DSC 630
Laura Hoffmann
Assignment 5.2

Optimal Hotel Recommendations Report

Exploratory Analysis

While conducting exploratory analysis, I developed a histogram to check the distribution of the hotel_cluster variable, because that is the target variable for this project. Afterward, I created functions to help extract both months and years from the columns containing times and created separate columns for check in months and years and check out months and years. Later on correlations between the predicting variables were checked and most of the time columns were highly correlated (because most people check in and out of hotels within the same month/year) so most of those variables were removed with the exception of check in month and check in year.

Following this I created an aggregate pivot table that mapped each of the hotel clusters and prepared them to merge with the original data. Because we were only interested in the instances of actual bookings, all other data was dropped from the original data frame and the result was then merged with the destination.csv file and the aggregate pivot table. This merged data table is what I wanted to divide into the training and testing sets in order to build the predictive models.

Modeling

For each of the models I built I used them alongside sklearn's preprocessing.standardscaler which standardizes the features by removing the mean and scaling to unit variance.

Originally, I attempted Naive Bayes, the Guassian Naïve Bayes classifier because it was the easiest to work with, and the target variable seemed independent from most of the predictors in terms of correlation. However, this model had the worst accuracy with around 13%. Next, I tried K-Nearest Neighbor classifier, which should've been able to group like users, assigning them to clusters which should be successful in predicting the hotel cluster target variable. With the knn classifier I used 3 as the number of neighbors. This one performed slightly better with around a 32% accuracy.

After that I attempted Random Forest Classifier with n_estimators = 150, max_depth = 10, and random_state = 0. I used random forest to build such decision trees that should create a more stable prediction. This performed only slightly worse than the knn classifier around 30%. Last I fit a Support Vector Machine or SVM model. This one is a supervised model used for classification, regression and detecting outliers. With a specified shape of 'ovo' I was able to accomplish a better accuracy of about 40%

I'm glad that the purpose was not to get great results because these were definitely not the best. Working with such a large quantity of data like this with predictors that are not highly correlated made this a challenge. While the results were not great, I was able to gain a little bit of a better understanding of each of the models used and why to use them.