

Cab Booking Cancellation

By Nivetha Sivaprakasam

Introduction

As technology improves the importance for a safe and fast transportation is in need. Many people rely on cab services, whether it be long distance or point to point. However; traffic, late bookings, and vehicle issues often lead to bookings getting cancelled or forgotten. A new issue is the confusion between a mobile booking or online booking which might not be received by the company. Due to the lack of service the company is held accountable for the mistakes of the drivers. In this paper I will explore the effects of different variables and how they affect the chance of a cab getting cancelled or not. This data could be used by the government, cab companies, and airports to improve their business.

Dataset

For this project I used the dataset from kaggle's class competition "Predicting Cab Booking Cancellations". The data comes from YourCabs, a trustworthy cab service in Bangalore, India. Similarly the company has issues with cab bookings getting cancelled due to the unavailability of the car, causing customers to search for an alternative. There are two csv files, one containing the predictive nature (cab.csv) and the actual values (cab_score.csv). The csv file contains various id's used to distinguish between the customer (based on the mobile number), booking, and vehicle. The package id is a binary value that distinguishes between the number of hours driven and kilometers. The travel type id is a unique id that distinguishes between the type of travel (1=long type, 2=point to point, 3=hourly rental). Similarly the other id's show the from_city, to_city, from_area and to_area value. The date and time is also included to see if there is a correlation between the 'from_date' and 'to_date'. As mentioned in the introduction online and mobile bookings are taken into consideration to see if they impact the chance of a Cab getting cancelled.

Data Wrangling

In terms of wrangling the data I first worked on the package_id and decided to set the NULL values to 0, so that NULL values would represent the package_id value of 0. For

the to_date variable I worked on setting an origin, typically 01-01-1900, and modifying the format to be 'mm/dd/yyyy'. For better understanding of the data and future modeling purposes I decided to parse the from_date and to_date into month, day, and time creating the variables from_month, from_day, from_time, to_month, to_day, and to_time. For parsing the dates I made sure the format of the from_date was 'MM/DD/YYYY H:M' and then I modified that format to be m, d, and H:M. I set the variable to be numeric since I ran into issues while modeling. For the to_month I set the NULL values to be 0 since I ran into issues with modeling.

```
mydata$from_date <- as.POSIXct(mydata$from_date, format='%d/%m/%Y %H:%M')
mydata$from_month <- format(mydata$from_date, format='%m')
mydata$from_month<-as.numeric(mydata$from_month)
mydata$from_month
```

For a better comparison, I choose to compare the difference in days between from_date and booking_created. To create this variable I set the format of the booking_created and from_date to be 'MM/DD/YYYY H:M' and I got the difference by using difftime. Afterwards to set the value to be numeric I used as.numeric and rounded the number for easy reference.

```
mydata$booking_created <-
as.POSIXct(mydata$booking_created,format='%m/%d/%Y %H:%M')
mydata$from_date <- as.POSIXct(mydata$from_date,format='%m/%d/%Y %H:%M')
mydata$diffs<- difftime(mydata$from_date ,mydata$booking_created , units =
c("days"))
mydata$diffs<-as.numeric(mydata$diffs)
mydata$diffs<-round(mydata$diffs)
mydata$diffs
```

Finally, I wrote the data frame to a csv file (cab_final.csv).

diffs	from_month	from_day	from_time	to_month
0	1	1	2:00	0
0	1	1	9:00	0
0	1	1	3:30	0
0	1	1	5:45	0
0	1	1	9:00	0
1	1	1	22:30	0
0	1	1	9:45	1
0	1	1	11:00	0
0	1	1	16:00	0
0	1	1	11:00	0
0	1	1	11:00	0
0	1	1	12:45	0
0	1	1	12:30	0
0	1	1	12:45	0
0	1	1	15:00	0
1	1	2	2:00	0
1	1	2	0:30	1
11	1	12	8:00	1

Confusion Matrix/ROC

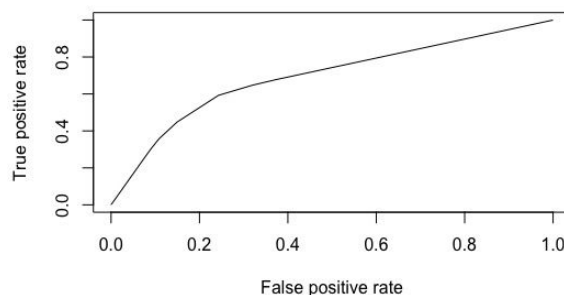
Confusion matrix is used to describe the performance of a classification model on a set of test data for which the true values is known. It contains the values of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The caret package contains functions that are helpful in creating the confusion matrix, calculating sensitivity, specificity, and 95% confidence interval.

	predicted	
actual	0	1
0	7937	101
1	602	46

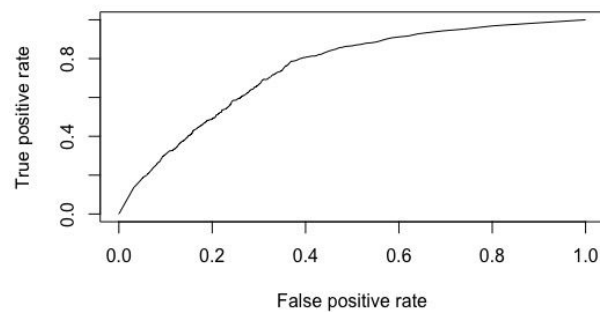
Accuracy: 0.9190

This is an example run with a model that contain the variables (**Car_Cancellation ~ travel_type_id + vehicle_model_id + online_booking + mobile_site_booking + from_lat + from_long + from_date + from_area_id**) which display the confusion matrix. The accuracy is calculated by hand using the formula: $\frac{A+D}{A+B+C+D}$, which would equate to $(7937 + 46) / (7937 + 46 + 602 + 101)$.

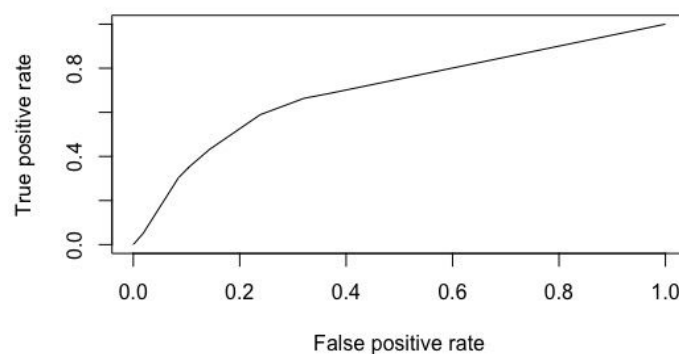
Another measure to compare the accuracy and validity of the variables is the ROC curve. It plots the true positive rate (Specificity) against the false positive rate (Sensitivity) for the different possible cut points. A curve above the line ($y = x$) is considered a valuable classifier, while a curve below is considered a poor classifier. The function ROC is used to plot the curve and calculate the AUC (area under the curve). Based on the AUC, we can decide whether the variable is a good or bad classifier. If the value is close to 0 it is considered poor, 0.5 is considered ok, and close to 1 is a valuable classifier.



The ROC curve above is for the variable Car_Cancellation. By using the predict function that predicts values based on a linear model I was able to display the ROC curve and calculate the AUC. Based on the AUC value of **0.6595**, we can say the model is a good classifier.



The ROC curve above is similar to the previous model except the data for this model is balanced equally. By setting a variable and equating it to a randomForest function containing the variables I was able to create the ROC curve and calculate the AUC. Based on the AUC value of **0.7494** we can say the predictor is getting better.



The ROC curve above is for an unbalanced model in which the Car_Cancellation is balanced to 0.67 compared to the other variables which are balance 0.33. By setting the parameters to `c(0.67, 0.33)` I was able to balance the random forest unevenly and create a ROC curve and calculate the AUC. Based on the AUC value of **0.6984** we can the predictor is good.

Logistic Regression

Logistic regression is the statistical method for analyzing data in which there is at least one independent variable that could determine an outcome. In this case the outcome would be whether the cab was cancelled or not which is determined by the value 1 (cancelled) or 0 (not cancelled). In order to find the variables that impact the cab cancellation I used both the Weight of Evidence and Information Value.

Weight of Evidence/ Information Value

The Weight of Evidence is used to determine the predictive power of an independent variable compared to the dependent. The woe package is used and contains the function `iv.mult`. This function can also be used to determine the information value (IV). The information value is used to determine the importance of the variable. Based on the IV value I was able to determine whether the variable is useful or not. If the IV value is less than 0.02 the variable is considered not useful, 0.02 to 0.1 means the predictor is weak, 0.1 to 0.3 is a medium strength variable, 0.3 to 0.5 is a strong indicator, and greater than 0.5 represents a suspicious relationship. The run below represents the IV value for the variable `diffs` in which the IV determines the relationship to be Medium.

Information Value 0.13

```
[[1]]
```

	variable	class	outcome_0	outcome_1	pct_0	pct_1	odds	woe	miv
1	diffs	(;0.5)	24593	2434	0.6	0.8	0.8	-0.2	0.04
2	diffs	<0.5;)	15706	698	0.4	0.2	1.7	0.6	0.09

```
sql
```

```
1 when diffs < 0.5 then -0.241729021585819
```

```
2 when diffs >= 0.5 then 0.558924104552018
```

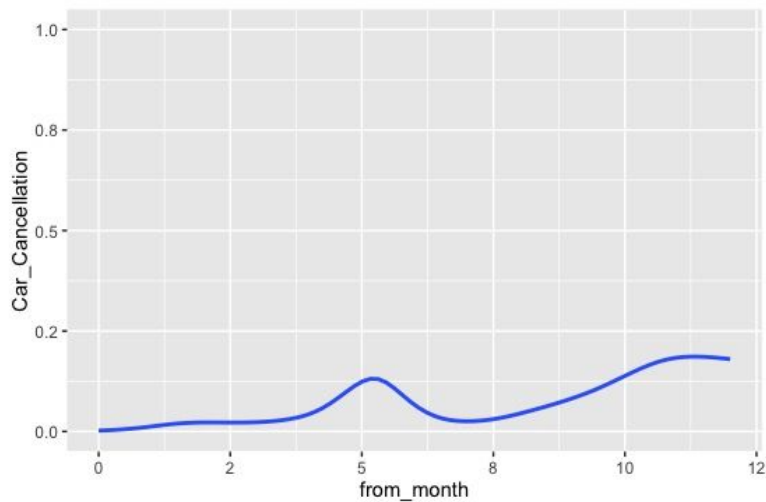
I decided to test 10 variables excluding the id variables to determine 4 important variables. By repeating this call:

`iv.mult(mydata,"Car_Cancellation",vars=c("online_booking"))`, with other variables I was able to determine the importance of the variables.

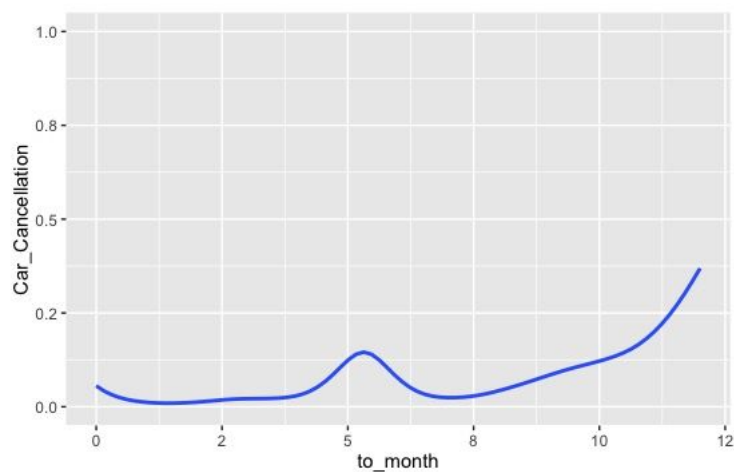
Modeling

For the logistic modeling I used the library `ggplot2` and the function `ggplot` to plot the curve. Based on the variable I was able to see either an increasing or decreasing slope

in the curve. By checking the WOE and IV values I decided to model the variables from_month, online_booking, diffs, and to_month to see if the values impact the chance of a cab getting cancelled.

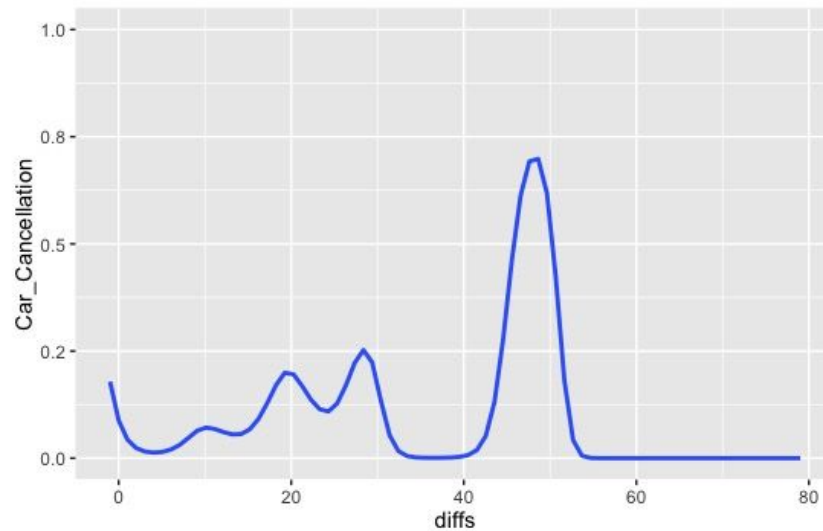


This curve is for the variable from_month and Car_Cancellation. Based on the curve we can predict as the from_month increases the chances for car cancellation increases. Around the month June and November is when the highest chance for car cancellation occurs compared to July and January.

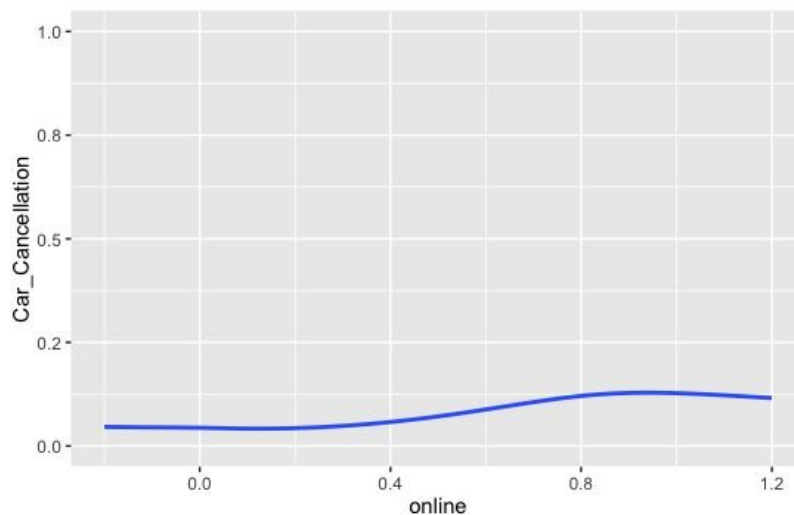


This curve represents the variable to_month and the chance of a cab getting cancelled. As the month increases, we can see an increase in Car_Cancellation. Similar to the

from_month curve there is an increase around June and November, and a decrease in January and July.



This curve represents the variable diffs which is the difference in days from the booking created and from date. The chances for a cab getting cancelled is highest around 50 days and 0 days; while the lowest is 60 to 80 days.



This curve represents the chance of Car Cancellation for the variable online_booking. If the online_booking is 1 the cancellation chance is higher compared to a value of 0. For all the models I set the y limit to be from 0 to 1, for accuracy purpose.

Decision Tree

A decision tree is a method that uses a tree like model of decisions and their consequences, including the number of nodes, and chance of probability. To calculate the most important variables it is important to use information gain.

Information Gain

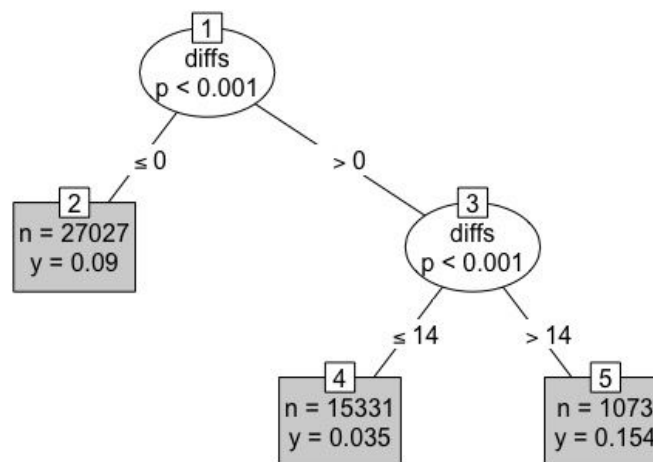
Information gain is a method to see how much information is gained by doing the split using the particular variable. The higher the value the more important the variable is.

```
attr_importance
from_month      1.216516
```

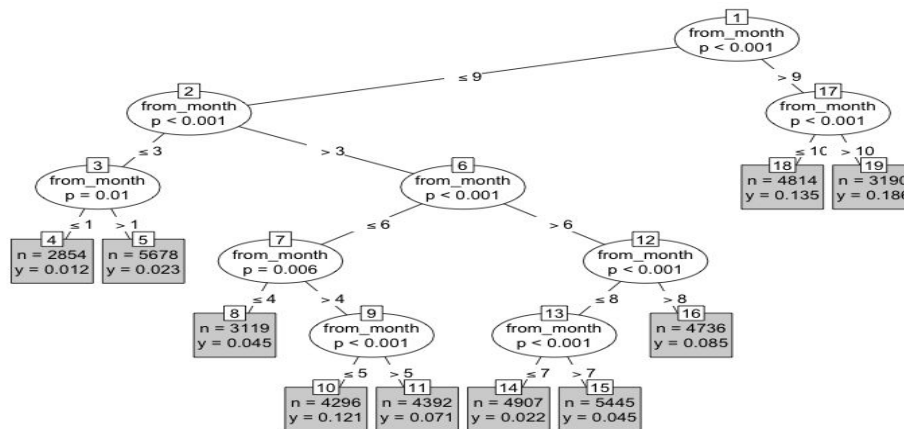
Above is an example run for the variable from_month. I decided to model the variables from_month, to_month, and diffs to see if there was a difference based on the value.

Modeling

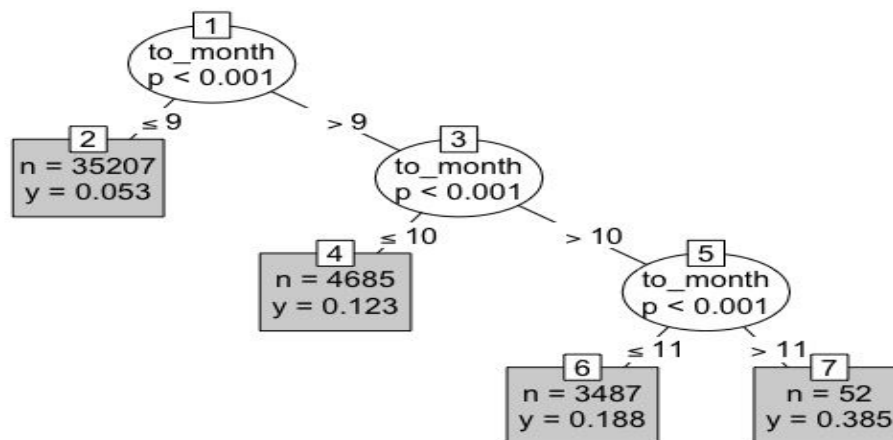
For the decision tree I used the package party which contains the function ctree. The function takes the parameter of the y value, x value, and dataset; while the plot function takes the type of plotting. In this case I set the plot type to be simple in order to clearly see the results.



Based on the tree above, we can see the decisions made to the variable diffs. Starting from node 1 we can decide which way to go based on the diff value. The value 'n' represents the number of records which that value, while 'y' represents the chance of cab cancellation. The most common value is < 0 and the highest chance of cab cancellation occurs in a diff value greater than 14.



The tree above is used for the variable from_month and displays the variable possibilities from 1 to 12. Based on the tree we can see the most common value is for the month January while the highest chance for cancellation occurs in November. The lowest chance for cab cancellation occurs around January.



This tree represents the decision for the variables to_month and Car_Cancellation. Based on the tree we can see that the most common value is for the months before September, while the least common value are for the months November and December.

The months with the highest chance of cancellation are November and December, and the least chance are for the months September and before.

Results

The graphs presented above show that the chance of cab cancellation is higher during popular holidays including Christmas hence an increase during December and November. One common problem to notice if the difference in days is low there is a high chance of not getting a cab, while a longer difference decreases the chance of cancellation. One major issue with the data was that most of the variables were id with no legend, including the car id, area id, and city id. If there was a key it would have been helpful to use the id to gain more results. For the variable package_id, a value of 1 (4 hr & 40 km) and 7(12 hrs & 120 km) resulted in the higher chance of cancellations compared to 3(6 hrs & 60 km). As expected for mobile bookings, if the value is 1 then the chance of cancellation is high compared to 0. A travel type of long distance has a low chance of cancellation possibly since they are expensive and companies would prefer long distance over short distances.

Future/Recommendations

In the future companies should try to prioritize their cabs to help customers early morning and late nights to get them to their locations safely. Companies should take safety over money as their priority; however, this will take some time to achieve. Companies should give an early notice for a change so customers could find a backup. Customers should prepare early in advance for cab bookings and should prioritize the use for long distance. Customers should avoid online and mobile booking and resort to phone bookings to ensure a cab. In general simple changes and backup plans can ensure a customer has a cab instead of being estranged. In the future this study can be done in other countries to see if a location has an impact on the cancellation. The distance traveled can be of great use for research and company records.