Phase-3

Forecasting house prices accurately using smart regression techniques in data science

Student Name: Boopalan N

Register Number: 410623244015

Institution: Dhaanish Ahmad college of engineering

Department: Computer Science and Engineering

Date of Submission: 17-05-2025

Github Repository Link:

---

1. Problem Statement

The real estate market is highly dynamic and influenced by numerous factors including economic conditions, location, infrastructure, and property features. Predicting house prices accurately is a challenging task due to this complexity. The aim of this project is to create a robust regression-based machine learning model that can forecast house prices based on historical data and multiple relevant features. This will enable potential buyers, sellers, and real estate agents to make informed decisions and improve market efficiency.

---

2. Abstract

In this project, we explore the application of advanced regression techniques to accurately predict house prices. We start by collecting a comprehensive dataset with diverse features affecting house prices. The data is then cleaned and preprocessed, addressing missing values and encoding categorical data. Through exploratory data analysis (EDA), we gain insights into the relationships between features and prices, guiding effective feature engineering. Multiple regression algorithms including linear regression, regularized regression (Ridge, Lasso, ElasticNet), and ensemble tree methods (Random Forest, XGBoost) are implemented and compared. Hyperparameter tuning and cross-validation enhance the model's generalization. The best-performing model achieves high accuracy, and the project optionally culminates in a deployment-ready web application for real-time predictions.

3. System Requirements

Hardware:

- **Processor:** Intel i5/i7 or equivalent (higher recommended for faster training)
- **RAM:** Minimum 8 GB, ideally 16 GB or more
- **Storage:** At least 10 GB free for datasets and libraries

Software:

- **Operating System:** Windows 10/11, macOS, or Linux
- **Programming Language:** Python 3.8 or later
- **Libraries:**
  - Data manipulation: pandas, numpy
  - Visualization: matplotlib, seaborn
  - Machine learning: scikit-learn, xgboost, lightgbm
  - Deployment: Flask, Streamlit, or Gradio
- **IDE/Notebook:** Jupyter Notebook, VSCode, or PyCharm

4. Objectives

- To collect and preprocess a comprehensive housing dataset with relevant features.
- To analyze data trends and uncover relationships between house attributes and prices.
- To engineer new features that enhance the predictive power of models.
- To implement various regression techniques and compare their performance.
- To optimize model parameters through hyperparameter tuning and validation.
- To build a deployment-ready predictive model enabling real-time price estimation.

---

5. Flowchart of Project Workflow

Start
↓
Data Collection
↓
Data Cleaning & Preprocessing
↓
Exploratory Data Analysis (EDA)
↓
Feature Engineering
↓
Train-Test Split
↓
Model Development
↓
Model Evaluation & Hyperparameter Tuning
↓
Best Model Selection
↓
Deployment (Optional)
↓
End

6. Dataset Description

The dataset used for this project consists of over 1,400 observations of house sales with approximately 80 features covering property characteristics. Key features include:

Feature	Type	Description
SalePrice	Numeric	Target variable — selling price of the house
LotArea	Numeric	Size of the lot in square feet
OverallQual	Categorical	Overall material and finish quality (1-10)
YearBuilt	Numeric	Year the house was built
Neighborhood	Categorical	Geographic area or community
TotalBsmtSF	Numeric	Total basement area in square feet
FullBath	Numeric	Number of full bathrooms
BedroomAbvGr	Numeric	Number of bedrooms above ground

---

7. Data Preprocessing

- **Handling Missing Values:** Missing data is analyzed feature-wise. For numerical features, missing values are imputed using mean or median. For categorical variables, the most frequent category or a new category ‘Unknown’ is assigned.
- **Encoding Categorical Variables:** Categorical features are converted into numerical format via One-Hot Encoding or Label Encoding depending on the number of categories and model requirements.
- **Feature Scaling:** Numerical features are scaled using StandardScaler to normalize distributions and improve model convergence.
- **Outlier Detection and Treatment:** Outliers are identified

using the Interquartile Range (IQR) method or Z-score thresholds. Extreme values are capped or removed to avoid distortion in model training.

---

8. Exploratory Data Analysis (EDA)

- **Distribution Analysis:** Histograms and KDE plots show the skewness of price and other variables.
- **Correlation Matrix:** A heatmap reveals strong positive correlation between SalePrice and OverallQual, GrLivArea, and GarageCars.
- **Boxplots:** Identify how categorical variables like Neighborhood influence house prices.
- **Scatter Plots:** Visualize relationships, e.g., SalePrice vs. LotArea, to detect linear or nonlinear trends.
- **Multicollinearity Check:** Variance Inflation Factor (VIF) analysis is done to detect highly correlated predictors which may degrade model performance.

---

9. Feature Engineering

- **New Features:**
  - HouseAge = CurrentYear - YearBuilt
  - TotalBathrooms = FullBath + 0.5 * HalfBath + BsmtFullBath + 0.5 * BsmtHalfBath
  - IsNew = Indicator if the house was built within last 5 years
- **Feature Transformation:** Log-transform skewed features such as LotArea and SalePrice to reduce skewness and stabilize variance.
- **Feature Selection:** Using techniques like Recursive Feature Elimination (RFE) and feature importance from tree-based

models to retain most influential features.

---

10. Model Building

- **Baseline Model:** Linear Regression to establish a performance benchmark.
- **Regularization:** Ridge and Lasso regression to handle multicollinearity and select relevant features.
- **Ensemble Methods:**
  - Random Forest Regressor for robustness and non-linear pattern capturing.
  - Gradient Boosting (XGBoost/LightGBM) for state-of-the-art predictive accuracy.
- **Support Vector Regression:** Applied to capture non-linear relationships with kernels.
- **Cross-Validation:** 5 or 10-fold cross-validation to reduce overfitting.

---

11. Model Evaluation

Metric	Description
Mean Absolute Error (MAE)	Average absolute difference between predicted and actual prices.
Root Mean Squared Error (RMSE)	Penalizes larger errors more strongly, providing insight on prediction accuracy.
R-squared (R^2)	Percentage of variance explained by the model (higher is better).

---

12. Deployment

- **Web Interface:** Using Flask or Streamlit to build a web app where users input house features and get predicted prices instantly.
- **API Endpoint:** Create an API for integration with other applications.
- **Cloud Deployment:** Host the app on Heroku, AWS, or Google Cloud for accessibility.
- **User Experience:** Simple and intuitive forms with input validation and clear output display.

---

13. Source Code

```
# House Price Prediction Backend using Flask
from flask import Flask, request, jsonify
import numpy as np
import joblib

# Initialize the Flask app
app = Flask(__name__)

# Load the pre-trained model and scaler
model = joblib.load('house_price_model.pkl')
scaler = joblib.load('scaler.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()

    # Extract features from the incoming request
    features = np.array([[
        float(data['lotArea']),
        float(data['yearBuilt']),
        float(data['totalRooms']),
```

```
        float(data['overallQual']),
        float(data['garageArea']),
        float(data['fullBath'])
    ]])

    # Scale the features
    features = scaler.transform(features)

    # Predict using the pre-trained model
    prediction = model.predict(features)

    # Return the result as JSON
    return jsonify({'price': round(prediction[0], 2)})

if __name__ == '__main__':
    app.run(debug=True)
```

---

14. Future Scope

- **Integration with Geospatial Data:** Incorporate maps and proximity to amenities.
- **Use of Deep Learning:** Apply neural networks like DNNs or CNNs for improved pattern recognition.
- **Time Series Analysis:** Incorporate temporal trends for price forecasting over time.
- **Automated Feature Engineering:** Use AutoML tools to discover new features.
- **Expand to Other Markets:** Adapt model for commercial real estate or rental price prediction.

---

15. Team Members and Roles

Name	Role	Responsibilities
Karthik k	Project Manager	Coordination, planning, and deadlines
Eswar S	Data Engineer	Data collection, cleaning, preprocessing
Boopalan N	Data Scientist	Modeling, EDA, feature engineering
Harshath S	Software Developer	Deployment, web app development
Jayakaran K	Documentation Specialist	Report writing, presentation preparation