Glassdoor Data Science Internship Project

by **Jin Xie** (bruce.jinxie@gmail.com)

April 9, 2018

All of the code, prediction and report files can be found in the zip file attached to the email or the github page below:

- `https://github.com/kof900/Glassdoor-Intern-Project`
  - "`input`" folder contains the Train and Run data.
  - "`output`" folder contains the Run data with prediction of likelihood of "isWon".
  - Jupyter notebook file, "`Glassdoor Data Challenge.ipynb`", is in the default path.
  - "`report`" folder contains the report latex and pdf file.

Answers to Question 3 are in the following.

a. First evaluate the data - what initial observations can you make from this data? Are there any concerns about the data?

This is a binary classification problem. In general, a logistic regression model should work. The predictors have both categorical and continuous variables. There are a lot of missing values in the dataset. I also note that it's an imbalanced binary classification problem. "isWon=True" only takes 10.9% of all the observations in the Train data. I have several concerns regarding the data.

(a) Imbalanced Binary Classification Problem.

Data is very unbalanced. Only 10.9% of the observations are the winning cases ("isWon=True") in the train data. The remaining 89.1% are all failure cases ("isWon=False").

Table 1: Case Proportion in the Train Data

Case	Proportion
isWon	0.109
nonWon	0.891

It suggests us that accuracy can be deceptive. If we predict every case is failure case, i.e. "isWon=False", we could have a 89.1% accuracy. But it's meaningless. We should use some criteria such as AUC or FNR.

(b) Duplicates in the Train Data.

"EmployerId" should be unique. However, I observe that there are 2 observations in the train data having the same "employerId". The "employerId" is "2bX4R".

(c) Some “employerId” in the Run Data can be found in the Train Data.

In other words, we could potentially cheat on “employerId”. We just use the value in the Train data as the prediction in the Run data since they can be found in the Train data. I did not do this since it is not the purpose of this project. But one could dramatically increase the model performance by cheating in this way.

(d) Missing Values.

There are so many missing values in the Train Data. Some methods such as tree methods will be able to deal with missing values directly. However, some algorithm such as logistic regression model cannot deal with missing values unless imputing the missing values beforehand. Otherwise we will have to delete all the cases containing any missing values. The missing values can be a big problem

Table 2: Missing Values in the Train Data

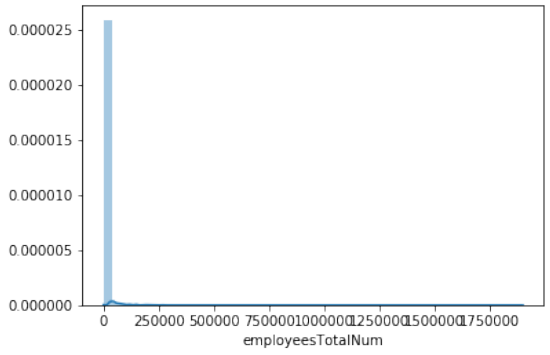
Feature	Missing Count	Proportion
isWon	0	0.000
subsidiary	0	0.000
numJobs	0	0.000
basePayAmount	0	0.000
has_interviews	0	0.000
timeOnSite	0	0.000
employerId	0	0.000
ATS__c	4	0.000
employeesTotalNum	172	0.010
Industry	602	0.036
RespondedContacts	2674	0.158
intDifficulty	2839	0.168
intDuration	3149	0.186
freshContent	3292	0.195
followers	3292	0.195
contentCount	3292	0.195
awards	3292	0.195
photos	3292	0.195
starRating	3956	0.234
monthOfSignupPageViews	4708	0.279
clicks	6037	0.357

(e) Outliers.

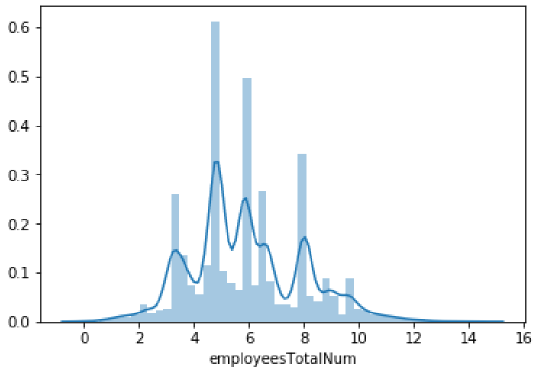
There a lot of outliers in the Train data. And it could harm the analysis.

(f) Predictor is skewed and in large scale.

Some predictors such as clicks have a very large range and the scale is very large. They are also very skewed. It suggests that some transformation may be needed. I did $\log(x)$ or $\log(x+1)$ transformations on some predictors depending on whether it’s skewed and has zero values. For example, for “employeesTotalNum”, the original and log transformed data histograms can be found as below.



(a) Original

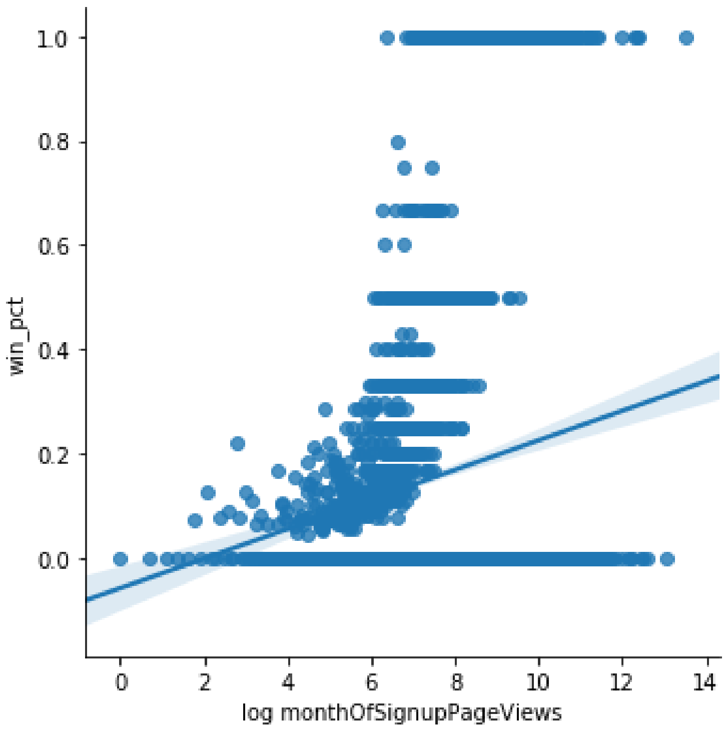


(b) log Transformed

b. Imagine you are presenting your results to a non-technical audience. Summarize your findings - What are the top 3 insights that you can draw from this data set? Share your plots and findings.

(a) The higher "monthOfSignupPageViews" the employer has, the higher likelihood it would have to become a client.

Figure 2: winning likelihood vs. Log monthOfSignupPageViews

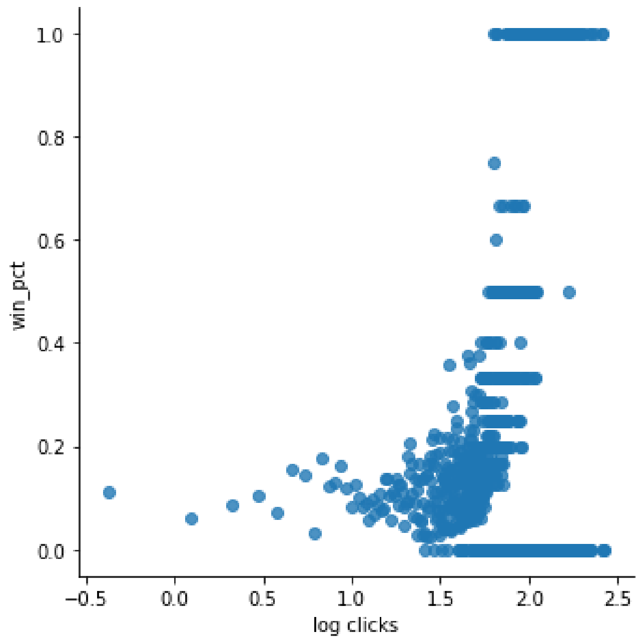


From the above plot, we can easily see that with "log monthOfSignupPageViews" increasing, the winning likelihood also has an increasing trend.

(b) The higher "clicks" the employer has, the higher likelihood it would have to become a client.

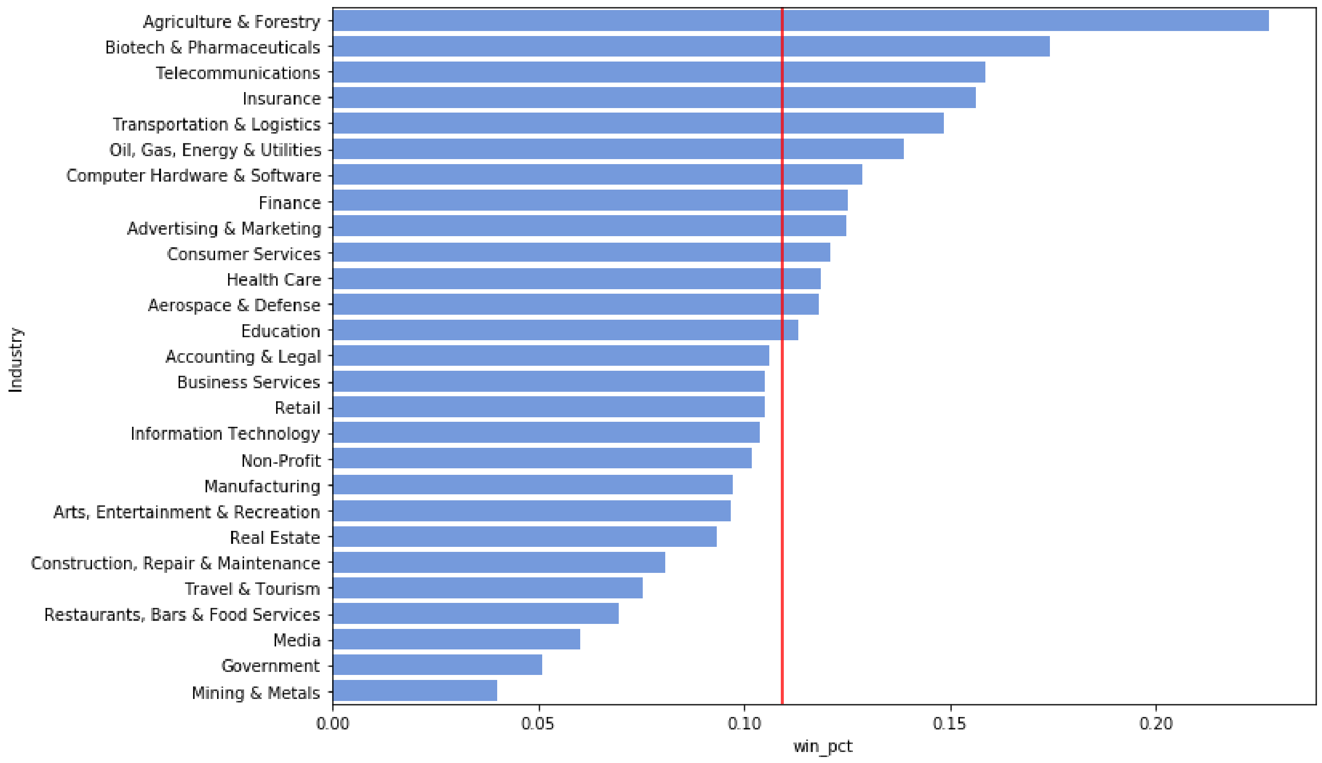
From the above plot, we can easily see that with "log clicks" increasing, the winning likelihood also has an increasing trend.

Figure 3: winning likelihood vs. Log clicks



(c) I made a barplot using winning likelihood by industries as in Figure 4.

Figure 4: winning likelihood by industries (red vertical line is the average for all)



Clearly, “Agriculture & Forestry” is the industry area that has the highest winning likelihood. We should put more effort on this industry area rather than “Mining & Metals” industry area, which has the lowest winning likelihood.

c. Train and test a model to predict how likely an account will be won (meaning they become a paying customer). Youre welcome to just pick a few features (please describe your reasoning in picking the features). For this task, use the data set `data_challenge_traindata.csv`.

For this part, I trained 3 different models.

(a) Gradient Boosting Tree method using the cutting-edge python library LightGBM.

(b) Logistic Regression only with the features that don't have any missing values.

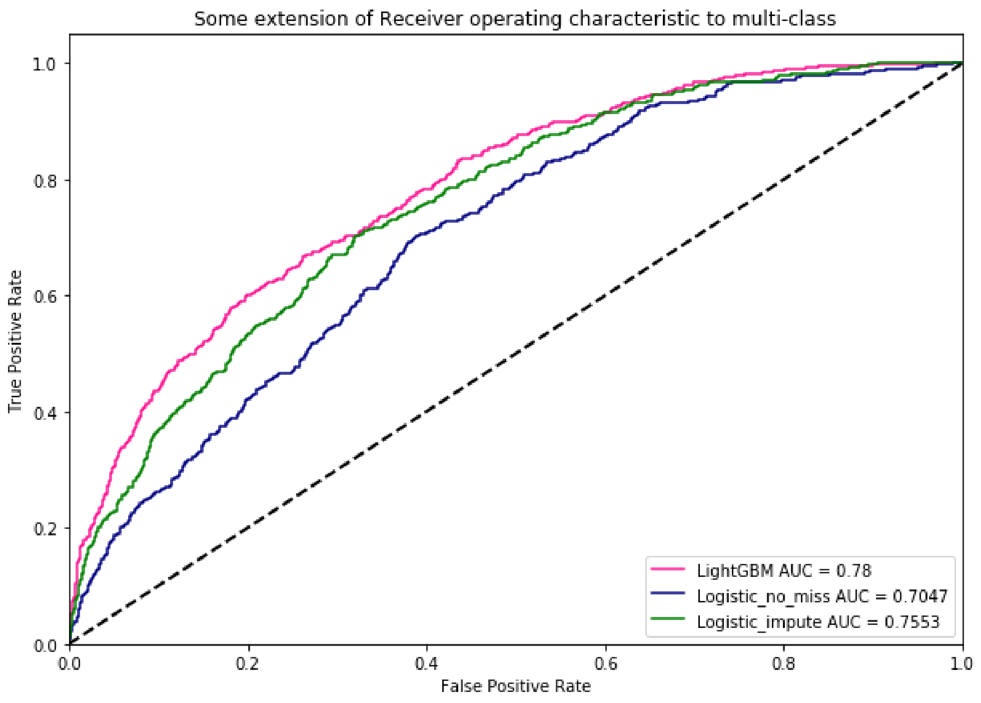
(c) Logistic Regression with imputing the missing values using mean value.

Their performances will be discussed in the next questions.

d. Briefly describe in a few words: How well does the model work? Which are the most important features? What about accuracy?

As we discussed in question **a**. Accuracy is deceptive since we can easily get 89.1% accuracy by predicting every case as failure. Instead, we use Area Under the Curve (AUC) from Receiver Operating Characteristic (ROC) curve.

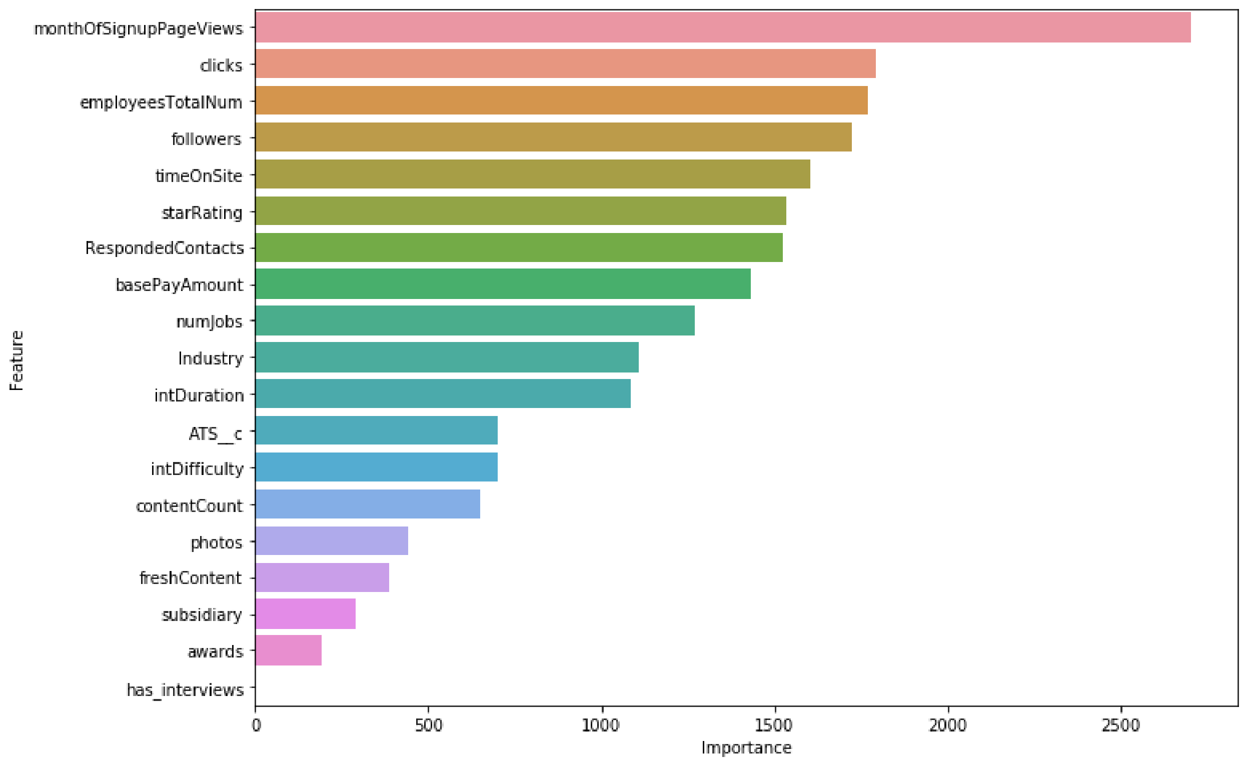
Figure 5: ROC curve for 3 methods



As we can observe from the ROC curve (Figure 5), LightGBM outperforms both Logistic Regression models. And Logistic Regression model with imputation has better performance than Logistic Regression only on non-missing features.

Figure 6 is the feature importance plot from LightGBM model. We can see that "monthOfSignupPageViews", "clicks" and "employeesTotalNum" are the top 3 important features.

Figure 6: Feature importance from LightGBM



e. Run your model on the dataset `data_challenge_rundata.csv`. What insights can you draw?

After fitting the model to the Run data (`data_challenge_rundata.csv`), I list the top 10 "employerId" that has the highest winning likelihood. I also list the bottom 10 "employerId" that has the lowest winning likelihoods. In other words, we should reach out more actively to the top 10 employers.

Table 3: Top and Bottom Prediction on Winning Likelihood in the Run Data

Top		Bottom	
employerId	Winning Likelihood	employerId	Winning Likelihood
uPI8q	0.896	8tMdC	0.003
DSjqi	0.882	yW4fr	0.004
4VJt2	0.875	4bAHQ	0.004
Z7ZPH	0.825	VQP0r	0.004
tRv6U	0.802	TUUsk	0.004
Dgawj	0.792	MDehi	0.004
4363z	0.791	Mc024	0.004
zpIcy	0.791	6p9qR	0.004
i3lry	0.790	yYrbE	0.004
iQNj7	0.785	cDiJK	0.004

f. If you run out of time/had extra time: Please explain the next steps you would take.

I have a few thoughts in my mind for future works.

(a) Screen out outliers by looking at each feature.
(b) Use EM algorithm to impute the missing values rather than just using the mean.
(c) Try several other different models and do model ensemble.