Property Price Prediction Engine

This project implements a **house price prediction model** using **XGBoost Regressor** on real estate data. It is designed to estimate property prices based on various features such as location, house size, number of bedrooms and bathrooms, lot size, and more.

Initially, the model achieved an R^2 score of 0.56. To optimize performance, the following techniques were applied:

- Removing irrelevant columns
- Using Target Encoding instead of Label Encoding for categorical features
- Feature Engineering (price_per_sqft, bed_bath_ratio, lot_per_bed)
- Switching from RandomForest to XGBoost Regressor

These improvements increased the R^2 score to 0.8.

Dataset

- The dataset file is `realestatedata.xlsx` and is available in this GitHub repository.
- It contains property listings with columns such as: `price`, `house_size`, `bed`, `bath`, `acre_lot`, `city`, `state`, `zip_code`, `status`, etc.
- Preprocessing steps include:
- Cleaning column names
- Handling missing values
- Dropping irrelevant columns (`street`, `brokered_by`)
- Target encoding categorical features (`city`, `state`, `zip_code`, `status`)
- Feature engineering: `price_per_sqft`, `bed_bath_ratio`, `lot_per_bed`

> **Note:** Ensure the dataset file (`realestatedata.xlsx`) is in the project directory before running the code.

Requirements

- Python >= 3.8
- Libraries:
- pandas
- numpy
- scikit-learn
- xgboost
- matplotlib
- seaborn

Install required libraries using:

```
pip install pandas numpy scikit-learn xgboost matplotlib seaborn
```

Usage

1. Load the dataset:

```
df = pd.read_excel("realestatedata.xlsx")
```

1. Run the script to train the model, evaluate it, and visualize feature importance.

2. Key outputs include:

3. **R^2 Score**: Shows how well the model explains the variance in house prices.

4. **Predicted Price**: Example prediction for a sample house from the test set.
5. **Feature Importance Plot**: Shows which features contribute most to the prediction.

Features

- Handles missing values and categorical encoding
- Feature engineering for better model performance
- Log transformation of target variable to reduce skewness
- Uses XGBoost for robust regression performance
- Visualizes feature importance for model interpretability

Evaluation

The model is evaluated using **R^2 Score**, which measures the proportion of variance in the target explained by the model.

Example output:

```
R² Score: 0.8
Predicted Price: $450,000.00
```

Future Improvements

- Experiment with additional features such as year built, property type, or neighborhood amenities.
- Hyperparameter tuning using GridSearchCV or Optuna.
- Deploy the model as a web API for real-time price prediction.

Author

Karanam Sumanth

- B.E. in Information Technology, 2025
- GitHub: https://github.com/SumanthRaoKaranam