1. Introduction

This project involves the analysis and modeling of property prices using a dataset containing various features related to housing in a specific region. The goal is to create predictive models for property prices and explore the relationships between different variables. The code is written in Python and utilizes various libraries such as Pandas, NumPy, Seaborn, Scikit-learn, Matplotlib, Pycaret, and XGBoost.

2. Data Loading and Cleaning

The initial steps involve loading the dataset and cleaning it to prepare for analysis. The dataset is loaded from a CSV file, and preliminary exploration is conducted. Duplicate values are removed, and missing values in the 'mrtdist' column are addressed by dropping the entire column. Additionally, several irrelevant columns are dropped to focus on relevant features. A scatter plot is created to visualize the geographical distribution of property prices.

3. Exploratory Data Analysis (EDA)

The exploratory data analysis section involves visualizing the data to gain insights into its distribution and relationships. Histograms, pair plots, correlation heatmaps, and bar plots are used to analyze numerical and categorical features. These visualizations provide a better understanding of the dataset's characteristics and help identify potential patterns.

4. Data Transformation

To prepare the data for modeling, log transformation is applied to the resale price column. This transformation is beneficial for linear models, enhancing their predictive performance.

5. Model Building and Evaluation

The main focus of the project is to build regression models to predict property prices. The Pycaret library is utilized for an automated setup of the regression task. Several regression algorithms, including Random Forest, Extra Trees, Gradient Boosting, Linear Regression, K-Nearest Neighbors, and XGBoost, are employed. The models are evaluated using various metrics such as R-squared,

Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

6. Hyperparameter Tuning

Hyperparameter tuning is performed on selected models, specifically Random Forest and XGBoost, using GridSearchCV. This process involves searching for the best combination of hyperparameters to optimize the model's performance.

7. Model Pipelines

Pipeline structures are implemented for Random Forest, Extra Trees, Decision Tree, Linear Regression, K-Nearest Neighbors, and XGBoost models. These pipelines include data preprocessing steps such as label encoding and standard scaling, followed by model training and evaluation.

8. Feature Importance

Feature importance is analyzed for the best-performing model using Random Forest. The importance of each feature is visualized to understand which variables contribute the most to predicting property prices.

9. Model Tuning with Ridge Regression

Ridge Regression, a linear model with regularization, is employed with hyperparameter tuning using GridSearchCV. This step involves standardizing the features and searching for the best alpha value to optimize the model's performance.

10. Final Model Evaluation

The final models are evaluated on the test set using various metrics to assess their predictive capabilities. The R-squared value, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error are reported for each model.

11. Conclusion

The documentation concludes with a summary of the entire process, including the initial data exploration, cleaning, feature engineering, model building, hyperparameter tuning, and final evaluation. Recommendations for future

improvements or additional analyses may be suggested based on the outcomes of the project.