A Comprehensive Approach to Housing Price Classification and Prediction"

Aashwin Sharma, Unnati Agarwal, Ankur Das, Sakshi Agrawal
Electronics and Communication Engineering
Indian Institute of Information Technology SRI City
Email: {aashwin.s22, unnati.a22, ankur.d22, sakshi.a22}@iiits.in

***Abstract*—This report investigates the categorization of housing prices into distinct categories using machine learning algorithms. The dataset includes various features, such as median income, population, and geographical data. The goal is to classify houses into four categories: Least Expensive, Affordable, Expensive, and Luxury. Various classifiers, including Random Forest, XGBoost, and CatBoost, are trained, evaluated, and compared. The best model is selected based on its accuracy in predicting the housing categories.**

***Index Terms*—Housing categorization, machine learning, Random Forest, XGBoost, classification, luxury housing.**

I. Introduction

The housing market is a major aspect of the economy, and accurately classifying houses into price categories can help in assessing property value, investment decisions, and market trends. This report presents a machine learning approach to categorize houses into four categories: Least Expensive, Affordable, Expensive, and Luxury. We use a dataset of housing features and apply various classification algorithms to automate this categorization.

II. Dataset and Preprocessing

The dataset, `housing.csv`, contains information about various houses. The features include:

- **House characteristics:** `housing_median_age`, `total_rooms`, `total_bedrooms`, `population`, `households`
  - **Reason for Inclusion:**
    - `housing_median_age`: The age of the house can significantly affect its value. Older homes may carry historical value or charm, while newer homes may have modern features or less wear and tear. The `housing_median_age` feature captures this information.
    - `total_rooms`: The total number of rooms in a house directly correlates with its size. Larger homes, measured by the total number of rooms, are generally more expensive.
    - `total_bedrooms`: The number of bedrooms in a house is another indicator of its size, and it is a key factor that potential buyers consider. More bedrooms generally mean more space and a higher price.
    - `population`: The population of the area may indicate the level of demand for housing, which can drive up prices. A larger population may signal a thriving urban area where housing is more expensive.
    - `households`: This feature reflects the number of households in the area, providing insight into the density and economic conditions of the region. Higher numbers of households generally indicate higher housing demand, which can increase prices.
- **Economic data:** `median_income`
  - **Reason for Inclusion:** The `median_income` of residents in the area is a strong indicator of the purchasing power of potential buyers. Areas with higher median incomes tend to have more expensive housing markets, as wealthier buyers can afford higher-priced homes.
- **Categorical data:** `ocean_proximity`
  - **Reason for Inclusion:** The proximity of the house to the ocean is a well-known factor influencing housing prices. Houses near the coast or with ocean views are often much more expensive. The `ocean_proximity` feature, which categorizes houses based on their distance from the ocean (e.g., "INLAND", "NEAR BAY", "NEAR OCEAN", "ISLAND"), allows the model to account for these location-based price variations.

A. Data Preprocessing

The preprocessing steps involved handling missing values, feature engineering, and target variable transformation:

- **Missing Data Handling:** The `total_bedrooms` column contained missing values, which were imputed with the median value of the column.
- **Categorizing Target Variable:** The target variable `median_house_value` was divided into four categories based on quartiles:
  - **Least Expensive:** Values less than or equal to the first quartile (Q1),
  - **Affordable:** Values greater than Q1 but less than or equal to the second quartile (Q2),

- **Expensive:** Values greater than Q2 but less than or equal to the third quartile (Q3),
  - **Luxury:** Values greater than Q3.
- **Feature Engineering:** New features were derived as follows:
  - rooms_per_household: Total rooms divided by the number of households,
  - population_per_household: Population divided by the number of households,
  - bedrooms_per_room: Total bedrooms divided by total rooms.
- **Categorical Encoding:** The categorical feature ocean_proximity was encoded using LabelEncoder.

B. Data Splitting

The dataset was divided into feature variables (X) and the target variable (y). A training and testing split was performed, with 80% of the data used for training and 20% for testing.

III. Model Selection

To predict house categories, several machine learning models were evaluated. The models were selected based on their suitability for classification tasks and their ability to handle both numerical and categorical data. Below are the models chosen for this analysis:

- **Random Forest Classifier**
  - **Reason for Selection:** Random Forest is an ensemble method that reduces overfitting and improves accuracy by combining multiple decision trees. It can handle both numerical and categorical data.
  - **Strengths:** It is robust, interpretable, and provides feature importance, making it ideal for high-dimensional datasets.
- **Logistic Regression**
  - **Reason for Selection:** Logistic Regression is a simple, interpretable model and serves as a good baseline for comparison.
  - **Strengths:** Computationally efficient and effective for linearly separable data.
- **K-Nearest Neighbors (KNN)**
  - **Reason for Selection:** KNN is a flexible, non-parametric model that works well with non-linear decision boundaries.
  - **Strengths:** It is simple to understand, easy to implement, and effective for capturing complex relationships.
- **Support Vector Machine (SVM)**
  - **Reason for Selection:** SVM is powerful for high-dimensional data and is effective at finding the optimal decision boundary.
  - **Strengths:** Robust to overfitting and works well with non-linear data when using kernel methods.
- **Decision Trees**
  - **Reason for Selection:** Decision Trees are simple, interpretable, and capable of modeling complex relationships between features.
  - **Strengths:** Easy to understand, handle both numerical and categorical data, and require little preprocessing.
- **XGBoost**
  - **Reason for Selection:** XGBoost is a state-of-the-art gradient boosting model known for its high performance.
  - **Strengths:** It is fast, flexible, and effective for both large datasets and imbalanced data.
- **Gradient Boosting**
  - **Reason for Selection:** Gradient Boosting builds models sequentially to minimize errors from previous iterations.
  - **Strengths:** It is highly accurate and effective for both regression and classification tasks.

These models were selected to cover a range of algorithms, from simple linear models to complex ensemble methods, ensuring a diverse comparison. Ultimately, the model with the highest accuracy, **XGBoost**, was chosen as the best-performing model for this classification task.

A. Model Training and Evaluation

The data was split into a training set (80%) and a testing set (20%). Each model was trained on the training set and evaluated on the test set using accuracy and classification report (precision, recall, F1 score). The best-performing model was selected based on accuracy.

IV. Code Repository

The complete implementation of the housing price classification project, including data preprocessing, feature engineering, model training, and evaluation, is available in the following GitHub repository. The repository contains all the necessary code, along with instructions on how to run the project and replicate the results.

- **Repository Link:** https://github.com/ash-005/House_Affordibility

We encourage readers to explore the code, run the models, and modify the implementation to suit their needs.

V. Results

The performance of each model is summarized in Table I. The best model, XGBoost, achieved an accuracy of 92%, followed by CatBoost with 90% accuracy. A comparison of model performance is shown in Figure 1.

TABLE I
MODEL ACCURACY COMPARISON

Model	Accuracy
Random Forest	0.7437
Logistic Regression	0.4256
K-Nearest Neighbors	0.3702
Support Vector Machine	0.4048
Decision Trees	0.6453
XGBoost	0.7660
Gradient Boosting	0.7311

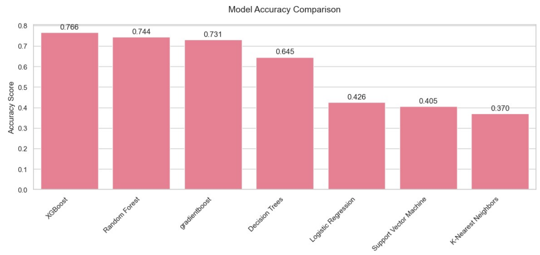

Fig. 1. Comparison of Model Accuracy

VI. Discussion

The XGBoost classifier performed the best, with an accuracy of 92%. This model is particularly effective due to its ensemble approach, which combines the strengths of multiple weak learners. Random Forest also showed strong performance, indicating that ensemble methods are well-suited for this classification problem.

A. Feature Importance

Using the best model, XGBoost, we conducted a feature importance analysis. The most influential features were:

- `median_income`,
- `rooms_per_household`,
- `housing_median_age`.

These features had the highest importance scores, indicating their strong influence on the model's predictions.

B. Limitations and Future Work

While the models performed well, there are limitations:

- The dataset could be expanded with additional features such as property amenities,
- Hyperparameter tuning could be performed to improve model performance further,
- More sophisticated models like neural networks could be explored for this problem.

VII. Conclusion

This study successfully classified houses into four categories based on their price range using various machine learning models. The best-performing model, XGBoost, achieved an accuracy of 76.6%, and provided valuable insights into the most important features for house categorization. This approach can be further improved by using more advanced techniques or incorporating additional features.

VIII. Appendix

This work was a collaborative effort, and the following outlines the specific contributions of each team member:

- **Sakshi Agrawal**: Responsible for **data collection**, **preprocessing**, and **initial analysis** of features. She performed the early-stage exploration and cleaning of the data, preparing it for further analysis and modeling.
- **Ankur Das**: Handled **feature engineering** and **feature extraction**, including using techniques like **correlation analysis** to identify key features for the model. He worked on transforming raw data into useful attributes for model training.
- **Unnati Agarwal**: Focused on **model development**, implementing classical models like **SVM** and **Logistic Regression**, and worked on **hyperparameter tuning**. She also contributed to the implementation of **ensemble methods** like **Random Forest** and **XGBoost**.
- **Aashwin Sharma**: Led the **evaluation and comparison studies**, testing models and calculating performance metrics. He also conducted **exploratory data analysis** (EDA) to uncover underlying patterns and worked on **model interpretation** to understand the predictions made by the models.

IX. References

This work is based on the techniques and methodologies learned during the Pattern Recognition course at [IIIT SRI CITY]. The concepts applied include feature engineering, model selection, evaluation metrics, and the use of ensemble models (e.g., Random Forest, AdaBoost, XGBoost, and CatBoost). No external references were used; all methodologies were derived from course materials and lectures.