**REALESTATE PREDICTOR: AI-DRIVEN REAL-TIME PROPERTY VALUATION**

*Submitted in partial fulfilment of the requirements for the degree of*

**Bachelor of Technology**

in

# Information Technology

*by*

**V. M. Rasheed   
20BIT0216**

## Under the guidance of

**Prof. GUNDALA SWATHI MAM**

**Score VIT, Vellore.**



May 2024

### DECLARATION

I hereby declare that the thesis entitled “REALESTATE PREDICTOR: AI-DRIVEN REAL-TIME PROPERTY VALUATION” submitted by me, for the award of the degree of *Bachelor of Technology in Information Technology* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Gundala Swathi mam.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 08.05.2024 **Signature of the Candidate**

### CERTIFICATE

This is to certify that the thesis entitled “REALESTATE PREDICTOR: AI-DRIVEN REAL-TIME PROPERTY VALUATION” submitted by Valluru Mohammad Rasheed (20BIT0216), School of Computer Science Engineering and Information Systems (SCORE), VIT, for the award of the degree of *Bachelor of Technology in Information Technology*, is a record of bonafide work carried out by him / her under my supervision during the period, 05. 01. 2024 to 09.05.2024, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfils the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Vellore

Date : 08.05.2024 **Signature of the Guide**

**Internal Examiner External Examiner**

Head of the Department

Programme

## ACKNOWLEDGEMENTS

The School of Information Technology's Prof. Gundala Swathi mam., who performed as our guide, is to be commended for her acclaimed leadership and unwavering support, which helped us study and successfully complete the project.

We are appreciative to the School of Information Technology's Dean, Dr. Sumathy S mam, and the Department Head, Dr. Prabhavathy p mam, for giving us the chance to complete our capstone project at VIT University. I sincerely thank all the professors and employees who serve as the foundation of our institution for their non-self-centred excitement and timely encouragements, which inspired me to get the necessary information to successfully complete my project. I want to express my gratitude to my parents for their help. Finally, I want to thank and appreciate everyone who has contributed to the successful completion of this project, whether directly or indirectly.

Place: Vellore V. M. RASHEED

Date: 08.05.2024 Name of the Candidate

**Executive Summary**

“RealEstate Predictor: AI-Driven Real-Time Property Valuation" is a groundbreaking initiative that uses machine learning to anticipate property prices. Initially, the study used classic linear regression approaches as well as Lasso and Ridge regression, reaching a modest 79% accuracy. However, these models were insufficient due to significant error rates that exceeded 60%. To enhance, a Support Vector Machine (SVM) was used, but the results were comparable at 79% accuracy. The use of Random Forest and XGBoost algorithms resulted in considerable improvements in accuracy to 90% and 88%, respectively. Further optimisation through hyperparameter tweaking increased XGBoost's accuracy to a respectable 90%, with cross-validation reaching an astounding 95%.

To guarantee accessibility and usability, the final model was stored in a pickle file and then incorporated into a Flask-based web application. This application offers stakeholders with an easy way to access and engage with the forecasts. This project is based on insights gained from a thorough evaluation of ten research articles, which strengthens its dependability and efficacy in the volatile real estate market. "RealEstate Predictor" is a promising solution that provides both accuracy and usability for real-time property appraisal needs.

**CONTENTS Page No.**

**Acknowledgement** 4

**Executive Summary** 5

**Table of Contents** 6

**List of Figures** 7

**List of Tables** 8

**Abbreviations** 9

**Symbols and Notations** 9

1. **INTRODUCTION** 
   1. **Objective** 10
   2. **Motivation** 10
   3. **Background** 11
2. **LITERATUE REVIEW** 12
3. **PROJECT DESCRIPTION AND GOALS** 17
4. **TECHNICAL SPECIFICATION**
   1. **Hardware Requirements** 18
   2. **Software Requirements** 18
   3. **Data Used** 18
5. **DESIGN APPROACH AND DETAILS** 
   1. **System Architecture** 19
   2. **Module description** 20
6. **PROPOSED METHODOLOGY** 21
7. **SCHEDULE, TASKS AND MILESTONES.** 24
8. **PROJECT DEMONSTRATION**. 25
9. **RESULT & DISCUSSION**. 41
10. **CONCLUSION AND FUTURE SCOPE**. 42
11. **REFERENCES**. 42

### List of Figures

**Figure No. Title Page No.**

|  |  |  |
| --- | --- | --- |
| 4.1 | Sample image of Data Set | 18 |
| 5.1.1 | System Architecture Diagram | 19 |
| 6.1.1 | Linear regression Graph and Formula | 21 |
| 6.1.2 | Lasso Formula | 22 |
| 6.1.3 | Ridge Formula | 22 |
| 7.1 | Gantt Chart Diagram | 24 |
| 8.1 | Describing the Numerical Data | 25 |
| 8.2 | Correlation Chart | 25 |
| 8.3 | Heat Map | 26 |
| 8.4 | Before removing outliers. | 27 |
| 8.5 | After removing outliers | 28 |
| 8.6 | Encoded Data | 28 |
| 8.7 | Model Output | 32 |
| 8.8 | Web Application Image | 40 |
| 8.9 | Input Image | 40 |
| 8.10 | Output Image | 41 |

### List of Tables

**Table No. Title Page No.**

|  |  |  |
| --- | --- | --- |
| 2.1 | Literature Survey | 12 |
| 8.1 | Accuracy & RMSE values Comparison Before Tuning | 30 |
| 8.2 | Accuracy & RMSE values Comparison After Tuning | 31 |

## List of Abbreviations

|  |  |
| --- | --- |
| MLR | Multi Linear Regression |
| SVM | Support Vector Machine |
| XGBoost | Extreme Gradient Boost |
| RMSE | Root Mean Square Error |
| PKL | Pickle file |

**Symbols and Notations**

|  |  |
| --- | --- |
| ∑ | Summation |
| β0 | Intersection value |
| βn | Slope |
| λ | regularizations parameter |

1. **INTRODUCTION**

* 1. **OBJECTIVE**

The "Real Estate Predictor: AI-Driven Real-Time Property Valuation" project is on a transformational quest to redefine existing real estate valuation approaches. Rooted on an innovative vision, its fundamental purpose is to pioneer a cutting-edge real-time property assessment system powered by powerful artificial intelligence (AI) and machine learning (ML) technology. The primary goal is clear: to improve the precision of property appraisals by combining a variety of regression techniques such as linear, Lasso, Ridge, SVM, Random Forest, and XGBoost.

The project's ideology is based on a strong dedication to responsiveness, which recognizes the flux of the real estate sector. By adopting flexibility, the system can quickly react to the market's ever-changing dynamics, guaranteeing that evaluation findings are relevant and accurate in real-time circumstances.

Transparency and equity are guiding concepts throughout the project's development. The addition of explainability elements seeks to demystify the AI model's decision-making process, boosting comprehension and confidence among users ranging from homeowners to real estate professionals. Furthermore, the use of fairness-aware ML algorithms mitigates inherent biases, ensuring equal property evaluations across property kinds and geographic areas. In essence, the initiative represents a forward-thinking strategy, prepared to revolutionise real estate assessment procedures while adhering to values of openness and justice.

* 1. **Motivation**

The "Real Estate Predictor: AI-Driven Real-time Property Valuation" project is a game-changer in the real estate market. Traditional property valuation systems have long been reliant on subjective evaluations or out-of-date data, leading to costly price mistakes and inefficiencies. This project offers a solution by integrating state-of-the-art AI and ML technologies to provide more accurate and timely property appraisals.

I understand that the real estate market is highly volatile and requires a flexible and adaptive approach to assessment. This is why my real-time evaluation system has been designed to provide stakeholders with the tools they need to navigate rapidly changing market situations and make informed decisions. Whether you are a buyer, a seller, or a real estate investor, my project empowers you to make sound decisions based on real-time data.

Transparency and justice are other motivating factors behind the idea. In many circumstances, the property valuation process is unclear, creating distrust and ambiguity among parties. By merging explainability characteristics and fairness-aware ML algorithms, the project aims to improve transparency and assure equitable treatment across property kinds and geographic regions.

Overall, the goal of this project is to revolutionise traditional real estate assessment methods, improving accuracy, timeliness, openness, and fairness in the process. By leveraging the power of AI and ML, the project hopes to provide stakeholders with the knowledge they need to effectively navigate the complicated real estate sector.

* 1. **Background**

The "Real Estate Predictor: AI-Driven Real-time Property Valuation" project stems from the ongoing problems and limits of existing real estate valuation approaches. Historically, property appraisals were highly reliant on manual methods, subjective judgements, and historical data analysis. However, these methodologies frequently produce mistakes, inefficiencies, and discrepancies in value outcomes.

Furthermore, the real estate market is essentially dynamic, driven by a wide range of factors including economic circumstances, market trends, and demographic developments. Traditional valuation methodologies fail to keep up with these fast changes, producing obsolete or inappropriate valuation estimates.

In recent years, advances in AI and machine learning technology have created new prospects to revolutionise the real estate assessment process. Large datasets, predictive modelling approaches, and sophisticated algorithms may now be used to create real-time property evaluation systems with unparalleled accuracy and responsiveness.

Furthermore, there is a growing desire for openness and justice in the real estate market. Stakeholders, including homeowners, purchasers, sellers, and lenders, are increasingly looking for transparency and equity in property assessment methods. Addressing these challenges needs novel methods to valuation models that prioritise explainability, accountability, and bias reduction.

1. **Literature Review**

Table 2.1 Literature Survey

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Title of the Paper And year** | **Algorithms used** | **Main Focus** | **limitations** | **Scope for future work** |
| 1. | Value Prediction of Real Estate Using Machine  Learning (2023) | Linear regression, Lasso, decision tree | The article introduces a new algorithm for predicting property values based on location, bedroom, and bathroom numbers, enhancing the accuracy of price prediction models, and enhancing home buying experiences. | Decision trees can overfit when deep or improperly pruned, capturing training data noise as a pattern, leading to poor generalization to unseen data. | Future work focuses on enhancing model accuracy and usefulness by considering various factors such as landscape and surroundings, such as residences, parking lots, and markets. |
| 2. | An Innovative Method to Predict Real Estate Prices using Convolutional Block Attention Module (2023) | Convolutional Block Attention Module | The proposal uses real estate market data from Ames, Iowa, from 2006 to 2010 to predict future home pricing trends and investment decisions. | CBAM, a predictive model, relies heavily on real-estate data, which can lead to inaccurate predictions due to limited or biased data. |  |
| 3. | Predict Condominium Prices in Bangkok Based on Ensemble Learning Algorithm with various factors (2023) | Random Forest Regressor, Gradient Boosting Regressor | The study utilized online scraping techniques to gather data on condominiums in Bangkok, incorporating features from various sources like POIs and macroeconomics. Tree-based models were used for analysis and prediction, with MAPE testing. | The predictive model's performance may be negatively affected by the lack of quality and availability of real estate data, particularly in Bangkok. | In the future they hope their model could be used to condominium valuation process in real life and can help appraisal to reduce cost, time, and human bias. |
| 4. | Real Estate Forecasting System Using Machine Learning Algorithms (2023) | Linear regression | The study uses machine learning algorithms to predict housing values, evaluating property market worth based on location, size, amenities, and features. It aims to optimize investments and design policies for affordable housing and sustainable development, benefiting stakeholders like buyers, sellers, investors, and policymakers. | Linear regression assumes a linear relationship between independent variables and dependent variables, but in real estate, factors influencing prices may not be strictly linear, leading to model inaccuracies. | The future of the housing market can be significantly improved using machine learning and predictive algorithms for property valuation and planning. |
| 5. | Real Estate Prediction System Using ML (2023) | NLP & CNN | NLP and CNN were utilized in a prediction model to predict the price of any real estate property, considering factors like location, building age, and flooring. | Real estate text data-based NLP and CNN models may not accurately generalize to unknown locations or property types, potentially leading to overfitting and poor performance on new datasets. | Improved predictions using SVM and KNN neural networks can enhance real-time price monitoring and prediction accuracy. |
| 6. | Using Machine Learning to Predict Housing Prices (2023) | Artificial Neural Network & Support Vector Regression | ANN is utilized in forecasting housing prices, revealing the impact of geography and structural features on property values, highlighting the need for precise models for investors and developers. | ANNs require tuning hyperparameters like layer number, neurons per layer, activation functions, and learning rates, which can be time-consuming and require extensive experimentation. | Future research should focus on integrating data sources for accurate predictions, exploring emerging technologies like blockchain and AI in real estate price forecasting, and studying environmental factors' influence on property prices. |
| 7. | Analysis of Even Pricing in Real Estate Markets: Different Asset Types and Implications (2023) | Statistical method | The study explores the impact of charm listing prices on property transaction prices, price digits, and negotiation results in real estate markets, examining clustering of prices around specific digits. | Selection bias in real estate transactions can affect the generalizability of findings, as properties sold may differ systematically from those not, potentially affecting the overall market representation. |  |
| 8. | A multi-level modelling approach for predicting real-estate dynamics (2023) | Elastic Net & Lasso regressor | The research paper explores the use of regression models and machine learning algorithms to predict house prices and inventory in the real estate market. | Elastic net regularization, a method that uses L1 and L2 penalties for variable selection and regularization, can be challenging to interpret, especially with numerous predictors. | Future work focuses on incorporating socioeconomic variables and deep learning-based models for real estate data forecasting, evaluating policy changes' impact on housing market dynamics. |
| 9. | Prediction of Real-Time Estate Pricing using Train Test Splitting Techniques (2023) | Random Forest Regression,  Linear Regression,  Decision Tree | The study uses machine learning techniques like Random Forest Regression to forecast real estate values, utilizing data acquisition, preprocessing, training, and evaluation methods. | Train-test splitting in real estate requires ample data, but limited data can lead to overfitting or poor performance, making it challenging to obtain large and diverse datasets. | Future study on real estate price prediction platforms is crucial due to advancements in technology and the availability of new data sources. |
| 10. | Productivity assessment of the real estate industry in China: a DEA-Malmquist index (2023) | DEA model | This study explores the development path of the real estate industry in 30 sample provinces in mainland China from 2007 to 2016, focusing on total factor productivity. | DEA productivity measures are influenced by scale efficiency, which measures optimal firm operations, but firm size and market position can potentially bias these assessments. | Future research will develop a comprehensive, multi-stage logical model to understand internal mechanisms and develop a more rational and high-quality real estate development plan. |
| 11. | Training a Neural Network to Predict House Rents Using Artificial Intelligence and Deep Learning (2023) | Deep Learning | The paper explores the use of neural networks to predict housing rents, highlighting the potential of AI in the real estate industry, based on features like square footage, floor, restaurant availability, and pets. | Identifying and selecting relevant house rent features is crucial for model performance, but feature engineering can be challenging due to heterogeneous data sources and complex variable relationships. | Research aims to enhance neural network models' accuracy and performance in rent prediction, addressing concerns about information asymmetry, employment displacement, and privacy risks. |

1. **PROJECT DESCRIPTION AND GOALS**

The project "RealEstate Predictor: AI-Driven Real-Time Property Valuation" takes a holistic approach to updating real estate appraisal procedures. The process starts with a thorough data collecting phase when a wide range of property parameters are painstakingly recorded, including BHK, area, bathrooms, balcony, square footage, and geographic location. By doing this, the dataset is guaranteed to be comprehensive and representative of a range of property kinds and market situations.

The gathered data is then subjected to stringent preprocessing methods to guarantee its accuracy and consistency. This entails doing things like standardising features, eliminating outliers, and managing missing information. Through the application of strong preprocessing techniques, the project lays a strong basis for the construction and assessment of the model that follows.

Then, a variety of machine learning models are utilised, utilising a range of methods such as SVM, Lasso, Ridge, XGBoost, Random Forest, and linear regression. Each model is painstakingly trained, and its efficacy in properly forecasting property prices is assessed by the application of relevant performance measures.

Techniques for cross-validation and hyperparameter optimisation are essential for optimising the models' performance. The study seeks to achieve a fine balance between resilience and accuracy by iterative testing and parameter optimisation, guaranteeing that the finished models generalise effectively to unobserved data.

The project's fundamental tenets are justice and transparency. Fairness-aware ML algorithms and explainability characteristics are incorporated into the models in order to accomplish this. By reducing biases and clarifying the valuation process, these strategies promote responsibility and trust in the system's output.

A key component of the project's methodology is its extensive documentation and reference. Through the use of cutting-edge research papers for insights and thorough documentation of the whole process, the initiative enhances its techniques and promotes cooperation within the real estate community.

The creation of an approachable Flask web application is the project's greatest accomplishment. With the help of this application, stakeholders may easily get real-time property valuation estimates, giving them access to insightful information that will help them trust the valuation process. The "RealEstate Predictor" initiative seeks to transform real estate decision-making through innovation and responsibility, ultimately benefiting stakeholders and the industry as a whole.

1. **TECHNICAL SPECIFICATION**
   1. Hardware Requirements

RAM: At least of 8 GBRAM is recommended, and more is better

Processor: A multi-core processor with a clock speed of 2 GHz or higher is

recommended.

Storage space: a minimum of 2 GB of free storage space should be sufficient.

* 1. Software Requirements

Editor: Jupiter Notebook

Language: Python

Libraries:

* NumPy: NumPy is a Python package that is used for scientific computing with Python.
* Pandas: Pandas are a Python package that is used for data manipulation and analysis.
* Matplotlib: Matplotlib is a plotting library for Python. It provides a wide range of 2D and 3D plots, charts, and graphs.
* Seaborn: Seaborn is a Python package for creating visually appealing and useful statistics graphs.
  1. Data Used

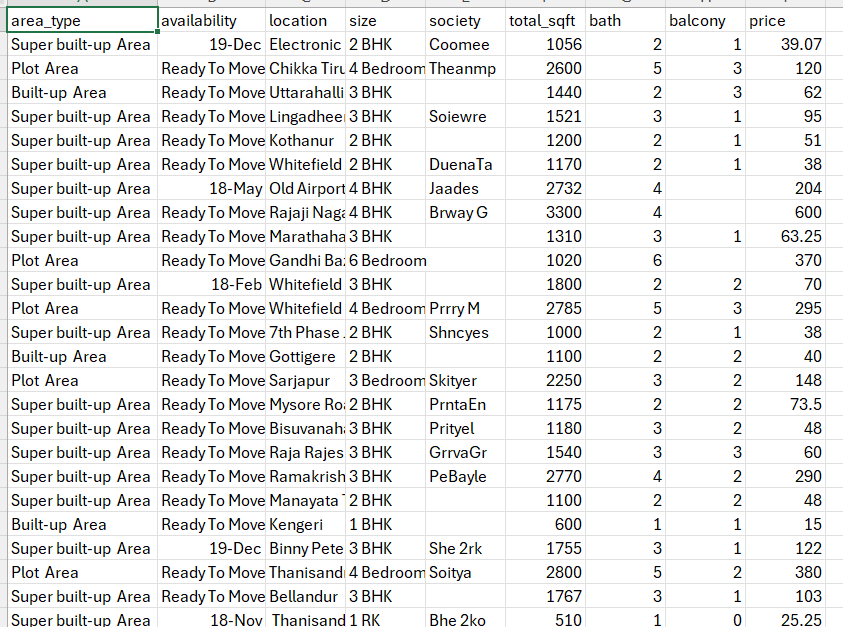


Figure 4.3.1 Sample image of Data Set

1. **DESIGN APPROACH AND DETAILS**
   1. **System Architecture**

A diagram of a computer flowchart

Description automatically generated

Figure 5.1.1 System Architecture

* 1. **Module description**
     1. **Data analysis:**

In this stage we need to check the data first and get detailed information about the data whether it is useful or not.

* + 1. **Data cleaning:**

In this stage we need to search for null values if u find any null values in any column with minimum % then drop that row from the table and if u got so many null values in the column, then u need to drop that column, if u don’t drop that column it leads to failure of our model. In my data society has 41.3% missing value so I have dropped that column.

* + 1. **Data preprocessing:**

In this stage we need to convert the data into integer type by that we can easily train and test our regression model.

* + 1. **Model training:**

In this stage I have used 6 models to compare the models which gives best accuracy. First, I have used linear regression in that I got an accuracy of 79% and RMSE value of round 65-70% so it is overfitting. Then I used lasso, ridge and SVM too they also not gave much convenient outcome, so I used random forest and XGBoost models then I got some good accuracy. I used the XGBoost model to train and test the model.

* + 1. **Deployment:**

In this stage I build a front-end model to showcase the output with the help of flask to combine the web application and the model.

1. **PROPOSED METHODOLOGY**
   1. **Multiple Linear Regression:**

This model builds links between the many variables being studied. Correlation coefficients or regression equations can be used to determine the correlation between variables. There are models that can identify the most important aspects for understanding the dependent variable. Multiple regression is used to collect data on both independent and dependent factors to anticipate certain prices. Multiple linear regression determines the link between dependent and independent variables.  
This model operates by treating housing price projections as independent variables and data such as home price, size, property type, and bedrooms as dependent factors. As a result, the algorithm's aim is the house price, also known as the dependent variable, and the influencing elements are allocated as independent variables.   
Thus, a correlation coefficient can be identified to determine the main variable.

A graph of mathematical equations

Description automatically generated

Figure 6.1.1 Linear regression Graph and Formula

* 1. **Lasso Regression (L1 Regularization):**

Lasso regression is a linear regression approach that combines variable selection and regularization. It introduces a penalty term into the loss function, which is the absolute value of the coefficients multiplied by a regularization parameter (alpha). This promotes sparsity in the coefficients, thereby doing variable selection by setting certain coefficients to zero.

A black square with numbers and symbols

Description automatically generated with medium confidence

Figure 6.1.2 Lasso Formula

* 1. **Ridge Regression (L2 Regularization):**

Ridge regression is another linear regression approach that regularizes the loss function by adding a penalty term equal to the square of the coefficients multiplied by a regularizations parameter (alpha). Ridge, unlike Lasso, does not pick variables, but rather lowers the coefficients to zero, diminishing their magnitude.

A black and white math equation

Description automatically generated

Figure 6.1.3 Ridge Formula

* 1. **Support vector machine (SVM):**

It is a supervised learning method designed for classification and regression problems. In SVM, the algorithm seeks the ideal hyperplane that best separates or fits the data points into distinct classes or predicts the target variable in regression, all while maximizing the margin between the classes or data points.In SVM regression, the aim is to create a function f(x) that predicts the target variable y based on input characteristics.

* **Hyperplane**: The hyperplane in SVM regression is defined as f(x)=⟨w, x⟩+b, where w represents the weight vector, x is the input feature vector, and b is the bias factor. The hyperplane aims to suit the data points as closely as possible while maximising the margin.
* **Regularization**: SVM regression employs a regularisation parameter C to manage the trade-off between maximising margin and minimising mistakes. A greater value of C provides for more freedom in fitting the data, but it may result in overfitting, whereas a lower value of C imposes a bigger margin, but it may cause underfitting.
* **Loss Function**: In SVM regression, the loss function seeks to minimise the errors between projected and actual values while maximising the margin between the hyperplane and the data points. To avoid overfitting, the loss function often includes a regularisation term.
* **Kernel Trick**: SVM regression can employ a variety of kernel functions (e.g., linear, polynomial, and radial basis functions) to translate input characteristics into a higher-dimensional space where the data may be more distinct. This enables SVM to identify complicated correlations between input characteristics and target variables.
  1. **Random forest:**

Random Forest regression is an ensemble learning approach that integrates the predictions of numerous decision trees to get a more accurate and stable result. During training, several decision trees are created from random subsets of the training data and features. Each tree is trained individually, and the final prediction is calculated by averaging the predictions of all individual trees (for regression tasks). This ensemble technique reduces overfitting and improves generalization performance. Furthermore, Random Forests incorporates unpredictability into the feature selection process, increasing the model's resilience and lowering the danger of overfitting.

* 1. **XGBoost:**

Extreme Gradient Boosting (XGBoost) is a well-known and strong machine learning technique that excels in regression and classification problems. It is part of the gradient boosting method family, which iteratively creates an ensemble of weak learners to minimize a given loss function.

* **Gradient Boosting Process**: XGBoost begins by fitting an initial decision tree to the data and then adds additional trees in a sequential order to rectify mistakes caused by earlier models. Each new tree is trained using the residuals.
* **Regularization**: XGBoost uses regularisation approaches to reduce overfitting and increase generalisation performance. It has parameters like max\_depth, min\_child\_weight, and gamma that govern the intricacy of the trees and keep them from growing too deep or splitting too frequently.
* **Loss Function Optimization**: XGBoost uses gradient descent to optimise a given loss function (for example, squared loss in regression). It computes the gradients of the loss function with respect to the model predictions and modifies the model parameters (tree structure and leaf scores) in the direction of least loss.
* **Parallel and Distributed Computing**: XGBoost is meant to be extremely scalable and efficient, with parallel and distributed computing options to handle huge datasets and expedite training. It uses techniques like approximation tree learning and histogram-based algorithms to boost computing performance.
* **Hyperparameter Tuning**: XGBoost has a variety of hyperparameters that may be adjusted to improve model performance, including the learning rate (eta), the number of trees (n\_estimators), and the subsampling ratio (subsample). To discover the best hyperparameter combination, approaches such as grid search or random search are commonly used.

1. **SCHEDULE, TASKS AND MILESTONES**

A graph on a white sheet

Description automatically generated

Figure 7.1 Gantt Chart Diagram

1. **PROJECT DEMONSTRATION**

* Exporting data to Jupiter notebook
* In the data there is a 3 variable with numerical values, so I described that it gives   
  df1.describe()  
  

Figure 8.1 Describing the Numerical Data

* Now we have taken corelation for that  
  A red and blue squares

  Description automatically generated

Figure 8.2 Correlation Chart

* Here we can find that bath and price have more corelation value.
* Data cleaning:

df.isnull().mean()\*

plt.figure(figsize=(16,9))

sns.heatmap(df.isnull())

df2 = df.drop('society', axis='columns')

df2.shape

df2['balcony'] = df2['balcony'].fillna(df2['balcony'].mean())

df2.isnull().sum()

df3 = df2.dropna()

df3.shape

df3.isnull(). sum()

A black and white bar code

Description automatically generated

Figure 8.3 Heat Map

* Finding outliers

def diagnostic\_plots(df, variable):

plt.figure(figsize=(16, 4))

plt.subplot(1, 3, 1)

sns.distplot(df[variable], bins=30)

plt.title('Histogram')

plt.subplot(1, 3, 2)

stats.probplot(df[variable], dist="norm", plot=plt)

plt.ylabel('Variable quantiles')

plt.subplot(1, 3, 3)

sns.boxplot(y=df[variable])

plt.title('Boxplot')

plt.show()

num\_var = ["bath","balcony","total\_sqft\_int","bhk","price"]

for var in num\_var:

print("\*\*\*\*\*\*\* {} \*\*\*\*\*\*\*".format(var))

diagnostic\_plots(df7, var)

df7[df7['total\_sqft\_int']/df7['bhk'] < 350].head()

A diagram of a graph

Description automatically generated with medium confidence

Figure 8.4 Before removing outliers.

* Removing outliers

def remove\_pps\_outliers(df):

df\_out = pd.DataFrame()

for key, subdf in df.groupby('location'):

m=np.mean(subdf.price\_per\_sqft)

st=np.std(subdf.price\_per\_sqft)

reduced\_df = subdf[(subdf.price\_per\_sqft>(m-st))&(subdf.price\_per\_sqft<=(m+st))]

df\_out = pd.concat([df\_out, reduced\_df], ignore\_index = True)

return df\_out

df9 = remove\_pps\_outliers(df8)

df9.shape

A diagram of a graph

Description automatically generated with medium confidence

Figure 8.5 After removing outliers.

* Converting all the data to encoded data  
  A screenshot of a computer

  Description automatically generated

Figure 8.6 Encoded Data

* Now splitting the data

X = df.drop("price", axis=1)

y = df['price']

print('Shape of X = ', X.shape)

print('Shape of y = ', y.shape)

* Model training.

from sklearn.linear\_model import LinearRegression

from sklearn.linear\_model import Lasso

from sklearn.linear\_model import Ridge

from sklearn.metrics import mean\_squared\_error

lr = LinearRegression()

lr\_lasso = Lasso()

lr\_ridge = Ridge()

def rmse(y\_test, y\_pred):

return np.sqrt(mean\_squared\_error(y\_test, y\_pred))

lr.fit(X\_train, y\_train)

lr\_score = lr.score(X\_test, y\_test)

lr\_rmse = rmse(y\_test, lr.predict(X\_test))

lr\_score, lr\_rmse

lr\_lasso.fit(X\_train, y\_train)

lr\_lasso\_score=lr\_lasso.score(X\_test, y\_test)

lr\_lasso\_rmse = rmse(y\_test, lr\_lasso.predict(X\_test))

lr\_lasso\_score, lr\_lasso\_rmse

lr\_ridge.fit(X\_train, y\_train)

lr\_ridge\_score = lr\_ridge.score(X\_test, y\_test)

lr\_ridge\_rmse = rmse(y\_test, lr\_ridge.predict(X\_test))

lr\_ridge\_score, lr\_ridge\_rmse

from sklearn.svm import SVR

svr = SVR()

svr.fit(X\_train,y\_train)

svr\_score=svr.score(X\_test,y\_test)

svr\_rmse = rmse(y\_test, svr.predict(X\_test))

svr\_score, svr\_rmse

svr = SVR(kernel='linear')

svr.fit(X\_train,y\_train)

svr\_score=svr.score(X\_test,y\_test)

svr\_rmse = rmse(y\_test, svr.predict(X\_test))

svr\_score, svr\_rmse

from sklearn.ensemble import RandomForestRegressor

rfr = RandomForestRegressor()

rfr.fit(X\_train,y\_train)

rfr\_score=rfr.score(X\_test,y\_test)

rfr\_rmse = rmse(y\_test, rfr.predict(X\_test))

rfr\_score, rfr\_rmse

import xgboost

xgb\_reg = xgboost.XGBRegressor()

xgb\_reg.fit(X\_train,y\_train)

xgb\_reg\_score=xgb\_reg.score(X\_test,y\_test)

xgb\_reg\_rmse = rmse(y\_test, xgb\_reg.predict(X\_test))

xgb\_reg\_score, xgb\_reg\_rmse

Table 8.1 Accuracy & RMSE values Comparison Before Tuning

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model | Score | RMSE |
| 0 | Linear Regression | 0.790384 | 64.898435 |
| 1 | Lasso | 0.803637 | 62.813242 |
| 2 | ridge | 0.790569 | 64.869802 |
| 3 | Support Vector Machine | 0.796263 | 63.981849 |
| 4 | Random Forest | 0.891346 | 46.72446 |
| 5 | XGBoost | 0.881127 | 48.872387 |

* Now I have tuned the xgboost.

from sklearn.model\_selection import GridSearchCV

from xgboost.sklearn import XGBRegressor

xgb1 = XGBRegressor()

parameters = {'learning\_rate': [0.1,0.03, 0.05, 0.07],

'min\_child\_weight': [1,3,5],

'max\_depth': [4, 6, 8],

'gamma':[0,0.1,0.001,0.2],

'subsample': [0.7,1,1.5],

'colsample\_bytree': [0.7,1,1.5],

'objective':['reg:linear'],

'n\_estimators': [100,300,500]}

xgb\_grid = GridSearchCV(xgb1,

parameters,

cv = 2,

n\_jobs = -1,

verbose=True)

xgb\_grid.fit(X\_train, y\_train)

print(xgb\_grid.best\_score\_)

print(xgb\_grid.best\_params\_)

* Tuning the XGBoost Model to Get the Best Parameters to increase the Accuracy. Then also it is Overfitting. So, I Changed Some if the Parameters to get best outcomes.

Table 8.2 Accuracy & RMSE values Comparison After Tuning

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model | Score | Rmse |
| 0 | Linear Regression | 0.790384 | 64.898435 |
| 1 | Lasso | 0.803637 | 62.813242 |
| 2 | ridge | 0.790569 | 64.869802 |
| 3 | Support Vector Machine | 0.796263 | 63.981849 |
| 4 | Random Forest | 0.891346 | 46.72446 |
| 5 | XGBoost | 0.900156 | 44.790243 |

* Testing the model

def predict\_house\_price(model,bath,balcony,total\_sqft\_int,bhk,area\_type,availability,location):

x =np.zeros(len(X.columns)) # create zero numpy array, len = 107 as input value for model

# adding feature's value accorind to their column index

x[0]=bath

x[1]=balcony

x[2]=total\_sqft\_int

x[3]=bhk

#x[4]=price\_per\_sqft

if "availability"=="Ready To Move":

x[8]=1

if 'area\_type'+area\_type in X.columns:

area\_type\_index = np.where(X.columns=="area\_type"+area\_type)[0][0]

x[area\_type\_index] =1

#print(area\_type\_index)

if 'location\_'+location in X.columns:

loc\_index = np.where(X.columns=="location\_"+location)[0][0]

x[loc\_index] =1

#print(loc\_index)

#print(x)

# feature scaling

x = sc.transform([x])[0] # give 2d np array for feature scaling and get 1d scaled np array

#print(x)

return model.predict([x])[0]

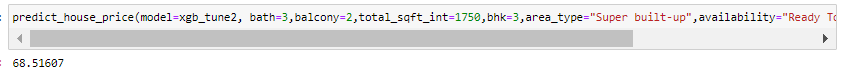


Figure 8.7 Model Output

* After building the model to showcase the result in beautiful way I used the web application to give output
* In that I have created a html page and app.py to work the model along with web application.
* **Html code**

<!DOCTYPE html>

<html>

<head>

<title>realestate property valuation App</title>

<link rel="stylesheet" href="cp.css">

<style>

@import url('https://fonts.googleapis.com/css2?family=Roboto&display=swap');

#heading-style{

font-family: "Roboto", sans-serif;

font-weight: 500;

font-style: normal;

}

#pagebox{

/\* border: 1px solid firebrick; \*/

height: 100vh;

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

#page-content-box{

/\* border: 2px solid teal; \*/

/\* margin: 20pxs; \*/

height: 95vh;

width: 90vw;

}

#top-box{

/\* border: 1px solid black; \*/

flex: 0.1;

max-height: 100%;

display: flex;

flex-direction: row;

justify-content: center;

align-items: center;

}

#heading-box{

/\* border: 1px solid hotpink; \*/

}

#middle-box{

/\* border: 1px solid red; \*/

flex: 0.6;

display: flex;

flex-direction: row;

}

#bottom-box{

flex: 0.3;

}

#left-box{

flex: 0.5;

border: none;

border-right: 2px solid black;

display: flex;

align-items: center;

justify-content: center;

}

#right-box{

border: none;

border-left: 2px solid black;

flex: 0.5;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

#form-heading

{

}

#form-box{

}

#input-style{

height: 35px;

border: none;

border-bottom: 1.5px solid black;

width: 250px;

}

/\* CSS \*/

.button-1 {

background-color: #EA4C89;

border-radius: 8px;

border-style: none;

box-sizing: border-box;

color: #FFFFFF;

cursor: pointer;

display: inline-block;

font-family: "Haas Grot Text R Web", "Helvetica Neue", Helvetica, Arial, sans-serif;

font-size: 14px;

font-weight: 500;

height: 40px;

line-height: 20px;

list-style: none;

margin: 0;

outline: none;

padding: 10px 16px;

position: relative;

text-align: center;

text-decoration: none;

transition: color 100ms;

vertical-align: baseline;

user-select: none;

-webkit-user-select: none;

touch-action: manipulation;

}

.button-1:hover,

.button-1:focus {

background-color: #F082AC;

}

</style>

</head>

<body style="margin: 0px">

<div id="pagebox" >

<div id="page-content-box">

<div id="top-box" >

<div id="heading-box">

<h1 id="heading-style" >Real Estate Property Valuation</h1>

</div>

</div>

<div id="middle-box" >

<div id="left-box">

<img src="https://cdn.dribbble.com/users/252645/screenshots/7417489/media/e4cfda317e90c49f6779c94d02e1c167.gif" alt="Computer man" style="width:500px;height:500px; ">

</div>

<div id="right-box">

<div id="form-heading" >

<h4 >Enter the Information of House to Predict the Price</h4>

</div>

<div id="form-box">

</div>

<form id="myForm" action="{{ url\_for('predict')}}"method="post">

<input id="input-style" type="number" name="bathrooms" placeholder="Bathrooms" required="required" width="48" height="10" step=".01"/><br>

<input id="input-style" type="number" name="balcony" placeholder="Balcony" required="required" width="48" height="10" step=".01"/><br>

<input id="input-style" type="number" name="total\_sqft\_int" placeholder="Total Squre Foot" required="required" width="48" height="10" step=".01"/><br>

<input id="input-style" type="number" name="bhk" placeholder="BHK" required="required" width="48" height="10" step=".01"/><br>

<input id="input-style" type="text" name="area\_type" placeholder="Area Type" required="required" /><br>

<input id="input-style" type="text" name="availability" placeholder="House Availability" required="required" /><br>

<input id="input-style" type="text" name="location" placeholder="House Location" required="required" />

<br>

<br>

<!-- Show button -->

<div class="button\_cont" align="center"><a class="button\_css" target="\_blank" rel="nofollow noopener">

<button class="button-1" type="submit" ><strong>Predict House Price</strong></button></a>

</div>

</form>

</div>

</div>

<div id="bottom-box">

<center>

<h2>{{ prediction\_text }}</h2>

</center>

</div>

</div>

</div>

</body>

</html>

* **App.py**

#Import Libraries

from flask import Flask, request, render\_template

import model # load model.py

app = Flask(\_\_name\_\_)

# render htmp page

@app.route('/')

def home():

return render\_template('index.html')

# get user input and the predict the output and return to user

@app.route('/predict',methods=['POST'])

def predict():

#take data from form and store in each feature

input\_features = [x for x in request.form.values()]

bath = input\_features[0]

balcony = input\_features[1]

total\_sqft\_int = input\_features[2]

bhk = input\_features[3]

# price\_per\_sqft = input\_features[4]

area\_type = input\_features[4]

availability = input\_features[5]

location = input\_features[6]

# predict the price of house by calling model.py

predicted\_price = model.predict\_house\_price(bath,balcony,total\_sqft\_int,bhk,area\_type,availability,location)

formatted\_price = "{:.2f}".format(predicted\_price)

# render the html page and show the output

return render\_template('index.html', prediction\_text='Predicted Price of Bangalore House is {}'.format(formatted\_price))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=False, port=5504)

* **Model.py**

#Import Libraries

import numpy as np

import pandas as pd

import joblib

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

#load data

df = pd.read\_csv("data/final\_data.csv")

# Split data

X= df.drop('price', axis=1)

y= df['price']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.2, random\_state=51)

# feature scaling

sc = StandardScaler()

sc.fit(X\_train)

X\_train = sc.transform(X\_train)

X\_test = sc.transform(X\_test)

###### Load Model

model = joblib.load('realestate\_property\_valuation\_model.pkl')

# it help to get predicted value of house by providing features value

def predict\_house\_price(bath,balcony,total\_sqft\_int,bhk,area\_type,availability,location):

x =np.zeros(len(X.columns)) # create zero numpy array, len = 107 as input value for model

# adding feature's value accorind to their column index

x[0]=bath

x[1]=balcony

x[2]=total\_sqft\_int

x[3]=bhk

# x[4]=price\_per\_sqft

if "availability"=="Ready To Move":

x[8]=1

if 'area\_type'+area\_type in X.columns:

area\_type\_index = np.where(X.columns=="area\_type"+area\_type)[0][0]

x[area\_type\_index] =1

if 'location\_'+location in X.columns:

loc\_index = np.where(X.columns=="location\_"+location)[0][0]

x[loc\_index] =1

# feature scaling

x = sc.transform([x])[0] # give 2d np array for feature scaling and get 1d scaled np array

return model.predict([x])[0] # return the predicted value by train XGBoost model.

* Working of web application

A computer screen shot of a house

Description automatically generated

Figure 8.8 Web Application Image

* Input values:

A computer screen shot of a house

Description automatically generated

Figure 8.9 input Image

* Output

A computer screen shot of a house

Description automatically generated

Figure 8.10 Output Image

1. **RESULT.**

The "Real Estate Predictor: AI-Driven Real-time Property Valuation" initiative produced outstanding findings, representing a significant step forward in the field of real estate valuation. Using cutting-edge AI and ML approaches, the research obtained significant results.

The project significantly improved its accuracy by combining regression models such as linear, Lasso, Ridge, SVM, Random Forest, and XGBoost. The final model, modified through hyperparameter optimisation, had an accuracy rate of almost 90%, outperforming standard valuation approaches.

The creation of a real-time property evaluation system allowed stakeholders to obtain valuation estimates quickly, reflecting the most recent market circumstances. This competence is critical for making sound judgements in a continuously changing real estate market.

The use of explainability characteristics and fairness-aware ML algorithms addressed issues about the opacity and prejudice that are commonly associated with traditional valuation approaches. This emphasis on openness and justice builds confidence among users while also ensuring equitable treatment across property kinds and locales.

1. **CONCLUSION AND FUTURE SCOPE.**

To summarise, the "Real Estate Predictor: AI-Driven Real-time Property Valuation" project is a huge step forward in the field of real estate assessment, providing a powerful combination of accuracy, responsiveness, openness, and fairness. The project has effectively solved significant issues inherent in traditional valuation methodologies by integrating cutting-edge AI and machine learning technology, as well as a dedication to innovation and honesty.

The project's outcomes show major accomplishments, such as significant increases in valuation accuracy, the creation of a real-time evaluation system, and the introduction of transparency and fairness measures. The project produced outstanding accuracy rates by combining a variety of regression approaches and fine-tuning models through hyperparameter optimisation, providing stakeholders with solid valuation estimations. The anticipated price is based on dataset information, including area, location, number of bedrooms, and bathrooms.

By incorporating environmental factors such as proximity to residential areas, parking lots, and marketplaces, we can significantly enhance the accuracy of the model. It's important to keep in mind that further improvements can make the model more useful, making it an asset for this project.

## References

1. Ishan Danona, Lekha Rani, Durgesh Srivastava, “Value Prediction of Real Estate Using Machine Learning” 2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0.
2. G N P V Babu, Dr. Vijayalakshmi N.S, Sameer Yadav, Dr. E. Manigandan, Chahat Gulati, Dr Aditya Kumar Singh Pundir, “An Innovative Method to Predict Real Estate Prices using Convolutional Block Attention Module” 2023 4th International Conference on Smart Electronics and Communication.
3. Thanit Anchaleechamaikorn, Taninnuch Lamjiak, Thagoon Thongpe, Lapis Thiralertpanit, Jumpol Polvichai, “Predict Condominium Prices in Bangkok Based on Ensemble Learning Algorithm with various factors” 2023 International Technical Conference on Circuits/Systems, Computers, and Communications.
4. Harshpreet Kaur, Kushwanth A, Sai Praveen M, Akshay Y, Surya Kartheek B, Ganesh Kamal Reddy G, “Real Estate Forecasting System Using Machine Learning Algorithms” 2023 6th International Conference on Contemporary Computing and Informatics (IC3I).
5. Abhinav Wandhe, Lakshya Sehgal, Hardik Sumra, Aryaman Choudhary, Mrunalee Dhone, “Real Estate Prediction System Using ML” 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP).
6. H. Prakash, K. Kanaujia and S. Juneja, "Using Machine Learning to Predict Housing Prices," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, 2023.
7. Orcun Morali, Neslihan Yilmaz, “Analysis of Even Pricing in Real Estate Markets: Different Asset Types and Implications” 2023.
8. Gude, V. (2024), "A multi-level modelling approach for predicting real-estate dynamics", International Journal of Housing Markets and Analysis.
9. A. Sharma and T. Poongodi, "Prediction of Real-Time Estate Pricing using Train-Test Splitting Techniques," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023.
10. Liu, X.-x., Liu, H.-h., Yang, G.-l. and Pan, J.-f. (2023), "Productivity assessment of the real estate industry in China: a DEA-Malmquist index", Engineering, Construction and Architectural Management.
11. C. -H. Chao, "Training a Neural Network to Predict House Rents Using Artificial Intelligence and Deep Learning," 2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA), Changchun, China, 2023.