Healthcare Project

Phase I

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

Executive Summary

In present days, hospitals started to track patients' activities and keep the their medical records for having a better understanding about the patients' health status after treatment and the performance of medical treatment. Hence, patients' return rate becomes a crucial criterion to measure how well a hospital is performing. 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.

Before conducting models, we excluded some redundant variables and carefully preprocessed selected variables that might be considered in our future models. Besides, we identified some important variables and figured out the interesting relationships between these variables through visualization, which greatly affect our decisions in building and advancing our models.

We applied logistic regression model, linear regression model, tree classification model and K-Nearest Neighborhoods (KNN) model to predict the patients return status based on our processed dataset. Among these four models, we discovered that the performance of logistic model is best that 75.73% of our prediction is correct.

Going forward we will:

- Attempt to figure out a better way to preprocess the dataset due to its large appearance of missing values.
- Attempt to consider the interaction variables by conducting the visualizations of two variables or calculating their correlation.
- Attempt to re-evaluate the variable selections through backward elimination and forward selection procedures.
- Test additional types of model, such as LDA and QDA.

Data Insights

Structure of the data:

- The data is structured by twenty-six predictor variables and one target variable (RETURN). The dataset includes numeric variables of AGE and CHARGES and other categorical variables. There are three types of categorical variables: level, category and binary. As for the 'level' type, ACUITY_ARR, RISK and SEVERITY are classified into certain levels. For instance, ACUITY_ARR varies from level of 1 to 5 in the order of acuity level. As for 'category' type, ED_RESULT, RACE, ETHNICITY, etc. are classified into different categories. For example, RACE is classified into "Not Hispanic or Latino"," Hispanic or Latino", "Declined to Answer" and "Unknown", these four categories. As for the 'binary' type, CONSULT_ORDER, CONSULT_CHARGE, etc. are defined as indicator variables, meaning the value is either zero or one.

Interpreting the data:

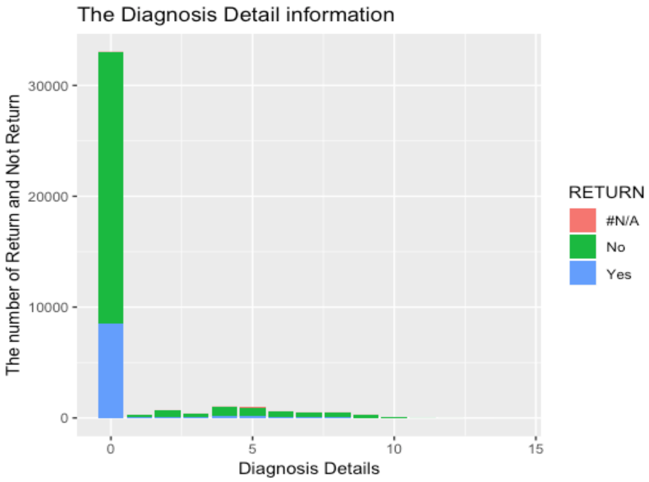
When browsing through the whole data, we observed a large number of blanks in certain columns, especially in Admit_Result, Consult_In_Ed, Risk and Severity, so we tried to figure out the reason why it happened. By understanding the logic behind each column in file "Hospitals_Dictionary", we founded out the relationships between these columns based on domain knowledge in the following:

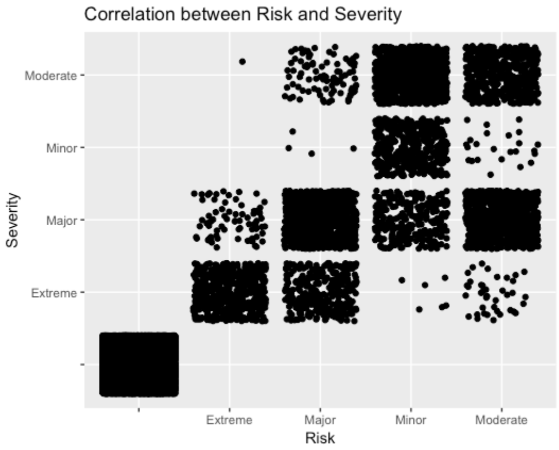
- If a patient was disposed to admission after leaving the emergency room, the ADMIT_RESULT, the RISK and SEVERITY of this patient will be recorded.
- If a patient was disposed to admission after leaving the emergency room, the patient has the probability to request a consultation.
- Only if a patient 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.
- Based on previous three relationships: if a patient was not disposed to admission after leaving the emergency room or left the hospital, the data of "Admit_Result, Consult_Order, Consult_Charge, Consult_In_Ed, Risk and Severity" generally were not recorded or set to be zero.

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.

<u>Preprocessing the data:</u>

1. Removing redundant variables:



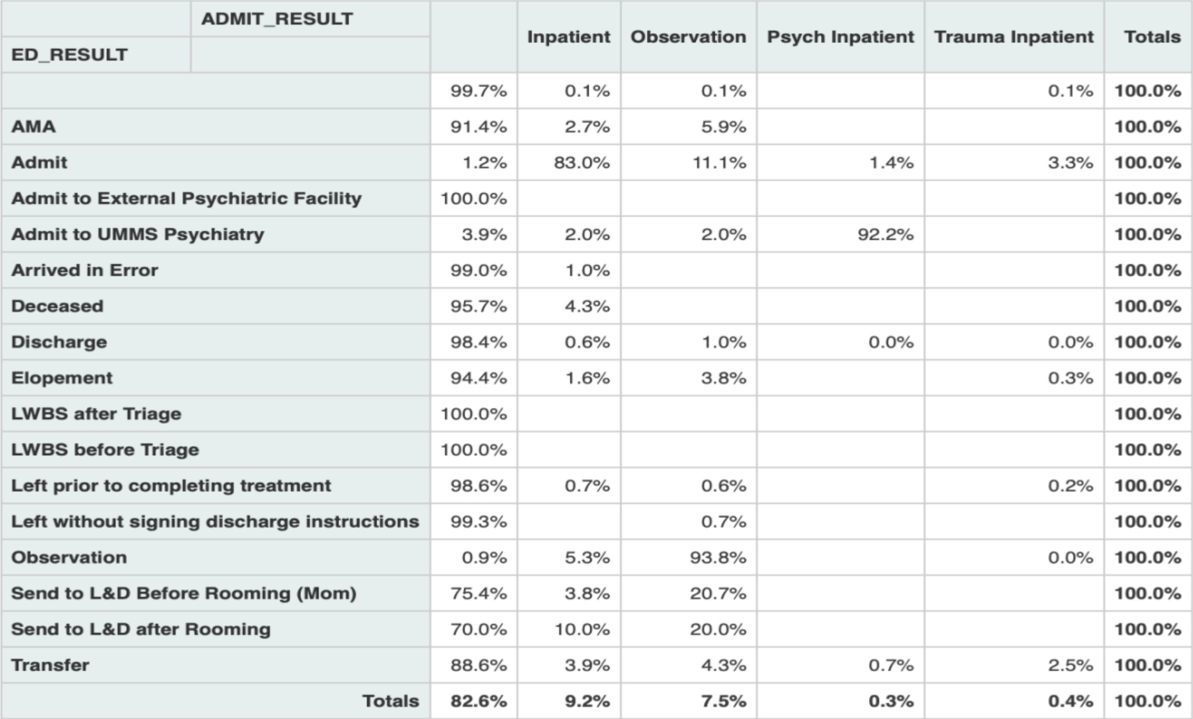


Our first step is to remove the redundant variables based on domain knowledge. We dropped the INDEX and DIAG_DETAILS that does not provide actual information or contains many missing values.

For example, bar plot that shows the distribution of DIAG_DETAILES , which indicates there are more than 30000 values are zero while non-zero values only take a small proportion of total dataset. Moreover, after calculating the difference between time variables, such as patients' hour, day and month between arrival and departure that implies the duration of hospitalization, we noticed that for hours and weekdays are the same. On the other hand, we decided not use the month variables since only 413 out of total dataset 38221 are different in month. As a result, we dropped all the columns that related to the duration of hospitalization.

Since perfect collinearity may exist between RISK and SEVERITY and it is possible to drop one of these variables if they are perfectly correlated, we plotted a graph showing the correlation between these two variables. According to the graph, the points are discrete, showing that the relationship between RISK and SEVERITY is not perfectly collinear. So, it is necessary to keep these two variables as our predictors.

2.Separate Dataset:



	ADMIT_RESULT		Inpatient	Observation	Psych Inpatient	Trauma Inpatient	Totals
ED_RESULT							
		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%**

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 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 this 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.

3. Reorganize categorical and numeric variables

When getting insights of the value in each applied variable, we found out that some of categories in the same feature can be grouped together by the concepts and data sizes of these categories. Thus, we did the following modifications:

- Before splitting the data set:
  - We found there are some minorities categories in ETHNICITY, RACE, and FINANCIAL_CLASS. Base on the meaning of these categories, the minorities were converted to “other”.
  - We separated the categories of ACUITY_ARR by ‘-’ into a number and a string. Using number, we found the frequency of these categories as a normal distribution. Thus, in order to balance the data size of each category, we combine “1-Immediate” and “2-Emergent” as one category, and combine “4-Less Urgent” and “5-Non-Urgent” as one category.
  - Base on the concepts of categories of ADMIT_RESULT, we combined existing categories to “Admit”.
  - In order to forbid large values in CHARGE to effect target variable excessively, we scaled CHARGE to the range 0 to 1 and renamed it as SCALE_CHARGES.
- After splitting the data set by ED_RESULT:
  - We found there are some minorities categories of ED_RESULT can be identified as one category, so we converted “LWBS after Triage” and “LWBS before Triage” to LWBS and adopted the same preprocess approach to other categories in this feature. Finally, categories of ED_RESULT are reclassified into 5 groups: LWBS, Admit, Observation, Transfer and Other.

4. Dealing with missing value and redundant information in variables:

General Rule of dealing with missing value:

- If the number of missing value is very small, we deleted the entire record.
- If the number of missing values is relatively big, we filled the missing value with mean.
- If the missing value is a categorical variable, we filled the missing value with proper content.
  GENDER: There are two columns with blank information are deleted.
  ADMIT_RESULT & RISK & SEVERITY: Missing values are replaced with "Not Admit".
  CHARGE: Missing values are replaced with mean of all CHARGE (without missing value).
  RETURN: We deleted the columns that are blank or the RETURN values are not 0/1.
  ACUITY_ARR: This variable mainly contains two part: digit and its corresponding acuity level. For example, 3-Urgent. We separate the digit and the acuity level, but only extracted the digits and converted them as factor variable. Blank columns are filled with 0 because we would like to classify them as "other".
  CHARGE: Missing values are replaced with mean of all CHARGE(without missing value).

5. Converting text data to corresponding number:

Except the numeric variables AGE and CHARGE, categorical variables were converted to digit value but still used as factors. Since we would like to apply KNN as one of the models trained, only numeric value can be accepted by KNN. Converting all the categorical predictors into digital value can make sure all the retained variables could be used in KNN model.

Modeling Insights

So far, we've tried logistic, linear, tree and KNN in both datasets out_hos and in_hos. Based on the consideration of needs in evaluating our models' performance, we split a validation data set out of each data set. After extracting all the rows that have 'No return' or 'Deceased', we used 70% of the whole data set as training data set, 30% as validation data set.

For evaluation, we look at the accuracy. For the best performance, we chose different cutoff value in different data set. The baseline of validation data set of out_hos is 0.7391642 and in_hos is 0.817913. Below is the table of the results we got from our model. The model for out_hos contains the variables except ADMIT_RESULT, CONSULT details, RISK and SEVERITY, model for in_hos has all variables on the other hand.

	Out_hos accuracy	out_hos baseline	In_hos accuracy	In_hos baseline
Logistic	0.757344	0.7391642	0.8187579	0.817913
Linear	0.7551269	0.7391642	0.817913(bl)	0.817913
Tree	0.7391642(bl)	0.7391642	0.817913(bl)	0.817913
KNN	0.7474781(k=91)	0.7391642	0.817913(bl)	0.817913

We first looked at logistic regression. For the out_hos data set, the model shows that nearly all the variables are highly significance, but SAME_DAY and DIAGNOSIS are not significant at all. The number of SAME_DAY=1 has only 276 observations compare to more than 20,000 observations of SAME_DAY=0. But things get worse in the second data set. AGE, SCALE_CHARGES are the only two significant variables while other variables are basically insignificant. And because the number of ‘Not Admit’ class in SEVERITY is the same as RISK, the row of ‘SEVERITY Not Admit’ in the summary of the model becomes all NA, which means they overlap each other. So we may continue identifying the useful variables and have a more reasonable classified method in grouping the small amount categorical variables in future work, and therefore have a more in_depth recognition on our criterion in splitting the data set.

About the linear regression model, it has a quite similar result and model as logistics model using same cutoff in out_hos data set. But the second is worse than using the baseline. So we will not consider it in our future implementation.

In tree model, the models for both data set give us 0 for all terminal nodes. Tree model doesn’t give us more predicting power than the baseline. We thought the poor performance might cause by sparse variable. We supposed that the tree could not capture the feature of Return(‘Yes’) and the number of 0 in each nodes is huge. Hence, we turned to regression tree to see whether our assumption is correct or wrong. We noticed that, in the regression tree, the tree shape is similar and the number in the terminal node are all below 0.4. The result seemed match our assumption. Therefore, all the validating data would be predicted as 0 and we got the baseline accuracy. We will reconsider the way that we utilize this model until we have a deeper understanding in why this kind of situation will happen.

As for the kNN, we first gave the categorical variables a number instead of character in each column and then changed them to numeric data. Now we can use all the numerical data to do our kNN. After testing different values of k, we chose the best one in model. Unfortunately, kNN isn’t that good.

Conclusion and future work

Above all, logistic regression has the highest accuracy. Although using the whole data set to train the model can get a higher overall accuracy than out_hos, the difference between model accuracy and baseline accuray in out_hos is higher than in the whole data set. We notice that all models in in_hos have really bad performances, worse than baseline accuracy. It is a warning to us that we should consider if it is necessary to split the data or is a good choice we split the data by ED_RESULT.

In the future work, we will focus more on identifying the useful variables and find out a more reasonable classified method in grouping the small amount categorical variables by discovering more valuable information from analysis. Therefore, a more in-depth recognition on our criterion in splitting the data set will be realized. Also, we will try additional types of model such as QDA and LDA.