Delhi Flat Price Predictor

Project Report

Prepared by:
Harsh

Role:
Data Analyst

Abstract

The Delhi Flat Price Predictor project aims to develop a machine learning model capable of accurately predicting the prices of flats in Delhi based on various features such as location, area, furnishing status, and more. The primary objective is to assist potential buyers, sellers, and real estate agents in making informed decisions by providing reliable price estimates.

This report details the entire process, from data collection and preprocessing to model development and deployment. Utilising a comprehensive dataset of latest flat prices in Delhi, several predictive models were trained and evaluated. The best-performing model was selected based on its accuracy and reliability, and subsequently deployed as a web application using Flask and Render.

Key findings indicate that our model achieves high accuracy with minimal error, making it a valuable tool for the Delhi real estate market. Additionally, the project highlights the importance of data cleaning, feature engineering, and model tuning in achieving optimal results.

Future work will focus on expanding the dataset, incorporating additional features, and continuously refining the model to enhance its predictive capabilities.

Introduction

Background

Delhi, the capital city of India, has a dynamic and rapidly growing real estate market. The demand for residential properties in Delhi is influenced by various factors, including location, economic conditions, infrastructure development, and social trends. With the increasing complexity of the real estate landscape, there is a growing need for tools that can provide accurate and actionable insights into property pricing.

Motivation

Traditional methods of estimating property prices often rely on manual comparisons and subjective assessments, which can lead to inaccuracies and inconsistencies. The Delhi Flat Price Predictor project seeks to address these limitations by leveraging machine learning techniques to offer precise price predictions based on data and various influential factors. This approach aims to enhance decision-making for buyers, sellers, and real estate professionals.

Objectives

The primary objectives of the Delhi Flat Price Predictor project are:

- Data Collection: Gather comprehensive data on factors affecting flat prices in Delhi.
- Data Exploration and Cleaning: Ensure data quality through analysis and preprocessing.
- Exploratory Data Analysis (EDA): Analyse and visualise the data to uncover insights and patterns.
- Model Development: Train machine learning models to predict flat prices.
- Model Evaluation: Select the best-performing model based on accuracy.
- Deployment: Launch a web application for real-time price predictions.

Scope

The project focuses on developing a predictive model for Delhi's real estate market, covering data preprocessing, exploratory analysis, model training, and deployment via a web application.

Data Collection

Data Sources

The data for the Delhi Flat Price Predictor was sourced from MagicBricks, a popular real estate website that provides comprehensive listings of residential properties.

Data Types

The dataset includes the following types of information:

- Property Name: The name or identifier of the property.
- Property Title: Title or description of the property listing.
- Property Type: Type of property, such as Flat.
- Property Size: Carpet area of the property.
- Furnishing: Details about the furnishing status (e.g., furnished, semi-furnished, unfurnished).
- BHK: Number of bedrooms, halls, and kitchens in the property.
- City / Locality: The city or locality where the property is located.
- Price: Total price of the property.
- Price (SQFT): Price per square foot of the property.

Method

The data was collected using web scraping with the following steps:

Setup:
- Utilised Selenium and BeautifulSoup for data extraction.
- Configured the Safari driver to automate browser actions.

URL Construction:
- Created dynamic URLs based on city, BHK, property type, and budget ranges.

Data Extraction:
- Automated scrolling to load more data on the webpage.
- Parsed the page content to extract property details such as name, title, size, BHK, furnishing, and prices.

Data Storage:
- Compiled data into Pandas DataFrames.
- Saved the final dataset in a CSV file and HTML content for further reference.

Error Handling:
- Logged errors and missed URLs to track issues during data collection.

Data Exploration and Cleaning

Loading Data
To begin the analysis, loaded the dataset containing information about flat prices in Delhi using Python libraries such as Pandas and NumPy. The data was imported from a CSV file into a Pandas DataFrame for easier manipulation and analysis.

Initial Exploration
After loading the data, performed an initial exploration to understand the structure and contents of the dataset. Checked the total number of rows and reviewed the first few rows to get a sense of the data.

Data Cleaning
To ensure the quality and consistency of the data, several cleaning steps were performed:

Data Type Conversion
Converted the 'BHK' column to an integer type to ensure numerical operations could be performed accurately.

Property Size Standardisation
The 'Property Size' column contained various units such as 'sqft', 'qyrd', and 'sqm'. So, filtered the rows to keep only those with these units and then standardised the values to square feet for consistency.

Furnishing Status
Filtered the rows based on the 'Furnishing' column to include only those that specified 'Semi-Furnished', 'Unfurnished', or 'Furnished' statuses.

Price Cleaning and Conversion
The 'Price Total' column was renamed to 'Price (INR)' and cleaned to remove any non-numeric characters. The cleaned prices were then converted to numerical values in Indian Rupees (INR).

Calculating Price per Square Foot
Calculated the price per square foot and added this as a new column, rounding off the values for clarity.

Locality Extraction
Created a dictionary of localities and wrote a function to populate a new column 'City/Locality' based on the property titles. Also refined this column to include more specific regional information.

Saving the Cleaned Data
Finally, the cleaned dataset was saved to a new CSV file for further analysis.

Exploratory Data Analysis (EDA)

Overview
The primary goal of this EDA is to understand the structure, patterns, and anomalies within the cleaned dataset of Delhi flat prices. The analysis includes descriptive statistics, distribution visualisations, outlier detection and removal, and correlation analysis to uncover relationships between various features.

Data Loading and Initial Inspection
The dataset was loaded using pandas, and initial inspection was performed to understand its structure and content.

Feature Selection
To focus on the relevant features, the dataset was filtered to include key columns such as 'Property Type', 'City/Locality', 'BHK', 'Property Size (sqft)', 'Furnishing', 'Price (INR)', and 'Price (per sqft)'.

Descriptive Statistics
Descriptive statistics were generated to summarise the central tendency, dispersion, and shape of the dataset's numerical features.

Distribution of Categorical Variables
The frequency distribution of categorical variables was analysed to understand the distribution of properties across different localities, BHK configurations, and furnishing statuses.

Distribution Analysis
Box plots and histograms were used to visualise the distribution of key numerical features, identify outliers, and understand the spread of the data.

Property Size (sqft) And Price (INR)
The distribution of property sizes was visualised, and the Interquartile Range (IQR) was calculated to identify and remove outliers. Similarly, the distribution of property prices was analysed, and outliers were identified and removed using the IQR method.

Relationship Analysis
The relationship between property size and price was visualised using a scatter plot to identify any trends or patterns.

Furnishing Analysis
The distribution of properties based on their furnishing status was analysed using a bar plot.

Correlation Analysis
One-hot encoding was used to convert the 'Furnishing' categorical feature into numerical values. A correlation matrix was then created to identify relationships between numerical features.

Model Development

Overview

The goal of the model development phase was to build a predictive model for estimating the prices of flats in Delhi. This involved preprocessing the data, building and training a machine learning model, and evaluating its performance.

Data Preprocessing

Before building the model, the data was preprocessed to ensure it was in the appropriate format for training. The preprocessing steps included:

- Selecting Relevant Features: Only the relevant columns were selected from the dataset, including 'City/Locality', 'BHK', 'Property Size (sqft)', 'Furnishing', and 'Price (INR)'.
- Price Conversion: The 'Price (INR)' column was converted from INR to lakhs to simplify the values.
- One-Hot Encoding: Categorical variables ('City/Locality' and 'Furnishing') were converted into numerical values using one-hot encoding.
- Outlier Removal: Outliers in the 'Property Size (sqft)' and 'Price (INR)' columns were identified using the Interquartile Range (IQR) method and removed.

Model Building and Training

The processed data was then used to build and train an XGBoost regression model. The steps included:

- Separating Features and Target Variable: The dataset was split into features (X) and target variable (y).
- Splitting Data: The data was split into training and testing sets, with 80% of the data used for training and 20% for testing.
- Model Training: An XGBoost regression model was built and trained using the training data.
- Model Evaluation: The model's performance was evaluated using the R-squared score on the test data.
- Model Saving: The trained model and feature columns were saved for future use.

Model Evaluation

The performance of the XGBoost model was evaluated using the R-squared score, which was found to be 0.74. This indicates that the model explains 74% of the variance in the property prices, suggesting a good fit.

Model Deployment

Overview

The deployment phase involved creating a web service that allows users to interact with the trained XGBoost model for predicting flat prices in Delhi. This was achieved by developing a Flask application, which was then deployed on the Render platform. The deployed API can be accessed and used through the my portfolio website to predict property prices based on user inputs.

Components

The deployment process included three main files:

- app.py: The main application file containing the Flask app.
- xgboost_model.pkl: The trained XGBoost model.
- feature_columns.pkl: The feature columns used in the model.

Application Code

The Flask application provides endpoints for predicting house prices and retrieving available locations. The key functionalities include:

Loading the Model and Features:

- The trained model and feature columns are loaded using joblib.
- A list of available locations is extracted from the feature columns.

Preprocessing Function:

- The preprocess_data function prepares input data by selecting required columns and one-hot encoding categorical variables.

Endpoints:

- Home (/): A welcome message for the API.
- Locations (/locations): Returns a list of available locations for prediction.
- Predict (/predict): Accepts input data and returns the predicted house price.

Deployment:

- The application was deployed on the Render platform under the name “Project API.”
- The API can be accessed through the my portfolio website for real-time price predictions.

Summary

The model deployment phase successfully created a web service using Flask that hosts the flats price prediction model. This service was deployed on Render, enabling seamless integration with the my portfolio website. Users can now interact with the model through a web interface, inputting property details to receive predicted prices. This deployment enhances the accessibility and usability of the predictive model, allowing for real-time property price estimations.

Conclusion

The "Delhi Flat Price Predictor" project aimed to develop a robust and accurate model for predicting flat prices in Delhi based on various property attributes. This comprehensive project involved several key steps, from data collection and preprocessing to model development and deployment. The following points summarise the key accomplishments and learnings from each phase:

Data Exploration and Cleaning

I began by exploring the dataset to understand its structure and identify any data quality issues. Essential steps included handling missing values, detecting and removing outliers, and transforming variables to suitable formats. This rigorous cleaning process ensured the dataset was ready for accurate and reliable model training.

Exploratory Data Analysis (EDA)

Through detailed EDA, gained valuable insights into the distribution and relationships of various features within the dataset. Visualisations such as box plots, histograms, and scatter plots helped to understand the data better and informed our feature engineering decisions. The correlation analysis further identified significant relationships between features, guiding the model-building process.

Model Development

The XGBoost algorithm was chosen for its superior performance and scalability. After preprocessing the data and applying feature engineering techniques, trained the XGBoost model and achieved an R-squared score of 0.74. This indicated a strong predictive capability, capturing a substantial portion of the variance in flat prices based on the input features.

Model Deployment

Deploying the model involved creating a Flask web application, enabling users to interact with the model via a user-friendly API. The application was successfully hosted on the Render platform, making it accessible through the my portfolio website. This deployment phase demonstrated the practical applicability of the model in a real-world setting, allowing users to predict flat prices dynamically.

Conclusion

Overall, the "Delhi Flat Price Predictor" project was a success, showcasing the entire data analytics pipeline from data preparation to model deployment. Key learnings included the importance of thorough data cleaning, the value of detailed exploratory analysis, and the effectiveness of the XGBoost algorithm for regression tasks. The successful deployment of the model underscores its potential utility in the real estate market, providing valuable price predictions for potential buyers and sellers.

The project not only honed technical skills in data analysis, machine learning, and web development but also emphasised the importance of integrating these skills to create end-to-end solutions. The final product is a testament to the power of data-driven approaches in solving real-world problems, and it sets a solid foundation for future enhancements and applications in other domains.