This project aims to predict the median house value in California using various regression models. Here's an overview of the process and what each part of the code does:

Importing Libraries and Defining Functions

We start by importing libraries for data manipulation (pandas, numpy), visualization (matplotlib), machine learning (sklearn), and statistical analysis (scipy). The Plot_features function plots feature importance for a given list of columns and their importance scores.

Data Loading and Preprocessing

The dataset is loaded and several new features are created:

- Rooms_per_Household
- Bedrooms_per_Household
- Population_Density
- Income_per_Person
- bedrooms_per_room

The data is then sorted by Median_Income and columns are reordered for consistency.

Feature Scaling and Balancing

Features (X) and the target (y) are separated. RandomOverSampler is used to balance the data, and StandardScaler normalizes the features. A new DataFrame is created with the normalized data.

Outlier Removal

Outliers are identified using z-scores and removed to ensure the data is clean for model training.

Linear Regression Model

A Linear Regression model is trained and evaluated using mean squared error (MSE) and mean absolute error (MAE).

Random Forest Regressor Model

A Random Forest Regressor model is trained with hyperparameter tuning using GridSearchCV to find the best model parameters.

Decision Tree Regressor Model

A Decision Tree Regressor model is trained with hyperparameter tuning using RandomizedSearchCV.

Model Comparison

The scores of the three models are compared and visualized in a bar chart to determine which model performs the best.

In summary, this code provides a comprehensive approach to predicting house values by creating new features, balancing and normalizing the data, removing outliers, and employing multiple machine learning models with hyperparameter tuning. The final step compares the model performances to select the best one for predictions.