data. The objective was to tokenize and pre-process the data and further 23pprox. 23ng it to get the sentiments of the tokens. The inverse matrix document was used for this analysis. They have taken a sample of 1000 tweets related to the topic “CEPC” to generate the report. The results obtained are positive 58% and negative 38% and neutral 7%. Disadvantages of this approach is that Accuracy can decrease when a large limit is set, as there was no learning based approach used.

Ramandeep Singh Kathuria and team has worked on Real Time Sentiment Analysis On Twitter Data Using Deep Learning(Keras) The objective was to develop a sentiment analysis by using deep learning models. The methodology involves the usage of the NLTK, and machine learning algorithms like naïve bayes and deep learning algorithms like keras and tensor flow along with word2vec method They extracted the live data from twitter by using the twitter API which can be only accessed by twitter developer account. They obtained a accuracy of 86% when their model was evaluated. The disadvantages of this approach is that The classifiers and the deep learning models used requires a high computing resources, which makes the model unfit for huge datasets.

## Existing System

The existing system for Twitter sentiment analysis involves collecting tweets related to a specific topic or keyword through a dataset, preprocessing the data to normalize it, applying sentiment analysis techniques to determine the sentiment polarity of each tweet, evaluating the results, and visualizing them for interpretation. The approach also relies on classifiers and deep learning models that require high computing resources. This makes the model unsuitable for handling huge datasets, as the computational requirements could become a limiting factor for its effectiveness and deployment.

## Disadvantages of Existing system

The existing system only collected hashtag- type data, which means that the replies to and from users are missing. This limitation restricts the comprehensiveness of the data collected and may lead to an incomplete understanding of the overall sentiment or opinions expressed.

The use of a dataset taken from Kaggle instead of live data for analysis is another disadvantage. Live data would provide more up-to-date and real-time insights, whereas using a pre-existing dataset can introduce potential biases and inaccuracies that might not reflect the current situation.

Another disadvantage is the use of deep learning, and CNN models cannot be efficiently applied to huge datasets without significant preprocessing, as it may lead to overfitting or underfitting. This limitation restricts the scalability and practicality of the approach when dealing with large amounts of data.

## Proposed System

The objective of the project “Real-Time Twitter Sentiment Insights: A Web-Based Analysis” is to develop a robust and efficient system that can perform sentiment analysis on real-time Twitter data. The specific objectives include:

1. Collecting Real-Time Twitter Data: Implement mechanisms to retrieve real-time tweets from the Twitter API related to a specific topic or event of interest.

2. Preprocessing and Cleaning of Data: Apply appropriate text preprocessing techniques to clean the collected tweets by removing noise, special characters, URLs, and irrelevant information.

3. Sentiment Analysis Classification: Utilize machine learning or natural language processing techniques to classify the cleaned tweets into positive, negative, or neutral sentiment categories.

4. Real-Time Analysis and Visualization: Perform sentiment analysis in real-time and visualize the sentiment distribution using graphs, charts, or other visualization techniques.

6. Insights and Applications: Extract meaningful insights from the sentiment analysis results to gain a better understanding of public sentiment and its impact on specific topics, events, or brands. Explore potential applications of sentiment analysis in areas such as market research, brand reputation management, and public opinion analysis.

By achieving these objectives, the project aims to provide a valuable tool for analyzing the sentiment of real-time Twitter data, enabling users to gain actionable insights from the vast amount of information available on the platform.

## Chapter 3 ANALYSIS

## Introduction:

Analysis is the process of dissecting something into its component elements in order to understand what each one does and how they interact with one another.

Machine learning is a subfield of Artificial Intelligence (AI) that works with algorithms and technologies to extract useful information from data. Machine learning methods are appropriate in big data since attempting to manually process vast volumes of data would be impossible without the support of machines. Machine learning in computer science attempts to solve problems algorithmically rather than purely mathematically. Therefore, it is based on creating algorithms that permit the machine to learn. However, there are two general groups in machine learning which are supervised and unsupervised. Supervised is where the program gets trained on pre- determined set to be able to predict when a new data is given. Unsupervised is where the program tries to find the relationship and the hidden pattern between the data.

## Software Requirement Specification

### User requirements

**Operating System**: Windows 8/8.1/10, Linux and Mac

### Software requirements

* + - * Python 3.2 and above
      * tweepy, textbold
      * Matplotlib
      * flask
    1. **Hardware requirments**
* Processor: 2 gigahertz (GHz) or faster processor.
* RAM: 4 gigabytes (GB) for 32-bit or 8 GB for 64-bit.
* Hard disk space: =16GB.

**RAM:** 4 GB. If your PC has less than 4GB of memory, there are sometimes options for upgrading to get additional RAM. As the API calls which model goes through requires 4GB and above.

## Content Diagrams of Project

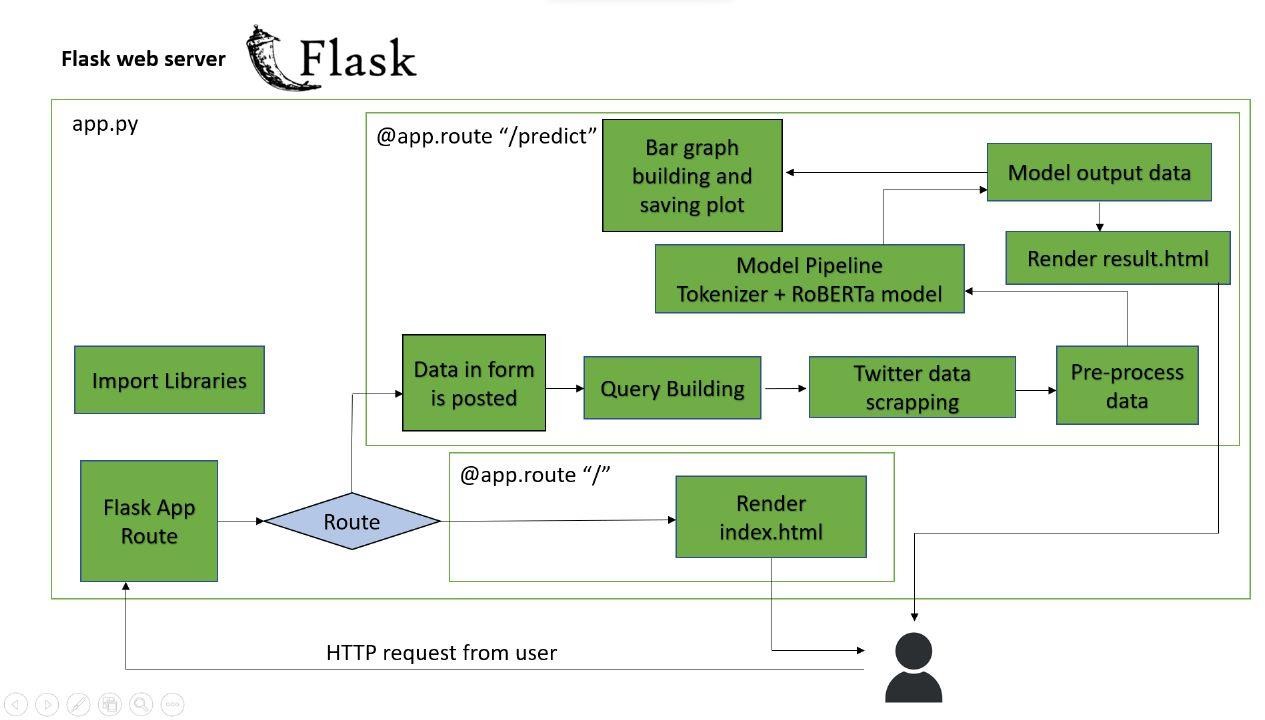


Fig. 3.3.1: Flowchart of prediction model

## Algorithms and Flowcharts

### Linear Regression:

Linear Regression is a supervised technique used to estimate the relationship between one dependent variable and more than one independent variables. Identifying the correlation and its cause-effect helps to make predictions by using these relations. To estimate these relationships, the prediction accuracy of the model is essential; the complexity of the model is of more interest. However, Linear Regression is prone to many problems such as multicollinearity, noises, and overfitting, which effect on the prediction accuracy. Regularised regression plays a significant part in Linear Regression because it helps to reduce variance at the cost of introducing some bias, avoid the overfitting problem and solve ordinary least squares (OLS) problems. There are two types of regularisation techniques L1 norm (least absolute deviations) and L2 norm (least squares). L1 and L2 have different cost functions regarding model complexity

### Lasso Regression:

Least Absolute Shrinkage and Selection Operator (Lasso) is an L1- norm regularised regression technique that was formulated by Robert Tibshirani in 1996. Lasso is a powerful technique that performs regularisation and feature selection. Lasso introduces a bias term, but instead of squaring the slope like Ridge regression, the absolute value of the slope is added as a penalty term.

Lasso is defined as: 𝐿 = 𝑀𝑖𝑛(𝑠𝑢𝑚 𝑜𝑓 𝑠𝑞𝑢𝑎𝑟𝑒𝑑 𝑟𝑒𝑠𝑖𝑑𝑢𝑎𝑙𝑠 + 𝛼 ∗

|𝑠𝑙𝑜𝑝𝑒|) (1) Where 𝑀𝑖𝑛(𝑠𝑢𝑚 𝑜𝑓 𝑠𝑞𝑢𝑎𝑟𝑒𝑑 𝑟𝑒𝑠𝑖𝑑𝑢𝑎𝑙𝑠) is the Least Squared Error, and 𝛼 ∗ |𝑠𝑙𝑜𝑝𝑒| is the penalty term. However, alpha 𝑎 is the tuning parameter which controls the strength of the penalty term. In other words, the tuning parameter is the value of shrinkage. |𝑠𝑙𝑜𝑝𝑒| is the sum of the absolute value of the coefficients

Cross-validation is a technique that is used to compare different machine learning algorithms in order to observe how these methods will perform in practice. Cross-validation method divides the data into blocks. Each block at a time will be used for testing by the algorithm, and the other blocks will be used for training the model. In the end,

the results will be summarised, and the block that performs best will be chosen as a testing block. However, 𝛼 is determined 4 by using cross-validation. When 𝛼 = 0, Lasso becomes Least Squared Error, and when 𝛼 ≠ 0, the magnitudes are considered, and that leads to zero coefficients. However, there is a reverse relationship between alpha 𝑎 and the upper bound of the sum of the coefficients 𝑡. When 𝑡 → ∞, the tuning parameter 𝑎 = 0. Vice versa when 𝑡 = 0 the coefficients shrink to zero and 𝑎 → ∞. Therefore, Lasso helps to assign zero weights to most redundant or irrelevant features in order to enhance the prediction accuracy and interpretability of the regression model. Throughout the process of features selection, the variables that still have non-zero coefficients after the shrinking process are selected to be part of the regression model. Therefore, Lasso is powerful when it comes to feature selection and reducing the overfitting.

### Ridge Regression:

The Ridge Regression is an L2-norm regularised regression technique that was introduced by Hoerl in 1962. It is an estimation procedure to manage collinearity without removing variables from the regression model. In multiple linear regression, the multicollinearity is a common problem that leads least square estimation to be unbiased, and its variances are far from the correct value. Therefore, by adding a degree of bias to the regression model, Ridge Regression reduces the standard errors, and it shrinks the least square coefficients towards the origin of the parameter space. Ridge formula is: 𝑅 = 𝑀𝑖𝑛(𝑠𝑢𝑚 𝑜𝑓

𝑠𝑞𝑢𝑎𝑟𝑒𝑑 𝑟𝑒𝑠𝑖𝑑𝑢𝑎𝑙𝑠 + 𝛼 ∗ 𝑠𝑙𝑜𝑝𝑒 2 ).Where 𝑀𝑖𝑛(𝑠𝑢𝑚 𝑜𝑓 𝑠𝑞𝑢𝑎𝑟𝑒𝑑 𝑟𝑒𝑠𝑖𝑑𝑢𝑎𝑙𝑠) is the Least Squared Error, and 𝛼 ∗ 𝑠𝑙𝑜𝑝𝑒is the penalty term that Ridge adds to the Least Squared Error. When Least Squared Error determines the values of parameters, it minimises the sum of squared residuals. However, when Ridge determines the values of parameters, it reduces the sum of squared residuals. It adds a penalty term, where

𝛼 determines the severity of the penalty and the length of the slope. In addition, increasing the 𝛼 makes the slope asymptotically close to zero. Like Lasso, 𝛼 is determined by applying the Cross-validation method. Therefore, Ridge helps to reduce variance by shrinking parameters and make the prediction less sensitive.

**Chapter 4 DESIGN**

# INTRODUCTION

This section describes the working of the various modules in the model and the way in which they are communicating data with each other. This model is divided into two parts:

#### User Interface

* + - This is the page which displays all the data requirements and results of the data.

#### Back End

* + - In this all the analysis is done based on data set provided.

## UML Diagrams

#### Sequence Diagram

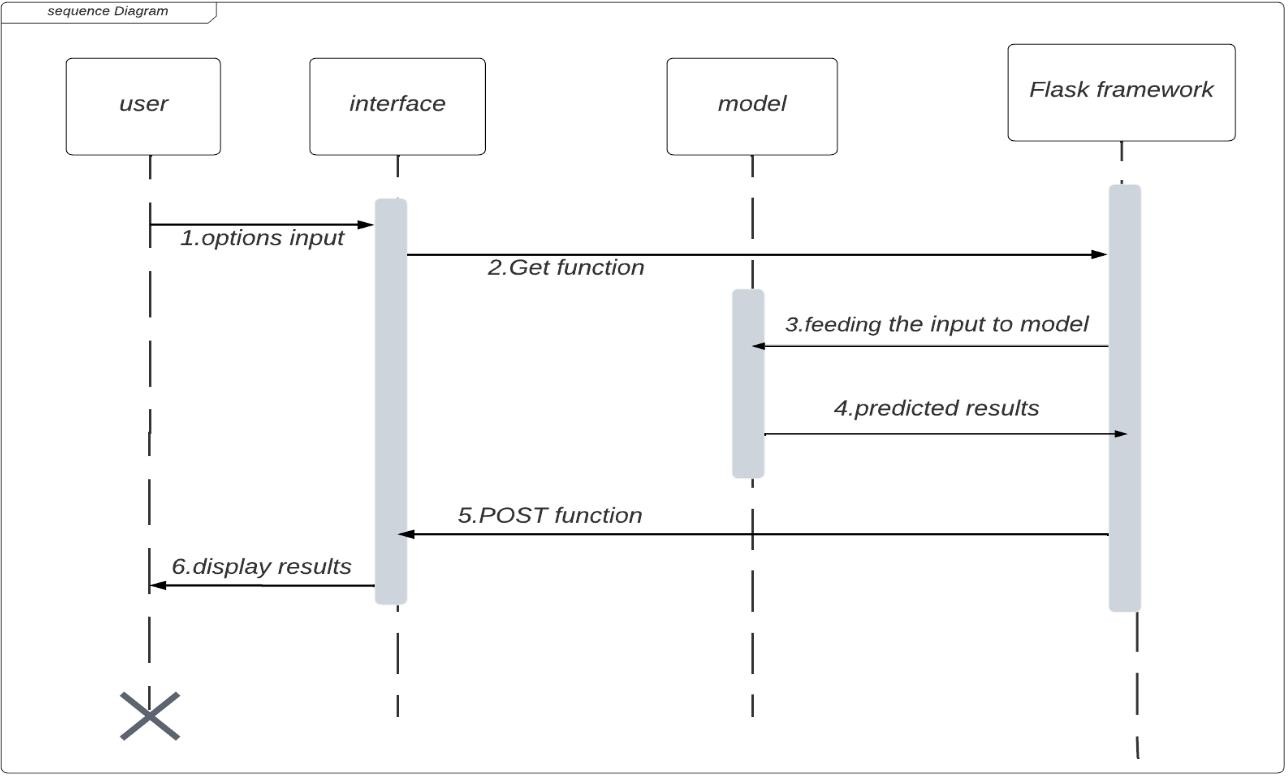


Fig. 4.2.1: Sequence diagram for house price prediction

#### DFD Diagrams

**Level 0**

**prediction model**

**OutPut**

**Input**

Fig. 4.2.2 DFD leve0 for house price prediction

#### Level 1

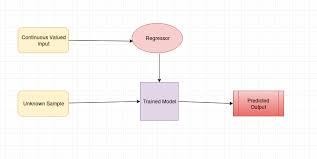


Fig. 4.2.3 DFD leve0 for house price prediction

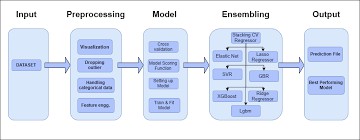


Fig. 4.2.4 DFD for house price prediction

#### Abstract view of model

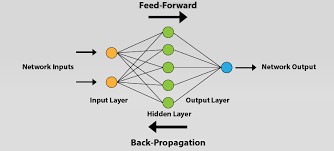


Fig. 4.2.5 abstract view of house price prediction model

## Module design and organization

#### Data Collection

In this module, we collect different datasets using predefined libraries in Python.

#### Data Refining

The data may have some null values and words in which machine

learning won’t accept the characters. It only accepts numeric data.

#### Training and Testing

We split our data into training and testing data using a test split from the sci-kit learn library.

#### Finding accuracy

We find the accuracy score and correlation matrix for the test target variable to the predicted values. We have used sklearn to import the accuracy score.

#### Saving model

We have used the best model from the all the tested models and used the best one with highest accuracy for predicting the values.

**Chapter 5 IMPLEMENTATION & RESULTS**

# INTRODUCTION

Implementation is one of the phases of the Software Development life Cycle. The implementation phase of software development involves the translation of design specifications source code and debugging, documentation and unit testing of the source code. In this project, we are going to develop a website which takes the input from users as features of the house. Based on the inputs we will predict the suitable crop to the farmer.

# EXPLANATION OF KEY FUNCTIONS

* In this project, the streamlit module is used to create a webpage for the machine learning model.
* Using python, the main code of the model to predict crops is executed.
* The Numpy package is used to take the inputs in multi- dimensional array form.
* Pickle package is used to store the files in bytes form.

# METHOD OF IMPLEMENTATION

### Approach

In this project, the total code is divided into 3 parts.

### Part 1 - trained\_model

It contains the main code of the model about how to pre-process the dataset. Dataset is divided into a training dataset and testing dataset and algorithm classifiers are used to get the results.

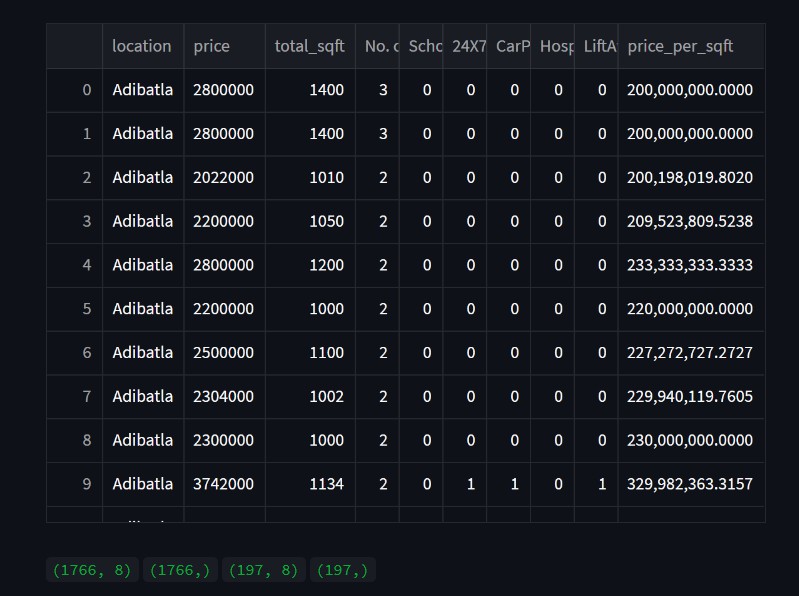


Fig. 5.3.1.1: Training data overview

### Part 2- predictive\_system.py

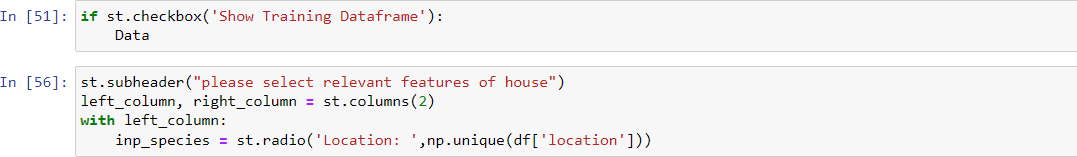
Using Stream we will create a user interface to print the result of the input. In this, we will dump the main code of the model using the pickle model.



Fig. 5.3.1.2: Stream lit code for frontend

#### Part 3- webapp.py

This module takes .py module which is used to run the webpage. Stream lit module is used to run the webpage.



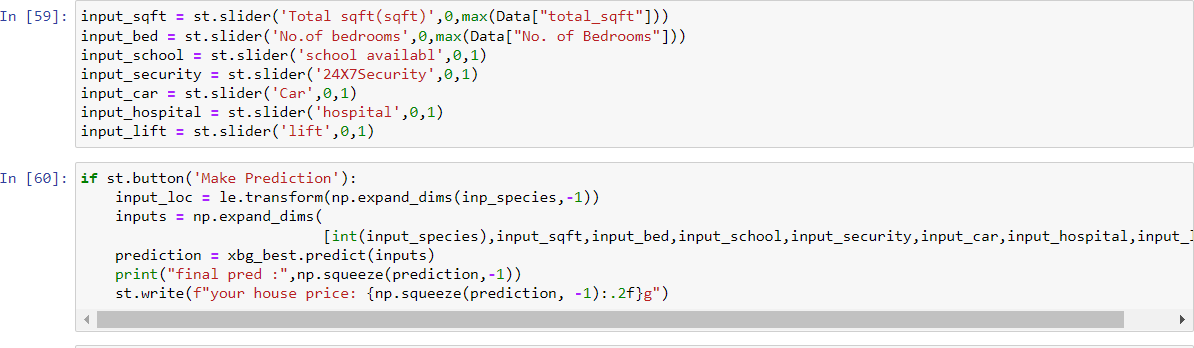
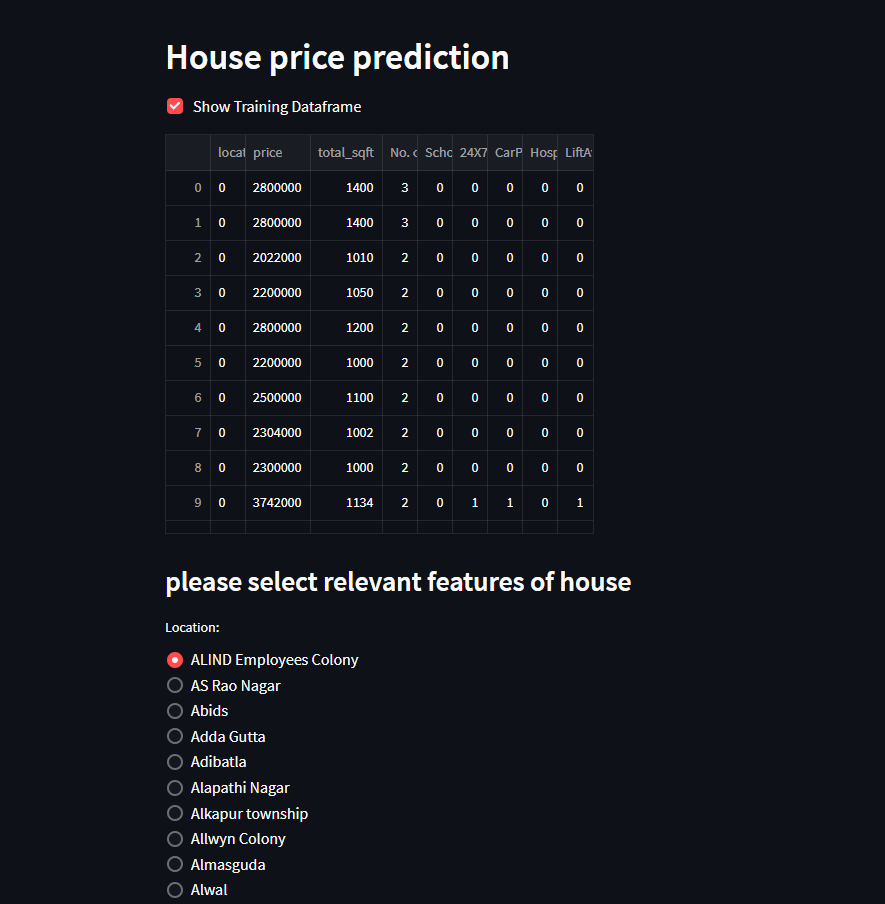


Fig 5.3.1.3: Deployment code for model

### Output Screens



Screen 5.3.2.1: Web application screen

### 5.3.2 Result Analysis



Screen 5.3.2.1: Web application screen2

**Chapter 6**

# TESTING AND VALIDATION

### INTRODUCTION

Software testing serves as the final assessment of specification, design, and code generation and is a crucial component of software quality assurance. Testing is done to look for mistakes. Testing is the process of looking for any flaws or weaknesses in a piece of work. It offers a technique to test the efficiency of parts, subassemblies, and assemblies. It is the act of testing software to make sure it complies with user expectations and meets user needs without failing in an unacceptable way.

Software engineering testing methodologies are testing approaches, strategies, or techniques used to test a particular product to verify its usability. It ensures that the product performs as intended and has no negative impacts when utilised outside of its intended scope. Software testing strategies cover a wide range of testing techniques, including integration, security, and performance testing**.**

* + - Functional testing involves application testing in accordance with business requirements and a variety of test types intended to ensure that every component of the software acts exactly how the users would expect it to.
    - Unit testing is a type of software testing that ensures that each individual piece of software is functioning flawlessly for the intended purpose at the code level.
    - Integration Examination Each unit is rigorously tested before being merged with other units to produce modules or components that are intended to carry out particular functions.
    - System testing is the black box testing method used to evaluate the integrated system as a whole and ensures it meets all specific requirements.
    - Non-functional testing techniques use a variety of test kinds that are concentrated on a piece of software's operational features.
    - The non-functional testing method known as performance testing is used to predict how an application will act in certain scenarios.
    - Security evaluations Finding system flaws and security concerns is the aim of security testing.
    - By evaluating a product with actual consumers, usability testing can determine how user-friendly a product is.

### Design of test cases and scenarios

Using train\_test\_split method we have divide a single data set into training and testing data of desired lengths

Here we have taken 90 percent of data for training the model and 10 percent data for testing the model using test\_size as 0.10 which indicates 10 percent of the data.

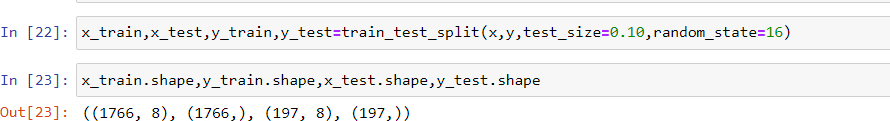


Fig 6.2.1: Code for splitting the training and testing data

### Validation Output:



Screen 6.3.1: Output displaying the predicted house price

When cross verified with the values in the available dataset, we came to know that the developed model predicted the outcome correctly.

#### Validation report

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **locatio n** | **sqft** | **bedroo ms** | **schoo l** | **24x7**  **securi ty** | **car** | **ho spi**  **tal** | **Expected Output** | **Result** |
| Adibatl  a | 1400 | 3 | 0 | 0 | 0 | 0 | 2800000 | True |
| BHEL | 1010 | 3 | 0 | 0 | 0 | 0 | 2022000 | True |
| Alwal | 1134 | 2 | 0 | 1 | 1 | 0 | 3724200 | True |
| JUNTU | 1350 | 3 | 0 | 1 | 0 | 1 | 5202000 | False |
| BHEL | 1100 | 2 | 2 | 0 | 0 | 0 | 2500000 | True |

Table 6.3.1: Expected output vs resultant output

## Chapter 7 CONCLUSION AND FUTURE WORK

**Conclusion:**

* Machine learning serves to be one of the important components of the growing field of data science.
* By using statistical methods, algorithms are trained to make classifications predictions.
* Machine learning and data mining techniques are valuable in disease diagnosis. The ability to predict diabetes early serves a vital role in the patient's appropriate treatment procedure.
* We have subjected our dataset to various algorithms in this project and compared their respective accuracies.
* The study shows a comparison between the regression algorithms predicting house prices in Hyderabad. The results were promising for the public data due to it being rich with features and having strong correlation.
* Four machine learning algorithms were applied to the public Hyderabad house price dataset and trained and validated against a test dataset.
* Finally, we have realized that Lasso regression tends to have the maximum accuracy of 88 percent when compared with accuracies of linear regression (79.2%), ridge (75.3%).

## Future Enhancement:

Future work on this model could be divided into two main areas to improve the result even further. Which can be done by:

* The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.
* Make use of the available features and if they could be combined as binning features has shown that the data got improved.
* The model will be deployed in the app so that could be available to every one to estimate the price of an house.

# REFERENCES

#### Form Journal of Sweden University:

[1]. Ahmad Abdulal & Nawar aghi: “House Price Prediction”, Kristianstand university Sweden, pp. 3 – 28,, Issued on spring semester 2020 .

#### Form Book the Hundred-page Machine Learning Book:

[2]. Andriy Burkov: “The Hundred-page Machine Learning”,

pp. 105, Quality South Asia Edition, 2006.

#### Form Research Papers:

[3]. I.J. Modern Education and Co mputer Science, 2020, 6, 46-54 Published Online December 2020 in MECS [(ht](http://www.mecs-press.org/))t[p://www.mecs-press.org/)](http://www.mecs-press.org/)) DOI: 10.5815/ijmecs.2020.06.04

#### Form Publication of IJRASET:

[4]. Authors: Mr. Piyush Chordia, Mr. Pratik Konde, Ms. Supriya Jadhav, Hrutik Pandhare, Prof. Shikha Pachouly DOI Link: <https://doi.org/10.22214/ijraset.2022.40466>

**Git resource used:** [https://github.com/Rohit7594/Banglore-House-Price-](https://github.com/Rohit7594/Banglore-House-Price-Predictor) [Predictor](https://github.com/Rohit7594/Banglore-House-Price-Predictor)