Healthcare Project

Phase II

Group member:
Yunjia Jiang , Hanran Weng, Lingtian He, Tong Cheng, Siyu Chen

Executive Summary

In present days, 30 days readmission rate becomes a crucial performance criterion for hospitals, because it somehow affects hospitals' reputation and represents their treatment effectiveness. Therefore, hospitals started to track patients' activities and kept the their medical records for having a better understanding about the patients'health status after treatment and the performance of medical treatment. Our project is strive to predict whether a patient will return to a hospital within 30 days after being discharged (defined as *Return* variable)by analyzing the dataset given by 5 hospitals.

As for this project, we believed that the cost for miss prediction of patient readmission is pretty high, because readmission implies that hospital may not give a proper treatment to patient at first time and even increase staff's burden of taking care of more patients. Therefore, we would like to focused on the false negative rate(FNR) in our prediction,which means we predict the patient will not return to hospital within 30 days, but actually he or she return.

Before conducting models, after analyzing the concepts of each predictors, and the relationship between predictors and target variables, we reasonably divided the dataset into three groups for our future models based on the the dispositions of the patient from the emergency department. One includes patients who were diagnosed deceased; one includes patients who were left the hospital without diagnosis; another includes patients who were diagnosed in the hospitals. We seperate overall data set into two data sets: the test data set, for the model comparison, and the training data set, for selecting the best parameters by cross-validation or BayesOptimization. Applied several models to predict the patients return status such as random forest, KNN and XGBoost, and discovered that XGBoost with randomly search parameters has the lowest false negative rate and the highest accuracy on testing data set. It is the most suitable model for our target to reduce FNR and increase accuracy.

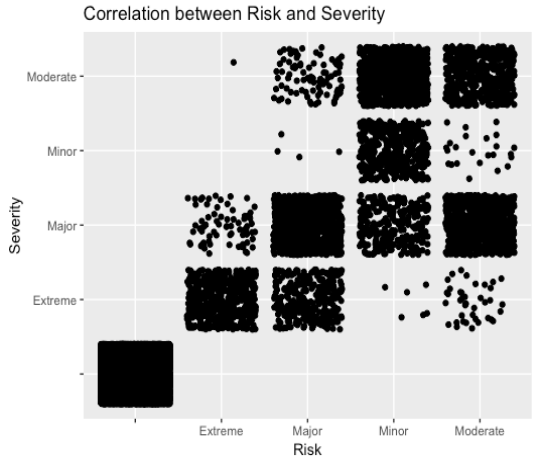
Data Insights and Analysis

Interpreting the data:

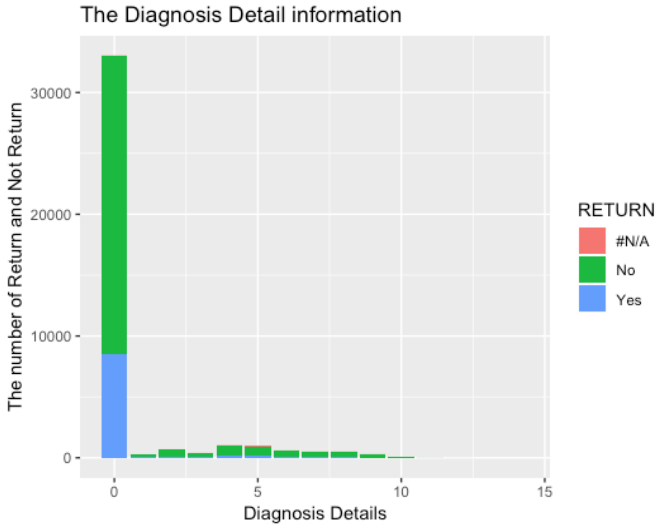
At very first, we would like to try a k-means method to classify the data set into groups. Since we already know the outcome of our predictions is Return status(Yes or No), the desired number of clusters K is 2. We ran K-means method once, but it is hard to interpret and understand the cluster means for the categorical variables. Therefore, we converted them to dummy variables just for have a better ideas about the cluster means of every category in each variable. For example, the K-means method will display cluster means of Hospital A, B,C,D and E so that we could have a better idea about how the each cluster is related to the hospitals. Based on observation of cluster means, the overall percentage explained is 48.3% and we listed following variables that have more clear distinctions between two groups:

	ED_RESULT Admit	ED_RESULT Discharge	DC_RESULT Home or Self Care	ADMIT_RESULTNot Admitted	CONSULT_ ORDER	CHARGES
Group 1	0.1026606	0.61148027	0.7923838	0.81477330	0.05470606	5089.375
Group 2	0.8951220	0.01463415	0.3073171	0.02439024	0.85365854	163439.395

As we can see from the table, most patient in group 1 are not Admitted for both ED_RESULT and ADMIT_RESULT. Also, on average, group 1 patient paid much less than group 2 patient and most of their DE_RESULT are “Take Home or Self Care”. Based on the observation, we would like to classify group 1 patients’ return status is No, while Yes for group 2 patients. We also conducted the table with real return status and cluster groups, and the calculated the percentage that classified wrong, which is 25.67%. In such condition, we admit that cluster means did a really good job to show the distinctions between two groups, but the high false negative rate implies K-means is not a good method for us to classify the return status.



Then, we glimpsed the concepts of each columns, there are two predictors attracting our attention, Severity and Risk. According to the definitions of these two predictors, we assume if the patient has higher severity of injury or illness, he or she will also have higher risk of death. we tried to find are there any correlations between predictors. Displaying all the observations by RISK and SEVERITY these two variables, the distribution of these observations by the categories of each variable can be discovered. From this graph, among patients whose the illness or the injury is diagnosed as moderate level, few of them whose Risk level was diagnosed as Major and only one patient whose risk level is diagnosed as Extreme. Among the patients whose injury or illness is diagnosed as Extreme, only few of them are diagnosed as having moderate or minor risk of death. There is an obvious correlation between SEVERITY and RISK observed: the higher severe injury or illness a patient has, the larger opportunity this patient is exposed to the death.



When exploring the relation between the predictors and target variable, we came up with some intuitive thoughts.One of them is that the more diagnosis detail information the patient has, the more carefully diagnosis this patient will get, resulting in a larger probability of this patient won't return the hospital. Thus, drawing a bar chart by RETURN and DIAG_DETAILS shows the distribution of all observations in these two dimensions. According to this chart, most of the diagnosis detail information of patients who return the hospitals is 0, which may prove our intuitive thought to some extent.

When browsing through the whole data, we observed a large number of blanks in certain columns, specially in Admit_Result, Consult_In_Ed, Risk and Severity, so we tried to

figure out the reason why it happened. By understanding the logic behind each column in file “Hospitals_Dictionary” , we founded out the relationships between these columns based on domain knowledge in the following:

- If a patient was disposed to admission after leaving the emergency room, the ADMIT_RESULT, the RISK and SEVERITY of this patient will be recorded.
- If a patient was disposed to admission after leaving the emergency room, the patient has the probability to request a consultation.
- Only if a patient requested a consultation by a speciality physician, will the hospital probably charge this consultation for this patient. Also, whether the consultation was requested by the emergency department or not was recorded.

So, we would like to conclude that values in ED_RESULT will logically affect the values in ADMIT_RESULT, and consequently has impact on the following variables: Consult_Order, Consult_Charge, Consult_In_Ed, Risk and Severity.

Separating Dataset:

As we stated before that there are plenty of blank information in certain columns because they are strongly associated with the ED_RESULT. Therefore, we conducted a pivot table(see Appendix) to display the relationship between ADMIT_RESULT and ED_RESULT. Based on this pivot table, we could better interpret how likely each ADMIT_RESULT generated by ED_RESULT would be. For example, when the ED_RESULT is “Admit”, the probability that blank occurs is 1.2%, while the “LWBS after Triage”, having the total number 2754, lead to blank ADMIT_RESULT at all. As this condition is highly followed by the relationships we observed in the interpreting, we would like to divided the dataset into three groups for building corresponding models based on the content of ED_RESULT. By doing so, patients with different ED_RESULT will be fit to its relative model to predict RETURN. We listed the following ways to separate the dataset:

- When ED_RESULT is “Deceased” or DC_RESULT is “Expired”, we will classify the patient’s RETURN to 0 since these information imply the patient would not be able to back to hospital due to death.
- When ED_RESULT is “Admit to External Psychiatric”, “Arrived in Error”, “AMA”, “ Discharge”, “Elopement”, “Left prior to completing treatment”, “Left without signing discharge instructions”, “LWBS after Triage” and “LWBS before Triage”, we would like to define those observations as one dataset and build a model for this dataset without considering the Consult_Order, Consult_Charge, Consult_In_Ed, Risk and Severity for them. That means we can keep the missing value in those eliminating variables without preprocessing.

- When ED_RESULT is “Admit”, “Admit to UMMS Psychiatry”, “Observation”, “Send to L&D after Rooming”, “Send to L&D Before Rooming (Mom)” and “Transfer”, we would like to define those observations as another dataset and build a model considering all the applied variables.

Preprocessing the data:

When dealing with the missing value of variables, we preprocess them generally before separating the dataset and specifically after separating.

Before separating the dataset: we fill the missing value in GENDER, ACUITY_ARR, RACE and ETHNICITY with their majority level, and fill the missing value in ED_RESULT and ADMIT_RESULT with ‘Not Admitted.’

After separating the dataset: we fill the missing value in CHARGES with each dataset’s mean of CHARGES, since we thought for each dataset the CHARGES might be different; and we also fill the missing value in RISK and SEVERITY with their major levels in the dataset considering those values.

Regrouping categorical variables:

Some categorical variables have many levels in the dataset, such as FINANCIAL_CLASS and ED_RESULT. Because the occurrence of each level is very different, we decide to regroup those levels to avoid possible error after further separating the training set into training, validation and testing. There are three general rules. First, we want to keep as much levels as we could because we only care about prediction not inference. So if the level appears more than 100 times(major levels) in the dataset, we will keep that. For minority levels(less than 100 times), we first examine their return probabilities and regroup all minority levels into a new level notated “Other”. The third rule is that, we will make sure the levels within ‘Other’ have similar return probability, so if the return probability of a minority level is relatively high(comparing with the mean return probability of that categorical variable), we will combine it with a major level with similar return probability.

Adding interaction variable:

As stated above, we only care about prediction but not inference. We try to use as many as variables as we can as long as they give good result. After tried some combination, we found the combination of CHARGESxHOSPITAL, RISKxAGE, SEVERITYxAGE gives us best prediction results. So, we decided to use this combination to run models.

Model Evaluation

As stated above, we assume the cost for FNR is higher than the others. So along with the accuracy, a lower FNR is also our criteria.

By now, we've tried logistic, KNN, naiveBayes and ensemble method including RandomForest, and XGBoost in both datasets *out_hos* and *in_hos*. Under the consideration of needs in evaluating our models' performance, we split a 30% testing data set out of each data set after extracting all the rows that have 'No return' or 'Deceased'. We later use the testing data to evaluate the performance across all models to find the best model.

For evaluating the best parameter in each model, we also use cross validation method to separate out a validating data set when training each model. So in total, we have an overall testing data set, a training dataset and a validating data set for each model solely. Our criteria for the best model and best parameter are both accuracy and FNR (False Negative Rate).

In order to achieve the best performance of each model, we use the validating data set to test their best parameters. The parameter we tuned for each model are:

- **Logistic:** cutoff;
- **KNN:** k;
- **NaiveBayes:** cutoff;
- **RandomForest:** ntree, mtry, min_child_weight, max_delta_step, cutoff;
- **XGBoost:** max_depth, eta, gamma, subsample, colsample_bytree, min_child_weight, max_delta_step, nround, cutoff;

	Out_hos Accuracy	Out_hos FNR	Out_hos Baleline	In_hos Accuracy	In_hos FNR	In_hos Baseline
Logistic	0.7451046	0.8398533	0.7261925	(bl)		0.817602
KNN	(bl)		0.7261925	(bl)		0.817602
naiveBayes	0.7302092	0.7176039	0.7261925	(bl)		0.817602
RandomForest	0.7556485	0.8398533	0.7261925	(bl)		0.817602
XGBoost	0.7694979	0.72022	0.7261925	(bl)		0.817602

And the number of folds we used for cross validation is 5. After we find the best parameters in the whole training data, which includes the validating data, we use the best parameter to test on the testing data. Table below shows the accuracy, TNR of each model and the baseline of the testing data. For those models whose accuracy is worse than the baseline, we set their accuracy to the baseline with a (bl) mark and therefore they don't have an FNR.

We can see that XGBoost dominant other models, it has the highest accuracy and the second lowest FNR. Compared to the first lowest FNR model naiveBayes, XGBoost has only 0.5% increase in FNR, but 4% increase in accuracy, therefore we will consist on using XGBoost for the future testing. As for the worst model KNN, we thought the poor performance might cause by sparse variable. We supposed that the KNN could not capture the feature of Return because of some hidden reason.

After testing all the models, we notice an interesting insights of data. All the models cannot beat the baseline of the second data set. At first, we try many ways to increase the accuracy, including SMOTE and upsampling, but the accuracy just couldn't get higher. So we change a way to think about it: What if the second model itself is not that good? We then go back to check our KMeans result, and assign the cluster back to the observation, calculate the RETURN mean and the cluster's mean. From the table above, we can calculate that the mean of RETURN is 0.182398, but the cluster mean is only 0.102754. That's an interesting finding that the some of the observations have very similarity in variables and therefore assign to the same cluster, but these observations have different RETURN. These 8% people may have very same characteristic but have different label. We assume that this situation affects the prediction of our model (affect KNN mostly), but so far, it's just a hypothesis and needed to be further verify.

After we get the best model, we use the whole training data set to process cross validation. There are two ways that we use to tune the parameter for XGBoost. The first one is the basic cross validation. By assign a range of numbers to each parameter, each iteration will randomly choose one number for each parameter to run on the validating data. After it find a lower validation loss than the former one, the new parameter will be update and save. The second one comes from an idea that among all the tested models, we notice that naiveBayes model's FNR are the lowest. It behaves well on predicting people that return. So other than the basic XGBoost, we try to find a way to combine these two methods, to see if we can decrease our high FNR and keep the high accuracy. Then we find a tuning method called *Bayesian Optimization of XGBoost Parameters*.

Rather than randomly search the parameter, this method is an optimization scheme that uses Bayesian models based on Gaussian processes to predict good tuning parameters. The best parameter by randomly search give us the same parameter like before, accuracy and FNR remains the same. Bayesian Optimization give us an accuracy 0.763517 and FNR 0.72786, both of the criteria are lower than the randomly search, so our final prediction sticks on the randomly search plan.

Before the final prediction of the data, we finally combine our training data and testing data, training with the best parameters we get. The final model then uses on predicting the unseen testing data. We get a accuracy 78.60% on unseen data, 3.15% higher than the baseline.

Appendix:

Graph:

Pivot Table

ED_RESULT \ ADMIT_RESULT		Inpatient	Observation	Psych Inpatient	Trauma Inpatient	Totals
	99.7%	0.1%	0.1%		0.1%	**100.0%**
AMA	91.4%	2.7%	5.9%			**100.0%**
Admit	1.2%	83.0%	11.1%	1.4%	3.3%	**100.0%**
Admit to External Psychiatric Facility	100.0%					**100.0%**
Admit to UMMS Psychiatry	3.9%	2.0%	2.0%	92.2%		**100.0%**
Arrived in Error	99.0%	1.0%				**100.0%**
Deceased	95.7%	4.3%				**100.0%**
Discharge	98.4%	0.6%	1.0%	0.0%	0.0%	**100.0%**
Elopement	94.4%	1.6%	3.8%		0.3%	**100.0%**
LWBS after Triage	100.0%					**100.0%**
LWBS before Triage	100.0%					**100.0%**
Left prior to completing treatment	98.6%	0.7%	0.6%		0.2%	**100.0%**
Left without signing discharge instructions	99.3%		0.7%			**100.0%**
Observation	0.9%	5.3%	93.8%		0.0%	**100.0%**
Send to L&D Before Rooming (Mom)	75.4%	3.8%	20.7%			**100.0%**
Send to L&D after Rooming	70.0%	10.0%	20.0%			**100.0%**
Transfer	88.6%	3.9%	4.3%	0.7%	2.5%	**100.0%**
Totals	**82.6%**	**9.2%**	**7.5%**	**0.3%**	**0.4%**	**100.0%**