COLUMABIA UNIVERSITY - JOHNSON&JOHNSON

AutoML Prediction Machine of Adverse Outcomes Following Hip Fracture Surgeries

Fall 2020 Capstone Project - Final Report

Mike Wang
Zining Fan
Qiang Zhao
Siyuan Wang

Contents

1 Introduction

The AutoML Prediction Machine Capstone Project is sponsored by Johnson Johnson, with a primary goal to focus on hospital level and predict the risk of readmission, emergency room visit, and mortality of the patients within the 90 days after receiving the hip fracture surgery. We would like to build an automated search engine which generates the hospital rate of readmission, emergency room visit, and mortality calculated by the patients' prediction outcome. For this semester, we would only focus on readmission due to the limitation on time and computing power. The project has great potential value because hip fractures are a major public health burden and Johnson & Johnson are working with hospitals to reduce complications after hip fracture surgeries.

The work of our team will be a continuation of what has been done in the previous two years. The team of last year showed us two approaches on tackling this problem. The first approach was based on the population model with the assumption that all the hospitals share the same model. The second approach was based on the hospital model with the assumption that each individual hospital has its unique model. Then they compared the results for both approaches and came to the conclusion that hospital models can predict emergency room and readmission outcomes better than an entire dataset based model, but could not predict mortality better. Last year's result is mainly focused on broad comparison between the two different models but this pattern is not consistent for all the hospitals. We would extend their work to automatically selecting the best model for each hospital that gives the highest evaluation score with acceptable computation time.

Our ultimate goal is to predict the risk of readmission of the patients within the 90 days after receiving the hip fracture surgery and then use the prediction to generate the rate of readmission for each hospital. To achieve it, we applied a two-step procedure. For the first step, we developed models that can accurately predict the individual patient's risk of readmission through three approaches including population model, hospital model and ensemble model. The second step is to use the best approach from above to generate the readmission rate for each hospital, and to compare it with the readmission rate generated from the regression models using hospital level data.

2 Dataset and Exploratory Data Analysis

For this project, we use Medicare data along with other public data such as, census. Our dataset has over 430,000 observations, and each observation is a case of a unique patient who has done hip fracture surgery. For each observation, it has 247 variables, covering feature types from hospital information, personal information, medical record, diagnosis and procedure of surgery as well as other follow-up information. There are over 3000 hospitals in total, but most of them have observation numbers below 300 as shown in Figure 1. Our target variable, readmission flag, is binary with 0 indicating the patient will not be readmitted within the 90 days after receiving hip fracture surgery and 1 vice versa. Figure 1 also shows that the target variable is highly imbalanced in our dataset with the population mean around 23% positive rate.

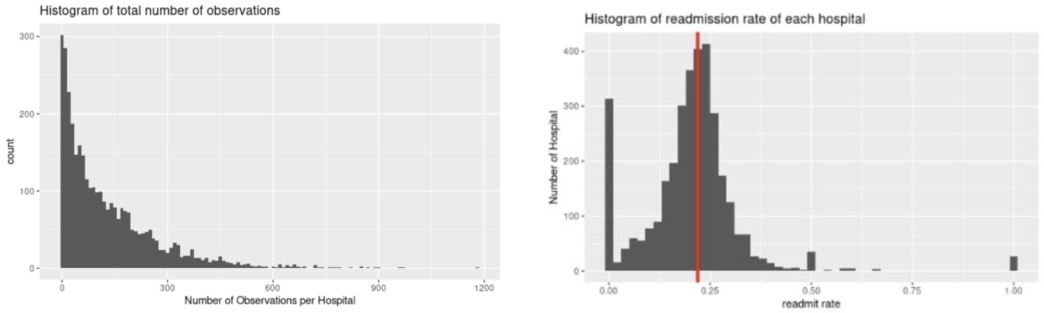

Figure 1: Hospital Size and Readmission Rate

We came up with an assumption that there might exist a temporal dependency for the readmission rate, for which it would decline over time in our dataset as medical technology will improve through time. We plotted the readmission rate by date and conducted ACF and PACF tests as figures below.

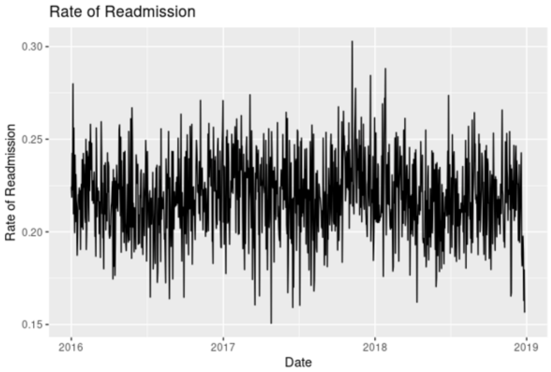

Figure 2: The Time Series Plot of Readmission Flag

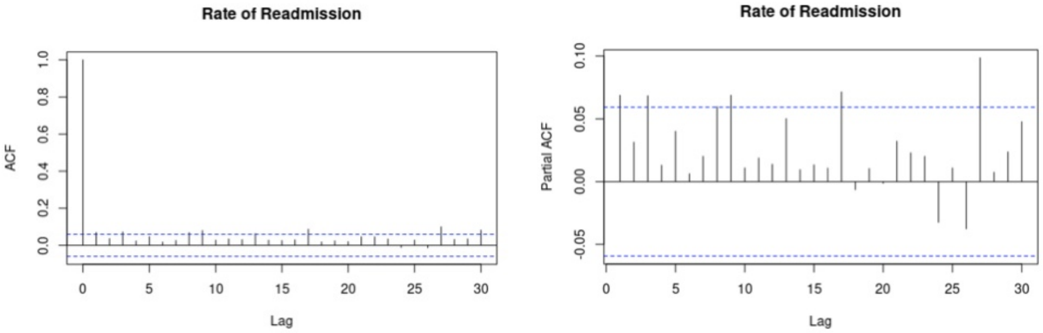

Figure 3: ACF and PACF plots of Readmission Flag

We could not see any declining trend, which means no significant temporal dependency is detected on population level. But we could not rule out the possibility of temporal dependency on hospital level. So we added time related features such as month and day of week in our model to generated feature set combinations for further testing.

For the data prepossessing task, we removed features with leaking information and collinearity, which gave us 137 features left. Since the logistic regression model only takes in numerical features, we need to apply encoder to categorical features. However, some features have too many classes. To avoid this issue, we regrouped them through grouping by type, keeping the top class and dropping the variables with repeating information. We applied an encoder approach that combines both one hot and target encoder with a threshold of 5 classes. For categorical features with less than 5 classes, we used one hot encoder, and for the rest we used target encoder. The threshold of 5 classes was determined by the limitation of our system memory.

3 Modeling Approaches & Results

To achieve our goal of predicting future readmission rate for each hospital, we followed below two steps. First, developed models that can predict whether a patient would be readmitted within the 90 days after receiving hip fracture surgery, given the patient's information and hospital's information. To develop the models, we tried below three approaches: population model, hospital models, and ensemble models. Then we selected the best approach from above to make predictions on the patients' risk of readmission in the prediction horizon. Subsequently, we calculated the future readmission rate of each hospital by taking the average of the patients' readmission indicators belonging to that specific hospital. Then we compared it with the readmission rate generated from using the regression models based on hospital level data.

3.1 Develop Classification Models on Patient Level Data

3.1.1 Sampling

After categorical feature transformation and data encoding, we further applied data sampling to facilitate the training process and model selection. We first removed hospitals with less than 100 observations as they were ineffective in the model construction and prediction. Then we removed all hospitals with abnormal distribution in the target variable such as hospitals with 0% and 100% positive outcomes. The selected hospitals also must have over 10 negative and positive values for target variables due to Medicare data rule of not showing less than 11 data entries at a time. Furthermore, considering the geographical elements that might affect our results, we also sampled the observations by state to guarantee that all states are covered in the general observation pool.

3.1.2 Population Model

Population model means that we are going to use the whole train dataset to train our model. Three models are applied to our data, regularized logistic regression, random forest and XGBoost. Regularized logistic regression is easy to implement and efficient to train.

Random forest and XGBoost are boosting methods, which have high accuracy and prevent overfitting. Then, each model will be trained to 4 different feature sets as shown below, base set, Boruta set, base set with time features, Boruta set with time features. The base set includes all the features after initial data cleaning. Boruta set is the features selected by the Boruta algorithm. Time features are the features we added to the data set that relates to time, such as month, day of week, and number of days since 2016 before admission. Then we compared their performances and picked the best model.

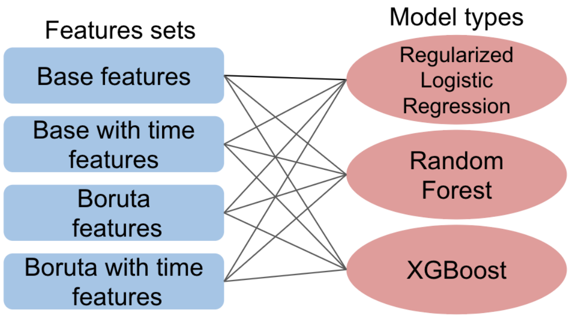

Figure 4: 12 Combinations for Population Model

In order to overcome the issue that our data set is highly imbalanced, we tried different sampling methods. First, we started with 3000 observations in the population model due to the limitation of RStudio server memory. The results are shown in Figure 5.

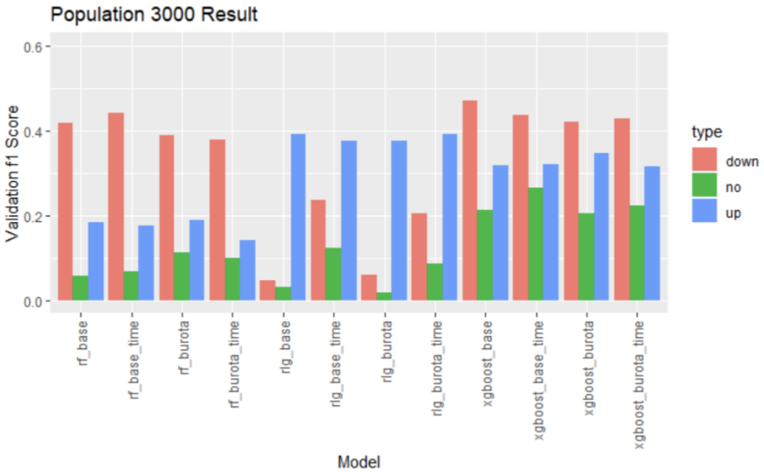

Figure 5: Bar Chart of Population Model Result of 3000 Cases

We applied downsampling, upsampling and no sampling to the train set when training the models. According to Figure 5, upsampling enhances performance in regularized logistic regression since the model generally performs better with more data. However, the downsampling has generally better performance in random forest and XGBoost models since tree models are more likely to be affected by noises and downsampling could remove some noises in the training set. No sampling always does worse. Also, downsampling requires less memory to train than upsampling and no sampling, therefore, from a general perspective, downsampling is the best approach.

Next, we want to check whether one hot encoder improves model performance or not. We selected 20 percent of the data and applied downsampling. We created two feature groups, one used one hot and target encoders after the categorical features regrouping, the other group only applied target encoder on categorical features before categorical features regrouping. The result is shown Figure 6.

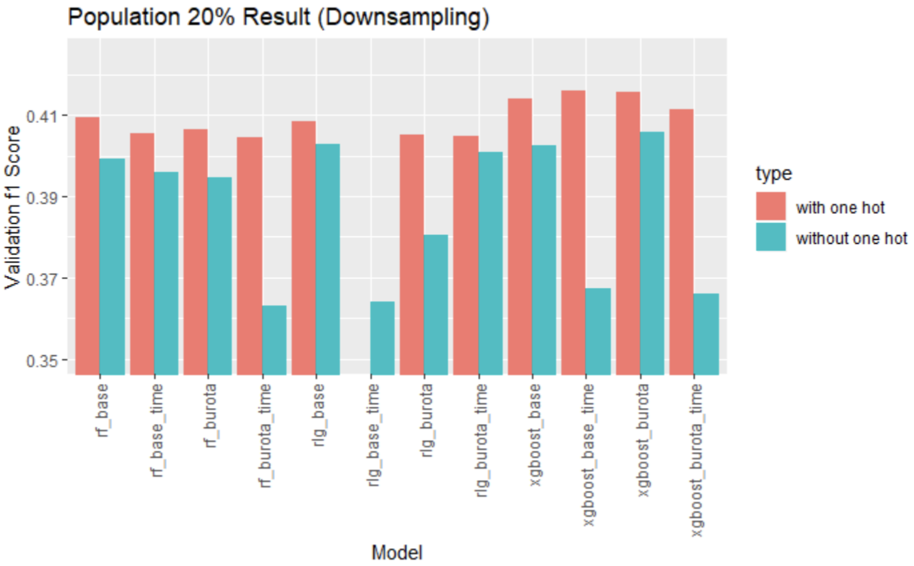

Figure 6: Bar Chart of Population Model Result of 20 % Cases

It is clear that data with one hot encoder always performs better. Although only the regularized logistic regression with base set plus time features "rlg_base_time" performs worse with one hot encoder, its overall performance is worse than others so we would not choose this combination as our final population model. For the rest of the model combinations, we did not see one combination that significantly outperformed the other. Since Boruta feature selections, regularized logistic regression, and XGBoost take much training time and more memories, we chose to use random forest with base set plus time features (with both one hot and target encoders) as our final approach on population level model.

Finally, we were able to train a random forest model with base set plus time related features on 100% data with downsampling on the training set. We did the train/validation/test split by chronological order and split the data proportional with ratio 0.7: 0.15: 0.15. Split data by chronological is significant in this case, since we want to make sure our model can be used to predict future readmission rate by current information. After data splitting, we applied downsampling and one hot encoder to our data set. The following is the detail of this. The training set originally contains 244,352, and it reduced down to 110,372 observations after downsampling. The validation set contains 52,369 observations with 11,819 positive cases. Since downsampling and random forest will get different outputs on the same dataset every time. We train our model on several trails and select the best one. The result is shown in the Table 1. Finally, we select the trail 4 model as our final population model since it has the highest validation F1 score.

[table]xcolor

Trail	Cross Validation F1	Validation F1	Log Loss	AUC	Accuracy
1	0.6073	0.3670	0.8200	0.5300	0.3200
2	0.6085	0.4066	0.6850	0.6379	0.5544
3	0.6058	0.3714	0.7868	0.5620	0.3585
4	0.6062	0.4075	0.6798	0.6398	0.5568
5	0.6063	0.4068	0.6879	0.6377	0.5537
6	0.6058	0.4038	0.6907	0.6339	0.5459

Table 1: Validation Scores

3.1.3 Hospital Model

There are 3378 hospitals in total after we group the Medicare data by hospital. After grouping by hospital, we removed features with 0 variance within groups, after which there are 87 features left. For each hospital, we ordered the data by time and then used 70% as the train set, 15% as the validation set and 15% as the test set. On account of the train set size, we excluded the hospitals with less than 100 observations. In addition, to make sure that the validation set has at least 10 positive observations and at least 10 negative observations then our results would not be biased, we only selected hospitals with at least 67 positive and negative observations.

After this, we have 367 hospitals left. The size of each hospital is shown in Figure 7. The majority of the hospitals have 200 to 500 patients' information.

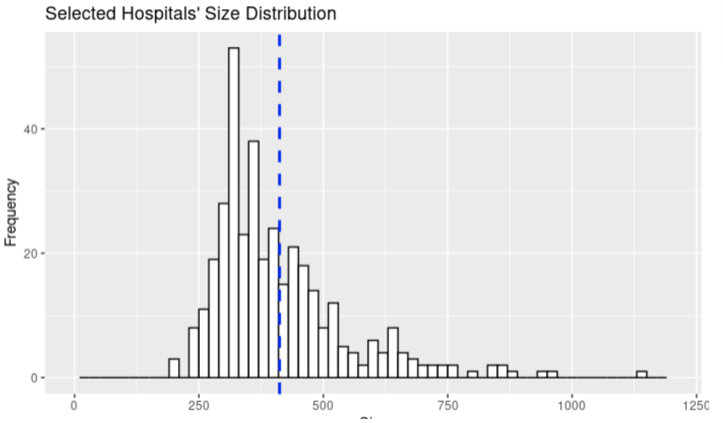

Figure 7: Histogram of Hospital Size after Data Cleaning

Then, we upsampled the train data for each hospital. For each hospital, we ran 12 models as shown in Figure 8, which come from 3 types of models: regularized logistic regression, random forest and XGBoost; and 4 types of feature sets: Boruta features, Boruta features with time related features, Kbest features and last year's top 9 features.

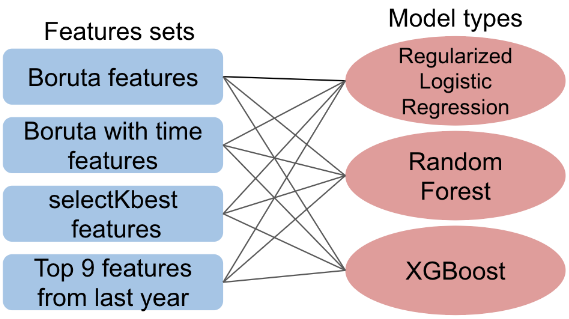
Features sets
Model types
Boruta features
Boruta with time features
selectKbest features
Top 9 features from last year
Regularized Logistic Regression
Random Forest
XGBoost

Figure 8: 12 Combinations for Hospital Model

Kbest features set is features selected by the selectkbest approach using the "FSinR" package with chiSquared method. We set k equal to the number of observations in the upsampling train set divided by 20. Last year's top 9 features are the most commonly used features across all hospital models after the Boruta algorithm from last year's report. Since the Boruta algorithm tends to select more features than needed after upsampling, we included the last two sets to avoid the potential overfitting of the Boruta method. In total, we ran 12 combinations for each hospital and selected the best model for each hospital based on the F1 score on the validation set.

Figure 9 shows the results of hospital models for 367 hospitals.

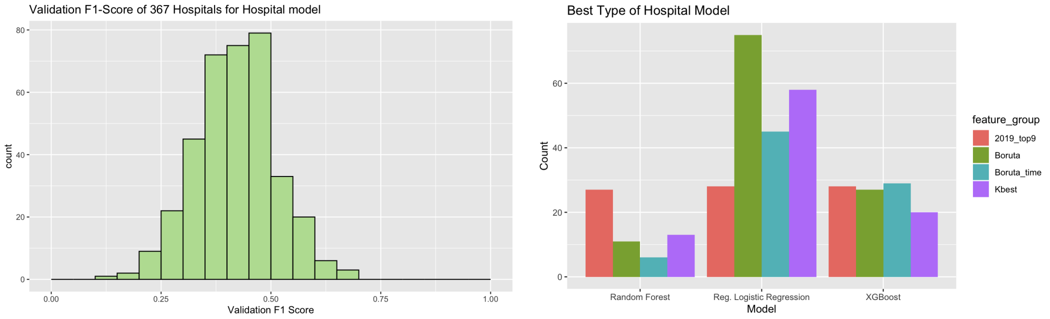

Figure 9: Results of Hospital Model

The left plot is the histogram of the F1 score on validation sets, and we could find that the majority of hospitals have the F1 score between 0.3 and 0.5. The right plot is the bar chart of the select best model type from 12 models, and it shows that regularized logistic regression with Boruta features has the highest frequency among all 12 models.

3.1.4 Ensemble Model

Ensemble models are the combination of population model and hospital model. For each hospital, we assigned α to the population model and $1 - \alpha$ to the hospital model, which includes the 12 combinations the same as the hospital model above. For each ensemble model, we selected the best α based on the F1 score on the validation set. Thus, we have a total 12 ensemble models for each hospital and we selected the best one based on F1 score on the validation set.

Figure 10 shows the results of ensemble models for 367 hospitals.

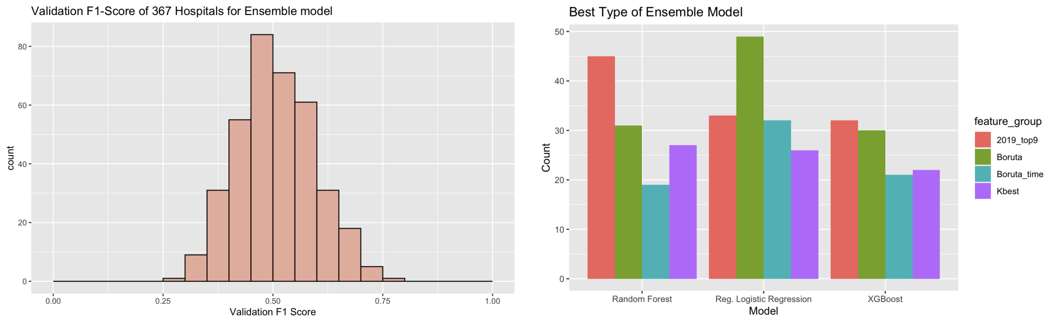

Figure 10: Results of Ensemble Model

The left plot is the histogram of the F1 score on validation sets, and we could find that the majority of hospitals have the F1 score between 0.4 and 0.6. The right plot is the bar chart of the select best model type from 12 models, and it shows that the type of models does not influence the performance. This is reasonable because we have incorporated the effect of the population model, which evens out the effect of the different hospital models.

3.1.5 Comparing Modelling Approaches

Figure 11 shows the density graphs of the result validation F1 scores density graphs for all three approaches. The ensemble model overall has higher validation F1 score than population model and hospital model. The density graphs for population model and hospital model are highly overlap each other, we could not make a clear conclusion on which models performed better.

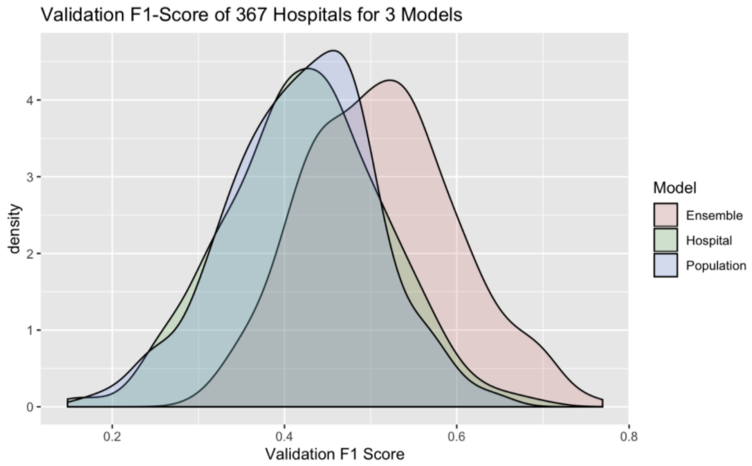

Figure 11: Density Plot of Comparison of Three Models

Thus, instead of looking at overall level, we plot the scatter plot in Figure 12 with scores for each individual hospital. Each vertical line represents the results of a single hospital, the blue dot is the score for the population model, the green dot is the score for hospital models, and the red dot is the score for the ensemble models. As we can see from the plot, the ensemble models always have the highest score for every hospital. However, for some hospitals, the population model has better scores than the hospital models, and vice versa. This is reasonable since by design, the ensemble models always pick the best α for the combination of population model and hospital models that gives the best validation F1 score.

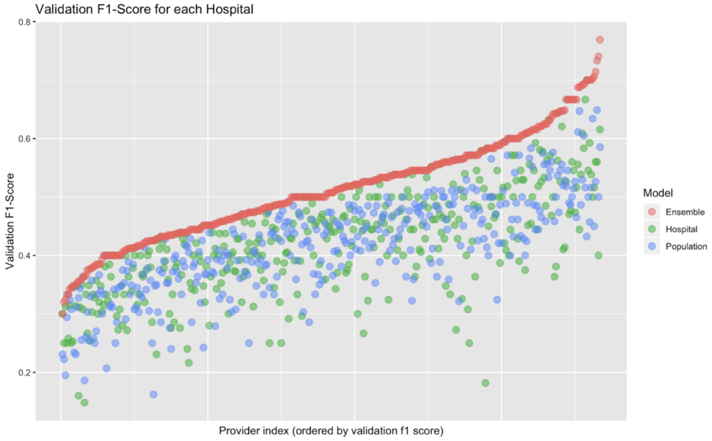

Figure 12: Scatter Plot of Comparison of Three Models

Thus, in order to further justify whether the ensemble method is the best approach, we gathered all 367 hospitals validation sets into one accumulative validation set and all 367 hospitals test sets into one accumulative test set, then applied all three approaches on them.

For the population model, we would just use a single model for all the patients in the set. For hospital models, we would use the best hospital model for each patient with respect to their hospital. For ensemble models, we would use the best ensemble model for each patient with respect to their hospital. We also added random guessing, which is randomly selecting between 0 and 1 through a uniform distribution, as the baseline for comparison. The results are in the Tables 2&3.

Model	Validation F1	Log Loss	AUC	Accuracy
Population Model	0.4196	0.6822	0.6404	0.5623
Hospital Model	0.4211	0.8201	0.6074	0.6442
Ensemble Method	0.5007	0.6497	0.6603	0.6900
Random Guessing	0.3210	0.9915	0.5504	0.5037

Table 2: Validation Scores

Model	Validation F1	Log Loss	AUC	Accuracy
Population Model	0.4197	0.6891	0.6326	0.5543
Hospital Model	0.3331	0.8905	0.5479	0.5894
Ensemble Method	0.3705	0.6784	0.5945	0.6079
Random Guessing	0.3307	0.9892	0.5073	0.5760

Table 3: Test Scores

From the above tables, we could find that the ensemble model has the best performance among all in the accumulative validation set. However, the population model has the best test F1 score among all. We also discovered for the hospital and ensemble models, there is inconsistency between the validation F1 score and test F1 score.

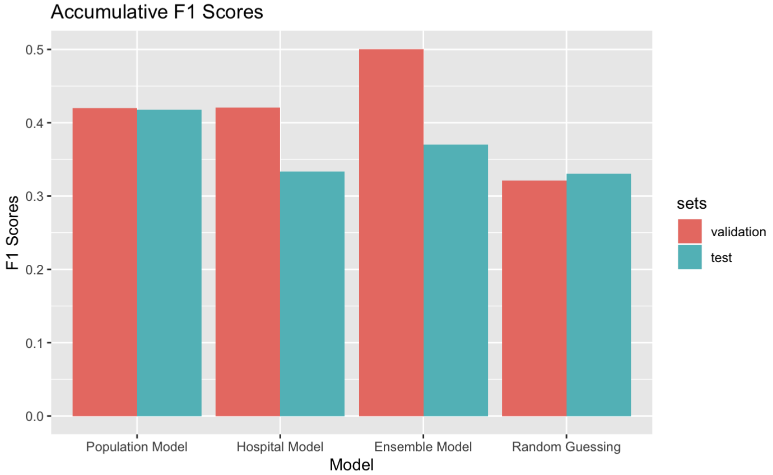

Figure 13: Bar Chart of Accumulative Scores

As shown in Figure 13, their test F1 scores are much lower than their validation F1 scores. Also, their test scores are just slightly better than random guessing's. This shows that our hospital models and ensemble models are biased and not generative. The finding is not too surprising as the majority of our hospital contains 200 to 500 observations, which is not enough to create a generative model. Therefore, our best approach is to use the population model, where its validation and test scores are consistent, and we can see a significant improvement from random guessing.

3.2 Calculate Next Year Hospital Readmission Rate

3.2.1 Generate Readmission Rate Prediction

After we selected the population model as the best approach to prediction the readmission indicator for each patient, we then applied the model to predict the patients' risk of readmission in the prediction horizon. Our data is patient level data and we only have previous years' records, so there is no next year's patients' information. Thus, we took the exact copy from last year's data except for the readmission indicator as next year's data. Then we obtained the next year patients' risk of readmission through predicting with the population model. Finally, we calculated each hospital's next year readmission rate by taking the average of that hospital's next year's patients' risk of readmission.

3.2.2 Regression Approaches

In order to solve the problem of not having next year's patient's information, we came up with a different approach. Instead of looking at patient data, we reorganized it into hospital data by taking the average of the patient information and group by hospital and year. When taking the average, for a continuous variable, we would take the average directly. For example, the age of the patient would transform to the average patient's age at the hospital. For categorical variables, we applied one hot encoder first and then took the average of all the dummy columns. For instance, for the diagnosis code containing classes 'S', 'M', and 'T', after transformation, it will become the percentage of the patient with the diagnosis code of 'S', 'M', and 'T' respectively. Since our data contains records from 2016 to 2018, for each hospital, we can obtain its last year average information, last year readmission rate, and next year readmission rate. Then we can train a regression model with input as last year average information and last year readmission rate, and output as next year readmission rate.

After removing hospitals with less than 100 observations and less than 10 negative or 10 positive classes for the purpose of excluding extreme cases, we have around 3000 hospital average data after the transformation. Then we apply train, validation, and test split with 0.7, 0.15, and 0.15 ratio with stratify on state as we want to make sure our sets have the same distribution in state. Then we tried linear regression models with ridge, lasso, and elastic regularization along with random forest and XGBoost regression models. Finally, Figure 14 shows the root mean square errors on the validation set using the population model on patient level data and all the regression models on hospital level data. We also added in the results of using last year readmission rate and the mean of the next year readmission rate in the training set as prediction for the baselines.

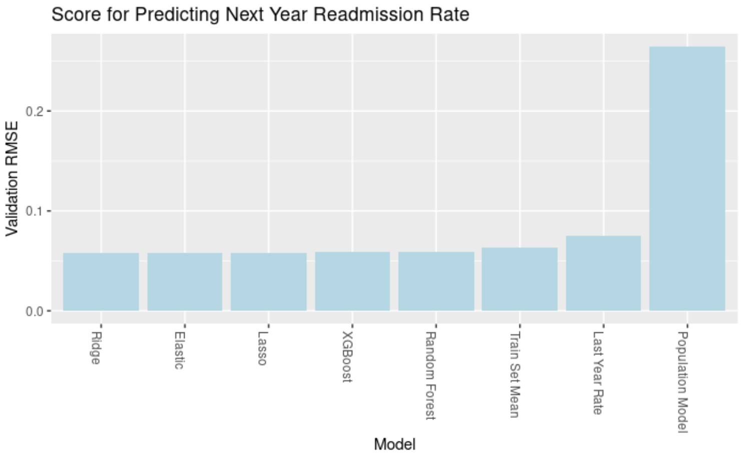

Figure 14: Bar Chart of the Prediction RMSE on Validation Sets

As you can see from Figure 14, the population model has the worst result of over 0.25 RMSE. Using the last year readmission rate as prediction has a RMSE around 0.07, and train set mean has a RMSE around 0.065. All the regression models have the best and similar RMSE around 0.06. This is expected as our population model used last year's patient information as next year's and the population model's F1 validation score is around 0.4. It would perform worse than using last year's readmission rate. Since our input for regression models contains last year's readmission rate as a feature, it is expected that they will have better performance than using last year's readmission rate.

4 Conclusion

In conclusion, we have developed three approaches, population model, hospital model, and ensemble model, to predict the patient's risk of readmission. The population model is the best approach among the three as hospital model and ensemble model are biased due to the small data set. However, the classification on patient level data does not help to predict future hospital readmission rate if using exactly the same past data as prediction horizon data. It could only do as good as just using last year's readmission rate as prediction even if we can 100% predict the readmission flag correctly. Instead, using the regression method to directly predict the hospital's future readmission rate on hospital level data is the better approach.

5 Future Work & Ethical Concerns

One could continue exploring other regression models and feature sets combinations that give the better result. Then after obtaining the best result, one could create a search engine through Shiny, which the user could get the prediction on hospital readmission rate when entering the information of the hospital and its patients' data.

Since the data consist of medical information of the patients, one should be aware of not exposing the patient information to third parties for privacy concerns. One advantage of using regression models over hospital level data is that one would have less chance of exposing individual patient information to the public since the individual patient information is preprocessed into hospital level data prior to model training.

References

[1] *6.2 - Binary Logistic Regression with a Single Categorical Predictor, PennState Eberly College of Science*, The Pennsylvania State University, 2018, online.stat.psu.edu/stat504/node/150/.

[2] *Assumptions of Logistic Regression*, Statistics Solutions, 2020, www.statisticssolutions.com/

[3] Bliss, Thompson, et al. *Comparing Population Level and Hospital Level Predictions of Adverse Outcomes Following Hip Fracture Surgeries*, Columbia University, 2019.

[4] Kuhn, Max, *The caret Package*, 2019, http://topepo.github.io/caret/index.html

[5] Kumar, Naresh. *Advantages and Disadvantages of Random Forest Algorithm in Machine Learning*,The Professionals Point, 2019, theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-random.html.

[6] Kumar, Naresh. *Advantages of XGBoost Algorithm in Machine Learning*, The Professionals Point, 2019, theprofessionalspoint.blogspot.com/2019/03/advantages-of-xgboost-algorithm-in.html.

[7] Kursa, Miron B. & Rudnicki, Witold R. *Feature Selection with the Boruta Package, Journal of Statistical Software, Foundation for Open Access Statistics, vol. 36(i11).*, 2010, https://ideas.repec.org/a/jss/jstsof/v036i11.html