### An Efficient Predictive Analysis Model for House Rent using Random Forest and Gradient Boosting Algorithm

**Project** Report submitted in partial fulfillment of the **requirements**

for the **degree** of

**Bachelor of Technology**

in

**Computer Science and Engineering**

of

**Maulana Abul Kalam Azad University of Technology**

By :

**Ayan Dhua**, Roll No. - **10900120100 Biltu Pal**, Roll No. - **10900120102**

Under the **guidance** of :

Mrs. **Soumi Ghosh**

**Assistant Professor**, **C.S.E. Department Department** of **Computer Science and Engineering**



**Netaji Subhash Engineering College**

Techno City, Garia, Kolkata – 700152

**Academic Year of 2020 – 2024**

# Certificate

This is to certify that this project report titled – “**An Efficient Predictive Analysis Model for House Rent using Random Forest and Gradient Boosting Algorithm**”, submitted in partial fulfillment of requirements for the award of the degree Bachelor of Technology (B.Tech.) in the program Computer Science and Engineering of Maulana Abul Kalam Azad Universityof Technology is a faithful record of the original work carried out by:

**Ayan Dhua** – Roll No. **10900120100** – Regd. No. 201090100110073 of 2020-21

**Biltu Pal** – Roll No. **10900120102** – Regd. No. 201090100110071 of 2020-21

under my guidance and supervision.

It is further certified that it contains no material, which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except the assistances drawn from other sources, for which due acknowledgement has been made.

Date:

Dr. **Piyali Chatterjee**, Mrs. **Soumi Ghosh**,

Head of the Department, Assistant Professor,

**Dept.** of **Computer Science & Engg.**, **Dept.** of **Computer Science & Engg.**, **Netaji Subhash Engineering College**, **Netaji Subhash Engineering College**, Techno City, Garia, Techno City, Garia,

Kolkata – 700152 Kolkata - 700152

# Declaration

We, hereby, declare that this project report titled - “**An Efficient Predictive Analysis Model for House Rent using Random Forest and Gradient Boosting Algorithm**”, is our own original work carried out as an under graduate student in Netaji Subhash Engineering College except to the extent that assistances from other sources are duly acknowledged.

All **sources** used for this project report have been fully and properly **cited**. It contains no material which to a substantial extent has been submitted for the award of any degree/diploma in any institute or has been published in any form, except where due acknowledgement is made.

|  |  |  |
| --- | --- | --- |
| Students’ Names: | Signatures: | Dates: |
| **Ayan Dhua** |  |  |
| **Biltu Pal** |  |  |

# Certificate of Approval

We, hereby, approve this dissertation titled as – “**An Efficient Predictive Analysis Model for House Rent using Random Forest and Gradient Boosting Algorithm**”, carried out by:

**Ayan Dhua** – Roll No. **10900120100** – Regd. No. 201090100110073 of 2020-21

**Biltu Pal** – Roll No. **10900120102** – Regd. No. 201090100110071 of 2020-21

under the guidance of Mrs. **Soumi Ghosh** of **Netaji Subhash Engineering College**, Kolkata in partial fulfillment of requirements for award of the degree **Bachelor of Technology** (B.Tech.) in the program **Computer Science and Engineering** of **Maulana Abul Kalam Azad University of Technology**.

Date:

Examiners’ Signatures:

# Acknowledgement

We would like to express our special words of gratitude to our **project mentor** and **supervisor** Mrs. **Soumi Ghosh**, who gave us the golden opportunity to do this wonderful project. She also guided us at every stage in the way of completing our target.

In addition, we also thank our college – **Netaji Subhash Engineering College**, for providing us with proper infrastructure and facilities required toexecute this project.

Secondly, we would not forget to remember our **parents** for their motivation, encouragement, and moreover, for their timely support and guidance till thecompletion of our project work.

Last but not least, we are also thankful to our **friends** for providing us with some of the research articles we used, which helped us a lot in finalizing thisproject within the limited time frame.

**Ayan Dhua Biltu Pal**

Dated:

# Abstract

Our proposed model efficiently predicts house rents by leveraging key characteristics identified through feature selection using random forest regression. The rent prediction is then performed using gradient boosting, ensuring high accuracy and reliability.

Regression analysis is widely used in machine learning for statistical modeling. Random Forest, a supervised algorithm, employs Ensemble Learning and Bootstrap Aggregating for accuracy. Gradient Descent optimizes parameters to minimize error, boosting performance.

Accurate house rent prediction aids tenants, landlords, and real estate businesses in decision-making. Our model uses a house rent dataset, applying feature scaling and pre-processing for standardization. Random Forest Regression identifies key features, which are then used in Gradient Descent Regression to forecast rents. This method achieves low Absolute Mean Deviation and high Mean Accuracy, enhancing profitability and adaptability to various real estate markets.

# Contents

|  |  |  |
| --- | --- | --- |
| **#** | **Chapters** | **Pages** |
| 1 | **Introduction** | 01 |
| 2 | **Literature Review** | 02 – 21 |
| 3 | **Our Proposed Work** | 22 – 25 |
| 4 | **Result Analysis** | 26 – 44 |
| 5 | **Conclusion** | 45 |

**Chapter 1**

# Introduction

Predicting house rent prices using machine learning can help landlords, tenants, and real estate professionals. The rental market is influenced by many factors like location, property size, amenities, and trends, making it complex. Traditional methods often struggle to predict rent accurately, but machine learning can handle this complexity by using large amounts of data and advanced algorithms.

Machine learning models, such as regression algorithms, decision trees, and ensemble methods, analyse historical rental data to predict future prices accurately. These models consider various factors, including location and property features, to make reliable predictions. By using techniques like feature selection and data preprocessing, these models can improve performance and provide useful insights.

The main goal is to create a strong model that can predict rental prices based on important features. This involves collecting and cleaning data, choosing the right machine learning algorithms, and evaluating their performance. Accurate predictions can help tenants find affordable housing, assist landlords in setting fair prices, and allow real estate companies to offer better market analysis.

This study examines different machine learning methods, aiming to develop a model that not only predicts rent accurately but also highlights the key factors affecting rental prices. Using these advanced techniques can make the housing market more efficient and transparent for everyone involved.

**Chapter 2**

# Literature Review

### Data Preparation:

**2.1**

Data preparation is the process of getting your data ready for analysis. It involves several steps to ensure that your data is accurate, complete, and formatted correctly.

First, you need to collect your data from various sources such as databases, spreadsheets, or APIs. Once collected, you must check for any errors, missing values, or inconsistencies in the data. This may involve cleaning the data by removing duplicates, correcting spelling mistakes, or filling in missing information.

Next, you need to organize the data in a format that is suitable for analysis. This includes structuring the data into rows and columns, renaming variables, and creating new variables if needed.

After organizing the data, you may need to transform it to make it more suitable for analysis. This could involve aggregating data, converting data types, or scaling variables.

Finally, you should validate the prepared data to ensure that it accurately represents the real-world phenomena you are studying. This may involve running checks or performing exploratory data analysis to identify any issues or patterns in the data.

Overall, data preparation is essential for ensuring that your analysis is based on reliable and high-quality data.

### Encoding:

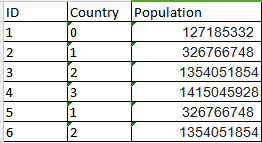
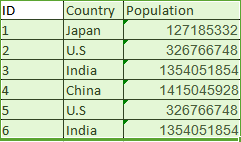
**2.2**

In ML models, it is often required to convert the categorical i.e. text features to its numeric representation. The accuracy of the model may shift by large scale by using the right encoding at the right scenario. It is an important pre- processing step for structured Datasets in Supervised Learning. Python provides some encoders for this purpose; one of them is ‘**LabelEncoder**’.

The ‘**LabelEncoder**’ encode the features/labels with values between **0** to ‘**n\_classes – 1**’ where ‘**n**’ is the number of distinct labels. If a label repeats,it assigns the same value to as assigned earlier.

### Example

In this sample Data, the ‘Country’ feature is encoded using ‘LabelEncoder’.



**Comparison of Un-encoded Data (on left) and Encoded Data (on right)**

**Tab. 1**

## Limitation

**Label Encoding** converts the data in machine readable form, but it assigns a unique number (starting from 0) to each class of data. This may lead to thegeneration of priority issue in training of Datasets. A label with high value may be considered to have high priority than a label having lower value.

## Feature Scaling

**2.3**

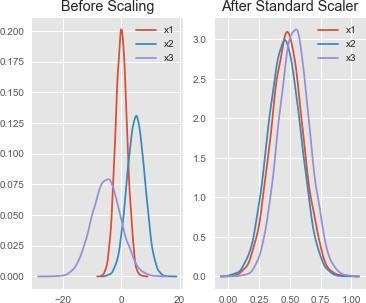
Feature Scaling is an important technique to **standardize** the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

The three most popular feature scaling methods are discussed here.

* **Standard Scaler (or, Standardization)**

The Standard Scaler assumes that the data is **normally distributed** within each feature and will scale them such that the distribution is now centered around 0, with a standard deviation of 1. The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

**… (1)**

If data is not normally distributed, this is not the best scaler to use.

**Example:** Upon feature scaling somesample data with the Standard Scaler,all features are now on the same scalerelative to one another.

**Comparison of Distribution of Actual (on left) and Scaled Data (on right)**

**Fig. 1**

### Regression Analysis

**2.4**

Regression Analysis is defined as follows :

In statistical modeling, **Regression Analysis** is a set of statistical processes for estimating the relationships between a dependent variable (also called the **Outcome** or **Target Variable**) and one or more independent variables (also called the **Predictors** or **Features**).

Regression helps investment and financial managers to value assets and understand the relationships between variables, such as commodity prices and the stocks of businesses dealing in those commodities.

Regression can help finance professionals as well as professionals in other businesses. Regression can also help predict sales for a company based on weather, previous sales, GDP growth, or other types of conditions.

### Assumptions Made

**2.5**

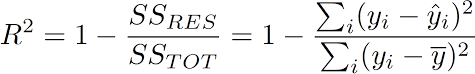
In order to interpret the output of a regression as a meaningful statistical quantity that measures real-world relationships, researchers often rely on a number of classical assumptions. These often include:

* The sample is representative of the population at large.
* The independent variables are measured with no error.
* Deviations from the model have an expected value of zero.
* The variance of the residuals is constant across all observations (also termed as **homoscedasticity**).
* The residuals are uncorrelated with one another.

**Metrics used for Evaluation of Regression Models**

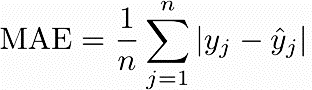
**2.6**

The most popular metrics for comparing and **measuring performance** of Regression Models, which will be used to compare the later- mentioned models, include:

* **R-Squared –** It determines the proportion of variance in the dependent variable that can be explained by the independent variable. This corresponds to the overall quality of the model. The **higher** the **R2**, the **better** the model.

**… (4)**

* **Mean Absolute Error –** It measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. It is robust against the effect of outliers. The **lower** the **MAE**, the **better** the model.

**… (5)**

### Types of Regression Models

**2.7**

Regression is done using a variety of algorithms or methods. Two of the mostpopular ones, which will be used, are mentioned below:

##### Gradient Boosting Regression

* + **XGBoost Regression**

These are discussed in detail in the next Sections

### Gradient Boosting Regression

**2.8**

Gradient Descent Regression is a fundamental optimization algorithm extensively used in machine learning for fitting models to data. Specifically applied in linear regression, it aims to minimize the disparity between predicted and actual values by iteratively adjusting model parameters. The algorithm operates by initializing random values for parameters, typically slope and intercept, and then systematically updating them in the direction of the steepest descent of the loss function. This iterative process continues until convergence, where further parameter adjustments yield negligible improvement. By continuously refining model parameters, Gradient Descent Regression efficiently navigates the parameter space to find optimal values. Its simplicity, versatility, and ability to handle large datasets make it a cornerstone in machine learning. From predicting housing prices to analyzing stock market trends, Gradient Descent Regression underpins numerous applications, showcasing its significance in modern data- driven decision-making processes.

### Algorithm

* **Initialize**: Start with a random value for the slope and intercept of the line.
* **Calculate Gradient**: Compute the gradient of the loss function with respect to the parameters (slope and intercept).
* **Update Parameters**: Adjust the parameters (slope and intercept) in the direction that minimizes the loss function.
* **Repeat**: Keep iterating steps 2 and 3 until convergence, meaning the parameters don't change significantly anymore

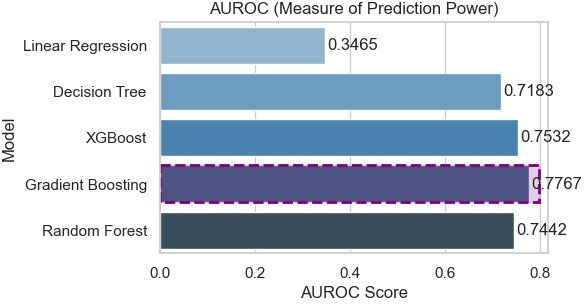
### Pros and Cons

**Pros:**

* + Versatile: Can be applied to various types of machine learning problems.
  + Efficiency: It's efficient for large datasets.
  + Flexibility: Works well with many types of loss functions.

**Cons:**

* + Convergence: May converge to a local minimum rather than the global minimum.
  + Sensitivity to Learning Rate: Performance heavily depends on choosing the right learning rate.
  + Not Suitable for All Problems: Some problems may have non- convex loss functions where gradient descent may struggle.



**Performance Highlight of Gradient Descent compared to other popular Algorithms**

**Fig. 2**

### XGBoost Regression

**2.9**

**eXtreme Gradient Boosting** (popularly known as **XGBoost**) is a Decision Tree based ensemble learning algorithm that employs a gradient boosting framework. It was developed as a research project, at the University of Washington, by **Tianqi Chen** and **Carlos Guestrin**, in 2016.

**Gradient Boosting** is a special case of boosting, where errors in sequentialmodels are minimized by the Gradient Descent algorithm.

XGBoost is an **optimized Gradient Boosting** algorithm where it fits a modelon the gradient of loss generated from the previous step, and is a perfect scalable combination of software enhancements and hardware optimization techniques to yield superior results using fewer computing resources in the shortest amount of time. [7]

### Algorithm

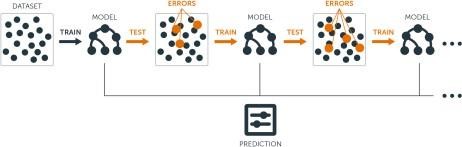
Given a Dataset with '**X**' being the **Feature set** and '**Y**' being the **Target variable**, the Gradient Boosting algorithm proceeds as follows:

**Step 1 –** A model to the data is fitted, **F1(x) = Y**

**Step 2 –** A model to the **residuals** is fitted,**h1(x) = Y - F1(x) Step 3 –** A new model is created, **F2(x) = F1(x) + h1(x)**

[ Note: **F2** is boosted version of **F1** ]

**Step 4 –** This is repeated till '**m**' iterations, **Fm(X) = Fm - 1(X) + hm -1(X)**



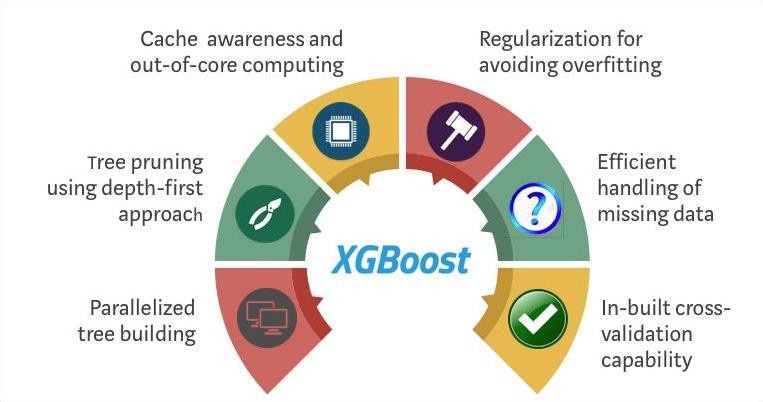
**Boosting of Decision Trees in XGBoost Algorithm**

**Fig. 3**

XGBoosting modifies this algorithm, so that it works with any **differentiable Loss function**, because Gradient Boosting alone does not produce equallypromising results for all Loss functions.

### Salient Features

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble Tree methods that apply the principle of boosting weak learners (like **CART**s generally) using the **Gradient Descent** architecture. However, the XGBoost improves upon the base GBM framework through the implementations of **System optimizations** and **Algorithmic enhancements**.



**Salient Features of XGBoost Algorithm**

**Fig. 3**

### System Optimizations

* **Parallelization:** XGBoost approaches the process of **sequential tree building** using parallelized implementation. This is possible due to the interchangeable nature of loops used for building base learners; the outer loop that enumerates the leaf nodes of a tree, and the second inner loop that calculates the features. This nesting of loops limitsparallelization because without completing the inner loop (more computationally demanding of the two), the outer loop cannot be started. Therefore, to improve run time, the order of the loops is interchanged using initialization through a global scan of all instances and sorting using parallel threads. This switching leads to improvement of the algorithmic performance by offsetting any parallelization overheads in computation.
  + **Tree Pruning:** The stopping criterion for Tree splitting within the GBM framework is **greedy** in nature, and depends on the negative loss criterion at the point of split. XGBoost uses ‘**max\_depth**’ parameter as specified instead of criterion first, and starts pruning trees backward. This ‘**depth-first**’ approach improves the computational performance significantly.
  + **Hardware Optimization:** XGBoost has been designed to make **efficient use** of hardware resources. This is accomplished by cache awareness by allocating **internal buffers** in each thread to store gradient statistics.Further enhancements such as use of ‘**out- of-core**’ computing optimizeavailable disk space while handling big data-frames that do not fit into memory.

### Algorithmic Enhancements

* + - **Regularization:** It penalizes more complex models using both

**LASSO** (**L1**) and **Ridge** (**L2**) regularization to prevent overfitting.

* + - **Sparsity Awareness:** It naturally admits **sparse features** for inputs by automatically ‘learning’ best missing value depending on training loss and handles different types of sparsity patterns found in the Dataset more efficiently.
    - **Continued Training:** It supports further boosting an already fitted modelon new data.
    - **Weighted Quantile Sketch:** It uses the distributed weighted Quantile Sketch algorithm to effectively find the optimal split points among the weighted datasets.
    - **Cross-validation:** It comes with built-in cross-validation method at eachiteration, taking away the need to explicitly program this search and to specify the exact number of boosting iterations required in a single run.

### Performance Record

As observed the Gradient Boost model has the best combination of prediction **performance** and processing **time** compared to other algorithms. Other rigorous benchmarking studies have produced similar results.

### Feature Selection

**2.10**

Feature Selection is a process of selection of a subset of relevant features (or Predictors) from all the features, which is used to build the learning model. It is one of the core concepts in Machine Learning which influences and impacts the performance of the model, as the features are used to train the model. Irrelevant or partially relevant features can negatively impact the model performance. [9]

With large number of features, data analysis is challenging to the engineers in the field of Machine Learning and Data Mining. Feature Selection gives an effective way to solve this problem by **removing irrelevant** and **redundant features**, which thereby can reduce computation time, improve learning accuracy, and facilitate a better understanding for the learning model.

### Benefits Obtained

**2.11**

The benefits of performing Feature Selection are as follows:

* + **Improvement in Accuracy:** Less misleading data means the modelingaccuracy improves.
  + **Reduction of Overfitting:** Less redundant data means less opportunityto make decisions based on noise.
  + **Shortening of Training Time:** Fewer data points reduce the algorithmcomplexity and models train faster.

The number of features to keep is decided based on a **trade-off** between **Predictive Accuracy** versus **Model Interpretability**, because if there are less number of features, then it is easy to interpret the model, less likely to overfit but it will give low prediction accuracy, while if there are large number of features then it is difficult to interpret the model, more likely to overfit but it will give high prediction accuracy.

### Methods Employed

**2.12**

Among many methods available for performing Feature Selection, the twomost popular ones, which will be used, are discussed here:

* **Correlation Matrix with Heatmap –** Correlation states how the features are **related** to each other or the Target variable. It can be either positive (increase in one value of feature increases the value of the Target variable) or, negative (increase in one value of feature decreases the value of the Target variable). The Heatmap makes it easy to identify which features are most related to the Target variable; the Heatmap of correlated features are plotted using the '**seaborn**' library.

In this project, the same is used to overview the relation of all the features in the Dataset with the Target variable, during the Exploratory Analysis.

* **Feature Importance Property –** The importance of each feature in the dataset can be obtained by using the '**feature\_importances\_**' property of the model. It provides a score for each feature; the higher the score, more **important** or **relevant** is the feature towards the prediction of the Target variable. It is an inbuilt class that comes with many Tree Based Regressors.

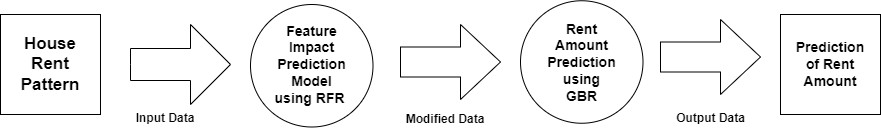
In this project, the Random Forest Regression algorithm is applied for computing and recording the possible order of feature importance, and then, extracting the top five features from the Dataset.

**Chapter 3**

## OUR PROPOSED WORK

Our proposed model efficiently predicts house rents by leveraging essential features identified through random forest regression. These key characteristics are then used in gradient boosting for rent prediction, ensuring high accuracy and reliability.

Predicting house rent accurately benefits both landlords and tenants. For landlords and companies, understanding the rental patterns and amounts tenants are willing to pay can stabilize rental income and enhance competitiveness. By capturing tenants' preferences and behaviors across various activities, personalized rental experiences can be created.



**Block Diagram of our Project**

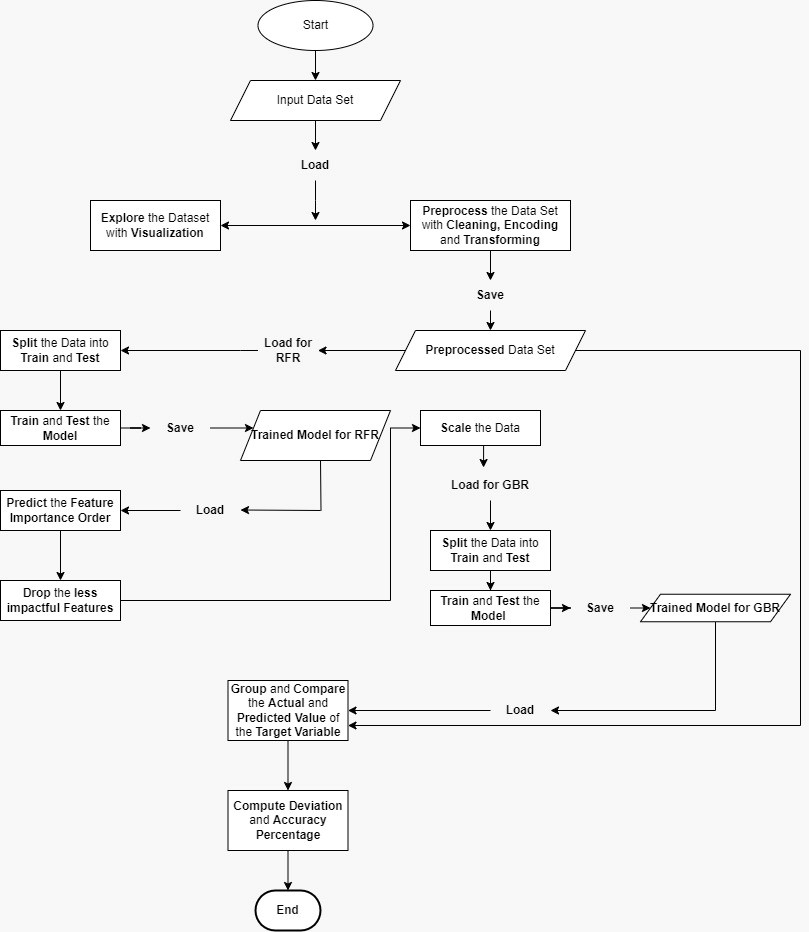
**Fig 4**

Our methodology involves two models: the first is a Random Forest Regression (RFR) model that identifies the most impactful features from the tenant and property data. The second model, Gradient Boost Regression, uses these features to predict the rent amount. This dual- model approach ensures that only the most relevant data influences rent predictions.

This system aids landlords in setting fair and competitive rents while helping tenants understand the appropriate rent for a property, ensuring they don't overpay. By leveraging these predictive models, both parties benefit from more accurate and fair rental pricing.

### Flow Chart

**3.1**



**Detailed Flow Chart of the Project**

**Fig. 5**

As depicted by the Flow Chart, the input Dataset is explored visually and pre- processed using several techniques. Then, it is split into two; one is used for training and testing the Random Forest Regression Model, which thenpredicts the impactful features, and this is fed into the second model which is built with Gradient Boost Regression. Finally, the Target variable is predicted for the whole Dataset, and compared with actual values of the Target variable to measure the Absolute Mean Deviation and Mean Accuracy percentages.

### Algorithm

**3.2**

Step 1: Libraries and functions are loaded. Step 2: The dataset is loaded.

Step 3: Exploratory analysis is done with visualization. Step 4: Data is preprocessed.

Step 4.1: Variables are introduced.

Step 4.2: None (missing) values are removed. Step 4.3: Abnormal data is removed.

Step 4.4: Categorical data is transformed. Step 5: Data is split into train and test sets.

Step 6:The Random Forest Regressor model is fitted, trained, and tested. Step 7: Important features are predicted by the RFR model.

Step 8: Less important features are dropped. Step 9: Data is scaled.

Step 10: Data is split into train and test sets again.

Step 11: The Gradient Boosting Regressor (GBR) model is fitted, trained, and tested.

Step 12: The actual and the predicted values are grouped and compared.

**Chapter 4**

# Result Analysis

#### Dataset Used

**4.1**

The Dataset we used here was, collected through a survey. It contains some information about house demographic along with the corresponding renting amounts. It consists of **4746 instances** with **12 features**. [12]

The **variables** of the Dataset are given as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| **Posted On** | Categorical | Identification of the date of publish |
| **BHK** | Numeric | No. of rooms present |
| **Rent** | Numeric | Amount asked for the property |
| **Size** | Numeric | Size of the property |
| **Floor** | Categorical | Floor at which property is available |
| **Area Type** | Categorical | Category of the City |
| **Area Locality** | Categorical | Locality of property |
| **City** | Categorical | Current City of the property |
| **Furnishing Status** | Categorical | Current situation property |
| **Tenant Preferred** | Categorical | Category of tenant |
| **Bathroom** | Numeric | No. of bathrooms present |
| **Point of Contact** | Categorical | Handed by who |

**Variable Description of the Dataset**

**Tab. 2**

Here, the Rent amount indicated by the variable '**Rent**' is the Target

variable, and the remaining features will be used as independent variables to predict the amount of Rent of a Property.

#### Libraries and Functions Used

**4.2**

In this Section, according to the Step 1 of the Algorithm, the following needed Libraries and Functions are imported and used for the Project:

|  |  |  |
| --- | --- | --- |
| **Libraries / Functions** | **Parent Library** | **Description / Requirements** |
| **numpy** | - | Data Cleaning |
| **pandas** | - | Data Processing |
| **pyplot** | matplotlib | Data Visualization |
| **seaborn** | - | Data Visualization |
| **LabelEncoder** | sklearn.preprocessing | Data Encoding with Label Encoder |
| **RandomForestClassifier** | sklearn.ensemble | Build Random Forest Classifier Model |
| **RandomForestRegressor** | sklearn.ensemble | Build Random Forest Regression Model |
| **GradientBoostingRegress or** | sklearn.ensemble | Build Gradient Boosting Regressor Model |
| **DecisionTreeRegressor** | sklearn.tree | Controls randomness of estimator |
| **XGBRegressor** | xgboost | Build XGBoost Regression Model |
| **LinearRegression** | sklearn.linear\_model | Implementation of Linear Regression |
| **StandardScaler** | sklearn.preprocessing | Feature Scaling with Standard Scaler |
| **mean\_squared\_error** | sklearn.metrics | Evaluate Regressors with MAE |
| **r2\_score** | sklearn.metrics | Evaluate Regressors with R-Squared |
| **train\_test\_split** | sklearn.model\_selection | Split Dataset into Train and Test Sets |

**Libraries and Functions imported for the Project**

**Tab. 3**

#### Exploratory Data Analysis

**4.3**

In this Section, according to the Steps 2 and 3 of the planned Algorithm, the input Dataset is examined and visually analyzed.

The **input** Dataset has values as follows:



**Overview of the values of the input Dataset**

**Tab. 4**

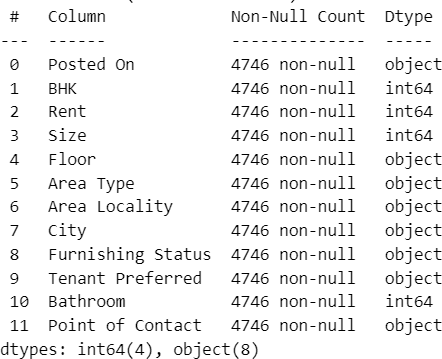
Statistically, the input Dataset is as follows:



**Statistics of the values of the input Dataset**

**Tab. 5**

The types of the variables are identified as follows:



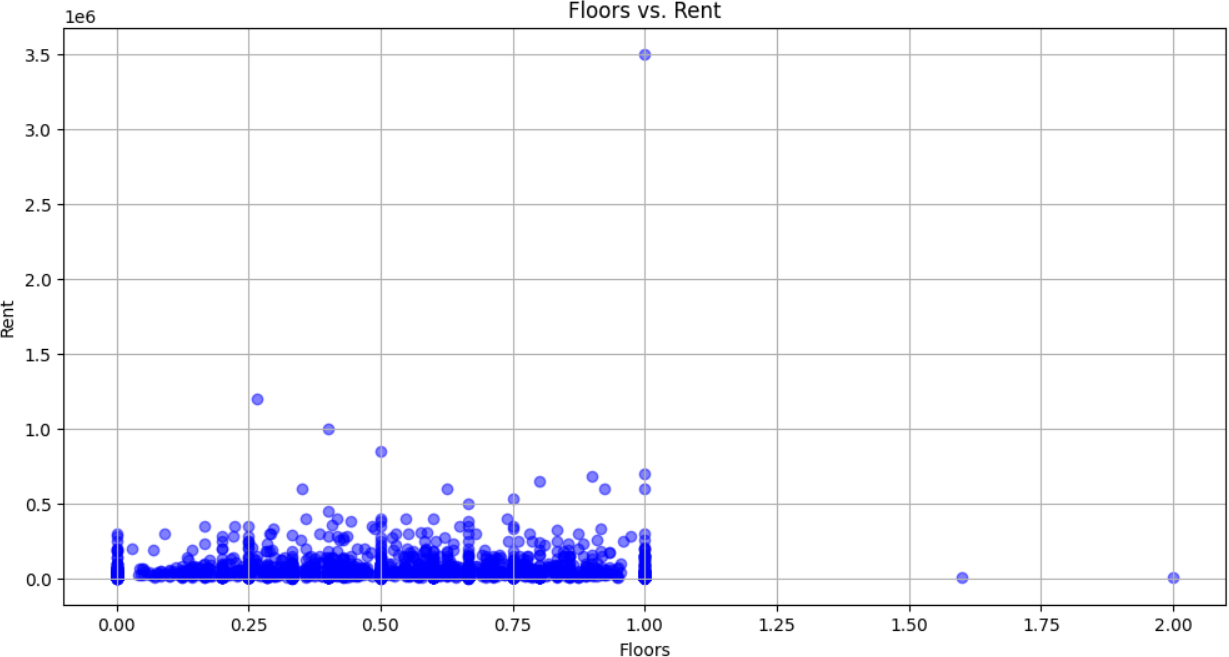
**Variable types of the input Dataset**

**Fig. 6**

#### Visualization of Data Trends

**4.4**

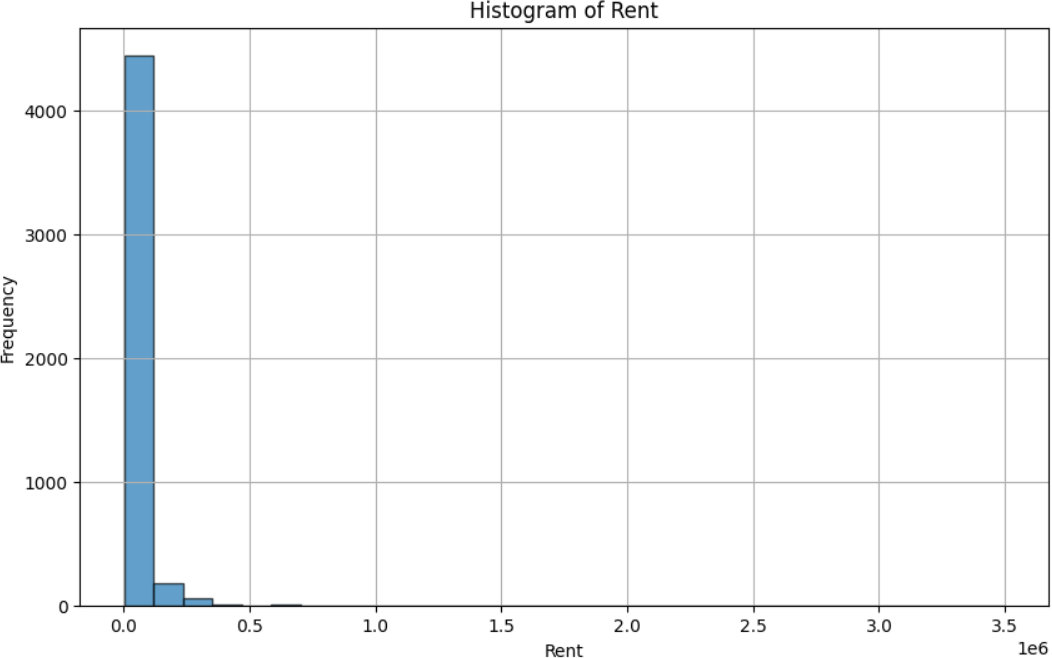
The Floor to Rent relation is visualized as below:



**Scatterplot visualizing Floors Vs. Rent**

**Fig. 7**

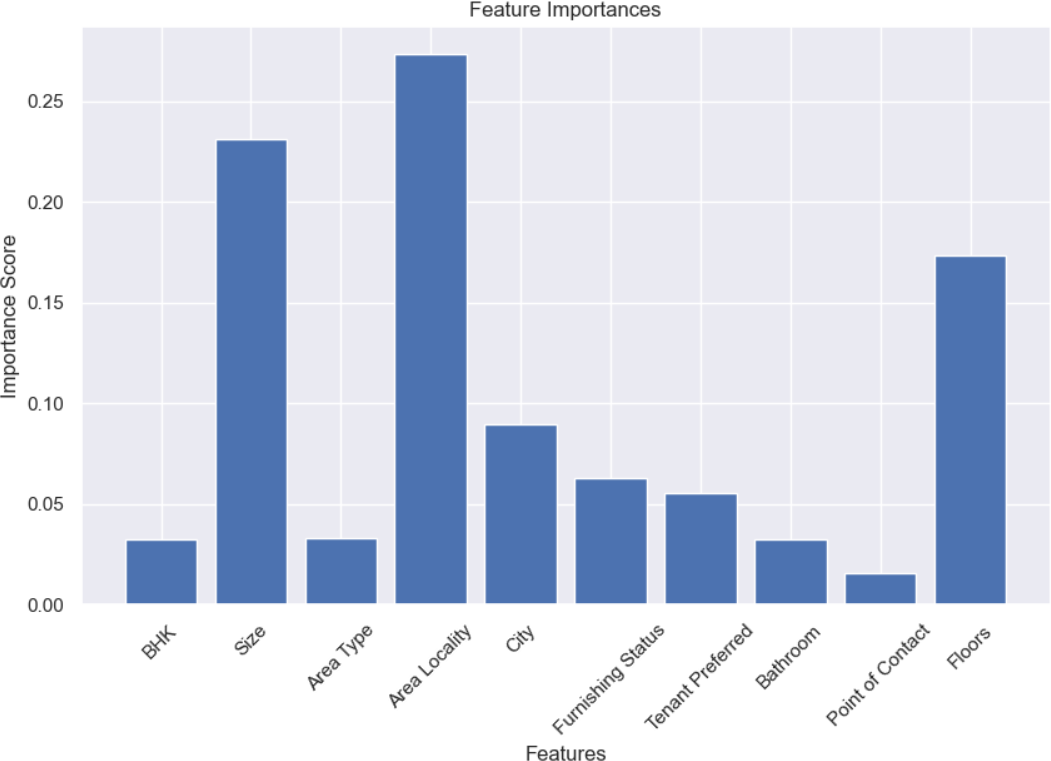
A graph for the Rent and its frequency, is visualized as below:



**Graph of Rent vs Frequency**

**Fig. 8**

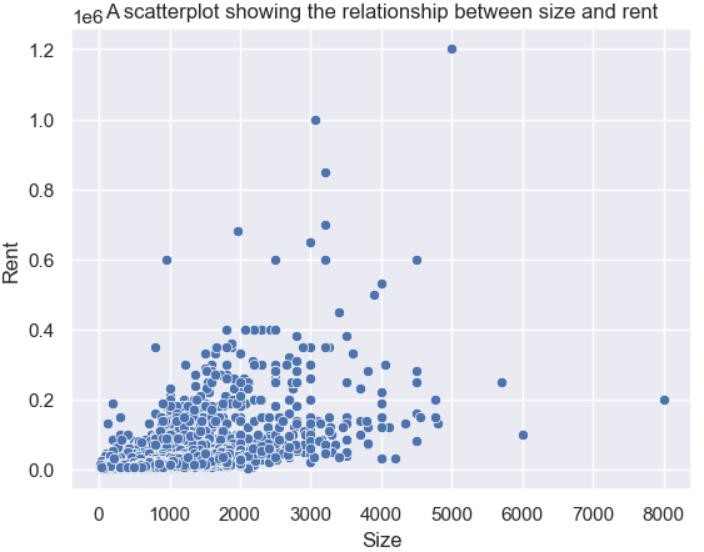
The graph of Features versus their Importance Score, is visualized as below:



**Graph of Feature Importance: Features vs Importance Score**

**Fig. 9**

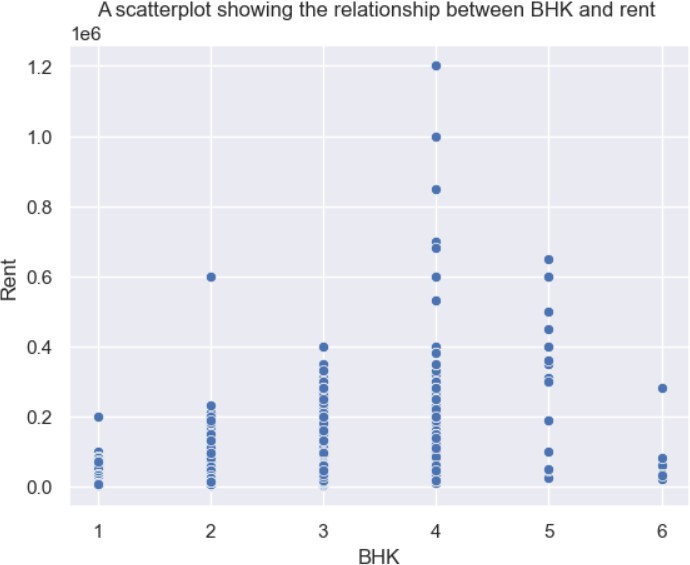
Visualization of Size to Rent:



**Scatterplot showing the relationship between Size and Rent**

**Fig. 10**

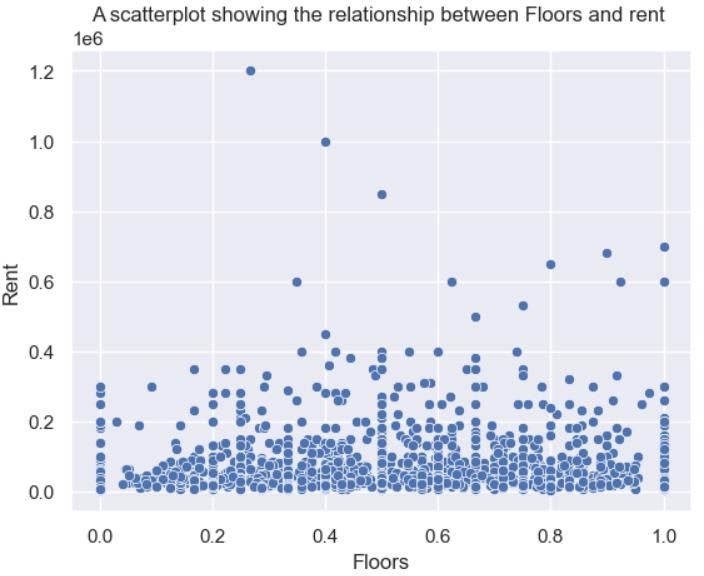
Relationship between BHK and Rent, is visualized as below:



**Scatterplot showing BHK vs RENT plotting**

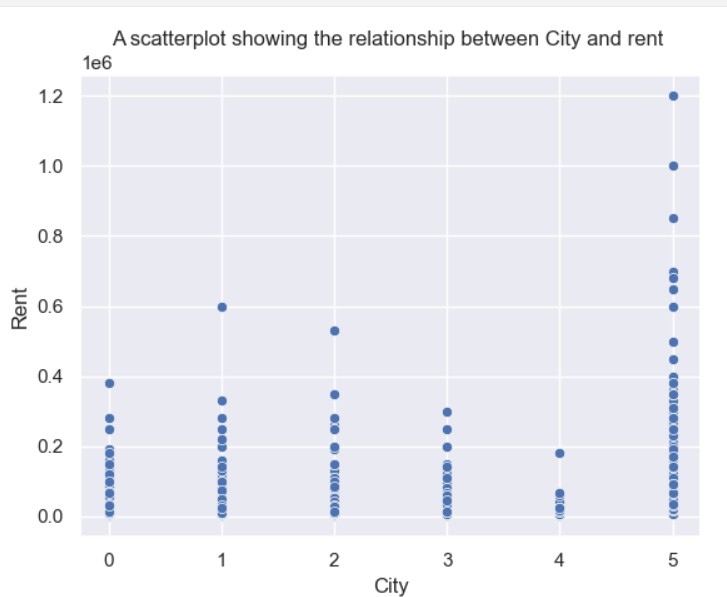
**Fig. 11**

Relation between Floors to Rent, is visualized below:



**Scatterplot visualizing the relation of Floors to Rent**

**Fig. 12**

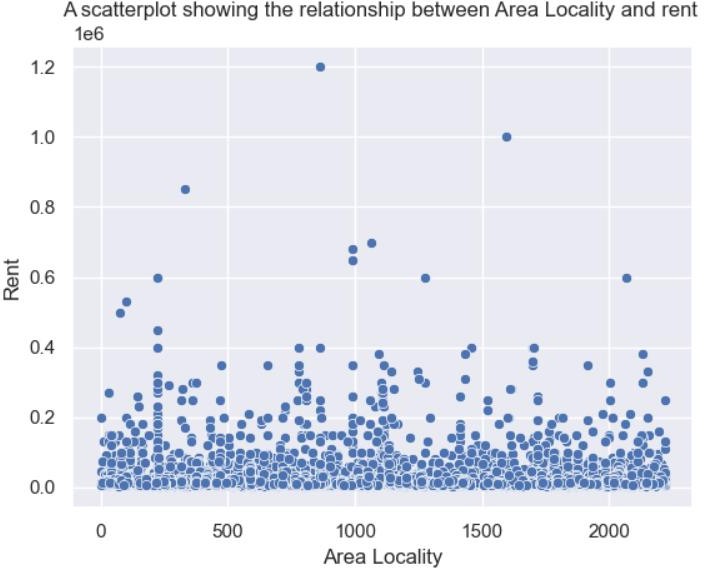
This graph visualizes the relation between City and Rent:

**Scatterplot for City vs Rent**

**Fig. 13**

Relationship between Area Locality and Rent is visualized as below:

**Fig. 14**



**Graph showing relationship between Area Locality and Rent**

The **Correlation Matrix** with **Heatmap** is shown as below before the pre- processing of the input Dataset begins:

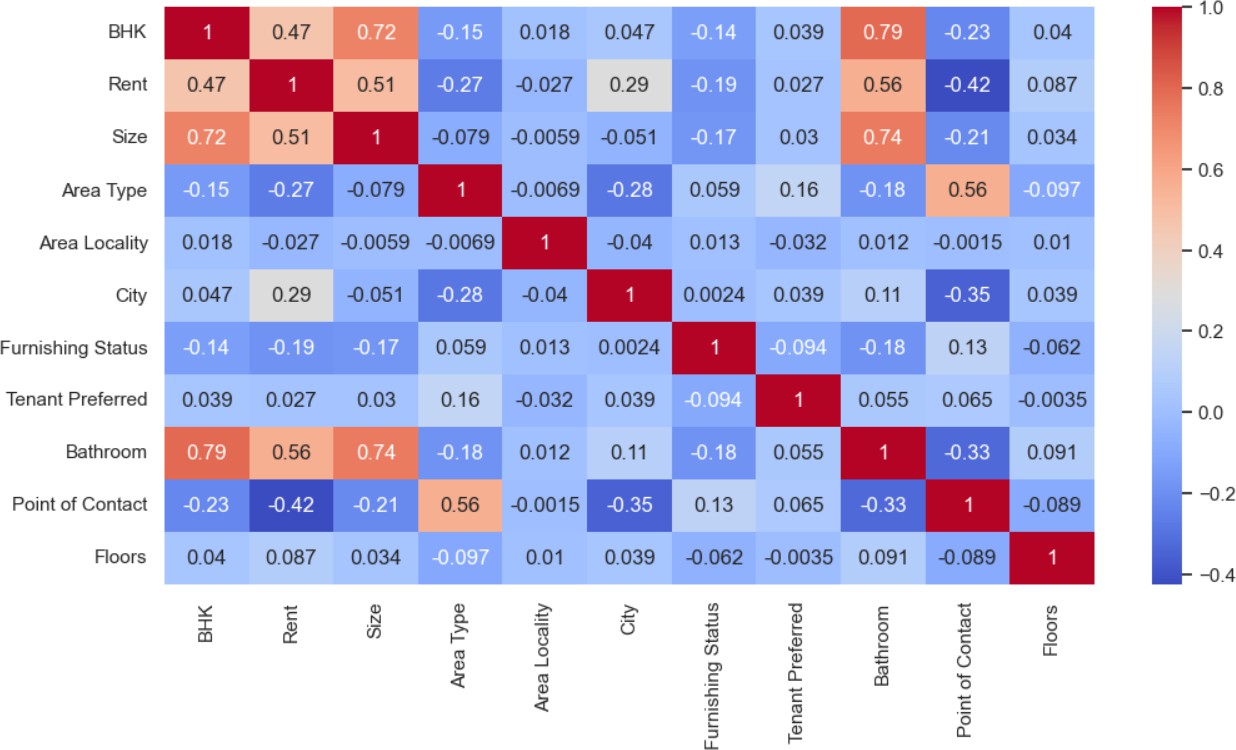


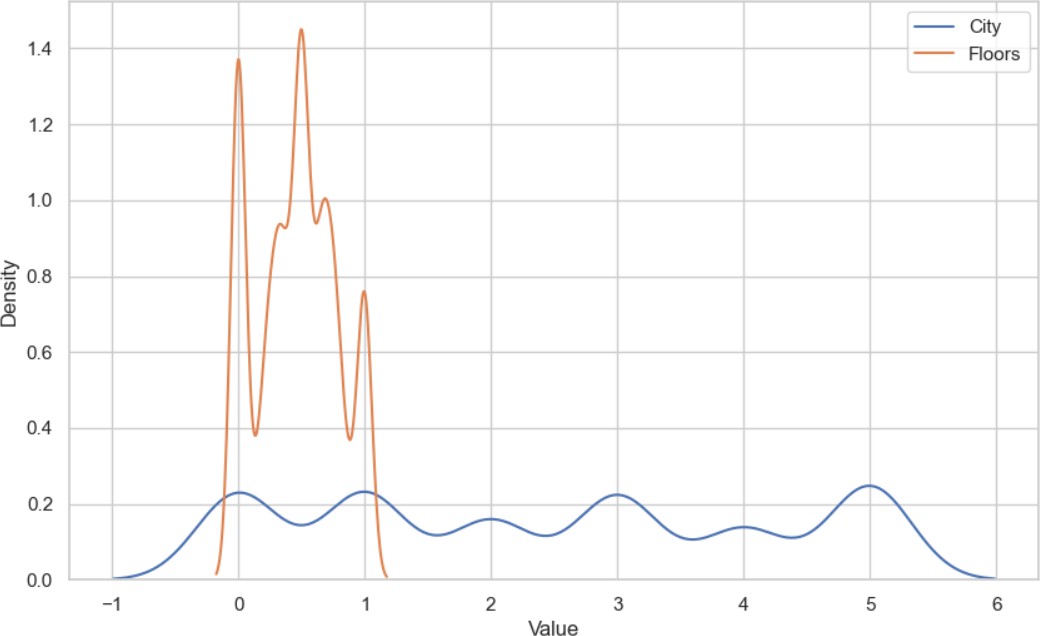
Fig.15

**Correlation Matrix**

#### Preparation

**4.5**

The distribution of the input Dataset, before feature scaling, is as follows:



**Distribution of the Features in the Unscaled input Dataset**

**Fig. 16**

In this Section, according to the Steps 4 and 5 of the planned Algorithm, the input Dataset is prepared by cleaning, encoding, filling and transforming.

As a result of introduction of indicator variables (**‘Current Floor’ and ‘Total Floor’**) and encoding of dummy variables into the Dataset, the total number of features grew from 12 to 14. Below is the pre-processed Dataset:



**Growth in number of Features after 1st pre-processing of the input Dataset**

**Tab. 6**

As a result of introduction of another indicator variable (**‘Floors’**) and encoding of dummy variables into the Dataset, the total number of features grew from 14 to 15 this time



**Overview of the values after 2nd pre-processing of Dataset**

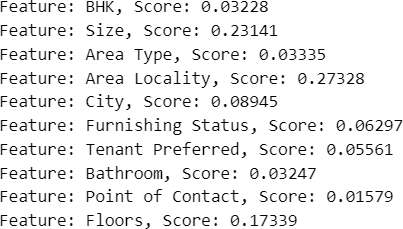
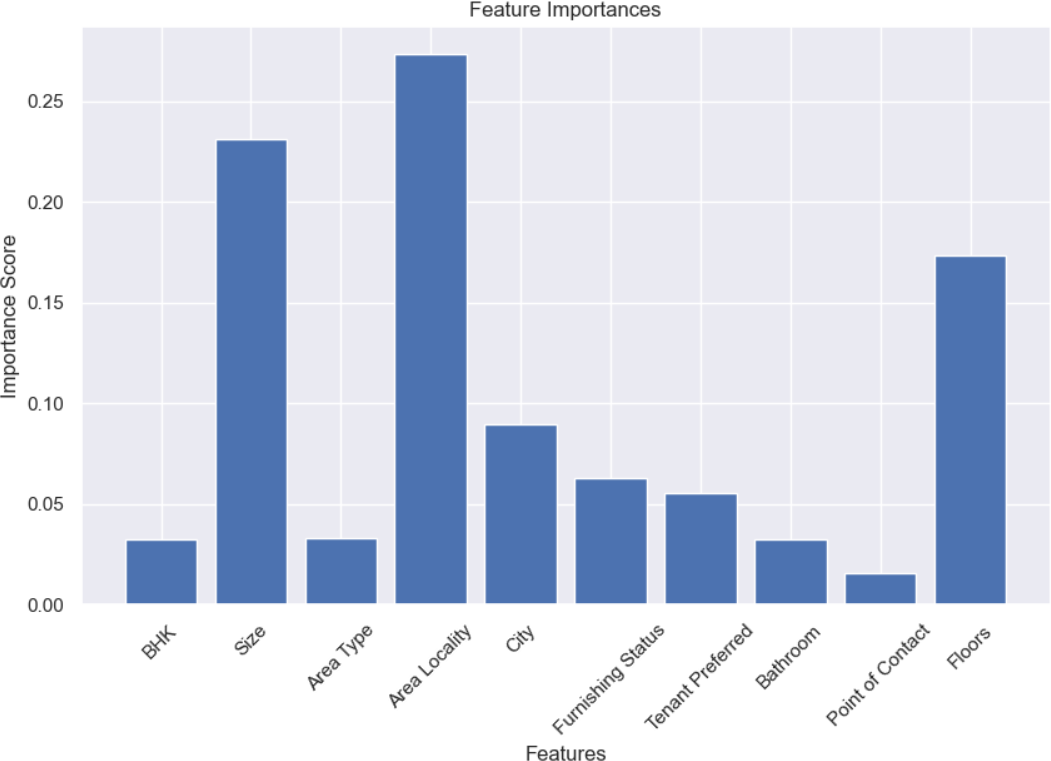
**Tab. 7**

#### Feature Selection with Random Forest Model

**4.6**

Feature selection is done using Random Forest Model dropping

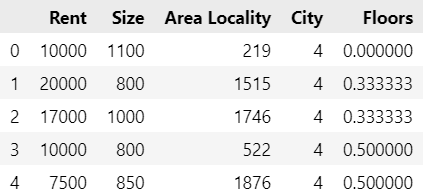
:’Posted On', 'Rent', 'Floor', 'Current Floor', 'Total Floors’ and visualizing the correlation of the features



**The values of features selection for RF Model and the graph plotting the same**

**Fig.17**

Further dropping - ’ BHK', 'Area Type’, ‘Furnishing Status’, ‘Tenant’, ‘Preferred', 'Bathroom', 'Point of Contact’



**Overview of the values of the Scaled Dataset**

**Tab. 8**

We utilized the standard Scaler, known for its robustness to outliers within the dataset. This scaler was chosen as optimal for the Random Forest Regression Model. Additionally, it ensures uniformity by converting all variable types in the dataset to 'float64'.

As the RFR Model is trained and tested, it is used for predicting the **Feature Importance Order,** from this Feature Importance Order, the **top five features** are **selected** and the **rest** are **dropped**.

This is tabulated as follows:

|  |  |
| --- | --- |
| **Objective** | **Features** |
| **Selection** | Rent, size, Area Locality, City, Floors |
| **Rejection** | Posted On, BHK, Floor, Area Type, Furnishing Status, Tenant Preferred Bathroom, Point of Contact, Current Floor, Total Floors |

**Features Selection and Rejection based on the predicted Feature Importance Order**

**Tab. 9**

#### Target Prediction with Gradient Boosting Model

**4.7**

The distribution of scaled datasets reveals that the Standard Scaler is the most suitable, dropping redundant features and eliminating most outliers. It's selected as the best scaler for the Gradient Boosting Regression Model, converting all variable types to 'float64'.

Following this, the dataset's independent and dependent features have reduced, enhancing accuracy potential for the Gradient Boosting Model. The Train set is fitted into the optimized model, evaluated on both Train and Test Sets.

Subsequently, the trained Gradient Boosting Model predicts the target variable for the entire input dataset, dropping less impactful features and scaling with the same scaler used in training. The predicted dataset is then post-processed and saved, retaining all original input features.

**Comparison Among Different Algorithms**

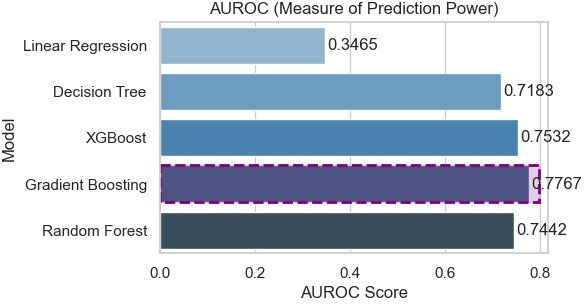
**4.8**

The table below compares different algorithms using R² Score and RMSE Score:

|  |  |  |
| --- | --- | --- |
| **Name** | **R2Score** | **Rmse** |
| **Linear Regression** | 0.3465 | 43420.91 |
| **Decision Tree** | 0.7183 | 28528.93 |
| **XGBoost** | 0.7532 | 26684.64 |
| **Random Forest** | 0.7442 | 27166.27 |
| **Gradient Boosting** | 0.7767 | 25396.49 |

**Variable Description of the Comparison Dataset**

**Tab. 10**



**Prediction power of different**

Fig.18

**Chapter 5**

# Conclusion

Our proposed house rent prediction model using Machine Learning algorithms can **save** an enormous amount of **time** and **effort** for students, corporate, or anybody seeking a house to rent and also eliminate most problems of brokers. Our model is robust to outliers in data and can make predictions on unknown data with practically high scale precision. It holds immense potential to transform the way property holders interact with their tenants. Our model have the ability to catch up more valuable insights into the recent price and patterns involving renting of property at most probable conditions.

In our proposed model, Random Forest as well as Gradient Boosting Regression algorithms are used in conjunction to construct a predictive model to analyze rent of property of a Dataset. It predicts the rent of property based on different features like **size, floor, area locality**, etc. selected from a given pool of twelve features based on the range of impact they have on the target variable. It is observed to have an **Absolute Mean Deviation percentage** of 0.7767 and **Mean Accuracy percentage** of 25396.49 which is highly accurate for a medium scale Dataset like the one used.

Our innovative house rent predictor offers a clear and detailed explanation of the predictive analysis, making it an invaluable resource for students, job seekers, and anyone relocating and searching for rental accommodations in unfamiliar areas.

# REFERENCES

1. Asniar & Kridanto Surendro, "**Predictive Analytics for Predicting Customer Behavior**", 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), IEEE, 12th September 2019.
2. Shichao Zhang, Chengqi Zhang & Qiang Yang, "**Data Preparation for Data Mining**",Applied Artificial Intelligence, ResearchGate Conference Paper, May 2003.
3. Sebastian Raschka, "**About Feature Scaling and Normalization and the Effect of Standardization for Machine Learning Algorithms**", ResearchGate Conference Paper,July 2014.
4. Gülden Kaya Uyanık & Neşe Güler, "**A Study on Multiple Linear Regression Analysis**", 4th International Conference on New Horizons in Education, 20th January 2014.
5. Jehad Ali, Rehanullah Khan, Nasir Ahmad & Imran Maqsood, "**Random Forests and Decision Trees**", International Journal of Computer Science Issues (IJCSI), Volume 9, September 2012.
6. Gerard Biau, “**Analysis of a Random Forests Model**”, Journal of Machine LearningResearch 13, April 2012.
7. Tianqi Chen & Carlos Guestrin, "**XGBoost: A Scalable Tree Boosting System**", August 2016.
8. Ting Hu & Ting Song, "**Research on XGBoost Academic Forecasting and Analysis Modelling**", Journal of Physics: Conference Series, Volume 1324, The Second International Conference on Physics, Mathematics and Statistics, 22nd- 24th May 2019.
9. Huan Liu, "**Feature Selection**", Encyclopedia of Machine Learning, Springer, 2011.
10. Jianyu Miao & Lingfeng Niu, "**A Survey on Feature Selection**", Promoting BusinessAnalytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016), 6th August 2016.
11. **Dataset used**: https://[www.kaggle.com/datasets/iamsouravbanerjee/house-rent-](http://www.kaggle.com/datasets/iamsouravbanerjee/house-rent-) prediction-dataset/data