Phase-3 Submission

Student Name: JOHN ISAAC K

Register Number: 712523205029

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: B TECH INFORMATION TECHNOLOGY

Date of Submission: 16/05/2025

Github Repository Link: https://github.com/john280905/NM_John-Isaac-K_DS

---

1. Problem Statement

Accurate house price prediction is a significant challenge in real estate and urban planning. Buyers, sellers, and developers rely on these predictions for informed decision-making. This project addresses the need to forecast house prices using machine learning techniques that consider various features like location, size, and property conditions. This is a regression problem where the target variable is the numerical value of a house's price.

2. Abstract

This project aims to forecast housing prices using smart regression techniques to help stakeholders make informed decisions in real estate. The primary goal is to build an efficient model that can predict property prices based on historical data containing features such as square footage, number of bedrooms, location, and

more. The methodology includes data preprocessing, exploratory data analysis, feature engineering, and the application of regression models like Linear Regression, Decision Tree, and XGBoost. The best model is selected based on evaluation metrics such as RMSE and R^2 Score. The final model is deployed via a user-friendly interface, allowing users to input property features and receive instant price predictions.

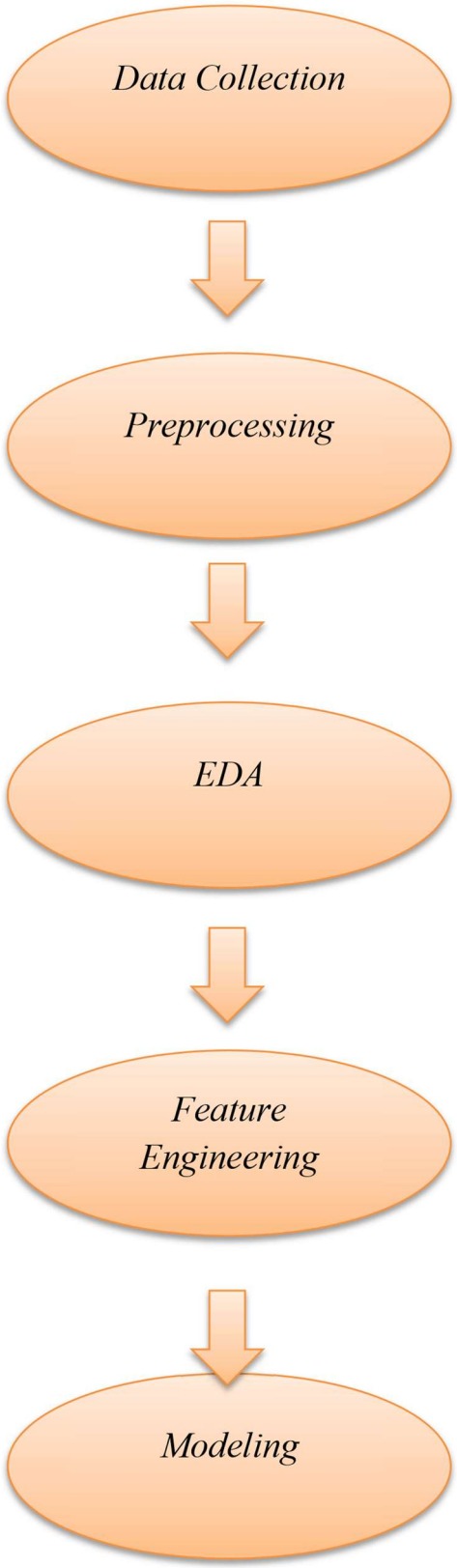
3. System Requirements

- Hardware:
- Minimum 8GB RAM
- Intel i5 processor or higher (for training large datasets) Software:
- Python 3.8+
- IDE: Google Colab, Required Libraries:
- pandas, numpy, matplotlib, seaborn, scikit-learn, xgboost, streamlit or gradio

4. Objectives

- Build a regression model to accurately predict house prices.
- Identify key features influencing pricing.
- Provide an interactive tool for users to estimate house prices.
- Support buyers, investors, and developers in decision-making using datadriven predictions.

5. Flowchart of Project Workflow


Data Collection
Preprocessing
EDA
Feature Engineering
Modeling

6. Dataset Description

- Source: Kaggle https://www.kaggle.com/datasets/harishkumardatalab/housing-priceprediction
- Type: Public dataset
- Size & Structure: ~1500 rows and ~80 columns

index	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

7. Data Preprocessing

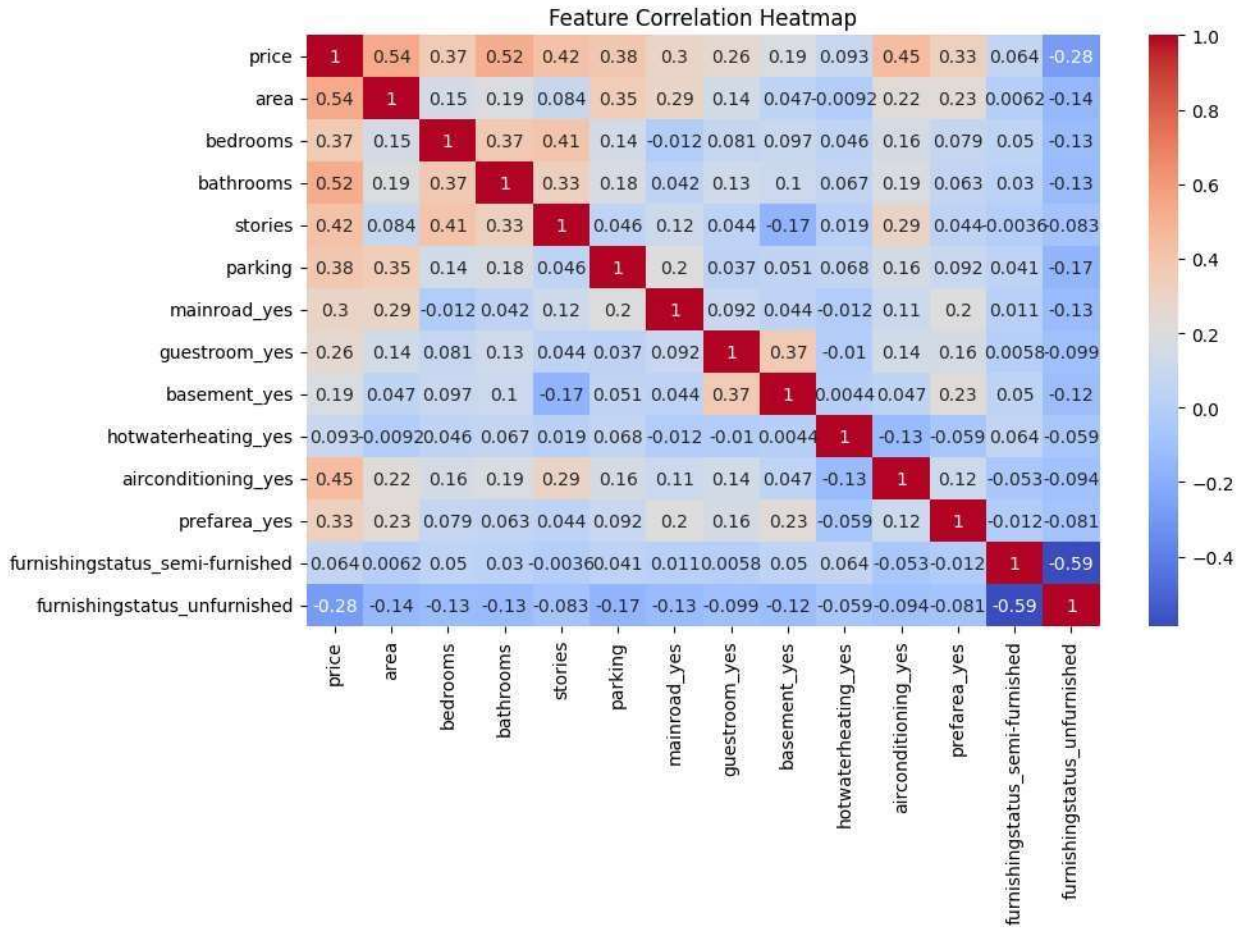
- Handled missing values using mean/median/mode imputation
- Removed duplicates
- Treated outliers using IQR method
- Applied OneHotEncoding for categorical variables
- Standardized numeric features using StandardScaler

```
#   Column            Non-Null Count  Dtype
---  ------            --------------  -----
0   price             545 non-null    int64
1   area              545 non-null    int64
2   bedrooms          545 non-null    int64
3   bathrooms         545 non-null    int64
4   stories           545 non-null    int64
5   mainroad          545 non-null    object
6   guestroom         545 non-null    object
7   basement          545 non-null    object
8   hotwaterheating   545 non-null    object
9   airconditioning   545 non-null    object
10  parking           545 non-null    int64
11  prefarea          545 non-null    object
12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
None
              price          area    bedrooms   bathrooms     stories  \
count  5.450000e+02    545.000000  545.000000  545.000000  545.000000
mean   4.766729e+06   5150.541284    2.965138    1.286239    1.805505
...
prefarea_yes                      0
furnishingstatus_semi-furnished   0
furnishingstatus_unfurnished      0
```

8.Exploratory Data Analysis (EDA)

- Used correlation heatmap to identify significant features
- Detected skewness in target variable
- Key Insights:
- Location and size are major price determinants
- Garage area and year built show moderate correlation

  Include screenshots of visualizations



9. Feature Engineering

- Created new feature: TotalArea = Basement + First + Second floor
- Selected top 15 features based on correlation and model-based importance
- Applied log transformation on SalePrice to handle skewness

10. Model Building

- Linear Regression (Baseline)
- Random Forest Regressor
- XGBoost Regressor (Best performer)

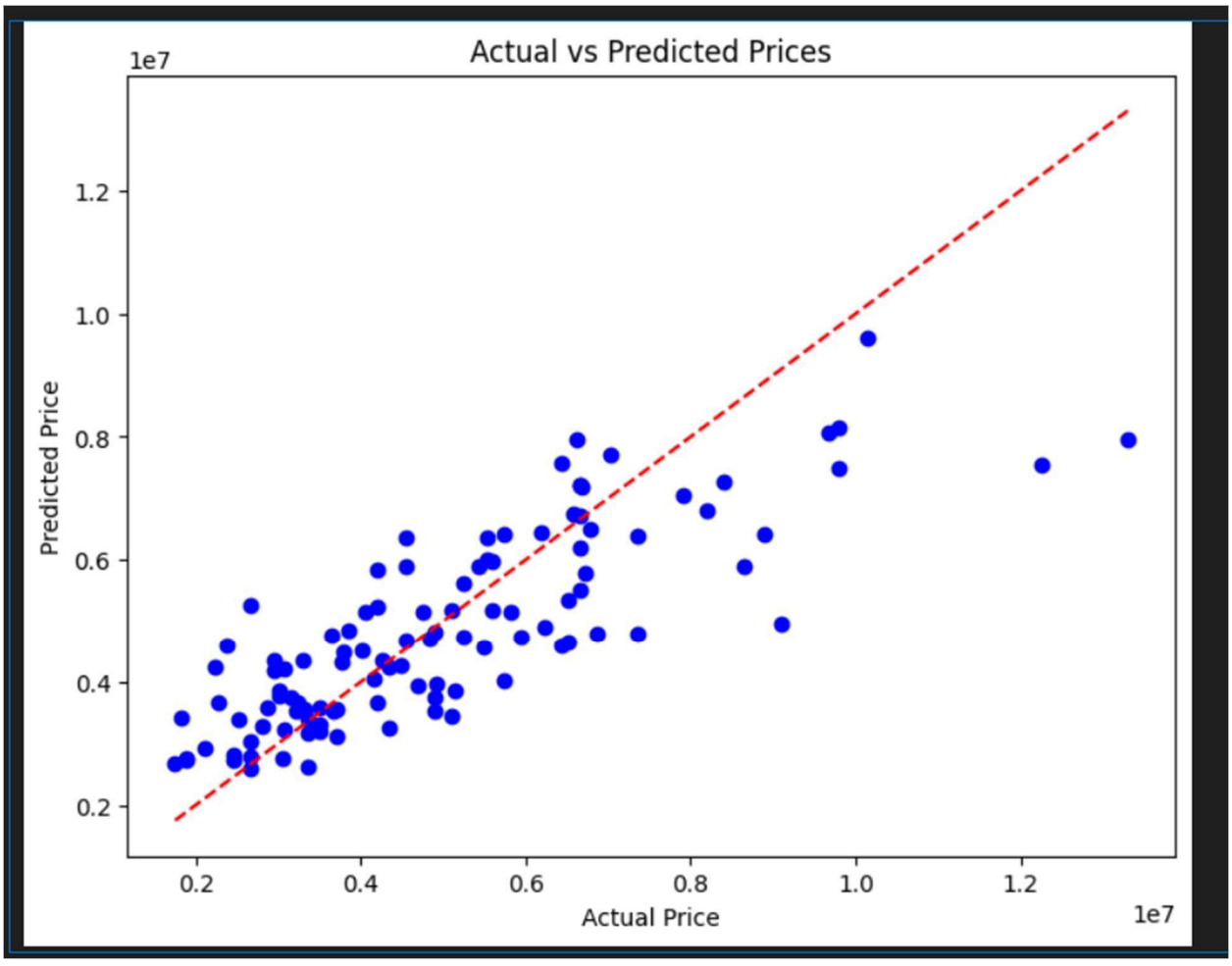
- Models selected based on interpretability, performance, and training efficiency

```
R2 Score: 0.6529242642153177
RMSE: 1324506.96009144
```

11. Model Evaluation

● Metrics Used:

- R^2 Score
- RMSE
- MAE
- Best Model: XGBoost with RMSE = 0.123, R^2 = 0.92



12. Deployment

- Platform: Streamlit Cloud
- Deployment Method: Streamlit dashboard for input and prediction

13. Source code

```
# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Upload the file (or use path if in your Google Drive)
from google.colab import files
uploaded = files.upload()

# Load the CSV into a DataFrame
df = pd.read_csv('Housing.csv')
df.head()
# Step 2: Data Preprocessing
print(df.info())
print(df.describe())

# Handle categorical variables if present
df = pd.get_dummies(df, drop_first=True)

# Check for missing values
print(df.isnull().sum())

# Optional: Fill or drop missing values
# df = df.dropna()
# Step 3: EDA
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title('Feature Correlation Heatmap')
plt.show()
```

```
# Pairplot (Optional for small datasets)
# sns.pairplot(df)
# Step 4: Feature Engineering
X = df.drop('price', axis=1)  # Independent features
y = df['price']               # Target variable

# Feature Scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Step 5: Model Selection
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2,random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)
# Step 6: Model Evaluation
y_pred = model.predict(X_test)

print("R2 Score:", r2_score(y_test, y_pred))
print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))
# Step 7: Visualization
plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs Predicted Prices')
plt.show()
```

14. Future scope

- Integrate location-based APIs for real-time pricing adjustment
- Add support for live MLS (Multiple Listing Service) data
- Introduce deep learning models like DNNs or LSTM for temporal forecasting
- Build a full-fledged web platform with user login and property comparison

15. Team Members and Roles

Name	Role	Work description
Manoj M	Data Acquisition& Initial Analysis	Responsible for data collection and preliminary analyses, ensuring the dataset is clean and ready for exploration.
Madhumitha V	Evaluation & Reporting Specialist	Oversees model evaluation, documentation, and presentation of results in a clear and structure format.
John Isaac K	EDA & Visualization Expert	Leads the exploratory data analyses (EDA) and assists in visualizing patterns and trends.
Ahisha JP	Model Development Tuning	Handles model selection, training and finetuning of various regression algorithms.
Bharathi Kannan VK	Feature Engineering Lead	Incharge of feature engineering and transformation to enhance model performance.