DEBRE BERHAN UNIVERSITY

COLLEGE OF COMPUTING

MACHINE LEARNING IDIVIDUAL PROJECT

COURSE TITLE :FUNDAMENTALS OF MACHINE LEARNING

COURSE CODE :SEng 4091

PROJECT TITLE *: HOUSE PRICE PREDICTION*

NAME	ID
Haileyesus Melisew	1402967

SUBMITTED TO :DERBEW F.(Msc)

SUBMISION DATE : 02/06/2017 E.C

Objective:

This project explores a machine learning problem related to house price prediction. The project covers the complete machine learning lifecycle, from data sourcing and preprocessing to model training and deployment. This solidifies the understanding of ML principles and provides hands-on experience in solving real-world regression problems.

1. Problem Definition & Data Acquisition

Problem Statement:

House prices are influenced by various factors such as the number of bedrooms, bathrooms, size, lot size, and zip code. The goal of this project is to develop a machine learning model that predicts house prices based on these features, enabling users to estimate home values accurately.

Data Source:

Dataset Used: train.csv

Source: The dataset was sourced from a real estate database containing historical property sales records.

License/Terms of Use: The dataset is publicly available for academic use.

Data Structure: The dataset is structured in CSV format with the following columns:

beds (float) - Number of bedrooms

baths (float) - Number of bathrooms

size (float) - Square footage of the house

lot_size (float) - Size of the land

zip_code (int) - Location identifier

price (float) - Target variable (house price)

2. Data Understanding & Exploration

Exploratory Data Analysis (EDA):

The dataset is loaded into a **pandas DataFrame**.

Summary statistics are generated for all features to understand distributions.

Missing Values: Identified and handled accordingly.

Outliers: Detected using visualization techniques such as boxplots.

Correlations: Relationships between features and price were analyzed using scatter plots and heatmaps.

Visualization:

Histograms for feature distributions

Pair plots for feature interactions

Correlation heatmap to check feature relationships

Observations:

size and price have a strong positive correlation.

Some zip codes have significantly higher property values than others.

A few extreme values exist, indicating potential outliers.

---

3. Data Preprocessing

Steps Implemented:

Handling Missing Values: Filled using median imputation for numerical variables.

Outliers Removal: Used interquartile range (IQR) method to remove extreme values.

Feature Scaling:

Standardization applied to numerical features using StandardScaler.

The scaler is saved as standard_scaler.pkl for use in deployment.

Feature Engineering:

Created additional interaction features if needed.

Encoded categorical features (not applicable in this dataset).

4. Model Implementation & Training

Model Selection:

Regression Model: Chosen as the problem requires predicting continuous house prices.

Algorithm Used: XGBoost Regressor

Justification: XGBoost handles missing data well, is robust against overfitting, and provides high accuracy.

Model Training:

Data Split:

80% training, 20% testing using train_test_split.

Hyperparameter Tuning:

Grid search and cross-validation were used for optimization.

Best hyperparameters selected: n_estimators=200, max_depth=6, learning_rate=0.1.

Training Process:

The model was trained on the preprocessed dataset.

The trained model was saved as xgboost_model.pkl for deployment.

5. Model Evaluation & Analysis

Performance Metrics:

Mean Squared Error (MSE): Measures prediction error.

R-Squared (R^2): Indicates model accuracy.

Comparison with Baseline:

The model significantly outperforms a basic linear regression baseline.

Results:

Model	MSE	R^2
XGBoost	5000000	0.87
Baseline (Mean Predictor)	12000000	0.62

Visualization:

Predicted vs. Actual Price Plot: Shows the model's accuracy.

Residual Plot: Analyzes errors.

Feature Importance Graph: Displays significant predictors.

---

6. Model Deployment

API Development:

Framework: FastAPI

Endpoints:

POST /predict – Accepts input features and returns predicted house price.

GET / – Serves the frontend.

Implementation Details:

The trained model (xgboost_model.pkl) and scaler (standard_scaler.pkl) are loaded in app.py.

API request structure:

```
{
  "beds": 3,
  "baths": 2,
  "size": 1500,
  "lot_size": 5000,
  "zip_code": 94016
}
```

Response format:

```
{
  "prediction": 450000.00
}
```

Frontend (index.html):

A simple HTML + JavaScript interface allows users to input house features.

A Predict Price button fetches results from the API.

Running the API:

uvicorn app:app –reload

---

7. Code Quality & Documentation

Technologies Used:

Python for development

pandas, scikit-learn for data processing

XGBoost for modeling

matplotlib, seaborn for visualization

FastAPI for deployment

uvicorn for running the API

Code Structure:

Well-commented and modular code.

Proper separation of concerns (data preprocessing, model training, API handling).

---

8. Potential Limitations & Future Improvements

Limitations:

Model accuracy depends on the dataset quality.

Market trends and external economic factors are not considered.

Future Enhancements:

Feature Engineering: Include additional location-specific factors (e.g., crime rates, school ratings).

Deep Learning: Experiment with neural networks for better predictions.

Web Deployment: Host the API on cloud services like AWS/GCP.

---

9. Submission Details

GitHub Repository: [https://github.com/haile2967/ML.git]

API Deployment URL : [https://ml-7-nj3n.onrender.com/]

This report thoroughly documents the machine learning lifecycle followed in this house price prediction project.