Group Project 1 Report

By Isaac Aguilar, Jeffrey Troll, & Sam Lee

Once we had gathered as a group, we compared the feature engineering that each of us did and tried selecting the best feature engineering since all of us used K-Nearest Neighbors as our models for our individual projects. We all had similar ideas about cleaning the features, extracting some other features, and doing feature selection. However, after conducting many tests with all kinds of really cool features (like one that measured the neighborhood price based on weighted distances).

These were the preprocessing steps we came up with:

* Create columns for the most important amenities.
* Convert amenities into a new column that has the number of amenities in the place.
* Create year columns from date-related variables and remove old columns.
* Extract city and state and remove the host\_location variable.
* Convert text percentages into decimals.
* Convert the list of verifications to the number of verifications.
* Extract bathroom and bathroom type from the bathroom column.
* Impute missing values with the mean for numerical and the mode for categorical variables.
* Standardize the scale for the numerical values.
* Create dummy variables for the categorical variables.
* Do feature selection with correlated feature variables and with the Select Percentile function.

We also tried removing, capping, and upsampling outliers, but none of that worked in the end. We noticed that doing more drastic feature selection improved our test sample predictions' accuracy, but it did not help with the actual Test predictions we submitted to Kaggle. We also noticed that tuning hyperparameters with cross-validation was more effective when using the Mean Absolute Error instead of the Mean Squared Error.

These are all the models that we tried:

|  |  |
| --- | --- |
| * KNN * Linear Regression (No penalization) * Ridge * Lasso * SVM | * Random Forest * XGBoost * Stack (KNN & RF) * Stack (KNN & XGB) * Stack (SVM & XGB) |

We found that improving our accuracy for the test sample data would not guarantee an improvement in the Test data submitted to Kaggle. So, we all tried different models with different features to cover as much ground as we could and reported our findings through text.

In the end, the Stack model using K-Nearest Neighbors and Random Forest as our base-learners model and using a random forest as our meta-learner model gave us our result, which was a MAE of 111.5 in Kaggle, improving all our individual submission scores by far.

As previously mentioned, this model did worse than others in the out-of-sample MAE but performed better in the actual Test data. The out-of-sample MAE was 127.62. We got models that were around the 90s for the out-of-sample MAE with worse scores for the Test Data.

The following are the features that seem to be most important for the prediction:

* City\_Victoria
* minimum\_maximum\_nights
* availability\_365
* availability\_60
* property\_type\_Room in a boutique hotel