Predicting Student Outcomes

Cody Rorick and Hannah Swickheimer
MA 5790
October 2024

Abstract:

Students dropping out of University in the middle of a degree is problematic for a variety of reasons. One of the most critical reasons being that the student spends money on tuition without having the degree necessary to obtain a job in their field of study. This can harm the student as it sets them back without providing the ability to recoup lost time and money. Data was collected at Capacitação da Administração Pública in Portugal in order to determine what characteristics might potentially make up a student that is at risk of dropping out. Having this information would allow third parties to step in early and intervene before the student has already dropped out. In order to determine which characteristics are actually important in predicting a dropout, different regression models will be tested and evaluated.

Table of Contents

1 Background

By the time a student has dropped out of school, it is already too late to intervene in their education. It is important to get additional support to the student well before they have reached the point of dropping out. This requires knowledge of demographic/academic/social-economic information from students early on in their academic career, and then the result of that academic career. This can give the investigator an idea of what factors may contribute to future students dropping out before graduation.

A program was created by SATDAP - Capacitação da Administração Pública under grant POCI-05-5762-FSE-000191 in Portugal to collect various information about students for this study. The intention of the study was to use machine learning techniques to identify the most at risk students in order to reduce academic dropout.

Given in the dataset is an indicator of whether or not the student has graduated, dropped out, or is still enrolled at the end of normal duration of courses. All of the other data in this dataset is information about the student prior to enrollment or during the beginning of their enrollment. If there is a strong relationship between the early indicators and the resulting status of the student at the end of normal course enrollment, the study will have successfully identified at-risk students, and can take action to reduce the academic dropout rate.

2 Variable Introduction and Definitions:

There are 35 predictor variables collected from a population of 4424 student samples. The response variable is the Target variable listed below that indicates whether or not a student has graduated, is still enrolled, or has dropped out. Below is a list of all of the variables that were used with descriptions of the information that they contain. The naming of the variables is consistent with how the names were given in the dataset. The case study that this data was collected and utilized for was published in 2021.

Marital status - *Factor indicating the student's marriage*
Application mode - *department applied for*
Daytime/evening attendance - *Factor variable indicating if a student attends day or evening classes*
Previous qualification - *Highest education achieved before this enrollment*
Previous qualification (grade) - *Grade of previous qualification (between 0 and 200)*
Nacionality - *Factor variable indicating the nationality of the student*
Mother's qualification - *Highest education of mother of student*
Father's qualification - *Highest education of father of student*
Mother's occupation - *Industry mother of student works in*
Father's occupation - *Industry father of student works in*
Admission grade - *Grade of the student upon entry*
Displaced - *Factor indicating whether or not the student has been displaced*
Educational special needs - *Factor indicating whether or not the student has educational special needs*
Debtor - *Factor indicating whether the student is a debtor*
Tuition fees up to date - *Factor indicating whether the student has up to date tuition fees.*
Gender - *Factor indicating the gender of the student*
Scholarship holder - *Factor indicating whether or not the student holds a scholarship*
Age at enrollment - *Age of the student upon entry*
International - *Factor indicating whether or not the student is international*
Curricular units 1st sem (credited) - *Number of curricular units credited in the 1st semester*
Curricular units 1st sem (enrolled) - *Number of curricular units enrolled in the 1st semester*
Curricular units 1st sem (evaluations) - *Number of evaluations to curricular units in the 1st semester*
Curricular units 1st sem (approved) - *Number of curricular units approved in the 1st semester*
Curricular units 1st sem (grade) - *Grade average in the 1st semester (between 0 and 20)*

Curricular units 1st sem (without evaluations) - *Number of curricular units without evaluations in the 1st semester*
Curricular units 2nd sem (credited) - *Number of curricular units credited in the 2nd semester*
Curricular units 2nd sem (enrolled) - *Number of curricular units enrolled in the 2nd semester*
Curricular units 2nd sem (evaluations) - *Number of evaluations to curricular units in the 2nd semester*
Curricular units 2nd sem (approved) - *Number of curricular units approved in the 2nd semester*
Curricular units 2nd sem (grade) - *Grade average in the 2nd semester (between 0 and 20)*
Curricular units 2nd sem (without evaluations) - *Number of curricular units without evaluations in the 1st semester*
Unemployment rate - *Unemployment rate (%)*
Inflation rate - *Inflation rate (%)*
GDP - *Gross Domestic Product*
Target - *Three category classification (dropout, enrolled, and graduate) at the end of the normal duration of the course*

3 Preprocessing of the predictors

The first step taken was to convert nominal variables that had more than two levels into binary “dummy variables”. The dummy variables served the purpose of allowing the multi-level predictors to be used in statistical models. After this, near- zero variance was checked for amongst the predictors. Low variability indicates that the predictor doesn't change much regardless of what values the response variable is, so it won’t help us in prediction. Seven of the predictors were shown to have low variability so these were removed. This dataset did not contain any missing values, so imputation was not necessary. We did an outlier check by viewing the histograms of continuous predictors and there weren’t any outliers, which was expected as the repository of the data stated that all unexplained outliers were already removed. Due to the lack of outliers, spatial sign transformation was deemed to be unnecessary.

a. Correlations

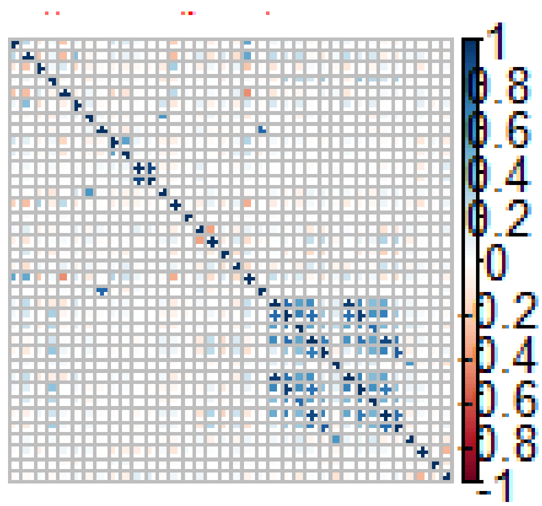


Figure 1. Correlation plot for student success data

We chose a correlation of 0.7 as a criteria to indicate a higher correlation between predictors. Eight predictors had correlations above this threshold. The predictors indicating credit enrollment and approval for the 1st and 2nd semesters had the highest correlation amongst predictors. PCA was performed for logistic regression, linear discriminant analysis, and KNN for dimension reduction, which removed unnecessary noise from predictors. Regularization in PLSDA and the penalized GLM model do not require PCA, so PCA was not necessary for these models.

b. Transformations

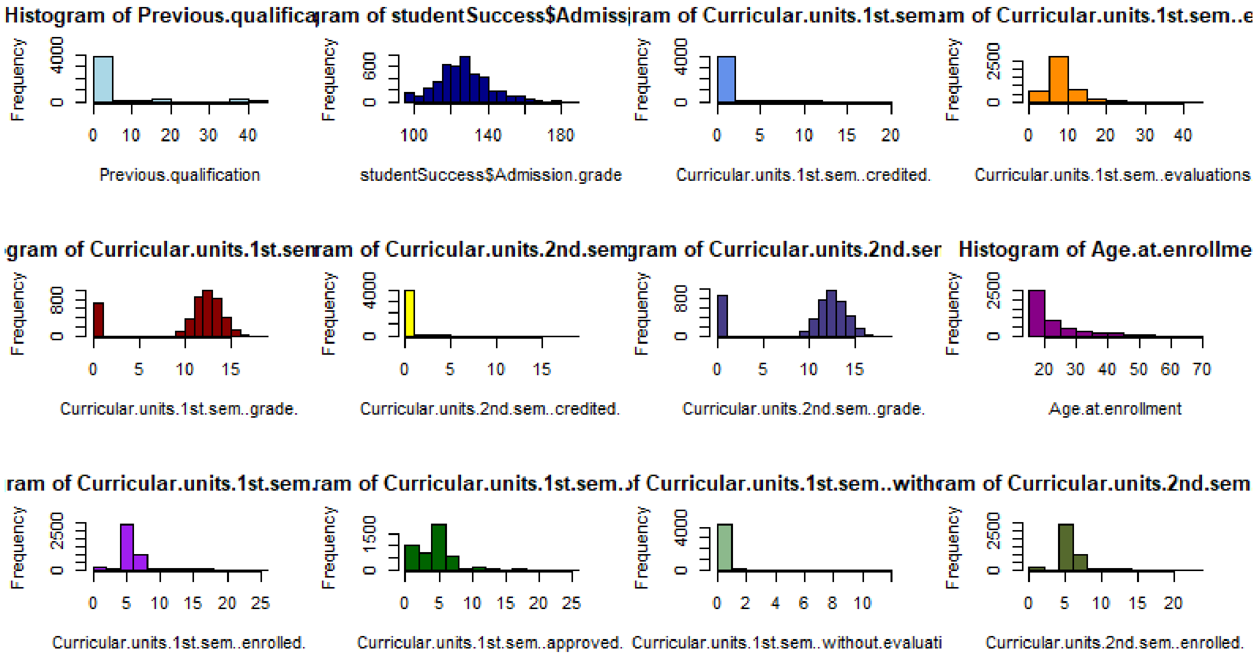


Figure 2. Distributions of skewed predictors

Marital.status	4.396781234
Application.mode	0.392769235
Application.order	1.879774573
Course	-3.806552522
Daytime.evening.attendance.	-2.505537676
Previous.qualification	2.869260050
Previous.qualification..grade.	0.312655359
Nacionality	10.696740185
Mother.s.qualification	0.001977136
Father.s.qualification	-0.298494697
Mother.s.occupation	5.335606967
Father.s.occupation	5.391515151
Admission.grade	0.530240105
Displaced	-0.194336045
Educational.special.needs	9.148769112
Debtor	2.433001498
Tuition.fees.up.to.date	-2.347460964
Gender	0.620857879
Scholarship.holder	1.164081053
Age.at.enrollment	2.053595052
International	6.100690964
Curricular.units.1st.sem..credited.	4.166222082
Curricular.units.1st.sem..enrolled.	1.617943168
Curricular.units.1st.sem..evaluations.	0.975974528
Curricular.units.1st.sem..approved.	0.765742859
Curricular.units.1st.sem..grade.	-1.567082365
Curricular.units.1st.sem..without.evaluations.	8.201838342
Curricular.units.2nd.sem..credited.	4.631677018
Curricular.units.2nd.sem..enrolled.	0.787579149
Curricular.units.2nd.sem..evaluations.	0.336269026
Curricular.units.2nd.sem..approved.	0.306071721
Curricular.units.2nd.sem..grade.	-1.312759491
Curricular.units.2nd.sem..without.evaluations.	7.262773225
Unemployment.rate	0.211907279
Inflation.rate	0.252204237
GDP	-0.393801022

Figure 3. Skewness values of predictors

Out of the 19 continuous predictors in the dataset, 13 of them were considered skewed when comparing their absolute value to a skewness threshold of 1. In order to remediate the effects of skewed variables, a box cox transformation was applied to them for the models that don't already perform their own regularization. The penalized GLM model for instance did not require this transformation for skewed predictors. The next transformation we considered was centering and scaling, since many of the variables were measured on different scales. This scaling difference can be observed by comparing the units on the "studentSuccess$Admission grade" histogram and comparing it to the "Previous.qualification" histogram for example in the figure above. To correct for this effect, predictors were centered and scaled so that equal weighting could be placed on them in the models.

4 Splitting of the Data:

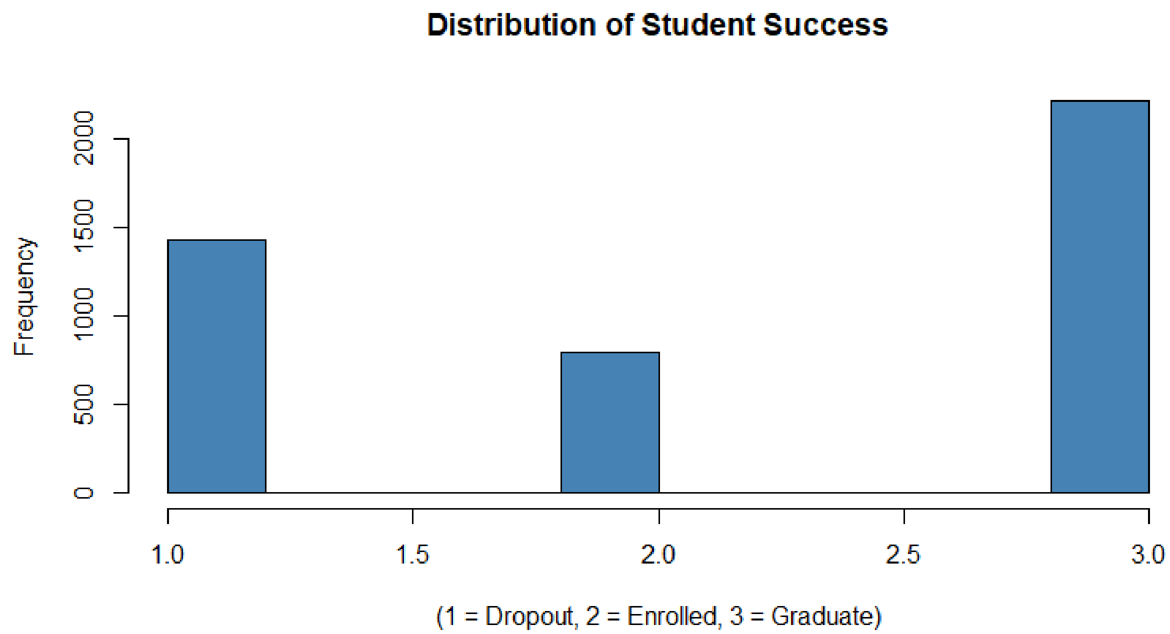


Figure 4. Distribution of outcome

We had 4424 samples to use in our model fitting process. We decided that this was enough samples to split the data into both a training and a testing set. The split decided upon was to use 80 percent for the training data and 20% for the test data. The response variable that we were predicting had 3 nominal levels. Students were either enrolled, graduated, or dropped out. In order to decide how to split the data up, we first wanted to see how balanced these levels were. As shown above, the response variable was very imbalanced between the three different outcomes, so it was clear that using a stratified random sampling approach to split the data made the most sense. This allowed representative samples of the responses to be used in both the training and testing sets. This resulted in 3,541 samples in the training set and 883 samples in the test set. A cross validation approach was used for resampling in the training data to train our models. Each model was tested with 10 cross validation iterations and a 75% train/25% predict split for each resample.

5 Model Fitting

Both linear and non-linear classification models were trained to the data. Considering the outcome has three classes with unbalanced distribution, Kappa was used to evaluate model performance, as it takes into account the distribution of classes. Table 1 and Table 2 summarize the optimal parameters and their performance of the training models. See Appendix 1 for figures demonstrating the parameters of the models and their Kappa values.

Classification model	**Tuning Parameters**	**Kappa**	**Accuracy**
Logistic Regression	decay = 0.1	0.3684	0.6456
Linear Discriminant Analysis	dimen = 2	0.3661	0.6449
PLSDA	ncomp = 2	0.3777	0.6546
Penalized GLM	α = 0.2, λ = 0.01	0.4307	0.6749

Table 1. Optimal tuning parameters and their performance of linear classification models on training set

Classification model	**Tuning Parameters**	**Kappa**	**Accuracy**
k-NN	k = 7	0.3267	0.6079
Nonlinear Discriminant Analysis	subclasses = 2	0.3606	0.6345
Neural Network	size = 3; decay = 0.0001	0.4227	0.6659
Flexible Discriminant Analysis	degree = 1; nprune =26	0.3833	0.6526
Support Vector Machine	sigma = 0.0412; C = 1	0.3556	0.6442
Naive Bayes	laplace = 0; usekernal = T; adjust = 1	0.3568	0.6224

Table 2. Optimal tuning parameters and their performance of non-linear classification models on training set

The top two models were selected to predict on the test set for the final model selection. Table 3 below demonstrated the Kappa value from both Penalized GLM and Neural Network models. Overall, the Penalized GLM was determined to be the best model for predicting, with a slightly higher Kappa value.

Classification Model	Kappa	Accuracy
Penalized GLM	0.4938	0.7067
Neural Network	0.4770	0.6874

Table 3. Summary of Penalized model and Neural Network

Table 4 shows the confusion matrix of the Penalized GLM and Table 5 shows the sensitivity and specificity between the three classes. It is important to consider sensitivity in our model, considering the goal of the study is to reduce academic dropout, and it is important to have accurate predictions in the outcome to determine what factors can be contributing to academic success. The classes dropout and graduate have a satisfactory sensitivity rate, while enrolled has a concerningly low sensitivity.

Prediction	**Dropout**	**Enrolled**	**Graduate**
Dropout	210	39	44
Enrolled	35	35	35
Graduate	39	84	362

Table 4. Confusion matrix of Penalized GLM

Statistics	**Dropout**	**Enrolled**	**Graduate**
Sensitivity	0.7113	0.1900	0.8889
Specificity	0.8664	0.9572	0.6652

Table 5. Sensitivity and Sensitivity within classes using Penalized GLM

6 Summary

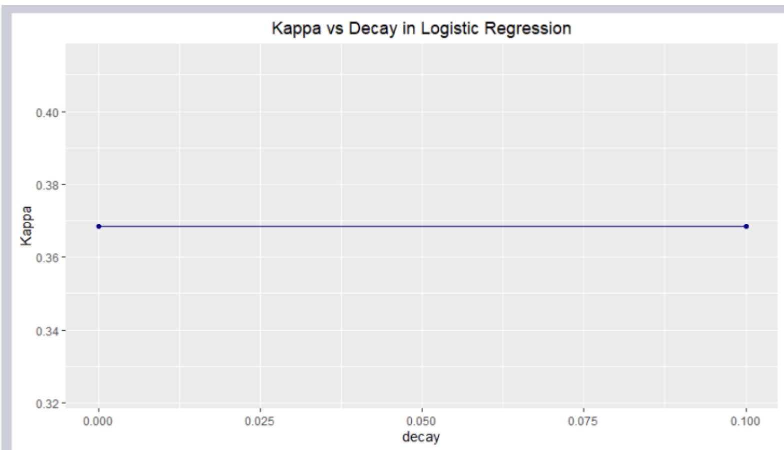
We have concluded that the optimal model is the Penalized GLM model as this model had the highest Kappa at 0.4307 and the highest accuracy rate at 0.6749 of all of the models. We thought the results from this model were good but could be improved given the breadth of the predictors that were given. One possible avenue to improve accuracy would be to group enrolled and graduated students into one category and dropped out students into another category, as a lot of the accuracy was lost when attempting to differentiate between graduated and enrolled students, and this information isn't of particular importance to us when we are trying to predict students that are at risk of dropping out. One problem to consider with this model build in predicting at risk students is the lack of diversity in the student population sampled. This study was done collecting samples of students only in Portugal, which may bias the model in one way or another. To get a better representation and understanding of at-risk students, it would be good to expand the sample to other geographical areas.

Appendix: Supplemental Material for Categorical Outcome Models with Three Levels

A. Linear classification models

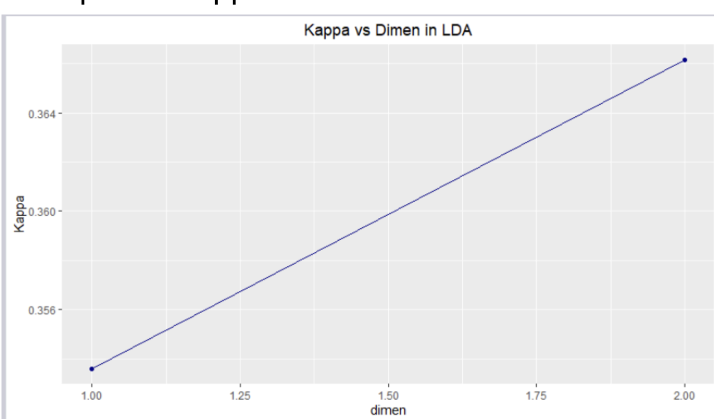
Logistic Regression model:

The optimal Kappa for the logistic regression model was found at a decay of 0.1



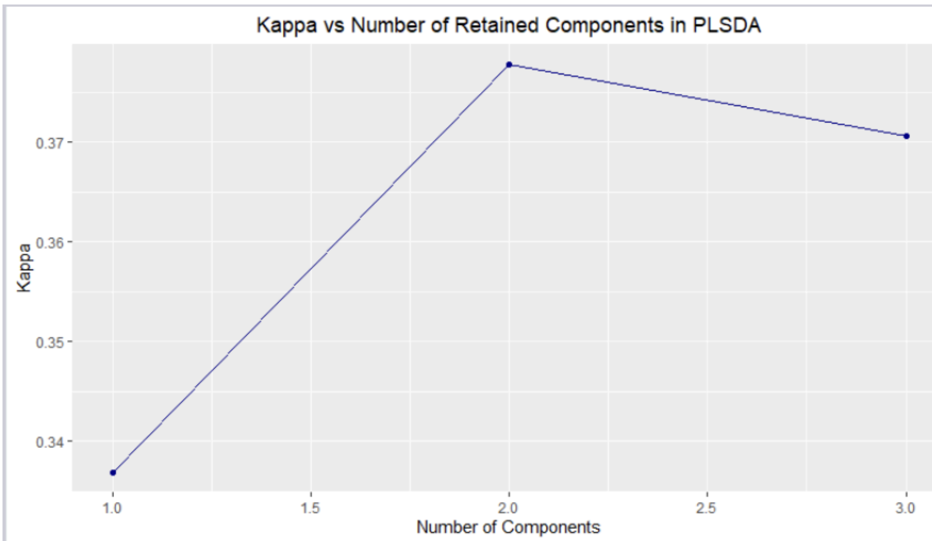
Linear Discriminant Analysis model:

The optimal Kappa in the LDA model was found when using 2 dimensions.



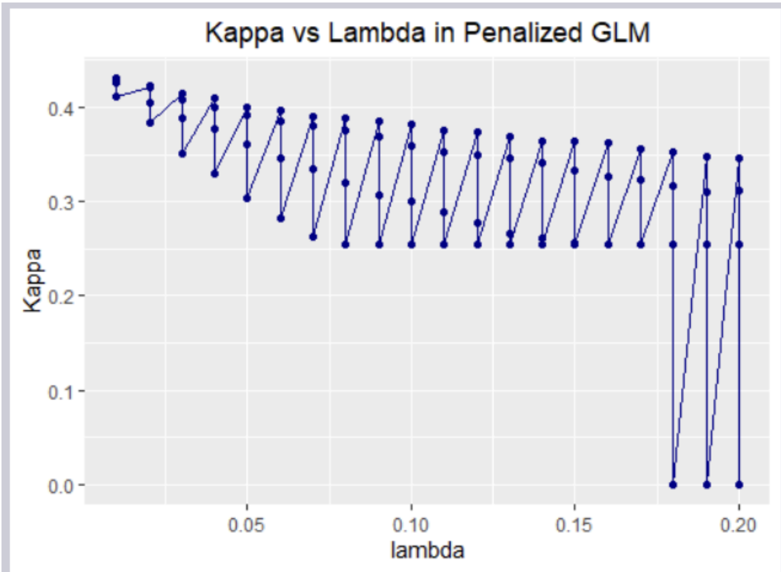
Partial Least Squares Discriminant Analysis model:

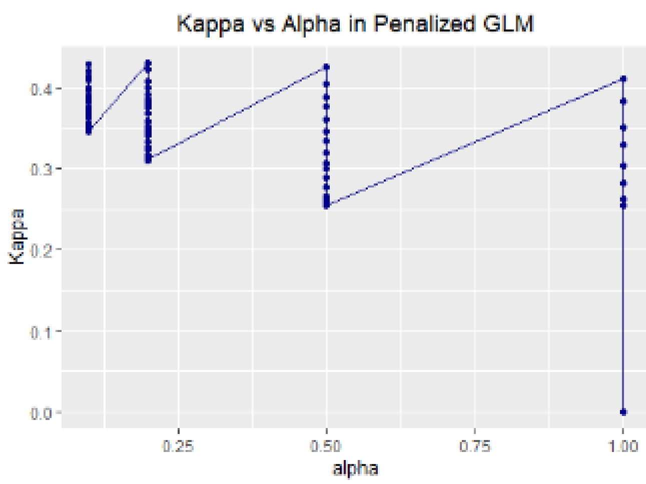
The optimal number of components for Kappa was found to be 2.



Penalized Generalized Linear Model:

α = 0.2, λ = 0.01 are the parameters to optimize Kappa in the penalized general linear model

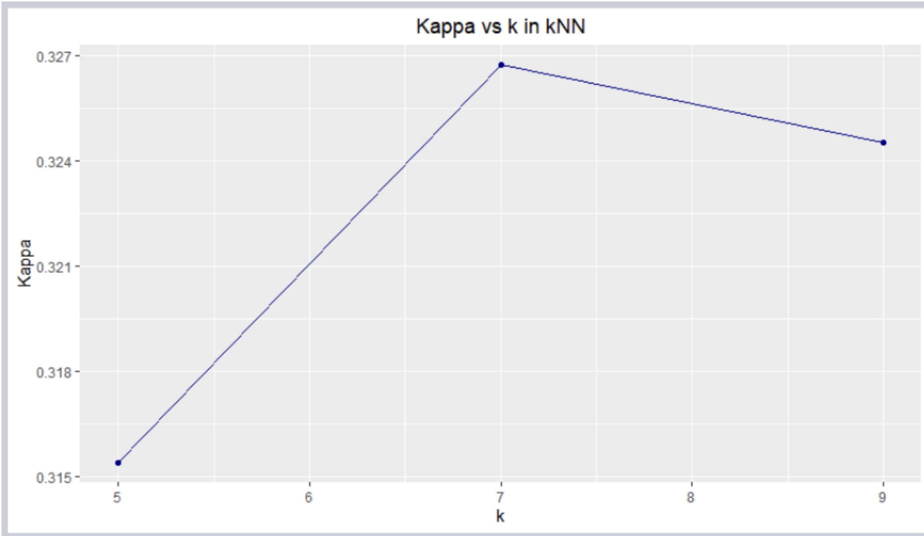




B. Nonlinear Classification models

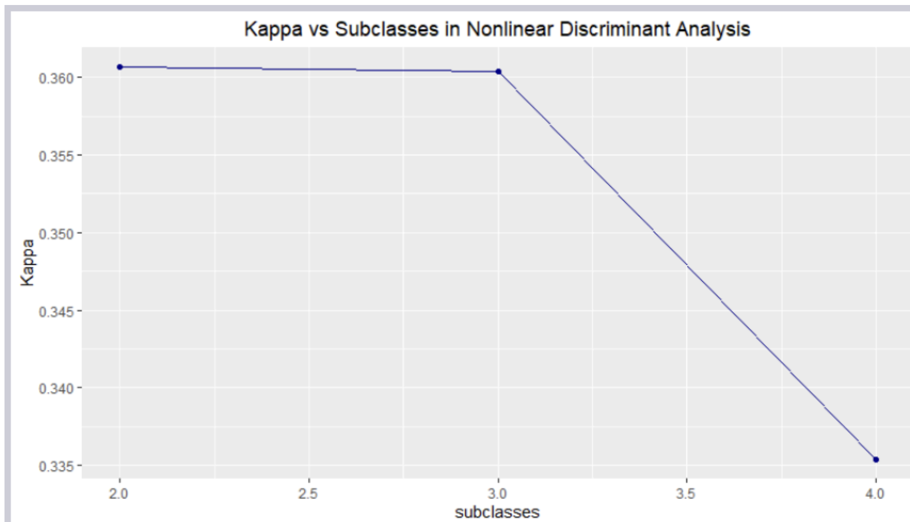
K Nearest Neighbors model:

The optimal number of neighbors was found to be 7 for the largest Kappa.


Kappa vs k in kNN
0.327
0.324
0.321
0.318
0.315
Kappa
5
6
7
8
9
k

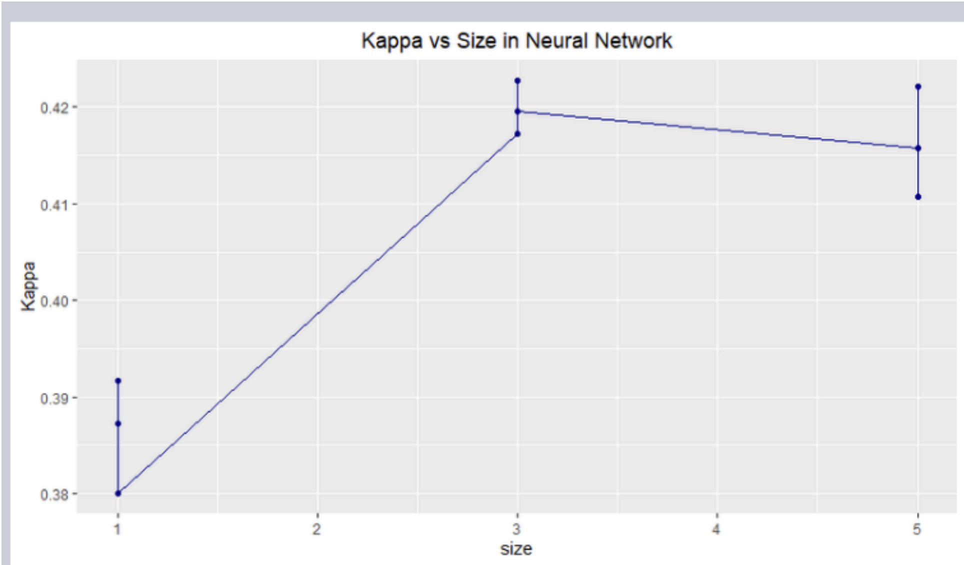
Mixed Discriminant analysis model:

The optimal Kappa was found with 2 subclasses for the Mixed Discriminant Analysis model.

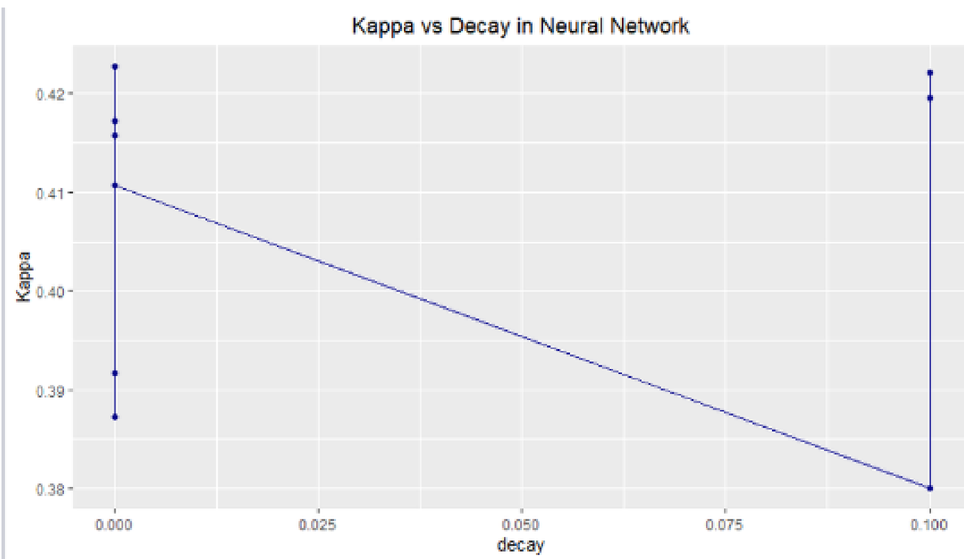


Neural Network:

size = 3; decay = 0.0001 are the parameters to optimize Kappa in the neural network model



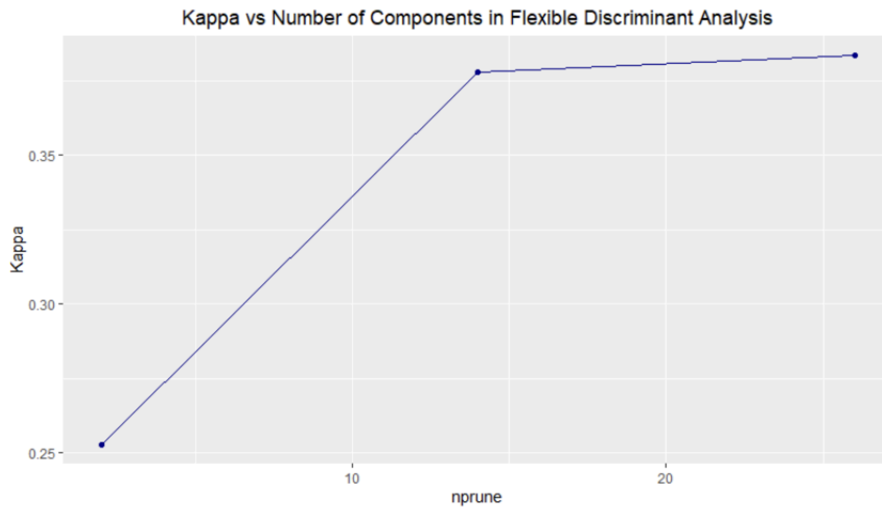
Kappa vs Size in Neural Network
0.42
0.41
0.40
0.39
0.38
Kappa
1
2
3
4
5
size



Kappa vs Decay in Neural Network
0.42
0.41
0.40
0.39
0.38
Kappa
0.000
0.025
0.050
0.075
0.100
decay

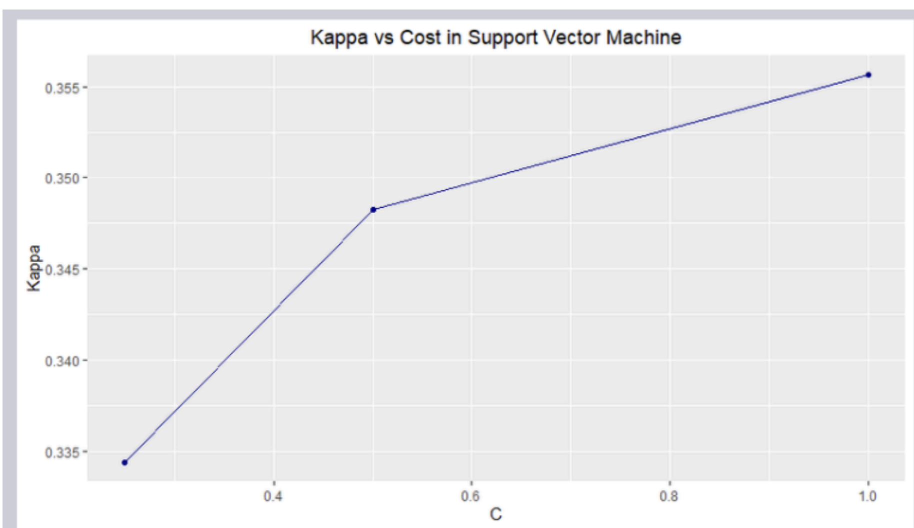
Flexible Discriminant Analysis model:

degree = 1; nprune =26 are the parameters that maximized Kappa in the FDA model



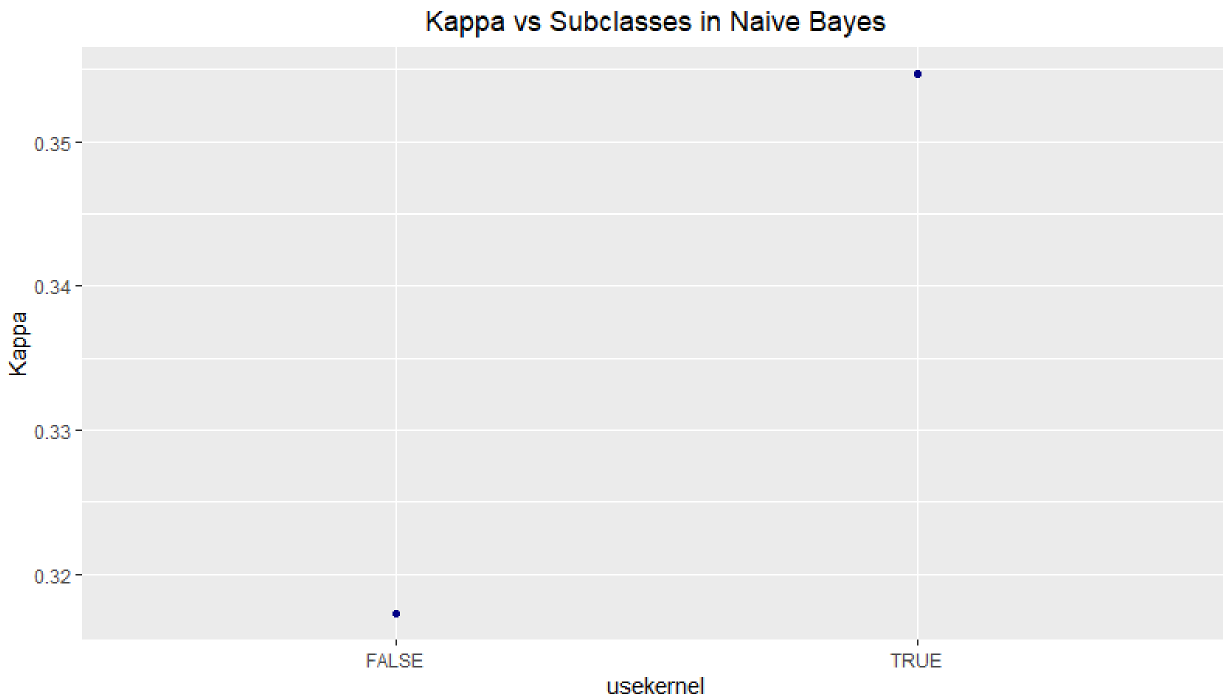
Support Vector Machine model:

sigma = 0.0412; C = 1 are the parameters that maximized Kappa in the SVM model



Naive Bayes model:

laplace = 0; usekernal = T; adjust = 1 are the parameters that maximized Kappa in the naive bayes model.



R Code:

```
# Loading in needed packages

library(caret)

library(moments)

library(corrplot)

library(AppliedPredictiveModeling)

library(ggplot2)

library(e1071)

library(glmnet)

library(MASS)

library(pamr)

library(pROC)

library(sparseLDA)

library(dplyr)

# Reading data into dataframe, data is seperated by semicolon so using that as
delimeter

setwd("/Users/crorick/Documents/MS\ Applied\ Stats\ Fall\
2023/MA5790/group_project/R_code")

studentSuccess <- read.csv('data.csv', sep = ';', header = TRUE)

# Getting a view of what the data looks like

str(studentSuccess)

studentSuccess$Target

# check distribution of response

par(mfrow = c(1,1))
```

```
hist(as.numeric(factor(studentSuccess$Target)), col = "steelblue",

    main = "Distribution of Student Success",

    xlab = "(1 = Dropout, 2 = Enrolled, 3 = Graduate)")

# Histogram shows very different frequencies for the various levels, a stratified sampling approach should be taken

#### pre-processing ###

# check for missing values

nas_bycol <- colSums(is.na(studentSuccess))

sum(is.na(studentSuccess))

# no missing values

# separate response from predictors

success_ouctome <- studentSuccess$Target # saves outcome separately

success_predictors <- studentSuccess

success_predictors$Target <- NULL # removes outcome from predictor set

str(success_predictors)

str(success_ouctome)

success_ouctome <- as.factor(success_ouctome)

## check skewness

# histograms of predictors

par(mfrow = c(3,4))

#hist(Previous.qualification, col = "lightblue")

hist(studentSuccess$Admission.grade, col = "darkblue")

hist(studentSuccess$Curricular.units.1st.sem..credited., col = "cornflowerblue")

hist(studentSuccess$Curricular.units.1st.sem..evaluations., col = "darkorange")
```

```
hist(studentSuccess$Curricular.units.1st.sem..grade., col = "darkred")

hist(studentSuccess$Curricular.units.2nd.sem..credited., col = "yellow")

# hist(Curricular.units.2nd.sem..evaluations.) # not skewed trying to lower margins

hist(studentSuccess$Curricular.units.2nd.sem..grade., col = "darkslateblue")

#hist(Unemployment.rate)

#hist(GDP)

#hist(Previous.qualification..grade.)

hist(studentSuccess$Age.at.enrollment, col = "darkmagenta")

hist(studentSuccess$Curricular.units.1st.sem..enrolled., col = "purple")

par(mfrow = c(2,2))

hist(studentSuccess$Curricular.units.1st.sem..approved., col = "darkgreen")

hist(studentSuccess$Curricular.units.1st.sem..without.evaluations., col =
"darkseagreen")

hist(studentSuccess$Curricular.units.2nd.sem..enrolled., col = "darkolivegreen")

#hist(Curricular.units.2nd.sem..approved.)

hist(studentSuccess$Curricular.units.2nd.sem..without.evaluations., col =
"darkslategrey")

#hist(Inflation.rate)

# Getting indices of columns that do contain continuous variables so we can find the
skewness of these variables

continuous_cols <- sapply(studentSuccess, is.numeric)

skew.values <- apply(studentSuccess[ ,continuous_cols], 2, skewness)

print(skew.values)

# will need transformation

# make categorical dummy vars
```

```
dummies <- dummyVars(~ ., data = success_predictors)

success_new_pred <- predict(dummies, success_predictors)

# only numeric values

success_pred_numeric <- success_predictors %>% select_if(is.numeric) # select only
numeric predictors

corrplot(cor(success_pred_numeric))

# check for multicollinearity

corr_pred <- cor(success_new_pred)

corrplot(corr_pred)

corr_pred2 <- cor(continuous_cols)

# check how many are highly correlated

high_cor <- findCorrelation(corr_pred, cutoff = 0.7)

length(high_cor)

str(high_cor)

without_high_cor <- success_new_pred[, - high_cor]

length(without_high_cor)

length(high_cor)

str(without_high_cor)

# 8 highly correlated predictors

# will need remedied - PCA for all besides PLSDA

# check for near-zero variance

degenerate_predictors <- nearZeroVar(success_new_pred)

degenerate_predictors
```

```
seg_data <- without_high_cor[, - degenerate_predictors] # new object without near-zero variance predictors

length(seg_data)

str(seg_data)

head(seg_data)

head(degenerate_predictors)

# 7 near-zero variance predictors removed

#### split data ####

training_rows <- createDataPartition(success_ouctome, p = 0.8, list = FALSE)

# training set

train_predictors <- seg_data[training_rows,]

train_response <- success_ouctome[training_rows]

# test set

test_predictors <- seg_data[-training_rows,]

test_response <- success_ouctome[-training_rows]

# set 10-fold CV

ctrl <- trainControl(method = "cv", number = 10)

### tune models ###

# logistic model

set.seed(123)

log_tune <- train(train_predictors, train_response,

            method = "multinom",

            preProc = c("BoxCox", "center", "scale", "pca"),

            metric = "Kappa",
```

```
              trControl = ctrl)

# lda model
lda_tune <- train(train_predictors, train_response,
              method = "lda2",
              preProc = c("BoxCox", "center", "scale", "pca"),
              metric = "Kappa",
              trControl = ctrl)



# plsda model
plsda_tune <- train(train_predictors, train_response,
              method = "pls",
              preProc = c("center", "scale"),
              metric = "Kappa",
              trControl = ctrl)

# penalized model
glmnGrid <- expand.grid(.alpha = c(.1, .2, .5, 1),
                   .lambda = seq(.01, .2, length = 20))

penalized_tune <- train(train_predictors, train_response,
               method = "glmnet",
```

```
                 preProc = c("center", "scale"),
                 metric = "Kappa",
                 tuneGrid = glmnGrid,
                 trControl = ctrl)

# KNN model
knn_tune <- train(train_predictors, train_response,
                method = "knn",
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
                trControl = ctrl)

# Nonlinear Discriminant Analysis
set.seed(476)
nda_tune <- train(train_predictors, train_response,
                method = "mda",
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
                trControl = ctrl)

# neural networks - nonlinear
nnet_tune <- train(train_predictors, train_response,
                method = "nnet",
```

```
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
                trControl = ctrl)

# flexible discriminant analysis
fda_tune <- train(train_predictors, train_response,
                method = "fda",
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
                trControl = ctrl)

# svm
svm_tune <- train(train_predictors, train_response,
                method = "svmRadial",
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
                trControl = ctrl)

# naive bayes
naivebayes_tune <- train(train_predictors, train_response,
                method = "naive_bayes",
                preProc = c("BoxCox", "center", "scale", "pca"),
                metric = "Kappa",
```

```
                 trControl = ctrl)


# model tuning results
log_tune
lda_tune
plsda_tune
penalized_tune
knn_tune
nda_tune
nnet_tune
fda_tune
svm_tune
naivebayes_tune



# fit models to test data
log_model_test <- predict(log_tune, newdata = test_predictors)
lda_model_test <- predict(lda_tune, newdata = test_predictors)
plsda_model_test <- predict(plsda_tune, newdata = test_predictors)
glm_model_test <- predict(penalized_tune, newdata = test_predictors)
knn_model_test <- predict(knn_tune, newdata = test_predictors)
nda_model_test <- predict(nda_tune, newdata = test_predictors)
nnet_model_test <- predict(nnet_tune, newdata = test_predictors)
```

```r
fda_model_test <- predict(fda_tune, newdata = test_predictors)
svm_model_test <- predict(svm_tune, newdata = test_predictors)
naiveb_model_test <- predict(naivebayes_tune, newdata = test_predictors)

## plot tuning parameters ##

# log plot
ggplot(data = log_tune$results, aes(x = decay, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Decay in Logistic Regression") +
  theme(plot.title = element_text(hjust = 0.5))

# LDA plot
ggplot(data = lda_tune$results, aes(x = dimen, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Dimen in LDA") +
  theme(plot.title = element_text(hjust = 0.5))

# PLSDA plot
ggplot(data = plsda_tune$results, aes(x = ncomp, y = Kappa)) +
  geom_line(colour = "darkblue") +
```

```
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Number of Retained Components in PLSDA",
       x = "Number of Components") +
  theme(plot.title = element_text(hjust = 0.5))

# penalized glm plot
ggplot(data = penalized_tune$results, aes(x = alpha, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Alpha in Penalized GLM") +
  theme(plot.title = element_text(hjust = 0.5))

ggplot(data = penalized_tune$results, aes(x = lambda, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Lambda in Penalized GLM") +
  theme(plot.title = element_text(hjust = 0.5))

# knn plot
ggplot(data = knn_tune$results, aes(x = k, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs k in kNN") +
```

```
  theme(plot.title = element_text(hjust = 0.5))

#nda plot
ggplot(data = nda_tune$results, aes(x = subclasses, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Subclasses in Nonlinear Discriminant Analysis") +
  theme(plot.title = element_text(hjust = 0.5))

#nnet plots
ggplot(data = nnet_tune$results, aes(x = decay, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Decay in Neural Network") +
  theme(plot.title = element_text(hjust = 0.5))

ggplot(data = nnet_tune$results, aes(x = size, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Size in Neural Network") +
  theme(plot.title = element_text(hjust = 0.5))

# fda plot
```

```
ggplot(data = fda_tune$results, aes(x = nprune, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Number of Components in Flexible Discriminant Analysis") +
  theme(plot.title = element_text(hjust = 0.5))

# svm plot
ggplot(data = svm_tune$results, aes(x = C, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Cost in Support Vector Machine") +
  theme(plot.title = element_text(hjust = 0.5))

ggplot(data = svm_tune$results, aes(x = sigma, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
  labs(title = "Kappa vs Sigma in Support Vector Machine") +
  theme(plot.title = element_text(hjust = 0.5))

# naive bayes plot
ggplot(data = naivebayes_tune$results, aes(x = usekernel, y = Kappa)) +
  geom_line(colour = "darkblue") +
  geom_point(colour = "darkblue") +
```

```
  labs(title = "Kappa vs Subclasses in Nonlinear Discriminant Analysis") +
  theme(plot.title = element_text(hjust = 0.5))


# confusion matrix for test set
log_confusion <- confusionMatrix(log_model_test, test_response)
lda_confusion <- confusionMatrix(lda_model_test, test_response)
plsda_confusion <- confusionMatrix(plsda_model_test, test_response)
glm_confusion <- confusionMatrix(glm_model_test, test_response)
knn_confusion <- confusionMatrix(knn_model_test, test_response)
nda_confusion <- confusionMatrix(nda_model_test, test_response)
nnet_confusion <- confusionMatrix(nnet_model_test, test_response)
fda_confusion <- confusionMatrix(fda_model_test, test_response)
svm_confusion <- confusionMatrix(svm_model_test, test_response)
nbayes_confusion <- confusionMatrix(naiveb_model_test, test_response)

log_confusion
lda_confusion
plsda_confusion
glm_confusion
knn_confusion
nda_confusion
nnet_confusion
```

```
fda_confusion

svm_confusion

nbayes_confusion


# fit best model




# important predictors for best model (GLM)


plot(varImp(penalized_tune, metric = "Kappa")) # importance plot
```