Data Science Study: YourCabs.com Cancellations

Authors: Jasmine Touton and Akhil Chugh

Section: 81 Baltimore

Business Understanding

At YourCabs.com, the company will sometimes need to cancel a booking made by a customer due to unavailability of car. This causes grave inconvenience for customers who plan to take a cab and are suddenly left without a mode of transportation to their desired destinations. It is a severe customer service problem, and could cause loss of market share, and therefore profits, if left to continue unchecked. YourCabs.com must figure out how to identify the individual cases where cab cancellation might happen and head them off through a creative customer service solution like altering the booking process or alerting and re-scheduling customers at risk of cancellation well in advance.

YourCabs.com collects a variety of data characteristics for its cab bookings, including car cancellations. A number of characteristics could contribute directly to whether a booking will be cancelled. Through data mining – setting our target variable as cab cancellations and analyzing which characteristics contribute most to it – we will better understand how to restructure the cab booking process and customer service methods to lessen cancellations. Those variables that contribute the most will need to be tweaked to control customer expectations or restructured. For example, if we were to find that the online booking was a large indicator of cab cancellations, we would need to re-assess how and when cabs are dispatched after the online booking cancellations (is there a way to do so without error or more efficiently?)

The current YourCabs.com website offers different packages for booking. You can book a regular cab based on pick up location, destination, time and day or you can choose a package that includes hours/distance (example: 6 hours and 60 kms). The packages, if used often by customers, likely help reduce the cab cancellation problem; cabs would be assessed as busy up to a set amount of hours and the system would be able to more accurately predict if cabs will be available to avoid overbooking. Unfortunately, packages aren’t used often as a booking option.

Customers are also currently automatically prompted to book a return trip when they book their original trip even if it may be several days (or weeks) away. While this may be helping the problem by assisting in planning for cab availability, it also may be

complicating planning processes for the company by making bookings several weeks out. Our data mining processes will help us to assess the usefulness of these current marketing tactics while allowing us to determine if there might be other marketing techniques, operations re-organization needs, or customer service methods to employ.

Data Understanding & Preparation

The data used to address our business problem was retrieved from Kaggle: Predicting cab booking cancellations for a cab company in Bangalore: https://inclass.kaggle.com/c/predicting-cab-booking-cancellations

The data fields included in the source data are demonstrated in **Exhibit 1**.

The problem and data set lent itself to be analyzed as a classification problem, as we wanted to predict which bookings would be cancelled. The training data set from Kaggle contained 43,431 transactions (or rows) and 20 variables (columns). After the initial analysis of the data and our business problem, it was evident that the target variable was “Car_Cancellation” – whether the booking was cancelled or not.

Through the initial analysis, it became obvious that the data had a large number of Null values for the particular variables. These Null values were affecting the data while running visualizations, as well as attempting to run decision trees on the data. There were also a number of variables that were unnecessary to solving our problem. As a result, all these variables were removed to clean up the Null’s from the data, for the purposes of the analysis and evaluation. Variables removed included: “id, user_id, from_area_id, to_area_id, from_city_id, to_city_id, to_date, from_lat, from_long, to_lat, to_long, Cost_of_error”.

New variables were also created from the existing variables in the data set to help with the prediction of whether each individual in the population of the data set would be considered a cancelled cab or not.

The variable “difftime” was created to represent the elapsed time between the date/time the booking was created online or on the phone, and the date/time of the cab departure. This variable was calculated by subtracting the “booking_created” from the variable

“from_date”. The logic was that the elapsed time between the two dates and times could have a bearing on cab cancellations. **Exhibit 2** shows the code for these calculations.

The variables “month_booked_for”, “dayofweek_booked_for” and “hour_booked_for” were created to represent the month, day of the week, and the hour, respectively, that the cab was reserved for. We created them because each of these elements individually might have a bearing on whether cabs are cancelled. For example, certain months, days of the week or hours in a day, might have a higher rate of cancellations than others, which would help derive deployment insight from our model. See **Exhibit 3** for the code.

Visualizations

With new variables in place, the attributes in the data were mainly of discrete numerical values: integers, factors and dates. We began producing graphs through R to do some preliminary analysis with this information. We looked to test whether some of the correlations we suspected with difftime, month_booked_for, hour_booked_for, and online_booking might be true.

We looked first at frequencies of booking to get a better idea of our data’s distribution. The first bar graph we looked at was frequency of cancellations during specific months (see **Exhibit 4)**; it indicated a greater frequency in summer months and less so in winter months, with almost zero in month 12. The data appears to follow a typical distribution of booking of cabs based on our own intuition and therefore does not appear skewed.
We also looked at frequency of hours (see **Exhibit 5**). The hours most frequently booked for are between 7 a.m. and 10 a.m. and again between 4 p.m. and 7 p.m. indicating rush hours. This also appears, intuitively, to be a good sampling of data to use since those are typical peak times.

Next, we looked at frequency of Travel Type to help determine preferences of customers for deployment techniques later on (see **Exhibit 6**). Overwhelmingly, users who booked used point-to-point travel (Travel Type 2) over Travel Type 1 (distance) or Travel Type 3 (hourly) rentals. This indicates a customer preference for using the point-to-point system. Customers want to book based on point-to-point rather than choosing some type of package.

Next, viewed variable correlation to determine whether we could see some preliminary patterns that might show up in our decision tree. We suspected that people who booked online might see more cab cancellations due to some communications obstacle between the system and the dispatcher. We found that a significant number of cancellations (over 15,000) resulted happened to customers who booked online (see **Exhibit 7**). Online booking is likely to play out in our modeling as an important factor to cancellation.

We wanted to see whether the difftime variable (difference in days from time of booking to time of departure) had something to do with the month that customers booked and felt like perhaps holiday months would be booked more often in advance, which would help with our solution. We saw a pretty even distribution of months and difftime when difftime is less than 40. When it is more than 40, however, it seems to be booked for primarily the last half of the year (months 8 - 12) (see **Exhibit 8**).

Another consideration: perhaps there is a month that people are booking for well in advance (i.e. the difftime variable) where car cancellations are happening. We could definitely head off problems in this month if that were the case. We used a 3D scatter plot to visualize this in **Exhibit 9**. Month 11 does appear to have a stronger number of calculations than the rest of the month, and the difftime measure extends further, up to 40 days and perhaps past that. This can begin to make us think about solutions: a promotion for booking in November could be if you book less than 20 days out, you get a discount, resulting in less overall bookings that month

Modeling

The business problem and the data for the cab cancellation is a classification problem, where the target is to predict whether each cab booking will be classified as a cancellation or not. Since this a quantifiable target being predicted and there is more than enough historical data (500+), this problem is considered supervised data mining.

We decided to apply the Decision Tree model to the data set to create segments in the data. We wanted to be able to understand which variables and segments had the most impact with respect to the target variable: “Car_Cancellation”.

We chose the Decision Tree model over a Regression model. Since our data was not of a continuous nature, the regression model would not be very beneficial. Additionally, based on the characteristic of our data set - containing a lot of categorical variables - the decision tree model would allow for a large number of levels in the tree. The regression model would need null values at the different levels, which would complicate the analysis. The overall ease of use and understanding makes the decision tree the best option for analyzing the data.

Looking closely at the decision tree (see **Exhibit 10**) and the print function of the decision tree (see **Exhibit 11**), we can get a better idea of which factors contribute the most to the predicted probabilities of the cabs that have cancellations. The most important factors are "month_booked_for" variable, the "hour_booked_for" variable, and the"online_booking" variable.

The cabs that do cancel fall into one of three branches of the decision tree. The groups help us to better define what to target in deployment. They are characterized as:

Group 1 (*must pass all the criteria to be characterized as part of the group)*

- month_booked_for = 05, 10, 11 (Bookings that were made for May, October or November)
- hour_booked_for = 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 22, 23(Bookings that were made for the times other than between 5pm to 10pm)
- online_booking < 0.5 (Bookings were not made online)
- difftime >= 16 (The difference between when the booking was made and when it as made for is greater than 15.8 days)
- month_booked_for = 05, 10 (Bookings that were made for May or October)
- hour_booked_for =! 12, 23 (Bookings that were made for the times other than the hour of 12pm and 11pm)

Group 2 *(must pass all the criteria to be characterized as part of the group)*

- month_booked_for = 05,10,11 (Bookings that were made for May, October or November.)

- hour_booked_for = 17,18,19,20,21 (Bookings that were made for the times between 5pm to 10pm
- online_booking < 0.5 (Bookings were not made online)
- difftime >= 15 (The difference between when the booking was made and when it as made for is greater than 15.3 days)
- month_booked_for = 11 (Booking that were made for May or November)

Group 3 *(must pass all the criteria to be characterized as part of the group)*

- month_booked_for = 05,10,11 (Bookings that were made for May, October or November.)
- hour_booked_for = 17,18,19,20,21 (Bookings that were made for the times between 5pm to 10pm
- online_booking >= 0.5 (Bookings were made online)
- difftime < 0.19 (The difference between when the booking was made and when it as made for is less than 0.186 days)
- dayofweek_booked_for = Friday, Monday, Sunday, Thursday (The day of the week the cab was booked for is a Monday, Thursday, Friday or Sunday)

Evaluation

From the “Cost_of_error” variable supplied in the data, we are told that a cost is incurred if the booking is misclassified. For an un-cancelled booking, the cost of misclassification is 1. For a cancelled booking, the cost is a function of the cancellation time relative to the trip start time, with a fixed penalty of 100 units given if they cancel within 15 minutes of the booking time.

This information can be translated into an expected value calculation to enumerate the possible outcomes of our business problem. Since there is no profit gained from the product/service being sold in our cancel cab bookings problem, we can translate the “Cost_of_error” penalty into a profit. The average value of the penalty in the “Cost_of_error” was 8 units, which will be given as a profit when our model predicts the booking will be cancelled, and in actuality it is cancelled. In situations when the model predicts the booking to be cancelled, but actually there is no cancellation, we will incur a cost of 1 (cost of misclassification). In the situations where the model predicts there is no cancel, we will have neither a cost, nor a benefit, since we are assuming that there is no

opportunity cost. See Cost/Benefit information in **Exhibit 12**. According to our expected benefit calculation, we should target bookings as long as the estimated probability of cancellation is greater than 11%.

Next, we tested the accuracy of the model based on the threshold rate of 11%. We set the random seed to 1, and drew 20% of the dataset, 8,686 indices, out of 43,431 to be the test dataset. The remaining indices were saved as the training dataset. Using the same set of variables that were created and selected as most relevant we built a training decision tree model. We then evaluated the test dataset by applying the model to the test data and getting the predicted values (see **Exhibit 13**).

FALSE	TRUE
7270	1416

From the prediction class error on the test data, we can calculate the accuracy:

Accuracy = Number of correct decisions made / Total number of decisions made
= 7270 / (7270 + 1416)
= 0.8370 = 83.70%

Based on this information and our initial assumptions, the predictive model is 83.70% accurate. Because this accuracy percentage can yield misleading results (some errors are more costly than others), we constructed a confusion matrix to allow a more detailed analysis than mere proportion of correct guesses (See code in **Exhibit 14**). The confusion matrix translates into a set of expected rates. Using these expected rates and the cost/benefit information from our assumptions, we are able to compute the expected profits of $6.39 per booking that would result from using this model.

Knowing that the expected profits of being able to properly predict the cancellation of a booking can help increase the profits by $6.39 (per booking) helps validate the usefulness of this model. YourCabs.com can further use this information by testing hypothetical changes and process improvements within the company. They can then verify the these improvements using the same baseline predictive model and compare

the resulting expected profits to determine if the improvements are actual benefits and should be deployed.

Deployment

The months of May, October, and November were the largest indicators of whether the cab may be late, which might indicate some seasonal disruptions in our service. As data miners at YourCabs.com, we would task our traffic coordinators to take a comprehensive look at city congestion in those months and find out where the problem locations/routes are. Perhaps there are large events or holidays that clog up typical traffic routes during these months. If we can pinpoint the cause of the seasonal disruption, we can then send maps or training information to the cab drivers who use YourCabs.com requiring them to seek alternate routes specifically in those months.

The next node on the decision tree showed the characteristic of hours. For the "problem" hours -- between 5 and 10 p.m. -- a similar traffic analysis as to the one mentioned above could be applied. Additionally, YourCabs.com could institute a fare increase to lessen the demand for this peak times since the problem may simply be supply side (not enough cabs). With a fare increase, those who really want to use the service will pay more money for YourCabs.com (increasing our revenues), and those who do not will no longer cause strain on the YourCabs.com system.

Online booking is the third most important factor to indicate cancellation in our decision tree model, which means there is likely a disconnect between how many online bookings the system allows and how many cabs are actually available at that time/location. As data miners, we would recommend our software developers test the current YourCabs.com search functionality to see whether it is returning incorrect availability information. If the system proved capable, we would recommend checking the lead times between when a reservation is made and when the dispatcher reserves the cab. Perhaps, there are too many overlapping requests coming in, resulting in passengers getting double booked. As we work to repair our communication and search functionality, we may choose to issue email confirmations after bookings. Customers would be alerted on the website that their cab is only confirmed as booked once they receive the email confirmation. This would allow the dispatching team to assure a cab is not double-booked before scheduling that customer, cutting down on cancellations.

Difference in time between when the booking was made and when the booking is scheduled for also shows up on our decision tree model. In two cases, a person might cancel if the difference in time is greater than 14 or 16 days. One way to remedy this, since all of these cancellations are also affected by online booking, is make it impossible to book online for more than two weeks out. This would require that customers that need to book early speak with a representative, facilitating a more error-free booking process.

The tactics listed above face a major hurdle: they involve manipulating the behavior of the customer to allow more time for the current dispatch system to work properly. While manipulating the customer through marketing might be easier than manipulating operations, in the long-term it could push demand to another part of the system that cannot handle it. For example, making customers pay more between the hours of 5 and 10 p.m. would perhaps cause increased bookings at 3 and 11 p.m. (just outside that window) and again cause a strain to the system. YourCabs.com will need to use data miners periodically to evaluate the effects of deployment and adjust accordingly.

We must consider ethical impacts of our techniques; we are acting on information that tells us how people are booking, where they are heading to, and more. Data about how people travel is sometimes considered sensitive and private. YourCabs.com should have a disclaimer available on the site that indicates use of the service also allows for internal sharing of data.

There is a risk inherent in attempting to solve the cancellation problem based on characteristics we obtain about (only) our customers. We're not looking at the complete picture, like cab driver data, and that regardless of these fixes we may simply not have enough cabs in our system to run a smooth operation. If cab cancellation number does not drop significantly after deployment of our data mining insights, we will need to adjust our insights and re-deploy techniques. If the problem remains, perhaps the issue lies in our resources, i.e. the number of cabs in our service compared to the customer demand. If that's the case, we should shut down the website and operations until we can successfully develop the resources necessary (side note: YourCabs.com seems to have done just that).

Appendix:

Exhibit 1: ***Data Characteristics from Kaggle.com for YourCabs.com***

id - booking ID

user_id - the ID of the customer (based on mobile number)

vehicle_model_id - vehicle model type.

package_id - type of package (1=4hrs & 40kms, 2=8hrs & 80kms, 3=6hrs & 60kms, 4= 10hrs & 100kms, 5=5hrs & 50kms, 6=3hrs & 30kms, 7=12hrs & 120kms)

travel_type_id - type of travel (1=long distance, 2= point to point, 3= hourly rental).

from_area_id - unique identifier of area. Applicable only for point-to-point travel and packages

to_area_id - unique identifier of area. Applicable only for point-to-point travel

from_city_id - unique identifier of city

to_city_id - unique identifier of city (only for intercity)

from_date - time stamp of requested trip start

to_date - time stamp of trip end

online_booking - if booking was done on desktop website

mobile_site_booking - if booking was done on mobile website

booking_created - time stamp of booking

from_lat - latitude of from area

from_long - longitude of from area

to_lat - latitude of to area

to_long - longitude of to area

Car_Cancellation (available only in training data) - whether the booking was cancelled (1) or not (0) due to unavailability of a car.

Cost_of_error (available only in training data) - the cost incurred if the booking is misclassified. For an un-cancelled booking, the cost of misclassification is 1. For a cancelled booking, the cost is a function of the cancellation time relative to the trip start time.

Exhibit 2: ***Creating the Difftime Variable***

```
> cab_cancel_data$from_date_ptime<-strptime(cab_cancel_data$from_date,
"%m/%d/%Y %H:%M")
> cab_cancel_data$booking_created_ptime<-
strptime(cab_cancel_data$booking_created, "%m/%d/%Y %H:%M")
> cab_cancel_data$difftime<-
difftime(cab_cancel_data$from_date_ptime,cab_cancel_data$booking_created_ptime,u
nits="days")
```

Exhibit 3: ***Creating the Month_Booked_For, Dayofweek_booked_for, and Hour_booked_for Variables***

```
> cab_cancel_data$month_booked_for<-
as.factor(format(cab_cancel_data$from_date_ptime, "%m"))
> cab_cancel_data$dayofweek_booked_for<-
weekdays(as.Date(cab_cancel_data$from_date_ptime))
> cab_cancel_data$hour_booked_for<-
as.factor(format(cab_cancel_data$from_date_ptime, "%H"))
```

Exhibit 4: ***Frequency of Months when Bookings are Created For***

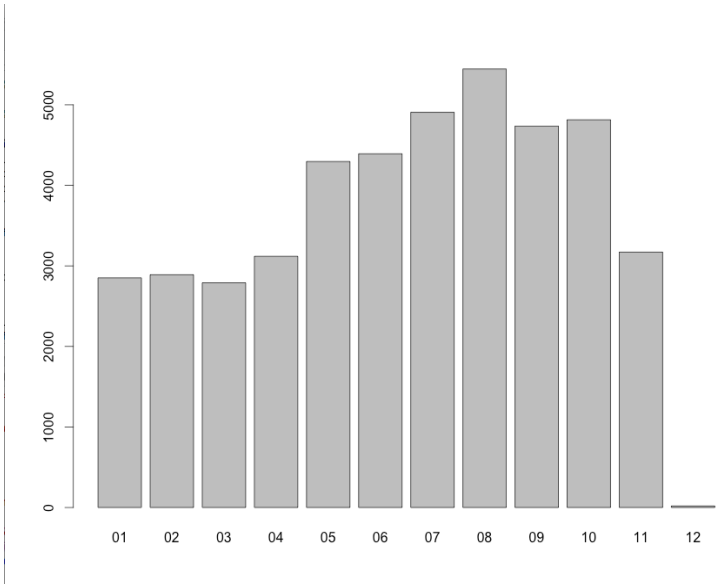

Exhibit 5: ***Frequency of Hours When Bookings are Created For***

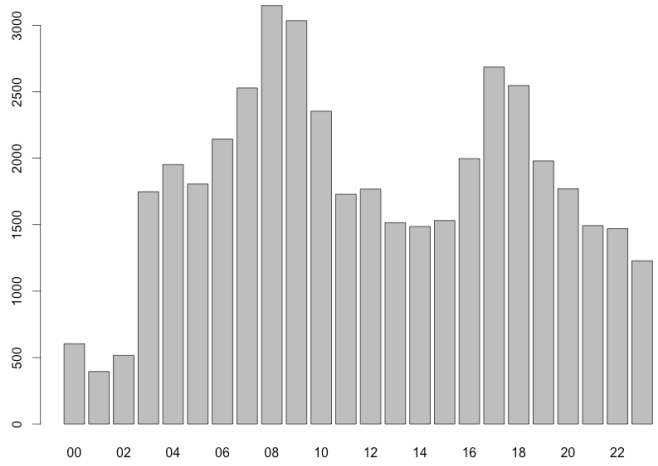

Exhibit 6: Travel Type Preferences

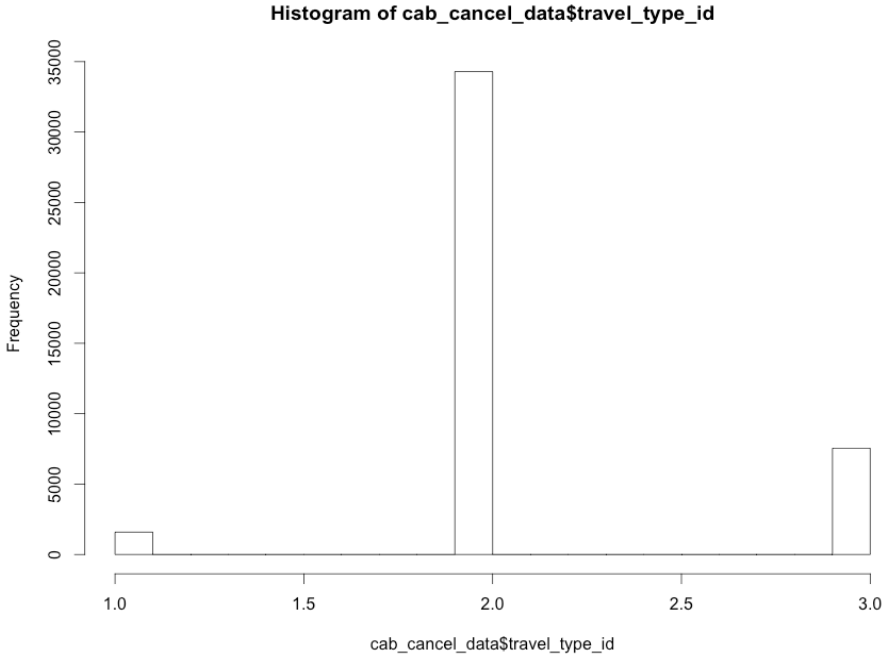

Exhibit 7: ***Cab cancellations for Online Booking***

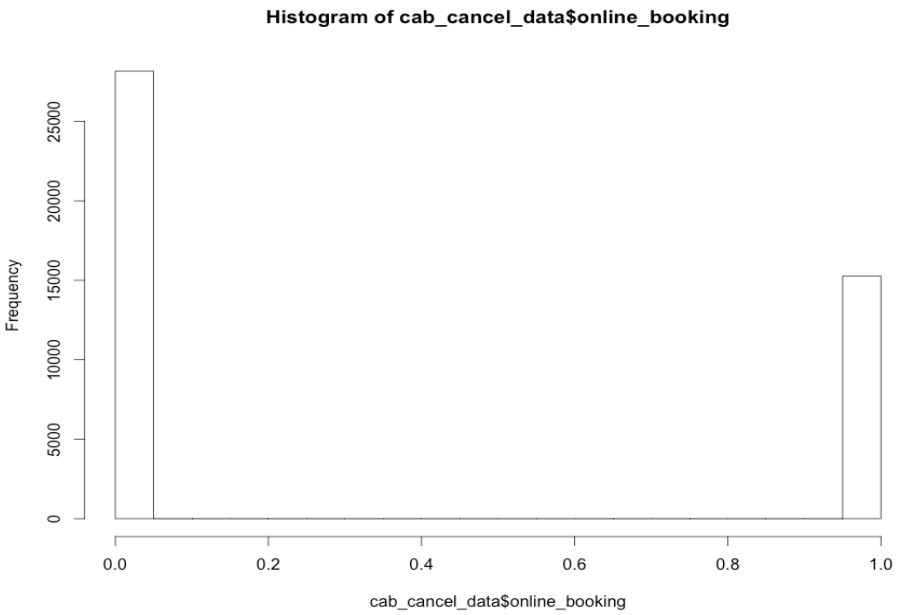

Exhibit 8: ***Is Month Booked For Correlated with DiffTime?***

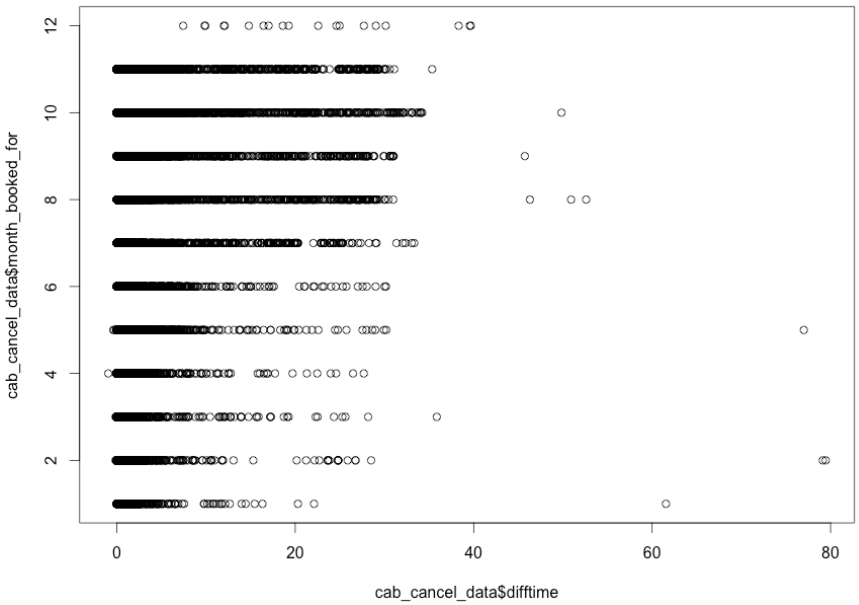

Exhibit 9: ***How does DiffTime and Month Booked For Effect Car Cancellation?***

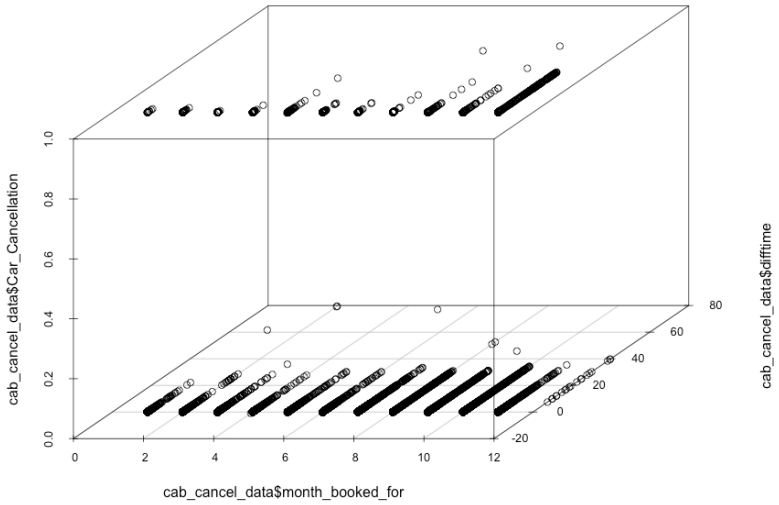

Exhibit 10: D*ecision Tree of Car Cancellation Dataset*

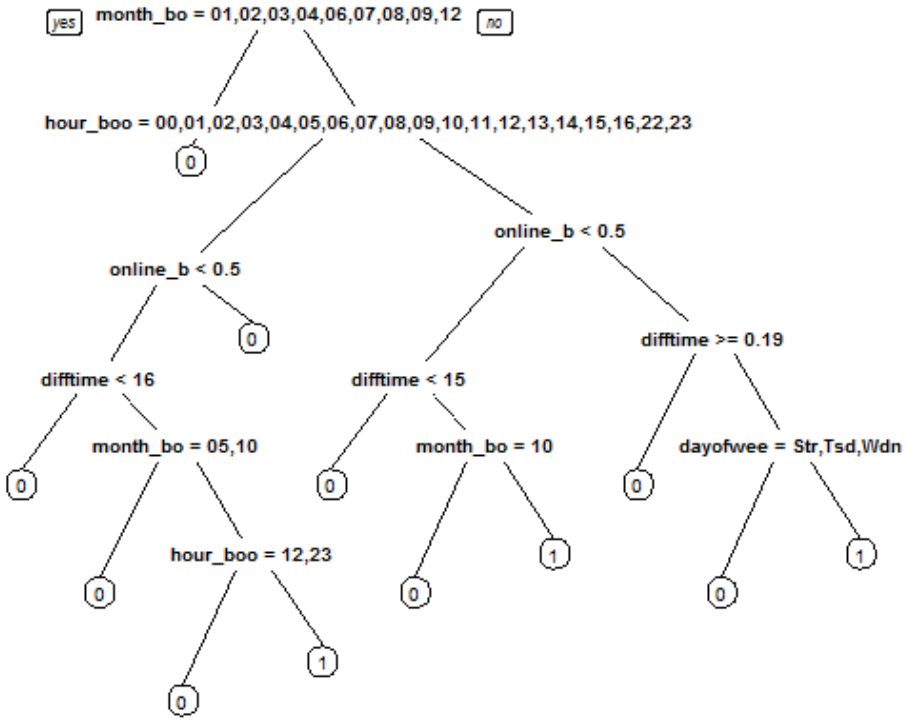

Exhibit 11: ***Print Function of DT Model on Car Cancellation Dataset***

```
> print(training_dt_model)        # model results
n= 43431

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 43431 3132 0 (0.92788561 0.07211439)
    2) month_booked_for=01,02,03,04,06,07,08,09,12 31150 1368 0 (0.95608347
0.04391653) *
    3) month_booked_for=05,10,11 12281 1764 0 (0.85636349 0.14363651)
      6) hour_booked_for=00,01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,22,23
9523 1085 0 (0.88606532 0.11393468)
       12) online_booking< 0.5 5674  442 0 (0.92210081 0.07789919)
         24) difftime< 15.85868 5446  350 0 (0.93573265 0.06426735) *
         25) difftime>=15.85868 228   92 0 (0.59649123 0.40350877)
           50) month_booked_for=05,10 116    1 0 (0.99137931 0.00862069) *
           51) month_booked_for=11 112   21 1 (0.18750000 0.81250000)
            102) hour_booked_for=12,23 10    0 0 (1.00000000 0.00000000) *
            103) hour_booked_for=06,07,08,09,10,11,14,15,22 102   11 1 (0.10784314
0.89215686) *
       13) online_booking>=0.5 3849  643 0 (0.83294362 0.16705638) *
      7) hour_booked_for=17,18,19,20,21 2758  679 0 (0.75380711 0.24619289)
       14) online_booking< 0.5 1648  283 0 (0.82827670 0.17172330)
         28) difftime< 15.34062 1525  220 0 (0.85573770 0.14426230) *
         29) difftime>=15.34062 123   60 1 (0.48780488 0.51219512)
           58) month_booked_for=10 60    4 0 (0.93333333 0.06666667) *
           59) month_booked_for=11 63    4 1 (0.06349206 0.93650794) *
       15) online_booking>=0.5 1110  396 0 (0.64324324 0.35675676)
         30) difftime>=0.1864583 604  119 0 (0.80298013 0.19701987) *
         31) difftime< 0.1864583 506  229 1 (0.45256917 0.54743083)
           62) dayofweek_booked_for=Saturday,Tuesday,Wednesday 185   79 0
(0.57297297 0.42702703) *
```

63) dayofweek_booked_for=Friday,Monday,Sunday,Thursday 321 123 1 (0.38317757 0.61682243) *

Exhibit 12: ***Cost/Benefit Information for Cab Cancellation Problem***

		Actual	
		Cancel	**No Cancel**
Predicted	**Cancel**	8	-1
	No Cancel	0	0

v_R =$8

v_{NR} =-$1

Expected value of targeting = $p_R(x)*v_R + [1-p_R(x)]*v_{NR}$

Expected value of targeting = $p_R(x)*100 + [1-p_R(x)]*(-1)$

Expected benefit of not targeting = 0

$p_R(x)*8 + [1-p_R(x)]*(-1) > 0$

$9p_R(x) > 1$

$p_R(x) > 0.11111$ → rounded to 0.11. Therefore, according to our expected benefit calculation, we should target bookings as long as the estimated probability of cancellation is greater than 11%.

Exhibit 13: ***Using Predict Function to Determine Class Error***

```
> test$dt_pred_prob<-predict(training_dt_model,test)[,2]
> test$dt_pred_class<-predict(training_dt_model,test,type="class")
> test$dt_pred_class<-test$dt_pred_prob> 0.11
> test$dt_pred_class_error<-test$Car_Cancellation!=test$dt_pred_class
> table(test$dt_pred_class_error
```

Exhibit 14: ***Confusion Matrix Construction and Expected Value Calculation***

```
> table(test$dt_pred_class,test$Car_Cancellation, dnn=c("predicted","actual"))  #
confusion table on test data
         actual
predicted    0    1
    FALSE 6987  366
    TRUE  1050  283
> exp_rate<-as.matrix(table(test$dt_pred_class,test$Car_Cancellation)/nrow(test))  #
expected rates
> exp_rate

                0          1
  FALSE 0.80439788 0.04213677
  TRUE  0.12088418 0.03258117
> cost_benefit<-matrix(c(8,0,-1,0),2)    # cost-benefit information (from external source)
> cost_benefit
     [,1] [,2]
[1,]    8   -1
[2,]    0    0
> exp_rate*cost_benefit                  # element-wise multiplication

                 0          1
  FALSE  6.43518305 -0.04213677
  TRUE   0.00000000  0.00000000
> exp_value=sum(exp_rate*cost_benefit)    # expected value
> exp_value
[1] 6.393046
```

Exhibit 15: Full Code Used

```
#---------- Starting from scratch-----------
library(rpart)
library(rpart.plot)
rm(list = ls())
cab_cancel_data<-
 read.csv("Kaggle_YourCabs_training.csv",stringsAsFactors=FALSE)
sapply(cab_cancel_data,class)
ncol(cab_cancel_data)
head(cab_cancel_data)

####################################
#        Creating Variables        #
####################################

#---------- Variable: Professor code for "difftime"
cab_cancel_data$from_date_ptime<-
 strptime(cab_cancel_data$from_date, "%m/%d/%Y %H:%M")
cab_cancel_data$booking_created_ptime<-
 strptime(cab_cancel_data$booking_created, "%m/%d/%Y %H:%M")
cab_cancel_data$difftime<-
 difftime(cab_cancel_data$from_date_ptime,cab_cancel_data$booking
 _created_ptime,units="days")

#---------- Variable: Month that cab was booked for
cab_cancel_data$month_booked_for<-
 as.factor(format(cab_cancel_data$from_date_ptime, "%m"))

#---------- Variable: Day of the week that cab was booked for
cab_cancel_data$dayofweek_booked_for<-
 weekdays(as.Date(cab_cancel_data$from_date_ptime))

#---------- Variable: Time the cab was booked for
cab_cancel_data$hour_booked_for<-
 as.factor(format(cab_cancel_data$from_date_ptime, "%H"))

#---------- Variable: Month the booking was made
#cab_cancel_data$booking_month<-
 as.factor(format(cab_cancel_data$booking_created_ptime, "%m"))

####################################
#        Decision tree model       #
```

```
###################################

# build a decision tree (classification tree) model
cancel_dt_model<-
 rpart(Car_Cancellation~vehicle_model_id+package_id+travel_type_i
 d+online_booking+mobile_site_booking+difftime
   +month_booked_for+dayofweek_booked_for+hour_booked_for,
 data=cab_cancel_data, method="class",
 control=rpart.control(cp=0.003))

rpart.plot(cancel_dt_model)   # tree plot
print(cancel_dt_model)        # model results
summary(cancel_dt_model)      # model result details

cancel_dt_pred<-predict(cancel_dt_model)  # get the predicted
 value - class probabilities (default)
head(cancel_dt_pred)

cancel_dt_pred_class<-predict(cancel_dt_model,type="class") # get
 the predicted value - class membership
head(cancel_dt_pred_class)

cab_cancel_data$dt_pred_prob<-cancel_dt_pred[,2]   # save the
 second column - predicted probability of default
cab_cancel_data$dt_pred_class<-cancel_dt_pred_class  # save the
 predicted class membership

head(cab_cancel_data)
#head(cab_cancel_data[which(cab_cancel_data$dt_pred_prob>0.7),])
 # return the customers whose predicted probability is greater
 than 70%
#head(cab_cancel_data[order(-cab_cancel_data$dt_pred_prob),])
 # sort customers by probability of default in descending order

# Cross-validation in rpart
dt_model_full<-
 rpart(Car_Cancellation~vehicle_model_id+package_id+travel_type_i
 d+online_booking+mobile_site_booking+difftime

 +month_booked_for+dayofweek_booked_for+hour_booked_for,
 data=cab_cancel_data, method="class",
 control=rpart.control(cp=0))
rpart.plot(dt_model_full)
printcp(dt_model_full)   # xerror, xstd - cross validation
```

```
 results
plotcp(dt_model_full)    # x-axis: model complexity, y-axis:
 error rate from cross-validation


####################################
#        Model Evaluation          #
####################################

# Cost-benefit information
cost_benefit<-matrix(c(100,0,-1,0),2)    # cost-benefit
 information (from external source)
cost_benefit


# Evaluation on training data  - provides no assessment of how
 well the model generalizes to unseen cases
cab_cancel_data$dt_pred_class_error<-
 !(cab_cancel_data$Car_Cancellation==cab_cancel_data$dt_pred_clas
 s)
table(cab_cancel_data$dt_pred_class_error)   .  # error rate =
 259/(259+9741) = 0.0259

# Holdout validation with the decision tree result
set.seed(1)   # set random seed
index <- sample(nrow(cab_cancel_data), 8686) # randomly select
 8686 indices out of 43,431 (20% of dataset)
test <- cab_cancel_data[index,]       # save 20% as a test
 dataset
training <-cab_cancel_data[-index,]   # save the rest as a
 training set

# build a model using the training set
training_dt_model<-
 rpart(Car_Cancellation~vehicle_model_id+package_id+travel_type_i
 d+online_booking+mobile_site_booking+difftime

 +month_booked_for+dayofweek_booked_for+hour_booked_for,
 data=cab_cancel_data, method="class",
 control=rpart.control(cp=0.003))

rpart.plot(training_dt_model)   # tree plot
print(training_dt_model)        # model results
summary(training_dt_model)      # model result details
```

```r
# Evaluation on test data - apply the model to the test dataset
 and get the predicted values
test$dt_pred_prob<-predict(training_dt_model,test)[,2]
test$dt_pred_class<-predict(training_dt_model,test,type="class")
head(test)

test$dt_pred_class<-test$dt_pred_prob> 0.11

test$dt_pred_class_error<-
 test$Car_Cancellation!=test$dt_pred_class
table(test$dt_pred_class_error)     # error rate=53/(1947+53) =
 0.0265


#test$dt_error<-test$dt_pred_class!=test$churn   # check whether
 the prediction is correct or not
#table(test$dt_error)
#test$dt_pred_class<-predict(training_dt_model,test,type="class")
#test$dt_pred_class<-test$dt_pred_prob> 0.1


####################################
#        Confusion Table           #
####################################

table(training$dt_pred_class,training$Car_Cancellation,dnn=c("pre
 dicted","actual")) # confusion table on training data
table(test$dt_pred_class,test$Car_Cancellation,
 dnn=c("predicted","actual"))  # confusion table on test data
exp_rate<-
 as.matrix(table(test$dt_pred_class,test$Car_Cancellation)/nrow(t
 est))  # expected rates
exp_rate
cost_benefit<-matrix(c(8,0,-1,0),2)    # cost-benefit information
 (from external source)
cost_benefit
exp_rate*cost_benefit                   # element-wise
 multiplication
exp_value=sum(exp_rate*cost_benefit)    # expected value
exp_value
```