|  |  |
| --- | --- |
| Internship Project Title | Intelligent Property Analyser |
| Name of the Company | TSC iON |
| Name of the Industry Mentor | Harish Kumar |
| Name of the Institute | IIT, Madras |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 11/02/2023 | 18/02/2023 | 45 | Windows 10, Chrome, VS Code | Python, Jupyter Notebook, SQL, Django, HTML, CSS |

**TABLE OF CONTENTS**

[**1. Acknowledgements** 3](#_Toc127822630)

[**2. Objectives** 3](#_Toc127822631)

[**3. Introduction** 3](#_Toc127822632)

[**4. Internship Activities** 4](#_Toc127822633)

[**5. Methods and Algorithms** 5](#_Toc127822634)

[**5.1 Importing Necessary Python Packages:** 5](#_Toc127822635)

[**5.2 Importing Dataset** 6](#_Toc127822636)

[**5.3 EDA** 6](#_Toc127822637)

[**5.4 Data Splitting and Feature Selection** 6](#_Toc127822638)

[**5.5 Machine Learning Models** 7](#_Toc127822639)

[**5.6 Model Evaluation** 9](#_Toc127822640)

[**5.7 Price Prediction** 9](#_Toc127822641)

[**5.8 Developing Web App using Django** 9](#_Toc127822642)

[**6. Assumptions** 10](#_Toc127822643)

[**7. Project Analysis** 10](#_Toc127822644)

[**7.1 EDA** 10](#_Toc127822645)

[**7.2 Model Evaluation** 14](#_Toc127822646)

[**7.3 Model Prediction** 15](#_Toc127822647)

[**7.4 Web App** 16](#_Toc127822648)

[**8. Conclusion** 16](#_Toc127822650)

[**9. Enhancement Scope** 17](#_Toc127822651)

[**10. Link to Code** 17](#_Toc127822652)

[**11. References** 17](#_Toc127822653)

**1. Acknowledgements**

I’m thankful to TCS iON for providing this internship. I'd want to offer my heartfelt gratitude and appreciation to my industry mentor Mr. Harish Kumar for his time & efforts and for helping me to complete my internship project successfully.

**2. Objectives**

In this project, we create an end-to-end web user interface-driven system powered by machine learning algorithms and built with Python-centric modules that can estimate the price of housing properties depending on a variety of conditions and parameters.

1. Predict the approximate price of a residential property based on several property-related criteria.
2. Create a web user interface enabling users to access the system and feed property data for analysis.
3. Design end to end ML-based system that has been trained with enough data to complete the task with acceptable accuracy.

**3. Introduction**

People and real estate agencies both buy and sell houses; people buy to live in or invest in, while agencies buy to run a business. In any case, we believe that everyone should receive exactly what they pay for. Over- and under-valuation in housing markets has always been a problem, and there are no proper detection measures in place.

A Buyer's main goal is to find their dream home with all of the features they require. Additionally, they search for houses/real estate with a price in mind, and there is no assurance that they will obtain the goods at a fair and not inflated price. Similarly, a seller seeks for a specific figure that they can put on the estate as a price tag, and this cannot be just a random estimate; much study is required to produce a property value. A system capable of precisely predicting pricing and catering to everyone's demands may be constructed by training an ML model with hundreds of thousands of data points.

In this project, we will create a web app for house price prediction. First, we'll define our app's data model. We'll also design a view for the web app that displays the data model and allows users to interact with it. Finally, we will build a prediction model using a variety of machine learning algorithms such as linear regression, decision trees, etc. The web app will then be created and deployed to a server using Django's web framework.

**4. Internship Activities**

**Stage 1: Develop ML model using python**

1. Data Collection
2. Data Preprocessing
3. Model Selection
4. Training the model
5. Evaluating model
6. Parameter Tuning
7. Making Predictions

**Stage 2: Setting up MySQL database**

1. Setup MySQL database
2. Database would be used by users/admins to populate records and predicted prices of properties

**Stage 3: Develop Django based web project**

1. Develop Web-UI
2. Users can view properties and get predicted prices
3. Designing report that displays predicted property prices

**5. Methods and Algorithms**

## **5.1 Importing Necessary Python Packages:**

1. **NumPy:** NumPy is a library for the Python programming language that provides support for large, multi-dimensional arrays and matrices, as well as a large collection of high-level mathematical functions. It allows for the efficient manipulation of data, including the creation of arrays, reshaping and slicing of arrays, element-wise manipulation of arrays, and the application of mathematical functions to arrays.
2. **Pandas**: Pandas is a Python library for data analysis and manipulation. It provides tools for working with tabular, multidimensional data, such as data frames and series. It provides a wide range of features for data manipulation, including merging, sorting, indexing, and data alignment. It also provides tools for statistical analysis, such as aggregation and groupby operations.
3. **Matplotlib:** Matplotlib is a Python library for plotting data. It provides a range of functions for plotting 2D and 3D data, including line, bar, histogram, scatter, and contour plots. It also includes a variety of features for manipulating and customizing plots, such as color palettes, legends, annotations, and interactive plots.
4. **Seaborn**: Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. It features several plot types for statistical data exploration, including scatter plots, box plots, violin plots, and regression plots. It also includes tools for visualizing univariate and bivariate distributions, and for visualizing linear relationships between variables.
5. **Scikit-learn**: Scikit-learn (sklearn) is a free software machine learning library for the Python programming language. It provides various tools for data analysis and machine learning, such as classification, regression, clustering, dimensionality reduction, and model selection. It also includes tools for text processing, feature extraction, and feature selection.
6. **Warnings**: The warnings module allows developers to filter out certain warnings or customize the formatting of warnings.

## **5.2 Importing Dataset**

Using the pandas.read\_csv() function, we can import a dataset in Python. This function returns a pandas DataFrame containing the data from the file and accepts a file path or URL as an argument. The data can then be filtered, sorted, and summarised using this DataFrame for a variety of data manipulation tasks.

## **5.3 EDA**

Exploratory data analysis (EDA) is the systematic analysis of data to identify patterns and relationships between various variables as well as to summarise the data's main characteristics. It is a crucial stage in the analysis of data and can give you access to data insights that more formal statistical techniques cannot provide. Python's pandas DataFrame object and associated functions and methods are frequently used for EDA. These enable quick and simple data manipulation, visualisation, and summarization. Plotting distributions, figuring out correlations, figuring out outliers, and fitting models to the data are a few examples of EDA tasks.

## **5.4 Data Splitting and Feature Selection**

Data splitting and feature selection are two important machine learning techniques. The act of splitting available data into two portions, usually for cross-validation purposes, is known as data splitting. One part of the data is used to train a model, and the other part is used to assess the model's performance. The process of selecting features from a larger set of features that are most relevant to a given response variable is known as feature selection. This can be accomplished through the use of metrics such as mutual information, variance inflation factor, or a variety of other methods. Feature selection is critical for reducing overfitting and improving model performance.

## **5.5 Machine Learning Models**

Machine learning models are algorithms that predict or make decisions based on data. These models are trained using data, which is then used to create a model capable of making accurate predictions or decisions based on new data. Machine learning models are frequently used to identify patterns in data and predict future data, as well as to make decisions about how to respond to new data.

**Linear Regression:**

Linear regression is a statistical technique for examining the relationship between two or more variables. It can be used to predict values based on past data and to explain variation in the response variable caused by variation in the explanatory variables. Linear regression models can be simple, with only one independent variable (simple linear regression), or complex, with multiple independent variables (complex linear regression) (multiple linear regression). In both cases, the model is a linear function of the independent variables, which means that for each unit of change in the independent variable, a fixed amount is added to the final result. When using multiple linear regression, the model is expressed as an equation with the independent variables on the right and the response variable on the left. The coefficients of the equation show how much a change in the independent variables is predicted to affect the response variable.

**Lasso Regression:**

Lasso regression is a type of regularized linear regression model that employs a lasso or L1 penalty term to reduce the size of parameter estimates. The lasso penalty is the absolute value of the sum of the estimated coefficients, and the goal is to minimize the sum of the squared residuals while keeping this constraint in mind. This means that the lasso model shrinks the coefficient estimates towards zero, which can aid in reducing model overfitting. Furthermore, because some of the coefficients may be set to zero during the optimization process, lasso regression can be used to perform feature selection.

**Ridge Regression:**

Ridge regression is a regularized linear regression model that employs an additional penalty term, known as the ridge or L2 penalty, to reduce the size of the coefficient estimates. The ridge penalty is equal to the sum of the squared coefficient estimates, and the goal is to minimize the sum of the squared residuals while keeping this constraint in mind. This means that the ridge model shrinks the coefficient estimates towards zero more slowly than the lasso model. This can help the model generalize to new data more effectively. Furthermore, because some of the coefficients may be set to zero during the optimization process, ridge regression can be used to perform feature selection.

**ElasticNet Regression:**

ElasticNet regression is a type of regularised linear regression model that combines the lasso and ridge regression L1 and L2 penalties. It is a hybrid of lasso and ridge regression, which means it employs both L1 and L2 regularisation. This combination of penalties reduces overfitting by shrinking the coefficient estimates towards zero while leaving some coefficients non-zero. Furthermore, because some of the coefficients may be set to zero during the optimization process, ElasticNet regression can be used to perform feature selection.

**Polynomial Regression:**

Polynomial regression is a type of regression analysis that models the relationship between the independent variable x and the dependent variable y as an nth-degree polynomial. This type of regression is useful for fitting nonlinear relationships and uncovering hidden patterns in data. Polynomial regression can also be used to forecast the dependent variable's future values.

A polynomial function is created in polynomial regression by adding together multiple polynomials of varying degrees, each of which is multiplied by a different coefficient. The coefficients are calculated by fitting the polynomial to the data points and minimising the difference between predicted and actual values. The more complex the relationship between the independent and dependent variables, the higher the degree of the polynomial.

**Decision Trees:**

Decision Tree Regression is a type of supervised learning technique used to predict continuous values. It is a nonparametric model that uses a decision tree structure to make predictions about the target variable. The decision tree is created by splitting the data based on the values of the features, and then making predictions based on the average of the values of the target variable in each group. This approach allows the model to capture non-linear relationships between the features and the target variable. It is also able to handle missing data and outliers, and can provide more accurate predictions than linear regression models.

**Random Forest:**

Random forest regression is a supervised learning algorithm that uses an ensemble of decision trees to predict a continuous target variable. This method is a powerful tool for machine learning, as it combines the power of multiple decision trees to produce a more accurate and reliable prediction than a single decision tree. The algorithm works by randomly selecting a subset of features from the dataset, which are then used as candidates to split the data into different branches. Each branch is then used to build a decision tree, which is then used to make a prediction. The predictions from all the trees are then combined to produce a single prediction. The random forest regression model has been found to be more accurate and robust than other machine learning algorithms.

**K-Nearest Neighbours:**

K-Nearest Neighbours (KNN) Regression is an instance-based algorithm that can be used for both classification and regression. In KNN regression, the output is a real value. The input consists of the k closest training examples in the feature space. The prediction is calculated by averaging the output variable for the k nearest neighbours. KNN regression is a simple yet powerful approach, as it is non-parametric and can capture non-linear relationships. It is also computationally efficient, as no training is required and the only thing that needs to be done is to store the training examples. KNN regression is commonly used for predicting continuous variables such as stock prices, sales, or house prices, and can also be used to predict categorical variables such as customer churn or customer segmentation.

## **5.6 Model Evaluation**

Model evaluation is the process of determining how well a model performs on a given dataset. This is typically accomplished by dividing the data into train and test sets, with the training set used to train the model and the test set used to evaluate the model's performance. Models are commonly evaluated using metrics such as accuracy, precision, recall, and F1 score.

## **5.7 Price Prediction**

Price prediction is the process of predicting the price of a product. This can be accomplished through the use of machine learning models such as linear regression, decision trees, etc. The model is trained using past data and then is used to predict prices.

## **5.8 Developing Web App using Django**

Django is a Python-based web development framework. It is intended to make web application development easier and more efficient. Django provides a set of modules, libraries, and tools for quickly and easily creating and deploying web applications. Django also includes built-in support for database access, user authentication, and other common web application tasks, making it a popular web development choice.

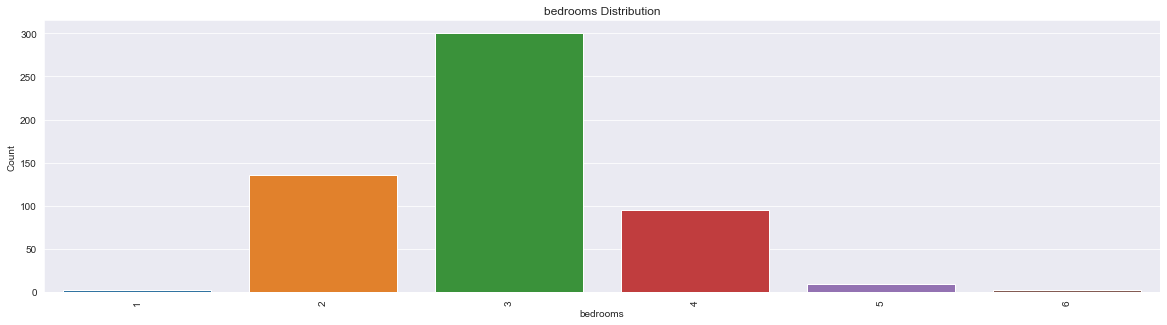
Here, we will specify the features that will be used in our application as well as the data model for our web application. The views will be written to display the data model and enable user interaction. The logic for our web app, which consists of the code that will handle user requests and produce the necessary responses, will then be created. The web app will be built using Django's web framework and then deployed to a server once all of these components are in place.

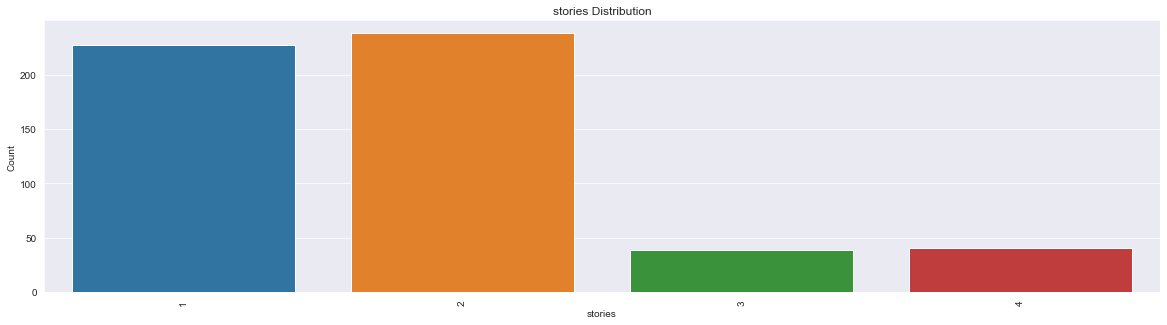
**6. Assumptions**

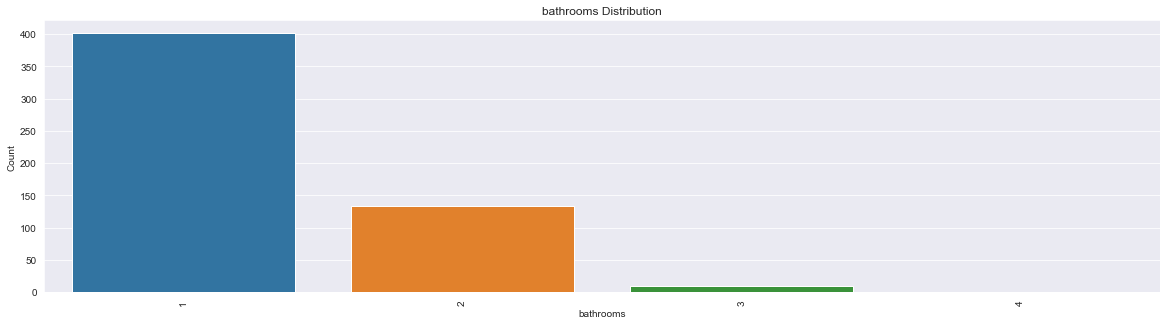
We have to predict housing prices using the given dataset. We assume that the data provided is free from errors and its features are related to target variable i.e., price.

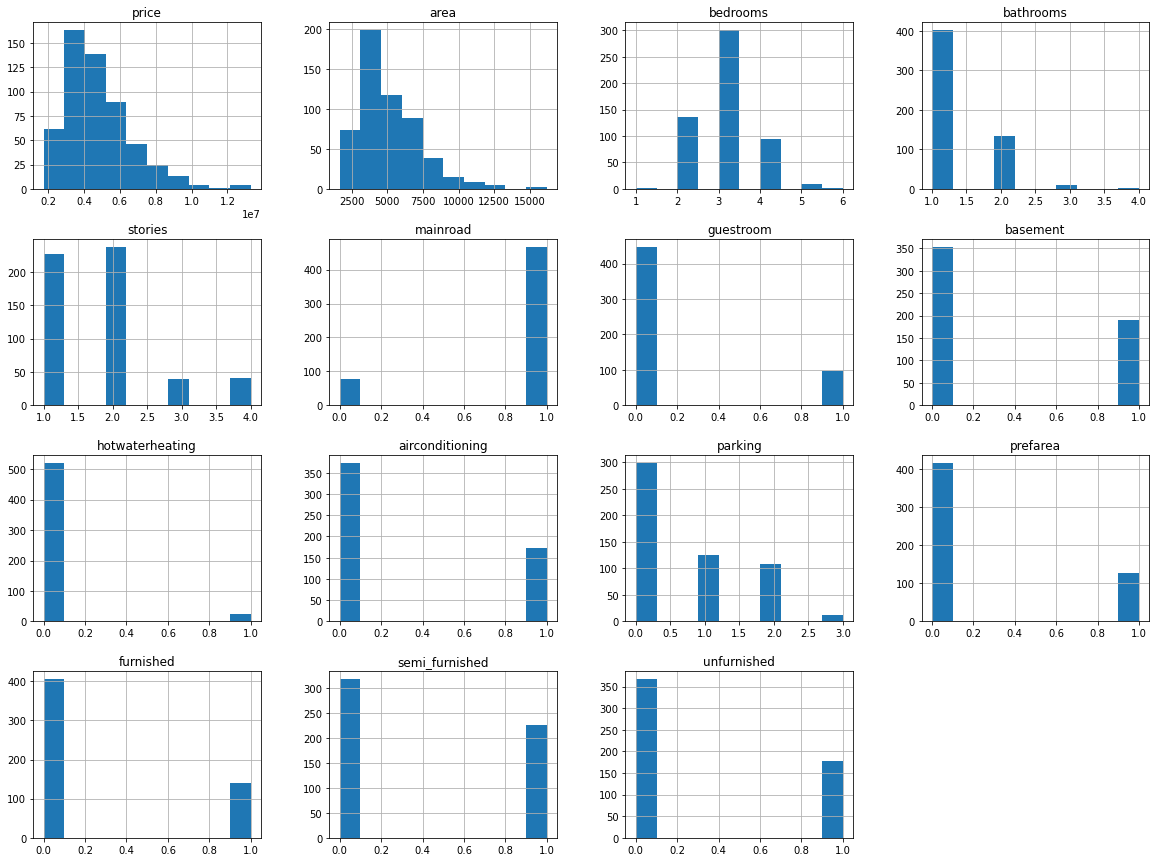
**7. Project Analysis**

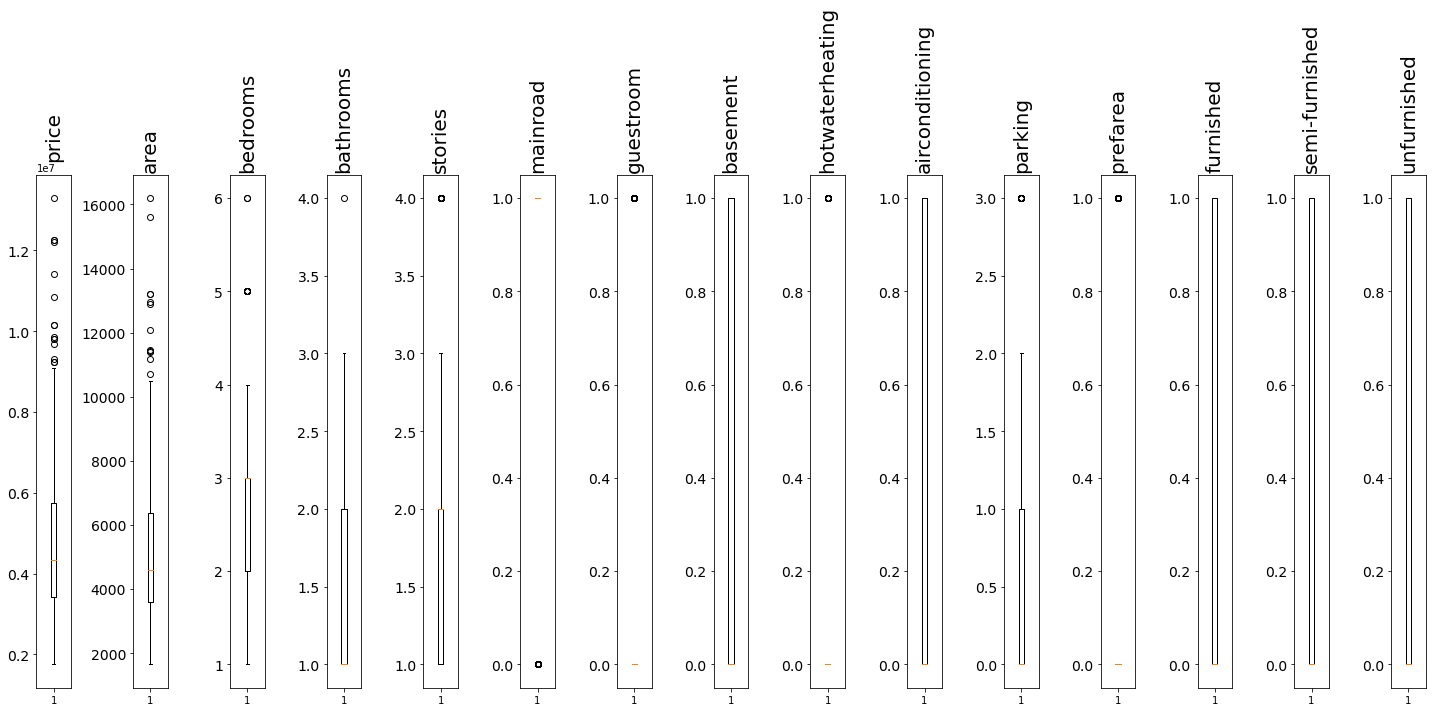
## **7.1 EDA**



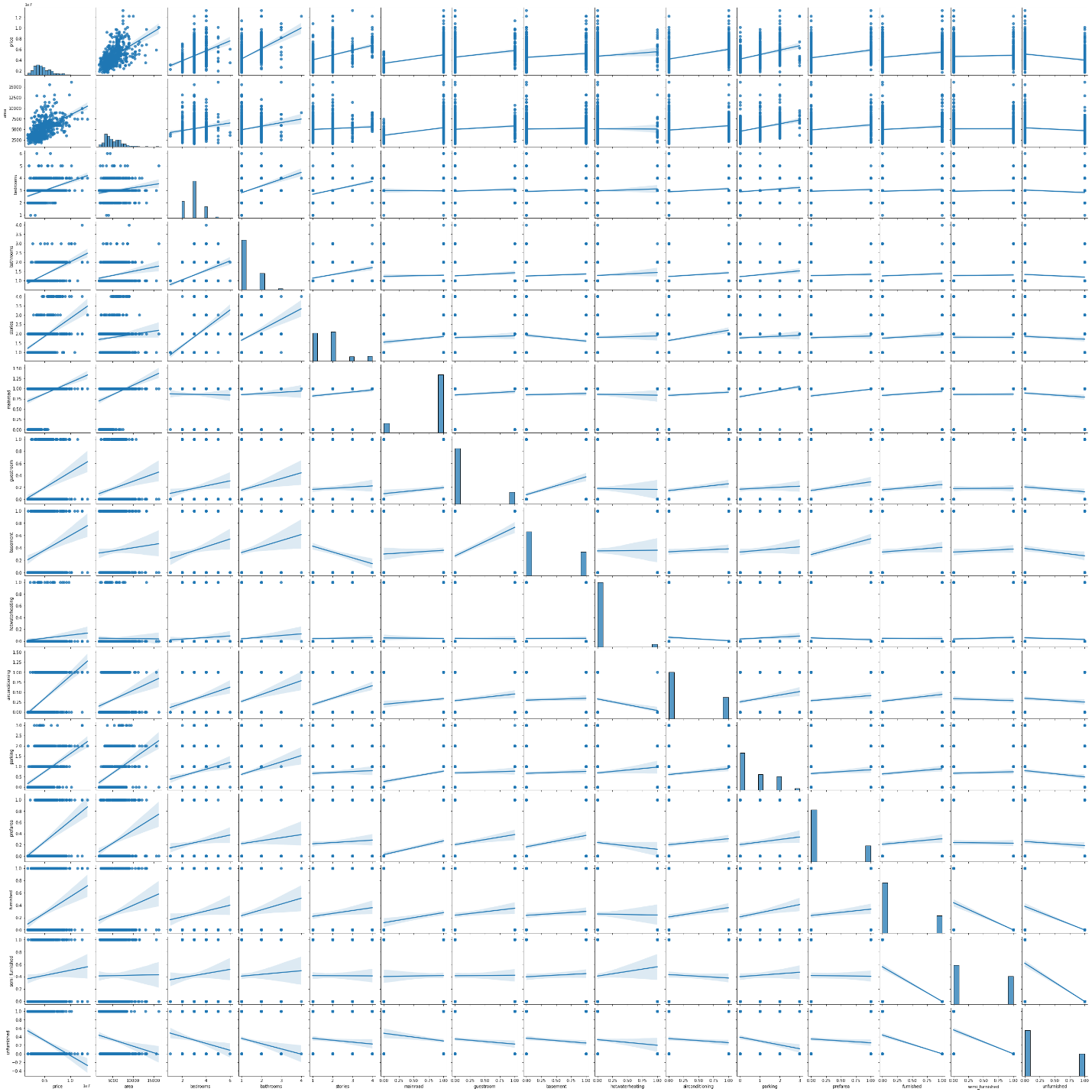


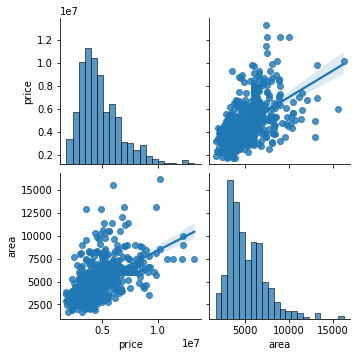
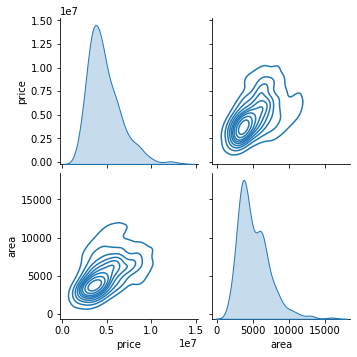






1. Most houses have 3 bedrooms.
2. Almost all houses have 1 & 2 stories
3. Most houses have single bathroom.
4. Housing price distribution appears to increase as the area code increases. Regression analysis might be used in further research on this.
5. The majority of the homes appear to be priced similarly across different area zones.
6. In this analysis, not many outliers are present.





|  |  |
| --- | --- |
|  | price |
| price | 1.000000 |
| area | 0.535997 |
| bathrooms | 0.517545 |
| airconditioning | 0.452954 |
| stories | 0.420712 |
| parking | 0.384394 |
| bedrooms | 0.366494 |
| prefarea | 0.329777 |
| mainroad | 0.296898 |
| guestroom | 0.255517 |
| furnished | 0.229350 |
| basement | 0.187057 |
| hotwaterheating | 0.093073 |
| semi\_furnished | 0.063656 |
| unfurnished | -0.280587 |

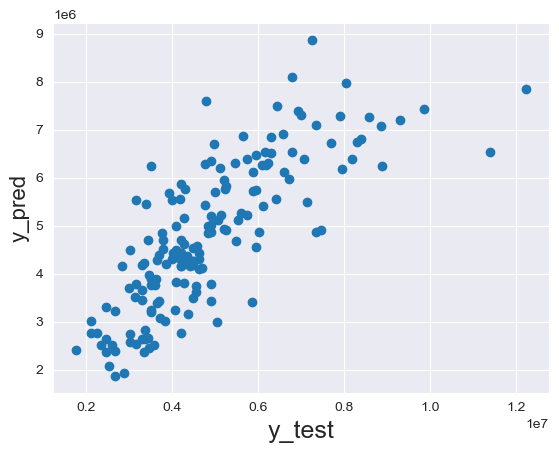
1. According to the regression analysis, area code and home price are positively correlated. It may be necessary to use residual plots to verify that this conclusion is accurate.
2. There is no direct correlation between price and other features.

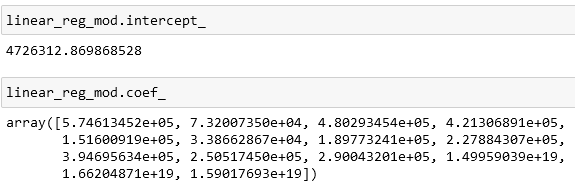
**7.2 Model Evaluation**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Models** | **Training Scores** | | | | **Testing Scores** | | | |
| **R2** | **MAE** | **MSE** | **RMSE** | **R2** | **MAE** | **MSE** | **RMSE** |
| Linear  Regression | 0.703 | 7.67E+05 | 1.06E+12 | 1.03E+06 | 0.613 | 7.87E+05 | 1.26E+12 | 1.12E+06 |
| Lasso  Regression | 0.703 | 7.67E+05 | 1.06E+12 | 1.03E+06 | 0.612 | 8.18E+05 | 1.26E+12 | 1.12E+06 |
| Ridge  Regression | 0.703 | 7.67E+05 | 1.06E+12 | 1.03E+06 | 0.612 | 8.17E+05 | 1.26E+12 | 1.12E+06 |
| ElasticNet  Regression | 0.677 | 7.78E+05 | 1.16E+12 | 1.08E+06 | 0.613 | 7.87E+05 | 1.26E+12 | 1.12E+06 |
| Polynomial  Regression | 0.787 | 6.47E+05 | 7.76E+11 | 8.81E+05 | 0.459 | 8.44E+05 | 1.56E+12 | 1.25E+06 |
| Decision Tree | 0.998 | 7.70e+03 | 4.47e+09 | 6.69e+04 | 0.177 | 1.09e+06 | 2.36e+12 | 1.53e+06 |
| Random Forrest | 0.833 | 5.52e+05 | 6.06e+11 | 7.78e+05 | 0.569 | 8.06e+05 | 1.24e+12 | 1.11e+06 |
| kNN | 0.559 | 9.15e+05 | 1.60e+12 | 1.26e+06 | 0.346 | 9.68e+05 | 1.88e+12 | 1.37e+06 |

From the above scores it is evident that linear regression is the best model.

**7.3 Model Prediction**





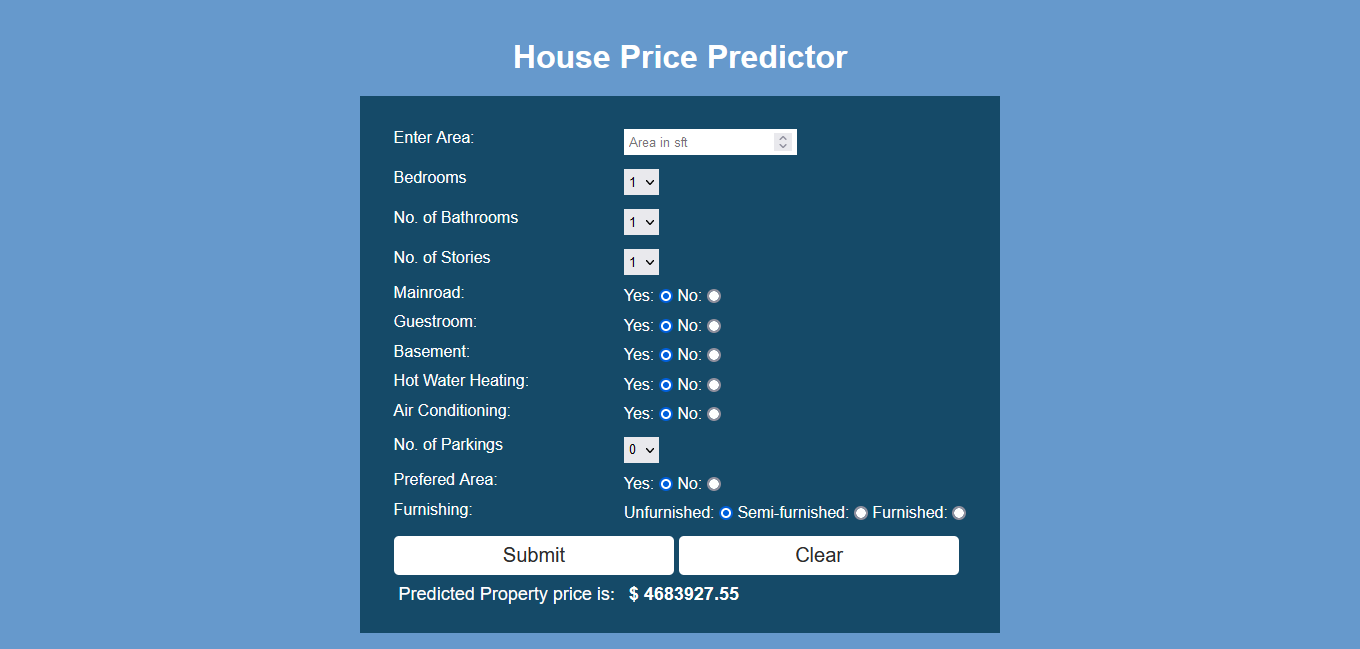
Sample prediction of test dataset:

|  |  |  |
| --- | --- | --- |
| **Linear Regression Model** | | |
| **SR** | **Actual Price** | **Predicted Price** |
| 0 | 4753000 | 5441065 |
| 1 | 8890000 | 6237737 |
| 2 | 7455000 | 4902441 |
| 3 | 3773000 | 4853289 |
| 4 | 3780000 | 4695593 |
| 5 | 3703000 | 3431977 |
| 6 | 7910000 | 7288361 |
| 7 | 3500000 | 3243561 |
| 8 | 4098500 | 4992553 |
| 9 | 5250000 | 5830185 |

|  |  |  |
| --- | --- | --- |
| **ElasticNet Model** | | |
| **SR** | **Actual Price** | **Predicted Price** |
| **0** | 4753000 | 5293042.31 |
| **1** | 8890000 | 6208419.19 |
| **2** | 7455000 | 4784511.42 |
| **3** | 3773000 | 4956305.38 |
| **4** | 3780000 | 4503912.47 |
| **5** | 3703000 | 3681162.87 |
| **6** | 7910000 | 6783972.12 |
| **7** | 3500000 | 3561016.23 |
| **8** | 4098500 | 5004110.66 |
| **9** | 5250000 | 5584245.96 |

1. We can see from the above table that error in prediction is up to 3 million.
2. Although some of the predicted prices are very close to actual price.
3. This is because our dataset is small and other price affecting features like material cost etc are not included in this dataset.

## **7.4 Web App**

****

1. We have created a web app using Django framework.
2. Its code is on github. The link to the code is provided below.

**8. Conclusion**

1. Area is the most important feature affecting price.
2. There is no direct correlation between price and other features.
3. Not many outliers are present.
4. Linear regression is the best fit model.

**9. Enhancement Scope**

1. Dataset is very small; hence more data needs to be acquired.
2. Lots of price dependent features are absent. Features such as structure\_type, flooring\_type, cladding\_type, ceiling\_type etc should be included. These features are very much price dependent and are in fact the main features affecting price.

**10. Link to Code**

<https://github.com/m-parth/Intelligent-Poperty-Analyser>

**11. References**

1. <https://www.djangoproject.com/>
2. <https://www.kaggle.com/datasets/yasserh/housing-prices-dataset>
3. <https://www.kaggle.com/code/ashydv/housing-price-prediction-linear-regression>
4. <https://dev.mysql.com/doc/>
5. <https://scikit-learn.org/stable/tutorial/index.html>
6. <https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html>
7. <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data>