|  |  |
| --- | --- |
| Internship Project Title | Intelligent Property Analyser |
| Name of the Company | TCS 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 |
| 09/02/2023 | 13/02/2023 | | 23.5 | | Windows 10 | Python, Jupyter notebook |
| Milestone # | 1 | Milestone: | | Predict price of housing property based on multiple factors | | | |

**TABLE OF CONTENTS**

[**1. Acknowledgements** 2](#_Toc127215298)

[**2. Objectives** 2](#_Toc127215299)

[**3. Introduction** 2](#_Toc127215300)

[**4. Internship Activities** 2](#_Toc127215301)

[**5. Methods and Algorithms** 3](#_Toc127215302)

[**6. Assumptions** 3](#_Toc127215303)

[**7. Project Analysis** 3](#_Toc127215304)

[**Model Evaluation** 6](#_Toc127215305)

[**Model Prediction** 6](#_Toc127215306)

[**10. Conclusion** 7](#_Toc127215307)

[**11. Enhancement Scope** 7](#_Toc127215308)

[**12. Link to Code** 7](#_Toc127215309)

[**13. References** 7](#_Toc127215310)

# **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**

# **4. Internship Activities**

Internship activities are divided in three stages which are as follows:

**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**

Steps Involved:

1. Importing the necessary Python packages into our environment
2. Importing house price data and performing EDA on it
3. Visualization of statistics on house prices
4. Data Splitting and Feature Selection
5. Using ML algorithms to model the data
6. Using the assessment metrics to evaluate model
7. Price prediction
8. Developing web app for price prediction

We have used these models for regression: Linear Regression, Lasso Regression, Ridge Regression, ElasticNet Regression, Polynomial Regression, Decision Tree, Random Forrest and kNN.

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 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 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 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 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 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.

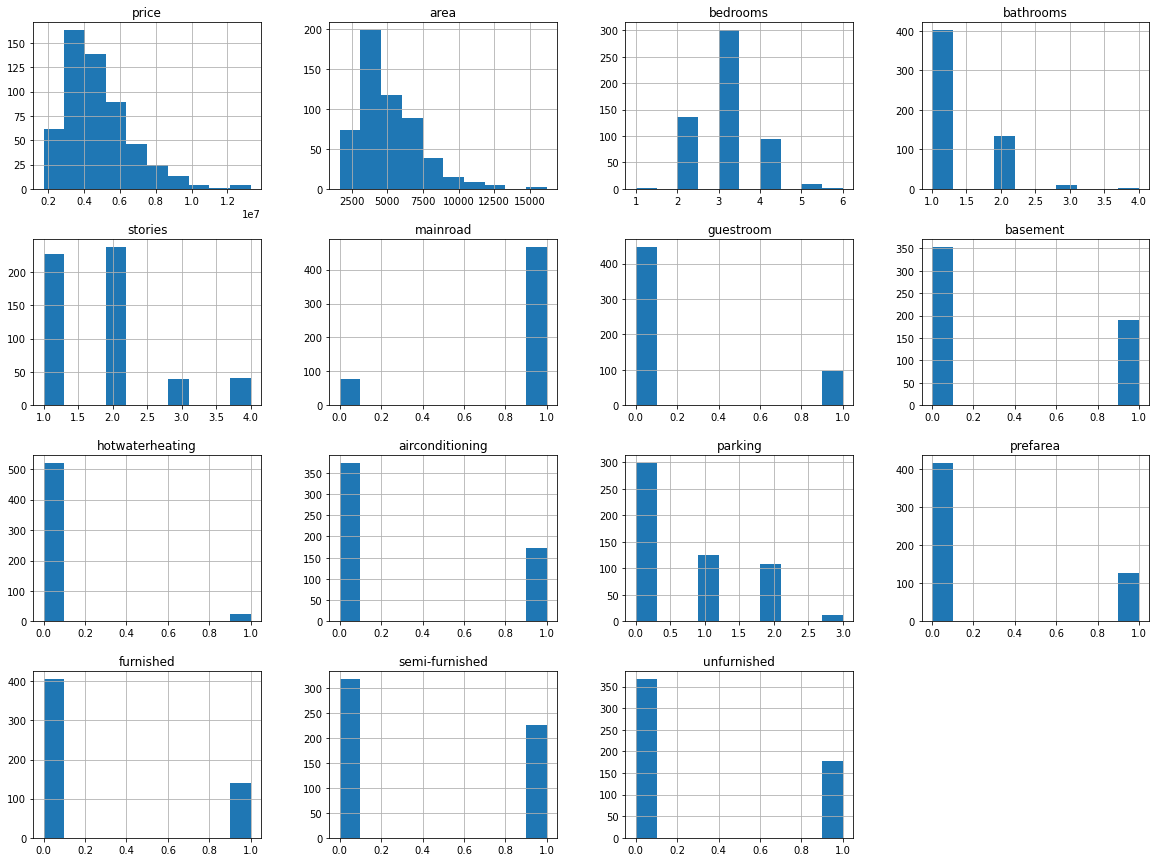
Random forest regression is a supervised learning algorithm that uses an ensemble of decision trees to predict a continuous target variable. This method is a powerful tool for machine learning, as it combines the power of multiple decision trees to produce a more accurate and reliable prediction than 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.

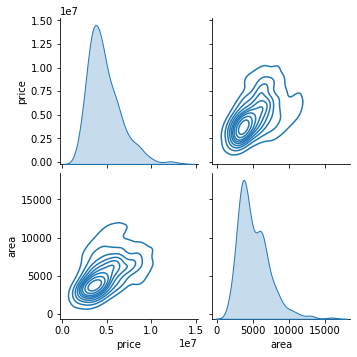
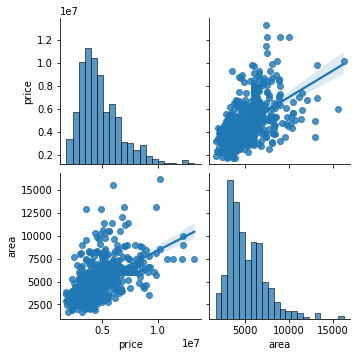
K-Nearest Neighbours (KNN) Regression is an instance-based algorithm that can be used for both classification and regression. In KNN regression, the output is a real value. The input consists of the k closest training examples in the feature space. The prediction is calculated by averaging the output 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.

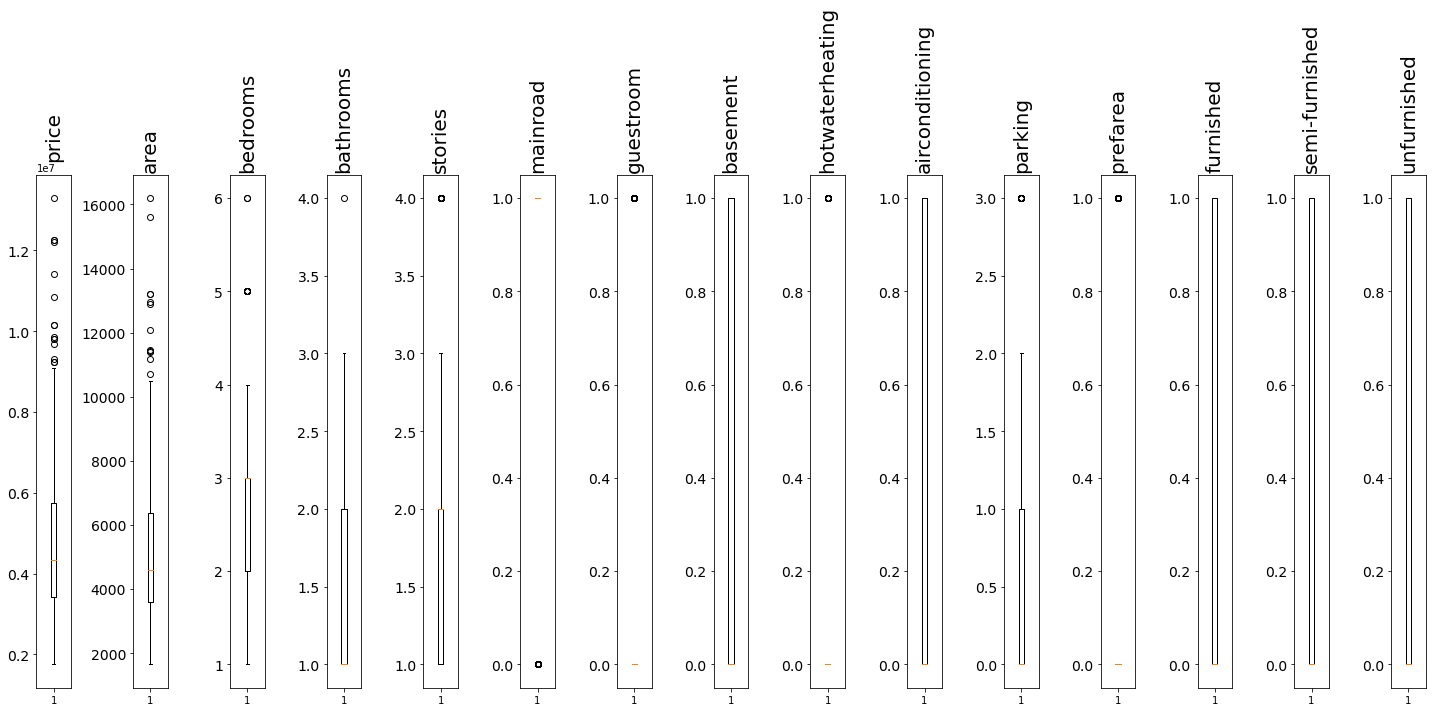
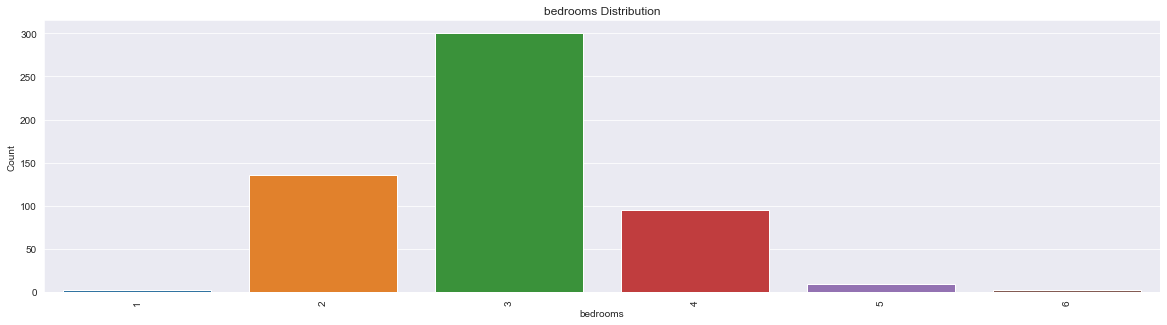
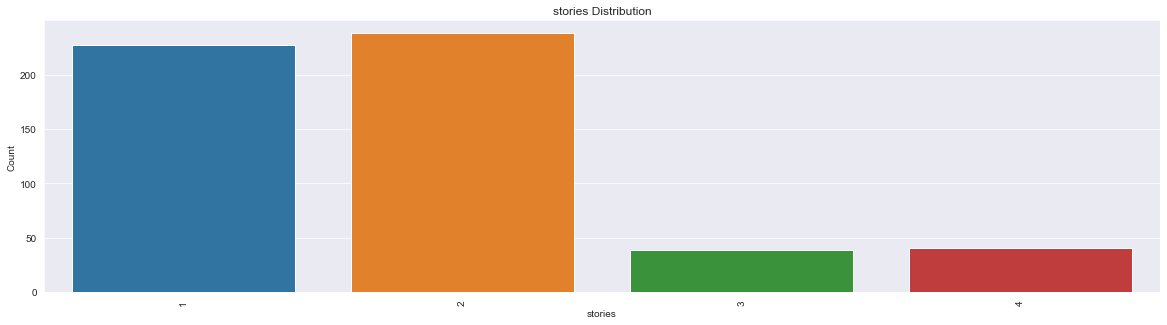
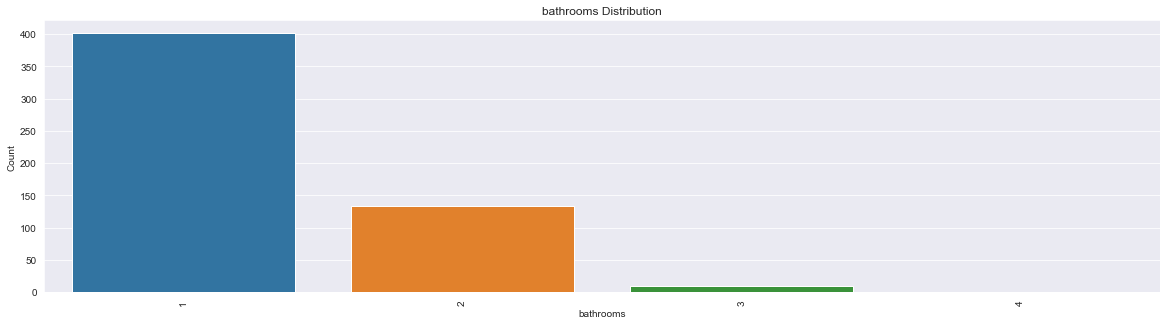
# **6. Assumptions**

We have to predict housing prices using the given dataset. We assume that the data provided is free from errors and its features are inter-related.

# **7. Project Analysis**







## **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 | 8.17E+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 |

## **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 |

# **10. Conclusion**

1. Area is the most important feature affecting price.
2. Linear regression is the best fit model.

# **11. 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.

# **12. Link to Code**

# **13. 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>