Project 2

Akshay Kumar Varanasi (av32826)

Instructions

Please submit both this completed Rmarkdown document and its knitted HTML, converted to PDF, on Canvas **no later than 4:00 pm on April 2nd, 2019**. These two documents will be graded jointly, so they must be consistent (as in, don't change the Rmarkdown file without also updating the knitted HTML!).

All results presented **must** have corresponding code. Any answers/results given without the corresponding R code that generated the result will be considered absent. All code reported in your final project document should work properly. Please bear in mind that **you will lose points** for the following:

- an R-code chunk with no comments
- results without corresponding R code
- extraneous code which does not contribute to the question
- printing out the entire data table

For this project, you will work with a dataset was extracted from the 1974 *Motor Trend* US magazine. It contains information about fuel consumption and 10 aspects of automobile design and performance for 32 automobiles.

```
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

The column contents are as follows:

- **mpg**: miles per US gallon.
- **cyl**: number of cylinders.
- **disp**: displacement (cubic inches).
- **hp**: gross horsepower.
- **drat**: rear axle ratio.
- **wt**: weight (1000 lbs).
- **qsec**: 1/4 mile time.
- **vs**: engine (0 = V-shaped, 1 = straight).
- **am**: transmission (0 = automatic, 1 = manual).

- **gear**: number of forward gears.
- **carb**: number of carbuerators.

Problems

Problem 1: (20 points) Make a logistic regression model that predicts transmission type (`am`) from gross horsepower (`hp`) and miles per galon (`mpg`). Make another logistic regression model that also predicts transmission type from gross horsepower alone. Show the summary (using `summary`) of each model below. Make a plot with two ROC curves, and explain which model better predicts transmission type. For this analysis, use the entire dataset as training data, and do not evaluate the model on test data.

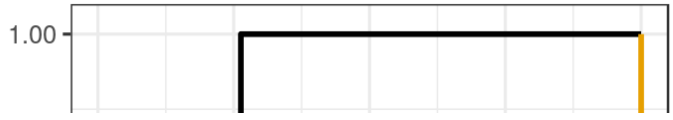
```
# model to use:
# am ~ hp + mpg

# Making logistic regression model for the above model.
glm_model1 <-glm(
  am ~ hp + mpg,
  data = mtcars,
  family = binomial
)

# Summary of the model 1
summary(glm_model1)
```

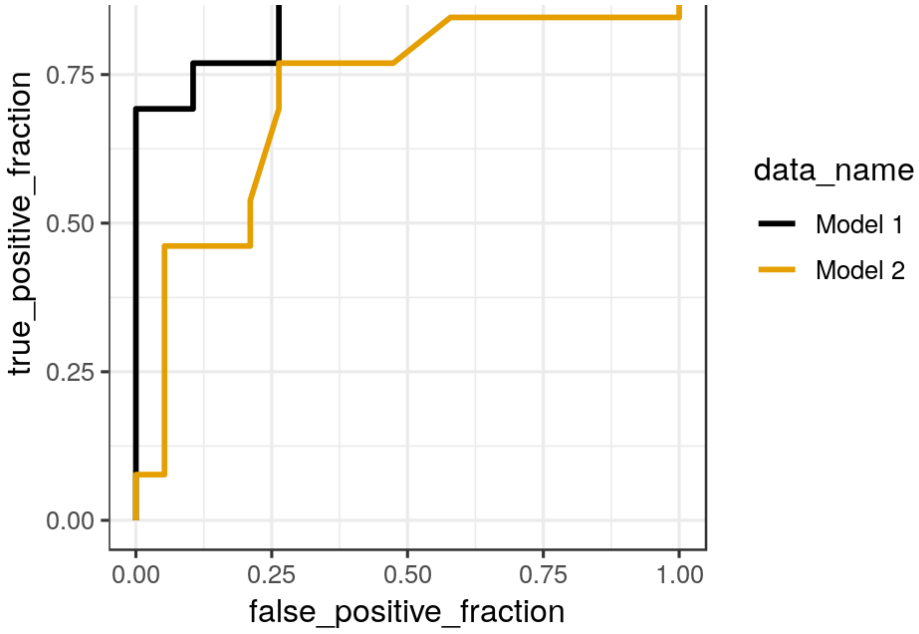
```
##
## Call:
## glm(formula = am ~ hp + mpg, family = binomial, data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.41460  -0.42809  -0.07021   0.16041   1.66500
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.60517   15.07672  -2.229   0.0258 *
## hp            0.05504    0.02692   2.045   0.0409 *
## mpg           1.25961    0.56747   2.220   0.0264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 19.233  on 29  degrees of freedom
## AIC: 25.233
##
## Number of Fisher Scoring iterations: 7
```

```
# model to use:
# am ~ hp


# Making logistic regression model for the above model.
glm_model2 <-glm(
  am ~ hp,
  data = mtcars,
  family = binomial
)

# Summary of the model 2
summary(glm_model2)
```

```
##
## Call:
## glm(formula = am ~ hp, family = binomial, data = mtcars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2955  -0.9968  -0.7818   1.1630   2.0379
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.776614   0.915429   0.848    0.396
## hp          -0.008117   0.006074  -1.336    0.181
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 41.228  on 30  degrees of freedom
## AIC: 45.228
##
## Number of Fisher Scoring iterations: 4
```

```
# results data frame for model 1
df_model1 <- data.frame(
predictor = predict(glm_model1, mtcars),
known_truth = mtcars$am,
data_name = "Model 1"
)

# results data frame for model 2
df_model2 <- data.frame(
predictor = predict(glm_model2, mtcars),
known_truth = mtcars$am,
data_name = "Model 2"
)

# Combining both the dataframes to plot ROC curves
df_combined <-rbind(df_model1,df_model2)

# Plotting the ROC curve for both the Models
ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)) +
geom_roc(n.cuts = 0) +
scale_color_colorblind()
```






```
# Calculating Area under the curve for models
p<-ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name))
+
geom_roc(n.cuts = 0)
data_name <- unique(df_combined$data_name)
data_name
```

```
## [1] Model 1 Model 2
## Levels: Model 1 Model 2
```

```
data_info <- data.frame(
data_name,
group = order(data_name)
)
left_join(data_info, calc_auc(p)) %>%
select(-group, -PANEL) %>%
arrange(desc(AUC))
```

```
## Joining, by = "group"
```

```
##   data_name       AUC
## 1   Model 1 0.9311741
## 2   Model 2 0.7125506
```

As we can see from the ROC curve and Area under curve both show that Model 1 is better than Model 2 because Model 1 uses miles per galon (`mpg`) in addition to gross horsepower (`hp`) which Model 2 uses

for prediction. More independent variables we use on which response variable depends, better is the model. We know that transmission type (`am`) depends on miles per galon (`mpg`) because the co-efficient obtained from the logistic regression for it is positive and relatively higher than that of gross horsepower (`hp`). And this makes sense because, we know that Manual transmission type cars have higher miles per galon(`mpg`) so it is reasonable to assume that higher mile per galon(`mpg`) is related to Manual transmission.

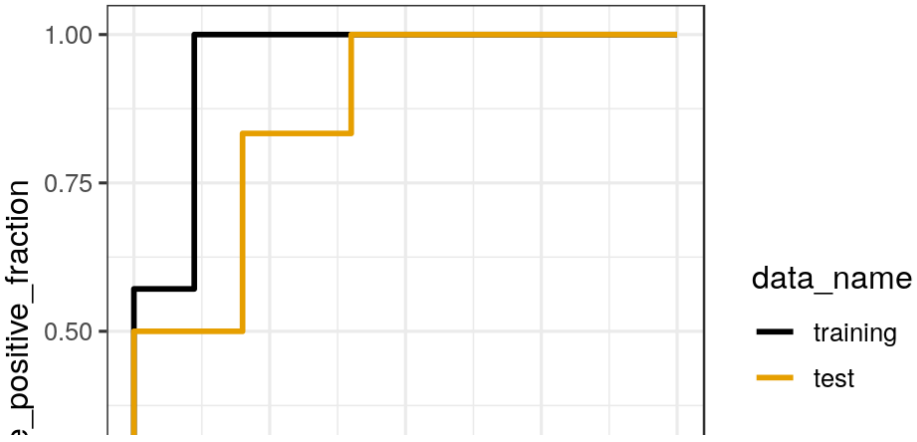
Problem 2: (40 points) We have now divided the `mtcars` dataset into a training and a test data set (`train_data` and `test_data`):

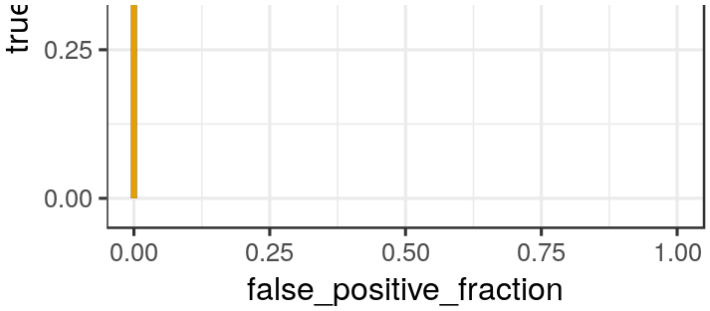
```
train_fraction <- 0.5 # fraction of data for training purposes
set.seed(123) # set the seed to make the partition reproductible
train_size <- floor(train_fraction * nrow(mtcars)) # number of observations in
training set
train_indices <- sample(1:nrow(mtcars), size = train_size)

train_data <- mtcars[train_indices, ] # get training data
test_data <- mtcars[-train_indices, ] # get test data
```

Fit a logistic regression model to predict transimission type on the training data set. Use the predictors `hp` and `mpg` to predict transimission type (`am`). Your code should be appropriately commented with high-level statements about the code's function. Using your model, predict the outcome on the test data set, and plot and discuss your results.

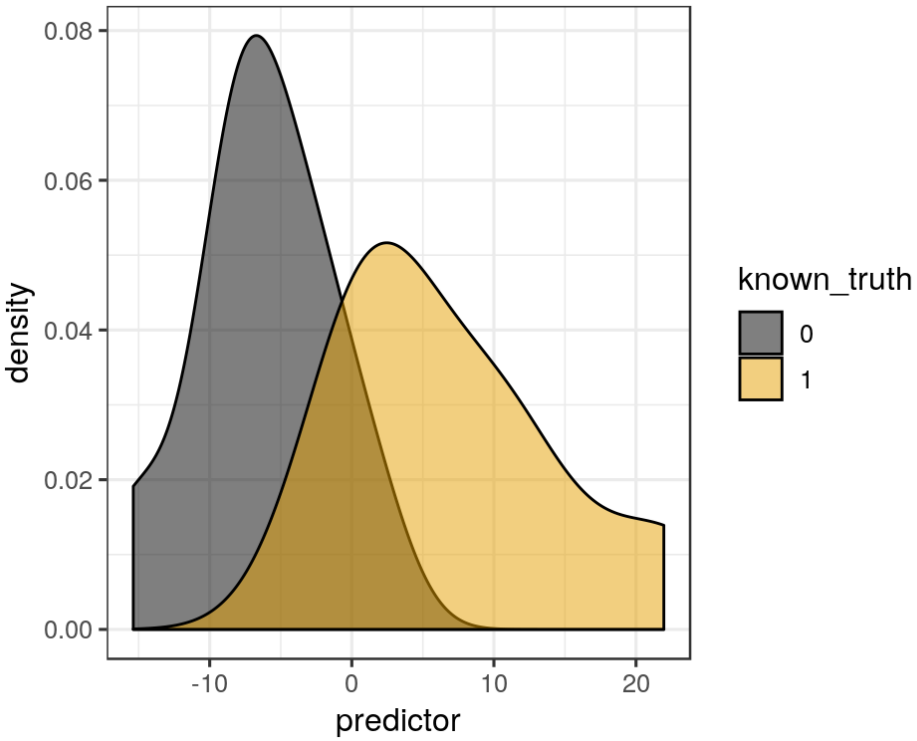
You should have two final plots: a plot with two ROC curves, one for the training and one for the test data set, and a density plot that shows how the linear predictor separates the two transmission types in the test data. Your discussion should, at least, cover the differences and similarities in model performance on the training vs. test data (including AUC) as well as a clear interpretation of each plot. Please limit your discussion to a maximum of 10 sentences.

```
# model to use:
# am ~ hp + mpg

# Making logistic regression model for the above model. (This time we use train
ing data only)
glm_model <-glm(
  am ~ hp + mpg,
  data = train_data, # Here we give only the training data instead of whole dat
aset
  family = binomial
)



# results data frame for training data
df_train <- data.frame(
predictor = predict(glm_model, train_data),
known_truth = train_data$am,
data_name = "training"
)

# results data frame for testing data
df_test <- data.frame(
predictor = predict(glm_model, test_data),
known_truth = test_data$am,
data_name = "test"
)

df_combined <-rbind(df_train,df_test)

ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)) +
geom_roc(n.cuts = 0) +
#xlim(0, 0.14) +
scale_color_colorblind()
```

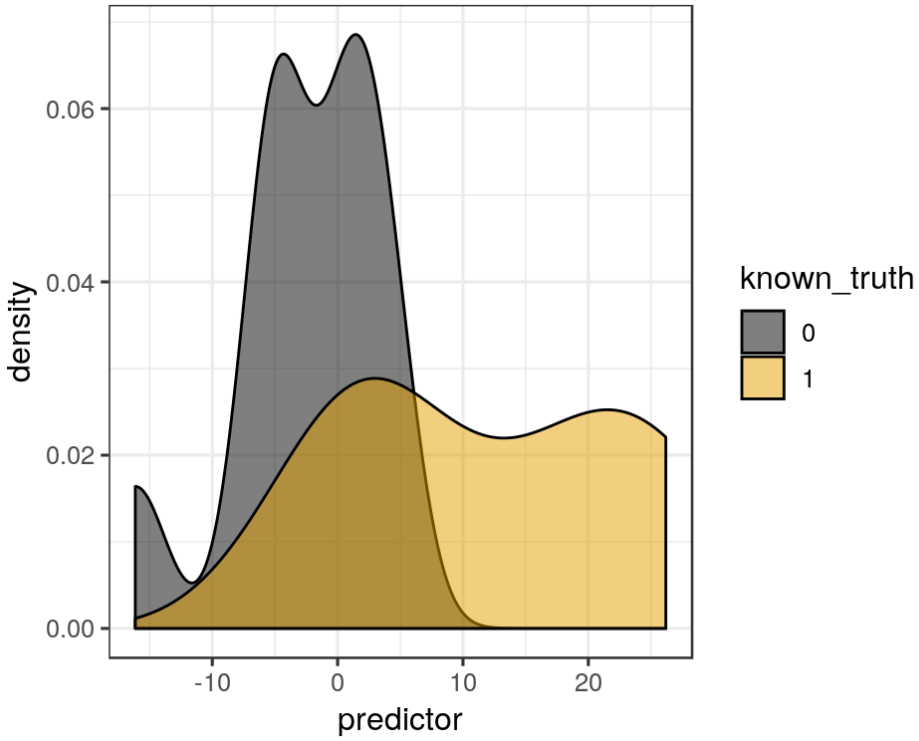





```
# results data frame for training data
df_train <- data.frame(
predictor = predict(glm_model, train_data),
known_truth = factor(train_data$am),
data_name = "training"
)

# Density plot
ggplot(df_train, aes(x = predictor, fill = known_truth )) +
geom_density(alpha = .5) +scale_fill_colorblind()
```



```
# results data frame for testing data
df_test <- data.frame(
predictor = predict(glm_model, test_data),
known_truth = factor(test_data$am),
data_name = "test"
)

# Density plot
ggplot(df_test, aes(x = predictor, fill = known_truth )) +
geom_density(alpha = .5) +scale_fill_colorblind()
```




```
# Calculating Area under the curve for model on training and test data
p<-ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name))
+
geom_roc(n.cuts = 0)
data_name <- unique(df_combined$data_name)
data_name
```

```
## [1] training test
## Levels: training test
```

```
data_info <- data.frame(
data_name,
group = order(data_name)
)
left_join(data_info, calc_auc(p)) %>%
select(-group, -PANEL) %>%
arrange(desc(AUC))
```

```
## Joining, by = "group"
```

```
##   data_name       AUC
## 1  training 0.9523810
## 2      test 0.8666667
```

We can see that model perfoms well on trained data compared to test data as both ROC and AUC are better for training data. This is because model was trained on training data so it is expected to perform better. The model does not perform better on testing data because we can see from the density plots that the density plot of training data and test data are different. Model expects to see distribution something like it is trained on(in this case training data distribution) to predict but test data distribution is different from it so the model is not able to perform as good as it performed on training data. As we can see the density of class 1 (Manual transmission) is flat and density of class 0 (Automatic transmission) has two hills (one small one) in test data unlike in the training data.

Problem 3: (40 points) Think of one **conceptual** question to ask about the dataset `mtcars`. You are welcome to use either the training, test, or full data set for this part. For your question, perform an exploratory statistical analysis (PCA, clustering, logistic regression, linear regression, ANOVA, etc.) with two corresponding figures. The analysis and plots *must* be multivariate (include at least three of the data columns). Discuss your findings, in particular how your analysis' results reveal (or don't reveal) an answer to your proposed question. Please limit your discussion to a maximum of 15 sentences.

To receive full credit for Part II, you will have to do the following:

- *Come up with one clear, conceptual question about the data, as explained above.*
- *The analysis must be multivariate (involve more than two columns of the data set at once).*
- *None of your work must repeat any part of the analysis of Part 1.*
- *For each plot, provide a justification for why you chose to make the type of plot that you made.*
- *Use different primary geoms for the two different plots.*
- *Provide an interpretation of your results and a response to your question.*

Conceptual question: *Please write your question here.*

Compare the predictors before and after PCA for classification of transmission type (`am`) using gross horsepower (`hp`), miles per galon (`mpg`) and displacement (`disp`) columns. (Use same training and test data which was used for previous question)

Please briefly describe your planned analysis and plots before doing them (5 sentences max). **Answer:** To compare the models before and after PCA for prediction, we first make dataset with PCA data. Then we make logistic regression model for both original and PCA data. We compare the model result using ROC curves and AUC values on test data.

As said, we first apply PCA on training and test data to get new data considering only gross horsepower (`hp`), miles per galon (`mpg`) and displacement (`disp`) columns.

```
# PCA on train data
train_data %>%
  select(hp,mpg,disp) %>%   # select only these columns (hp , mpg, disp)
  scale() %>%            # scale to 0 mean and unit variance
  prcomp() ->            # do PCA
  pca                    # store result as `pca`

# now display the results from the PCA analysis
pca_train <- data.frame(pca$x, am = train_data$am)
head(pca_train)
```

```
##                          PC1         PC2         PC3 am
## Merc 280         -0.7691093 -0.23144396  0.38912371  0
## Pontiac Firebird  0.8089344 -0.24101264 -1.01277660  0
## Merc 450SL        0.4266568 -0.02150589  0.01474263  0
## Fiat X1-9        -2.5391170  0.04804990 -0.12228144  1
## Porsche 914-2    -1.9950828  0.11562619 -0.19014002  1
## Mazda RX4 Wag    -1.1039624 -0.20506145  0.19839111  1
```

```
# PCA on test data
test_data %>%
  select(hp,mpg,disp) %>%   # select only these columns (hp , mpg, disp)
  scale() %>%            # scale to 0 mean and unit variance
  prcomp() ->            # do PCA
  pca                    # store result as `pca`

# now display the results from the PCA analysis
pca_test <- data.frame(pca$x, am = test_data$am)
head(pca_test)
```

```
##                           PC1        PC2        PC3 am
## Mazda RX4         -0.37720303  0.3302830  0.1297385  1
## Datsun 710        -0.92046048  0.3598877  0.2222460  1
## Hornet 4 Drive     0.01735294  0.0603692 -0.4411631  0
## Hornet Sportabout  1.29994337 -0.3560091 -0.3170457  0
## Valiant            0.11849582  0.5855675 -0.2400374  0
## Duster 360         2.33125256 -0.3719142  0.4761955  0
```

Now we make logistic regression models using 3 columns of training data and the data obtained using PCA on 3 columns of training data.

```
# am ~ hp + mpg + disp

# Making logistic regression model for the above model using original data.
glm_model1 <-glm(
  am ~ hp + mpg + disp ,
  data = train_data,
  family = binomial
)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Summary of the model 1
summary(glm_model1)
```

```
##
## Call:
## glm(formula = am ~ hp + mpg + disp, family = binomial, data = train_data)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.870e-05  -2.110e-08  -2.110e-08   2.110e-08   2.804e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -819.596 802631.514  -0.001    0.999
## hp               1.642   2060.985   0.001    0.999
## mpg             34.315  32784.796   0.001    0.999
## disp            -0.375    608.816  -0.001    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.1930e+01  on 15  degrees of freedom
## Residual deviance: 1.7973e-09  on 12  degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25
```

```
# am ~ pc1 + pc2 + pc3

# Making logistic regression model for the above model using pca data
glm_model2 <-glm(
  am ~ PC1 + PC2 + PC3,
  data = pca_train,
  family = binomial
)
```

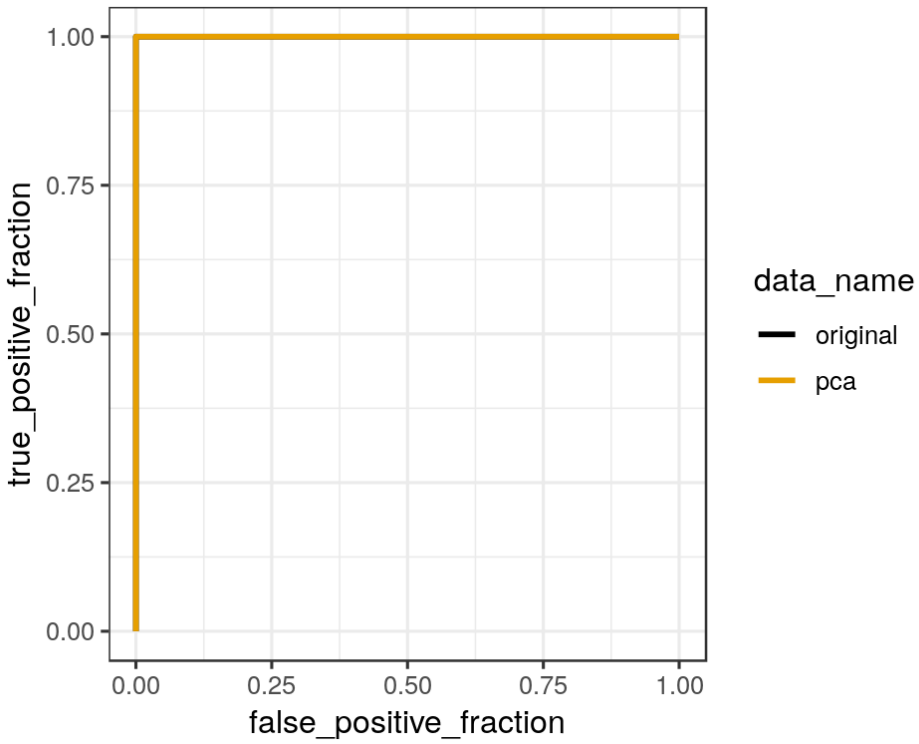
```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# Summary of the model 2
summary(glm_model2)
```

```
##
## Call:
## glm(formula = am ~ PC1 + PC2 + PC3, family = binomial, data = pca_train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.870e-05  -2.110e-08  -2.110e-08   2.110e-08   2.804e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.152  37232.541   0.000    1.000
## PC1            -69.232  67337.695  -0.001    0.999
## PC2            197.237 209857.817   0.001    0.999
## PC3            -78.022  96344.950  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.1930e+01  on 15  degrees of freedom
## Residual deviance: 1.7973e-09  on 12  degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25
```

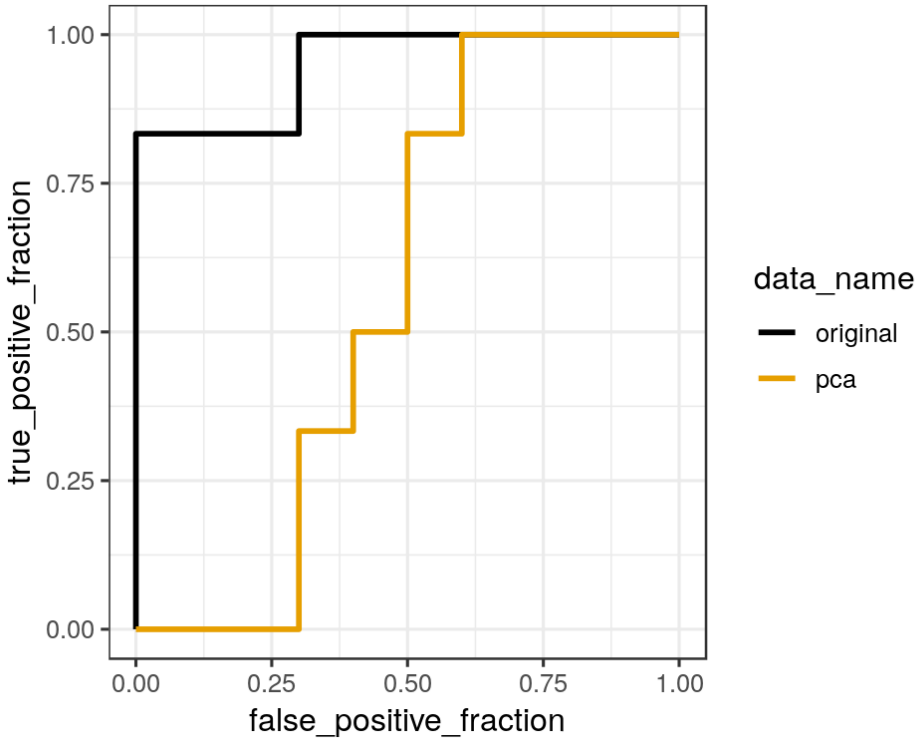
ROC curve is plotted for both to see which model is better on train data.

```
# Comparing model on training data

# results data frame for original data
df_original <- data.frame(
predictor = predict(glm_model1, train_data),
known_truth = train_data$am,
data_name = "original"
)

# results data frame for pca data
df_pca <- data.frame(
predictor = predict(glm_model2, pca_train),
known_truth = pca_train$am,
data_name = "pca"
)

df_combined <-rbind(df_original,df_pca)

# ROC curve of both
ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)) +
geom_roc(n.cuts = 0) +
scale_color_colorblind()
```



ROC curve is plotted for both to see which model is better on test data.

```
# results data frame for original data
df_original <- data.frame(
predictor = predict(glm_model1, test_data),
known_truth = test_data$am,
data_name = "original"
)

# results data frame for pca data
df_pca <- data.frame(
predictor = predict(glm_model2, pca_test),
known_truth = pca_test$am,
data_name = "pca"
)

df_combined <-rbind(df_original,df_pca)

# ROC curve of both
ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)) +
geom_roc(n.cuts = 0) +
scale_color_colorblind()
```



We now calculate AUC for both so that we can compare.

```
# Calculating Area under the curve for model based on original and pca data
p<-ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name))
+
geom_roc(n.cuts = 0)
data_name <- unique(df_combined$data_name)
data_name
```

```
## [1] original pca
## Levels: original pca
```

```
data_info <- data.frame(
data_name,
group = order(data_name)
)
left_join(data_info, calc_auc(p)) %>%
select(-group, -PANEL) %>%
arrange(desc(AUC))
```

```
## Joining, by = "group"
```

```
##   data_name       AUC
## 1  original 0.9500000
## 2       pca 0.5666667
```

As we can see, ROC plot and AUC values are better using model based on original data rather than model based on PCA data. Though both the models perform well on training data, model based on PCA data fails to perform well on test data. If we remember, PCA was used as means to transform the data such that it makes more sense or explains better in most cases like for prediction, but in this case it is not true.