AIS431: Intelligent Decision Support Systems

Predictive, Ranking, and Recommend Decision Support System for House Pricing

BY:

Aya Fouda 202002609

Nour Mohamed 202000119

Youssef Abbas 202001428

Abdulrahman Medhat 202002045

Mohamed Elsayed 202002092

Under The Supervision of:

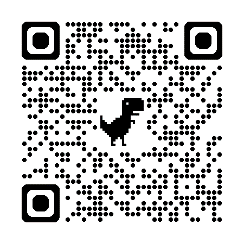
Dr. Inas Yassine

Eng. Kholoud Khaled

**Dataset**: <https://www.kaggle.com/datasets/juhibhojani/house-price>

**GitHub:** [**https://github.com/AyaFouda2002/IDSS\_FINAL\_PROJECT**](https://github.com/AyaFouda2002/IDSS_FINAL_PROJECT)

(Yarb 5 bouns 😊)

****

**Problem Definition**

A Predictive Decision Support System (PDSS) for house pricing in Intelligent Decision Support Systems (IDSS) leverages advanced analytics, machine learning algorithms, and vast datasets to provide accurate and dynamic pricing recommendations. By analyzing historical and real-time data on market trends, property features, location advantages, and economic indicators, this system offers precise valuation models. It aids stakeholders, including buyers, sellers, and real estate professionals, in making informed decisions by forecasting future market movements and evaluating property worth. The integration of PDSS in IDSS enhances decision-making efficiency, optimizes financial outcomes, and supports a transparent, data-driven real estate market.

**Objective**

* Develop a Predictive Decision Support System (PDSS) for house pricing within Intelligent Decision Support Systems (IDSS).
* Leverage advanced analytics, machine learning algorithms, and vast datasets.
* Provide accurate and dynamic pricing recommendations.
* Analyze historical and real-time data on market trends, property features, location advantages, and economic indicators.
* Offer precise valuation models for informed decision-making.
* Aid stakeholders in forecasting future market movements and evaluating property worth.
* Enhance decision-making efficiency, optimize financial outcomes, and support a transparent, data-driven real estate market.

**Introduction**

**Summary of Phase I:** In Phase I, we introduced the concept of a Predictive Decision Support System (IDSS) for house pricing, leveraging advanced analytics and machine learning algorithms to provide dynamic pricing recommendations. The system utilizes historical and real-time data on market trends, property features, location advantages, and economic indicators to offer precise valuation models. This IDSS aims to aid stakeholders, including buyers, sellers, and real estate professionals, in making informed decisions by forecasting future market movements and evaluating property worth. The integration of IDSS into Intelligent Decision Support Systems (IDSS) enhances decision-making efficiency, optimizes financial outcomes, and supports a transparent, data-driven real estate market.

**Purpose of Phase II:** Phase II focused on detailing the proposed design of the IDSS, providing a comprehensive literature review, examining current solutions and their limitations, and justifying our design decisions. We described the system architecture, outlined the functions of its components, and explained the methodologies applied in its development.

**Objective of the Final Phase:** This final report aims to present a detailed account of the completed Predictive and Ranking Decision Support System for house pricing. It covers the data preparation, model development, deployment, results, and insights. The report also highlights the practical implications and future enhancements for the system.

**Data Description**

1. House Attributes:

* + **Title**: The title or name of the listing.
  + **Description**: Detailed information about the house.
  + **Carpet Area**: The usable area within the house, excluding the thickness of the inner walls.
  + **Status**: The current status of the house (e.g., new, under renovation, ready to move in).
  + **Floor**: The level of the building on which the house is located.
  + **Transaction**: Type of transaction (e.g., resale, new booking).
  + **Furnishing**: The level of furnishing provided (e.g., furnished, semi-furnished, unfurnished).
  + **Facing**: The direction the house faces (e.g., north, south, east, west).
  + **Overlooking**: What the house overlooks (e.g., park, road, garden).
  + **Society**: The housing society or complex name.
  + **Bathroom**: Number of bathrooms.
  + **Balcony**: Number of balconies.
  + **Car Parking**: Information about car parking availability.
  + **Ownership**: The type of ownership (e.g., freehold, leasehold).

**2**. Financial Attributes:

* + **Amount (in rupees):** The price of the house.

3**. Location Attributes:**

* + Location: The location or city where the house is situated.

4**. Area Attributes:**

* + **Super Area**: The total area including balconies, proportionate share of common areas; sometimes includes wall thickness as well.

**Determining Agents**:

* + **Market Dynamics:** Demand and supply are influenced by economic conditions, population growth, and urbanization.
  + **Location:** The geographical position of a property, impacting its value based on connectivity, amenities, and neighborhood.
  + **Transaction Type**: Whether a property is a resale or new affects its pricing and market demand.
  + **Ownership Type**: Freehold or leasehold status, influencing transaction ease and property value.

**Association of Entities:**

* Transaction Type and Market Dynamics
* Property Features and Price
* Location and Price
* Society and Desirability

**Literature Review**

* **Advanced Analytics and Machine Learning in House Pricing**

The application of advanced analytics and machine learning in house pricing has seen significant growth, driven by the need for precise and dynamic pricing recommendations. Traditional models such as Multiple Linear Regression and ARIMA have been widely used in predicting house prices and forecasting future market movements. However, these models often fall short of capturing complex relationships and interactions between variables in the housing market.

* **Integration of Real-Time Data and Historical Data**

Combining real-time data acquisition with historical data enhances the accuracy and reliability of house price predictions. Real-time data sources include market trends, economic indicators, and property-specific attributes. Historical data provides a foundation for understanding long-term trends and patterns, allowing for more robust predictive models. The integration of these data sources helps in creating dynamic and responsive pricing models.

* **Feature Engineering and Data Preprocessing**

Feature engineering is a critical step in the development of predictive models. It involves creating new features that capture interactions between existing variables and standardizing numerical features to ensure they are on a comparable scale. Handling missing values is also crucial, as incomplete data can significantly impact model performance. Techniques such as filling missing values with the mode or dropping columns with high missing value counts are commonly employed.

* **Model Selection and Evaluation**

Various machine learning algorithms, including Linear Regression, Decision Trees, Random Forests, and Gradient Boosting, have been explored for house price prediction. Model performance is typically evaluated using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²). Random Forest has emerged as a robust model, offering high accuracy and the ability to handle non-linear relationships effectively.

* **User Interface and Practical Implementation**

The development of an interactive user interface is essential for making predictive models accessible to non-technical stakeholders. Tools like Gradio enable users to input property details and receive price predictions, enhancing the usability and practical application of the system. A user-friendly interface ensures that stakeholders, including buyers, sellers, and real estate professionals, can leverage the predictive capabilities of the system effectively.

* **Comparative Analysis with Existing Solutions**

Comparing the proposed system with existing solutions highlights its advantages and areas for improvement. For instance, while the IRJET system uses Multiple Linear Regression and ARIMA, the proposed system leverages Random Forest, which has shown better performance in handling complex relationships. The proposed system also integrates real-time data and provides a more interactive user experience through an advanced interface.

**Advantages and Disadvantages of the Literature Review**

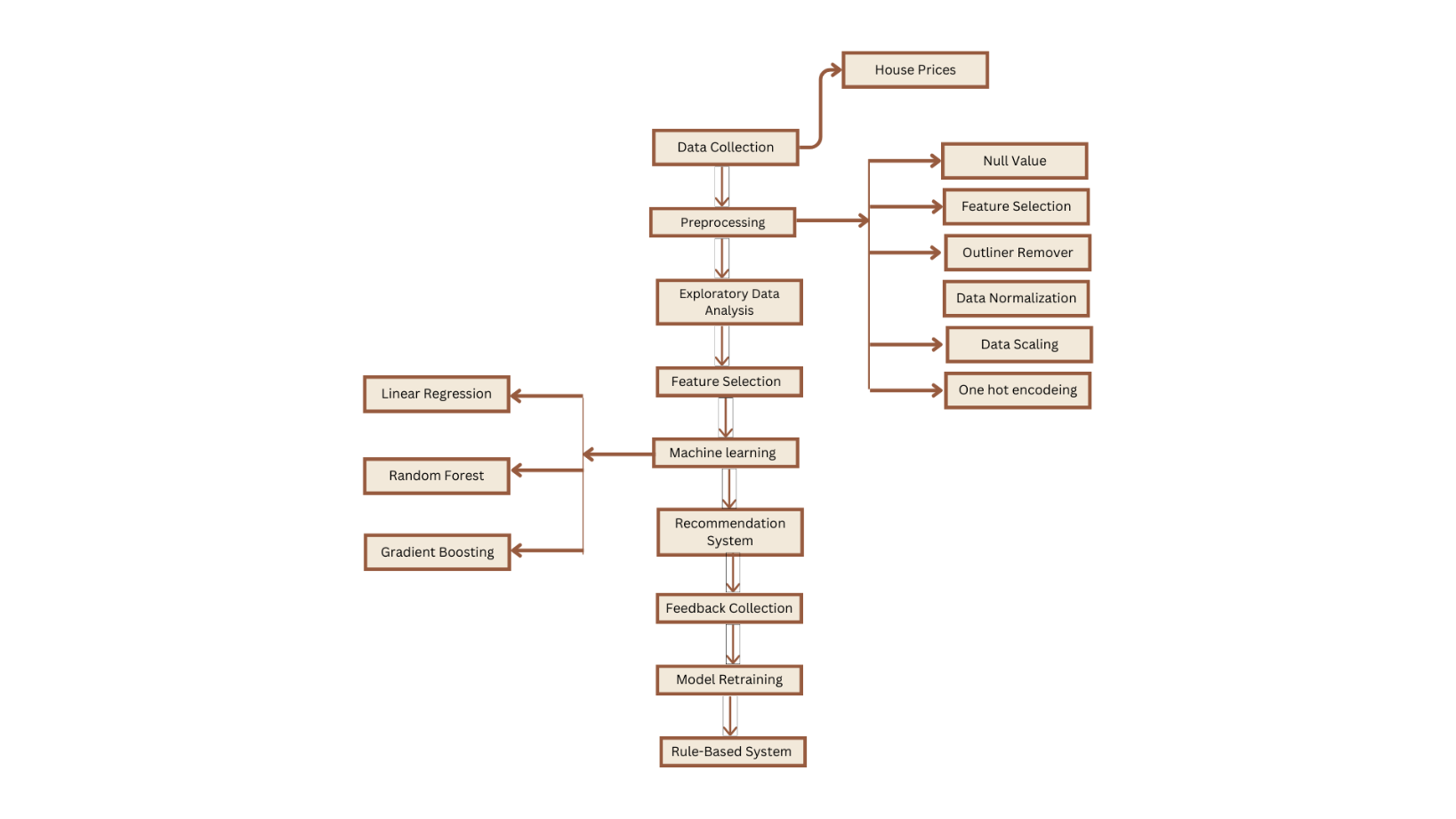
|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Comprehensive Data Integration | Complexity and Computational Resources |
| Enhances predictive accuracy and model reliability | - Requires significant computational resources |
| Real-time data acquisition for dynamic pricing | High complexity in model development and preprocessing |
| Advanced Analytics and Machine Learning | Handling Missing Data |
| Better performance with advanced algorithms | Dropping columns may lead to loss of valuable information |
| Effective feature engineering and preprocessing | Imputation methods can introduce biases |
| Improved Decision-Making | Scalability and Maintenance |
| Provides precise valuation models. | Resource-intensive to maintain real-time data integration. |
| Enhances decision-making efficiency. | Scalability issues with larger datasets |
| User-Friendly Interface | Dependency on Data Quality |
| Accessible to non-technical stakeholders | Accuracy relies on quality and completeness of data |
| Facilitates easy input and visualization | Inaccurate data leads to erroneous predictions |
| Comprehensive Model Evaluation | Limited Generalizability |
| Rigorous assessment with multiple metrics | Models may not generalize well to different markets |
| Highlights the system's strengths and improvements | Requires retraining for new contexts |
| Innovative Ranking System | User Adoption and Trust |
| Classifies properties as cheap, average, or expensive | Skepticism towards automated pricing recommendations |
| Adds value to user understanding of property value | Building trust requires consistent accuracy |

**Components of the Knowledge Base**

1. **Data Repository**:
   * **Historical Data**: Past house prices, economic indicators, and market trends.
   * **Real-Time Data**: Current market listings, economic conditions, and property-specific details.
   * **Demographic Data**: Population statistics, income levels, and other socio-economic factors.
   * **Geographic Data**: Location specifics, including neighborhood quality, accessibility, and amenities.
2. **Feature Repository**:
   * **Property Features**: Carpet area, number of bathrooms, balconies, parking space, furnishing status, facing direction, and property condition.
   * **Financial Features**: Listing price, transaction type (sale/rent), and ownership status.
   * **Location Features**: Proximity to landmarks, quality of local infrastructure, and neighborhood characteristics.
   * **Derived Features**: Interaction terms, normalized scales, and other engineered features.
3. **Model Repository**:
   * **Machine Learning Models**: Algorithms such as Random Forest, Gradient Boosting, Linear Regression, and Decision Trees.
   * **Evaluation Metrics**: Measures like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²) to assess model performance.
   * **Hyperparameters**: Optimal settings for model parameters to enhance predictive accuracy.
4. **Rule-Based Systems**:
   * **CLIPS (C Language Integrated Production System)**: For implementing rules that classify properties into categories like cheap, average, or expensive.
   * **Decision Rules**: Criteria for triggering specific actions or recommendations based on data inputs.
5. **User Interface and Interaction Data**:
   * **Interface Design**: Elements for user input, visualization of results, and interactive features.
   * **User Feedback**: Data collected from user interactions to refine models and improve system accuracy.
6. **Documentation and Knowledge Sharing**:
   * **System Documentation**: Detailed descriptions of the system architecture, data flows, and algorithm implementations.
   * **User Manuals**: Guides for users to interact with the system effectively.
   * **Knowledge Sharing Platforms**: Forums, wikis, and other collaborative tools to facilitate knowledge exchange among stakeholders.

#### **Building and Maintaining the Knowledge Base**

1. **Data Collection and Integration**:
   * Establish pipelines for continuous data acquisition from various sources.
   * Ensure data quality through validation, cleansing, and normalization processes.
2. **Model Development and Evaluation**:
   * Regularly train and test models using updated data.
   * Evaluate model performance and make adjustments based on evaluation metrics.
3. **Rule Creation and Management**:
   * Develop rules based on domain knowledge and expert input.
   * Update rules periodically to reflect changes in market conditions and user preferences.
4. **User Interface Development**:
   * Design intuitive and user-friendly interfaces.
   * Incorporate user feedback to improve functionality and usability.
5. **Documentation and Training**:
   * Maintain comprehensive and up-to-date documentation.
   * Provide training sessions and resources for users to maximize system utilization.
6. **Continuous Improvement**:
   * Monitor system performance and user satisfaction.
   * Implement continuous improvement processes to enhance the knowledge base and overall system effectiveness.



**Initial Data Exploration:**

* **Data Loading:** The dataset was loaded using Pandas.
* **Structure:** The dataset contains several columns with both numerical and categorical data.
* **Missing Values:** An initial check revealed significant missing values in certain columns.

**Data Preprocessing**

**Handling Missing Values:**

* Columns with a high number of missing values, such as 'Society', 'Car Parking', 'Super Area', 'Dimensions', and 'Plot Area', were dropped.
* Missing values in categorical columns like 'Description', 'Status', 'Furnishing', and 'Transaction' were filled with 'Unknown'.
* Numerical columns with missing values, such as 'Bathroom', were filled using the mode.

**Encoding Categorical Variables:**

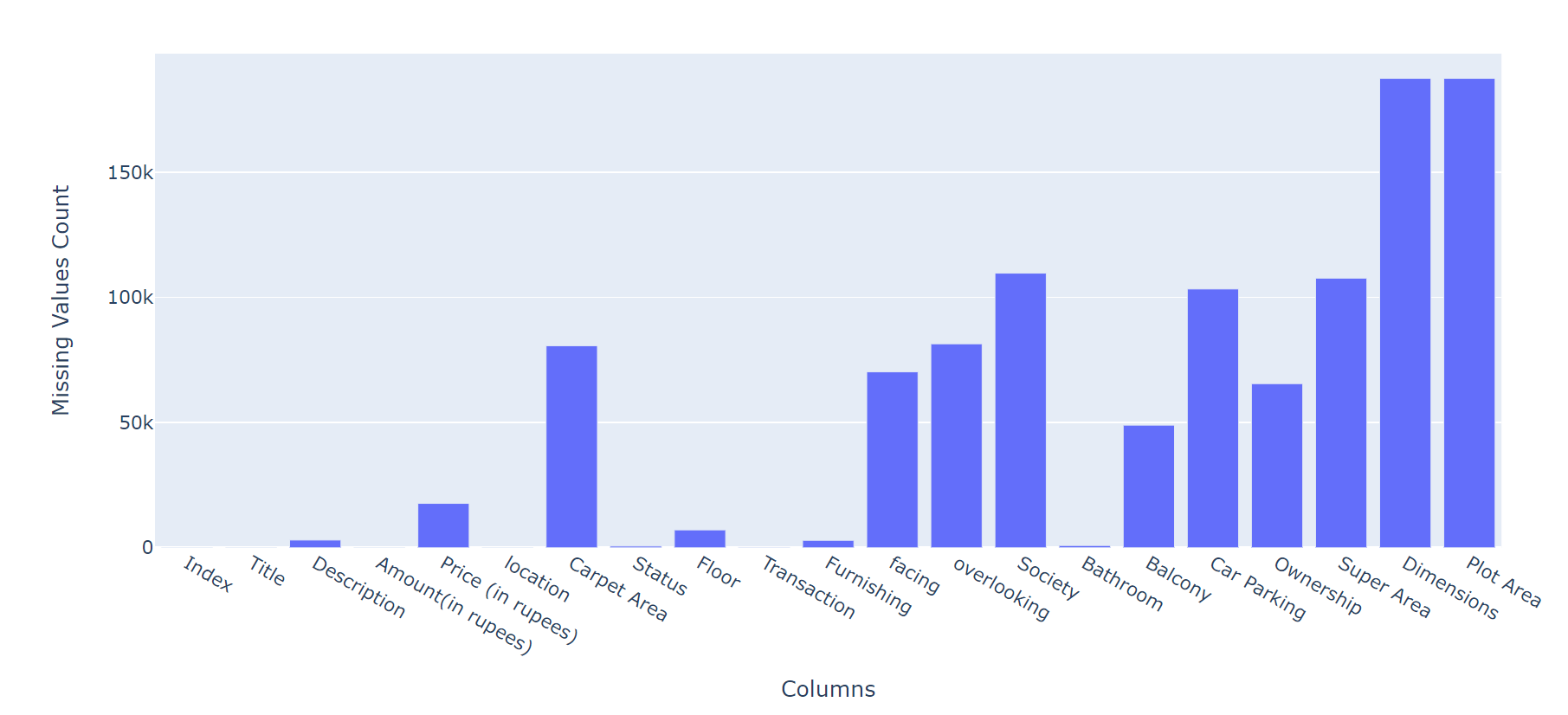
* Categorical features were encoded using label encoding and one-hot encoding as appropriate.
* Ensured that the encoding preserved the meaningful order of categorical values where necessary.

**Feature Engineering:**

* Created new features to capture interactions between existing features.
* Standardized numerical features to ensure they are on a comparable scale.

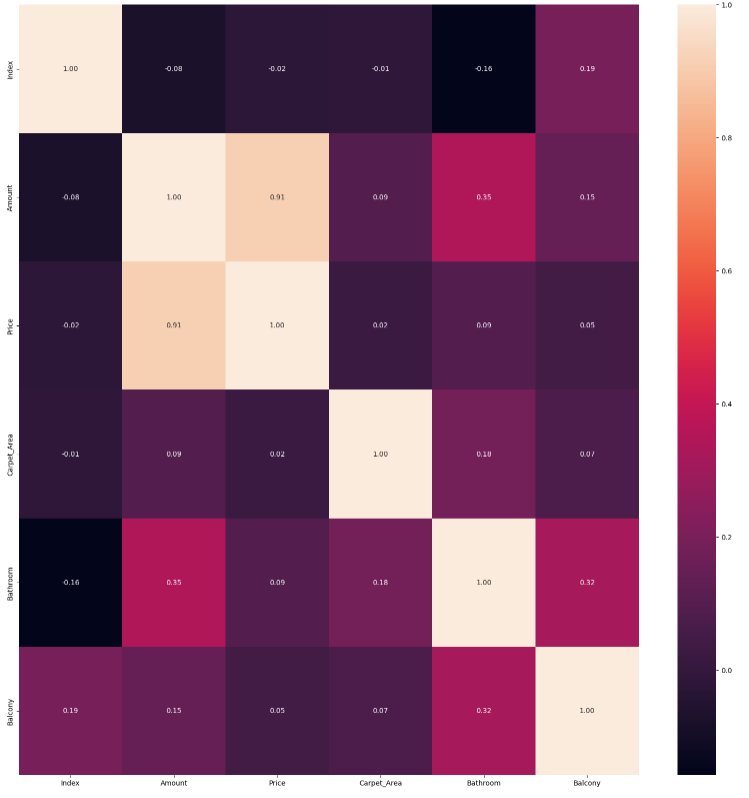
**Exploratory Data analysis (EDA)**

* Count of missing values in each column



This image is a bar chart that displays the count of missing values in each column of a dataset. The x-axis lists the columns in the dataset, while the y-axis shows the count of missing values. From the chart, it's evident that some columns have significantly more missing values than others. For example, the "Super Area," "Dimensions," and "Plot Area" columns each have over 150,000 missing values, making them the columns with the highest number of missing entries. Other columns like "location," "Carpet Area," "Society," and "Balcony" also show substantial counts of missing values, while some columns such as "Index," "Title," "Status," and "Floor" have relatively few or no missing values. This visualization helps identify which columns may require data cleaning or imputation efforts to handle the missing values effectively.

* Heatmap



This heatmap visualizes the correlation coefficients between different numerical variables in the dataset. Darker colors indicate stronger correlations, with the 'Amount' and 'Price' columns showing a high correlation of 0.91.

* Selecting the top 10 prices from the 'Price' column

A graph of a number of houses

Description automatically generated

This bar chart displays the top 10 highest house prices by city. Siliguri and Raipur have the highest prices, with Mumbai and Gurgaon showing the lowest prices among the top 10.

* Calculate the Max 'Price' for each 'location' and sort in descending order

A graph with blue and purple squares

Description automatically generated with medium confidence

* Calculate the value counts for each unique value in the 'location' column

This bar chart illustrates the top 10 locations by maximum house prices, with the x-axis representing different cities and the y-axis showing the maximum house prices in millions. Siliguri has the highest maximum house price at 4 million, followed by Raipur at around 3.5 million. Other locations such as Varanasi, New Delhi, Mumbai, Gurgaon, Goa, Chennai, Ahmedabad, and Kanpur have significantly lower maximum prices, all below 1 million. The color gradient, ranging from blue for lower prices to yellow for higher prices, emphasizes the stark contrast in maximum house prices between Siliguri, Raipur, and the other cities.

A graph of different colored squares

Description automatically generated

* Sunburst chart for the distribution of Furnishing

This bar chart displays the top 10 locations by the number of occurrences, with the x-axis representing the locations and the y-axis showing the count. New Delhi leads with 16,327 occurrences, followed by Gurgaon with 10,269. Other notable locations include Bangalore (4,357), Ahmedabad (4,294), Hyderabad (3,537), Chennai (3,015), Jaipur (2,818), Kolkata (2,663), Faridabad (2,563), and Greater Noida (2,211). Each bar is color-coded for easy distinction. The chart highlights New Delhi and Gurgaon as having significantly higher counts compared to other cities, emphasizing their prominence in the dataset.

A circular chart with text on it

Description automatically generated

This pie chart illustrates the distribution of house furnishings among three categories: Semi-Furnished, Unfurnished, and Furnished. The largest segment is Semi-Furnished, indicating that this category has the highest proportion among the three. The Unfurnished segment is slightly smaller but still substantial, while the Furnished segment is the smallest, representing the least common furnishing type in the dataset. The chart provides a clear visual representation of how houses are furnished, with a dominant preference for semi-furnished options, followed by unfurnished, and then fully furnished homes.

* Bathroom counts

A graph with purple and white bars

Description automatically generated with medium confidence

This bar chart depicts the frequency distribution of the number of bathrooms in a dataset. The x-axis represents the number of bathrooms, while the y-axis shows the frequency of each category. The chart reveals that the most common configurations are houses with 2 bathrooms (26,604) and 3 bathrooms (23,024), followed by houses with 4 bathrooms (9,407). Houses with 1 bathroom are less common (3,332), and those with more than 4 bathrooms are relatively rare, with only a few occurrences for each additional bathroom count up to 10. The color gradient, ranging from blue to yellow, visually distinguishes the frequency of each category, with higher frequencies shown in purple and lower frequencies in blue. This chart highlights that houses with 2 and 3 bathrooms are the predominant configurations in the dataset.

* correlation matrix

A graph with a arrow pointing up

Description automatically generated with medium confidence

This image is a correlation heatmap that visualizes the relationships between various features in a dataset. The matrix shows correlation coefficients, ranging from -1 to 1, between pairs of features. Darker shades of blue indicate stronger positive correlations, while lighter shades indicate weaker correlations. The diagonal line of dark blue squares shows that each feature is perfectly correlated with itself (correlation coefficient of 1).

Key observations:

- "Amount" and "Price" have a strong positive correlation, indicated by a dark blue square.

- Other features, such as "Bathroom" and "Balcony," show moderate correlations with certain features.

- Many features show weak or no significant correlation, as represented by the lighter blue squares.

This heatmap helps identify which features are closely related and which are relatively independent, aiding in feature selection and understanding relationships within the data.

**Model Development**

**Model Training:**

* **Data Splitting:** The dataset was split into training and testing sets (80% training, 20% testing).
* **Algorithms Considered:** We experimented with various regression models including Linear Regression, Decision Trees, Random Forests, and Gradient Boosting.
* **Training Process:** Each model was trained using the training data, with hyperparameter tuning performed to optimize performance.

**Model Evaluation:**

* **Metrics:** Models were evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R²).
* **Results:**
  + **Linear Regression:**
    - **R² Score:** 0.80
    - **MAE:** 3299.30
    - **RMSE:** 11132.77
    - **Predicted Prices:**
      * Index 176476: 3200.0, Predicted: 3292.709338
      * Index 132463: 10500.0, Predicted: 9691.401661
      * Index 154900: 3500.0, Predicted: 3514.514544
      * Index 94563: 6189.0, Predicted: 4891.425164
      * Index 66262: 7143.0, Predicted: 1390.753704
  + **Random Forest:**
    - **R² Score:** 0.97
    - **MAE:** 303.52
    - **RMSE:** 4283.31
    - **Predicted Prices:**
      * Index 176476: 3200.0, Predicted: 3200.94
      * Index 132463: 10500.0, Predicted: 10500.00
      * Index 154900: 3500.0, Predicted: 3630.98
      * Index 94563: 6189.0, Predicted: 6189.00
      * Index 66262: 7143.0, Predicted: 7143.00
  + **Gradient Boosting:**
    - **R² Score:** 0.85
    - **MAE:** 736.50
    - **RMSE:** 9526.50
    - **Predicted Prices:**
      * Index 176476: 3200.0, Predicted: 3496.296058
      * Index 132463: 10500.0, Predicted: 10279.201788
      * Index 154900: 3500.0, Predicted: 3608.345850
      * Index 94563: 6189.0, Predicted: 6112.771667
      * Index 66262: 7143.0, Predicted: 6313.572536

**Final Model Choice:**



* The best-performing model is the RandomForestRegressor with an R² Score of 0.97, MAE of 303.52, and RMSE of 4283.31.

A computer screen shot of a computer screen

Description automatically generated

This image displays the output of a house price prediction model, including user inputs and the model's recommended house price. The user provided various details: location number (80), carpet area (12,916.25546 square feet), furnishing number (1), facing number (0), overlooking number (0), number of bathrooms (3), number of balconies (2), and ownership number (1). The model uses these inputs, considering a total of 8 features both in the input data and the model. The recommended house price is 4,636.571833333333, based on these inputs.

Additionally, the output includes a generated title and description for the house: "Beautiful House with 3 Bathrooms." The description elaborates that the house, perfect for a family, features 3 bathrooms and 2 balconies, enhancing the living experience by allowing the enjoyment of views. The carpet area is 12,916.25546 square feet, and the house is furnished to level 1.

This output demonstrates the capability of the model to not only predict house prices based on specific inputs but also to generate descriptive details that provide a comprehensive overview of the property's features, making it easier for potential buyers or stakeholders to understand the offering.

A screenshot of a computer

Description automatically generated

* Being executed to predict house prices and classify properties based on user inputs. The user provides details about properties, such as location, carpet area, furnishing, facing direction, overlooking feature, number of bathrooms, number of balconies, and ownership type. The system then predicts the house price and classifies the property as "Expensive." Users give feedback on the predictions, with responses like "very bad" or "yes." The system records this feedback and offers to retrain the model using the new data. This iterative process allows the model to improve its accuracy over time, making the script a dynamic Decision Support System (DSS) that adapts based on user interactions.

A graph of a graph showing a long line

Description automatically generated with medium confidence

The image is a forecast plot generated by the Prophet model, illustrating the projected property prices from March 2024 to June 2025. The central blue line represents the median forecasted prices, showing a consistent upward trend, indicating an expected increase in property prices over time. The blue shaded area around the central line depicts the 95% confidence interval, suggesting the range within which the actual prices are likely to fall with 95% certainty, with the interval widening over time to reflect increasing uncertainty. The black dots represent historical property prices, aligning with their respective dates, which were used to train the model. This plot provides a clear visual of both the predicted price trend and the associated uncertainty over the forecast period.

A screenshot of a computer

Description automatically generated

The image shows the output of the Python script that searches for properties within a specified price range in a dataset. The user inputs a minimum price of 1560 and a maximum price of 12356. The script then filters the dataset to find properties with prices within this range and displays the results. The resulting data frame includes columns such as Amount, Price, location, Carpet\_Area, Furnishing, facing, overlooking, Bathroom, Balcony, and Ownership. Each row represents a property that meets the price criteria, with encoded categorical values for columns like location, Furnishing, facing, overlooking, and Ownership. The output successfully demonstrates the functionality of the script, showing a list of properties that fall within the specified price range

**Deployment**

**User Interface Development:**

* We developed an interactive user interface using Gradio, allowing users to input property details and receive price predictions.
* **Interface Features:**
  + Users can enter details such as location, carpet area, furnishing, facing, overlooking, number of bathrooms, balconies, and ownership type.
  + The interface outputs the predicted house price along with additional information such as the title, description, and amount from the dataset.

A screenshot of a computer

Description automatically generated

The provided Python code includes two functions, appartment\_ranking and appartment\_classify, which interface with CLIPS to classify properties. The appartment\_ranking function sends the price and carpet area of an apartment to CLIPS, which uses a set of rules defined in the ranking.clp file to classify the property. The classification result is extracted from the CLIPS output. Similarly, the appartment\_classify function sends the number of balconies and bathrooms to CLIPS, using rules defined in the apartment.clp file to classify the apartment. Both functions construct a command to run CLIPS, prepare the input as CLIPS commands, execute the command, and process the output to find and return the classification result. The code demonstrates the integration of Python with CLIPS to leverage rule-based classification in an expert system context.

The provided code leverages two CLIPS files, apartment.clp and ranking.clp, to classify apartments based on specific attributes. The apartment.clp file likely contains rules related to the characteristics of an apartment, such as the number of balconies and bathrooms, which are used to determine its classification. For instance, rules might specify that an apartment with more than two balconies and bathrooms qualifies as a "Luxury" apartment, whereas fewer balconies and bathrooms might classify it as "Standard" or "Economy." The Python function appartment\_classify prepares and sends this data to CLIPS, which processes it according to the rules in apartment.clp, and then returns the classification result. This approach allows for complex, rule-based decision-making processes to be managed outside of the main Python application, making the system more modular and easier to maintain.

A black background with white text

Description automatically generated

Similarly, the ranking.clp file contains rules for ranking apartments based on price and carpet area. This file likely defines thresholds or conditions under which apartments are categorized as "High-end," "Mid-range," or "Low-end" based on their price and size. The Python function appartment\_ranking assembles these inputs, sends them to CLIPS, and retrieves the resulting classification. By externalizing these rules in CLIPS files, the system can be easily updated with new criteria or adjusted to different market conditions without altering the core Python code. This separation of concerns enhances the flexibility and scalability of the application, allowing developers to fine-tune the classification logic or adapt it to new contexts by simply modifying the CLIPS rule files. This setup demonstrates the effective use of rule-based systems for complex decision-making tasks, where rules can be adjusted independently of the application logic.

A black screen with white text

Description automatically generated

**Results and Discussion**

**Model Performance:**

* **Random Forest Results:**
  + **R² Score:** 0.97
  + **MAE:** 303.52
  + **RMSE:** 4283.31
* The model demonstrated high accuracy and robustness in predicting house prices.

**Key Insights:**

* **Feature Importance:** Analysis of feature importance revealed that location, carpet area, and transaction type are among the most significant predictors of house prices.
* **Market Dynamics:** The model effectively captures market dynamics and provides valuable insights for decision-making.

**Impact of Features:**

* **Location:** The geographical position significantly influences house prices due to factors like connectivity, amenities, and neighborhood quality.
* **Property Features:** Characteristics such as carpet area, furnishing, and number of bathrooms greatly affect property value.

**Ranking System:**

* **Ranking Properties:** We implemented a ranking system using CLIPS to classify properties as cheap, average, or expensive based on price and carpet area.
  + **Cheap:** Properties with prices and carpet areas below the average.
  + **Average:** Properties with prices and carpet areas around the average but within the defined limits.
  + **Expensive:** Properties with prices and carpet areas significantly above the average.

**Conclusion**

**Summary:**

* The developed Predictive Decision Support System successfully provides accurate and dynamic house price predictions using advanced machine learning techniques.
* The system enhances decision-making for buyers, sellers, and real estate professionals by offering data-driven insights.
* Additionally, the implementation of a ranking system using CLIPS provides further classification of properties into categories of cheap, average, or expensive based on price and carpet area, aiding in more nuanced decision-making.

**Future Work:**

* Integrating additional data sources such as demographic and socio-economic data to further improve predictive accuracy.
* Continuously refining the model based on user feedback and evolving market conditions.
* Expanding the ranking system to include more attributes and refine classification criteria for better accuracy and user guidance.

**Comparison with Existing Solutions**

**Comparison with IRJET Paper:** The paper "House Price Prediction Forecasting and Recommendation System Using Machine Learning" by Ashutosh Sharma et al. focuses on using multiple linear regression for house price prediction and ARIMA for forecasting future prices. Their system also includes a content-based recommendation system to suggest properties based on user preferences. Key differences and improvements in our system include:

* **Model Selection:** While the IRJET paper uses Multiple Linear Regression and ARIMA, our system leverages Random Forest, which has shown better performance in handling non-linear relationships and interactions between variables.
* **Data Sources:** Our system integrates a wider range of data sources, including real-time data acquisition through APIs, enhancing the accuracy and reliability of predictions.
* **User Interface:** The interactive Gradio interface in our system provides a more user-friendly experience, allowing for easier manipulation of input data and visualization of results.
* **Evaluation Metrics:** We employ comprehensive evaluation metrics (MAE, MSE, R²) to rigorously assess model performance, whereas the IRJET paper primarily reports accuracy percentages.
* **Ranking System:** Our system includes a ranking mechanism using CLIPS to classify properties as cheap, average, or expensive, which is not present in the IRJET system.

**Pros and Cons of Existing Solutions:**

* **IRJET System:**
  + **Pros:** Simple implementation, effective use of linear regression and ARIMA for prediction and forecasting, incorporation of a recommendation system.
  + **Cons:** Limited to linear regression which may not capture complex relationships, reliance on a smaller set of features, and potential for overfitting with ARIMA in long-term forecasting.
* **Our Proposed System:**
  + **Pros:** Utilizes advanced machine learning models (Random Forest) for better performance, comprehensive data preprocessing and feature engineering, real-time data integration, and an interactive user interface. Includes a ranking system for classifying properties.
  + **Cons:** Higher computational complexity requires extensive data preprocessing and feature engineering, and potential challenges in maintaining real-time data integration.

**Advantages of Our Solution:**

* **Accuracy:** Improved predictive accuracy due to the use of Random Forest and comprehensive data preprocessing.
* **Flexibility:** The system is designed to be dynamically updated with new data, ensuring it adapts to changing market conditions.
* **User Experience:** The interactive interface enhances usability, making it accessible for a wider range of users, including non-technical stakeholders.
* **Insights:** Provides deeper insights into market dynamics and feature importance, aiding more informed decision-making.
* **Classification:** The additional ranking system helps users understand the relative value of properties based on price and carpet area.

**References**

* Park, B., & Bae, J. K. (2015). Predicting housing prices using machine learning techniques: Fairfax County, Virginia data analysis. Expert Systems with Applications, 42(6), 2928-2934.
* Hastie, T., Friedman, J., & Tibshirani, R. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.
* Natekin, A., and Knoll, A. (2013). Gradient boosting machines: a tutorial. Frontiers in Neurorobotics, 7, 21.
* Ashutosh Sharma, Pranav Sonawale, Deeksha Ghonasgi, Shreya Patankar (2020). House Price Prediction Forecasting and Recommendation System Using Machine Learning. International Research Journal of Engineering and Technology (IRJET), Volume 07, Issue 05.