Case Study 02: Taxi Cancellations

Taxi-cancellation-case.csv is the dataset for this case study.

Business Situation

In late 2013, the taxi company Yourcabs.com in Bangalore, India was facing a problem with the drivers using their platform—not all drivers were showing up for their scheduled calls. Drivers would cancel their acceptance of a call, and, if the cancellation did not occur with adequate notice, the customer would be delayed or even left high and dry. Bangalore is a key tech center in India, and technology was transforming the taxi industry. Yourcabs.com featured an online booking system (though customers could phone in as well), and presented itself as a taxi booking portal. The Uber ride sharing service started its Bangalore operations in mid-2014. Yourcabs.com had collected data on its bookings from 2011 to 2013, and posted a contest on Kaggle, in coordination with the Indian School of Business, to see what it could learn about the problem of cab cancellations. The data presented for this case are a randomly selected subset of the original data, with 10,000 rows, one row for each booking. There are 17 input variables, including user (customer) ID, vehicle model, whether the booking was made online or via a mobile app, type of travel, type of booking package, geographic information, and the date and time of the scheduled trip. The target variable of interest is the binary indicator of whether a ride was canceled. The overall cancellation rate is between 7% and 8%.

Assignment

1. How can a predictive model based on these data be used by Yourcabs.com? (5”)

2. How can a profiling model (identifying predictors that distinguish canceled/uncanceled trips) be used by Yourcabs.com? (5”)

3. Explore, prepare, and transform the data to facilitate predictive modeling. (30”)

Here are some hints:

- In exploratory modeling, it is useful to move fairly soon to at least an initial model without solving all data preparation issues. One example is the GPS information—other geographic information is available so you could defer the challenge of how to interpret/use the GPS information.

Can other meaningful features be extracted from GPS data such as Trip length? (hint: you can use geodesic distance from GeoPy package or you can code you own Haversine distance formula.)

- How will you deal with missing data, such as cases where Null/NaN is indicated? Are the classes imbalanced? What would you do to remedy that?
- Think about what useful information might be held within the date and time fields (the booking timestamp and the trip timestamp). Will it be more helpful if you can break timestamps into hour, day of week, month, and year?
- Think also about the categorical variables, and how to deal with them. Should we turn them all into dummies? Use only some?
- Try to find meaningful features from the features 'from_city_id' , 'to_city_id', 'from_area_id' and 'to_area_id'.

4. Fit several predictive models (at least three, such as, KNN, Logistic, Naïve-Bayes; or you can also try decision tree and SVM if you want) of your choice. Do they provide information on how the predictor variables relate to cancellations? (30")

5. Report the predictive performance of your model on the test set (The confusion matrix, accuracy, precision, recall, ROC-Curve, AUC, Precision-Recall Curve, AP, etc.). How well does the model perform? Can the model be used in practice? (25")

Notebook Organization: (5)

1. Make sure you have appropriate Headers and Section Titles
2. Please make sure you are addressing all the questions with appropriate question numbers.
3. The explanations and insights should be right below the plotted graphs.
4. Do not forget to import all necessary packages at the top.
5. Confirm that your notebook is run-able.
6. All Assignments to be submitted as a python jupyter notebook.
7. Example notebook name here: BUAN685_Case01_Initials01_ Initials02.ipynb

8. Typed answers and elaborations in Markdown cells.
9. Equations in Latex format in markdown cells
10. Python coding answers in jupyter notebook code cells
11. Make sure the code is readable and clear.
12. Make comments as much as possible to ensure readability.