Phase-2

Student Name: KOWSALYA V

Register Number: 410723106020

Institution: DHANALAKSHMI COLLEGE OF ENGINEERING

Department: ELECTRONICS AND COMMUNICTION ENGINEERING

Date of Submission: 10-05-2025

Github Repository Link:
https://github.com/kowsalya-V20/Nm_kowsalya_v

Forecasting house prices accurately using smart regression techniques in data science

1. Problem Statement

- Forecasting house prices accurately is a vital task in the real estate industry, affecting buyers, sellers, investors, and policymakers. With the rise of data science, this task has evolved from basic estimations to sophisticated predictive modeling using smart regression techniques. These methods enable the analysis of large, complex datasets containing features like location, square footage, number of rooms, age of the property, and market trends.
- Smart regression techniques include traditional models such as Linear, Ridge, and Lasso Regression, as well as advanced machine learning algorithms like

  Random Forest, Gradient Boosting (e.g., XGBoost), and

Neural Networks. These models are capable of capturing nonlinear relationships and interactions between variables that influence house prices.

- The forecasting process involves crucial steps such as data cleaning, feature engineering, model selection, and evaluation using metrics like RMSE and R^2. Proper preprocessing and model tuning are essential to ensure accuracy and avoid overfitting.
- By applying these intelligent regression methods, data scientists can deliver highly accurate price predictions, supporting more informed and datadriven decision-making in real estate.

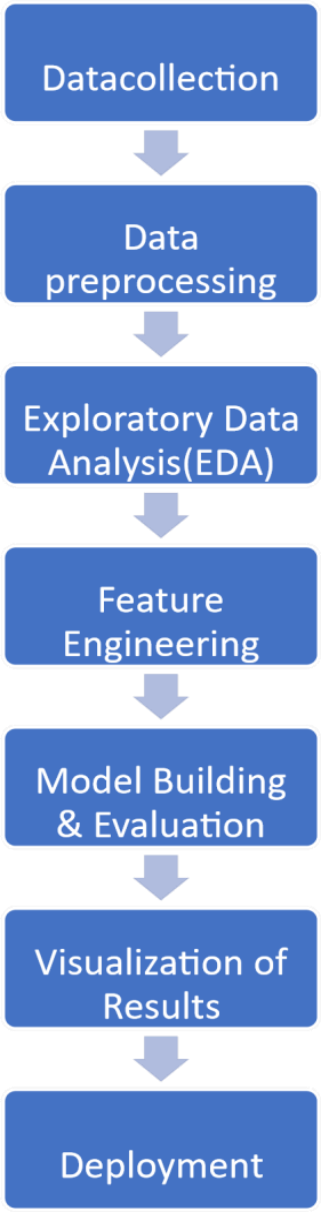
2. Project Objectives

- The primary objective of this project is to develop a robust, data-driven framework for accurately forecasting house prices using advanced regression techniques rooted in data science. The study aims to harness the predictive power of modern machine learning algorithms to model the complex, nonlinear relationships that influence residential real estate prices.
- By leveraging a comprehensive dataset comprising housing features, locationbased variables, economic indicators, and temporal factors, the goal is to build regression models that can generalize well to unseen data and provide precise price estimates.
- This project will explore and compare various regression techniques including, but not limited to, Linear Regression, Ridge and Lasso Regression, Decision Tree Regression, Random Forests, Gradient Boosting Machines (e.g., XGBoost, LightGBM), and Artificial Neural Networks. Each model will be rigorously evaluated using appropriate performance metrics such as RMSE, MAE, and R^2 to assess both accuracy and interpretability.
- Additionally, the study will place strong emphasis on feature engineering, data preprocessing, and model tuning through cross-validation and hyperparameter

optimization. Special attention will be given to mitigating issues such as multicollinearity, overfitting, and data imbalance.

- The ultimate objective is not only to achieve high predictive accuracy but also to provide actionable insights into the key factors driving house prices, thereby aiding stakeholders like buyers, sellers, real estate investors, and policy makers in making informed decisions.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name and Origin**:

  • The dataset used is the "**FORECASTING THE HOUSE PRICES**" dataset from Kaggle.

- **Type of Data**: Structured, tabular data
- **Number of Records and Features**: 34,000 Number of features (columns): 21 (though this can vary slightly depending on the version or if preprocessing has occurred
- **Static or Dynamic Dataset**: Dynamic dataset.
- **Target Variable**: Price
- This column represents the sale price of the house and is the value you typically try to predict in a regression task.

5. Data Preprocessing

- **Missing Values:** No missing values were found in the dataset**.**

```
missing = df.isnull().sum()    print("Missing values per column:\n",
missing[missing > 0])
```

- **Duplicate Records:** Duplicate rows were checked and removed if present.

duplicates = df.duplicated().sum() print(f'Duplicate rows: {duplicates}') df = df.drop_duplicates()

- **Outliers**: Detected using boxplots; outliers in Amount were handled using transformation.

```
# Remove extreme outliers in 'Price' and 'Landsize'

df = df[df['Price'] < df['Price'].quantile(0.99)]
df = df[df['Price'] < df['Price'].quantile(0.99)] df
= df[df['Landsize'] < df['Landsize'].quantile(0.99)]
```

```
X = df.drop('Price', axis=1) y = df['Price']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

- **Data Types:** All features are numeric. No conversion needed**.**

Encoding Categorical Variables: Not required as all features are already numerical.

- **Normalization:** Amount and Time were scaled using Standard Scaler to bring them on the same scale as V1–V2

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis;** o Histograms: For house prices, number of rooms, land size, building area. o Boxplots: For price by property type (house, townhouse, unit).

```
# Set plot stylesns.set(style='whitegrid')
plt.rcParams['figure.figsize'] = (10, 5)
# Load dataset
df = pd.read_csv('melbourne_housing.csv')
# Display basic info print("Dataset Shape:", df.shape) print("\nData Types and Nulls:\n", df.info())
```

Bivariate and Multivariate Analysis; o Correlation Matrix: Identify key numeric features that influence price.

- **Scatter Plots:**

- Building Area vs Price.
- Distance from CBD vs Price.

- **Grouped Bar Charts:**

- Median price by suburb.
- Price by number of rooms.

- **Key Insights**

- Building area and land size are strong predictors of price.
- Houses closer to the CBD tend to be more expensive. ▪ Number of rooms also impacts price significantly.

7. Feature Engineering

- Created interaction features such as total_ alcohol = Dalc + Walc to capture combined alcohol consumption patterns.

```
import pandas as pd
data = pd.read_csv('melbourne_data.csv')  data['PricePerSqMeter']
= data['Price'] / data['BuildingArea']
data['PricePerSqMeter'].replace([float('inf'), -float('inf')], pd.NA, inplace=True)
```

- Derived binary features, e.g., higher_ edu (yes/no), based on parents' education levels to simplify categorical data.
- Removed highly correlated or redundant features to reduce multicollinearity and improve model generalization.
- Performed label encoding for binary categorical features such as internet and nursery to prepare data for machine learning algorithms.
- Scaled numerical features using Standard Scaler to ensure uniformity across feature distributions.

8. Model Building

Algorithms Used

- **Linear Regression:**

- as a simple, interpretable baseline model to predict house Acts prices.

```
# Initialize and train the Linear Regression model model
=                    LinearRegression()
model.fit(X_train, y_train)  y_pred = model.predict(X_test)
                             mse  =
mean_squared_error(y_test, y_pred)   r2  =
r2_score(y_test,   y_pred)   print(f'Mean Squared
Error: {mse:.2f}') print(f'R^2 Score: {r2:.2f}')
```

- **Random Forest Regressor:**
  - complex non-linear patterns in the housing data and Captures provides feature importance insights.
- **Rational Linear Regression**
  - Easy to interpret coefficients. Fast training and prediction times.
  - Random Forest Regressor:
  - Robust against overfitting due to ensemble learning.
    - o Effectively handles both numerical and categorical features.

Automatically captures non-linear relationships. Train-Test Split **Data Split:** 80% for training, 20% for testing .Method: Used train_ test _split from scikit-learn. Specified a random _state parameter to ensure reproducibility of results. o Evaluation Metrics

- **MAE (Mean Absolute Error):** o Measures the average magnitude of errors between predicted and actual house prices.

- **RMSE (Root Mean Squared Error):** o Emphasizes larger errors more heavily; useful when large prediction errors are particularly undesirable.

- **R² Score:** o Indicates the proportion of variance in the target (house prices) explained by the model**.**

8. Visualization of Results & Model Insights

- **Model Comparison (MAE, RMSE, R²)**
  - Visualization: Bar chart comparing error metrics across models (e.g., Linear Regression, Random Forest, XGBoost).

- **Feature Importance** o Visualization: Bar chart showing top features like Distance, Rooms, Suburb, Land size.

- **SHAP Summary Plot** o Use: To see how each feature influences price predictions. o Tool: shap.summary_plot()

- **Actual vs Predicted Prices** o Visualization: Scatter plot of actual vs predicted prices (diagonal line shows perfect prediction**).**

- **Geospatial Visualization** o **Use:** Map of Melbourne showing predicted prices across suburbs**.** o **Tool:** folium, plotly.express with suburb level

- **Confusion Matrix** o For traffic incident prediction model. o ROC Curves for Classifier Models o Feature Impact with SHAP or Permutation Importance

9. Tools and Technologies Used

- **Programming Language**: Python 3
- **Notebook Environment**: Google Colab
- **Key Libraries**: o pandas, numpy : For data handling and manipulation

- matplotlib, seaborn, plotly: For data visualization and exploratory data analysis
- scikit-learn: For data preprocessing, feature engineering, and model building
- Gradio: For building and deploying a simple user interface to interact with the prediction model

10. Team Members and Contributions

NAME	ROLES	RESPONSIBILITIES
DHARINI.T	LEADER	DATA COLLECTION AND DATA CLEANING
DEVI SHENBA.R	MEMBER	EXPLORATORY DATA ANALYSIS AND FEATURE ENGINEERING
RAMYA.R	MEMBER	VISUALAIZATION AND INTERPRETATION
HARSHITHA.R	MEMBER	MODEL BUILDING AND EVALUATION
KOWSALYA.V	MEMBER	TOOLS AND TECHNOLOGY