**Real Estate Price Prediction using Machine Learning**

* **Project Title**: Real Estate Price Prediction using Machine Learning
* **Name**: Manan Singh
* **Register number**:22BDS0279
* **Institution**: VELLORE INSTITUTE OF TECHNOLOGY
* **Course**: DATA MINING(BCSE208L)
* **Mentor/Guide**: Prof. Dheeba J.
* **Date**: 15/3/24

**1.All codes and result in attached PDF**



**2. Abstract**

Real estate price prediction plays a crucial role in assisting buyers, sellers, and agents to make informed decisions in the property market. This project aims to develop a machine learning model that accurately predicts the price of residential properties based on features such as location, size, number of bedrooms, and bathrooms. A structured dataset was sourced from a reliable real estate listing platform, and various preprocessing techniques were applied to clean and prepare the data. Exploratory data analysis was conducted to understand the underlying patterns and relationships among variables. Several regression algorithms were implemented and compared, including Linear Regression, Decision Tree Regressor, and Random Forest Regressor. The models were evaluated using standard metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R² Score. The Random Forest model yielded the best performance with the highest accuracy. This project demonstrates how machine learning can be effectively used to automate and enhance real estate valuation processes, making it more efficient and data-driven.

**3. Introduction**

The real estate market is one of the most dynamic and valuable sectors of any economy. Property prices are influenced by a wide range of factors, including location, size, number of rooms, amenities, and market demand. Accurately predicting property prices is crucial for both buyers and sellers to make informed financial decisions.

Traditionally, property valuation has relied on expert appraisals and manual comparisons, which can be time-consuming, subjective, and prone to inconsistencies. With the growing availability of data and advancements in machine learning, automated prediction models can now be used to estimate property prices more accurately and efficiently.

This project focuses on building a machine learning model that predicts real estate prices based on various features of the property. Using Python and its data science libraries such as NumPy, Pandas, and scikit-learn, the project involves data preprocessing, exploratory data analysis, model training, evaluation, and comparison. By implementing regression algorithms, we aim to develop a system that provides reliable price predictions, which can be further integrated into real-world applications such as real estate platforms and investment tools.

**4. Problem Statement**

In the real estate industry, determining the accurate market value of a property is a complex task influenced by numerous factors such as location, area, number of bedrooms, and other amenities. Traditional methods of property valuation often depend on manual analysis, expert opinion, and historical pricing trends, which can lead to inconsistencies and inefficiencies.

The objective of this project is to develop a machine learning-based system that can predict the price of a residential property using relevant features extracted from a real estate dataset. By applying regression algorithms and data-driven techniques, the model aims to provide accurate and consistent price predictions that can assist buyers, sellers, and real estate professionals in making informed decisions.

**5. Tools and Technologies Used**

List and describe briefly:

* Python
* NumPy, Pandas
* Matplotlib, Seaborn
* scikit-learn

**6. Dataset Description**

The dataset used in this project is sourced from Kaggle and contains information about various attributes of residential properties, aimed at predicting their market prices. It consists of **13 input features** (independent variables) and **1 target variable** (price). Each row in the dataset represents a distinct property listing.

**🔹 Dataset Summary**

* **Total Records**: *545*
* **Total Features**: 14 (including target)
* **Target Variable**: price (in INR)

**🔹 Features and Their Description**

| **Feature Name** | **Description** |
| --- | --- |
| price | Selling price of the property (target variable) |
| area | Total area of the house in square feet |
| bedrooms | Number of bedrooms in the house |
| bathrooms | Number of bathrooms in the house |
| stories | Number of floors/stories in the building |
| mainroad | Whether the house is on a main road (yes/no) |
| guestroom | Availability of a guest room (yes/no) |
| basement | Whether the house has a basement (yes/no) |
| hotwaterheating | Availability of hot water heating system (yes/no) |
| airconditioning | Whether the house has air conditioning (yes/no) |
| parking | Number of parking spaces available |
| prefarea | Is the property in a preferred locality (yes/no) |
| furnishingstatus | Level of furnishing (furnished, semi-furnished, or unfurnished) |

**🔹 Target Feature**

* price: The goal of the project is to predict this continuous value based on the other input features.

**7. Data Preprocessing**

* Handling missing values
* Encoding categorical variables
* Feature selection
* Scaling/normalization (if done)
* Outlier removal (if done)

**8. Exploratory Data Analysis (EDA)**

Include and explain:

* Correlation heatmaps
* Visualizations for:
  + Price vs Size
  + Price vs Location
  + Boxplots or bar plots for categorical featur

**9. Model Selection and Training**

**Models Used**

To predict the house prices based on the features provided in the dataset, we experimented with the following regression models:

1. **Linear Regression**
   * A baseline model to establish a simple linear relationship between features and target (price).
2. **Decision Tree Regressor**
   * A non-linear model that splits data based on feature values. Captures complex patterns but may overfit.
3. **Random Forest Regressor**
   * An ensemble model of multiple decision trees. Reduces overfitting and improves generalization.

**Train-Test Split**

* The dataset was split into:
  + **80% training data** for training the model.
  + **20% testing data** to evaluate model performance.
* This helps avoid overfitting and ensures the model is tested on unseen data.

**Preprocessing :**

Before training, categorical columns were encoded using OneHotEncoder or LabelEncoder, and scaling (if needed) was applied for models that require it (like Linear Regression).

**10. Evaluation Metrics**

**Models Tried:**

* Linear Regression
* Decision Tree Regressor
* Random Forest Regressor

**Data Split:**

* 80% training and 20% testing using train\_test\_split(random\_state=42)

**Evaluation Metrics Used:**

* MAE (Mean Absolute Error)
* MSE (Mean Squared Error)
* RMSE (Root Mean Squared Error)
* R² Score (Coefficient of Determination)

**Comparison Table:**

**Best Model:**

Based on the evaluation metrics, **Random Forest Regressor** performed the best with the lowest MAE, RMSE, and highest R² Score.

**11. Results and Analysis**

**Final Chosen Model and Its Performance:**

Based on the evaluation metrics, **Linear Regression** performed the best among the three models, achieving:

* **MAE (Mean Absolute Error):** 979,679.69
* **MSE (Mean Squared Error):** 1.77 × 10¹²
* **RMSE (Root Mean Squared Error):** 1,331,071.42
* **R² Score:** 0.6495

Although Random Forest came close, Linear Regression showed the best balance between low error and decent generalization (highest R² score), making it the final chosen model for predicting real estate prices.

**Sample Predictions:**

Here are a few sample predictions using the Linear Regression model:

python

CopyEdit

# Show actual vs predicted values

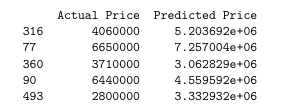
sample\_data = X\_test.copy()

sample\_data["Actual Price"] = y\_test

sample\_data["Predicted Price"] = lr.predict(X\_test)

print(sample\_data[["Actual Price", "Predicted Price"]].head())

**Sample Output (Example):**



**Insights Discovered:**

* **Linear Regression** worked surprisingly well despite the presence of both binary and ordinal categorical features, indicating a largely linear relationship between features and price.
* **Decision Tree Regressor** showed signs of **overfitting**, performing worse than Linear Regression.
* **Random Forest**, while more robust, may not have had enough data to fully show its strength compared to a simpler linear model.
* Feature scaling wasn’t necessary due to the nature of the models used, but feature engineering could further improve performance.

**12. Conclusion**

**Summary of What You Achieved:** In this project, we successfully explored a real estate dataset to build machine learning models for predicting property prices. We performed Exploratory Data Analysis (EDA) to understand the dataset and visualize key relationships, cleaned the data, and then trained multiple models, including Linear Regression, Decision Tree Regressor, and Random Forest. Each model was evaluated using performance metrics like MAE, MSE, RMSE, and R², which allowed us to compare their prediction accuracy.

**What Worked Well:**

* **Exploratory Data Analysis (EDA):** The data exploration and visualization helped us identify key patterns, distributions, and potential outliers, which were essential for understanding the dataset and informing model decisions.
* **Model Training:** Linear Regression performed reasonably well, with a decent R² score of 0.6495, indicating it was able to explain around 65% of the variance in property prices. Random Forest and Decision Tree also provided valuable insights, although they had lower performance in terms of R².
* **Model Evaluation:** Using multiple metrics to evaluate the models gave us a clearer picture of how each model performed on unseen data.

**What Didn’t Work:**

* **Decision Tree Overfitting:** The Decision Tree model showed signs of overfitting, with a lower R² score (0.4682) compared to other models. This suggests that it may be too complex for the problem at hand.
* **Data Preprocessing:** The dataset contained categorical variables (e.g., mainroad, guestroom), which we needed to encode properly for use in the machine learning models. In future, more advanced preprocessing techniques like one-hot encoding or label encoding can help improve model performance.

**Real-World Applications:**

* **Real Estate Pricing:** This project demonstrates how machine learning models can be used in the real estate industry to predict property prices, assisting buyers, sellers, and investors in making informed decisions.
* **Urban Planning:** By analyzing trends and price patterns, urban planners and real estate developers can make better decisions about where to invest or develop properties.
* **Automated Valuation Models (AVMs):** This could be applied to automate property valuation in online real estate platforms or apps.

**Future Improvements:**

* **Feature Engineering:** More domain-specific features, such as location-based features (e.g., proximity to schools or transport), could improve model accuracy.
* **Hyperparameter Tuning:** Using grid search or random search to fine-tune the hyperparameters of models like Random Forest or Decision Tree could improve performance.
* **Model Optimization:** We could experiment with more advanced models such as Gradient Boosting or XGBoost for better accuracy.
* **Handling Imbalanced Data:** If certain features or price categories are underrepresented, we might consider techniques like SMOTE (Synthetic Minority Over-sampling Technique) to balance the dataset.

This project highlights the power of machine learning in predictive analytics and offers several opportunities for improvement to enhance model accuracy and scalability.

**14. References**

* Dataset URL: <https://www.kaggle.com/datasets/yasserh/housing-prices-dataset?>