Blood Donation Prediction

Cecil Rivers

Monday, January 26, 2015

Objective:

This project focuses on the development of a model that will predict blood donations. This project is apart of a competition held by DrivenData (www.drivendata.org).

The dataset provided for this competition is from a mobile blood donation vehicle in Taiwan and supplied by the UCI Machine Learning repository. The objective of this project is to predict whether or not a donor will give blood the next time the vehicle comes to campus.

The analysis and model generation was performed in the R scripting language along with the generation of this document.

Data:

There are two datasets provided in this competition: a training and test dataset. The training dataset has six variables:

- Unique ID: Number which uniquely identifies the person donating blood.
- Months since Last Donation: Number of months since this donor's most recent donation.
- Number of Donations: Total number of donations that the donor has made.
- Total Volume Donated: Total amount of blood that the donor has donated in cubic centimeters.
- Months since First Donation: Number of months since the donor's first donation.
- Made Donation in March 2007: Indication whether a person donated in March 2007.

The test dataset has five features. All the features in the test dataset are the same as the training dataset except "Made Donation in March 2007" is missing.

The feature names have been shortened in order to make them easier to visualize in the R script. Below are the feature name transformations:

- Months since Last Donation -> recency
- Number of Donations -> frequency
- Total Volume Donated -> volume
- Months since First Donation -> time
- Made Donation in March 2007 -> march

The unique ID has been removed from both datasets, since the prediction model should be able to predict if any donor provides a future donation based on their previous behavior, not just specific donors.

Analysis

Below is a summary of the training dataset:

```
##     recency         frequency          volume           time      
##  Min.   : 0.000   Min.   : 1.000   Min.   :  250   Min.   : 2.00  
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.:  500   1st Qu.:16.00  
##  Median : 7.000   Median : 4.000   Median : 1000   Median :28.00  
```

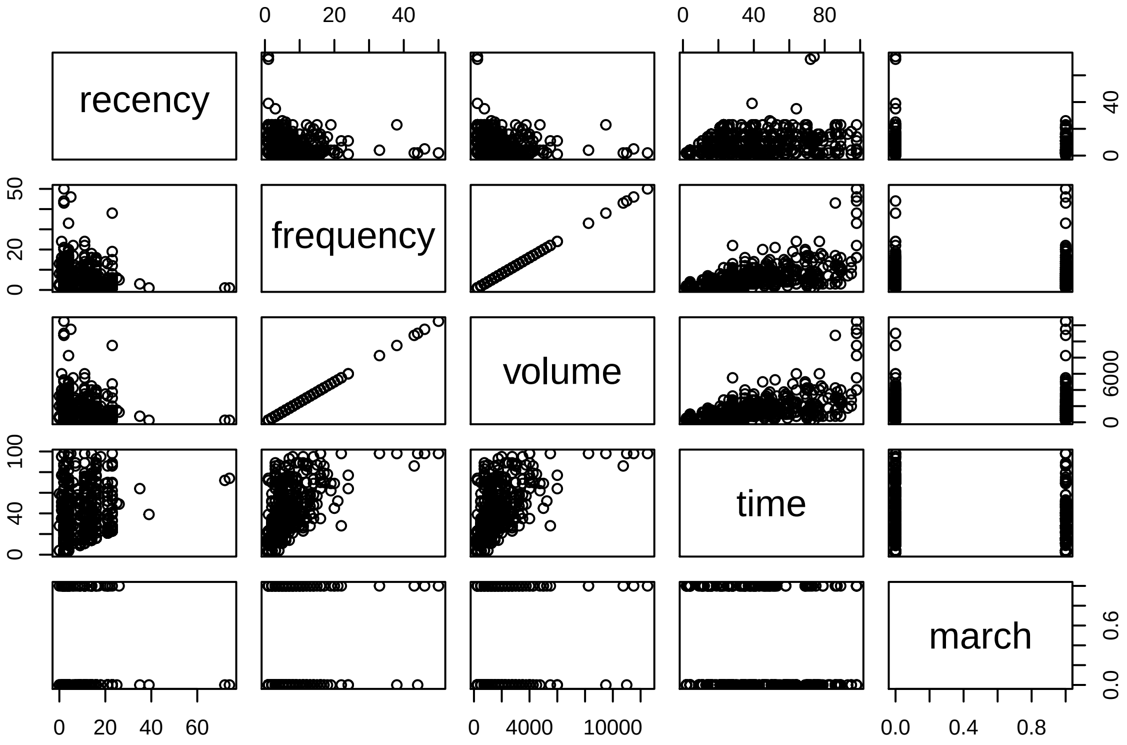

Figure 1: Figure 2. Scatter plot of features in the blood donation training dataset.

```
##  Mean   : 9.439   Mean   : 5.427   Mean   : 1357   Mean   :34.05
##  3rd Qu.:14.000   3rd Qu.: 7.000   3rd Qu.: 1750   3rd Qu.:49.25
##  Max.   :74.000   Max.   :50.000   Max.   :12500   Max.   :98.00
##      march
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2396
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

A scatter plot of the training dataset shows a correlation between the number of donations (frequency) and the total volume donated (volume) which seems reasonable since the more times a person donates the more blood will be collected in total.

The classification feature *march* (Made Donation in March 2007) shows there is a severe imbalance between between donations made and not made where the majority of the dataset weights heavily on donations not made.

Modeling

In order to determine the best model for the dataset, several models were scored using the AUC (area under the curve) based on the training dataset. A similar technique is described by John Mount and Nina Zumel

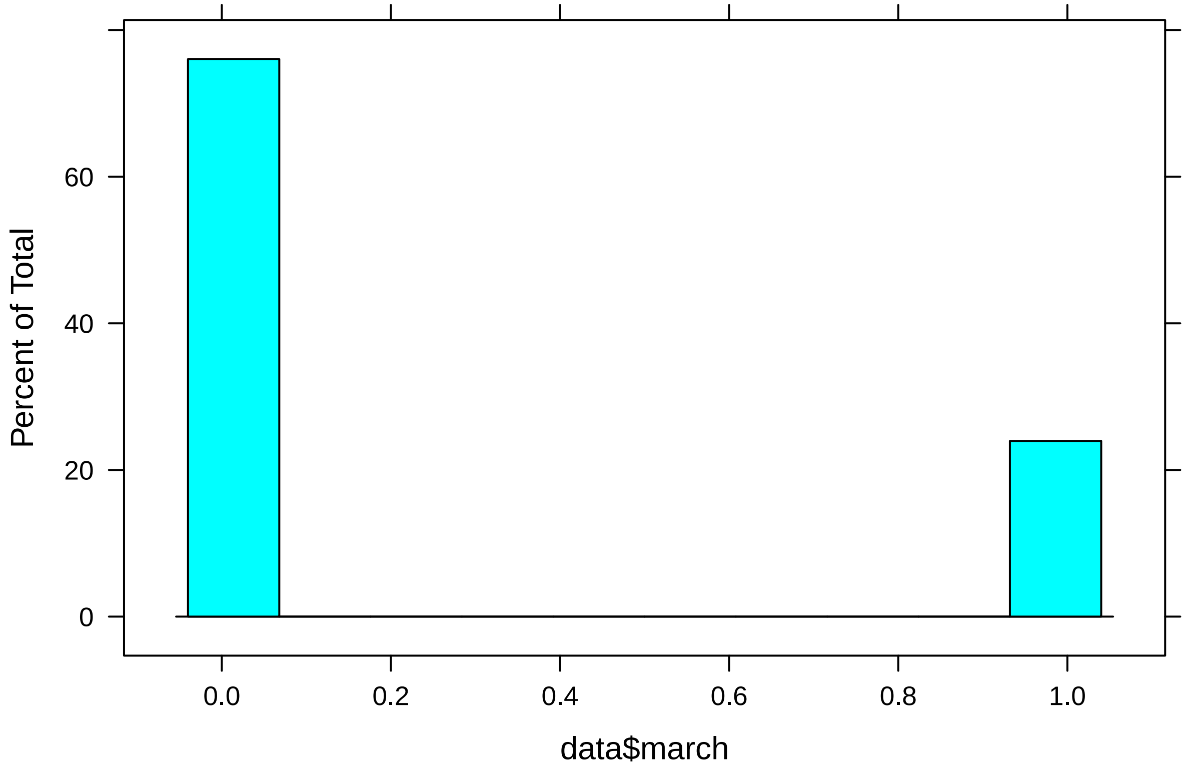

Figure 2: Figure 3. Histogram showing the distribution of made donations in March 2007.

in a paper on the Revolutions website called “How do you know if your model is going to work? Part 2: In-training set measures”. In this technique, all of the training data will be used to generate a model and then the model’s AUC will be calculated and compared to the AUC of other models generated using the same data. The model with the highest AUC score will be selected as the best prediction model.

The first models evaluated were simple logistic regression models like generalized linear model (glm). Initially all predictors in the training dataset were used to get a baseline.

```
## 
## Call:
## glm(formula = march ~ recency + frequency + time + volume, family = binomial(link = "logit"), 
##     data = data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.5102  -0.8079  -0.5273  -0.2427   2.5545  
## 
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -0.585643   0.201818  -2.902  0.00371 ** 
## recency     -0.091026   0.018955  -4.802 1.57e-06 ***
## frequency    0.129921   0.029102   4.464 8.03e-06 ***
## time        -0.018797   0.006588  -2.853  0.00433 ** 
## volume             NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 634.29  on 575  degrees of freedom
## Residual deviance: 556.61  on 572  degrees of freedom
## AIC: 564.61
## 
## Number of Fisher Scoring iterations: 5
```

The coefficients in the baseline glm model using all of the predictors shows that with all predictors present in the model, all predictors are significant (p-value < 0.05) except for the “Total Volume Donated” (volume) which was not defined in the model because of singularities. These singularities point toward the correlation shown in the scatter plot between the volume and frequency variables. To prevent the singularity, the log of the total volume donated was taken and a new glm was created.

```r
summary(glm.fit2)
```

```
## 
## Call:
## glm(formula = march ~ recency + frequency + time + log(volume), 
##     family = binomial(link = "logit"), data = data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.5391  -0.7241  -0.4895  -0.1952   2.7119  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -7.07215    1.57647  -4.486 7.25e-06 ***
## recency     -0.07623    0.01971  -3.867  0.00011 ***
## frequency    0.02855    0.03204   0.891  0.37283
## time        -0.03181    0.00752  -4.230 2.34e-05 ***
## log(volume)  1.06439    0.25330   4.202 2.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 634.29  on 575  degrees of freedom
## Residual deviance: 538.88  on 571  degrees of freedom
## AIC: 548.88
##
## Number of Fisher Scoring iterations: 5
```

To compare both models a receiver operating characteristic (ROC) curve is utilized in order to determine the performance of each binary classifier.

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit1),     print.auc = TRUE)
##
## Data: fitted(glm.fit1) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.749
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "blue", print.au
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

The ROC plot shows the GLM1 (all predictors) and GLM2 (all predictors where the log of the total volume of donations is considered). The AUC (area of the curve) is higher for GLM2 indicating it is a better model than GLM1.

Several variations of the generalized linear models were tested. These models consist of the following: all original features plus the log(volume) (GLM3), the original GLM2 feature with the addition of a 1 (GLM4), and the GLM4 features without the frequency features (GLM5).

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit1),     print.auc = TRUE)
##
## Data: fitted(glm.fit1) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.749
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "blue", print.au
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

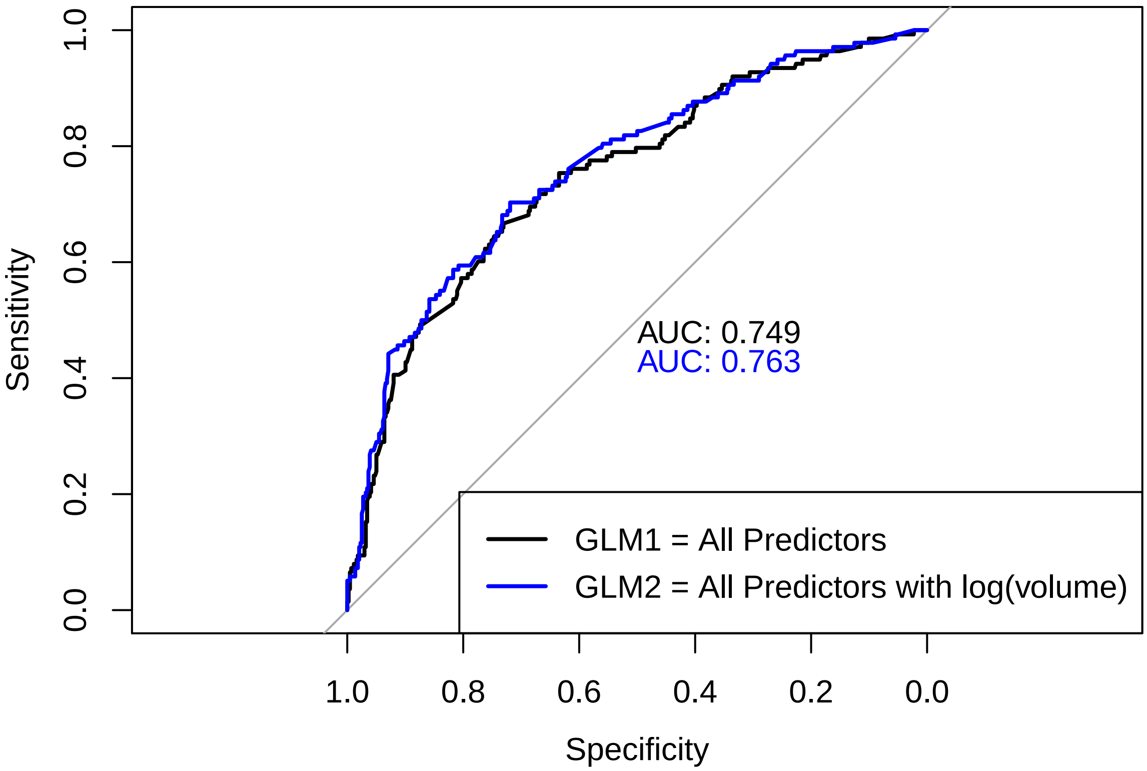

Figure 3: Figure 4. Receiver Operating Characteristic Curve for all predictors vs all predictors where the log of the total volume of donations.

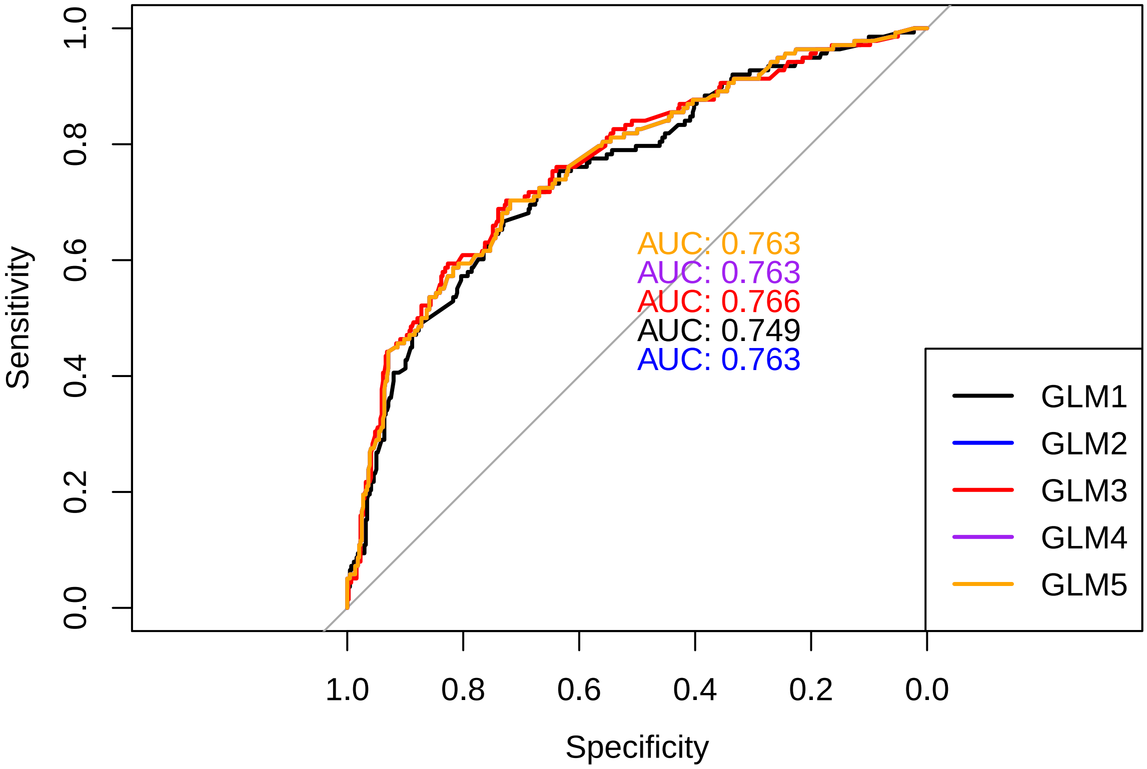

Figure 4: Figure 5. Comparison of five Generalized Linear Models.

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit3),     add = TRUE, col = "red", print.au
##
## Data: fitted(glm.fit3) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7661
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "purple", print
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "orange", print
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

The highest AUC was from GLM3, but that model had 3 singularities. After removing the singularities and the predictors with p-values > 0.05, the model reduced to the predictors in GLM2 and GLM4.

```
summary(glm.fit3)
```

```
##
## Call:
## glm(formula = march ~ . * . + log(volume), family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9264  -0.6998  -0.4761  -0.2610   2.5603
##
## Coefficients: (3 not defined because of singularities)
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -8.283e+00  2.868e+00  -2.888  0.00387 **
## recency           -5.517e-02  3.762e-02  -1.467  0.14243
## frequency          5.420e-02  1.340e-01   0.405  0.68579
## volume                    NA         NA      NA       NA
## time              -4.348e-02  1.611e-02  -2.699  0.00696 **
## log(volume)        1.255e+00  5.055e-01   2.482  0.01307 *
## recency:frequency -1.063e-02  5.128e-03  -2.074  0.03811 *
## recency:volume            NA         NA      NA       NA
## recency:time       1.066e-03  8.973e-04   1.187  0.23505
## frequency:volume  -2.956e-06  1.046e-05  -0.283  0.77754
## frequency:time     5.654e-04  1.554e-03   0.364  0.71591
## volume:time               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 634.29  on 575  degrees of freedom
## Residual deviance: 534.36  on 567  degrees of freedom
## AIC: 552.36
##
## Number of Fisher Scoring iterations: 5
```

Using the features of recency, frequency, time and log(volume) the generalized linear model was expanded to an generalized additive model (GAM). By varying the dimension of the smooth term in the GAM, the smoothing term that generates the best AUC was discovered.

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.12
```

Taking the maximum AUC from Figure 6 yields a smooth term dimension of 56. Figure 7 compares the previous GLM models to the GAM model using the smooth term dimension of 56.

```
gam.fit <- gam(march~s(recency,i) + s(frequency,i) + s(time,i) + s(log(volume),i),data = data,family=bi

plot.roc(data$march,fitted(glm.fit1),print.auc=TRUE)
```

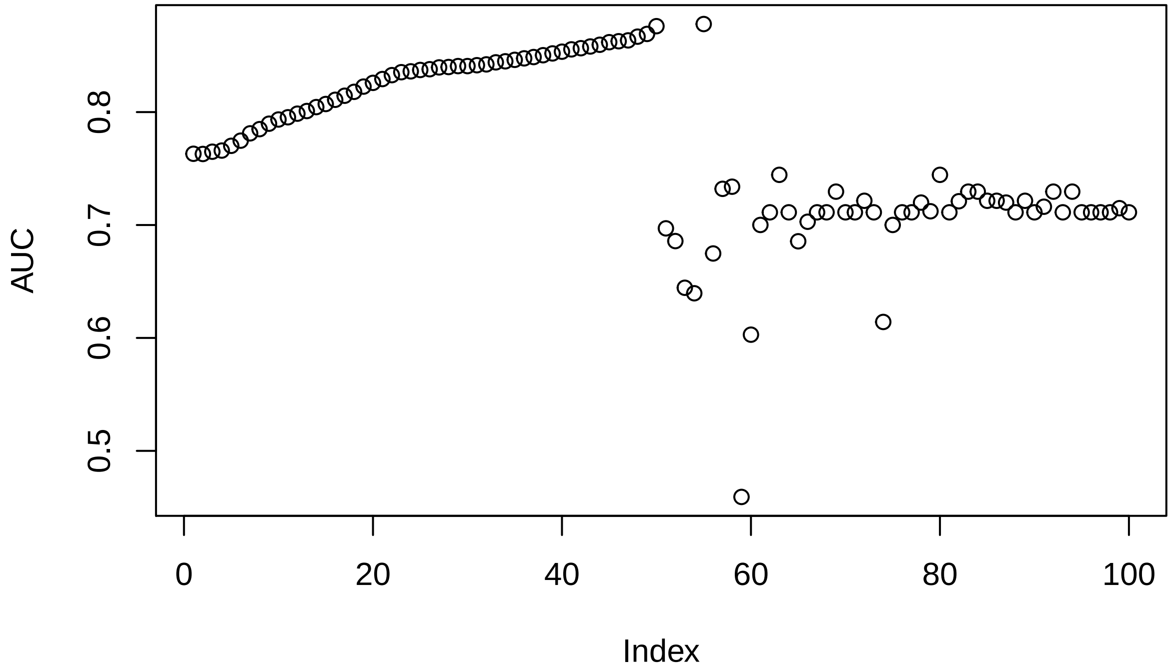

Figure 5: Figure 6. AUC vs Smoothing term Dimension for GAM

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit1),     print.auc = TRUE)
##
## Data: fitted(glm.fit1) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.749
```

```
plot.roc(data$march,fitted(glm.fit2),add=TRUE,col = "blue",
         print.auc=TRUE,print.auc.y = 0.45)
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "blue", print.au
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

```
plot.roc(data$march,fitted(glm.fit3),add=TRUE,col = "red",
         print.auc=TRUE,print.auc.y = 0.55)
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit3),     add = TRUE, col = "red", print.auc
##
## Data: fitted(glm.fit3) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7661
```

```
plot.roc(data$march,fitted(glm.fit2),add=TRUE,col = "purple",
         print.auc=TRUE,print.auc.y = 0.60)
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "purple", print
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

```
plot.roc(data$march,fitted(glm.fit2),add=TRUE,col = "orange",
         print.auc=TRUE,print.auc.y = 0.65)
```

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(glm.fit2),     add = TRUE, col = "orange", print
##
## Data: fitted(glm.fit2) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.7632
```

```
plot.roc(data$march,fitted(gam.fit),add=TRUE,col = "green",
         print.auc=TRUE,print.auc.y = 0.70)
```

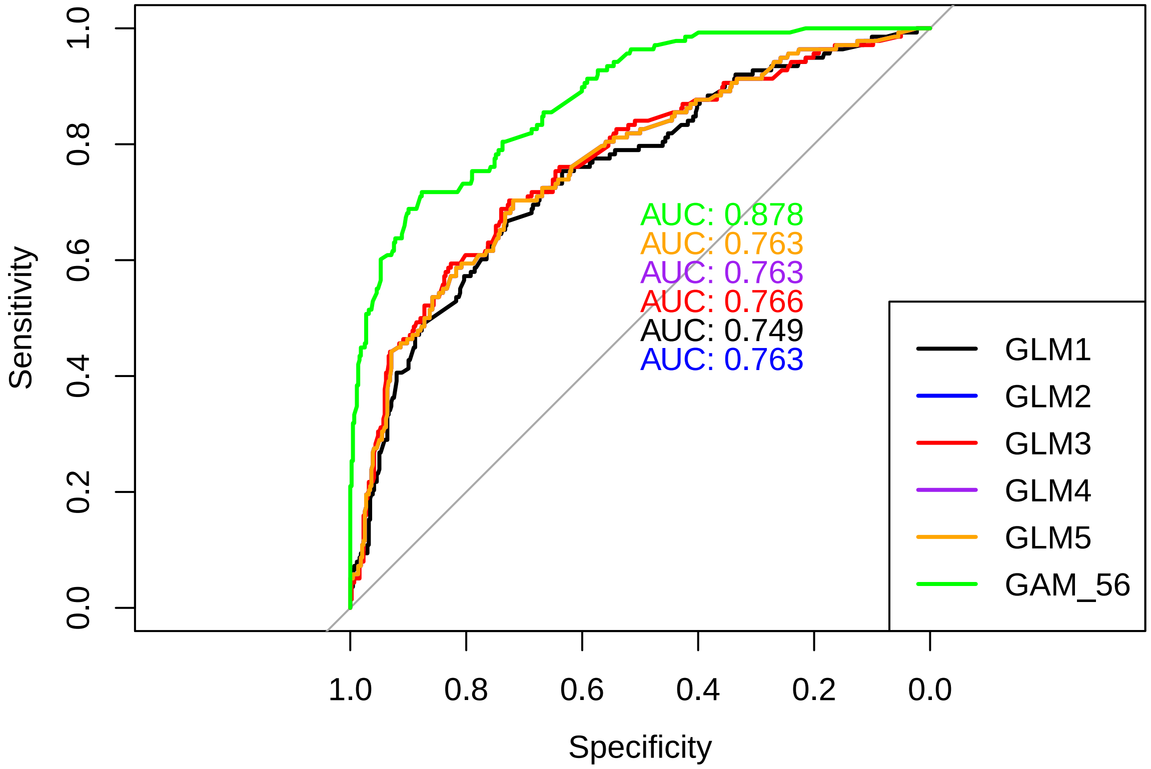

Figure 6: Figure 7. Comparison of GLM vs GAM

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(gam.fit),     add = TRUE, col = "green", print.au
##
## Data: fitted(gam.fit) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.878
```

```
legend("bottomright", c("GLM1", "GLM2","GLM3","GLM4","GLM5","GAM_56"),
       col=c("black", "blue","red","purple","orange","green"), lwd=c(2,2))
```

The next model investigated was a neural network because of their ability to address complex datasets. To build the neural network, the R package nnet was utilized. This package can generate single hidden layer neural networks for classification or regression models where regression models were selected for the neural network. Before the neural network was generated, the training data was normalized using R's *scale* function, then the number of units in the hidden layer was selected by running at various units in the hidden layer and using the AUC to determine which neural network provided the best performance. The maximum number of units in the hidden layer was limited to 200 due to the amount of processing time required to generate the model.

```
##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(model_nnet_10),     print.auc = TRUE)
##
```

```
## Data: fitted(model_nnet_10) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.813

##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(model_nnet_50),     add = TRUE, col = "blue", pr
##
## Data: fitted(model_nnet_50) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.951

##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(model_nnet_100),     add = TRUE, col = "red", pr
##
## Data: fitted(model_nnet_100) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.982

##
## Call:
## plot.roc.default(x = data$march, predictor = fitted(model_nnet_200),     add = TRUE, col = "purple",
##
## Data: fitted(model_nnet_200) in 438 controls (data$march 0) < 138 cases (data$march 1).
## Area under the curve: 0.9831
```

The neural network with 200 units in the hidden layer produced the highest AUC. This AUC was also higher than the GLM and GAM models. As a result of the higher AUC, the neural network with 200 hidden units was trained with the complete training dataset and the predicted results from the test dataset were submitted to the DrivenData competition.

Conclusion

The blood donation competition provided several opportunities for modeling such as: small dataset, imbalanced outcome and collinear features. By utilizing an AUC scoring to compare various models allowed for easy selection of the model; however, determine the validity of the AUC score needs further investigation. Another area of investigation is cross validation. By using the method of comparing AUC of models generated from the complete training dataset, the approximation of the model's performance would not be as accurate as using cross validation on portions of the training dataset. The third area of further investigation are alternative models that could provided better predictions such as SVM or random forest.

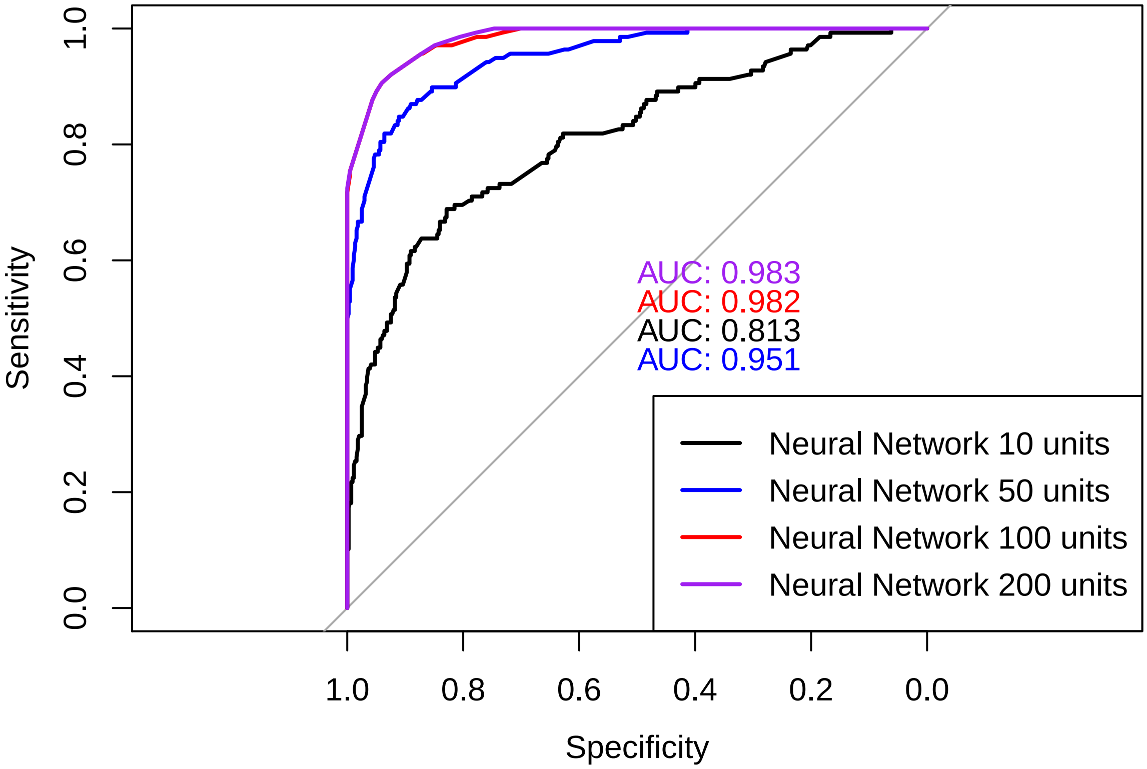

Figure 7: Figure 8. Comparison of neural networks by varying number of units in hidden layer