Competition-2 Report

Yan Junyu
School of Computing
Queen's University
Kingston, Canada
junyu.yan@queensu.ca
20188932

Ajibode Adekunle
School of Computing
Queen's University
Kingston, Canada
Ajibode.a@queensu.ca
20436193

Li Shengan
School of Computing
Queen's University
Kingston, Canada
16sl75@queensu.ca
20048177

I. Data exploration

We initiated this experiment by delving into the characteristics of the dataset. The dataset comprises three distinct data files, "C2T1_Train.csv", "C2T1_Test.csv", and "C2T1_IDs_mapping.csv" which we henceforth refer to as the train, test, mapping datasets. The train dataset consists of 90,766 rows with 50 columns, the test dataset contains 11,000 rows with 50 columns, and the mapping dataset contains 67 rows of different sections. In the train dataset, only two features exhibit significant missing values: max_glu_serum (85,581) and A1Cresult (75,927). Similarly, in the test dataset, three features show notable missing values: max_glu_serum (10,839), A1Cresult (8,821), and readmitted (11,000). The distribution of the target variable is unbalanced: NO (49,361), >30 (31,232), and <30 (10,173).

II. Data cleaning

First, we replace the string '?' with the conventional pd.NA, since NA can be handled later. We then visualize the result. We dropped the "weight" column since it is not too important in the prediction, and it has too many missing values. We check the number of unique values in each feature to eradicate redundancies, and then drop the features that have only one unique value.

III. Methodology

A. The used software

We utilized the Python 3.14 programming language along with the following libraries:

- Pandas
- NumPy
- Matplotlib
- Scikit-learn
- Joblib
- Imbalanced-learn
- Multi-Layer Perceptron (MLP) Classifier
- Random Forest Classifier

B. Instructions on how to download and install

Python can be downloaded from the official website*. After downloading, we can follow the instructions on the website for installation.

For all the libraries used, we can use pip†, such as *pip install library*.

C. Exploration, preparation and modeling

We explored each feature and visualized them one after the other using pie charts. Firstly, we visualized the Caucasian feature and found 2.4% missing values. These were filled with the mode. Next, we visualized the gender feature and found some unknown/invalid entries, which were deemed insignificant. However, we replaced these unknown/invalid entries with the mode of female, assuming they might possibly be female since the percentage of females is higher than males. Moving on to age, we found no missing values. We discovered that the age feature could be grouped, so we grouped it as follows: age 0-40 (0), 50-60 (1), 60-70 (2), 70-80 (3), 80-100 (4).

Furthermore, we explored the third dataset to map the admission ID to its description in the train data. Visualizing the result, we discovered that Emergency was the most prevalent admission type, followed by urgent attention. Similarly, we explored the discharge disposition ID. However, visualization was challenging due to the large number of labels in this category, causing overlap. Thus, we grouped them into rich_health_care, limited_health_care_not_home, home, expired_hospice, and no_info. It became evident that many people's discharge disposition was Home, followed by limited health care not at home.

We then explored the admission ID to understand the source of admission. It became apparent that the majority of admissions were from the emergency room, followed by physician referral. Next, we explored the time spent in the hospital by each patient. The majority

*https://www.python.org/downloads/
†https://pypi.org/project/pip/

of patients spent an average of 1-3 weeks. Additionally, we examined the number of lab procedures and found that the majority of patients had 0-1 procedures. The number of medications received by each patient was also investigated, revealing an average of 15 medications given to each patient.

Furthermore, we explored the outpatient visits of patients in the year preceding the encounter. Most had 0 outpatient visits, while the number of emergency visits was mostly 0 and the average number of inpatient visits was 1. We also delved into the diagnosis column, but due to the large amount of data on the x-axis, we categorized codes into 9 categories and used them as diagnoses of diseases falling under those categories. We found that most patients had an average of 9 diagnoses.

All other explorations, such as dosage changes for medication, feature clustering, and feature importance, can be found in our replication and submission package. We cannot itemize all here due to space limitations.

D. Other unfruitful approaches

Initially, we experimented with the Random Forest model, considering its reputation as one of the top classifiers in this domain. However, we observed that this model achieved 100% accuracy on the training data but only 55% on the test data, indicating clear overfitting. Consequently, we opted for the Multi-Layer Perceptron (MLP), a type of artificial neural network, as an alternative approach. Before settling for MLP, we also disabled SMOTE to explore its potential in enhancing the result. However, disabling SMOTE did not improve the outcome. Additionally, we attempted to convert all numeric data into categorical variables. Unfortunately, this adjustment had no effect on the result.

IV. Result

To achieve a good result, we first dropped some features that are not useful, such as encounter_id2 and patient_nbr2. We then used the remaining features as input and considered readmitted as the label. We applied one-hot encoding and split the data into training and test sets. To address the imbalance in the variable to be predicted, we applied SMOTE to oversample the training data exclusively.

Our baseline predictor was a random forest model, with a Training Accuracy of 1.0, testing accuracy of 0.57, Precision of 0.53, Recall of 0.57, F1-score of 0.53, and AUC-ROC of 0.64. We computed and visualized the feature importance, noting that the top five most important features were: num_lab_procedures (0.08), num_medications (0.07), time_in_hospital (0.06), age (0.04), and number_diagnoses (0.04). Due to observed overfitting in the training result of the Random Forest, we opted for a Multi-Layer Perceptron (MLP) to mitigate this issue.

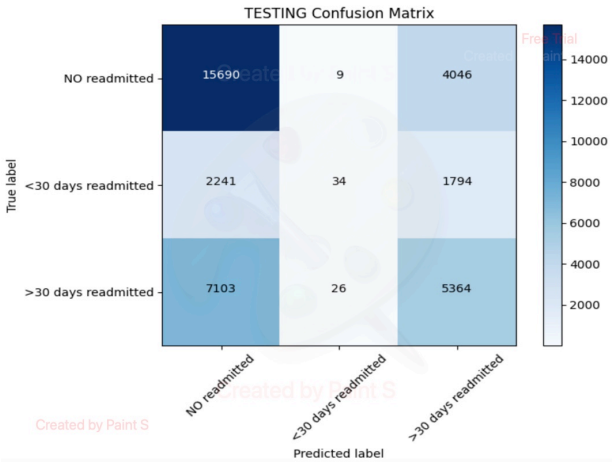

Fig. 1: The confusion matrix of the most performed models in the testing data

For training the MLP, we specified hidden_layer_sizes=(35, 70, 3), learning_rate_init=0.01, and max_iter=50000. This training process lasted for a couple of hours, resulting in an accuracy of 57%. Not satisfied with this performance, we experimented with hyperparameters. First, we decreased the learning rate, resulting in an accuracy of 54%, which was worse than before. Second, we changed the activation function from relu to logistic, but the accuracy remained at 54%. Third, changing the solver from adam to sgd led to an accuracy of 55%, still not an improvement. Fourth, increasing the complexity of the model by extending the hidden_layer_sizes from 3 to 4 resulted in an accuracy of 51%, which was even worse. Finally, simplifying the network by reducing the hidden_layer_sizes from 4 to 2 yielded the best result so far, with an accuracy of 58%. Figure 1 shows the confusion matrix of the most performed model in the testing data.1

Moving forward, we utilized the last trained model with simple complexity as a predictor on our main testing dataset. We made predictions for each row and appended them accordingly. The prediction results can be found in the csv file C2T1_Test_Labeled.csv within our replication package. This csv contains three columns: encounter_id, patient_nbr, and readmitted (the predicted column). Detailed information on all experiments, trained models, and confusion matrices for each model can be found in our replication package on GITHUB[‡].

[‡]https://github.com/16sl75/Hospital-Readmission