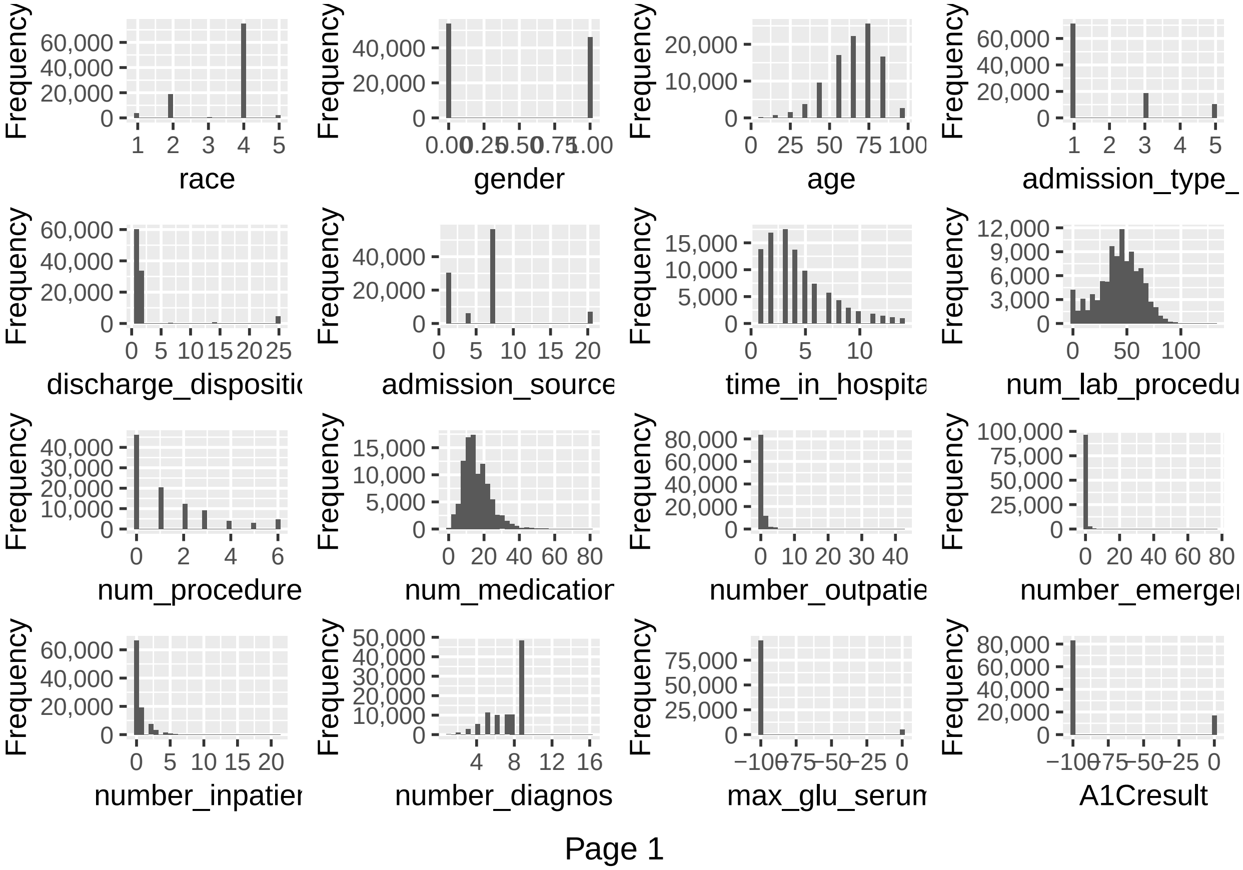
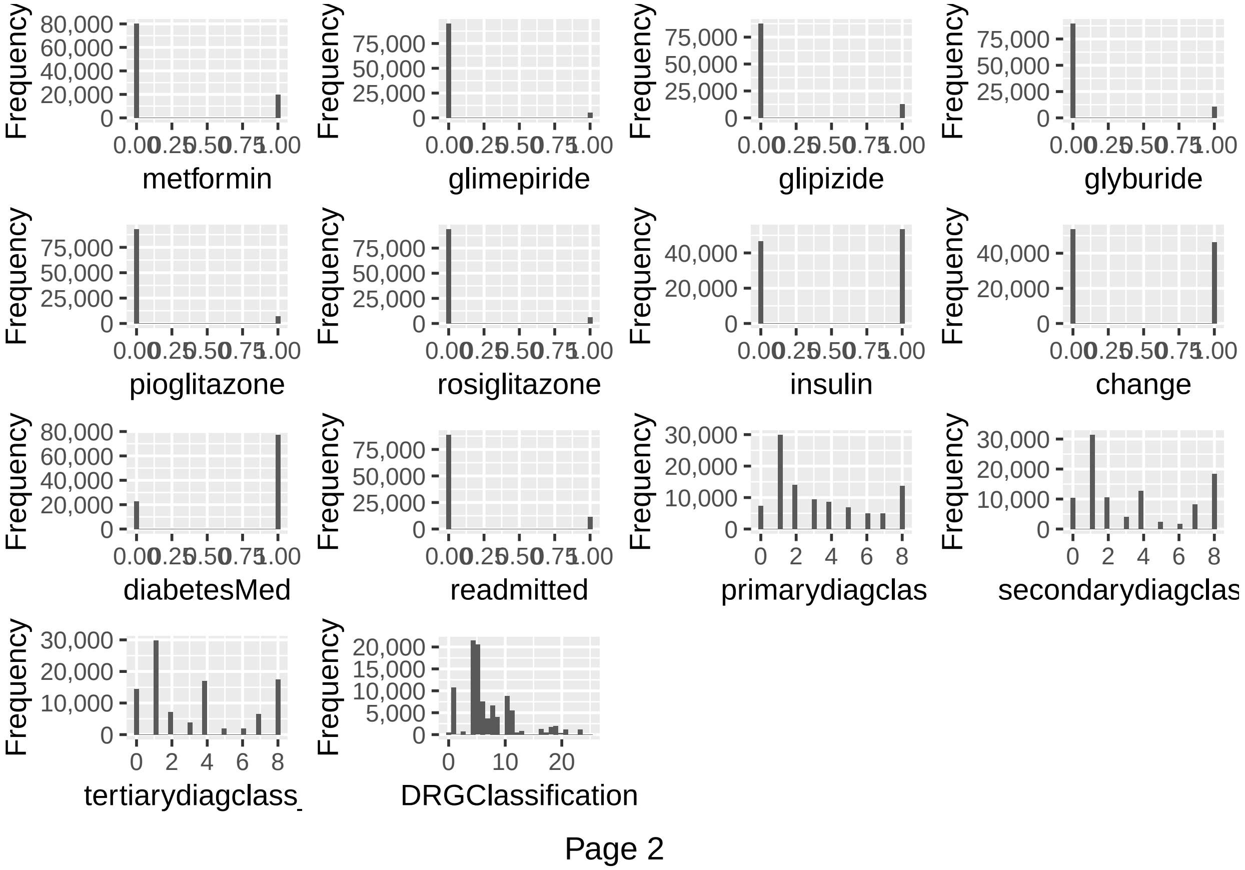
# Plots

*Dilip Ganesan*

*4/20/2019*

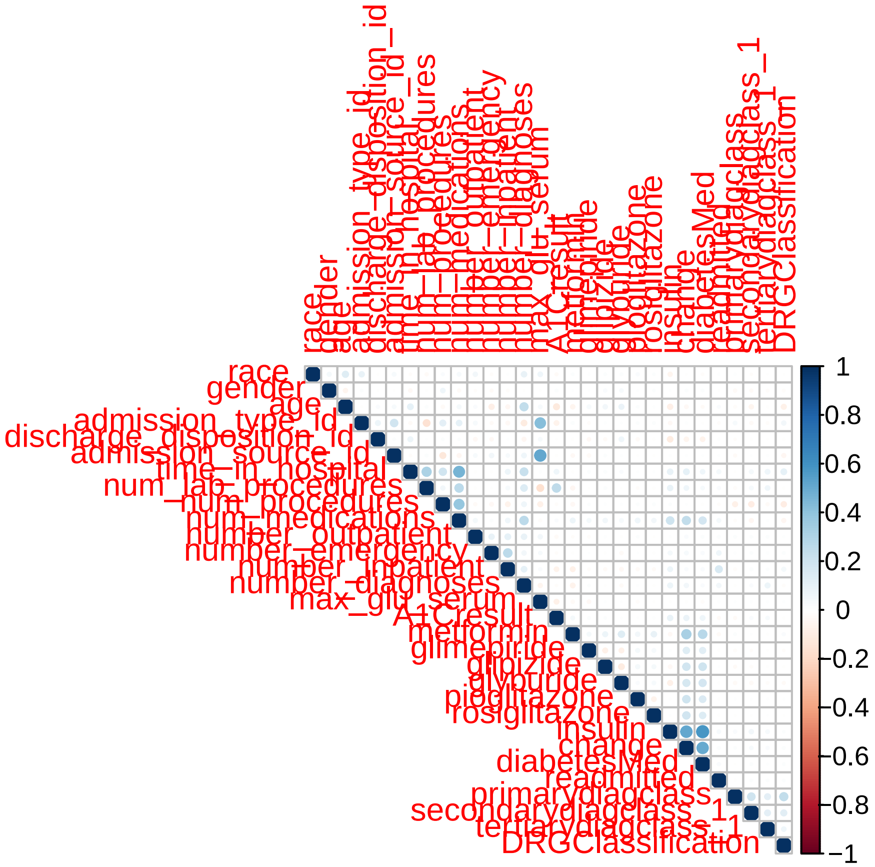
```
#Load data set
dataset <- read.csv("https://raw.githubusercontent.com/dilipganesan/DATA698/master/Cleaned_Data/Cleaned_
#dataset
# Cleaning the column X.
dataset = dataset %>%
            select(-c("X"))

dataset = dataset %>%
            select(-c("Primary_Diag","Secondary_Diag_1","Secondary_Diag_2", "Payment","diag_1","diag_2"




############ Removing some of medication data from data set because of class imbalance.
newdataset = dataset %>%
            select(c("race","gender","age", "admission_type_id","discharge_disposition_id","admission_so

##########Plots############################
# variable distributions

plot_histogram(newdataset)
```


Frequency
60,000
40,000
20,000
0
1 2 3 4 5
race
40,000
20,000
0
0.00 0.25 0.50 0.75 1.00
gender
20,000
10,000
0
0 25 50 75 100
age
60,000
40,000
20,000
0
1 2 3 4 5
admission_type_
60,000
40,000
20,000
0
0 5 10 15 20 25
discharge_dispositic
40,000
20,000
0
0 5 10 15 20
admission_source
15,000
10,000
5,000
0
0 5 10
time_in_hospita
12,000
9,000
6,000
3,000
0
0 50 100
num_lab_procedu
40,000
30,000
20,000
10,000
0
0 2 4 6
num_procedure
15,000
10,000
5,000
0
0 20 40 60 80
num_medicatior
80,000
60,000
40,000
20,000
0
0 10 20 30 40
number_outpatie
100,000
75,000
50,000
25,000
0
0 20 40 60 80
number_emerge
60,000
40,000
20,000
0
0 5 10 15 20
number_inpatier
50,000
40,000
30,000
20,000
10,000
0
4 8 12 16
number_diagnos
75,000
50,000
25,000
0
−100 −75 −50 −25 0
max_glu_serun
80,000
60,000
40,000
20,000
0
−100 −75 −50 −25 0
A1Cresult
Page 1



### Correlation Plots

```
cordata = cor(newdataset)
corrplot(cordata, method = "circle", type = "upper")
```



**Training and Test Data for Logistic Regression:**

As a first step in data preprocessing, splitting of training and test data set, is to check the class bias. In our dataset there is class bias in our target variable

Ideally, the proportion of events and non-events in the target variable should approximately be the same. So, lets first check the proportion of classes in the dependent variable readmitted.

```
knitr::kable(table(newdataset$readmitted ))
```

| Var1 | Freq |
|---|---|
| 0 | 88603 |
| 1 | 11347 |

```
### Checking of class bias. The number of events happening is less than events not happening. So for pr
```

Clearly, there is a class bias, a condition observed when the proportion of events is much smaller than proportion of non-events. So we must sample the observations in approximately equal proportions to get better models.

As a next step we are going through the process to remove class bias.

One way to address the problem of class bias is to draw the 0s and 1s for the trainingData in equal proportions. In doing so, we will put rest of the inputData not included for training into testData. As a result, the size of trainingData sample will be smaller that validation.

Once the trainingData and testData are created from our dataset, the next step is to create the Binary

Logistic Regression.

### Binary Regression Base Model

As first step, we are going to run our model using all the variables that are available in the data set. This includes DRGClassification also as predictor variable.

```
logitMod <- glm(readmitted ~ ., data=trainingData, family=binomial(link="logit"))
summary(logitMod)
```

```
##
## Call:
## glm(formula = readmitted ~ ., family = binomial(link = "logit"),
##     data = trainingData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8241  -1.0951  -0.2939   1.1718   1.8235
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.2471730  0.1678836  -7.429 1.10e-13 ***
## race                     -0.0165265  0.0181249  -0.912 0.361866
## gender                    0.0229196  0.0332869   0.689 0.491108
## age                       0.0062533  0.0011351   5.509 3.61e-08 ***
## admission_type_id        -0.0148371  0.0146160  -1.015 0.310047
## discharge_disposition_id  0.0032313  0.0031603   1.022 0.306558
## admission_source_id      -0.0098438  0.0042409  -2.321 0.020278 *
## time_in_hospital          0.0188556  0.0066201   2.848 0.004396 **
## num_lab_procedures        0.0008208  0.0009627   0.853 0.393887
## num_procedures           -0.0180566  0.0111157  -1.624 0.104287
## num_medications           0.0074724  0.0027006   2.767 0.005657 **
## number_outpatient        -0.0140258  0.0125846  -1.115 0.265055
## number_emergency          0.0712899  0.0208360   3.421 0.000623 ***
## number_inpatient          0.3123913  0.0140707  22.202  < 2e-16 ***
## number_diagnoses          0.0569270  0.0097778   5.822 5.81e-09 ***
## max_glu_serum             0.0026733  0.0009683   2.761 0.005764 **
## A1Cresult                -0.0011097  0.0004718  -2.352 0.018667 *
## metformin                -0.1697504  0.0479709  -3.539 0.000402 ***
## glimepiride              -0.1426876  0.0802128  -1.779 0.075262 .
## glipizide                -0.0394206  0.0554962  -0.710 0.477499
## glyburide                -0.0770857  0.0612274  -1.259 0.208028
## pioglitazone             -0.1051970  0.0673396  -1.562 0.118244
## rosiglitazone            -0.0701111  0.0705504  -0.994 0.320333
## insulin                  -0.0369879  0.0535855  -0.690 0.490031
## change                    0.0647218  0.0474893   1.363 0.172923
## diabetesMed               0.2325196  0.0597013   3.895 9.83e-05 ***
## primarydiagclass         -0.0154751  0.0066070  -2.342 0.019170 *
## secondarydiagclass_1      0.0079339  0.0058183   1.364 0.172695
## tertiarydiagclass_1       0.0175397  0.0057634   3.043 0.002340 **
## DRGClassification         0.0044458  0.0036723   1.211 0.226032
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 22020  on 15883  degrees of freedom
## Residual deviance: 20923  on 15854  degrees of freedom
## AIC: 20983
##
## Number of Fisher Scoring iterations: 4
```

```r
predicted <- predict(logitMod, testData, type="response")
```

### Summary

The summary(logitMod) gives the beta coefficients, Standard error, z Value and p Value. As a next step of summary analysis we have to look for variables don't turn out to be significant in the model (i.e. p Value turns out greater than significance level of 0.05). The following values are considered to be significant in our model. age, admission_source_id, time_in_hospital, number_emergency, number_inpatient, number_diagnoses, max_glu_serum, A1Cresult, metformin, diabetesMed, primarydiagclass, tertiarydiagclass_1. The above variables becomes the next set of variables for our step wise regression.

### Optimal CutOff:

The default cutoff prediction probability score is 0.5 or the ratio of 1's and 0's in the training data. But sometimes, tuning the probability cutoff can improve the accuracy in both the training and test dataset. The optimal cutoff is used to improve the prediction of 1's, 0's, both 1's and 0's and to reduce the misclassification error. Below we will compute the optimal score that we use to minimize the misclassification error for the model.

```r
#Optimal Cut Off
optCutOff <- optimalCutoff(testData$readmitted, predicted)[1]
optCutOff
```

```
## [1] 0.998445
```

### MisClassification Error:

Misclassification error is the percentage mismatch of predcited vs actuals, irrespective of 1's or 0's. The lower the misclassification error, the better is our model.

```r
#Classfication Error.
misClassError(testData$readmitted, predicted, threshold = optCutOff)
```

```
## [1] 0.0405
```

### VIF

From our corrplot analysis we did not find much correlation between our predictor variables and also between predictor variable and target variable. Further as next step in our regression analysis, we want to confirm the same by validating the variance inflation factor. We should check for multicollinearity in the model. As seen below, all predictor variables in the model have VIF well below 4.

```r
#VIF Factor
knitr::kable(car::vif(logitMod))
```

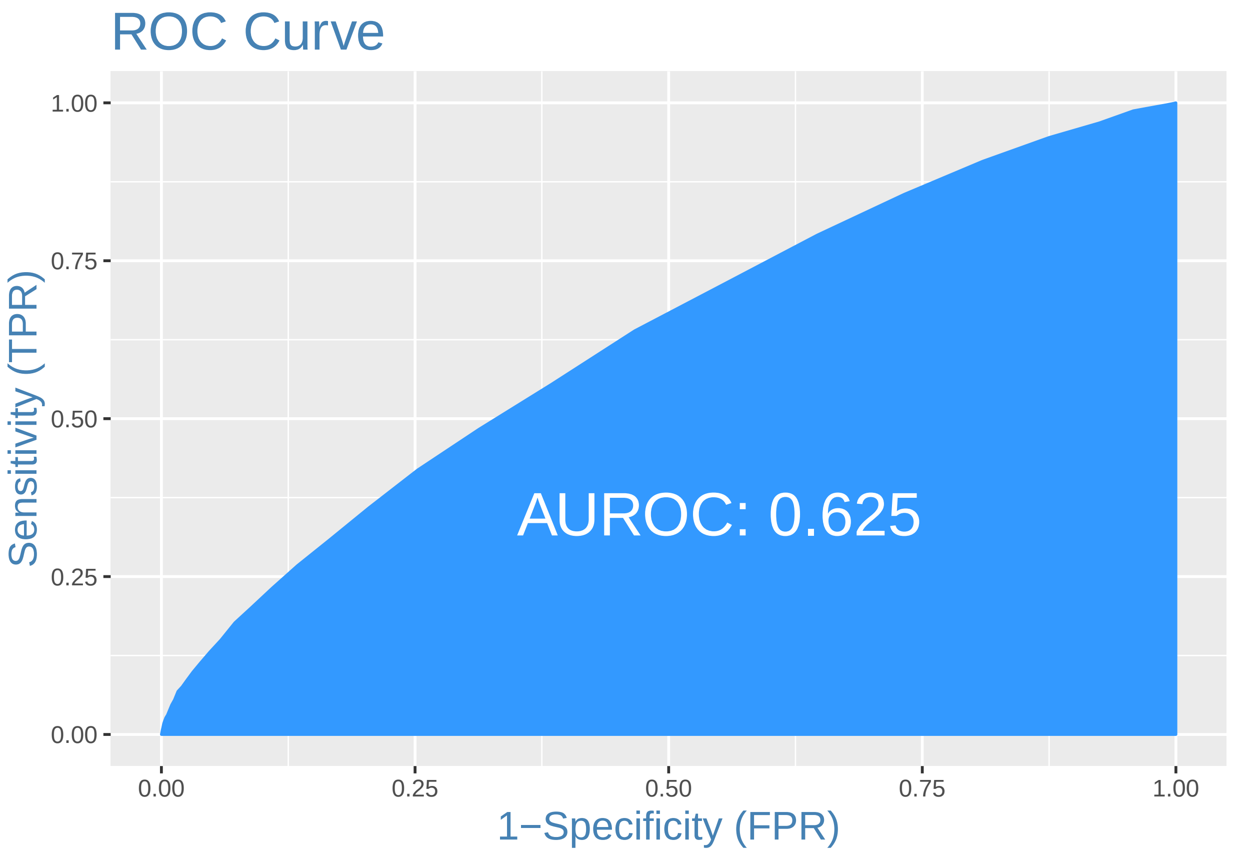
| | x |
|---|---:|
| race | 1.051333 |
| gender | 1.020962 |

| | x |
|---|---|
| age | 1.163690 |
| admission_type_id | 1.390994 |
| discharge_disposition_id | 1.037087 |
| admission_source_id | 1.421390 |
| time_in_hospital | 1.448778 |
| num_lab_procedures | 1.298425 |
| num_procedures | 1.290099 |
| num_medications | 1.746123 |
| number_outpatient | 1.035940 |
| number_emergency | 1.096536 |
| number_inpatient | 1.111531 |
| number_diagnoses | 1.223003 |
| max_glu_serum | 1.719808 |
| A1Cresult | 1.110510 |
| metformin | 1.304300 |
| glimepiride | 1.110022 |
| glipizide | 1.292438 |
| glyburide | 1.297341 |
| pioglitazone | 1.109716 |
| rosiglitazone | 1.100765 |
| insulin | 2.636983 |
| change | 2.087960 |
| diabetesMed | 2.226846 |
| primarydiagclass | 1.117049 |
| secondarydiagclass_1 | 1.073137 |
| tertiarydiagclass_1 | 1.035312 |
| DRGClassification | 1.115241 |

### ROC

Receiver Operating Characteristics Curve traces the percentage of true positives accurately predicted by a given logit model as the prediction probability cutoff is lowered from 1 to 0. For a good model, as the cutoff is lowered, it should mark more of actual 1's as positives and lesser of actual 0's as 1's. So for a good model, the curve should rise steeply, indicating that the TPR (Y-Axis) increases faster than the FPR (X-Axis) as the cutoff score decreases. Greater the area under the ROC curve, better the predictive ability of the model. We will not look at the curve for our model. From the below curve we can see our curve with AUROC value of 0.625. The value is decent value, though not good.

```r
#ROC Plot
plotROC(testData$readmitted, predicted)
```



```
#Confusion Matrix
predicted = ifelse(predicted > 0.5, 1, 0)
caret::confusionMatrix(factor(testData$readmitted), factor(predicted))
```

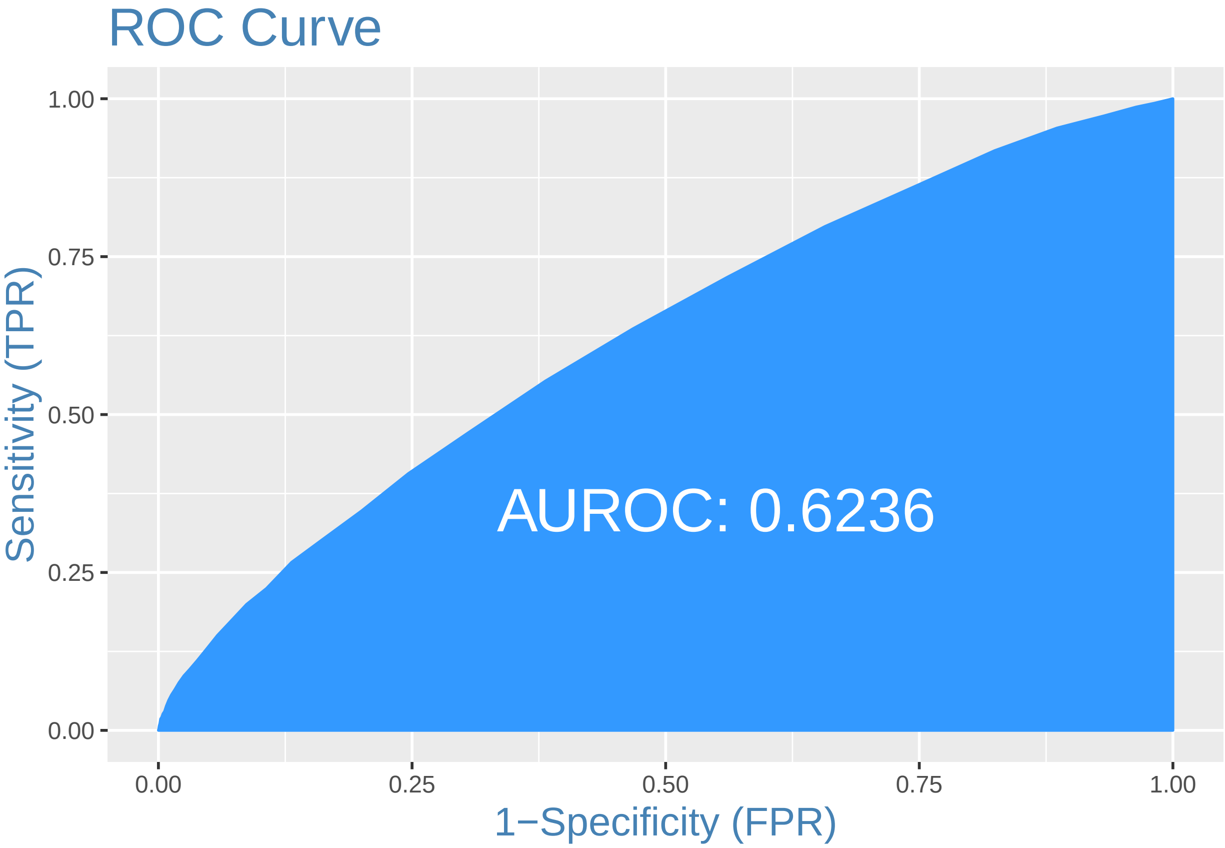
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 55415 25246
##          1  1760  1645
##
##                Accuracy : 0.6788
##                  95% CI : (0.6756, 0.6819)
##     No Information Rate : 0.6801
##     P-Value [Acc > NIR] : 0.8035
##
##                   Kappa : 0.0395
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.96922
##             Specificity : 0.06117
##          Pos Pred Value : 0.68701
##          Neg Pred Value : 0.48311
##              Prevalence : 0.68012
##          Detection Rate : 0.65918
##    Detection Prevalence : 0.95950
##       Balanced Accuracy : 0.51520
```

```
##
##        'Positive' Class : 0
##
```

From the confusion Matrix analysis, we come to the conclusion that our model Accuracy is 67.88%. We are able to predict with 67.88% accuracy that with DRG as a predictor variable the diabetic mellitus patient will get readmitted. When you look at the Sensitivity of our model it is pretty good. Sensitivity (or True Positive Rate) is the percentage of 1's (actuals) correctly predicted by the model. Which is what we are looking for in our readmission analysis.

### Binary Regression Base With Reduced Predictors:

From our first model we are going to drop those predictor variables which we find in statistically less signficant based on p-value(<0.05). With that analysis the list of variables which will be used for this model are. age, admission_source_id, time_in_hospital, number_emergency, number_inpatient, number_diagnoses, max_glu_serum, A1Cresult, metformin, diabetesMed, primarydiagclass, tertiarydiagclass_1



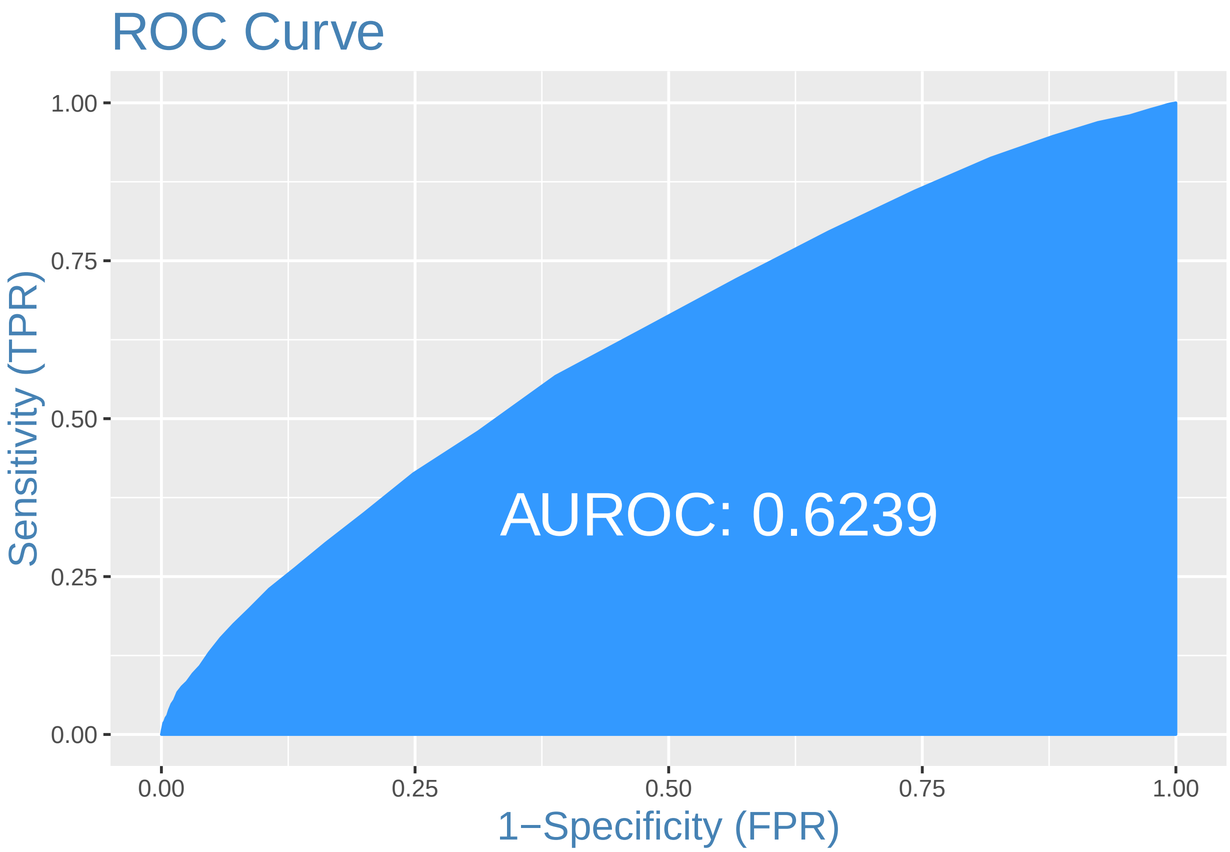
```
caret::confusionMatrix(factor(testData$readmitted), factor(predicted_2))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 55961 24700
##          1  1798  1607
##
##                Accuracy : 0.6848
##                  95% CI : (0.6816, 0.6879)
##     No Information Rate : 0.6871
##     P-Value [Acc > NIR] : 0.9228
##
##                   Kappa : 0.0393
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.96887
##             Specificity : 0.06109
##          Pos Pred Value : 0.69378
##          Neg Pred Value : 0.47195
##              Prevalence : 0.68707
##          Detection Rate : 0.66568
##    Detection Prevalence : 0.95950
##       Balanced Accuracy : 0.51498
##
##        'Positive' Class : 0
##
```

From our reduced predictor variable regression model, the Accuracy of our model has gone up, though not by a greater percent, but to some degree to a value of 68.48%. This shows that DRGClassification acts a negative parameter from logistic regression modelling prespective. We would like to see how the other logistic regression and ensemble models before drawing conclusions.

**Binary Regression Base With Log transformation:**

From the plots in our data preparation step, we found some of the variables are skewed either to the left or right. Out of those parameters, the parameters which are important to us as part of our Literature review are Age, Time_In_Hospital and DRGClassification. So in our base model we want to do a log transformation on these parameters and see whether the accuracy our model increases.



```
caret::confusionMatrix(factor(testData$readmitted), factor(predicted_3))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 55460 25201
##          1  1773  1632
##
##                Accuracy : 0.6791
##                  95% CI : (0.676, 0.6823)
##     No Information Rate : 0.6808
##     P-Value [Acc > NIR] : 0.8524
##
##                   Kappa : 0.0389
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.96902
##             Specificity : 0.06082
##          Pos Pred Value : 0.68757
##          Neg Pred Value : 0.47930
##              Prevalence : 0.68081
##          Detection Rate : 0.65972
##    Detection Prevalence : 0.95950
##       Balanced Accuracy : 0.51492
##
##        'Positive' Class : 0
```

##