https://www.kaggle.com/datasets/whenamancodes/fraud-detection?resource=download

```
data = read.csv("creditcard.csv")
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
## Loading required package: lattice
```

```
library(PRROC)
```

```
## Loading required package: rlang
```

```r
library(xgboost)
```

```
##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice
```

```r
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-10
```

```r
library(knitr)
```

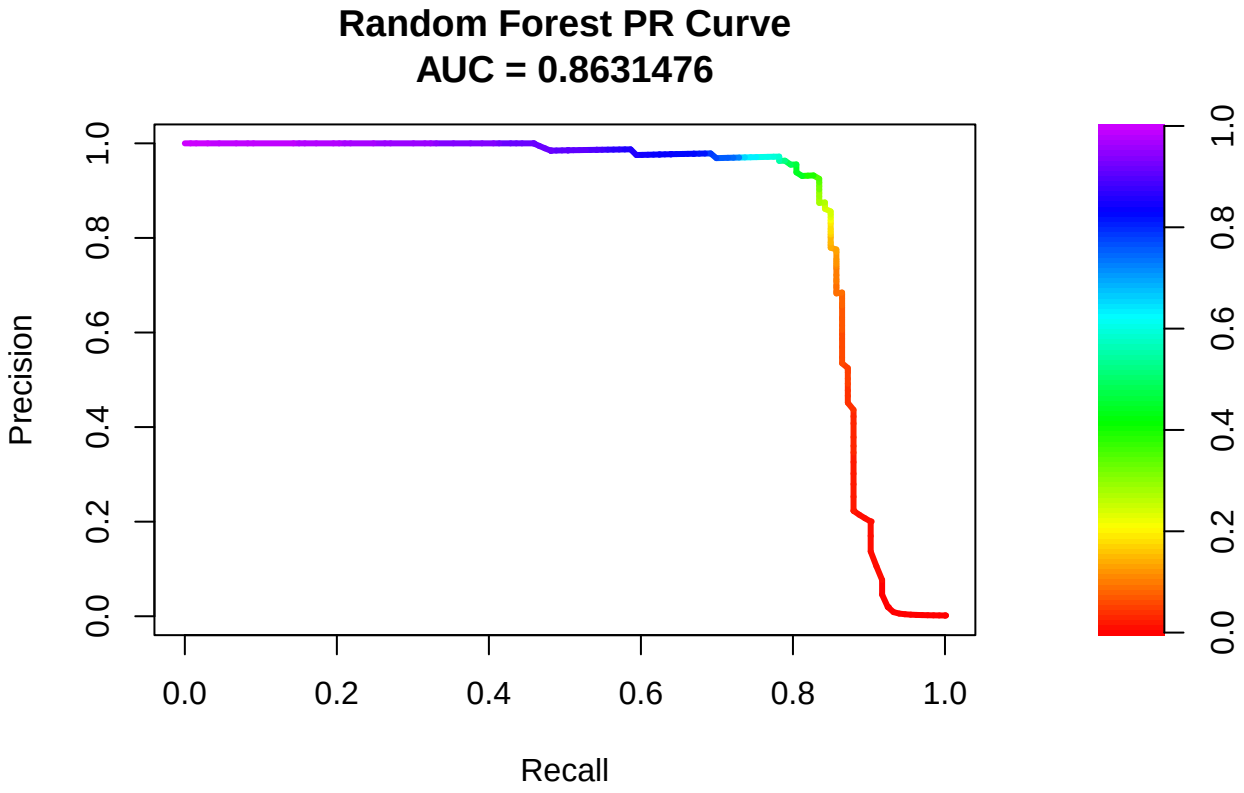
```r
set.seed(01242004)
trainIndeces = createDataPartition(data$Class, p=0.7, list = FALSE)
```

Test random forest

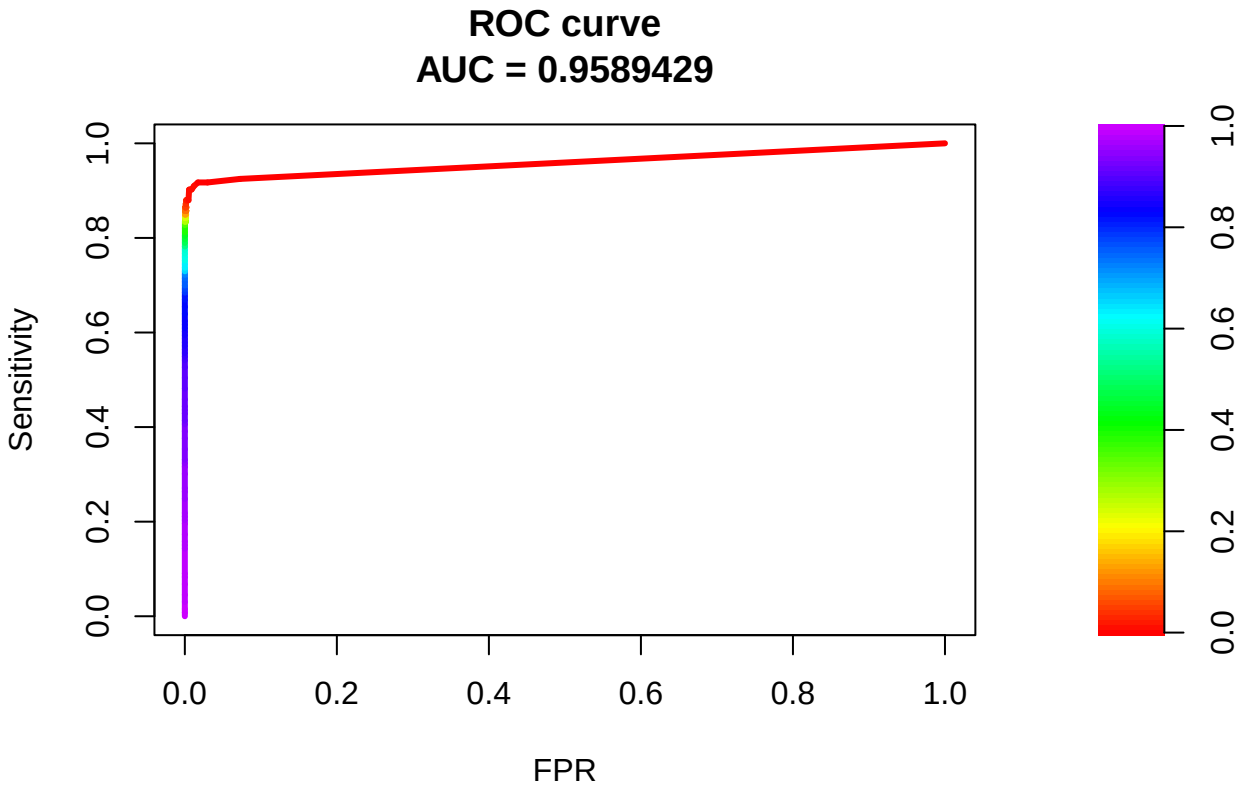
```r
RFModel = readRDS("Models/randomForest.rds")

prediction = predict(RFModel, as.matrix(data[-trainIndeces,-31]), type = "prob")

pr <- pr.curve(scores.class0 = prediction[,2], weights.class0 = data[-trainIndeces, "Class"], curve = T
plot(pr, main = "Random Forest PR Curve")
```




```
roc <- roc.curve(scores.class0 = prediction[,2], weights.class0 = data[-trainIndeces, "Class"], curve =
plot(roc)
```




```
print("Actual as rows - Predicted as columns")
```

```
## [1] "Actual as rows - Predicted as columns"
```

```
print("Prediction greater than 10%")
```

```
## [1] "Prediction greater than 10%"
```

```
table(ifelse(data[-trainIndeces, "Class"] == 1, "Fraud", "Legitimate"), ifelse(prediction[,2] >= 0.1, "
```

```
##
##              Fraud Legitimate
##   Fraud        114         19
##   Legitimate    46      85263
```

```
print("Prediction greater than 30%")
```

```
## [1] "Prediction greater than 30%"
```

```
table(ifelse(data[-trainIndeces, "Class"] == 1, "Fraud", "Legitimate"), ifelse(prediction[,2] >= 0.3, "
```

```
##
##              Fraud Legitimate
##   Fraud        111         22
##   Legitimate    12      85297
```

```
print("Prediction greater than 50%")
```

```
## [1] "Prediction greater than 50%"
```

```
table(ifelse(data[-trainIndeces, "Class"] == 1, "Fraud", "Legitimate"), ifelse(prediction[,2] >= 0.5, "
```

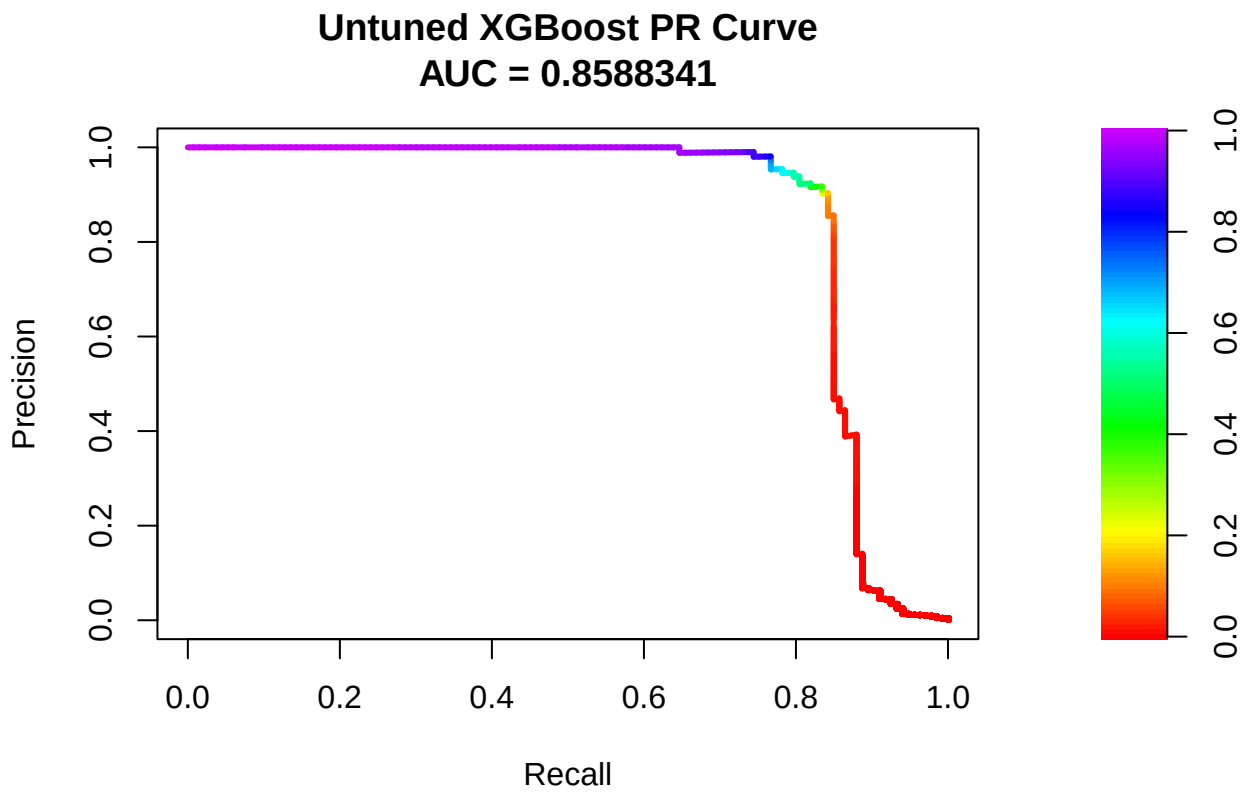
```
##
##              Fraud Legitimate
##   Fraud        105         28
##   Legitimate     4      85305
```

Test untuned xgBoost

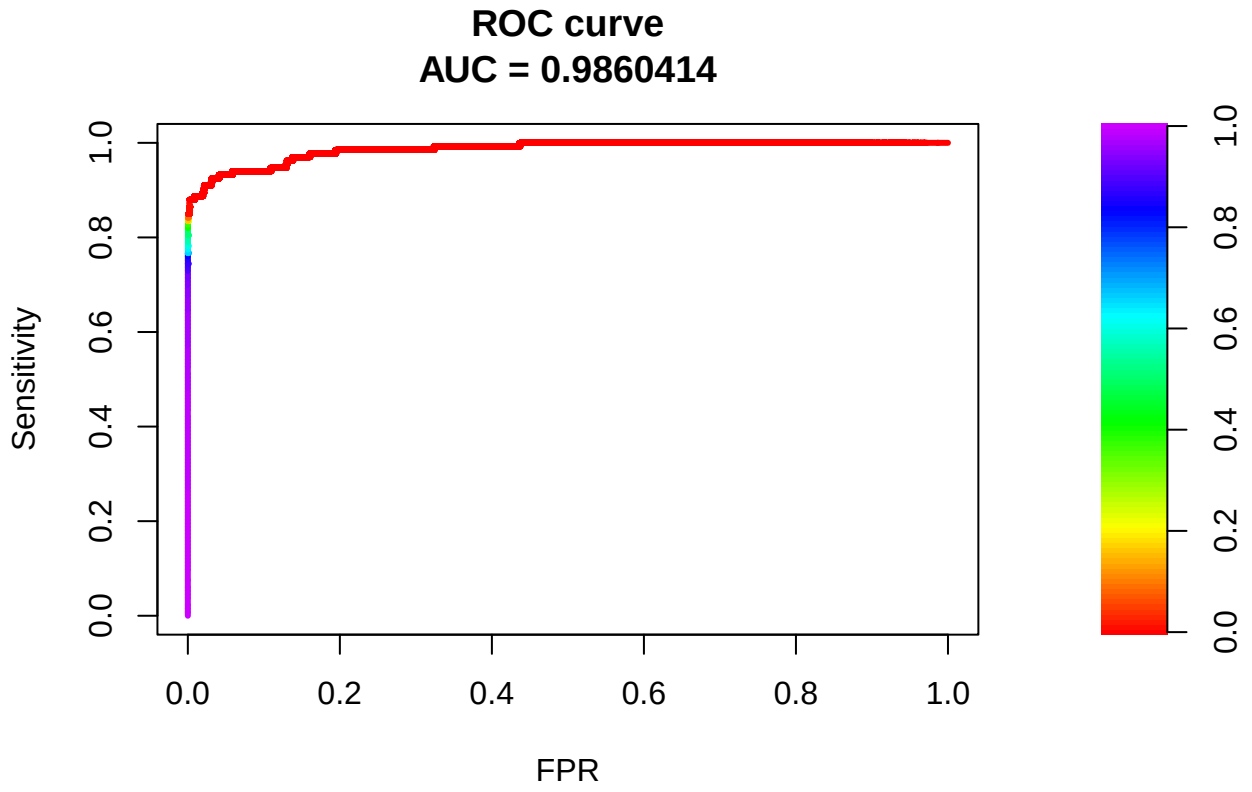
```
xgBoostUntuned = readRDS("Models/xgBoostUntuned.rds")

prediction = predict(xgBoostUntuned, as.matrix(data[-trainIndeces,-31]))

pr <- pr.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(pr, main = "Untuned XGBoost PR Curve")
```



```
roc <- roc.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(roc)
```




```
predicted = ifelse(prediction >= 0.3, 1, 0)
table(data[-trainIndeces, "Class"], predicted)
```

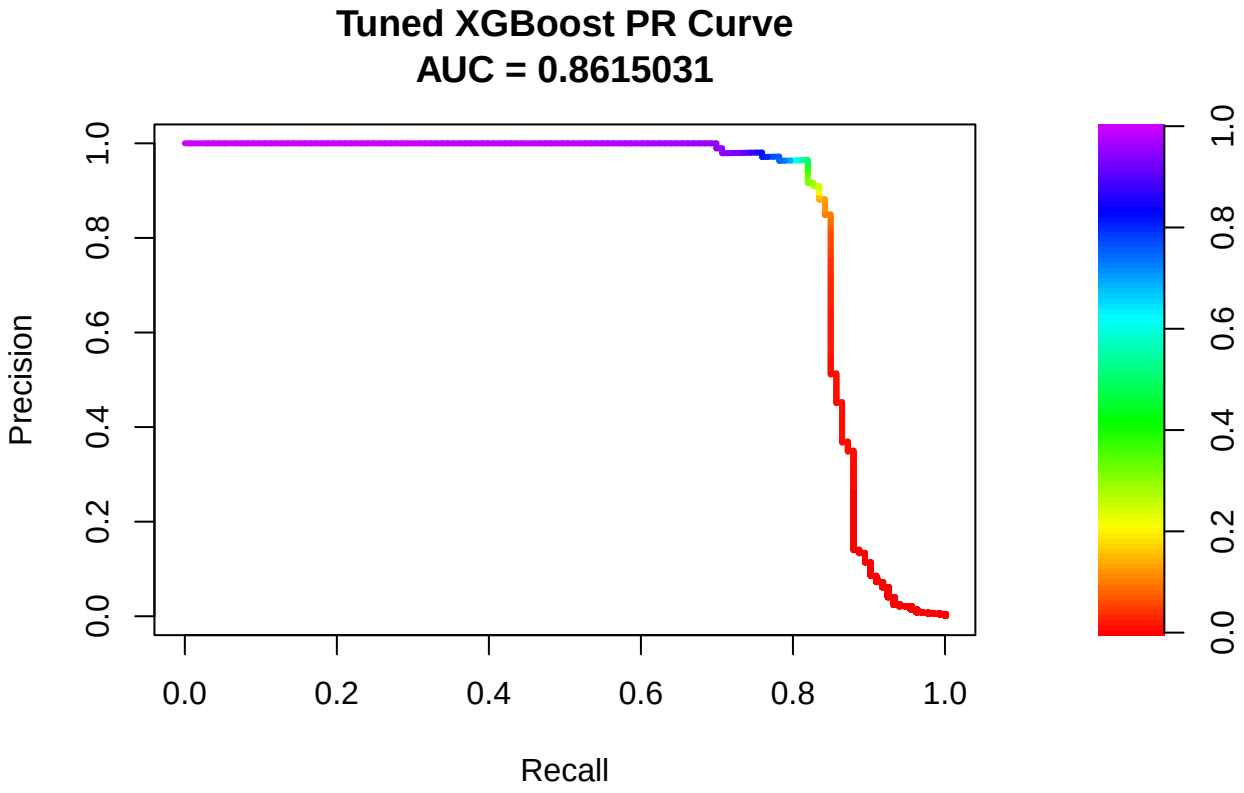
```
##    predicted
##         0     1
##   0 85298    11
##   1    22   111
```

Test tuned xgBoost

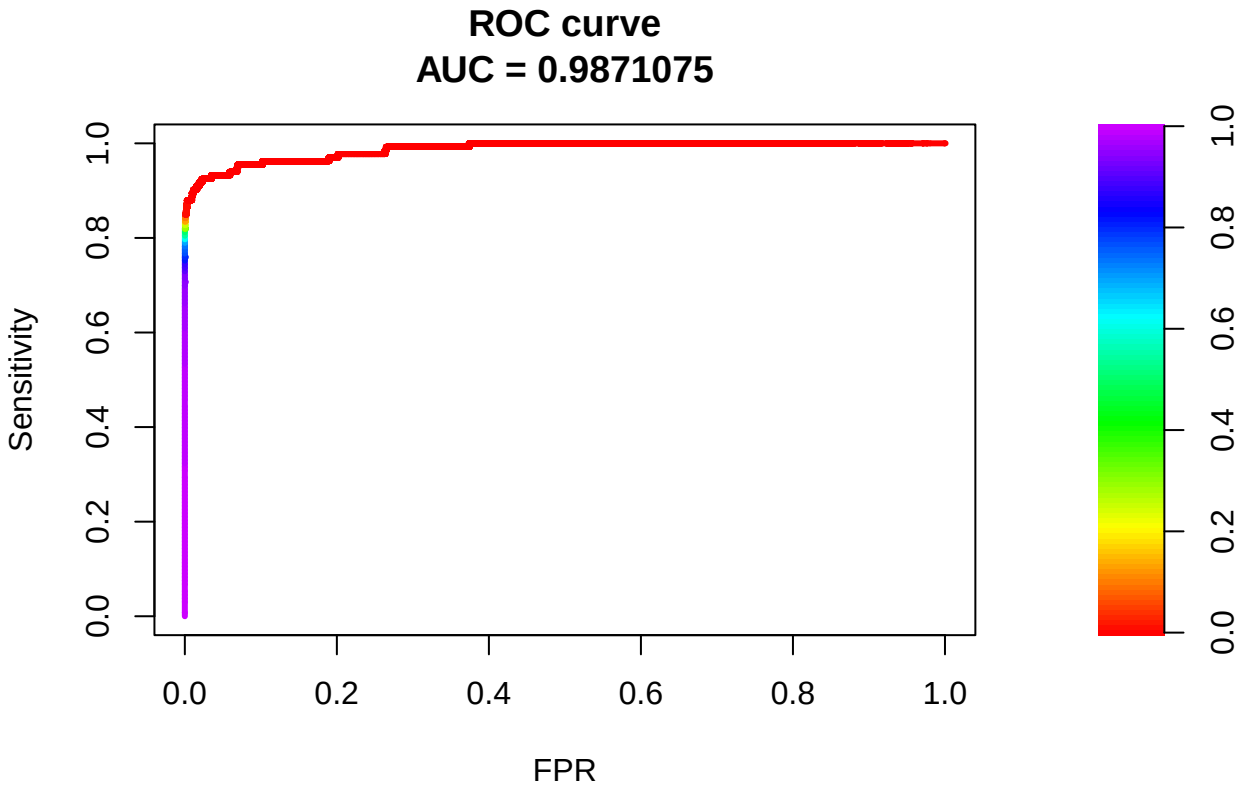
```
xgBoostTuned = readRDS("Models/xgBoostTuned.rds")

prediction = predict(xgBoostTuned, as.matrix(data[-trainIndeces,-31]))

pr <- pr.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(pr, main = "Tuned XGBoost PR Curve")
```




```
roc <- roc.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(roc)
```




```
predicted = ifelse(prediction >= 0.3, 1, 0)
table(data[-trainIndeces, "Class"], predicted)
```

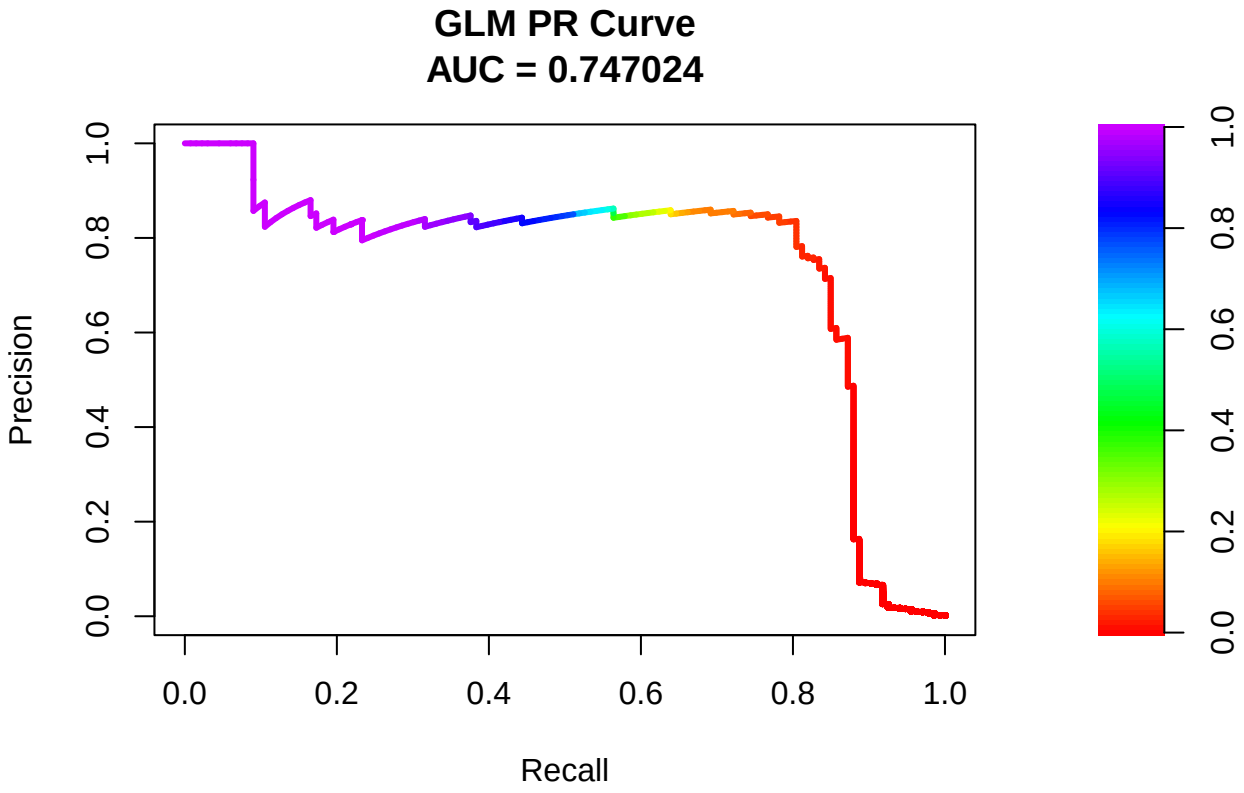
```
##    predicted
##         0     1
##   0 85299    10
##   1    24   109
```

Test GLM

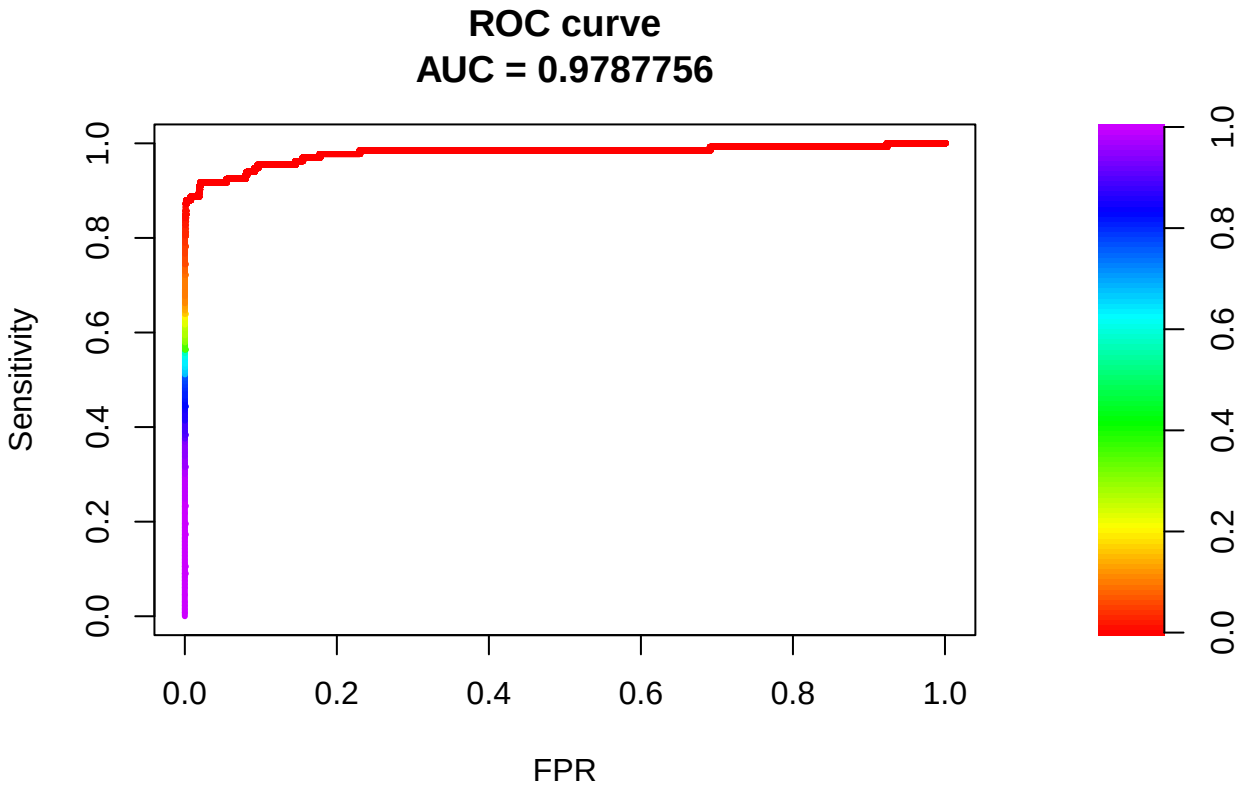
```
glm = readRDS("Models/glm.rds")

prediction = predict(glm, as.matrix(data[-trainIndeces,-31]), type = "response", s = 0.005)

pr <- pr.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(pr, main = "GLM PR Curve")
```




```
roc <- roc.curve(scores.class0 = prediction, weights.class0 = data[-trainIndeces, "Class"], curve = T)
plot(roc)
```




```
predicted = ifelse(prediction >= 0.3, 1, 0)
table(data[-trainIndeces, "Class"], predicted)
```

```
##    predicted
##         0     1
##   0 85295    14
##   1    55    78
```

Stop writing to pdf