ISE 5103 Intelligent Data Analytics

Group-4 Homework #7

Tushar Jayendra Mhatre, Roshini Talluru, Rahul Kataram

(a)

(i) We used the following five models:

- Logistic Regression
- XGBoost
- Decision Tree
- MARS
- Random forest

Note: Please refer to lines 373-539 in the R code.

Important Observations:

- We also tried the C5.0 tree model which is also in the R code. However, we weren't able to get accuracy and kappa values for this model.
- We also observed that the logistic regression model is the fastest model to run but XGboost is giving a better model performance.
- Random Forest is taking too long to run so we used less number of predictor variables to run the model which decreased the model performance.

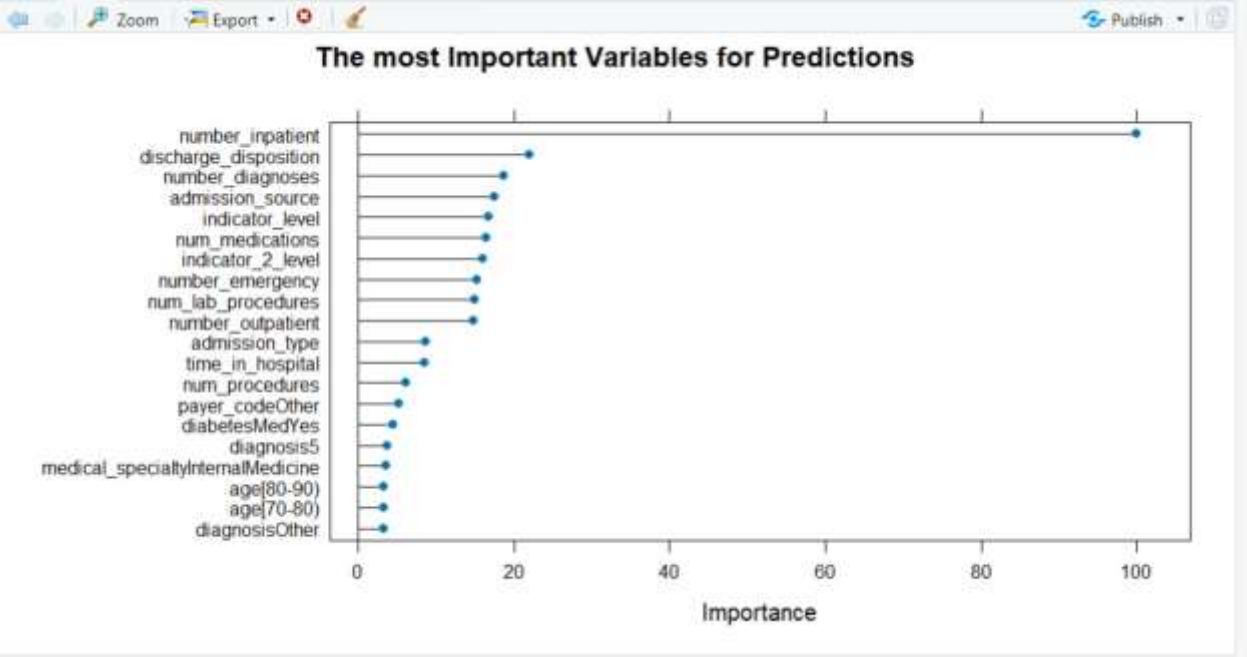
(ii) Model Performance Summarizations:

The below table shows the various model performance summarizations (accuracy, kappa value).

					CV Performance	
Model	Method	Package	Hyperparameter	Selection	Accuracy	Kappa
Logistic Regression	glm	stats	NA	NA	0.629	0.246
XGBoost Model	xgbTree	xgboost	nrounds, max_depth, eta, gamma, colsample_bytree, min_child_weight, subsample	c(400,450,500), c(3,4,5), c(0.05,0.1,0.5), 0.01, 0.6, 0, 0.75	0.641	0.275
Decision Tree	rpart	rpart	cp	0.00154	0.622	0.233
MARS	earth	earth	nprune	2	0.626	0.238
Random forest	rf	randomForest	mtryStart, stepFactor, improve	2, 1.5, 0.05	0.552	0.0681

(iii) Three potential "Insights" relating to Hospital Readmits:

We are considering the XGBoost model. The performance of the XGBoost is far better than compared with the others. Among the various approaches we did to increase model performance collapsing factors and missing value imputation increased the model performance.



The above graph gives us an idea of the significant and necessary variables which affect the data predictions.

We can retrieve the following insights from the graph:

Inpatient Visits and Readmission Correlation:

- The graph suggests a positive correlation between the number of inpatient visits and the likelihood of patient readmission. Higher inpatient visits seem to indicate an increased chance of subsequent readmission.
- Hospitals and healthcare providers should pay attention to patients with a history of multiple inpatient visits. These individuals may require closer monitoring, personalized care plans, and proactive interventions to mitigate the risk of readmission. Implementing strategies to improve care transitions and post-discharge follow-ups for frequent inpatient visitors could be beneficial.

Impact of Indicator Level on Readmissions:

- The indicator level (presumably indicating severity or risk level) is shown to have a significant influence on patient readmission. Higher indicator levels are associated with increased chances of readmission.
- Patients with higher indicator levels may require more intensive care or specialized interventions to prevent readmissions. Hospitals and healthcare providers can prioritize these patients for targeted interventions, early interventions, or specialized care plans tailored to their specific needs.

Equally Influential Features: Lab Procedures, Medication, and Diagnoses:

- The features related to the number of lab procedures, medication, and diagnoses exhibit nearly equal importance in predicting patient readmissions. Higher values in these features are associated with an increased likelihood of readmission.
- Monitoring and managing the number of lab procedures, medications prescribed, and diagnoses could be crucial in reducing readmission rates. Hospitals and healthcare providers may need to evaluate these factors more comprehensively and ensure appropriate management, potentially focusing on medication adherence, diagnostic accuracy, and streamlined lab procedures to optimize patient outcomes and reduce readmissions.

These insights highlight the importance of certain patient indicators and healthcare factors that significantly influence the likelihood of readmission. Leveraging this information can aid hospitals, doctors, and healthcare systems in developing targeted strategies and interventions aimed at reducing readmission rates and improving overall patient care

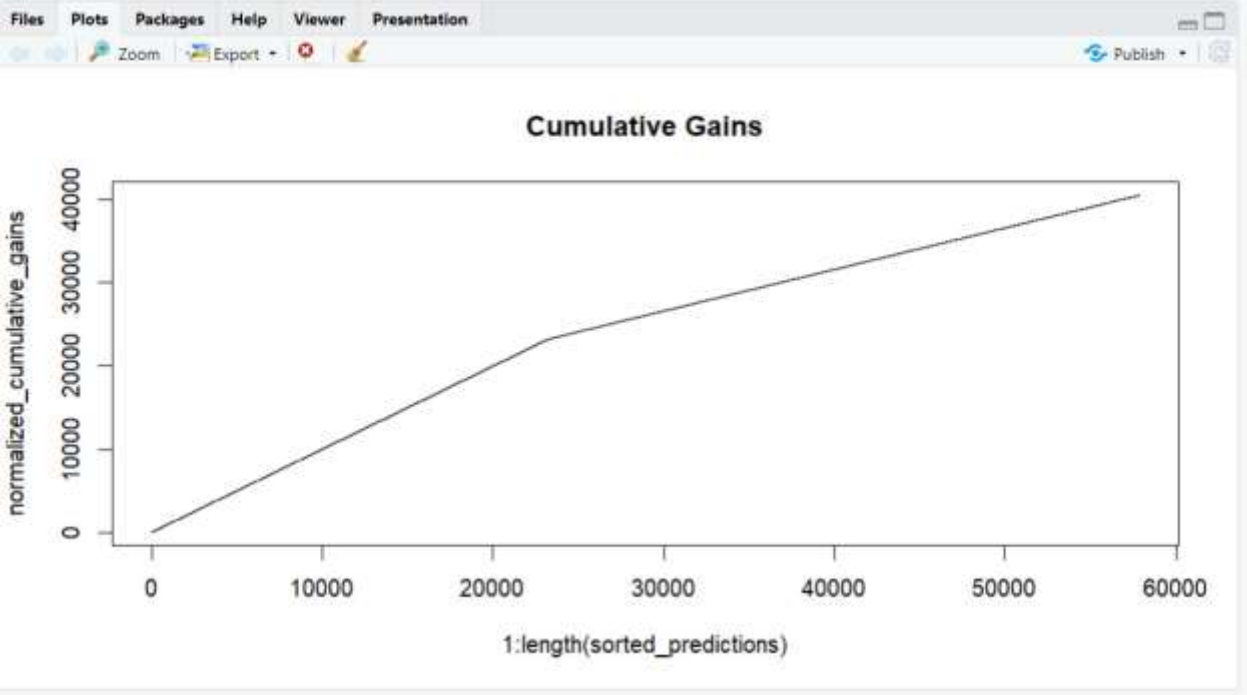
(iv) Performance Evaluation Techniques

Cumulative Gain:

Cumulative Gain is a metric used to evaluate the effectiveness of a model or a ranking system in capturing the relevant instances or outcomes, particularly in scenarios involving ranked predictions.

Cumulative Gain helps in understanding the proportion of relevant instances captured when considering a subset of the total predictions.

Below is the cumulative gain graph of XGboost



- From the plot, we can observe that it is an increasing graph which indicates that the model's ranked predictions are capturing relevant instances effectively as you move through the ranked list.
- An increasing CG graph suggests that relevant items are being appropriately prioritized and captured early in the ranked list. It implies that the model is successful in identifying and ranking relevant instances higher.
- The model is performing well in terms of ranking the most relevant items toward the top of the list. This is crucial in scenarios like recommendation systems or information retrieval, where presenting the most relevant results early is essential for user satisfaction. It is beneficial if we know the readmission rates earlier which can help us to control the increasing rates of inpatients.

XGBoost Model:

From below it can be seen, that 64.1% of the model predicted accurately.

```
> XGBoost_Model
eXtreme Gradient Boosting

57855 samples
   28 predictor
    2 classes: 'No', 'Yes'

Pre-processing: centered (124), scaled (124)
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 52069, 52070, 52070, 52070, 52069, 52069, ...
Resampling results:

  Accuracy  Kappa
  0.641     0.275

Tuning parameter 'nrounds' was held constant at a value of 500
Tuning
 held constant at a value of 0
Tuning parameter 'subsample' was held
 constant at a value of 0.75
```

Confusion Matrix:

A confusion matrix is a table often used to describe the performance of a classification model.

A confusion matrix typically consists of four terms:

True Positives (TP): These are the cases where the model predicted the positive class correctly.

True Negatives (TN): These are the cases where the model predicted the negative class correctly.

False Positives (FP): These are the cases where the model predicted the positive class incorrectly (it's actually negative).

False Negatives (FN): These are the cases where the model predicted the negative class incorrectly (it's actually positive).

Accuracy:

Accuracy, in the context of classification models, is a metric that measures the proportion of correct predictions made by the model overall predictions made.

It's calculated using the formula:

$$\text{Accuracy} = \frac{\text{Total Number of Predictions}}{\text{Number of Correct Predictions}}$$

In a confusion matrix, accuracy can be calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP+TN+FP+FN}$$

Where:

TP = True Positives (correctly predicted positive instances)

TN = True Negatives (correctly predicted negative instances)

FP = False Positives (incorrectly predicted positive instances)

FN = False Negatives (incorrectly predicted negative instances)

We were able to get 0.667 Accuracy in the confusion matrix for our XGboost model.

Kappa Value:

The kappa value indicates the level of agreement between predicted and observed classifications. If we have a kappa value less than 0.2 it means a weak agreement, 0.2-0.6 is considered a moderate agreement. Values greater than 0.6 are a very good agreement. We were able to get 0.3262 which falls under moderate agreement.

```
> confusionMatrix(xgb_Tr_preds,as.factor(Traincollapsed$readmitted),mode = "eve
rything")
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  22993 11626
       Yes  7639 15597

               Accuracy : 0.667
                 95% CI : (0.6632, 0.6709)
    No Information Rate : 0.5295
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3262

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7506
            Specificity : 0.5729
         Pos Pred Value : 0.6642
         Neg Pred Value : 0.6712
              Precision : 0.6642
                 Recall : 0.7506
                     F1 : 0.7048
             Prevalence : 0.5295
         Detection Rate : 0.3974
   Detection Prevalence : 0.5984
      Balanced Accuracy : 0.6618

       'Positive' Class : No

>
```

Log Loss:

Log Loss, also known as logarithmic loss or cross-entropy loss, is a metric used to evaluate the performance of a classification model, particularly in scenarios involving probabilistic predictions.

We were able to get a 0.6 log loss for our model. It is considered typically a good value but if we get less than that it means the model performance is still better.

```
> #Alternate way:Calculate LogLoss manually
> predicted_probs <- xgb_Tr_preds$Yes
> actual_labels <- as.numeric(Traincollapsed$readmitted == "Yes")
> log_loss <- -mean(actual_labels * log(predicted_probs) + (1 - actual_labels)
* log(1 - predicted_probs))
> print(log_loss)
[1] 0.6062757
>
```