ST537 Final Project

Mana Azizsoltani & Chris Comora

08 November, 2020

Contents

Introduction **2**

Methods **4**

Results **8**

Conclusion **8**

References **10**

Introduction

In the hospitality industry, there is currently a new push towards the implementation of different sorts of analytics solutions, especially in the realm of revenue management and optimization. Hotel rooms are a limited commodity, and once a room on a certain date is booked in advance, the hotel must guarantee the room to that customer. The problem is that the hotel takes on risk when they book the room, because the customer could cancel with relatively short notice, which leaves the hotel either with a vacant room or with no other choice than to downsell the room last minute. The impact on the bottom-line can be significant; research has shown that cancellations impacted almost 40% of on-the-books revenue in 2018 for some hotels.

In an attempt to solve the problem of revenue loss for canceled rooms, many hotels implement strict cancellation policies or overbooking. With the new push for analytics solutions, we look towards machine learning as a solution to the problem. For this project we will be assessing the accuracy, sensitivity, and specificity of various machine learning techniques when faced with predicting hotel cancellations. We will also compare and contrast the different models, looking at the advantages and disadvantages of each. Essentially, these machine learning models will be attempting to classify a given hotel booking as either canceled or not canceled. We will be running the following five binary classification models:

- Basic Classification Tree
- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Logistic Regression

The data sets used come from two different hotels located in Southern Portugal. The first hotel is a resort hotel located along the coast and the second one is located within a city. They have 31 variables that correspond to different booking information. The first hotel has about 40,000 records, while the second has just under 80,000. Each record corresponds to a booking from the period between July 1, 2015 and August 31, 2017. The data are in .csv format. No-shows are considered cancellations. Because of limitations on our computational power, we used the first 5,000 records of each data set.

Required Libraries

To run the code for the project, the following libraries are required:

- `caret`: to do the heavy lifting of training and tuning the models
- `tidyverse`: for all the data reading and wrangling
- `knitr`: for rmarkdown table outputs
- `rmarkdown`: for output documents
- `rpart`: fitting the basic classification tree
- `rpart.plot`: visualization of the classification tree
- `randomForest`: fitting random forest models
- `kernlab`: fitting the support vector machine
- `class`: fitting the k-nearest neighbor model

Variable Descriptions

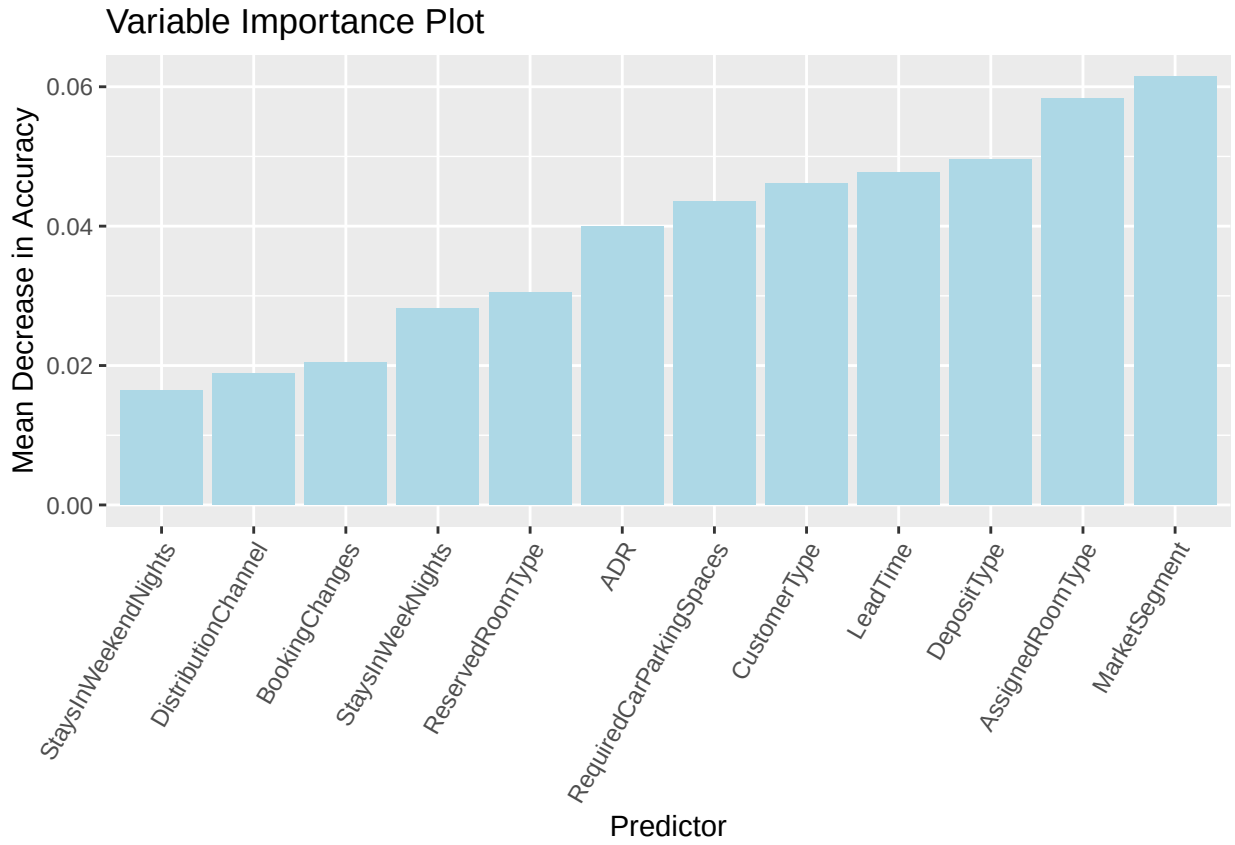
In the original data set, there are many more variables, but the variables that we will be using and talking about are described below.

- IsCanceled: value indicating whether or not the booking was cancelled
- PreviousCancellations: number of previous bookings that were cancelled by the customer prior to the current booking
- ADR: average daily rate of the room
- AssignedRoomType: code for the type of room assigned to the booking
- CustomerType: type of booking
- DepositType: indication on if the customer made a deposit to guarantee the booking
- LeadTime: number of days that elapsed between the entering date of the booking into the PMS and the arrival date
- MarketSegment: market segment designation
- RequiredCarParkingSpaces: number of car parking spaces required by the customer
- StaysInWeekNights: number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel

Methods

Variable Selection

After running an initial random forest model on the entire dataset with selected predictors, we looked at the variable importance of the predictors to decide which variables we would use in our final model. Variable importance, in the context on machine learning, refers to how much a given model "uses" that variable to make accurate predictions. In other words, the more a model relies on a variable to make predictions, the more important it is for the model.



Based on this plot, we chose the 8 predictors with the highest variable importance. The type of room being an important factor came as a surprise to us. We presume that it probably matters since basic, cheaper rooms will probably have more cancellations than suites. Similarly, the number of car parking spaces requested seemed like a rather bizarre predictor for cancellation. Maybe if a car space was requested, it makes the guest not at the mercy of an airline company when traveling to the hotel. We also decided to throw out the weekend stays variable on the basis that is was highly correlated with the weeknight stays variable as well as the reserved room type, since it was highly correlated with the assigned room type.

On the other hand, there were many variables that were highly important and *didn't* come as a surprise. For example, the average daily rate of the room and the deposit. When someone's money is on the line, it makes sense that they would take canceling their stay more serious, especially if they got an expensive room with an unfavorable cancellation policy. Likewise, customer type and market segment seemed intuitively important for prediction; certain clientele or certain sources of clientele could be more or less prone to cancellation.

Data Processing

Fortunately, the data sets that we were working with were very tidy; there were no missing values. We didn't standardize the data initially, but when training the support vector machine as well as the k-nearest neighbors models we centered and scaled the data. In fact, the data sets seemed really balanced in terms of cancellations; there is a reasonably similar amount of records that were canceled as records that were not.

As mentioned in the introduction, our focus is on the classification accuracy of machine learning methods. In particular, we will be using traditional cross-validation methods to assess the accuracy, sensitivity, and specificity of each of the five models. We split the data into a training and test data set in order to later evaluate the model's prediction accuracy. For this project we used a 75/25 split, training the data on the 75% and testing the trained models on the withheld 25%. We will then repeat this process over 5 folds of the data, averaging the results.

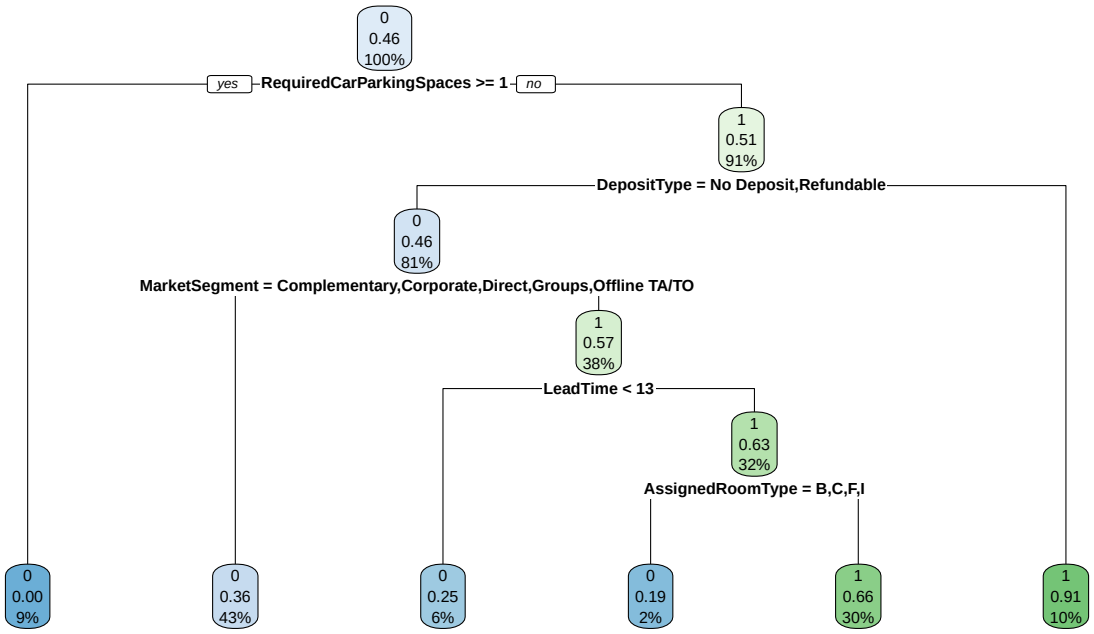
The tree-based and the K-nearest neighbors models require parameters to be tuned (more on those later). Since we used the `caret` package in R to fit all of our models, we used the tunes of the parameters that we used were deemed the "best" by the `train()` function.

Basic Classification Tree

The basic classification tree is based on partitioning the data into subgroups using simple binary splitting. Initially, all objects are considered a single group. Then, the group is binarily split into two subgroups based on the criteria of a certain variable. We then classify the observations in a specific region with majority vote. We want to grow the tree as big as we can, and then prune it back using cost-complexity pruning. This is done to not overfit the data, but pruning increases the bias. The pruning parameter needs to get tuned, which is done automatically in the `caret` package.

The advantage of using a basic classification is that it is easy to understand and has a good interpretability, which is not something that translates over to ensemble methods like the random forest. Additionally, the basic classification tree is not very computationally expensive, unlike the random forest and the support vector machine. We used the `caret` and `rpart` packages in R to fit our classification trees.

A visualization of the tree can be found below.



Random Forest

The random forest model is an ensemble tree-based method, which creates multiple trees from bootstrap samples and averages the results. Many bootstrap samples are created with replacement and then a classification tree is fitted on each bootstrap sample with a random subset of the predictors. Once a prediction has been made by all of the bootstrapped trees, the final classification is based on majority vote of the bootstrap predictions.

The advantage for using an ensemble method over a regular classification tree is that because there are many bootstrap samples being averaged together, there is less variance. This is similar to how the variance of the sample mean goes down as the sample size increases. Although it will probably increase our prediction accuracy, the random forest loses the interpretability that the basic classification tree has. Furthermore, the algorithm that is used to fit the random forest is very computationally expensive.

To train our model, we used the `randomForest` and `caret` packages in R. The maximum number of predictors for a bootstrap sample as well as the number of trees in the forest are both parameters that need tuning. Again, we will use the "best" tune, automatically given to us by the `caret` package.

Support Vector Machine

A support vector machine is a type of classification rule where it essentially maximizes the margin between groups by choosing the "line" with the widest "margin", but allowing for error. We put "line" in quotation marks because with higher-dimensional data, the "line" is actually a hyper-plane-thing. Similarly, the "margin" doesn't actually exist the in sense of a clean margin between two things, since we are allowing for some sort of error. The support vectors are the points closest to the middle that carry the most weight when

classifying, since they are the most prone to error and are deciding factors of where the classification line could go. We used a radial kernel function to deal with the non-linearity and higher-dimensions.

The support vector machine is pretty effective in higher-dimensional spaces, but doesn't perform very well with very large data sets with lots of noise. If we were to look at hotels with millions of records, a support vector machine may not be suitable. We fit our SVM using the `kernlab` and `caret` packages in R.

K-Nearest Neighbors (KNN)

K-NN is a "model free" approach, meaning that it doesn't assume any probability model on the data. Given an observation, $\boldsymbol{x}$, we want to find the k training observations that are "closest" to $\boldsymbol{x}$, and then classify the new observation using majority vote among these k neighbors. We define "closeness" based on Euclidean distance in our model. As this is the case, we need to center and scale the data, so that different scales of measurement are kept constant.

The K-nearest neighbors model is relatively intuitive and simple and has no underlying assumptions, making it a good model to consider. We are only using it for a binary classification, but it is also very easy to extend the K-NN model to multiple classes. On the other hand, the K-NN algorithm is very sensitive to outliers, high dimensional data, and imbalanced data.

The number (k) of closest neighbors is a parameter that needs tuning. We used the `class` and `caret` packages in R to fit this model. Based on our cross-validation, we used a k value of 5.

Logistic Regression

The logistic regression model uses the "logit" function $log(\frac{p}{1-p})$, which links the mean to the linear form of the regression model, $\boldsymbol{X\beta}$. Using it for binary classification, we round the fitted values either up or down to 1 or 0. The logistic regression function is defined as

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = x'\beta, \qquad \text{where } p(x) = (1+e^{-x'\beta})$$

The logistic regression model assumes that each observation is independent, that there is little or no multicollinearity among the predictors, and that there is a linear relationship between the predictor variables and log odds. This differs from the typical regression model, where the residuals must be normally distributed and have constant variance.

To fit the logistic regression model, we used the `glm()` function in base R. The advantages to using a logistic regression model are similar to those for the basic classification tree: it is not computationally expensive and easy to interpret the results. On the other hand, one of the major drawbacks to using a logistic regression model is that it is subject to the distributional assumptions on the data and errors mentioned above. If our assumptions are broken, our predictions may not be very reliable.

Results

Below are two tables of the accuracy, sensitivity, and specificity of each model fit for each hotel.

Table 1: H1 Model Results

	Accuracy	Sensitivity	Specificity
Basic Tree	0.6998	0.7600	0.6289
Random Forest	0.8030	0.8237	0.7787
SVM	0.7382	0.6919	0.7927
KNN	0.7438	0.7748	0.7073
Logit. Reg.	0.7022	0.7170	0.6847

Table 2: H2 Model Results

	Accuracy	Sensitivity	Specificity
Basic Tree	0.8279	0.9818	0.5305
Random Forest	0.8807	0.9235	0.7981
SVM	0.8375	0.9793	0.5634
KNN	0.8503	0.9101	0.7347
Logit. Reg.	0.8038	1.0000	0.4249

Looking at the output of the machine learning models on the two data sets, we can see that the Random Forest model had the highest accuracy (80% for hotel 1 and 87% for hotel 2), with the KNN and SVM models falling close behind it in the mid-70s and mid-80s, respectively. The worst performance in terms of prediction accuracy came from the logistic regression model, with an accuracy right around 70%. When considering the non-information rate of 54%, all of the models were effective in at least increasing the prediction accuracy by about 20%.

The fact that the random forest so drastically outperformed the basic classification tree did not come as a surprise, since as an average of many classification trees, ensemble methods generally improve prediction accuracy and reduce overall variance. We suspect that other ensemble methods, such as bagging or boosting, may also be able to get higher prediction accuracies. One of the main disadvantages to using ensemble techniques like the random forest is the fact that they tend to be very computationally intensive. The basic classification tree, logistic regression, and KNN models took significantly less time to run than the random forest and the SVM. If we wanted to run an ensemble method on a very large data set, it would be a nightmare with the minuscule computational power of our personal computers.

Conclusion

Hotels often suffer losses in revenue due to room vacancies that result from cancellations. Hotels can implement stringent cancellation policies as well as overbooking procedures to try and mitigate this loss in revenue. As analytics solutions are becoming more and more popular in the tourism industry, we sought to apply machine learning learning models in an attempt to solve this problem. Hotel managers could use the results of a machine learning algorithm to take action based on knowledge of forecasted cancellations or demand.

Based on the results of our study, we were able to predict the cancellation of a given room with about 75-80% accuracy. Although this is a *good* result considering the non-information rate of 54%, it is by no means excellent. These results indicate that machine learning algorithms could be a good technique for predicting

cancellations, but that there is room for improvement. With a more formal, industry expert-guided variable selection process, we believe that higher prediction accuracies can be achieved. Machine learning models could potentially be combined with other techniques to help optimize revenue from bookings.

Limitations

These two hotels come from hotels in Portugal, and although they are different kinds of hotels, we worry that the results may not be generalizable to the whole of the hotels industry. In our output we saw that the prediction accuracies for hotel 2 were higher than those for hotel 1. Given this variability among hotels, what can we say about smaller family hotels, motels, or hotels in more rural areas?

The bookings of different hotels in different areas may have different trends and their clients may have different needs. That being said, the bookings data available to us may have different information. Will we be able to achieve the same level of accuracy with different information?

There are many hotels that are smaller, family-owned, or just not franchised. As they cannot afford to have a team of data analysts or data scientists to look at trends in data, what does that say about the accessibility of these results?

Further Research

The hospitality industry is often split into to two sectors: accommodations and food and beverage. Many top-tier, elite restaurants suffer from a loss in revenue when customers cancel their reservations. Machine learning models could be applied in a similar manner for restaurants, using a different set of predictors. This could also be extended to different areas of hospitality such as transportation, events, and tours. Car rentals, for example, lose out of profit when they have a lot of cars sitting on their lot. Events also suffer from cancellations, as estimates for items, food, etc. are given based on the number of attendees.

We only used five supervised learning techniques, which is just a small fraction of all of the methods available. One method that comes to mind is the neural network. Could a neural network improve the accuracy of the predictions?

In a post-pandemic world, hotels are suffering more and more from vacant rooms and their consequent loss in revenue. How is COVID-19 affecting bookings? Has there been a large increase in customer cancellations? Would machine learning predictions still be relevant when looking at the same variables?

When speaking of the accessibility of results to non-incorporated hotels, could there be a way to make these tools available to everyone? Maybe there could be some sort of user-friendly software developed for people in the hospitality with little-to-no analytics skills.

References

1. 04/24/2019. "Global Cancellation Rate of Hotel Reservations Reaches 40% on Average." Hospitality Technology, 24 Apr. 2019, hospitalitytech.com/global-cancellation-rate-hotel-reservations-reaches-40-average.

2. N. Antonio, A. de Almeida and L. Nunes, "Predicting Hotel Bookings Cancellation with a Machine Learning Classification Model," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, 2017, pp. 1049-1054, doi: 10.1109/ICMLA.2017.00-11.

3. Antonio, Nuno, Ana de Almeida, and Luis Nunes. "An automated machine learning based decision support system to predict hotel booking cancellations." An automated machine learning based decision support system to predict hotel booking cancellations 1 (2019): 1-20.