CS6220 Data Mining - Final Project

So Man Amanda Au Yeung
Yian Chen
Chun Wei Tseng

https://github.com/Chun-Wei-Tseng/CS6220-FinalProject

1. Introduction

The rising problem of early readmissions among diabetic patients presents formidable challenges for healthcare, resulting in complications and financial penalties. Our project seeks to forecast the combination of features that contribute to early hospital readmissions within a 30-day window, with a focus on addressing gaps in preventive care and enhancing patient outcomes. In contrast to conventional models, we delve into a range of machine learning approaches to gain deeper insights into the factors influencing early readmissions, creating a synergy between healthcare and data mining.

2. Background

Diabetic patients frequently experience inadequate care, leading to complications and heightened hospital expenditures. The Center for Medicaid & Medicare Services (CMS) penalizes hospitals for elevated readmission rates, underscoring the financial impact and influencing quality ratings. Our project is dedicated to forecasting features that contribute early readmissions (within 30 days) for diabetic patients, addressing gaps in preventive care. In contrast to traditional regression models, we explore diverse models to gain a nuanced understanding of the factors contributing to early readmission. Positioned at the intersection of healthcare and data mining, this project aims to elevate patient outcomes, reduce costs, and optimize diabetes care practices.

3. Approach

3.1 Data Analysis

Our data consists of 101766 instances with 47 features and a target y label which shows the results of early readmission within 30 days of a patient being discharged. In our feature selection process, we dropped features that have a high proportion of missing values. We also dropped medication features that have a low correlation to our label. For our diagnoses features, there is primary diagnosis (diag_1), secondary diagnosis (diag_2), and tertiary diagnosis (diag_3). At first we tried feature hashing to try to include all the diagnoses, however this multiplied our features and did not reflect an improvement in our model predictions later on. Upon further research we found that we could essentially rely on the primary diagnosis and reduce our data noise by removing diagnosis 2 and 3 (Impact of

HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records). In addition,

Analyze data, discuss the imbalance of our class distribution and how we aimed to balance the dataset. Discuss how the imbalance affect our results. determine feature correlation, discuss what features are most useful and how we determined it (how we did dimensionality reduction/feature selection)

3.2 discuss what we're optimizing for

3.3 Results

3.1. Data & Data Analysis

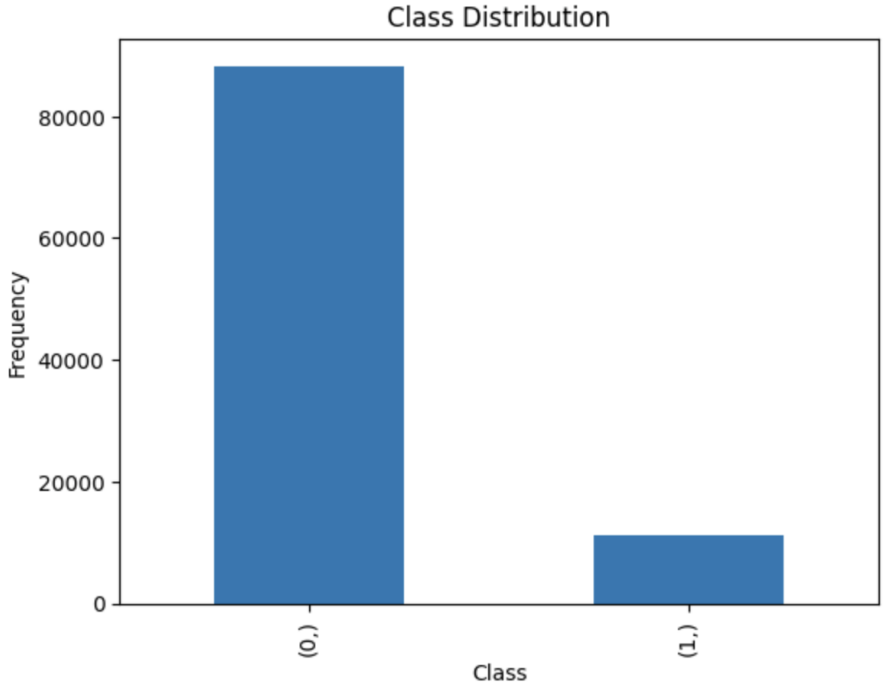
Our dataset encompasses 101,766 patient records, each described by 47 attributes. The target variable 'y' denotes whether a patient was readmitted within 30 days post-discharge. During feature selection, attributes with substantial missing data were omitted to ensure the integrity of the analysis. Medication-related features exhibiting negligible correlation with the target variable were also excluded to sharpen the predictive power of the model. The dataset initially included primary (diag_1), secondary (diag_2), and tertiary (diag_3) diagnosis categories. An initial attempt to incorporate these through feature hashing was made, but this technique expanded the feature space without yielding predictive performance gains. Subsequent to a detailed review, it became apparent that focusing exclusively on the primary diagnosis could enhance model accuracy while reducing complexity. This approach aligns with findings from an extensive clinical database analysis, which underscores the significance of the primary diagnosis in readmission rates(Strack,DeShazo,Gennings,Olmo,Ventura,Cios,Clore, 2014, Year).

Data Preprocessing

- Dropped columns in the data frame that have amount of missing data larger than the threshold we set (20%)
- Dropped medication columns with those that have less than 100 patient the medication
- Dropped secondary and additional secondary (diag_1, diag_2) diagnosis
- Dropped columns in the data frame that are not diverse in value
- Mapped ordinal categorical columns to integer value
- One-hot encoding non-ordinal categorical value
- Group values in the primary diagnosis column (diag_1) and apply one-hot encoding to them
- Concatenate data frames with categorical and numerical values
- Drop rows with NaN value in it

- Using train_test_split to split the data into two groups, for training, validation, and testing
- Using MinMaxScaler to normalize data before train them with different models

- Distribution of target values (y):

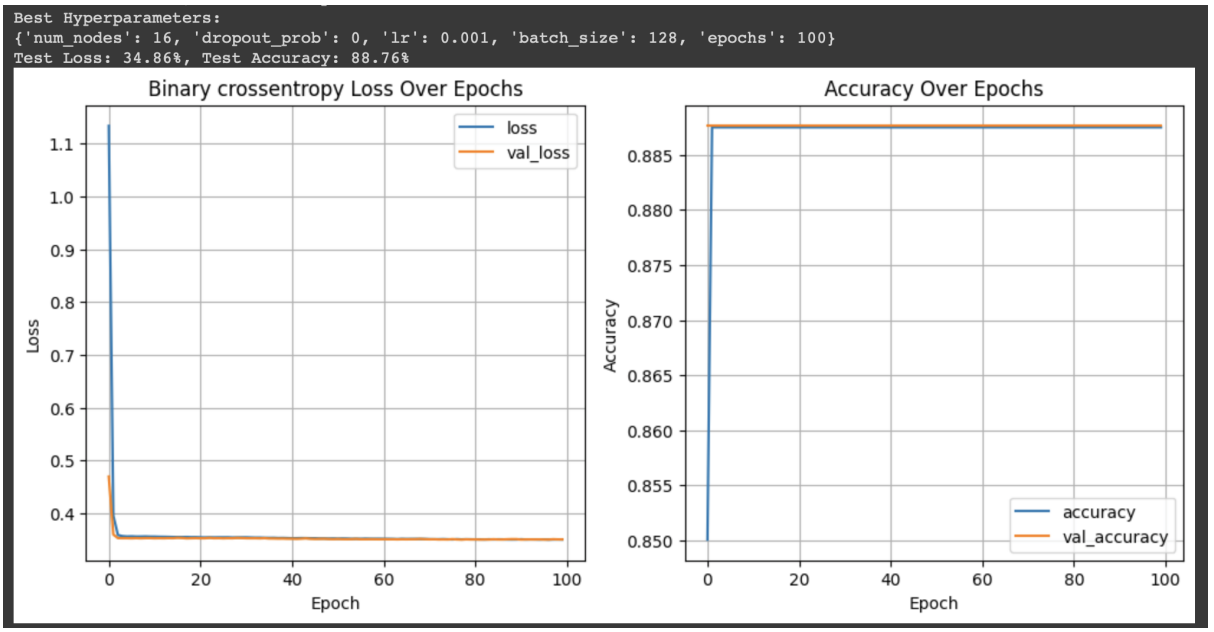


3.2. Implementation

- Original dataset has around 50 columns (features), and many of them have missing values, a wide range of values, or non-numerical values. This can cause a lot of noise when training models. We spent a lot of time deciding which columns or rows to drop, how to map categorical values to numerical values or one-hot encoding.
- Target value of the data is very unbalanced:
  - For our project, we considered patients readmitted within 30 days as 1, and patients readmitted over 30 days (or not readmitted) as 0. Turns out there are over 8 times more 0s than 1s.
- Optimizing the model with grid search method might not work for all the models:
  - When we were trying to optimize the SVM using grid search, the runtime turned out to be too long for computing.
- Optimizing the model with SMOTE and Stratified K-fold cross validation on the neural network for our severely imbalanced data also did not work. The results have shown little to no improvement.
- Within the neural network layer, the results have shown overfitting. Therefore, the dropout layer is incorporated within the keras layer to randomly set the

input unit to 0 with a frequency of rate at each step during training time, which prevents overfitting.

- The incorporated dropout layer with a rate of 0.2 has shown significant improvement, compared to the initial model without. It was also tested with different learning rate, number nodes, batch size, drop out rate to search for the least loss model. However, the run time took too long to finish the different combinations of different parameters (learning rate, number nodes, batch size, drop out rate). Thus, a compromised approach was used with a random choice of each value every time with iterations. After using the compromised approach, the parameters were reduced to make sure all combinations were tested for accuracy.
- Later on, regularization l1 was added instead of the dropout layer towards the neural network. The data was far less overfitting and the slope of the curve diminishes,which has faster convergence reaching maximum learning ability.



3.3.

4. Results & Evaluation

- Naive Bayes:
  - Accuracy: 88.76%
  - Area under curve for Receiver Operating Characteristic: 56.75%

- Support Vector Machine (SVM):
  - Accuracy: 88.76%
  - Area under curve for Receiver Operating Characteristic: 51.87%

- Neural Networks:

    - Accuracy with least loss model: 88.75%
    - Area under curve for Receiver Operating Characteristic: 60%

- Logistic Regression
    - Accuracy: 88.74%
    - Area under curve for Receiver Operating Characteristic: 53.22%

- Random Forest:
    - Accuracy: 89%
    - Area under curve for Receiver Operating Characteristic: 60%
- XGBoost:
    - Accuracy: 88.78%
    - Area under curve for Receiver Operating Characteristic: 62.93%

Comparison between methods:

- Performance:
    - Naive Bayes: low ROC-AUC
    - Support Vector Machine (SVM): low ROC-AUC
    - Neural Networks
    - Logistic Regression
    - Random Forest:
    - XGBoost: highest ROC-AUC

5. Conclusions

- All of the models we used have accuracy of predictions on the testing dataset around 88% or above, which seems to be really good. However, when we take the ROC-AUC (Area under curve for the Receiver Operating Characteristic), most of the models didn't receive a high score. Having ROC-AUC below 70, or even close to 50 or 60, for binary classification models means that they have poor performance. This can be caused by the unbalanced amount of targeting values ("Readmitted to hospital, 0 or 1"). XGBoost has benefits in this case. Compared to other models, it handles dataset with missing or unbalanced values a lot better. During the training process, it combines multiple weak learners to create a stronger learner, which typically has better generalization capability and is less likely to be

overfitted. XGBoost has built in regularization methods to handle missing values and imbalance data.
- When taking a small badge of a balanced dataset for testing, the result changed. Other models that have low ROC-AUC because of unbalanced dataset performed better now. On the other hand, XGBoost decreases in performance.

Citation:

- https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008
- https://www.cdc.gov/diabetes/health-equity/diabetes-by-the-numbers.html
- (Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, Strack,DeShazo,Gennings,Olmo,Ventura,Cios,Clore, 2014).