Daniel Addokwei Tetteh

Intelligent Data Analytics

Homework 7\_Group\_7

Kaggle (O)-Curiosity-7

2022-11-15

PROJECT DESCRIPTION

A hospital readmission is an episode when a patient who had been discharged from a hospital is admitted again within a specified time interval. Readmission rates have increasingly been used as an outcome measure in health services research and as a quality benchmark for health systems. Hospital readmission rates were formally included in reimbursement decisions for the Centres for Medicare and Medicaid Services (CMS) as part of the Patient Protection and Affordable Care Act (ACA) of 2010, which penalizes health systems with higher-than-expected readmission rates through the Hospital Readmission Reduction Program. The real-world clinical care data provided in this competition comes from multiple hospitals across the United States for several years.

**The challenge**: In this machine learning problem, you have the exacting task of trying to predict whether or not a patient will be readmitted to the hospital. The target takes on binary values where 0 implies that the patient was not readmitted, and 1 implies that the patient was readmitted. You are predicting the latter. The data and many details about the problem can be found on the course

Your classification model will be evaluated using log loss. Your predictions, therefore, will be probabilities of readmission. You will be evaluated on how well you can predict the hospital readmits on a test data set available on the Kaggle.com class competition page.

Here is the link to the competition website: https://www.kaggle.com/competitions/5103-ida-hm7-2022

**FINAL PROJECT REPORT**

-Important Libraries

library(xgboost)  
library(randomForest)  
library(knn)  
library(earth)  
library(kernlab)  
library(RevoScaleR)  
library(rpart.plot)  
library(dplyr)  
library(tidyverse)  
library(readr)  
library(VIM)  
library(plotly)  
library(corrplot)  
library(caret)  
library(car)  
library(AppliedPredictiveModeling)  
library(dplyr)  
library(MASS)  
library(lars)  
library(elasticnet)  
library(reshape)  
library(glmnet)  
library(lubridate)  
library(e1071)  
library(vtreat)  
library(binaryLogic)  
library(glmnet)  
library(tidymodels)  
library(Metrics)  
library(Amelia)  
library(naniar)  
library(outliers)  
library(ggpubr)  
library(DAAG)  
library(party)  
library(rpart.plot)  
library(pROC)  
library(tree)  
library(naniar)  
library(Amelia)  
library(mice)  
library(rpart)  
library(ROCR)  
library(rattle)  
library(adabag)  
library(ipred)

**Exploratory Data Analysis:** Data preprocessing steps include summary of data structure, division of data into numeric and non-numeric data sets, summary report of both numeric and non-numeric data sets, visualization of missing values and imputation of missing values.

From the data, I realized that admission\_type, discharge dispositions and admission source are nominal data, with each factor represented by a number. These were converted to factors and included in the non-numeric data set.

#Changining nominal data to factors and adding them to non-numeric set

Non\_numeric\_data$admission\_type <-as.factor(Numeric\_data$admission\_type)  
Non\_numeric\_data$discharge\_disposition <- as.factor(Numeric\_data$discharge\_disposition)  
Non\_numeric\_data$admission\_source <- as.factor(Numeric\_data$admission\_source)  
glimpse(Non\_numeric\_data)

A plot of readmitted vs patientID on the training data showed that there is an even distribution in the number of patients readmitted and not indicating a good balance within the target variable

Chart

Description automatically generated

Fig 1 Plot of target classes

**Data quality report**

Numeric Variables

Table

Description automatically generated with medium confidence

Data quality report for numeric variables

Non-numeric variables

Table

Description automatically generated

Data quality report for first few variables in non-numeric variables

**Missing Values Visualization and Imputation**

Chart

Description automatically generated

Fig 2-Missing values in numeric variables

Graphical user interface, application, table, Excel

Description automatically generated

Chart, treemap chart

Description automatically generated

Fig 3 Missing values in non-numeric variables

In the numeric dataset, missing values were observed in indicator\_level, time\_in\_hospital and number\_of\_lab\_procedures. For non\_numeric, missing values were identified in medical\_specialty, payercode, race and diagnosis. Predictive mean matching and mode imputations were used for numeric and non-numeric missing values imputations.

Chart

Description automatically generated

Fig 4 Correlation plot between numeric variables and target variables

**MODELLING**

1. Basic Model Development: A basic model was created in MARS using only numeric variables in good correlation with the target variable “readmitted”. The metrics of this model is reported in table 1 below. Selected numeric variables include number\_inpatient, number\_emergency, number\_outpatient, num\_medications, time\_in\_hospital, num\_lab\_procedures and number\_diagnoses. The model performed poorly
2. Next, a more rigorous data preprocessing was done with factor level collapsing, encoding and feature selection. The final data set was used in building five more classification models. i.e. Random Forest, K Nearest Neighbours, Support Vector Machines, Gradient Boosting and Logistics Regression. The results of these models are discussed in (a) below.

**Important Steps and Trends in Rigorous preprocessing**

1. The train and test data sets after the initial preprocessing were combined to form a new dataset.

# Train data

TrainOR =cbind(numeric1,Non\_numeric\_data\_1)  
glimpse(TrainOR)  
  
TrainOR = subset(TrainOR, select=-c(missing\_1,missing\_2,missing\_3))  
Train\_df=TrainOR  
  
####Test data  
testdata  
testdata= subset(testdata, select=-c(missing\_a,missing\_b))  
glimpse(testdata)  
  
  
# Combine the train and test data  
test\_df=testdata  
test\_df$readmitted = NA  
test\_df$readmitted = as.factor(test\_df$readmitted)  
prediction\_df <- rbind(Train\_df,test\_df)

1. To ensure proper feature selection and engineering, a relationship between some significant variables and the target are plotted as shown below.

Chart, bar chart, histogram

Description automatically generated

Chart, bar chart

Description automatically generated

Fig 5 – Exploratory relationship between some categorical variables and the target variables.

Encoding

Based on insights obtain from the data, some variables were dropped, and others encoded as indicated in the R code attached.

prediction\_df$gender <- as.numeric(as.factor(prediction\_df$gender))  
prediction\_df$age <- as.numeric(as.factor(prediction\_df$age))  
prediction\_df$payer\_code <- as.numeric(prediction\_df$payer\_code)  
prediction\_df$medical\_specialty <- as.numeric(prediction\_df$medical\_specialty)  
prediction\_df$diagnosis <- as.numeric(prediction\_df$diagnosis)  
#prediction\_df$readmitted <- as.numeric(as.factor(prediction\_df$readmitted))  
prediction\_df$insulin <- as.numeric(as.factor(prediction\_df$insulin))  
prediction\_df$diabetesMed <- as.numeric(as.factor(prediction\_df$diabetesMed))  
prediction\_df$metformin <- as.numeric(as.factor(prediction\_df$metformin))  
  
glimpse(prediction\_df)  
  
# Max Glu Serum  
prediction\_df$max\_glu\_serum <- as.numeric(as.factor(prediction\_df$max\_glu\_serum))  
prediction\_df$max\_glu\_serum[prediction\_df$max\_glu\_serum == 1] <- 1  
prediction\_df$max\_glu\_serum[prediction\_df$max\_glu\_serum == 2] <- 1  
prediction\_df$max\_glu\_serum[prediction\_df$max\_glu\_serum == 4] <- 1  
prediction\_df$max\_glu\_serum[prediction\_df$max\_glu\_serum == 3] <- 0  
  
# A1 Cresult  
prediction\_df$A1Cresult <- as.numeric(as.factor(prediction\_df$A1Cresult))  
prediction\_df$A1Cresult[prediction\_df$A1Cresult == 1] <- 1  
prediction\_df$A1Cresult[prediction\_df$A1Cresult == 2] <- 1  
prediction\_df$A1Cresult[prediction\_df$A1Cresult == 4] <- 1  
prediction\_df$A1Cresult[prediction\_df$A1Cresult == 3] <- 0  
  
# Repaglinide  
prediction\_df$repaglinide <- as.numeric(as.factor(prediction\_df$repaglinide))  
prediction\_df$repaglinide[prediction\_df$repaglinide == 1] <- 1  
prediction\_df$repaglinide[prediction\_df$repaglinide == 3] <- 1  
prediction\_df$repaglinide[prediction\_df$repaglinide == 4] <- 1  
prediction\_df$repaglinide[prediction\_df$repaglinide == 2] <- 0  
  
# Chlorpropamide  
prediction\_df$chlorpropamide <- as.numeric(as.factor(prediction\_df$chlorpropamide))  
prediction\_df$chlorpropamide[prediction\_df$chlorpropamide == 2] <- 1  
prediction\_df$chlorpropamide[prediction\_df$chlorpropamide == 3] <- 1  
prediction\_df$chlorpropamide[prediction\_df$chlorpropamide == 4] <- 1  
prediction\_df$chlorpropamide[prediction\_df$chlorpropamide == 1] <- 0  
  
# Glimepiride  
prediction\_df$glimepiride <- as.numeric(as.factor(prediction\_df$glimepiride))  
prediction\_df$glimepiride[prediction\_df$glimepiride == 1] <- 1  
prediction\_df$glimepiride[prediction\_df$glimepiride == 3] <- 1  
prediction\_df$glimepiride[prediction\_df$glimepiride == 4] <- 1  
prediction\_df$glimepiride[prediction\_df$glimepiride == 2] <- 0  
  
# Glipizide  
prediction\_df$glipizide <- as.numeric(as.factor(prediction\_df$glipizide))  
prediction\_df$glipizide[prediction\_df$glipizide == 1] <- 1  
prediction\_df$glipizide[prediction\_df$glipizide == 3] <- 1  
prediction\_df$glipizide[prediction\_df$glipizide == 4] <- 1  
prediction\_df$glipizide[prediction\_df$glipizide == 2] <- 0  
  
# Glyburide  
prediction\_df$glyburide <- as.numeric(as.factor(prediction\_df$glyburide))  
prediction\_df$glyburide[prediction\_df$glyburide == 1] <- 1  
prediction\_df$glyburide[prediction\_df$glyburide == 3] <- 1  
prediction\_df$glyburide[prediction\_df$glyburide == 4] <- 1  
prediction\_df$glyburide[prediction\_df$glyburide == 2] <- 0  
  
# Pioglitazone  
prediction\_df$pioglitazone <- as.numeric(as.factor(prediction\_df$pioglitazone))  
prediction\_df$pioglitazone[prediction\_df$pioglitazone == 1] <- 1  
prediction\_df$pioglitazone[prediction\_df$pioglitazone == 3] <- 1  
prediction\_df$pioglitazone[prediction\_df$pioglitazone == 4] <- 1  
prediction\_df$pioglitazone[prediction\_df$pioglitazone == 2] <- 0  
  
# Rosiglitazone  
prediction\_df$rosiglitazone <- as.numeric(as.factor(prediction\_df$rosiglitazone))  
prediction\_df$rosiglitazone[prediction\_df$rosiglitazone == 1] <- 1  
prediction\_df$rosiglitazone[prediction\_df$rosiglitazone == 3] <- 1  
prediction\_df$rosiglitazone[prediction\_df$rosiglitazone == 4] <- 1  
prediction\_df$rosiglitazone[prediction\_df$rosiglitazone == 2] <- 0

Finally, the train and test data sets are separated for the main modelling phase. The final data set had 29 columns consisting of both numeric variables and encoded categorical variables.

(a)

* Five different models were created using the following algorithms: Random Forest, Gradient Boosting, K-Nearest Neighbors, Penalized Logistic Regression, and Support Vector Machines as seen in the R-code attached. The performance of the models is summarized in the Table below. The best model was the gradient boost model with an accuracy and kappa score of 0.651 and 0.294.

**A picture containing table

Description automatically generated**

Table 1: Summary of model results

**Potential Insights from Model**

The gradient boost classification model was taken and evaluated to obtain relevant insights that may be valuable to stakeholders.

The following insights can be obtained from the feature importance table below.

Table

Description automatically generated

Fig 6. Variable importance from Gradient Boost model

1. From the variable importance chart, we can see that the top five variables that help in determining whether a patient will be readmitted include the number of inpatients, diagnosis, number of diagnoses, age, and number of emergencies. Based on this, hospitals should be able to predict the chances of a patient being readmitted and adjust. For instance, when a patient has many inpatient visits, there is a high chance that he or she will be readmitted.
2. The age, number of diagnoses and the number of emergency cases as shown in the feature engineering chart also contribute significantly to the readmission of a patient. Thus, insurance companies can tailor their insurance premium charges based on the age and medical health records of their customers (typically if they have had a high number of emergency cases or diagnoses in the past). From fig 5, we can see that probability of readmitting a patient based on their age group is fairly even. i.e. there is a close to an equal chance of readmitting a patient or not based on their age. However, number of readmission cases is very high as age of patients increases. Insurance companies can take a cue from this charge health insurance premiums based on age of customers.
3. The model also shows that the time a patient spends in a hospital, the type of admission and the payer code as well as whether a patient has diabetes are not very significant determinants of whether or not a patient will be readmitted. This can be very important to the government in regulating health insurance premium charges by private health insurance companies.

**Quantification of Performance Evaluation Techniques of Gradient Boost Model**

Here, I will use the following performance evaluation techniques; accuracy and KAPPA from final xgboost model, Recall from confusion matrix, and ROC curve

The best model after the 10-fold cross-validation with gridsearch for the xgboost model is shown in the table 1 above and reported the listed metric; Accuracy= 0.633 and Kappa= 0.257. The best tune was saved as “model\_final\_xgboost” as shown in the R-codes attached. The hyperparameter values are shown in the table and the fig below.



The result from the confusion matrix is as shown as follows,

Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **0** | 22871 | 12411 |
| **1** | 7761 | 14812 |

A screenshot of a computer

Description automatically generated with low confidence

Summary of Metrics

|  |  |  |
| --- | --- | --- |
| Metric | Value | Predictive Quality Quantification based on metric |
| Recall (from confusion matrix) | 0.544 | The recall is the ratio of true positive to the sum of true positive and false negative. That is how well does the model predict a class (1) correctly. This needs to be as high as possible. The recall value from our model diagnosis and performance evaluation indicates that about 54% of the time, the model will call a predicted class of “1” a “0” which is not good enough. The model thus require improvement. A more rigorous feature engineering or re-tuning. |
| xgboost\_model Accuracy |  | Accuracy is basically the ratio of correct to wrong predictions. A value of 0.633 means that the model makes a right prediction of whether a patient will be readmitted or not about 63% of the time. This is fairly good given the fact that there is an almost equal probability of predicting a “0” or a “1” from the original data set. i.e. the original data set has almost equal rows with target classes “0” and “1”. It had a good balance. The model however needs to be improved to an accuracy of about 0.8. |
| xgboost\_model Kappa | 0.257 | Kappa basically compares the observed and expected accuracies of a classifier. i.e. number of times the classifier correctly classifies observations as compared to its expected classification power. Usually a kappa between 0.2 and 0.4 is considered fairly good. |

ROC Curve

The ROC curve compares the True positive rate and the false positive rate of a classifier based on a threshold value. i.e. how many of the actual positives are classified as positive versus how many of the actual negatives are classified as positive. The threshold value taken here was 0.5 based on the balance observed on the original data. We can see that the true positive rate is higher than the false positive rate. The model classifies a target class of “1” as “1” better than it classifies a “0” as 1.

Chart, line chart

Description automatically generated

(b) The xgboost model was submitted on Kaggle and yielded a log loss of 0.63798. It placed within the top 15 on the leaderboard.