P8106 Midterm Project

Lin Yang

Data cleaning

```r
stroke_dat <- read.csv("healthcare-dataset-stroke-data.csv") %>%
  janitor::clean_names() %>%
  dplyr::select(-1) %>% #delete the id column
  filter(bmi != "N/A") %>% #remove missing bmi values
  filter(gender != "Other") %>%
  mutate(bmi = as.numeric(bmi),
         gender = as.numeric(factor(gender)) - 1,
         ever_married = as.numeric(factor(ever_married)) - 1,
         work_type = as.numeric(factor(work_type)) - 1,
         residence_type = as.numeric(factor(residence_type)) - 1,
         smoking_status = as.numeric(factor(smoking_status)) - 1,
         stroke = factor(stroke,
                         levels = c("0", "1"),
                         labels = c("neg", "pos")))

set.seed(1)
trainRows <- createDataPartition(y = stroke_dat$stroke, p = 0.8, list = FALSE)
stroke_train <- stroke_dat[trainRows, ]
stroke_test <- stroke_dat[-trainRows, ]

x_train <- stroke_train[ , -11]
y_train <- stroke_train$stroke

x_test <- stroke_test[ , -11]
y_test <- stroke_test$stroke
```

EDA

```r
#prevalence of stroke
stroke_dat %>%
  group_by(stroke) %>%
  summarize(count = n()) %>%
  mutate(proportion = round(count / sum(count), 3)) %>%
  knitr::kable()
```

stroke	count	proportion
neg	4699	0.957

stroke	count	proportion
pos	209	0.043

```
#distribution of gender
stroke_dat %>%
  group_by(gender) %>%
  summarize(count = n()) %>%
  mutate(gender = recode(gender, `0` = "female", `1` = "male")) %>%
  knitr::kable()
```

gender	count
female	2897
male	2011

```
#prevalence of hypertension
stroke_dat %>%
  group_by(hypertension) %>%
  summarize(count = n()) %>%
  mutate(hypertension = recode(hypertension, `0` = "no", `1` = "yes")) %>%
  knitr::kable()
```

hypertension	count
no	4457
yes	451

```
#prevalence of heart disease
stroke_dat %>%
  group_by(heart_disease) %>%
  summarize(count = n()) %>%
  mutate(heart_disease = recode(heart_disease, `0` = "no", `1` = "yes")) %>%
  knitr::kable()
```

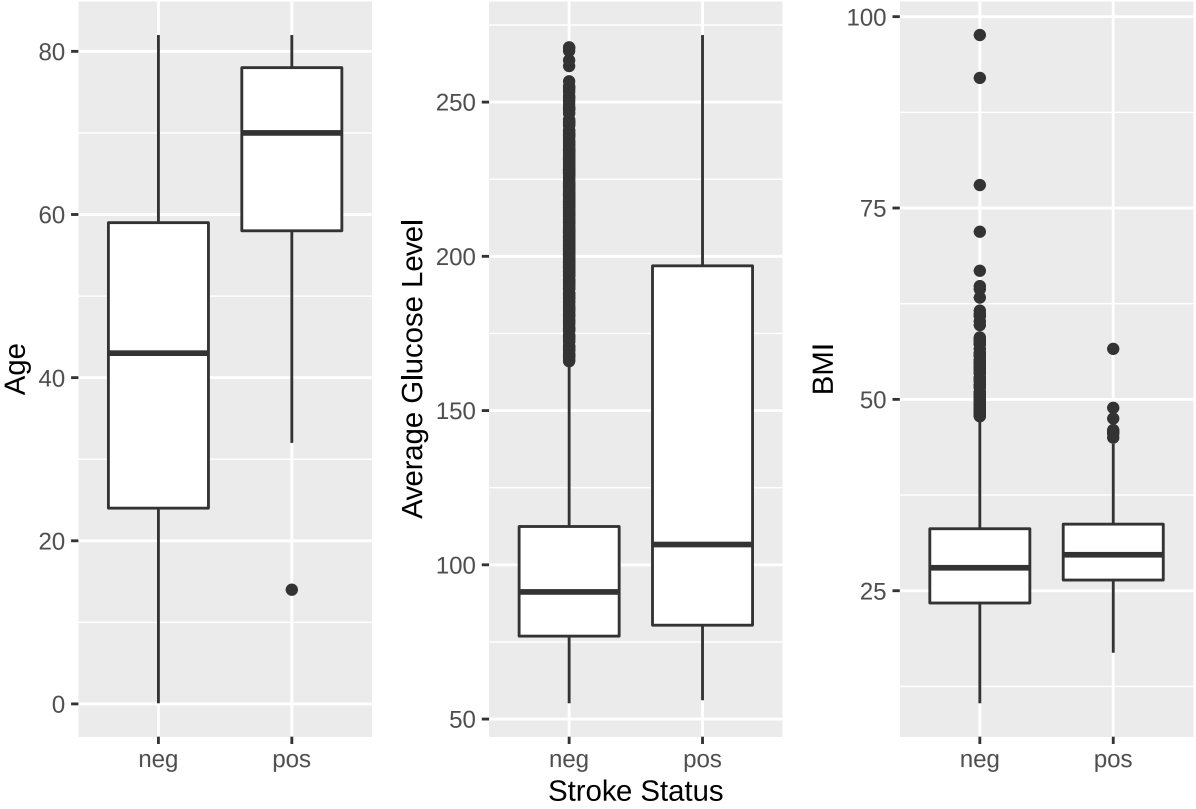
heart_disease	count
no	4665
yes	243

```
stroke_dat %>%
  group_by(ever_married) %>%
  summarize(count = n()) %>%
  mutate(ever_married = recode(ever_married, `0` = "no", `1` = "yes")) %>%
  knitr::kable()
```

ever_married	count
no	1704

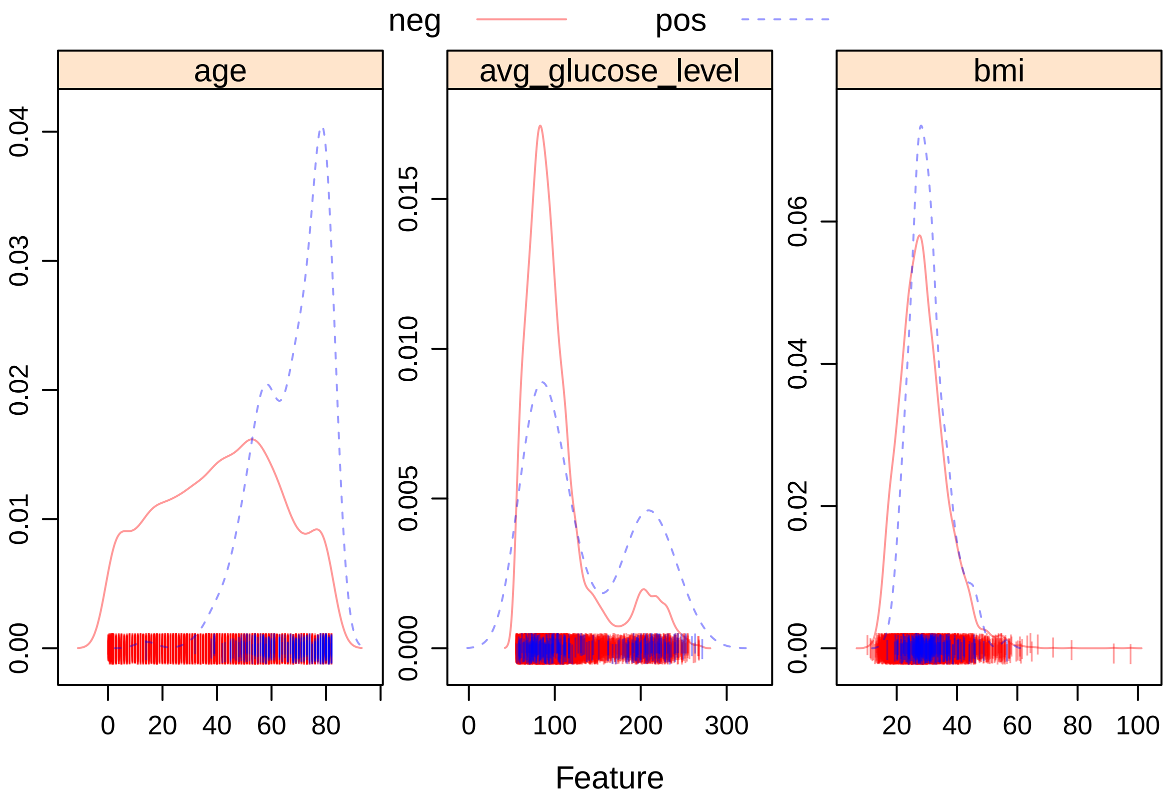
ever_married	count
yes	3204

```
#boxplots of continuous variables
age <- ggplot(data = stroke_dat, aes(x = stroke, y = age), group = stroke) +
  geom_boxplot() +
  labs(x = " ",
       y = "Age")

glucose <- ggplot(data = stroke_dat, aes(x = stroke, y = avg_glucose_level), group = stroke) +
  geom_boxplot() +
  labs(x = "Stroke Status",
       y = "Average Glucose Level")

bmi <- ggplot(data = stroke_dat, aes(x = stroke, y = bmi), group = stroke) +
  geom_boxplot() +
  labs(x = " ",
       y = "BMI")

age + glucose + bmi
```

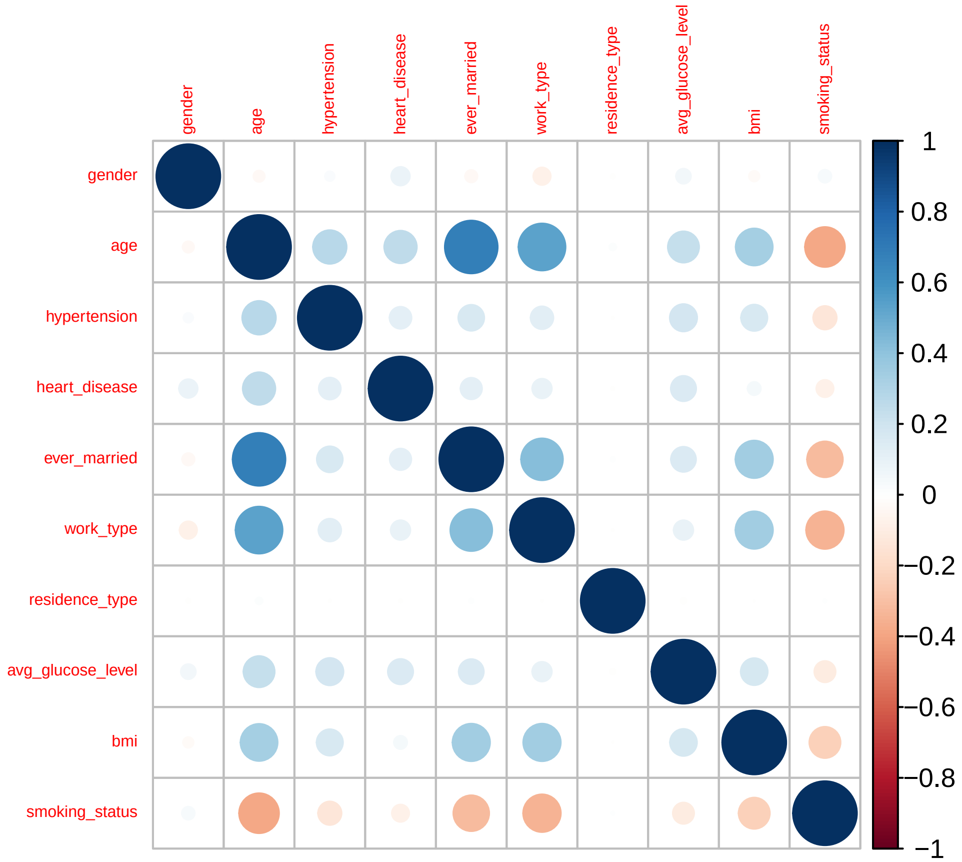


```
#density plots of stroke vs continuous variables
stroke_dat_con <- stroke_dat %>% dplyr::select(age, avg_glucose_level, bmi)
theme1 <- transparentTheme(trans = .4)
```

```r
trellis.par.set(theme1)

featurePlot(x = stroke_dat_con,
            y = stroke_dat$stroke,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch = "|",
            auto.key = list(columns = 2))
```



```r
#correlation plot of predictors
corrplot::corrplot(cor(stroke_dat[1:10]),
         method = "circle",
         type = "full",
         tl.cex = 0.5)
```



Fitting models

Logistic regression

```
#glm
fit.glm <- glm(stroke ~ .,
               data = stroke_train,
               family = binomial(link = "logit"))
summary(fit.glm)
```

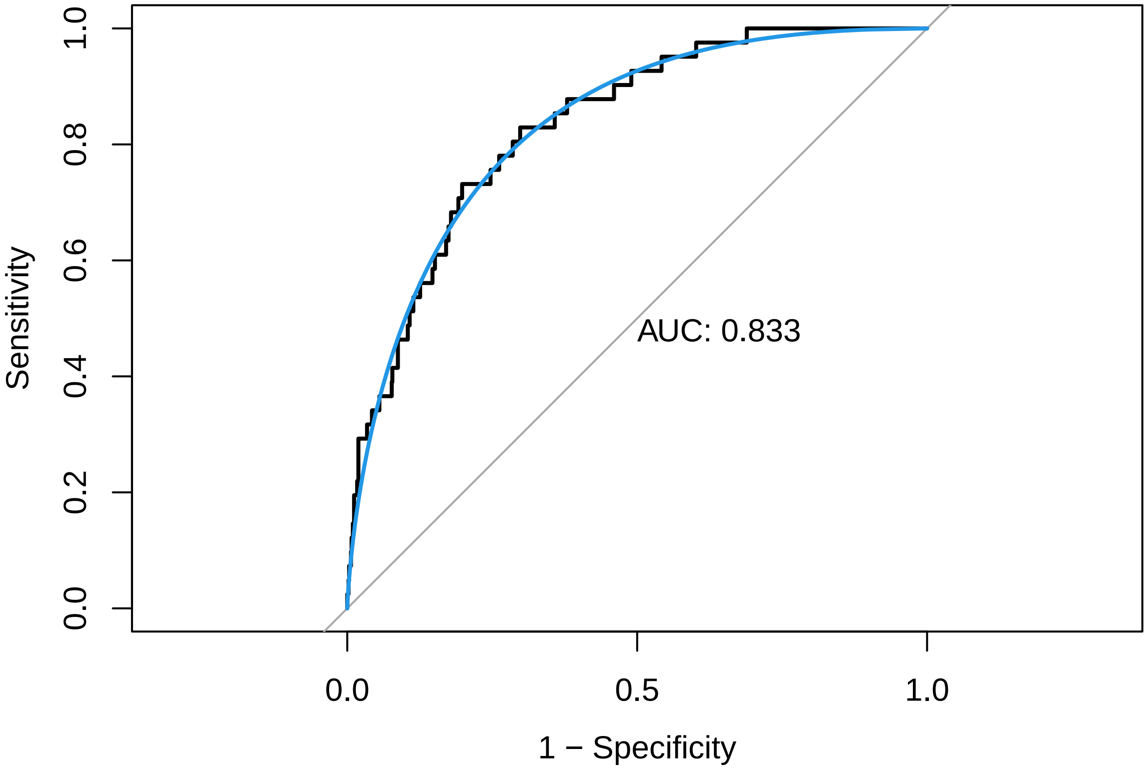
```
##
## Call:
## glm(formula = stroke ~ ., family = binomial(link = "logit"),
##     data = stroke_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1476  -0.2948  -0.1546  -0.0739   3.5252
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -7.639251   0.676093 -11.299  < 2e-16 ***
## gender              0.012272   0.170549   0.072  0.94264
## age                 0.073277   0.006693  10.949  < 2e-16 ***
## hypertension        0.482939   0.196059   2.463  0.01377 *
## heart_disease       0.317726   0.231726   1.371  0.17033
## ever_married       -0.155512   0.268260  -0.580  0.56211
## work_type          -0.095933   0.086350  -1.111  0.26658
## residence_type     -0.089320   0.166800  -0.535  0.59231
## avg_glucose_level   0.004323   0.001445   2.992  0.00277 **
## bmi                 0.005275   0.013167   0.401  0.68868
## smoking_status     -0.003868   0.081829  -0.047  0.96230
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1387.8  on 3927  degrees of freedom
## Residual deviance: 1098.4  on 3917  degrees of freedom
## AIC: 1120.4
##
## Number of Fisher Scoring iterations: 8
```

```
glm.pred.prob <- predict(fit.glm,
                     newdata = stroke_test,
                     type = "response")
glm.pred <- rep("neg", length(glm.pred.prob))
glm.pred[glm.pred.prob > 0.5] <- "pos"

confusionMatrix(data = factor(glm.pred),
                reference = stroke_test$stroke,
                positive = "pos")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg 939  41
##        pos   0   0
##
##                Accuracy : 0.9582
##                  95% CI : (0.9437, 0.9698)
##     No Information Rate : 0.9582
##     P-Value [Acc > NIR] : 0.5414
##
##                   Kappa : 0
##
##  Mcnemar’s Test P-Value : 4.185e-10
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.95816
##              Prevalence : 0.04184
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
```

```
##
##        'Positive' Class : pos
##
```

```
#ROC curve
roc.glm <- roc(stroke_test$stroke, glm.pred.prob)
plot(roc.glm, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glm), col = 4, add = TRUE)
```



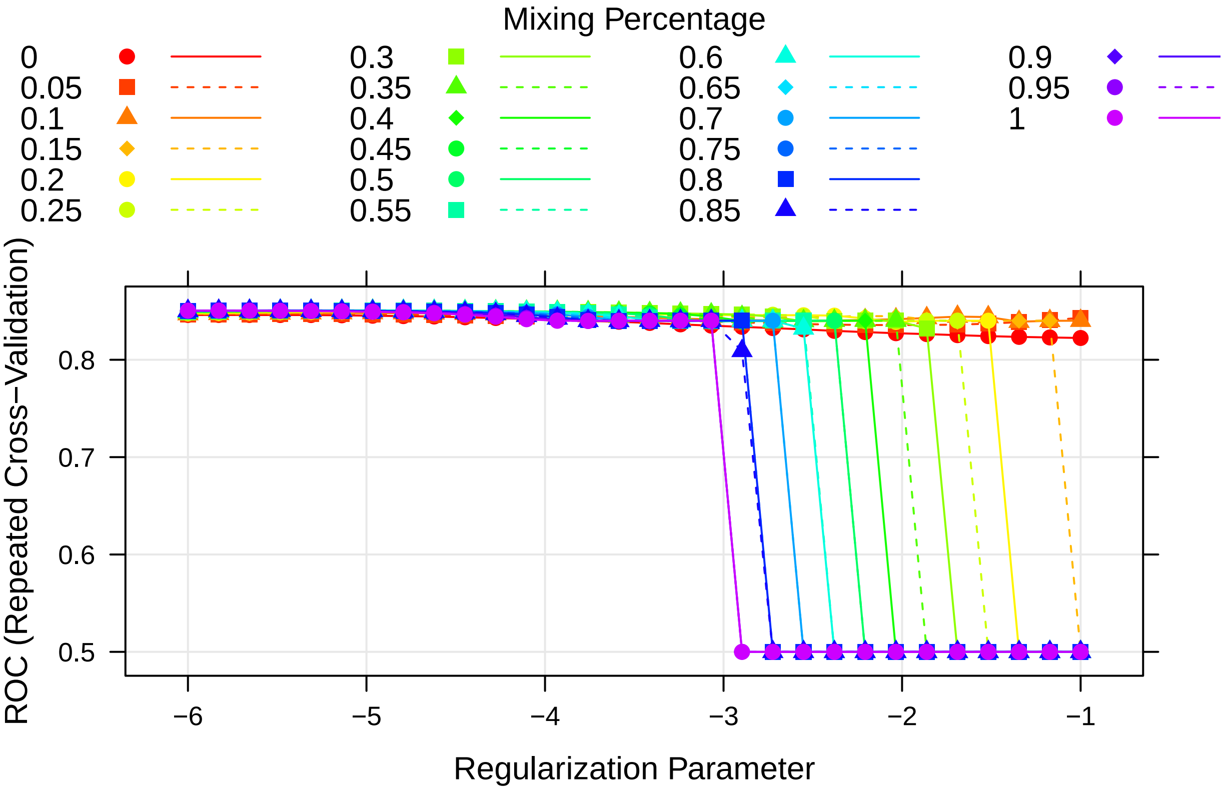
```
#fit a logistic regression model using caret for CV
ctrl <- trainControl(method = "repeatedcv", repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
set.seed(1)
model.glm <- train(x = stroke_train[ , 1:10],
                   y = stroke_train$stroke,
                   method = "glm",
                   metric = "ROC",
                   trControl = ctrl)
```

Penalized logistic regression

```
glmnGrid <- expand.grid(.alpha = seq(0, 1, length = 21),
                        .lambda = exp(seq(-6, -1, length = 30)))
set.seed(1)
model.glmn <- train(x = stroke_train[ , 1:10],
                    y = stroke_train$stroke,
                    method = "glmnet",
                    tuneGrid = glmnGrid,
                    metric = "ROC",
                    trControl = ctrl)

model.glmn$bestTune
```

```
##     alpha      lambda
## 277  0.45 0.006974342
```

```
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
              superpose.line = list(col = myCol))
plot(model.glmn, par.settings = myPar, xTrans = function(x) log(x))
```

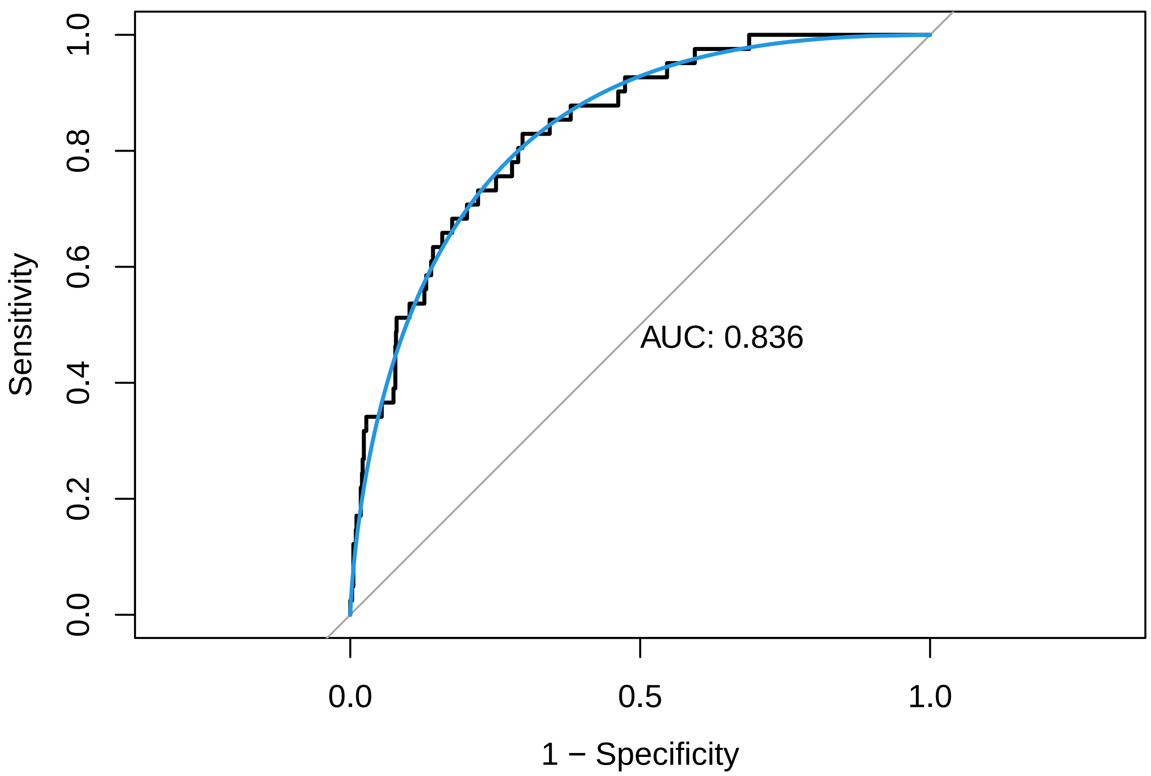


```
glmn.pred.prob <- predict(model.glmn,
                          newdata = stroke_test,
                          type = "prob")
glmn.pred <- ifelse(glmn.pred.prob$pos > 0.5, "pos", "neg")
```

```r
confusionMatrix(data = as.factor(glmn.pred),
                reference = stroke_test$stroke,
                positive = "pos")
```

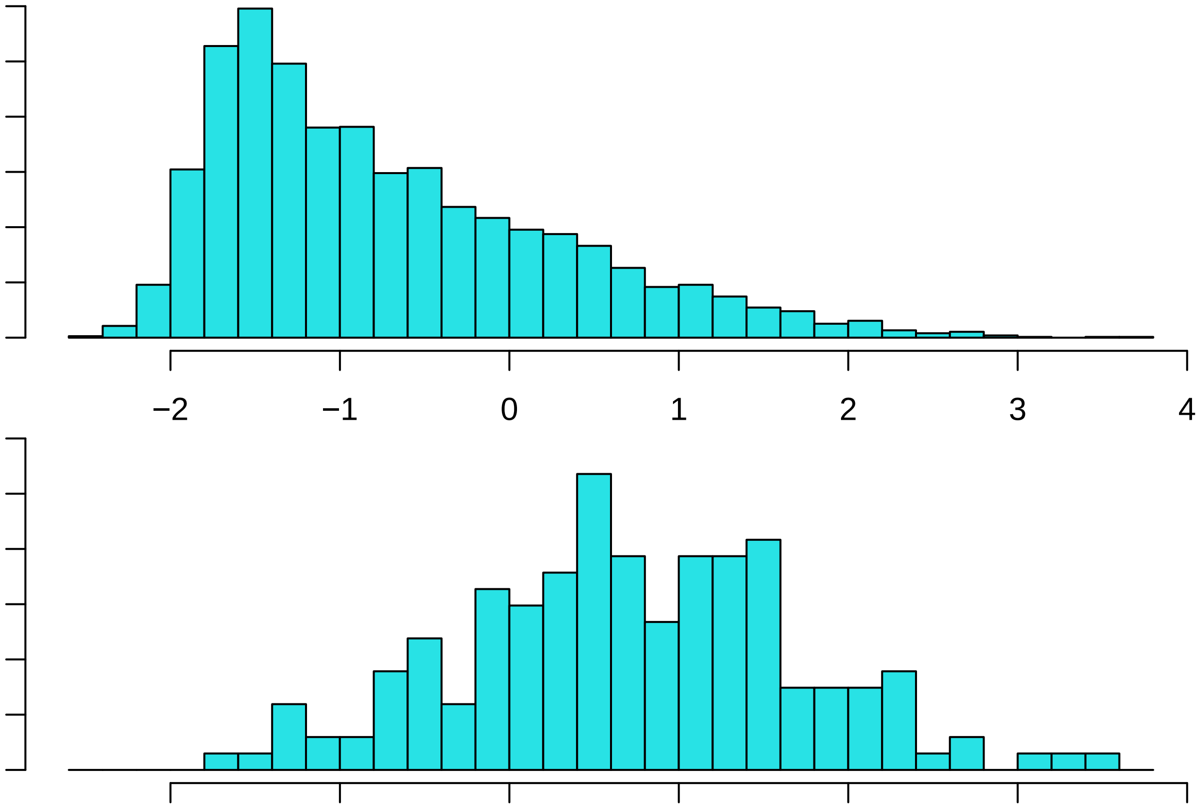
```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction neg pos
##        neg 939  41
##        pos   0   0
##                                           
##                Accuracy : 0.9582          
##                  95% CI : (0.9437, 0.9698)
##     No Information Rate : 0.9582          
##     P-Value [Acc > NIR] : 0.5414          
##                                           
##                   Kappa : 0               
##                                           
##  Mcnemar's Test P-Value : 4.185e-10       
##                                           
##             Sensitivity : 0.00000         
##             Specificity : 1.00000         
##          Pos Pred Value :     NaN         
##          Neg Pred Value : 0.95816         
##              Prevalence : 0.04184         
##          Detection Rate : 0.00000         
##    Detection Prevalence : 0.00000         
##       Balanced Accuracy : 0.50000         
##                                           
##        'Positive' Class : pos             
## 
```

```r
#ROC curve
roc.glmn <- roc(stroke_test$stroke, glmn.pred.prob[ , 2])
plot(roc.glmn, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.glmn), col = 4, add = TRUE)
```



LDA

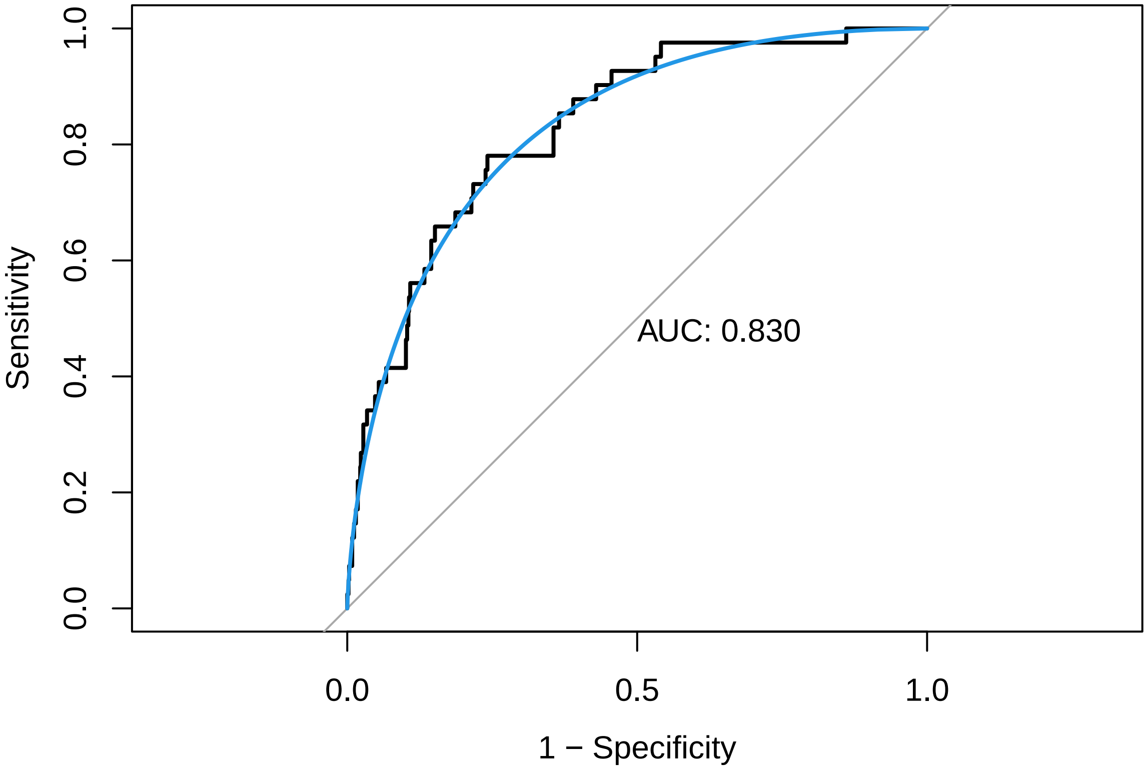
```
par(mar = c(1,1,1,1))
fit.lda <- lda(stroke ~ ., data = stroke_train)
plot(fit.lda)
```



```
lda.pred.prob <- predict(fit.lda, newdata = stroke_test, type = "prob")
posterior <- as.data.frame(lda.pred.prob$posterior)
lda.pred <- ifelse(posterior$pos > 0.5, "pos", "neg")

confusionMatrix(data = as.factor(lda.pred),
                reference = stroke_test$stroke,
                positive = "pos")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg 932  38
##        pos   7   3
##
##                Accuracy : 0.9541
##                  95% CI : (0.939, 0.9663)
##     No Information Rate : 0.9582
##     P-Value [Acc > NIR] : 0.7673
##
##                   Kappa : 0.1029
##
##  Mcnemar's Test P-Value : 7.744e-06
##
##             Sensitivity : 0.073171
##             Specificity : 0.992545
##          Pos Pred Value : 0.300000
##          Neg Pred Value : 0.960825
##              Prevalence : 0.041837
##          Detection Rate : 0.003061
##    Detection Prevalence : 0.010204
##       Balanced Accuracy : 0.532858
##
##        'Positive' Class : pos
##
```

```
#ROC curve
roc.lda <- roc(stroke_test$stroke, posterior[, 2])

plot(roc.lda, legacy.axes = TRUE, print.auc = TRUE)
plot(smooth(roc.lda), col = 4, add = TRUE)
```



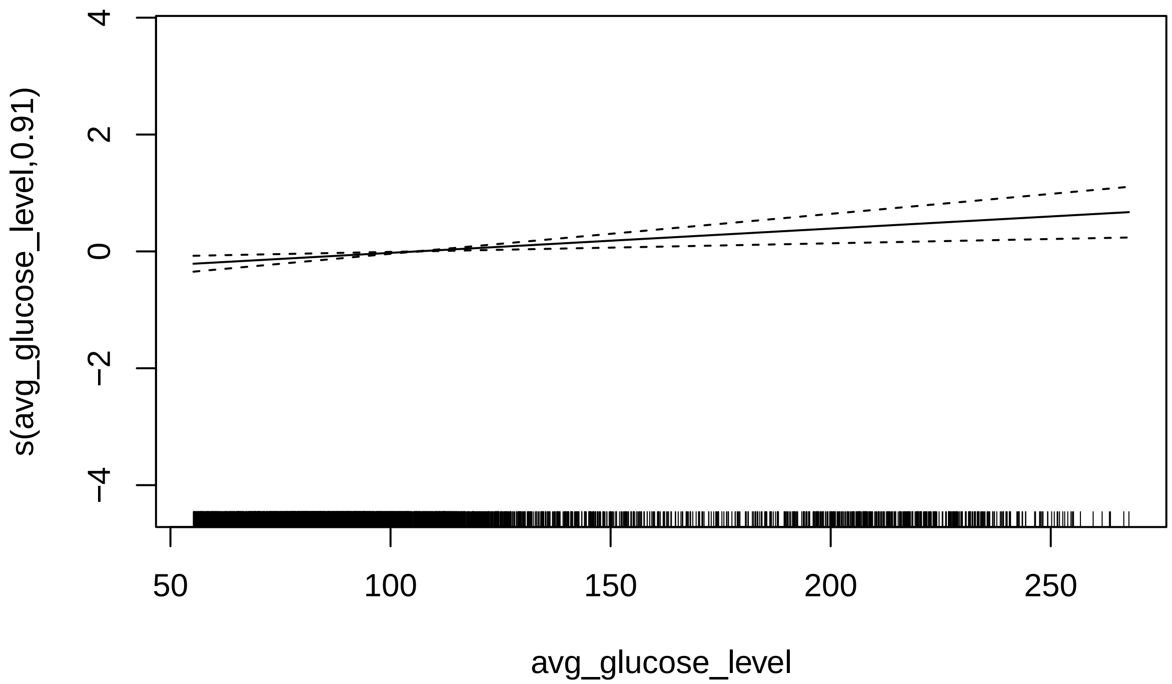
```
#use caret
set.seed(1)
model.lda <- train(x = stroke_dat[ , 1:10],
                   y = stroke_dat$stroke,
                   method = "lda",
                   metric = "ROC",
                   trControl = ctrl)
```

GAM

```
set.seed(1)
model.gam <- train(x = stroke_train[ , 1:10],
                   y = stroke_train$stroke,
                   method = "gam",
                   metric = "ROC",
                   trControl = ctrl)


model.gam$finalModel
```

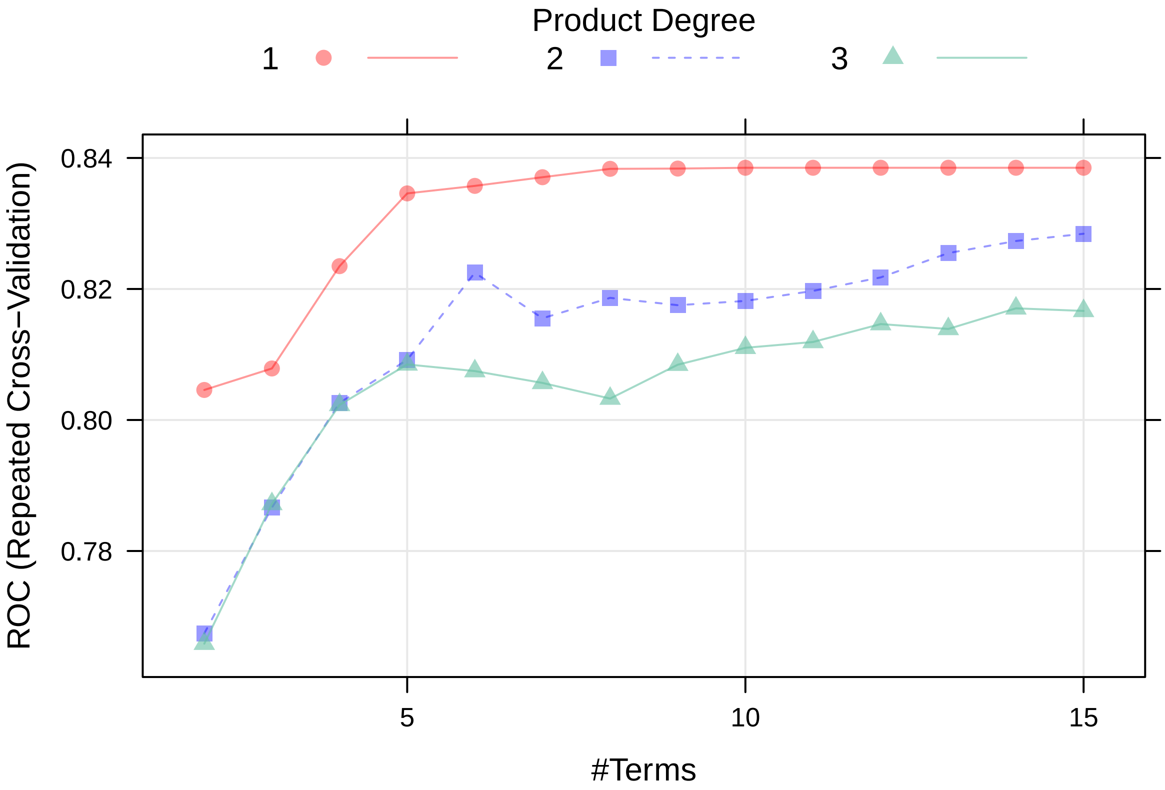
```
##
## Family: binomial
## Link function: logit
##
## Formula:
## .outcome ~ gender + hypertension + ever_married + residence_type +
##     smoking_status + work_type + s(age) + s(bmi) + s(avg_glucose_level)
##
## Estimated degrees of freedom:
## 3.6670 0.0004 0.9068  total = 11.57
##
## UBRE score: -0.7158818
```

```
plot(model.gam$finalModel, select = 3)
```



MARS

```
set.seed(1)
model.mars <- train(x = stroke_train[ , 1:10],
                    y = stroke_train$stroke,
                    method = "earth",
                    tuneGrid = expand.grid(degree = 1:3,
                                           nprune = 2:15),
                    metric = "ROC",
                    trControl = ctrl)

plot(model.mars)
```



```
coef(model.mars$finalModel)
```

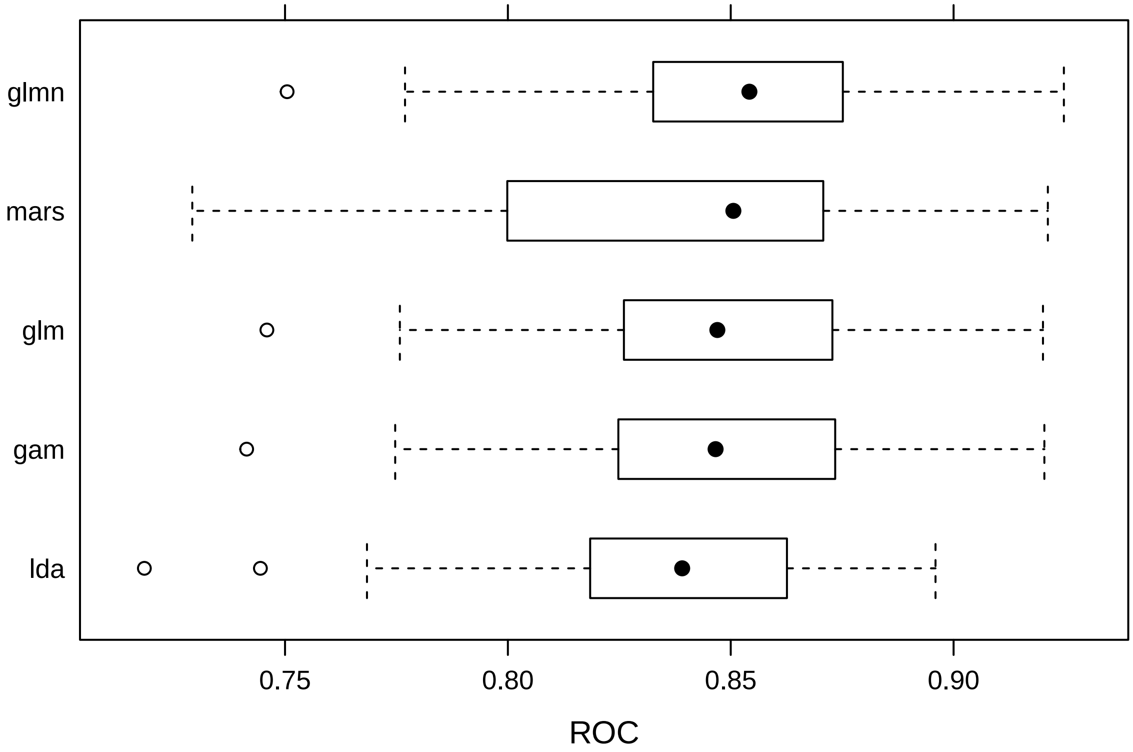
```
##                (Intercept) h(avg_glucose_level-100.6)
##               -5.207461830                0.005531031
##               hypertension              heart_disease
##                0.498466921                0.331263581
##                  h(age-81)             h(work_type-3)
##               -0.718808479               -0.334687398
##                  h(age-75)                  h(age-36)
##               -0.006336640                0.083775053
```

Model Comparison

```
res <- resamples(list(glm = model.glm, glmn = model.glmn, lda = model.lda, gam = model.gam, mars = model
roc_summary <- summary(res)$statistics[1]
roc_summary %>% knitr::kable()
```

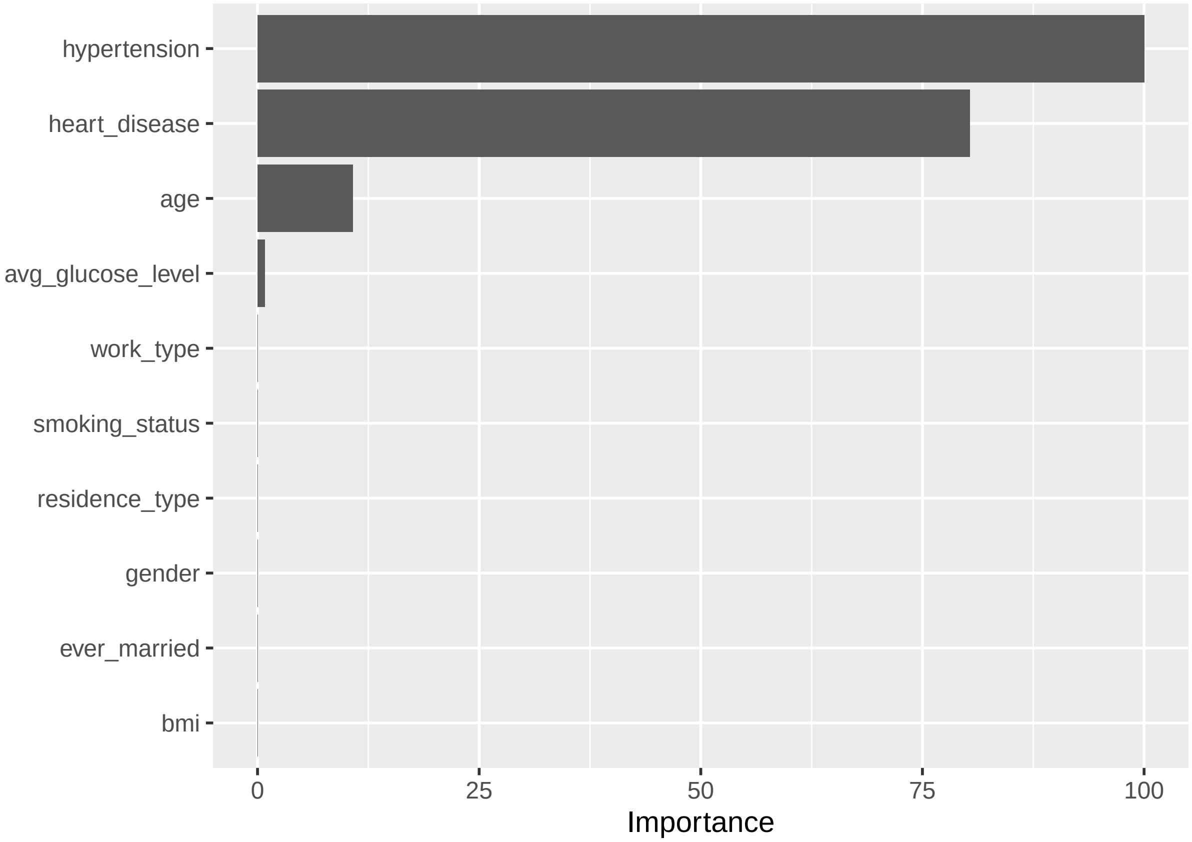
```
bwplot(res, metric = "ROC")
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
glm	0.7459324	0.8261499	0.8469962	0.8460861	0.8726924	0.9200563	0
glmn	0.7504693	0.8329161	0.8541927	0.8507349	0.8751173	0.9247497	0
lda	0.7184397	0.8185553	0.8391084	0.8351982	0.8620314	0.8959473	0
gam	0.7413955	0.8257196	0.8466051	0.8458014	0.8726631	0.9203692	0
mars	0.7291927	0.8008350	0.8505945	0.8385021	0.8699827	0.9211514	0



The penalized logistic regression model is selected to be the best model for predicting stroke because it has the highest AUC. It's ROC curve and confusion matrix using the test data are shown below

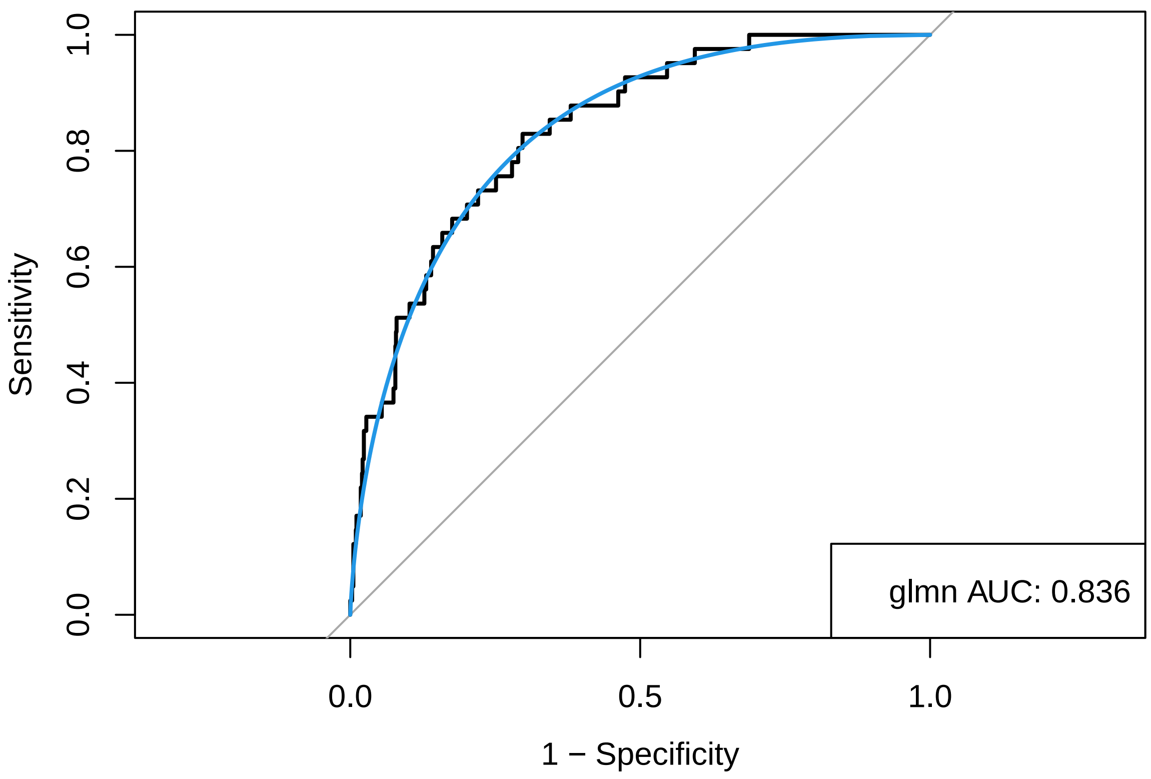
```
vip(model.glmn)
```



```
glmn.pred.prob1 <- predict(model.glmn, newdata = stroke_test, type = "prob")
roc.glmn1 <- roc(stroke_test$stroke, glmn.pred.prob1[, 2])
auc1 <- roc.glmn$auc[1]
auc1
```

```
## [1] 0.8361516
```

```
plot(roc.glmn1, legacy.axes = TRUE)
plot(smooth(roc.glmn1), col = 4, add = TRUE)
legend("bottomright", legend = paste0("glmn AUC", ": ", round(auc1, 3)), cex = 1)
```



```
glmn.pred1 <- rep("neg", nrow(glmn.pred.prob1))
glmn.pred1[glmn.pred.prob[, 2] > 0.5] <- "pos"

confusionMatrix(data = factor(glmn.pred1),
                reference = stroke_test$stroke,
                positive = "pos")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg 939  41
##        pos   0   0
##
##                Accuracy : 0.9582
##                  95% CI : (0.9437, 0.9698)
##     No Information Rate : 0.9582
##     P-Value [Acc > NIR] : 0.5414
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 4.185e-10
##
##             Sensitivity : 0.00000
##             Specificity : 1.00000
##          Pos Pred Value :     NaN
##          Neg Pred Value : 0.95816
##              Prevalence : 0.04184
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.00000
##       Balanced Accuracy : 0.50000
##
##        ’Positive’ Class : pos
##
```