CS-559-WS Group 12 Report

Github: https://github.com/ShaunMendes/house_price_prediction.git

Training Process

Exploratory Data Analysis:

1. **Handling Missing Values:** Features with a high percentage of missing values, strong correlations with other features, or low/zero variance were carefully dropped. For numeric features with missing values, we imputed them using the average value, while for categorical features, we filled in missing values with the most common category.
2. **Categorical Encoding:** We transformed categorical features into integers using scikit-learn's LabelEncoder, making them suitable for machine learning algorithms.
3. **Standardization:** To ensure consistent scaling, both the features and target prices were standardized using scikit-learn's StandardScaler. This step enhances the performance of various machine learning models.
4. **Dimension Reduction:** To reduce complexity and improve model efficiency, we applied Principal Component Analysis (PCA), reducing the feature space to 35 principal components.
5. **Target Variable Standardization:** After experimentation, we discovered that standardizing the target variable significantly improved numerical stability for k-means clustering and for downstream training stability. While calculating the Mean Squared Error (MSE), we destandardized the prices before evaluation.

All the transformation artifacts were saved to disk (trained_models directory) to ensure replication on the test set.

Model Building:

Once the data was preprocessed, kmeans was used to cluster the data. The kmeans cluster ids were used as ground truth labels for the classifier. Using the elbow method and a few experiments we determined that 3 clusters(groups) provided substantial training data for individual models and created good clusters. Based on grid search, we decided on a SVC classifier for partitioning the data.

The classifier's predictions were used to determine which model each sample belonged to. This process created 3 mini datasets, that would be used to train k models. For example, if the classifier predicted that a sample belonged to group 0, then that sample and its ground truth would contribute to training model 0.

Once the data was split into groups, each dataset was split into train, validation, and test sets. For each group we trained 4 models. The top 4 models were selected using optuna. Once the best hyperparameters were found, the final models were stacked and then the stacked model was trained by merging the validation and train datasets.

Cluster Classifier:

- Training Accuracy: 0.994
- Test Accuracy: 0.959

Note: We populated the test accuracy by comparing the result between cluster created by kmean as ground truth vs the ones created by SVC.

Group 0 Regressors:

- Ridge
- SVR
- MLPRegressor
- GradientBoosting

Stacked Model MSE on train data of group 0: 183,417,953

Group 1 Regressor:

- Ridge
- SVR
- MLPRegressor
- GradientBoosting

Stacked Model MSE on train data of group 1: 165,861,159

Group 2 Regressors:

- LinearRegression
- Ridge
- SVR
- MLPRegressor

Stacked Model MSE on train data of group 2: 317,436,738

Test Process

Test predictions were generated by running the preprocessed test data through the trained models.

The test data was preprocessed using the saved artifacts generated by the training step. The same features were dropped. Any nans were replaced with the values selected during the training process. Labels were encoded using the LabelEncoders that was fit during training. The Data was standardized using the StandardScaler that was fit during training. The remaining feature count was reduced to 35, using the PCA model that was fit during training.

Once the test data was preprocessed, the classifier was used to assign each sample to a cluster. Using the classifier predictions, the samples were split up into their assigned groups.

Each group of data was then passed into the group’s respective model for inference. Since the prices were standardized, the StandardScaler’s inverse_transform function was used to destandardize the predicted and ground truth prices. With the destandardized prices, the mse for the test data was then calculated.

Results:

MSE on test data for group 0: 425,196,806

MSE on test data for group 1: 1,356,699,627

MSE on test data for group 2: 370,421,718

Aggregate test MSE = 748,841,426

Aggregate test RMSE = 27,364.967