# ETC3250 Lab 10

*Di Cook*

*SOLUTION*

```
tr <- read_csv("../data/paintings_training_sub.csv")
ts <- read_csv("../data/paintings_test_sub.csv")
```

## Question 1

a. Build a linear discriminant analysis model to predict whether the painting is about flowers or cold theme.
b. Compute the error of the model for the test data.
c. Summarise the coefficients of the LDA classifier.

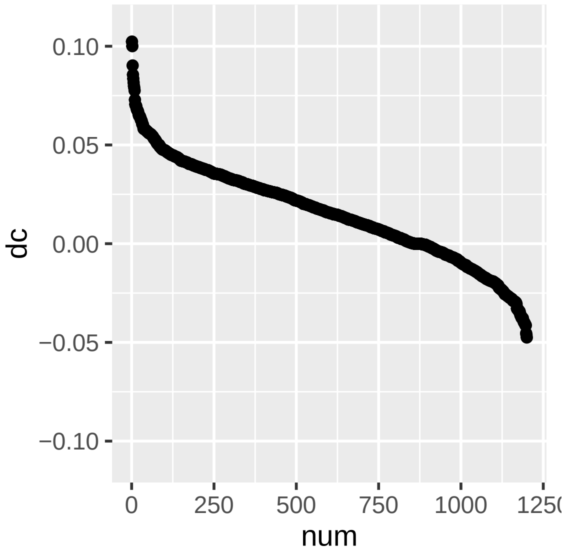
```
library(MASS)
tr_lda <- lda(class~., data=tr[,-c(1,2)], prior=c(0.5,0.5))
```

`You cannot run lda because it is not possible to estimate the covariance matrix. The function throws an error if you try to run it.`

## Question 2

a. Build a penalised linear discriminant analysis model to predict whether the painting is about flowers or cold theme.
b. Compute the error of the model for the test data. `Error is 0`
c. Summarise the coefficients of the Penalised LDA classifier. `Only 84 of the 1200 are bigger than 0.05, in magnitude. Most coefficients are between -0.05 and 0.05. This is a bit surprising, as we would expect a lot to be really at zero.`
d. Discuss how these differ fomr the LDA coefficients.`Can't do this becase the LDA model cannot be fitted.`

```
library(penalizedLDA)
cls <- ifelse(tr[,3]=="flowers", 2, 1)
set.seed(1)
tr_plda <- PenalizedLDA(as.matrix(tr[,-c(1:3)]), cls,
                        as.matrix(ts[,-c(1:3)]), lambda=0.001, K=1)
table(ts$class, tr_plda$ypred)
#
#           1 2
#   cold    7 0
#   flowers 0 5
length(tr_plda$discrim[tr_plda$discrim>0.05])
# [1] 84
df <- data.frame(num=1:1200, dc=sort(tr_plda$discrim, decreasing=TRUE))
ggplot(df, aes(x=num, y=dc)) + geom_point() + ylim(c(-0.11, 0.11))
```



## Question 3

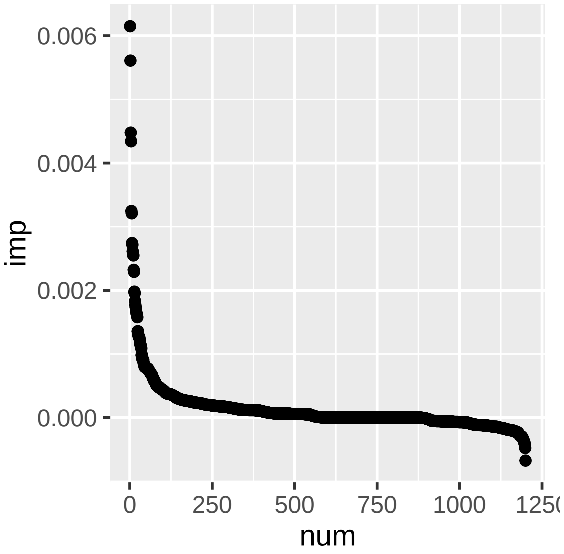
a. Build a support vector machine to predict whether the painting is about flowers or cold theme. `35 of the 45 observations are support vectors. Each has a really small coefficient. It would be interesting to look at the coefficients of the separating hyperplane - its possible that many have non-zero values, which would indicate that it has been affected by high-d low sample size.`
b. Compute the error of the model for the test data.`1/12=0.083`

```
library(e1071)
tr_svm <- svm(tr[,-c(1:3)], cls, kernel="linear")
psvm <- round(predict(tr_svm, ts[,-c(1:3)]), 0)
table(ts$class, psvm)
#          psvm
#           1 2
#   cold    7 0
#   flowers 1 4
tr_svm$index
#  [1]  5  7  9 10 11 12 13 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 31
# [24] 32 33 35 36 37 38 39 40 41 42 43 45
tr_svm$coefs
#                [,1]
#  [1,] -1.595726e-03
#  [2,] -1.117044e-03
#  [3,] -5.495303e-04
#  [4,] -2.315315e-03
#  [5,] -2.699798e-04
#  [6,] -6.993033e-04
#  [7,] -2.468889e-03
#  [8,] -2.800056e-04
#  [9,] -1.114680e-03
# [10,] -6.431594e-04
# [11,] -2.347893e-03
# [12,] -3.693279e-04
# [13,] -1.999740e-03
# [14,] -2.154866e-03
# [15,] -1.096979e-03
# [16,] -1.280236e-03
# [17,]  1.020084e-03
# [18,]  3.096888e-03
# [19,]  5.781053e-05
# [20,]  1.195442e-04
# [21,]  1.983927e-03
# [22,]  7.689723e-05
# [23,]  1.902882e-05
# [24,]  3.766256e-04
# [25,]  4.363271e-04
# [26,]  1.292646e-03
# [27,]  1.469611e-03
# [28,]  1.406445e-03
# [29,]  1.345161e-03
# [30,] -1.294380e-04
# [31,]  2.839299e-03
# [32,]  2.495546e-03
# [33,]  6.136562e-04
# [34,]  3.918769e-04
# [35,]  1.390740e-03
```

## Question 4

a. Build a random forest classifier to predict whether the painting is about flowers or cold theme.
b. Compute the error of the model for the test data.`0`
c. Compare the ten most important variables from random forest with that of penalizedLDA. Is there much overlap in the subset of variables?`Not a single common variable!`

```
library(randomForest)
tr$class <- factor(tr$class)
tr_rf <- randomForest(tr[,-c(1:3)], tr$class, ntree=1000, importance=TRUE)
prf <- predict(tr_rf, ts, type="class")
table(ts$class, prf)
#          prf
#           cold flowers
#   cold       7       0
#   flowers    0       5
df <- data.frame(num=1:1200, imp=sort(tr_rf$importance[,3], decreasing=TRUE))
ggplot(df, aes(x=num, y=imp)) + geom_point()
```



```
length(tr_rf$importance[tr_rf$importance[,3]>0.0007,3])
# [1] 62
tr_rf$importance[order(tr_rf$importance[,3])[1:10],]
#               cold       flowers MeanDecreaseAccuracy MeanDecreaseGini
# g170 -0.0007615440 -0.0006468254        -0.0006752137      0.024501197
# r190 -0.0004000000 -0.0005968254        -0.0004771930      0.010652684
# b151 -0.0003333333 -0.0005670996        -0.0004288515      0.027221195
# g158 -0.0001909091 -0.0006250000        -0.0003948413      0.015190476
# g280 -0.0005353535 -0.0002222222        -0.0003921569      0.008154762
# g66  -0.0003151515 -0.0004285714        -0.0003696970      0.021666052
# r234 -0.0004285714 -0.0002857143        -0.0003571429      0.001882353
# b197 -0.0003095238 -0.0003968254        -0.0003355263      0.018669586
# r2   -0.0004285714 -0.0002000000        -0.0003333333      0.005120000
# b243 -0.0002500000 -0.0004166667        -0.0003166667      0.007030303
names(tr[order(tr_plda$discrim, decreasing=TRUE)[1:10]])
#  [1] "b316" "b294" "b315" "b314" "b336" "g316" "g336" "g315" "g294" "b384"
```

## Question 5

a. Write a paragraph describing the xgboost algorithm, in your own words.

```
Boosting re-fits the classifier by re-weighting the observations. It is usually conducted
using tree classifiers. The predictions for each weighted tree are combined to give
final predictions for each class. XG is for extreme gradient boosting. This is a tweak
to the weight calculations to get to the minimum error quickly using a gradient descent
minimisation of the loss function.
```

b. Build an xgboost model to predict whether the painting is about flowers or cold theme.
c. Compute the error of the model for the test data.The arror varies a lot depending on the inputs. The lowest I got was 1/12=0.083.
d. Tweak the inputs to predict the test as best as you can.The best I got was using eta=0.5, nthread=20, nround=100.

```
library(xgboost)
dtrain <- xgb.DMatrix(as.matrix(tr[,-c(1:3)]), label = cls)
param <- list(max.depth = 2, eta = 0.5, silent = 1)
tr_xgb <- xgb.train(param, dtrain, nthread = 20, nround = 100)
pxgb <- round(predict(tr_xgb, as.matrix(ts[,-c(1:3)])), 0)
table(ts$class, pxgb)
#          pxgb
#           1 2
#   cold    6 1
#   flowers 0 5
```

## Question 6

Write a couple of paragraphs to compare and contrast the different classifiers for building a model on the paintings data.

This should be a discussion containing these pieces: LDA, we can't do. SVM doesn't need to estimate the variance-covariance matrix, so the model can be computed, but there are symptoms of an ill-specified model due to the many support vectors needed. The test error is low, though. This is similarly the case for penalizedLDA, the test error is perfect, but many variables have fairly high coefficients. XGBoost is fiddly to fit to get a low test error. Random forests is the easiest to explain, and has zero test error.

It should be noted that the training error for random forests is 0.24. That perhaps the test data was easy to perfectly separate was a lucky partition of the two groups. XGBoost is interesting, because the training error is 0, and the test error was low. It is difficult to tease apart the importance of variables for XGboost, which would involve examining the weights of each case.

There should also be a discussion on important variables. Particularly how there is no overlap in the top 10 for random forests and penalizedLDA. Because it is such a high-dimensional problem, it is likely that variables substitute for each other, that similar separation can be obtained with different subsets of variables.