PREDICT HOTEL BOOKING CANCELLATIONS

Marjorie Kinney	**m20210647**
Bruno Mendes	**m20210627**
Lucas Neves	**m20211020**
Farina Pontejos	**m20210649**

Business Cases in Data Science Fall 2022

NOVA Information Management School

Predicting Hotel Booking Cancellations

Business Case 2: Classification

Apex Pattern Deployers:

Marjorie Kinney	*m20210647*
Bruno Mendes	*m20210627*
Lucas Neves	*m20211020*
Farina Pontejos	*m20210649*

Business Cases for Data Science

NOVA Information Management School

March 2022

Introduction

Online travel agencies have gained an increased role in the hotel industry in the past 25 years. While these agencies provide high market exposure, they also charge a commission to use and have brought with them increased competition. To stay viable most hotels make their properties available to these travel agencies, which means offering their lowest price and generous cancellation policies. Deal seekers often book several hotels, and cancel those that are not the best. In order to maximize revenue, Hotel Chain C, like other hotel chains, implemented an overbooking policy that consists in allowing more bookings than there are available rooms for a certain time-period. The reason being that there is always a certain percentage of customers that cancel their bookings and open a spot for another customer. If there were no overbooking policy, more rooms would be left unused and, adding to the potentially missed revenue, the fixed costs for those rooms remain since they have to be maintained whether or not they are utilized.

The overbooking policy can be more or less aggressive, depending on the particular hotel's circumstances, but it always incurs a risk nonetheless. If the policy is too aggressive, there may be more customers than there are rooms at the time of check-in, in which case the hotel will suffer damage to its reputation and must pay to relocate these customers to another hotel.

Business Understanding

Hotel Chain C's H2, the subject of this project, is affected by high cancellation rates, representing 42% of all bookings and 43% of potential revenue. Taking that into account, the company pursued an aggressive overbooking policy which generated unacceptable costs.
To address this issue, this project aims to conduct a predictive analysis to determine which reservations are most likely to be canceled. In this way, the company can either reach out to individuals with incentives to prevent cancellation, or implement the right amount of overbooking to maximize room revenue.

Data Understanding

We used various statistical and visualization techniques to explore and better understand the nature of the available data. These techniques include a time series analysis to determine if seasonality exists in the data, as well as histograms and boxplots to analyze the distribution of the data and to detect potential outliers. See the appendix for visualizations of the data prior to preprocessing.

Data Preparation

Splitting the data
We split the data in two, one set to train the models and one to validate them. The split was done by year: data from 2015 and 2016 were used for training and 2017 for testing. We recognize that there may be no need for taking time into account in this instance, since it does not reflect changes in a certain subject over-time nor do we use exogenous variables that change over-time such as GDP. The reasoning for this split is one of practicality, we can more easily appraise the value of a real world implementation by simulating the situation - we use past data to estimate current and future bookings and evaluate accordingly.

Removal of duplicates
In examining the data, we also observed that there was a significant amount of duplicates that were removed from our train/test sets so that the model performance could be improved and no extra weight would be given to the same information. Nevertheless, the duplicates were kept in the validation set as it reflects reality and the actual reservations that occured.

Data transformation
Simple data transformations were applied to the data to allow for modeling. The categorical features were one hot encoded so that they could be used in the algorithms. While scaling was appropriate for some models, it was not appropriate for other models. The scaling of data was done only when needed, and just prior to modeling. Depending on the performance of each model, the data was not scaled at all, scaled using the standard scaler, or scaled using the MinMax scaler. Null values were only present in the features of Country, Company, and Children. Although the missing values were imputed with the mode for Country and Children respectively, all three features were eventually dropped. Trailing whitespace was removed if present in categorical features. We used an oversampling technique to account for the imbalances in the dataset. SMOTENC was chosen because it works for numerical and categorical data. The data was oversampled based on cancellation status.

Feature engineering
A categorical feature of season was added to the dataset based on the reservation arrival dates. The month, year, week number, day of month, month number, and date of week were also added for each reservation arrival date.

Feature selection
Several methods were employed for feature selection. Spearman and Pearson correlation heatmaps were created to compare the various correlations of the numeric features. ArrivalDateWeekNumber and ArrivalDateMonth were highly correlated, of course. Since ArrivalDateWeekNumber is more granular, it was kept, while ArrivalDateMonth was removed. See appendix for heatmaps.

In the hotel industry it is quite common for customers to change their booking's attributes, like the number of persons, stay duration, or room type preferences, either at the time of

their check-in or during their stay. It is also common for hotels not to know the correct nationality of the customer until the moment of check-in. Therefore, we removed ‘Adults’, ‘Children’, ‘Babies’, ‘Meal’, ‘Country’, ‘AssignedRoomType’, ‘ReservationStatusDate’, and ‘ReservationStatus’, while keeping the variables in Table 1. We consider it unlikely that having the wrong country or the incorrect number of children listed is the reason for canceling.

NumFeat	'LeadTime', 'StaysInWeekendNights', 'StaysInWeekNights', 'PreviousCancellations', 'PreviousBookingsNotCanceled', 'BookingChanges', 'DaysInWaitingList', 'ADR', 'RequiredCarParkingSpaces', 'TotalOfSpecialRequests'
CatFeat	'ArrivalDateMonth', 'MarketSegment', 'DistributionChannel', 'ReservedRoomType', 'DepositType', 'CustomerType', 'IsRepeatedGuest

Table 1 - Features used

Pair plots were created to determine if any significant relationships exist between the remaining numeric features and IsCanceled. SelectKBest was run to rank the features by their importance. No features were removed as a result of these investigations.

Success criteria

The business objectives in this case require that uncertainty about demand is diminished significantly. Forecasting net demand based on the reservations should allow the company to identify bookings with a high likelihood of cancellation, and use incentives to reduce those cancellations from approximately 42% to 20%. While accuracy is often used as a success criteria for a machine learning algorithm, in this case accuracy does not provide a comprehensive measure. Instead, F1 score was chosen as the best measure of success of the model. While accuracy measures the number of correct predictions divided by the total number of predictions, the F1 score takes into account the precision and the recall. In other words, it takes into account true and false predictions and outcomes.

In addition to providing the F1 score, the confusion matrix for the final model will also be provided. This will allow for a clear representation of true positives, false positives, true negatives, and false negatives. Finally, predictions for the final model will also be reported as a likelihood of cancelation. This will allow the business to use their judgment on a case-by-case basis as to what should be done to either incentivize the guest to fulfill their reservation, or to overbook.

Modeling

We tested several industry-standard machine learning algorithms that are fit for binary classification problems such as the present case. They were:

- Gradient Boosting
- XGBoost
- Random Forest
- Decision Tree
- k-Nearest Neighbors
- Logistic Regression
- Neural Network

To leverage the ‘wisdom of the crowd’, we also used the Voting Classifier as an ensemble method that takes into account the results from the previous other models.

In each case, several combinations of the available parameters were tested and analyzed using GridSearchCV. Simple K-fold cross validation was implemented on the test data to ensure the F1 estimates of the models were accurate. Refer to appendix for details about model parameters.

Evaluation

The F1 scores of each trained model were compared to determine the best solution. Finally, a confusion matrix for the final solution was constructed. The Trained Voting Classifier outperformed all other models. See the appendix for details.

Deployment

With the Voting Classifier trained on the available booking data from July 2015 to December 2016, we managed to correctly predict the outcome of roughly 75% of all bookings from January 2017 to August 2017 - this encompasses 73% of all cancellations, and 77% of non-cancellations.

Accordingly, 25% of the bookings’ outcomes were incorrectly predicted, in which there are two distinct cases:

1. The model predicted the customer would cancel but they did not, which accounts for about 46% of the incorrect predictions (11.5% of the overall bookings) and might entail known relocation costs;
2. The customer was predicted to not cancel but they did, which accounts for the remaining 54% (13.5% of the overall bookings).

To deal with this error, since no classification model is perfect, we recommend risk-mitigation strategies such as less aggressive overbooking during high seasons and more aggressive overbooking during low seasons. The reasoning here is that because there is higher demand in high season, canceled bookings will more easily be filled without overbooking. Although the company has stated a preference for solely reaching out to customers over using overbooking, we recommend a mixture of the two strategies.

Furthermore, to lower cancellation rates, since the mean days prior to arrival for cancellation is 49, with the median days prior to arrival for cancellation being 24, the company should reach out both 60 days and 30 days prior to arrival with incentives. Particularly, customers with a lower yet considerable probability of cancellation (eg. 50%) should be prioritized to be contacted more than those with a higher probability of cancellation (eg. 80%). In this way, a concentration of effort with those less likely to cancel might lead to better results, since it’s a question of resource availability for the task.

Additionally, the threshold for considering a booking as predicted to cancel is currently 50% or higher, but that can be altered. If the threshold is set higher, the model will be more careful

and will only predict a booking as a cancellation when it has a higher certainty. The drawback in this instance is that fewer predictions of cancellations will be made, meaning that there will be more cases where the model predicts a non-cancellation but a cancellation occurs (False Negative). The trade-off is that there will be less error among the cancellation predictions that are made. If the threshold is set lower, the contrary is the case. This is a matter for future discussion with the management - what is more important to be mitigated, the risk of having to relocate a customer or foregoing potential revenue? For example, the company may wish to set the threshold higher in the high season, so that fewer cancellations are predicted, and consequently the overbooking policy based on it would be less aggressive. The reverse could be done in low season. In the high season vacancies are not as big of an issue as in low seasons, since they can be more easily filled due to the increased demand.

Group bookings are more likely to cancel. To address this in the near-term the company may wish to implement stricter cancellation policies for group bookings. In the long-term a study should be conducted on the patterns of group bookings and their usual procedure. Group bookings are inherently harder to cancel for cheaper alternatives, at least during high seasons, since there's the question of the high number of rooms involved. The hotel can use this to their advantage.

In addition, since most guests cancel the day of arrival, the company may wish to implement a cancellation fee for reservations canceled within a short window, for example within 48 hours or 1 week of arrival.

Through a combination of overbooking and incentives, we estimate that if this model had been implemented in January 2017 until August 2017, the company could have increased their revenue by more than €650,000. This assumes that approximately 50% of predicted cancellations could be contacted with incentives, and of those, 50% decide to not cancel. Given that the average stay is worth €318, these 2,104 additional customers equate to an additional €669,072 revenue. If projected for the whole year, this equates to an annual increase in revenue of €1,003,608. This does not include any additional revenue obtained from overbooking likely cancellations.

Because new data will add additional insights, we recommend that the model be retrained if it begins to underperform, if a large unexpected event occurs (such as the COVID pandemic), if a large number of bookings occur, or after a set period of time (for example, a planned retraining each month).

In sum, this model provides significant predictive ability for cancellation which could help the company manage day-to-day resources to focus entirely on interacting with select customers in order to reduce cancellations. In the same way, it could assist in defining more robust overbooking and cancellation policies.

Appendix

Figure 1 - Plots of Numeric Variables by Class

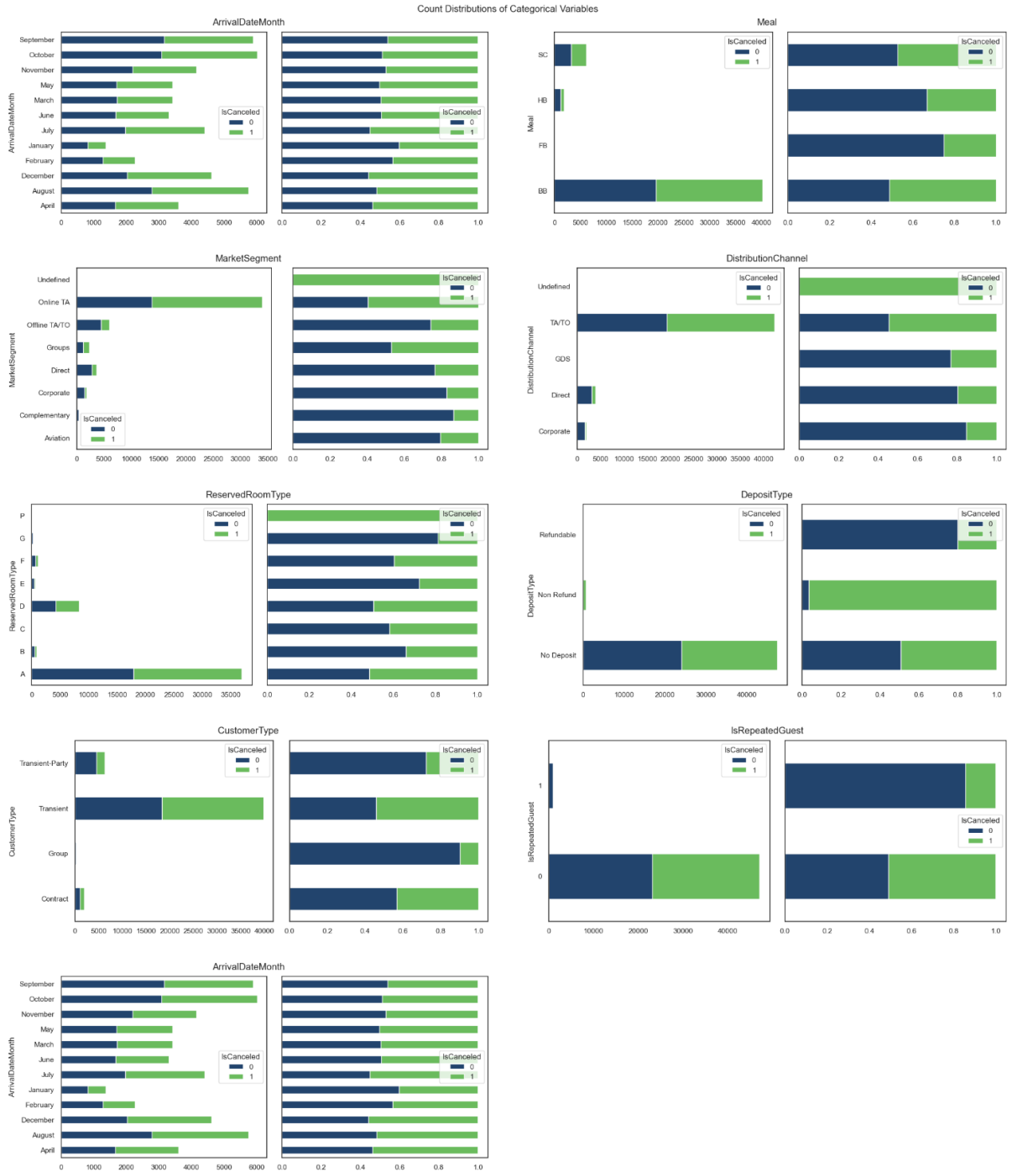

Figure 2 - Plots of Categorical Variables by Class

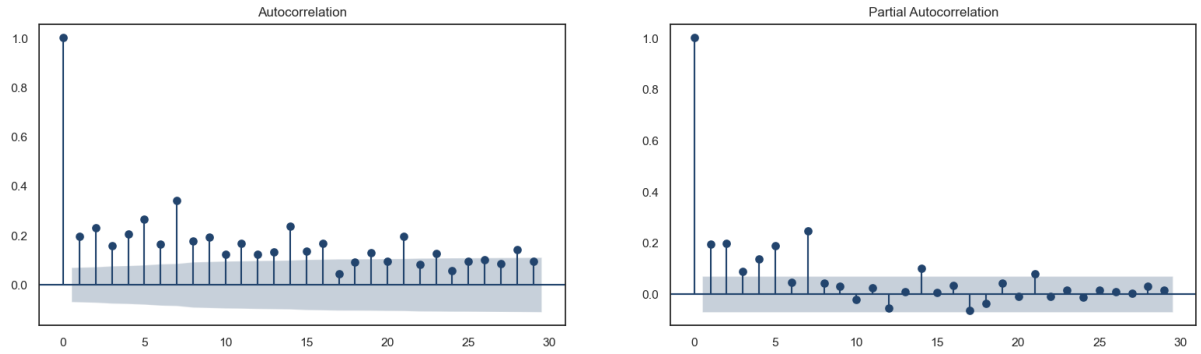

Figure 3 - Time Series Plots Showing 7-Day Seasonality

Figure 4 - Pearson and Spearman Correlation Plots for Numeric Features

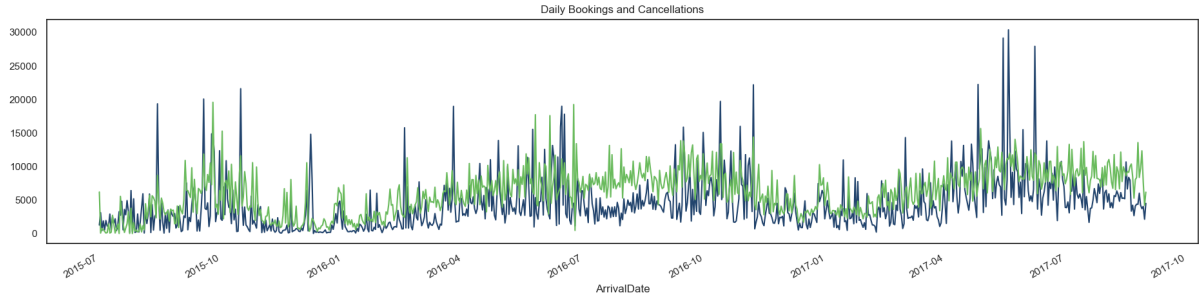

Figure 5 - Daily Bookings and Cancellations by Arrival Date

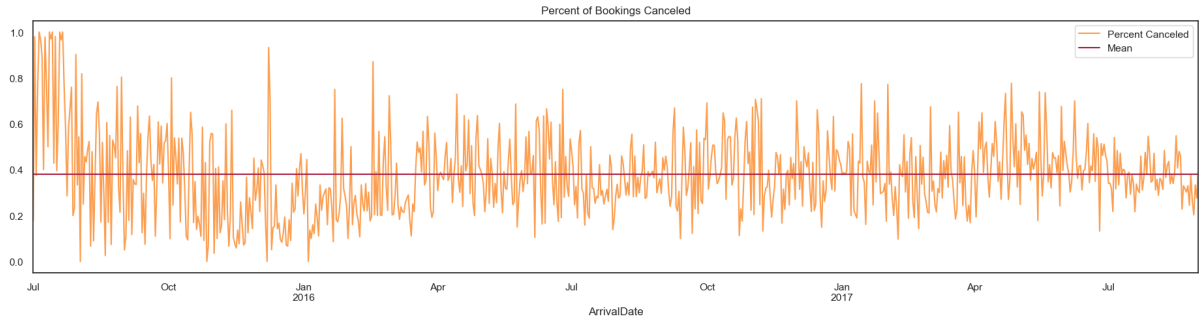

Figure 6 - Percent of Cancellations by Arrival Date

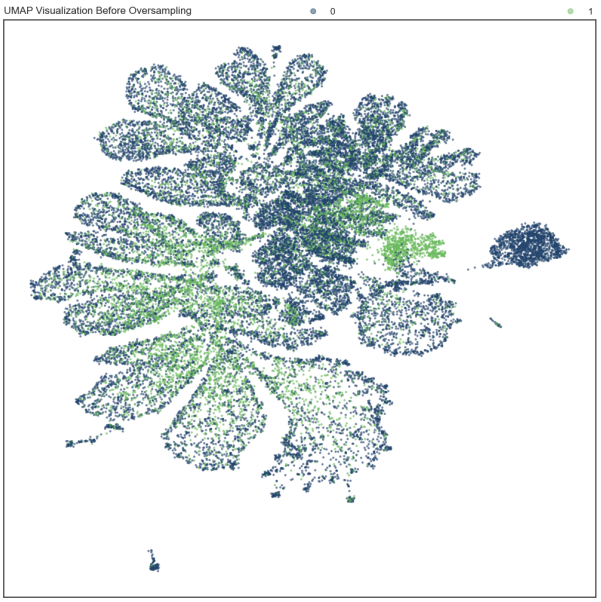

Figure 7 - UMAP Visualization Before Oversampling

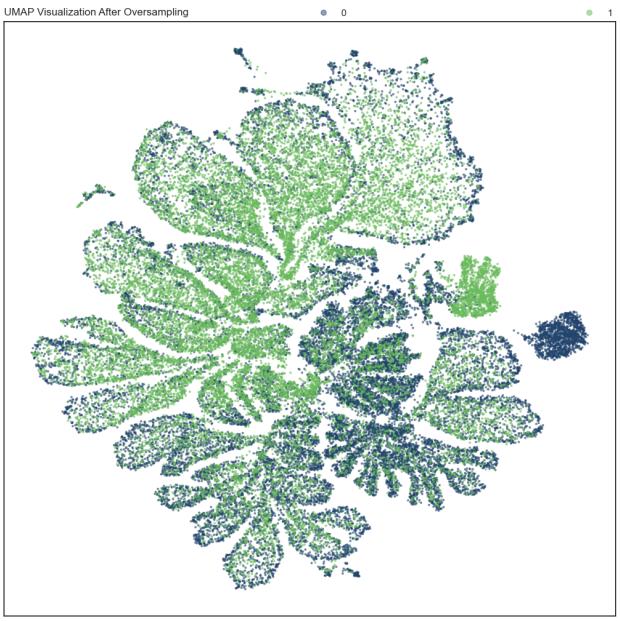

Figure 8 - UMAP Visualization After Oversampling

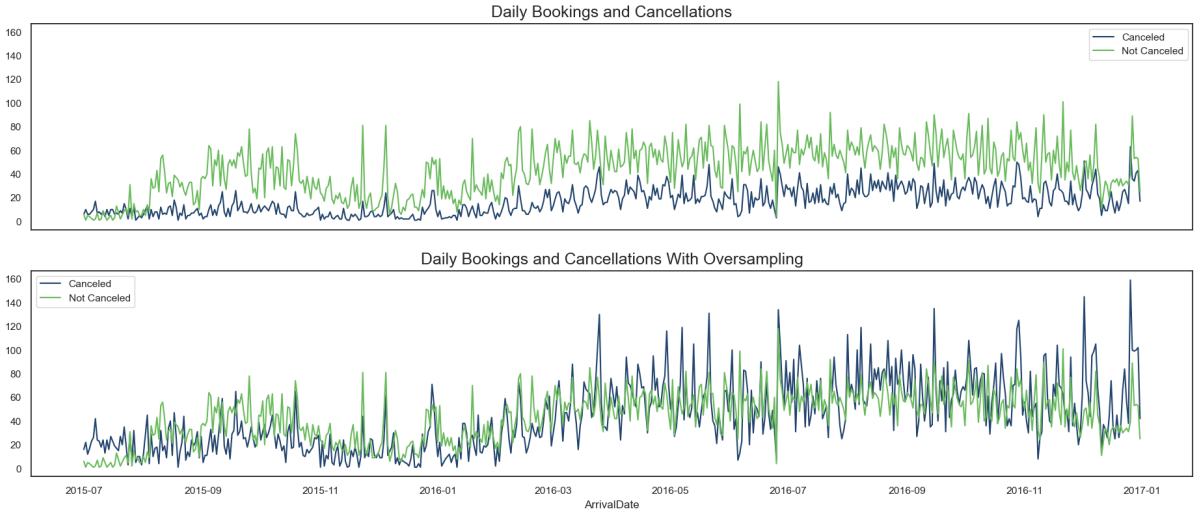

Figure 9 - Daily Bookings Before and After Oversampling

Gradient Boosting	n_estimators=100 max_features='log2' loss='exponential' learning_rate=0.2
XGBoost	subsample= 0.6, max_depth= 7 , eta= 0.1, colsample_bytree= 0.8, colsample_bynode= 0.9, colsample_bylevel= 0.6
Random Forest	'ccp_alpha'= 2e-05, 'criterion'= 'gini', 'max_depth'= None, 'min_samples_split'= 6, 'n_estimators'= 81,
Decision Tree	criterion= 'entropy', max_depth= 15, max_features= None, min_impurity_decrease= 0, min_samples_leaf= 1, min_samples_split= 11, min_weight_fraction_leaf= 0, ccp_alpha= 0.00014, splitter= 'best'
k-Nearest Neighbors	'metric': 'manhattan', 'n_neighbors': 10, 'weights': 'uniform'
Logistic Regression	'C': 10.0, 'penalty': '12', 'solver': 'saga'
Neural Network	'selectkbest__k'= 30, 'hidden_layer_sizes'= (10, 10, 10, 10), 'activation'= 'identity'

Table 2 - Model Parameters

Model	F1 Scores
VotingClassifierFitted	0.713764735815452
MLPClassifier	0.700423251660001
GradientBoostingClassifier	0.698295478940363
LogisticRegression	0.697858083044155
RandomForestClassifier	0.696275988119717
XGBClassifier	0.695974623598267
DecisionTreeClassifier	0.65784
KNeighborsClassifier	0.637708649468892
GaussianNB	0.555555555555556

Table 3 - Validation Scores of Different Models

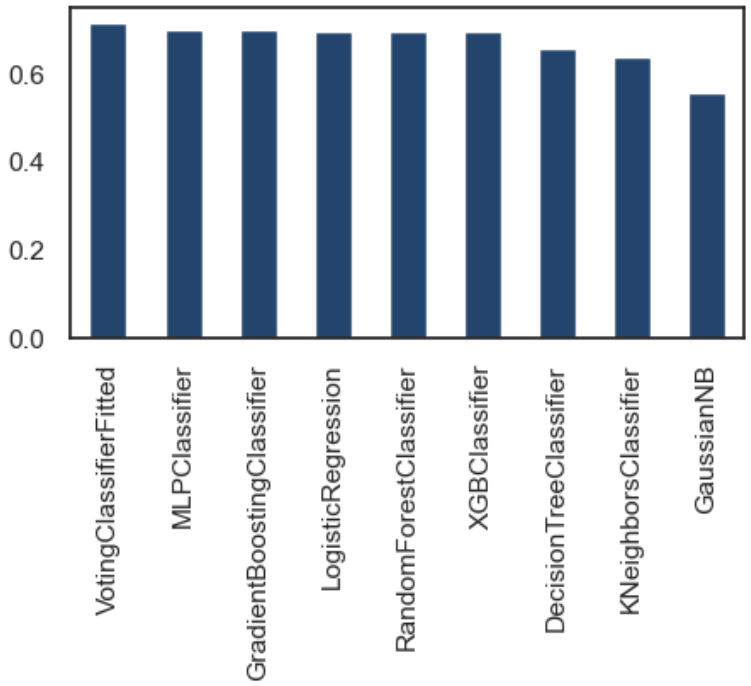

Figure 10 - Validation Scores of Different Models

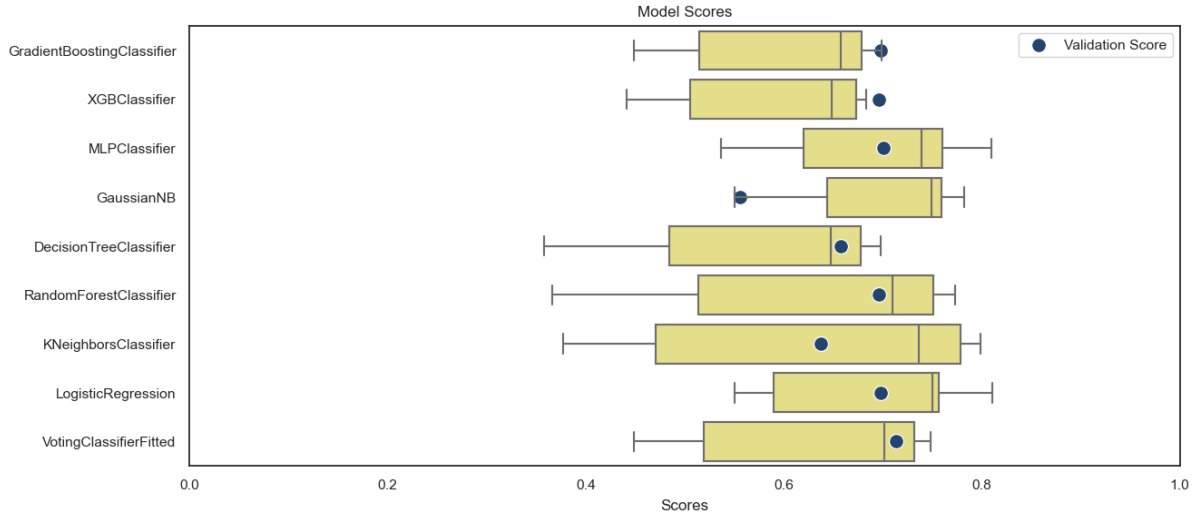

Figure 11 - Cross Validation Scores and Final Validation Score for the Models

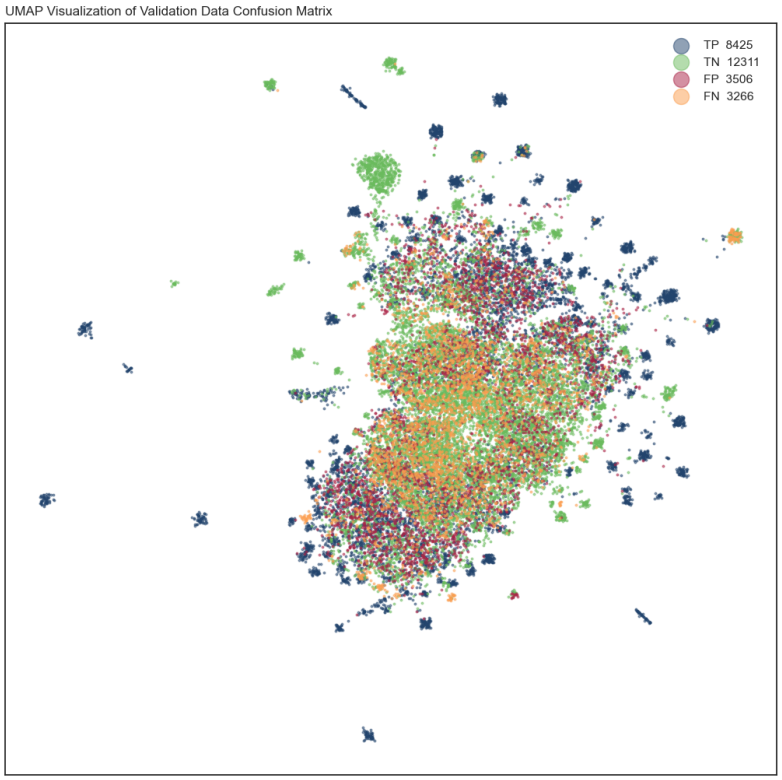

Figure 12 - UMAP Visualization of TP, FP, TN, and FN for Fitted Voting Classifier Model

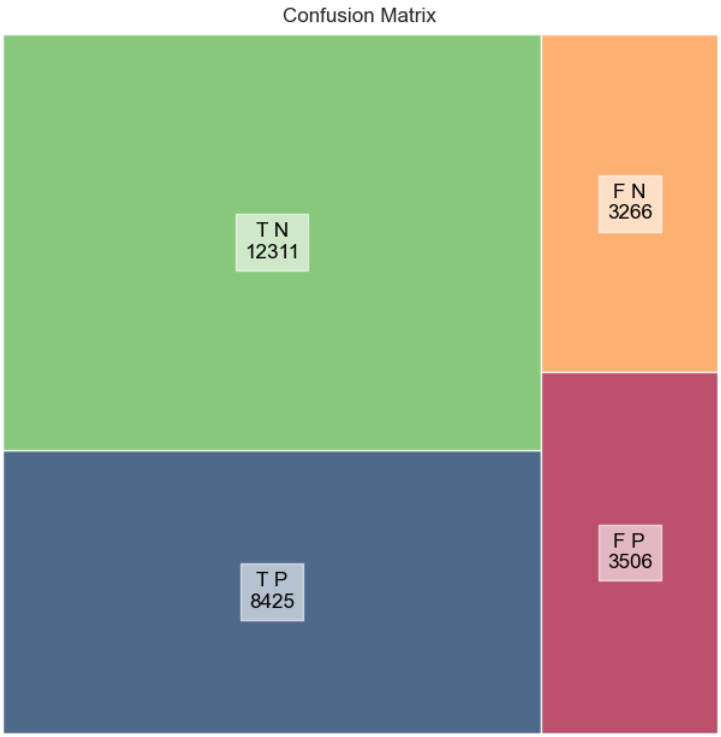

Figure 13 - Confusion Matrix for Fitted Voting Classifier