## **Phase-3 Submission Template – Data Analytics**

**Student Name:** VIDHYA.S  
 **Register Number:** 511923106032  
 **Institution:** PRIYADARSHINI ENGINEERING COLLEGE  
 **Department:** ELECTRONICS AND COMMUNICATION ENGINEERING  
 **Date of Submission:** 10/05/2025  
 **GitHub Repository Link:**

**1.Problem Statement**

In the real estate sector, determining the right price for a house is a complex and critical task. Prices can vary significantly based on multiple factors such as the location, size (area in square feet), number of bedrooms (BHK), availability of amenities, and other structural features. However, without data-driven insights, these prices are often estimated manually, leading to inaccurate pricing, undervaluation, or overpricing. This not only impacts customer trust but also affects the profitability and efficiency of real estate businesses.

To solve this, the House Price Prediction project is formulated as a **supervised regression problem**, where the goal is to predict the continuous numerical target—Price—based on independent variables. The analytical approach combines **descriptive analysis** to understand the dataset, **diagnostic analysis** to explore feature relationships, and **predictive modeling** using machine learning techniques. Accurate price predictions can help property developers, agents, and buyers make informed decisions, reduce negotiation time, and enhance transparency in the housing market.

**2. Abstract**

This project aims to design a robust machine learning model to predict house prices using a structured dataset consisting of features like location, area, number of bedrooms and bathrooms. The dataset was cleaned and preprocessed to handle missing values, encode categorical variables, and standardize numerical values. Exploratory Data Analysis (EDA) revealed key patterns in the data, including the strong influence of area and location on pricing. A Random Forest Regressor was implemented and fine-tuned, achieving a high R² score and low mean absolute error, demonstrating its effectiveness in predicting house prices.

The findings of this project provide valuable insights into real estate price dynamics and help realtors and developers understand what factors contribute most to property value. Furthermore, the model can be integrated into sales platforms to assist users in estimating house prices accurately before listing or purchasing. This enhances trust and efficiency in the housing market, making the solution beneficial for both consumers and professionals in the real estate industry.

**3. System Requirements**

To implement and execute the House Price Prediction project efficiently, certain **hardware and software prerequisites** must be met. These ensure smooth data processing, model training, and visualization without system crashes or slow performance. The requirements below are tailored for both academic and professional environments.

#### **3.1 Hardware Requirements**

* **RAM**: A minimum of **4 GB** is recommended. However, **8 GB or higher** will provide better performance, especially when working with larger datasets, performing EDA, or training ensemble models like Random Forest.
* **Processor (CPU)**: Intel **i3 or above**. For faster computations, especially when using scikit-learn models or performing visualizations, an **i5 or i7 processor** is preferable.
* **Storage**: At least **500 MB** of free space is required to store the dataset, notebook, plots, and exported files. More may be needed if dashboards or external tools like Tableau/Power BI are used.
* **Display**: A standard HD display (13” or larger) is sufficient for programming and viewing plots, though higher resolutions can help with detailed visual analysis.

#### **3.2 Software Requirements**

* **Operating System**: Windows, macOS, or Linux (the project is platform-independent since it runs in the cloud or on Python environments).
* **Python Version**: Python **3.7 or later** is required for compatibility with modern libraries.
* **Development Environment**:
  + **Google Colab** (preferred for beginners and collaboration): No installation required; runs in the cloud.
  + **Jupyter Notebook**: Recommended for local development and iterative experimentation.
  + **VS Code / PyCharm** (optional): Useful for modular Python scripting and professional development.

#### **3.3 Required Python Libraries**

The following Python packages are essential for implementing the data processing, analysis, modeling, and visualization components of the project:

| **Library** | **Purpose** |
| --- | --- |
| pandas | Data loading, cleaning, and manipulation |
| numpy | Numerical operations and array handling |
| matplotlib | Static data visualizations (bar charts, plots) |
| seaborn | Statistical data visualization (boxplots, heatmaps) |
| plotly | Interactive charts and dashboards (optional) |
| scikit-learn | Machine learning modeling and evaluation |
| openpyxl | Reading Excel files (.xlsx) |
| pandas-profiling | Automated EDA (optional, for quick summary) |

#### **3.4 Optional Tools**

* **Power BI / Tableau**: These tools are used for building interactive and business-friendly dashboards. If dashboard presentation is required as a deliverable, at least one of these is recommended.
* **Gradio or Streamlit**: These Python libraries allow quick deployment of ML models as interactive web apps for demonstration purposes.

**4. Project Objectives**

The primary objective of this project is to develop a reliable, accurate, and interpretable machine learning model that can predict house prices based on historical property data. By leveraging structured housing datasets, we aim to uncover the key attributes that influence house valuation, build a predictive model, and provide insights that can help stakeholders make informed real estate decisions.

#### **Key Technical Objectives:**

* **Build a predictive model** using machine learning algorithms (Linear Regression, Random Forest) to estimate house prices.
* **Identify and rank features** such as area, location, number of bedrooms (BHK), and bathrooms that have the most significant impact on property prices.
* **Perform thorough data preprocessing**, including handling missing values, outliers, categorical encoding, and feature scaling to ensure model quality.
* **Conduct Exploratory Data Analysis (EDA)** to understand data distributions, relationships, and trends in property pricing.
* **Visualize results and performance** using charts and metrics to interpret model behavior and communicate findings clearly.

#### **Expected Outputs:**

* A clean and processed dataset ready for modeling.
* Visual reports showing trends in house prices by region, BHK, area, and other factors.
* A trained and evaluated machine learning model with high accuracy and low error.
* Feature importance rankings to identify which attributes drive house prices.
* Predictive insights and suggestions for stakeholders (buyers, sellers, developers).

#### **Business Impact:**

* Helps **real estate agents** set fair and competitive property prices.
* Assists **buyers and investors** in making data-driven decisions.
* Enables **property listing platforms** to incorporate automated price estimation tools.
* Reduces the subjectivity and uncertainty in property valuation processes.

**5. Project Workflow (Flowchart)**

**Flow:**

**Data Collection**

**Evaluation**

**Recommendations**

**Visualization**

**Model Building**

**Feature Engineering**

**EDA**

**Data Cleaning**

**6. Dataset Description**

The dataset used in this project is titled **house\_price.csv** and serves as the foundation for training the prediction model and performing statistical analysis. It contains structured data representing various features of residential properties, including both numerical and categorical attributes that influence market value.

#### **Dataset Source:**

The dataset was collected from a real estate platform or compiled manually for academic purposes. It mimics data available on platforms like **MagicBricks**, **99acres**, or **Kaggle**, representing real-world housing features and pricing trends across different regions.

#### **Data Type:**

* **Structured Dataset**: The data is organized into rows and columns in a tabular format.
* **Format**: Comma-separated values (CSV), suitable for processing using Python libraries like pandas.

#### **Size of Dataset:**

* **Total Records (Rows):** Approximately **1,330** individual property listings.
* **Total Features (Columns):** **5 to 7** key attributes depending on version (after cleaning).

Typical columns include:

* **location** – Categorical: Region or locality where the house is situated.
* **area** – Numerical: Total built-up area of the property (in square feet).
* **bhk** – Numerical: Number of bedrooms.
* **bath** – Numerical: Number of bathrooms.
* **price** – Numerical: Target variable representing house price (in lakhs or INR).

Some datasets may also include:

* **furnishing**, **balcony**, **parking**, or **status** (e.g., ready to move, under construction).

#### **Static or Dynamic Nature:**

* The dataset is **static**, meaning it represents a snapshot in time and does not get updated automatically. This is suitable for building and evaluating models, but the model should be retrained periodically using fresh data for production use.

#### **Target Variable:**

* **price** – This is the dependent variable (label) that we aim to predict using regression algorithms.

#### **Sample Snapshot of Data:**

| **Location** | **Area** | **BHK** | **Bath** | **Price** |
| --- | --- | --- | --- | --- |
| Whitefield | 1200 | 2 | 2 | 65 |
| Indiranagar | 2400 | 4 | 4 | 180 |
| Koramangala | 1500 | 3 | 3 | 120 |

#### **Initial Observations:**

* Some inconsistencies such as duplicate entries and outliers in area and price were observed.
* Location is the only major categorical feature and requires encoding before modeling.
* Feature distributions show skewness that may require scaling or transformation for certain algorithms.

**7. Data Preprocessing**

Data preprocessing is a critical step in any machine learning pipeline, as it ensures that the dataset is clean, consistent, and suitable for modeling. In this project, we carried out several preprocessing tasks to prepare the house price dataset for analysis and machine learning. These steps include handling missing values, removing duplicates, transforming data types, encoding categorical variables, detecting outliers, and feature scaling.

**7.1 Handling Missing Values**

The dataset was checked for missing values using the df.isnull().sum() function. While most columns were complete, a few records had null entries, especially in the area, price, or bhk columns. Since these fields are crucial for prediction and cannot be imputed reliably without introducing bias, rows with missing values in these columns were **dropped**.

python

CopyEdit

df = df.dropna(subset=['price', 'area', 'bhk'])

#### **7.2 Removing Duplicate Records**

We used the df.duplicated().sum() function to identify any duplicate rows. Duplicates can distort the learning process by overemphasizing repeated patterns, so they were removed:

python

CopyEdit

df = df.drop\_duplicates()

#### **7.3 Data Type Conversion and Consistency**

All numerical fields such as price, area, bhk, and bath were ensured to be of the correct data types (i.e., float or integer). Some columns initially appeared as strings (especially if imported from Excel), and these were converted to numeric:

python

CopyEdit

df['area'] = pd.to\_numeric(df['area'], errors='coerce')

df['price'] = pd.to\_numeric(df['price'], errors='coerce')

After conversion, any resulting NaN values (from failed conversions) were again removed.

#### **7.4 Encoding Categorical Variables**

The dataset includes categorical columns such as location. Since machine learning models do not understand text-based data, we applied **Label Encoding** to convert the location column into numeric form using the LabelEncoder from sklearn:

python

CopyEdit

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

df['location\_encoded'] = le.fit\_transform(df['location'])

This new encoded column (location\_encoded) was used in the feature set for modeling.

#### **7.5 Outlier Detection and Treatment**

Outliers, especially in area and price, can skew the model's understanding of the data. We used **boxplots and the IQR method** to detect and remove extreme values. For instance, properties with unrealistically high prices or excessively large areas were filtered:

python

CopyEdit

Q1 = df['price'].quantile(0.25)

Q3 = df['price'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['price'] >= Q1 - 1.5 \* IQR) & (df['price'] <= Q3 + 1.5 \* IQR)]

#### **7.6 Feature Scaling (Standardization)**

Since features like area and price can have different scales, and some models (e.g., KNN, SVM) are sensitive to scale, **StandardScaler** was applied to normalize the numerical features:

python

CopyEdit

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled\_columns = ['area', 'bhk', 'bath']

df[scaled\_columns] = scaler.fit\_transform(df[scaled\_columns])

This step ensures that all features contribute equally to the model and avoids dominance by high-magnitude features.

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

Exploratory Data Analysis (EDA) is the phase where we explore the structure, patterns, relationships, and trends in the dataset before applying any machine learning models. In the House Price Prediction project, EDA helped uncover valuable insights about which features affect housing prices the most, detect data inconsistencies, and choose relevant features for modeling.

EDA was performed using Python libraries like **pandas**, **matplotlib**, **seaborn**, and **plotly** for interactive visuals. The analysis included both **univariate** and **bivariate/multivariate** exploration techniques.

#### **8.1 Univariate Analysis**

Univariate analysis focuses on understanding the distribution of individual variables:

* **Price Distribution**: A histogram of price showed a **right-skewed distribution**, indicating that most homes are priced in the low to mid-range bracket, with fewer high-end luxury properties.

python

CopyEdit

sns.histplot(df['price'], kde=True)

* **Area**: The area column showed a **wide variation**, with some extreme outliers representing very large homes or plots.
* **BHK and Bath Counts**: A count plot revealed that **2BHK and 3BHK** houses are the most common in the dataset, followed by 1BHK and 4BHK.

#### **8.2 Bivariate and Multivariate Analysis**

This step examines the relationships between features and how they jointly influence the target variable (price):

* **Area vs Price**:  
  A scatter plot of area vs price demonstrated a **positive linear trend**, as expected. Larger properties tend to be priced higher.

python

CopyEdit

sns.scatterplot(x='area', y='price', data=df)

* **BHK vs Price (Boxplot)**:  
  The boxplot showed that prices generally increase with BHK count, but the price variation becomes broader as the number of bedrooms increases. This is due to location and luxury features affecting pricing.

python

CopyEdit

sns.boxplot(x='bhk', y='price', data=df)

* **Correlation Heatmap**:  
  A heatmap of the numerical features (price, area, bhk, bath) showed a **moderate to strong correlation** between area and price, while bhk and bath showed weaker but positive correlation with price.

python

CopyEdit

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

* **Location vs Price**:  
  Average pricing by location revealed that certain areas (e.g., "Indiranagar", "Koramangala") consistently have higher property prices. This highlighted location as a significant categorical driver of price.

#### **8.3 Visualizations Used**

| **Visualization Type** | **Purpose** |
| --- | --- |
| Histogram (Price) | Understand price distribution and skewness |
| Boxplot (BHK vs Price) | Examine price variation across BHK categories |
| Scatter Plot (Area vs Price) | Detect trends and outliers in price vs area |
| Heatmap (Correlation) | Assess strength of relationships between features |
| Countplot (BHK, Bath) | Identify frequency of property types |

**9. Insights and Interpretation**

This section translates the visual and statistical findings from the Exploratory Data Analysis (EDA) phase into actionable business insights. These insights help us understand which features significantly influence house prices and how these relationships can support smarter pricing, investment, and marketing strategies in the real estate domain.

The dataset, which includes attributes such as area, bhk, bath, and location, revealed several important patterns that are essential for predictive modeling and practical decision-making.

#### **Key Business Insights:**

* **Larger area leads to higher property price**  
  A strong positive correlation was found between area and price. This suggests that area is one of the most important features to consider when estimating property value. However, the relationship is not perfectly linear—premium areas can command high prices even for smaller spaces.
* **Location significantly affects pricing**  
  Certain areas such as "Indiranagar", "Whitefield", and "Koramangala" showed consistently higher prices, even for properties with fewer bedrooms or smaller areas. This highlights the importance of location as a categorical variable and confirms that neighborhood prestige and connectivity are major price drivers.
* **3BHK and 2BHK homes are the most popular configurations**  
  These categories made up the majority of the dataset. 3BHK homes tend to offer a better balance between price and space, making them attractive to mid-range buyers. Prices rise with BHK count, but the rate of increase flattens after 4BHK, suggesting diminishing returns for higher configurations in certain localities.
* **Number of bathrooms moderately influences price**  
  Properties with 2 or more bathrooms tend to fetch higher prices, which aligns with buyer preferences for comfort and convenience. However, this influence is secondary to area and location.
* **Presence of outliers indicates luxury segment behavior**  
  A small number of homes with very high price and area values exist in the dataset. These likely belong to luxury or premium segments and can skew the model if not treated properly. Their removal or isolation improves model performance.

#### **Interpretation and Modeling Relevance:**

The insights from EDA not only improve our understanding of real estate pricing dynamics but also inform the feature selection and engineering process for model building. For example:

* Area and Location were retained as **key predictive features**.
* BHK and Bath were used in combination with Area to improve interpretability.
* Outliers were removed to prevent distortion in model learning.
* Categorical encoding was applied to Location to allow machine learning algorithms to learn from its hidden influence.

These insights laid the foundation for building a more robust and accurate prediction model, ensuring that the machine learning algorithm can learn relevant patterns from the data and generalize well to unseen property listings.

**10. Recommendations**

Based on the findings from the Exploratory Data Analysis (EDA) and the insights derived from model results, the following recommendations are proposed to enhance decision-making in property pricing, sales strategy, and future data usage. These are divided into **short-term tactical actions** and **long-term strategic initiatives** for stakeholders such as real estate developers, brokers, and listing platforms.

#### **10.1 Short-Term Actions**

**Incorporate data-driven pricing into property listings**  
Utilize the trained machine learning model to suggest price ranges for new property listings based on features like area, location, and number of bedrooms. This will improve pricing accuracy and increase buyer engagement.

**Focus marketing on mid-range BHK properties**  
Since 2BHK and 3BHK units are the most in-demand, allocate marketing resources and campaigns toward promoting these categories in popular areas such as Whitefield and Koramangala.

**Highlight value-for-money properties in premium locations**  
Use model insights to identify properties that are underpriced in high-demand areas and promote them to budget-conscious buyers as smart investments.

**Remove or adjust outlier listings**  
Flag property listings that fall far outside the expected price range (either too high or too low) to prevent user confusion and maintain credibility in pricing models.

#### **10.2 Long-Term Strategic Moves**

**Integrate the prediction model into a live web platform or CRM**  
Develop an interactive tool or app (using Streamlit or Gradio) where agents or users can input property details and instantly receive price predictions. This can enhance customer trust and streamline operations.

**Build a dynamic pricing system**  
Use the model’s predictions along with market trend data to create a dynamic pricing algorithm that updates with time, seasonality, and demand patterns—similar to models used in the airline and hotel industries.

**Expand dataset with more features and external APIs**  
Enhance the dataset with additional variables such as proximity to schools, public transport access, and crime rates. This can improve model accuracy and allow for hyper-local predictions.

**Retrain the model regularly**  
Since real estate markets are dynamic, retrain the prediction model every 3–6 months using new listings and transactions to maintain predictive performance and relevance.

These recommendations support the development of a **scalable, intelligent pricing tool** for the real estate industry. Implementing them can lead to better customer satisfaction, more efficient transactions, and smarter investment decisions.

**11. Visualizations / Dashboard**

Visualizations are a key component of this project, used to explore the dataset, communicate insights, and interpret model performance in a clear and intuitive way. Various types of charts and graphs were created using Python libraries such as **Matplotlib**, **Seaborn**, and **Plotly**. These helped in understanding the data distribution, feature relationships, and identifying trends that affect housing prices.

#### **11.1 Data Visualization Techniques Used**

Here are the core visualizations used in the project and what each represents:

| **Visualization Type** | **Purpose** |
| --- | --- |
| **Histogram (Price Distribution)** | Showed that house prices are right-skewed with most properties priced in the mid-range. |
| **Boxplot (BHK vs Price)** | Compared price variations across different bedroom configurations (1BHK, 2BHK, etc.). |
| **Scatter Plot (Area vs Price)** | Revealed a positive trend where larger houses generally have higher prices. |
| **Heatmap (Correlation Matrix)** | Helped identify strong correlations among numerical features like area, bhk, and price. |
| **Bubble Chart** (Optional) | Showed area vs price with bubble size representing bhk and color by location. Useful for showing multidimensional relationships in a single chart. |

#### **11.2 Key Visual Insights**

* Houses with larger **area** tend to have higher prices, which was visible in the scatter plots and heatmap correlations.
* The **location** variable introduces a lot of variability in price, as seen in boxplots segmented by location.
* **3BHK properties** generally have the most balanced pricing pattern with minimal outliers.
* **Outliers** were identified through boxplots and removed to improve model performance.

These visual tools were critical during **Exploratory Data Analysis (EDA)** and model evaluation, helping to validate assumptions and optimize feature selection.

#### **11.3 Interactive Dashboard (Optional – Power BI / Tableau)**

If advanced tools like **Power BI** or **Tableau** were used, the dashboard may include:

* **Dynamic filters** for location, BHK, or area size.
* **Interactive heatmaps** and bar charts to compare average prices by region.
* **Real-time input fields** to enter area/BHK and get a model-predicted price (if integrated with backend prediction models).

This allows business users and clients to explore the data visually without writing any code, making insights more accessible and actionable.

#### **Why Visualization Matters**

* It helps simplify complex relationships.
* It supports decision-makers who may not be data-savvy.
* It reveals hidden trends and patterns that might not be visible in raw numbers.
* It aids in model transparency and trust by explaining what influences predictions.

**12. Final Deliverables**

The following items have been compiled and submitted as part of the final project deliverables for the House Price Prediction system. These components represent the complete lifecycle of the project—from data analysis and modeling to documentation and reporting—and can be used for academic submission, deployment, or future improvement.

#### **12.1 Cleaned and Documented Jupyter/Colab Notebook**

* A fully functional Python notebook (.ipynb) hosted on **Google Colab** or available as a **Jupyter Notebook**.
* Contains:
  + Data import and inspection
  + Data preprocessing (missing values, encoding, scaling)
  + Exploratory Data Analysis (EDA) with visualizations
  + Model training and evaluation
  + Final predictions and interpretation
* All steps include clear code comments and markdown explanations for easy understanding.

#### **12.2 Machine Learning Model File (Optional)**

* The trained model (e.g., **Random Forest Regressor**) can be exported as a .pkl or .joblib file.
* This file can be reused for deployment in a web application, API, or dashboard tool without retraining the model.

#### **12.3 Dashboard File (Optional)**

* If **Power BI** or **Tableau** was used, the .pbix or .twbx file is included.
* Dashboard allows filtering by location, area, or BHK to dynamically explore pricing trends.
* Useful for presenting insights to real estate professionals or business stakeholders.

#### **12.4 Final Report Document**

* A **10–12 page report** in .docx or .pdf format.
* Includes:
  + Problem Statement
  + Objectives
  + Data Description
  + Preprocessing Steps
  + EDA and Visualizations
  + Model Performance
  + Insights, Recommendations, and Future Scope
* Enhanced with tables, charts, screenshots, and flowcharts.

**13. Source Code Structure**

**├── data/**

**│ └── house\_price.csv**

**├── notebooks/**

**│ └── house\_price\_prediction.ipynb**

**├── dashboard/**

**│ └── house\_price\_dashboard.pbix**

**├── report/**

**│ └── final\_report.docx/pdf**

**└── README.md**

**14. Future Scope**

While the current implementation of the House Price Prediction model offers a functional and valuable tool for estimating property values based on historical data, there are several areas in which the project can be expanded and enhanced. These future improvements aim to increase model accuracy, usability, real-world relevance, and integration with business processes.

#### **1. Real-Time Data Integration**

Currently, the model operates on a static dataset. In a real-world deployment, housing prices change frequently based on market dynamics, interest rates, and seasonal trends. By connecting the model to **real-time data sources** such as real estate APIs (e.g., Zillow, MagicBricks, or housing portals), the model can be retrained automatically at scheduled intervals to reflect the most recent trends.

#### **2. Geospatial and External Data Enrichment**

Future versions of the model can be enhanced by including **geospatial and contextual features** such as:

* Distance to public transport hubs, schools, and hospitals
* Nearby amenities like malls, parks, and offices
* Local crime rates, air quality, or noise levels  
  This can be achieved by integrating APIs such as Google Maps or OpenStreetMap. Adding such location intelligence would allow for more **granular and location-aware pricing**.

#### **3. Deep Learning and Ensemble Modeling**

Although current models like Random Forest perform well, future development could explore more advanced techniques like:

* **Gradient Boosting (e.g., XGBoost, LightGBM)** for better performance on tabular data
* **Neural Networks** for learning complex non-linear relationships
* **Stacking and Blending** to combine multiple models and reduce generalization error

These could yield even better results, especially in large-scale implementations.

#### **4. NLP for Feature Extraction from Descriptions**

Real estate listings often contain rich textual descriptions. Future iterations could use **Natural Language Processing (NLP)** to extract features such as:

* Renovated kitchen
* Sea view
* Gated community  
  This qualitative information can be quantified and included in the prediction model to further improve its precision.

#### **5. Deployment as a Web or Mobile Application**

A streamlined and user-friendly **web or mobile app** could be developed using tools like **Streamlit**, **Flask**, or **React Native**. This would allow end-users—homebuyers, sellers, and agents—to enter property details and instantly receive a predicted price, making the tool more accessible.

#### **6. Integration with CRM or Real Estate Portals**

The model could be integrated into Customer Relationship Management (CRM) systems or real estate listing websites. This would allow automatic suggestions of optimal pricing for new listings or generate alerts when properties are under/overpriced based on market norms.

*15.TEAM MEMEBERS:*

HARSHITHA.B

VIDHYA.S

VENKATESHWARI.B