## ABSTRACT

Hospital readmission is now considered as one index to measure the quality of healthcare services one hospital provides, since its cause can be broken into improper planning, inadequate resource allocation etc. In fact, many documents provide the statistics that a majority of readmissions are actually preventable. In this project, we concentrate on the case of diabetic patients and combine both carefully chosen feature engineering steps and different machine learning models such as Random Forest, Gradient Boosting, CNN to predict the 30-day readmission status of patients. Balanced accuracy is the main evaluation index and the prediction results of the highest balanced accuracy model can be used by hospitals to allocate resources more properly.

# 1. INTRODUCTION

Hospital readmission describes the situation of patients who have been discharged from hospitals but admitted later in a given period (Strack, et al., 2014). This situation can be caused by various reasons and has an impact on both patients and hospitals. For patients, hospitalization is very stressful, and the procedures taken may lead to further readmission, which increases the treatment cost for such patients. In the United States, it has been reported that 76% of 30-day readmissions in the records could be avoided (Commission, 2007). In addition, hospitals have to put resources in activities caused by unpredicted readmissions, such as allocating medical staff, managing external care providers, and conducting necessary cleaning. Therefore they want to find methods to identify the patients who can be given available resources to prevent readmission and reduce additional costs in the future. Readmission rates are also used as a quality benchmark for health systems, so low admission rates also help increase hospitals' reputations (Chopra, Sinha, Jaroli, Shukla, & Maheshwari, 2017).

In this project, we propose using machine learning models to predict the readmission of diabetic patients in a 30-day interval. Previous scholars have introduced several existing models (Mingle, 2017), which provide insights for our project. We continue on the topic and apply different processing and engineering steps to improve the prediction performance for our proposed models. Detailed descriptions of such steps and models will be presented in the methodology section later, followed by the discussion and interpretation of our results. Limitations and obstacles have been discovered during the project, and they will be discussed together with possible future works.

# 2. RELATED LITERATURE

## 2.1 Literature Review

The dataset used in the project is available to the public from the University of California, Irine, Machine Learning Repository. In terms of related work, plenty of papers and conference articles have worked on the Diabetes 130-US hospitals for years 1999-2008 Data Set in the recent ten years. This project aims to predict the readmission rate of Diabetes patients.

Some study papers used this dataset for the same purpose. Many pieces of literature try to use different machine learning models and statistical methods to predict readmission. Many of them not only pursue predictability but also want a certain degree of interpretability.

Some of them start with different processing procedures. For instance, in the data set, the attributes of the 'readmission' class indicate whether the patient is readmitted to the hospital, that is, three categories: ‘no’ (that is, no readmission),' <30’ (that is, readmission within thirty days), and '>30 '(That is, readmitted to the hospital after thirty days). Jia & Tian (2019), Ramirez & Herrera (2019) and Hempstalk & Morda (2016), all these authors transfer the classification into a binary classification problem by merging class labels 'NO' and '>30' into one class and keeping the '<30' intact. However, Alloghani et al. (2019) keep the three labels intact, treat the dataset as a multi-class problem, and focus more on the features' interpretations.

For the missing values, Qian and Hongwattanakul (2016) suggest that there are multiple features with a very high percentage of missing values that need to be dropped, such as Weight (97% missing values), Payer code (40% missing values), and Medical specialty (47% missing values). In addition, the Encounter ID and Patient Number as Indexes are also removed for excessive repetition of the same value. Jia & Tian (2019), Ramirez & Herrea (2019), Sharma et al. (2019), and Shibly et al. (2021) all remove the features that are mentioned above. Unlike these authors, Tamin & Iswari (2017) deal with

missing values by replacing the missing values with mean values for numerical type attributes and mode values for the nominal type attribute.

In the data pre-processing, ICD9 codes in the dataset indicated the primary, secondary, and tertiary diagnoses as suggested in the paper of Strack et al. (2014). To scale features for the machine learning models, Sharma et al. (2019), Hempstalk & Mordaunt (2016), Shibly et al. (2021) and Bhuvan et al. (2016) all regroup the ICD9 codes into 9 or 10 groups and replace the codes by binary variables. Ghazo (2019) reconstructed the original data from compressed data without losing the information with dimensionality reduction and numerosity reduction methods. In addition, the problem of imbalance between the readmitted class and non-readmission class has been addressed in many papers. Many authors address the problem with the same oversampling method named 'SMOTE' introduced by Nitesh et al. (2002).

Chopra et al. (2017) carried out some modeling evaluation utilized with Logistic Regression, Support vector machines, Decision Tree, Simple Neural Network, and Recurrent Neural Network for the identical classification task. The RNN model shows the highest predictability among all the models, and they were able to achieve maximum accuracy of 90.3 percent. However, after the result comparison has been conducted between our findings and theirs. One possible error they made could be that they split the dataset after processing with SMOTE oversampling, resulting in inaccurate results. Same as Ahamd et al. (2018), they also conducted a SMOTE oversample of the whole dataset before the train test split, resulting in unreliable results. The proper way to resample the dataset is to split the dataset into training and testing sets first and only resample the training test. Tarig (2016) used WEKA for data mining and used the Naive Bayes, Logistic Regression, and J48 for the prediction, which achieved the highest accuracy of 74.4% in their paper. Furthermore, Diviya and Rathipriya (2020) utilized an HLR Algorithm as a hybrid logistic regression which outperforms other algorithms and the advantage of this method in healthcare helps to analyze patients at the early stage and reduce the readmission risk. Damian (2017) tested with several models, including Random Forest, Extreme Gradient Boosted Trees

Classifier with Early Stopping, SVM classifier, Balanced Random Forest Classifier, Gradient Boosted Trees Classifier, etc in the highest AUC of 0.67.

# 1. RESEARCH/PROJECT PROBLEMS

## 1.1 Research/Project Aims & Objectives

In many cases, hospital readmission costs take the most portion of medical expenditure on the entire medical system, and it has been recorded that in 2011, hospitals in the U.S. spent $41.3 billion on unplanned readmission cases (Shinkman, 2014). Hospitals still need to allocate resources for such unplanned readmissions, making the operations even harder since they only have limited resources. This project focuses on the readmission situation of one specific disease, diabetes since it is the costliest disease in the U.S. (Waters & Graf, 2018). To be more specific, this project explores the patterns based on the medical records of one inpatient to predict whether this patient will be readmitted again after discharge. Readmission typically means this patient has not been given enough treatment during this stay. Such early judgment helps hospitals to allocate resources and staff more wisely so that they can put priority on patients who need more treatment. By taking advantage of such prediction and resource allocation, hospitals will achieve a better quality of medical services and lower expenditures.

## 1.2 Research/Project Questions

This project aims to explore and evaluate the performance and results of several machine learning models with respect to the prediction of diabetic readmission for patients who have been discharged within 30 days. In addition, this project explores the importance and relationship of key features and attributes within the medical records regarding diabetic readmission.

### 1.3 Research/Project Scope

In order to proceed with the project, a dataset of diabetic patients will be identified first, followed by the processing and exploratory data analysis on this dataset. Feature engineering steps will be conducted to improve the quality of the raw dataset. Different models will be implemented to explore the performance and results of the readmission prediction and in this project, we focus on Naïve Bayes, Logistic Regression, Random Forest, Support Vector Machine, Gradient Boosting, Convolutional Neural Network and XGBoosting models. They will be tuned with their best parameters. Accuracy, precision, recall, F1 score and balance accuracy will be the evaluation metrics and the dominant one we choose is balanced accuracy. Based on this, the best model will be chosen and discussed as well as the reflection and limitations of this project.

## 2. METHODOLOGIES

### 2.1 Methods

#### 2.1.1 Models implementation

To get a well-performed predictive model, we tried seven different models: logistic regression, random forest, Naive Bayes, support vector machine, CNN, gradient boosting, and XGboosting. The best parameters for each model are selected by grid search with five-fold cross-validation.

(1) Logistic regression

Logistic regression is a simple linear classification model for binary classification, which is viewed as the base model for this project. It predicts the logit of the response from the predictors, where logit is the natural logarithm of odds (Peng, Lee and Ingersoll, 2002). After grid search, the best estimators are as follows. ElasticNet is applied as a combination of L1 and L2 regularization with 0.7 of L1 ratio and solver of ‘saga’. C = 0.3 is set to control the regularization strength.

(2) Random forest

Random forest is an ensemble method for decision trees, reducing the overfitting effects in the decision trees (Fratello and Tagliaferri, 2018). It constructs multiple

decision trees on the training set and outputs the class selected by most trees (Ali, Khan, Ahmad and Maqsood, 2012). To get the best estimators, three parameters are tuned for the random forest:

- the minimum number of samples for a leaf node
- the maximum depth of the tree
- the minimum number of samples for splitting the internal nodes

With the grid search, the recommended estimators are 1 for the minimum samples for a leaf node, 1 for the max depth, and 2 for the minimum samples for splitting the internal nodes.

(3) Naive Bayes

Naïve Bayes is used to building the baseline of modeling and evaluation. It is one of the supervised learning algorithms which utilizes Bayes theorem and assumes strong independence among features. Here we use Complement Naïve Bayes since it is simple enough and is suitable for imbalanced data. It is also suitable for classifying based on discrete features (Sobran, 2013). Laplace smoothing is introduced as one of the parameters to address the 0-probability problem. In addition, since this is a binary classification problem, a uniform prior is used to train the model. Training of Naïve Bayes is speedy compared to other models. As mentioned previously, the result from this model is only used as a metrics baseline, so it is acceptable.

(4) SVM

The support vector machine (SVM) is a supervised machine learning model that uses classification methods to solve issues involving two groups of data or multi-class data. After providing an SVM model with labeled training data sets for each category, they can classify new data points. In other words, a support vector machine will produce the hyperplane that will optimally divide the labels (Noble, 2006).

Basically, the primary goal model is to identify the best hyperplane in N-dimensional space (N here is the number of features) that optimizes the margins from all classes and distinguishes between the different classes of data points. In order to achieve the best performance of the SVM model, we utilized the

grid search to find the best parameters combination with the scoring standard balanced accuracy. After that, we find the best parameters combination is kernel='rbf', C=100 and gamma=1.

(5) CNN

For the convolutional neural network model (Krizhevsky, Sutskever, and Hinton, 2012), since the input data is 2D, we reshape input data from 2D to 3D and use the Conv1D method to construct convolutional layers. Our model consists of three convolutional layers: The first Convolution layer has 32 filters, kernel_size of 3, and ReLU as an activation function. The second layer has 64 filters, and the third layer has 128 layers. A MaxPooling method is used to perform max pooling operation to downsample the input representation, where it takes the maximum value of each spatial window after each convolutional layer, respectively. After that, a flattening operation is used as a fully connected layer to flatten the input. Lastly, since this is a binary classification question, a sigmoid activation function generates the predicted label. An adam optimizer with a learning rate of 0.001 is used for backpropagation of the model, and a binary cross-entropy loss function is used to calculate the loss.

(6) Gradient boosting is a machine learning approach that provides a prediction model in the form of an ensemble of prediction models (Friedman, 2001), often decision trees, for regression, classification, and other problems. In order to receive a better model, the parameter adjustment is made to find a suitable number of estimators and learning rates. Finally, the recommended number of estimators is 100, and the learning rate is 0.1, which will lead to better balanced accuracy.

(7) XGBoosting (Chen, 2016) is a high-speed and high-performance implementation of gradient boosted decision trees. This method supports regression and classification predictive modeling problems. In our paper, first, XGBoosting determines each component's importance and ranks them according to importance. Second, most of the time, XGBoosting outperforms Logistic Regression and Random Forest. Third, XGBoosting also makes it possible to process data in parallel, resulting in a shorter total runtime time. In

the project, XGBoosting is used for our classification problem. After the first XGBoosting matrices are set, the cross-validation parameters are determined utilizing the GridSerach method with the 'scoring' parameter to 'balanced_accuracy.' We find the best parameters are colsamplee_bytree sets to 0.5; learning rate sets to 0.1, max_depth sets to 3, min_child_weight sets to 1, and subsample sets to 1. After the confusion matrix has been performed, the trained XGBoosting model can also plot the importance of the features. The 'bc_number_inpatient' feature is considered to be the most critical driver in the prediction.

## 2.2 Data Collection

The dataset contains clinical care data from 130 hospitals in the United States during 10 years (1999-2008) (Strack et al., 2014). The database contains more than 50 attributes that represent the situation of patients and hospital outcomes (Strack et al., 2014). Information was collected for the database that met the five criteria listed below.

(1) It is a hospitalization encounter (a hospital admission).

(2)In this case, the patient is classified as a diabetic encounter, meaning that it occurred during which any kind of diabetes was recorded into the system as a diagnosis.

(3) The duration of the visit was at least one day and no more than fourteen days.

(4) During the encounter, a series of laboratory tests were carried out.

(5) Medications were delivered during the time of encounter.

Based on these five rules, the ‘Diabetes 130-US Hospitals for Years 1999-2008 Data Set’ contained 101766 records with 50 attributes and was collected from more than 70000 patients with diabetes from 1999 to 2008 within the United States. Each record contains a readmission label indicating whether the patient had no readmission, had readmission for less than 30 days, or had readmission for more than 30 days. There are two data types, categorical and numerical, including demographics like race, age, gender, admission type, diagnosis, various kinds of medications, additional risk factors, etc.

## 2.3 Data Analysis

### 1. 2.3.1 Pre-processing and initial analysis

Initially, the dataset has 50 columns and 101766 observations. With respect to the goal of our project, we affirm that the column 'readmitted' is the label column or classes. To begin with, some missing values in the dataset must be dealt with pre-processing. The missing rate for each column with missing data is shown in Table 1. The variables' weight', 'medical_specialty,' and 'payer_code' have a significant proportion of missing values. Because the variable 'weight' has nearly 97 percent of missing value, it would not be considered for further analysis and modeling. Besides, since almost half of the values are missing on the 'medical_specialty' variable, the column could also be removed. The column' payer code' is the information about bill payer, which is irrelevant information for hospital readmission of patients so that it could be dropped. The mode variable 'Caucasian' fills in for the missing values in 'race.' Additionally, the value 'None' replaces the missing values in the variables' diag_1', 'diag_2', and 'diag_3'. Furthermore, the variables' encounter_id' and 'patient_nbr' are irrelevant attributes related to the readmission situation of patients, thus dropping them in the further analysis. After pre-processing, there are 44 attributes and one label column 'readmitted.'

| Column | Missing value rate % | Processing method |
|---|---|---|
| weight | 96.8584 | remove - too much missing values |
| medical_specialty | 49.0822 | remove - too much missing values |
| payer_code | 39.5574 | remove - irrelevant attributes |
| race | 2.2335 | replace with the mode 'Caucasian' |
| diag_3 | 1.3983 | replace with 'unknown' |
| diag_2 | 0.3517 | replace with 'unknown' |
| diag_1 | 0.0206 | replace with 'unknown' |

Table 1. The proportion of the missing value

Furthermore, there are three class labels ‘NO’, ‘>30’, ‘<30’ within the original dataset column ‘readmitted’ (see Figure 1). Because 30 days is the globally accepted standard time for counting a patient’s return as readmission, many hospitals failing to meet the standard time are penalized (Nakamura et al., 2014). In other words, when a patient returns to the hospital in less than 30 days, it could be considered poor healthcare and a waste of time and resources in the hospital. Thus, it is necessary to combine ‘NO’ and ‘>30’ and relabel them as ‘1’, which represents good healthcare, and only ‘<30’ as ‘0’ represents poor healthcare.

The dataset will be segmented into training (80% of the dataset) and testing (20% of the dataset), and all exploratory data analysis would be based on the train set. The train set has 81412 observations and the test set has 20354 observations.

## 2. 2.3.2 Exploratory data analysis

The primary goal of exploratory data analysis is to aid in the examination of data prior to forming any assumptions. It may assist in identifying apparent mistakes, as well as better understanding patterns within the data,and discovering intriguing relationships between variables. Thus, the exploratory data analysis of our project proceeds with two data types.

In Figure1, there are 43892 observations classified as ‘NO’, 28436 observations as ‘>30’ and s 9085 observations’ <30’ in the train set . Figure 2 shows the distribution of two classes of readmission with reclassification. Thus the counts of the binary classes are 72327 records with class ‘1’ and 9085 records with class ‘0’. After reclassification, the response class of the dataset is extremely imbalanced.

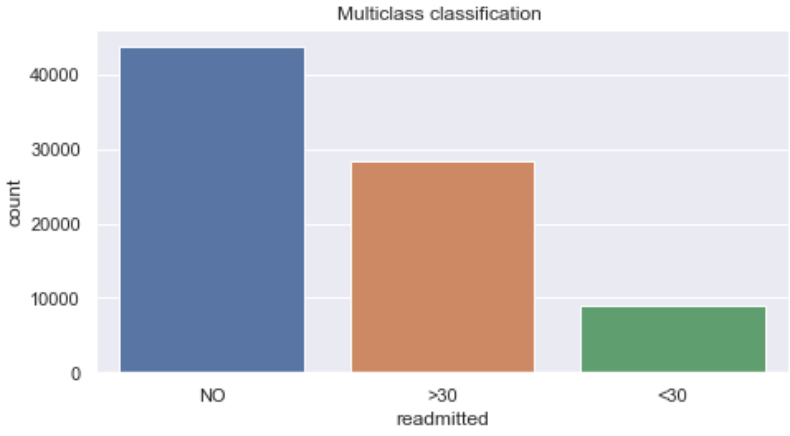


Figure 1. the distribution of original classes

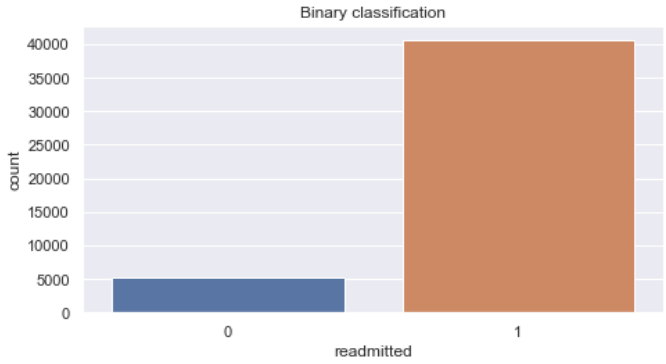



Figure 2. the distribution of classes after reclassification

In terms of categorical features, we utilized the count plot to visualize all distributions of categorical data, compare with nest variables in each feature, and then suggest possible further processing. For example, the count plot of 'examide' only contains one nested variable 'No'(Figure 3). It is meaningless for the classification problem, and we could remove the feature in the further processing.

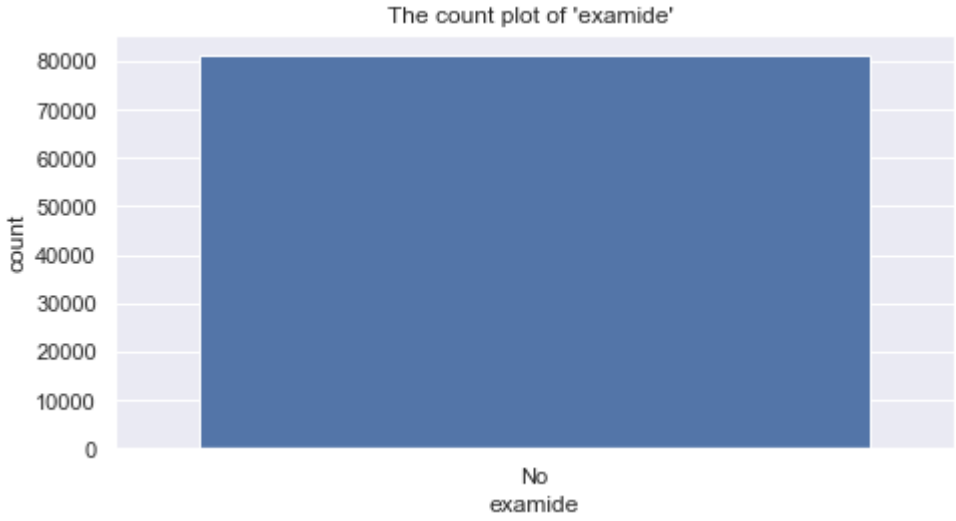



figure 3. the count plot of the feature 'examide'

Furthermore, according to the visualization we find there are some sparse features in the dataset. We plan to deal with this problem with two methods: one is to drop the low variance features after one-hot encoding of categorical features, the other is to regroup the nested variables with respect to some reference and materials. For example, the attribute 'race' count plot suggests removing some low variance after utilizing one-hot encoding. On the other hand, there are many nest variables in the feature 'discharge_disposition_id,' and we could regroup them as new categories with some trustworthy reference.

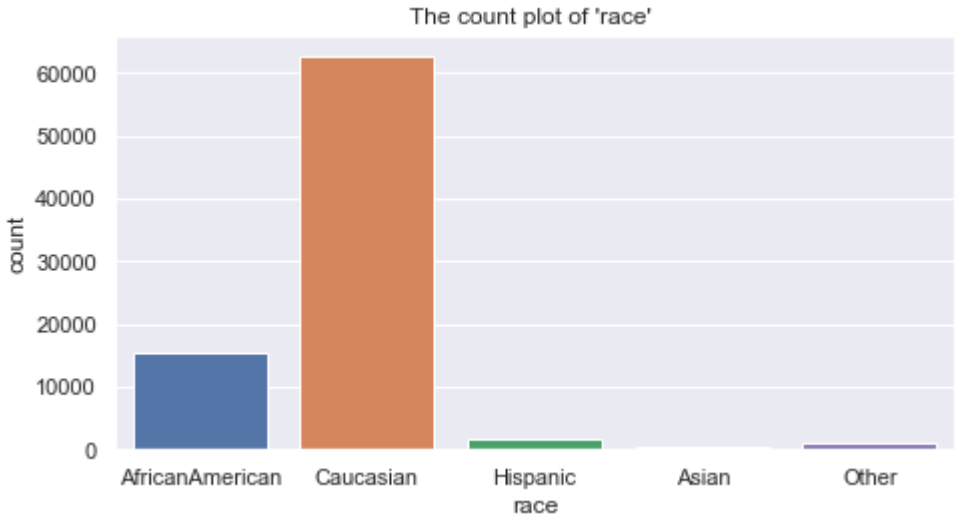



figure 4. the count plot of the feature ‘race’

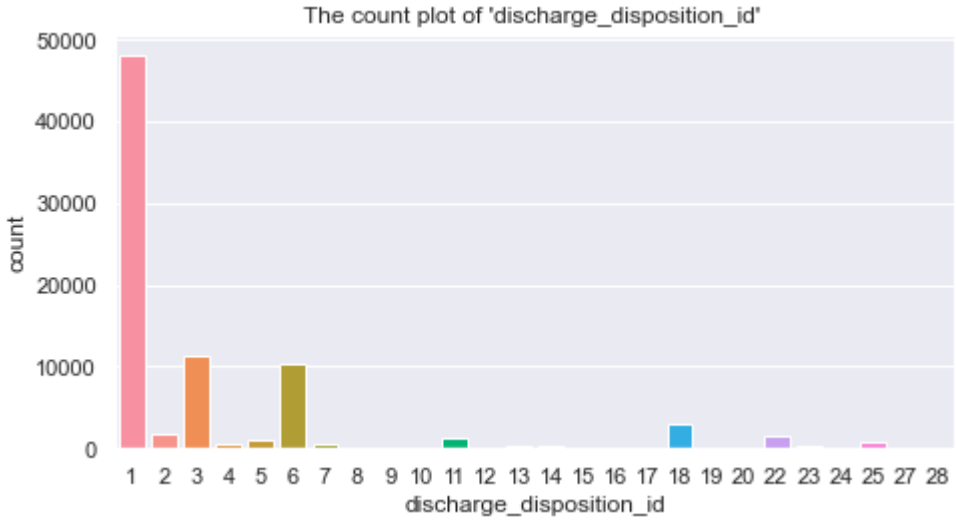



figure 5. the count plot of the feature ‘discharge_disposition_id’

In terms of numerical data, we utilized the displot function to visualize them and analyze their distribution. According to these visualizations, we find there is right skewness on numerical columns. For example, the distribution of ‘time_in_hospital’ is right-skewed. Although this is a classification problem and the impact of right skewness is hard to identify, we still remove such skewness to ease further processing.

Meanwhile, we make use of a heatmap to represent the correlation matrix of these numerical variables. In figure 6, there is no strong relationship since their correlation is relatively small among each other.

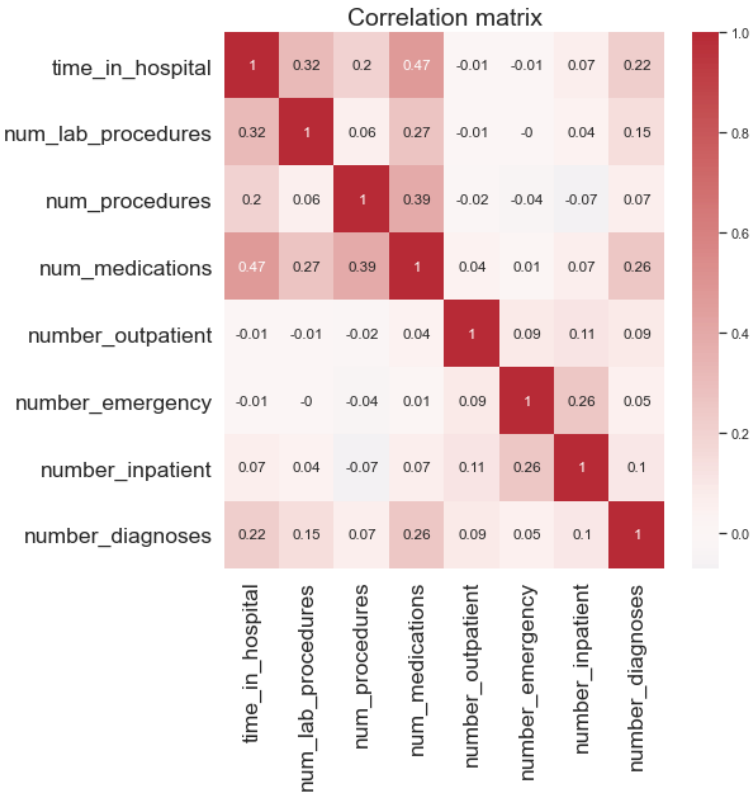



Figure 6. the heatmap of correlation matrix among numerical features

### 3. 2.3.3 Feature Engineering

The “diag_1, diag_2 and diag_3” columns store the ID of primary, secondary, and additional secondary diagnoses of patients. The IDs are coded as the first three digits of ICD9. We grouped each code into its corresponding diagnosis group and resulted in nine groups: Circulatory, Respiratory, Digestive, Diabetes, Injury, Musculoskeletal, Genitourinary, Neoplasms and Other (Strack et al., 2014).

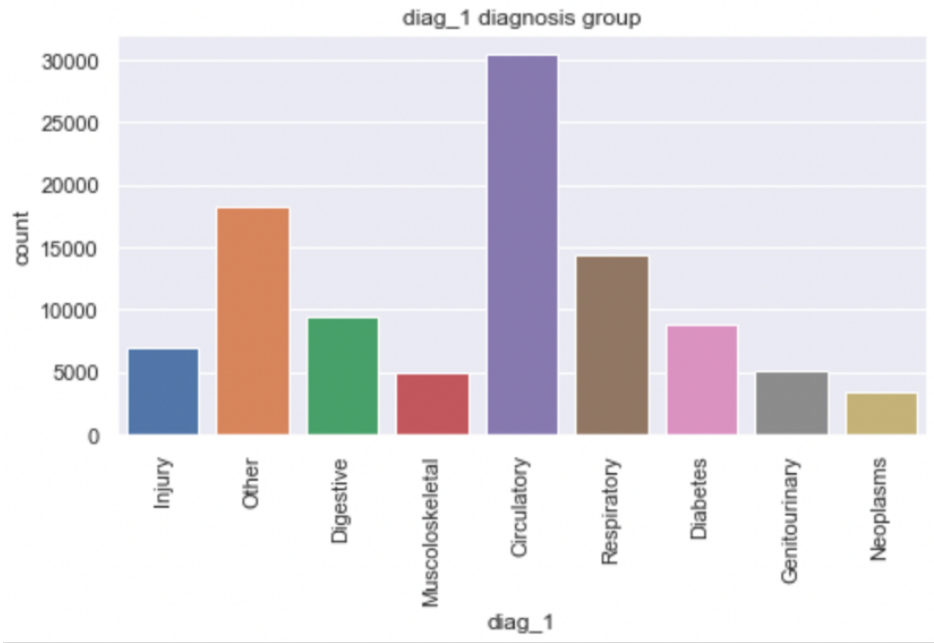



Figure 3. the distribution of primary diagnosis group

For the admission_type_id feature, each record represents the ID of the admission type. We replaced the ID with its actual type: Emergency, Urgent, Elective, Newborn, Not Available, Null, Trauma Center, and Not Mapped (Strack et al., 2014).

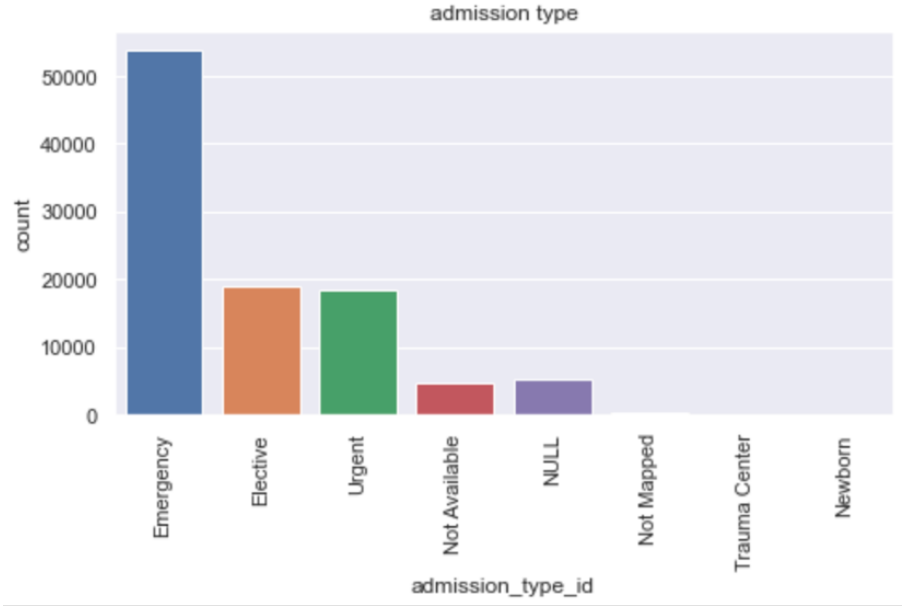



Figure 4. the distribution of admission type

For admission_source_id features, each record represents the ID of the admission source type. We replaced the ID with its actual source type: Referral, Transfer, Emergency room, Court/Law enforcement, Unknown, and delivery (HCUP, 2008).

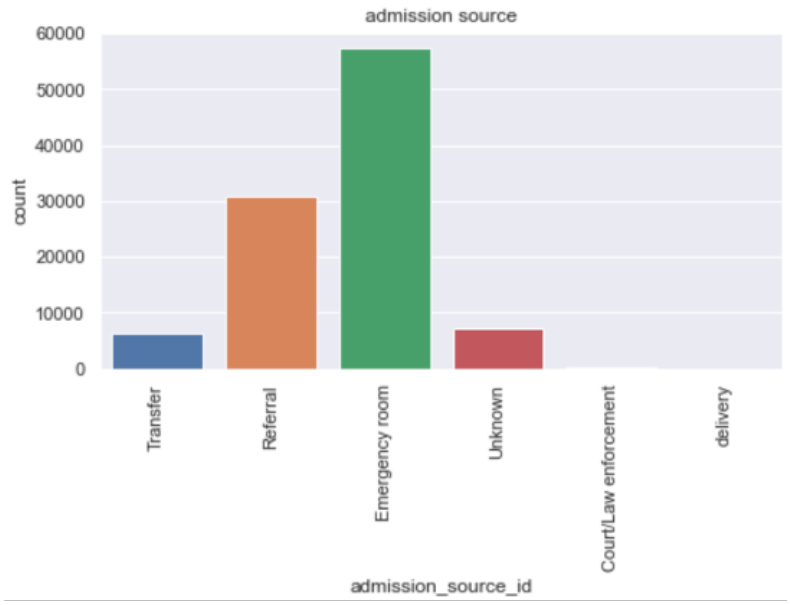



Figure 5. the distribution of admission source

According to The Healthcare Cost and Utilization Project, each ID in discharge_disposition_id belongs to one type of disposition of the patient. Like what we did for admission_type_id and admission_source_id columns, we replaced ID in discharge_disposition_id with the actual type: Routine, Transfer_Hospital, Transfer_Other, HHC, AMA, Expired, NULL, and Not mapped (HCUP, 2008).

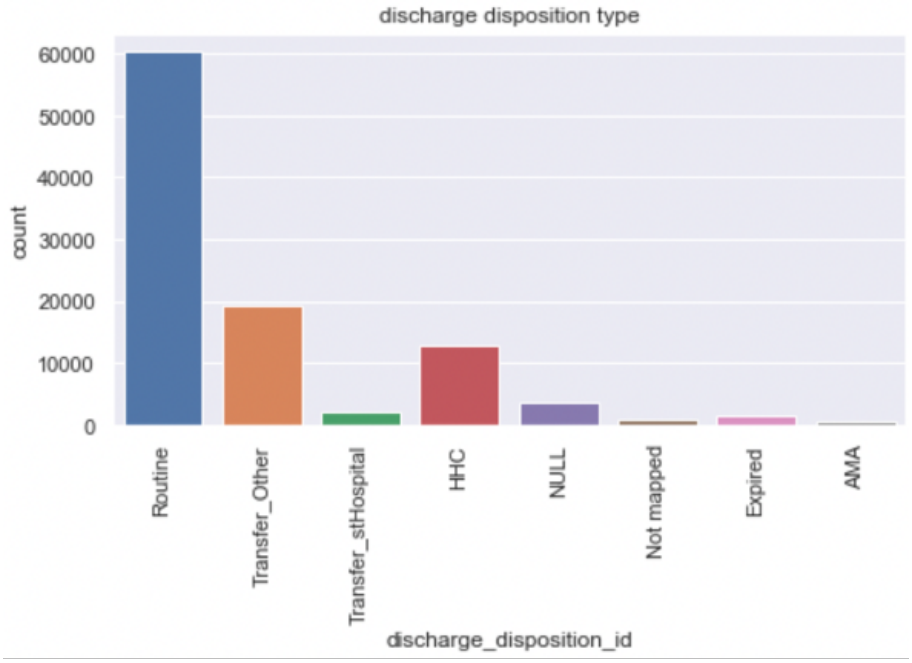



Figure 6. the distribution of discharge disposition type

There are 24 columns representing if there was any diabetic medication given to each patient for 24 types of medications. We recategorized the value of these columns since ‘No’ represents the medication was not given to the patient, we replaced ‘No’

with 0. ‘Up’ means the dosage was increased, ‘Down’ means the dosage was decreased and ‘Steady’ means the dosage did not change. We reclassified ‘Up’, ‘Down’, and ‘Steady’ with 1. Then we created one new variable, ‘number_24medications,’ representing the number of doses the patient takes, which sums all 24 columns’ values. Since changes in medication are also essential to classify readmission labels, we also replaced ‘No’ and ‘Steady’ as 0, ‘Up’ and ‘Down’ as one and created variable ‘number_change_24medications’, which adds these values together to see the number of changes of doses the patient took.

In the gender feature, three records have the value of ‘Unknown/Invalid,’ which are dropped.

In relation to age, it contains age intervals. Therefore, age is viewed as an ordinal variable and is transformed by ordinal encoding.

For numerical variables, we noticed that most numerical features have right-skewed distribution. We applied Box-Cox transformation so that non-normal variables could be transformed into normal shapes. Interquartile range box plots are used to show and remove outliers of the dataset. Values within quantile 0.25 to 0.75 are kept. After removing these outliers and 23 features related to 24 medications, there are only 45902 observations and 21 attributes on the dataset.

For categorical data features, the One-hot encoding technique is used. We use dummy variables to represent categorical data where dummy variables transfer original categorical data into a set of zero and indicate the absence or presence of categorical values. After one-hot encoding, our data features increase from 21 to 77.

The class imbalance problem has become one of the most major obstacles in terms of Machine Learning datasets nowadays. With an imbalanced dataset, the model makes predictions predominantly based on the majority class of the dataset but ignores the minority class, resulting in bad performance of predicting labels of the minority class. In this project, the Synthetic Minority Over-sampling Technique (SMOTE) is introduced to the training dataset since SMOTE creates synthetic minority class examples, increasing input data size and reducing the over-fitting problem of models (Chawla, Bowyer, Hall and Kegelmeyer, 2002). We only applied the SMOTE method on the train set, and it generated some artificial data points on the train set. Eventually, the train set has 81408 data points with respect to 40704 of class ‘1’ and 40704 of class ‘0’.

A Min-Max normalization is applied to the dataset to scale each feature to a range of zero and one.

Lastly, we analyse the mutual information between input variables and target labels. Mutual information measures the degree of mutual dependence between two variables, and analyzing the mutual information can tell us how much information of target labels can be obtained from a random input variable. 12 features have values of mutual information greater than 0.015, and the rest of the features contribute very little to obtaining target labels. Therefore, only those 12 features are kept as input datasets since the fewer features we use, the faster each model's training process can achieve.

# 3. RESOURCES

Hardware & Software

**Software:**

Python: Python 3 is the language we used to construct the project, including processing, EDA, model and evaluation.

Pandas: Pandas library is used to perform data loading and processing techniques to raw data such as column deletion, transformation and creation.

Seaborn and Matplotlib: Seaborn library and matplotlib library are used to plot graphs for initial data analysis and EDA including heatmaps count plots and boxplots of different features of data.

Numpy: Numpy library is applied to perform feature engineering to dataset such as changing data type of some columns and making some data value to null value.

Sklearn: We use the sklearn library to achieve many operations: train_test_split is used to split the original dataset into a training dataset and a testing dataset with a ratio of 8:2. Feature_selection is used to help us find the most useful features among all features to predict the result. PCA is used for feature reduction to reduce the features of the dataset and improve the speed of the model significantly. Logistic Regression, Random Forest, Naive Bayes, SVM and Gradient Boosting helps us to build our own logistic regression model, random forest model, naïve bayes model, support vector machine model and gradient boosting model. precision_score, recall_score, f1_score,

accuracy_score, roc_auc_score and balanced_accuracy_score are used for model parameter tuning and model evaluation, so that the performance of each model is measured and displayed.

Imblearn: We applied the imblearn library to perform SMOTE oversampling technique for our dataset to deal with the imbalanced data problem.

Tensorflow: Tensorflow library is mainly used to build the CNN model such as construction of convonlution layers, dense layers, compilation and fitting the model to our dataset.

Xgboost: Xgboosting model is achieved by applying Xgboost library.

### 3.1 Materials

WeChat and slack are communication tools we use for daily communication within the team and tutor.

Zoom is the platform for meeting within the team, with tutor and client.

All our documents and reports are stored and constructed in Google Drive and Google doc.

Github is the coding platform we use for upload and download codes.

We download the dataset through UCI Machine Learning Repository and Kaggle. We also viewed some previous work of the dataset from Kaggle.

Online libraries, including Google Scholar, the University of Sydney library, ACM Digital Library, and IEEE Xplore, are the libraries we used for searching past related papers and literature.

## 4. MILESTONES / SCHEDULE

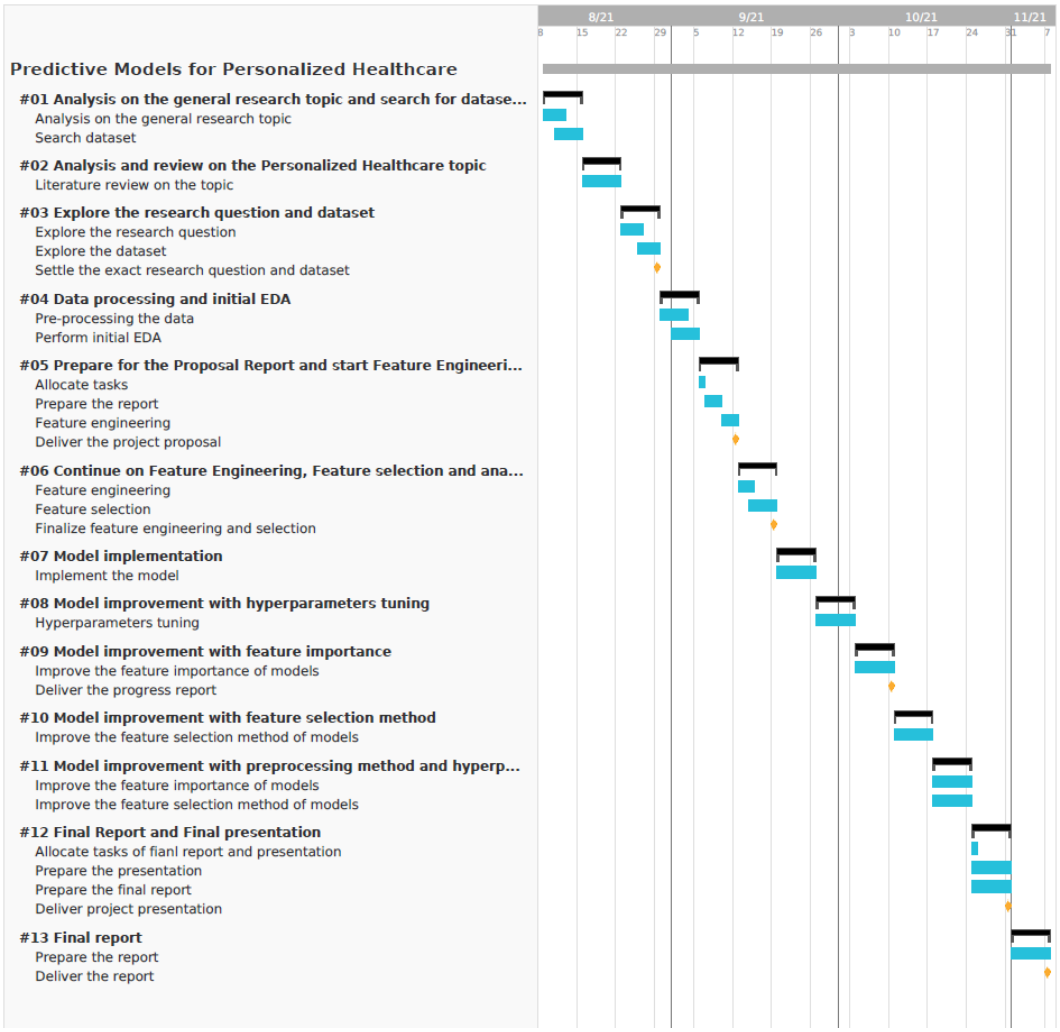
| # Week | Activities | Definition |
|---|---|---|
| Week-1 | Analysis on the general research topic and search for datasets | Investigate different topics and choose our project topic. After determining the topic, we search for related dataset. |
| Week-2 | Analysis and review on the Personalized Healthcare topic | Investigate the details about personalized healthcare. |

| | | |
|---|---|---|
| Week-3 | Explore the research question and dataset | Identify our research question and find a suitable dataset. |
| Week-4 | Data processing and initial EDA | Pre-process the data to be analyzable and perform exploratory data analysis on attributes. |
| Week-5 | Prepare for the Proposal Report and start Feature Engineering | Prepare the project proposal and start to encode data. |
| Week-6 | Continue on Feature Engineering, Feature selection and analysis | Finalize feature engineering. Select a subset of features based on previous findings and investigate the relationship within such features. |
| Week-7 | Model implementation | Implement the model we proposed. |
| Week-8 | Model improvement with hyperparameters tuning | Continue implementing the model we proposed, focus more on hyperparameters tuning on models. |
| Week-9 | Model improvement with feature importance | Continue implementing the model we proposed, focus more on feature importance. |
| Week-10 | Model improvement with feature selection method | Continue implementing the model we proposed, focus more on feature selection. |
| Week-11 | Model improvement with preprocessing method and hyperparameter tuning again | Continue implementing the model we proposed, focusing more on preprocessing methods. |
| Week-12 | Final Report and Final presentation | Prepare the final report and presentation. |
| Week-13 | Final Report (thesis) | Continue preparing the final report. |

2.

| # Week | Activities | Milestone | Date |
|---|---|---|---|
| Week-1 | Analysis on the general research topic and search for datasets | Confirm the project topic and dataset | 15/08/2021 |
| Week-2 | Analysis and review on the Personalized Healthcare topic | None | 22/08/2021 |
| Week-3 | Explore the research question and dataset | Settle the exact research question and dataset | 29/08/2021 |
| Week-4 | Data processing and initial EDA | None | 05/09/2021 |
| Week-5 | Prepare for the Proposal Report and start Feature Engineering | Deliver the project proposal | 12/09/2021 |
| Week-6 | Continue on Feature Engineering, Feature selection and analysis | Finalize feature engineering | 19/09/2021 |
| Week-7 | Model implementation | None | 26/09/2021 |
| Week-8 | Model improvement with hyperparameters tuning | None | 10/10/2021 |

| | | | |
|---|---|---|---|
| Week-9 | Model improvement with feature importance | None | 17/10/2021 |
| Week-10 | Model improvement with feature selection method | None | 24/10/2021 |
| Week-11 | Model improvement with preprocessing method and hyperparameter tuning again | None | 31/10/2021 |
| Week-12 | Final Report and Final presentation | Deliver project presentation | 07/11/2021 |
| Week-13 | Final Report (thesis) | Deliver the report | 14/11/2021 |




3.

## 3. RESULTS

In this project, we select 3 metrics for evaluation: balanced accuracy, F1, and AUC. All three metrics need to be calculated by the counts from the confusion matrix (Table xx) (Sokolova and Lapalme, 2009).

| Data class | Classified as positive | Classified as negative |
|---|---|---|
| Positive | True positive (TP) | False negative (FN) |
| Negative | False positive (FP) | True negative (TN) |
| Sum | Positive (P) | Negative (N) |

Table xx: Confusion matrix for binary classification (Sokolova and Lapalme, 2009)

Balanced accuracy is the average of true positive rate and true negative rate, which is a good metric for imbalanced data (Bekkar, Djemaa and Alitouche, 2013). Therefore, it is the primary criterion for evaluating our models.

| | | *Equation 1* |
|---|---|---|

Precision (P) measures the number of true positive cases over the total number of positive labels (Hossin and Sulaiman, 2015).

*Equation 2*

Recall (R) represents the proportion of the positive correctly labelled category over the total correctly classified categories (Hossin and Sulaiman, 2015).

*Equation 3*

F1 gives the harmonic mean between recall and precision (Hossin and Sulaiman, 2015). Therefore, we choose F1 instead of precision and recall.

*Equation 4*

AUC (Area Under the Curve) calculates the area under ROC (Receiver Operating Characteristics) curve, which identifies the model's ability to distinguish two labels (Narkhede, 2018). A basic ROC curve contains two axes: a true positive rate for the y-

axis and a false positive rate for the x-axis (Fawcett, 2006). Additionally, since it is used in many papers, we can use our AUC results to compare with others.

For related works, most of them use AUC as the evaluation metrics. The model with the highest AUC is conducted by Hempstalk and Mordaunt's (2016) logistic regression model. Under the 10-fold cross-validation, the average AUC is 0.67 with a 0.008 standard deviation (Hempstalk and Mordaunt, 2016). In this project, the best AUC we found is 0.6039 for random forest (Table xx), which is just a little lower than Hempstalk and Mordaunt's (2016) model. However, AUC and F1 may be misleading since they ignore the impact of low accuracy of the minority class.

Moreover, the data imbalance in the testing set would significantly affect AUC and F1 (Elazmeh, Japkowicz, and Matwin, 2006). Some of the other papers also use the standard accuracy score as their first consideration. It seems that they get high accuracy, but it is not reliable. For example, Jia and Tian (2019) get 0.9 accuracy on CNN, but they apply SMOTE before train test split so that there are a large number of fake observations in their test set.

Overall, gradient boosting has the best-balanced accuracy (Figure 1). However, although SVM becomes the best based on the F1 score, gradient boosting's F1 score is not bad (Figure 1). Besides, gradient boosting ranks second based on the AUC, 0.6022 (Table xx) (Figure 2), similar to the random forest's AUC results. Since balanced accuracy is our first choice of evaluation, gradient boosting is considered as the best model for our project.

| | **balanced accuracy** | **AUC** | **F1** |
|---|---|---|---|
| **Logistic Regression** | 0.5489 | 0.5764 | 0.7300 |
| **Random Forest** | 0.5828 | 0.6039 | 0.8107 |
| **Naive Bayes** | 0.5167 | 0.5635 | 0.6229 |
| **Gradient Boosting** | 0.5912 | 0.6022 | 0.7815 |

| | | | |
|---|---|---|---|
| **CNN** | 0.5482 | 0.5482 | 0.8908 |
| **XGB** | 0.5726 | 0.5726 | 0.7646 |
| **SVM** | 0.5245 | 0.5245 | 0.9140 |

table xx: detailed evaluation results table

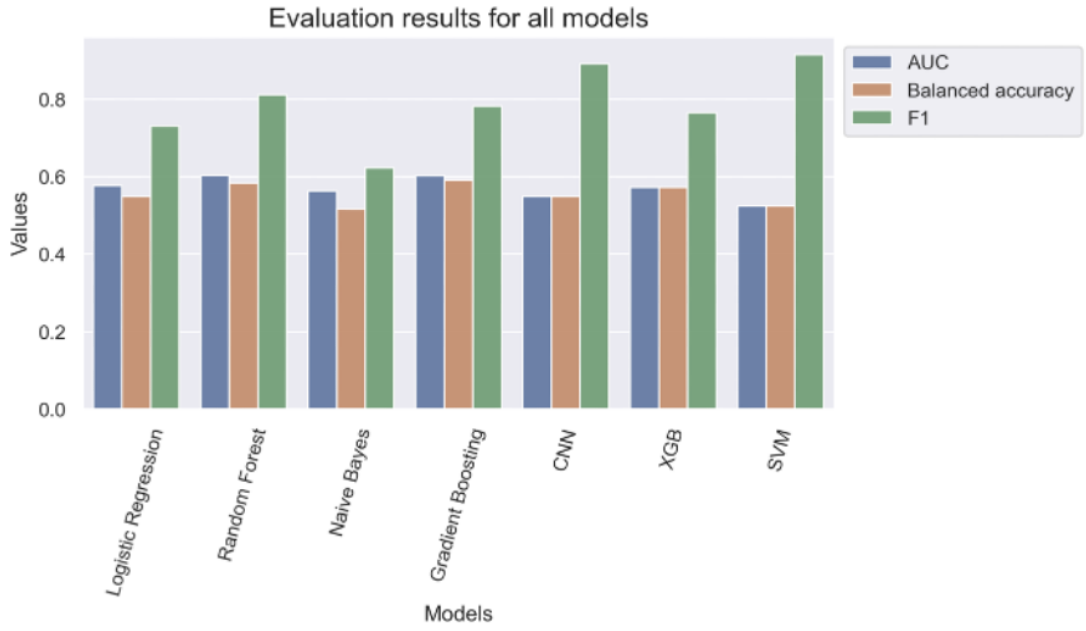



*Figure 1: Evaluation results*

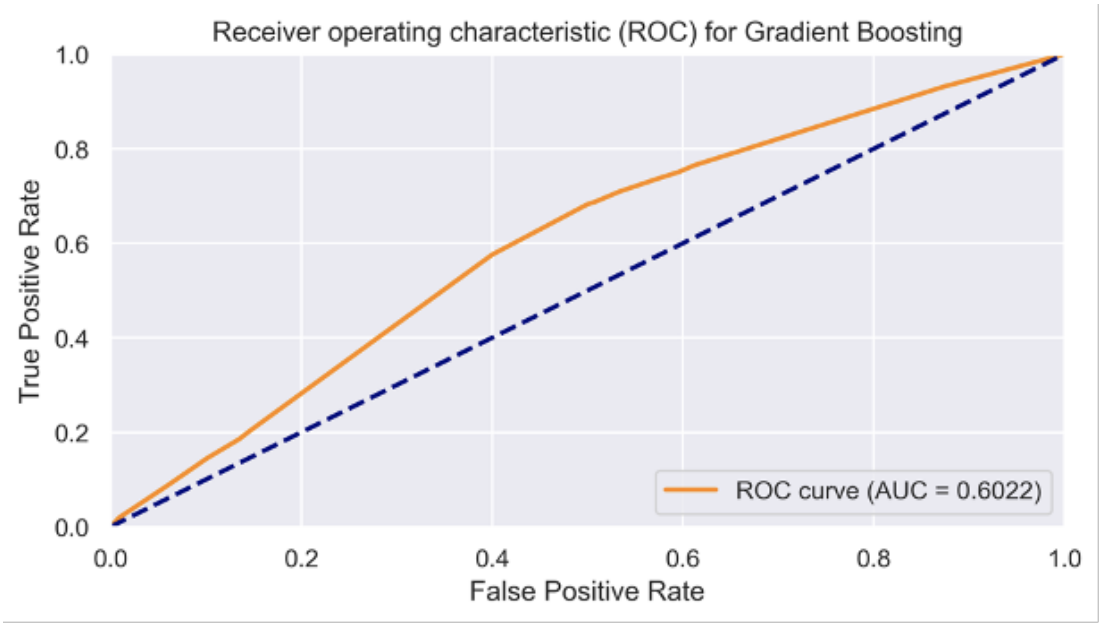


*Figure 2: Evaluation results*

# 4. DISCUSSION

Based on the evaluation, the best model selected is gradient boosting with the highest balanced accuracy of 0.5912. For implication, we expected that this model could assist doctors in diabetes to make more precise decisions on whether a certain patient requires readmission within 30 days. That is, before a doctor makes the final decision on the patients' readmission based on their knowledge, they can review their decision based on the prediction results by our model. Therefore, our model can reduce the time consumption for decision-making and the extra cost that occurred due to unplanned short-term readmission.

# 5. LIMITATIONS AND FUTURE WORKS

In conclusion, we built seven different machine learning models to predict diabetes readmission. Since we choose balanced accuracy and AUC as our evaluation method, the gradient boosting method has the best performance. Compared with other papers that use the same dataset, our gradient boosting model can achieve higher balanced accuracy of 59.12%. The pre-processing procedure we used is the reason we outperformed their findings. Our project uses the SMOTE oversampling method to handle the imbalance problem of the dataset. Then, the mutual information method is used to select important features to reduce useless features. It also can improve the running speed of the model.

Nevertheless, there are still parts of our model that can be improved. Firstly, the dataset can be updated. Since the dataset is from 10 years ago and people's eating habits have changed very quickly in recent years, the model may not be time-sensitive. If we want to create a model that can be directly applied to real life, we need the latest data. Secondly, the parameters of the model can probably be optimized because the current ones are only the ones we have adjusted in a limited time. Thirdly, this model can only be used on this dataset. It is not generalizable and can not be immediately extended to the overall health care system. It still depends on specific data and features.

# REFERENCES

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random forests and decision trees. International Journal of Computer Science Issues (IJCSI), 9(5), 272.

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl, 3(10).

Commission, M. P. (2007). Report to the Congress: promoting greater efficiency in Medicare.

Elazmeh, W., Japkowicz, N., & Matwin, S. (2006, September). Evaluating misclassifications in imbalanced data. In European Conference on Machine Learning (pp. 126-137). Springer, Berlin, Heidelberg.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

Fratello, M., & Tagliaferri, R. (2018). Decision trees and random forests. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, 1, 3.

Hempstalk, K., & Mordaunt, D. (2016). Improving 30-day readmission risk predictions using machine learning. In Health Informatics New Zealand (HiNZ) Conference (Vol. 2016).

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2), 1.

Jia, M., & Tian, F. (2019, December). Readmission prediction of diabetic based on convolutional neural networks. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)* (pp. 1990-1994). IEEE.

Mingle, D. (2017). Predicting diabetic readmission rates: moving beyond Hba1c. Current Trends in Biomedical Engineering & Biosciences, 7(3).

Nakamura, M. M., Toomey, S. L., Zaslavsky, A. M., Berry, J. G., Lorch, S. A., Jha, A. K., Bryant, M. C., Geanacopoulos, A. T., Loren, S. S., Pain, D., & Schuster, M. A. (2014). Measuring Pediatric Hospital Readmission Rates to Drive Quality Improvement. Academic Pediatrics, 14(5), S39–S46. https://doi.org/10.1016/j.acap.2014.06.012

Narkhede, S. (2018). Understanding auc-roc curve. Towards Data Science, 26, 220-227.

Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. The journal of educational research, 96(1), 3-14.

Shinkman, R. (2014, April 20). Readmissions lead to $41.3B in additional hospital costs. Retrieved from FIERCE Healthcare: https://www.fiercehealthcare.com/finance/readmissions-lead-to-41-3b-additional-hospital-costs

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information processing & management, 45(4), 427-437.

Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c Measurement on Hospital Readmission Rates:. BioMed Research International, 781670-11. doi:https://doi.org/10.1155/2014/781670

Wardhani, N. W. S., Rochayani, M. Y., Iriany, A., Sulistyono, A. D., & Lestantyo, P. (2019, October). Cross-validation metrics for evaluating classification performance on imbalanced data. In 2019 international conference on computer, control, informatics and its applications (ic3ina) (pp. 14-18). IEEE.

Waters, H., & Graf, M. (2018). The Costs of Chronic Disease in the U.S. Santa Monica: Milken Institute.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

HCUP. (2008). DISPUNIFORM - Disposition of patient, uniform coding. Retrieved from: https://www.hcup-us.ahrq.gov/db/vars/siddistnote.jsp?var=dispuniform

HCUP. (2008). ASOURCE - Admission source, uniform coding. Retrieved from: www.hcup-us.ahrq.gov/db/vars/siddistnote.jsp?var=asource.

Strack, B., DeShazo, J., Gennings, C., Olmo, J., Ventura, S., Cios, K., & Clore, J. (2014, April 3). *Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records.* Hindawi. https://www.hindawi.com/journals/bmri/2014/781670/

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

Sobran, N.M., Ahmad, A., & Ibrahim, Z. (2013). CLASSIFICATION OF IMBALANCED DATASET USING CONVENTIONAL NAÏVE BAYES CLASSIFIER.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology*, *24*(12), 1565-1567.

Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29, 1189-1232.

Ahmad Hammoudeh, Ghazi Al-Naymat, Ibrahim Ghannam, Nadim Obied. (2018). *Predicting Hospital Readmission among Diabetics using Deep Learning.* Procedia Computer Science 141 (2018) (pp. 484–489)

Alloghani, M., Aljaaf, A., Hussain, A., Baker, T., Mustafina, J., Al-Jumeily, D., & Khalaf, M. (2019). Implementation of machine learning algorithms to create diabetic patient re-admission profiles. BMC Medical Informatics and Decision Making, 19(Suppl 9), 253–253. https://doi.org/10.1186/s12911-019-0990-x

Bhuvan M S, Ankit Kumar, Adil Zafar, Vinith Kishore. (2016). Identifying Diabetic Patients with High Risk of Readmission. arXiv: 1602.04257v1 (pp.1-11)

Chopra, C., Sinha, S., Jaroli, S., Shukla, A., & Maheshwari, S. (2017, October). Recurrent neural networks with non-sequential data to predict hospital readmission of diabetic patients. In Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics (pp. 18-23).

Damian M. Predicting Diabetic Readmission Rates: Moving Beyond Hba1c. Curr Trends Biomedical Eng & Biosci. 2017; 7(3):555707. 007 DOI: 10.19080/CTBEB.2017.07.555715

Diviya Prabha, V., & Rathipriya, R. (2020). Readmission Prediction Using Hybrid Logistic Regression. In Innovative Data Communication Technologies and Application (pp. 702–709). Springer International Publishing. https://doi.org/10.1007/978-3-030-38040-3_80

Ghazo, E. (2019). Prediction of Diabetic Patient Readmission Using Hybrid Ensemble Learning. ProQuest Dissertations Publishing.

Hempstalk, K., & Mordaunt, D. (2016). Improving 30-day readmission risk predictions using machine learning. In Health Informatics New Zealand (HiNZ) Conference (Vol. 2016).

M. Jia and F. Tian, "Readmission Prediction of Diabetic based on Convolutional Neural Networks," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019, pp. 1990-1994, doi: 10.1109/ICCC47050.2019.9064477.

Qian Zhu, Akkati, A., & Hongwattanakul, P. (2016). Risk feature assessment of readmission for diabetes. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 538–543. https://doi.org/10.1109/BIBM.2016.7822578

Ramírez, J. C., & Herrera, D. (2019, June). Prediction of diabetic patient readmission using machine learning. In *IEEE Colombian Conference on Applications in Computational Intelligence* (pp. 78-88). Springer, Cham.

Sharma, A., Agrawal, P., Madaan, V., & Goyal, S. (2019). Prediction on diabetes patient's hospital readmission rates. Proceedings of the Third International Conference on Advanced Informatics for Computing Research, 1–5. https://doi.org/10.1145/3339311.3339349

Shibly, M. M. A., Tisha, T. A., & Mazumder, M. M. I. (2021). Predicting Early Readmission of Diabetic Patients: Toward Interpretable Models. In Lecture Notes in Electrical Engineering (Vol. 733, pp. 185–200). https://doi.org/10.1007/978-981-33-4909-4_14

Tamin, F., & Iswari, N. M. S. (2017). Implementation of C4.5 algorithm to determine hospital readmission rate of diabetes patient. 2017 4th International Conference on New Media Studies (CONMEDIA), 15–18. https://doi.org/10.1109/CONMEDIA.2017.8266024

Tarig Mohamed Ahmed. (2016). Using Data Mining To Develop Model For Classifying Diabetic Patient Control Level Based On Historical Medical Records. Journal of Theoretical and Applied Information Technology, Vol.87. No.2