Machine Learning Final Project

For this project, we use a subset of features from a famous olive oil dataset. The features I've chosen are not as predictive as the ones that I've excluded.

Features are acid measurements: palmitic, palmitol, stearic, linoleni, arachidi (you will see quite quickly that they vary somewhat in variability) These oils are classifiable into three macro-areas (variable name region): North, South, Sardinia

We have pre-split the data into: training (70%) and testing (30%). You will use the `caret` program to set tuning parameters.

SVM

Problem: Linear Kernel

We will compare and contrast svm fits using raw measures.

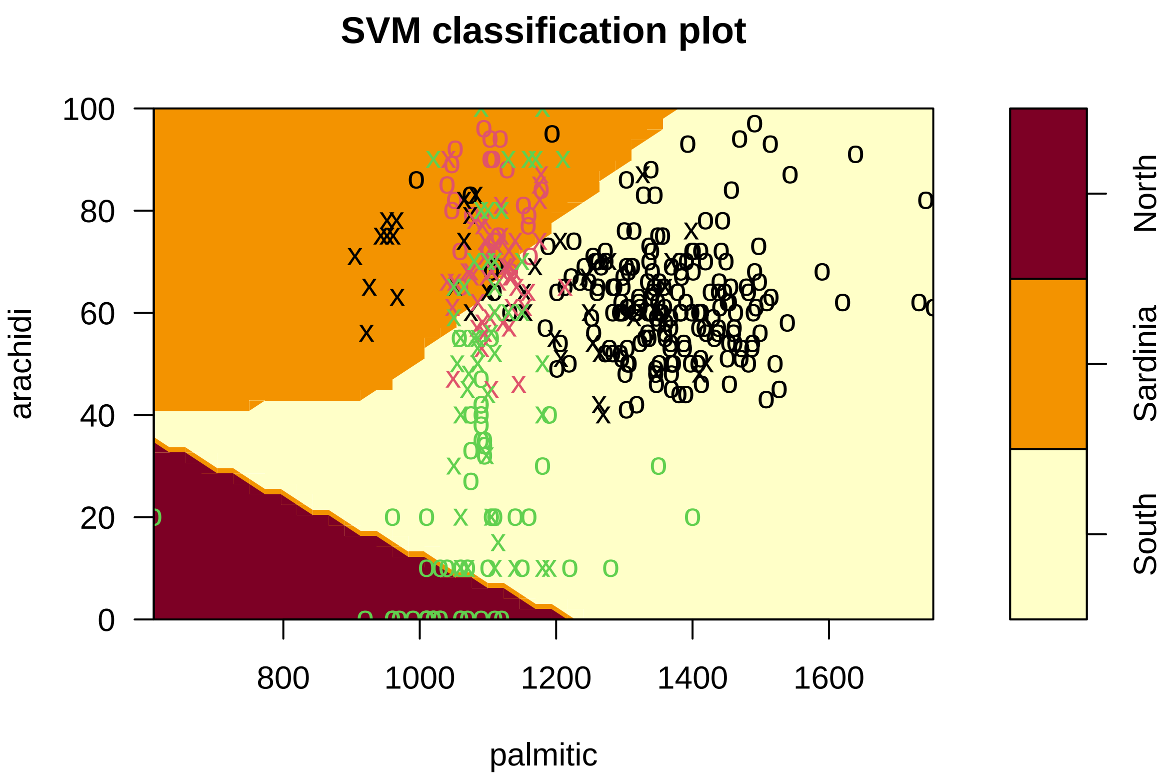
0. set.seed(2011001)
1. Run use the caret function to train a svm using 10-fold cross-validation, with the following formula: `region~.`, first with the linear kernel (in caret, use svmLinear2 method). Remember to specify 10-fold cv in the `trControl` (do not set repeats).
2. Refit the best model using `svm` so that it is easier to visualize
3. Visualize using plot on the final model chosen by each set of runs, using columns palmitic (x) and arachidi (y) for the plot, with slice set to the means of each of the remaining 3 features.
4. Predict on the TEST data. Examine accuracy and confusion matrices for this run.

```
# set seed
set.seed(2011001)

# train svm: 10-fold cross-validation
svm.linear.tc <- trainControl(method="cv",
                              number=10)

svm.linear <- train(region ~.,
              data = training,
              method = "svmLinear2",
              trControl = svm.linear.tc
)

# parameters of best model: 0.5
svm.linear$bestTune
```

```
##   cost
## 2  0.5
```

```
# refit the best model with svm
fit.lin <-svm(region~.,
              data=training,
              kernel="linear",
              cost = 0.50)
```

```r
# visualize plot on final model
plot(fit.lin,
     training,
     arachidi~palmitic,
     slice=list(stearic=mean(training$stearic),
                linoleni=mean(training$linoleni),
                palmitol=mean(training$palmitol)
                ))
```



```r
# predict on test data
svm.linear.pred <- predict(fit.lin, newdata = testing)

# confusion matrix
confusionMatrix(table(svm.linear.pred, testing$region))$table
```

```
##
## svm.linear.pred South Sardinia North
##        South       88        1     5
##        Sardinia     0       28    10
##        North        8        0    30
```

```r
# accuracy
confusionMatrix(table(svm.linear.pred, testing$region))$overall['Accuracy']
```

```
##  Accuracy
## 0.8588235
```

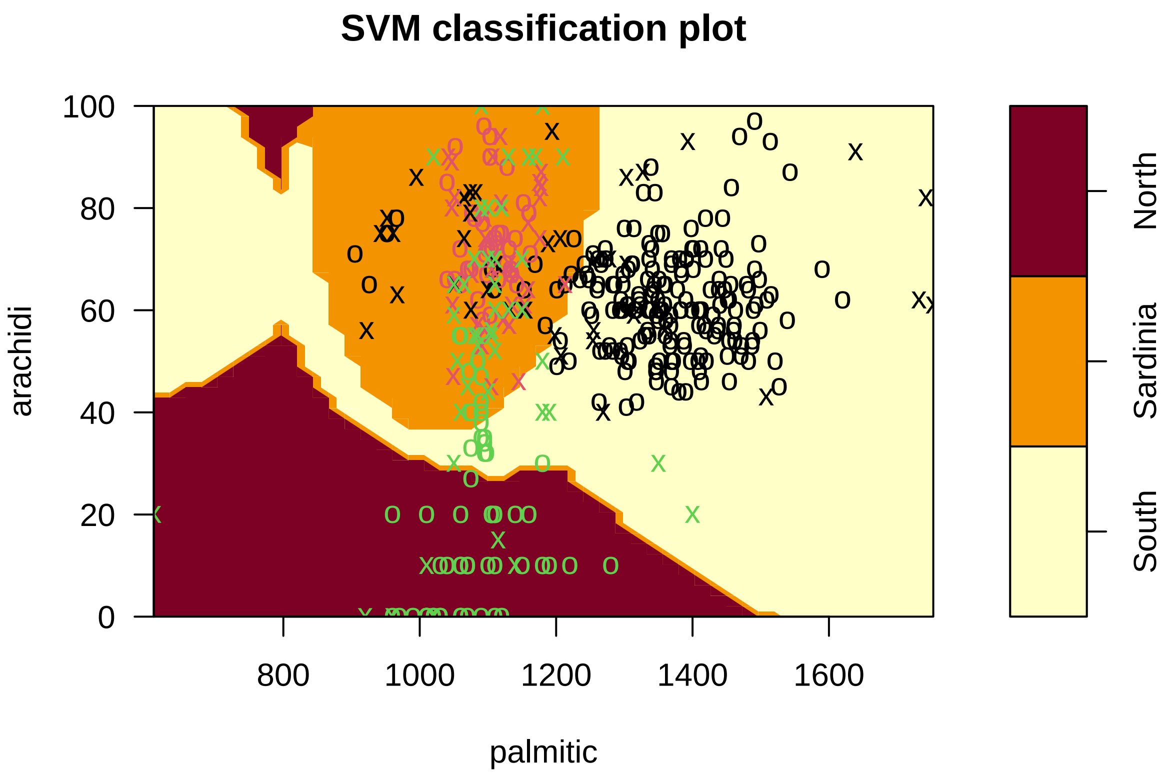
Problem: Radial Kernel

Repeat the 5 steps above with the radial kernel (caret's svmRadial method).

```r
# set seed
set.seed(2011001)

# train svm: 10-fold cross-validation
svm.radial.tc <- trainControl(method="cv",
                              number=10)

svm.radial <- train(region ~.,
              data = training,
              method = "svmRadial",
              trControl = svm.radial.tc
)

svm.radial$results
```

```
##       sigma    C  Accuracy     Kappa AccuracySD    KappaSD
## 1 0.3788049 0.25 0.9202220 0.8629769 0.02618833 0.04474769
## 2 0.3788049 0.50 0.9301001 0.8799133 0.02359113 0.03992319
## 3 0.3788049 1.00 0.9376001 0.8929265 0.02991909 0.05128058
```

```r
# refit the best model
fit.radial <-svm(region~.,
                data=training,
                kernel="radial",
                cost=1)

# visualize plot on final model
plot(fit.radial,
     training,
     arachidi~palmitic,
     slice=list(stearic=mean(training$stearic),
                linoleni=mean(training$linoleni),
                palmitol=mean(training$palmitol)
                ))
```



```r
# predict on test data
svm.radial.pred <- predict(fit.radial, newdata = testing)

# confusion matrix
confusionMatrix(table(svm.radial.pred, testing$region))$table
```

```
##
## svm.radial.pred South Sardinia North
##        South       92        0     5
##        Sardinia     1       27     7
##        North        3        2    33
```

```r
# accuracy
confusionMatrix(table(svm.radial.pred, testing$region))$overall['Accuracy']
```

```
##  Accuracy
## 0.8941176
```

Questions: all questions refer to performance on the test data

- Which kernel had better overall accuracy? **radial**
- Which region r is hardest to classify (using either kernel)? I.e., for which r is P(Prediction=r|Region=r) smallest (use TEST data results)? **North**
- Name there region that is better classified using the linear kernel? **Sardinia**
- The regions are different colored Xs and Os on the plot. Do they seem 'clumped' in a strongly curvilinear way? **Yes, the regions in the plot are separated in a manner that suggests curves would separate them better than lines, as the decision boundaries appear to be non-linear. The groups do not appear in clumps that look like rectangles, indicating a non-linear relationship between the variables.**

Regression Trees

Problem: rpart

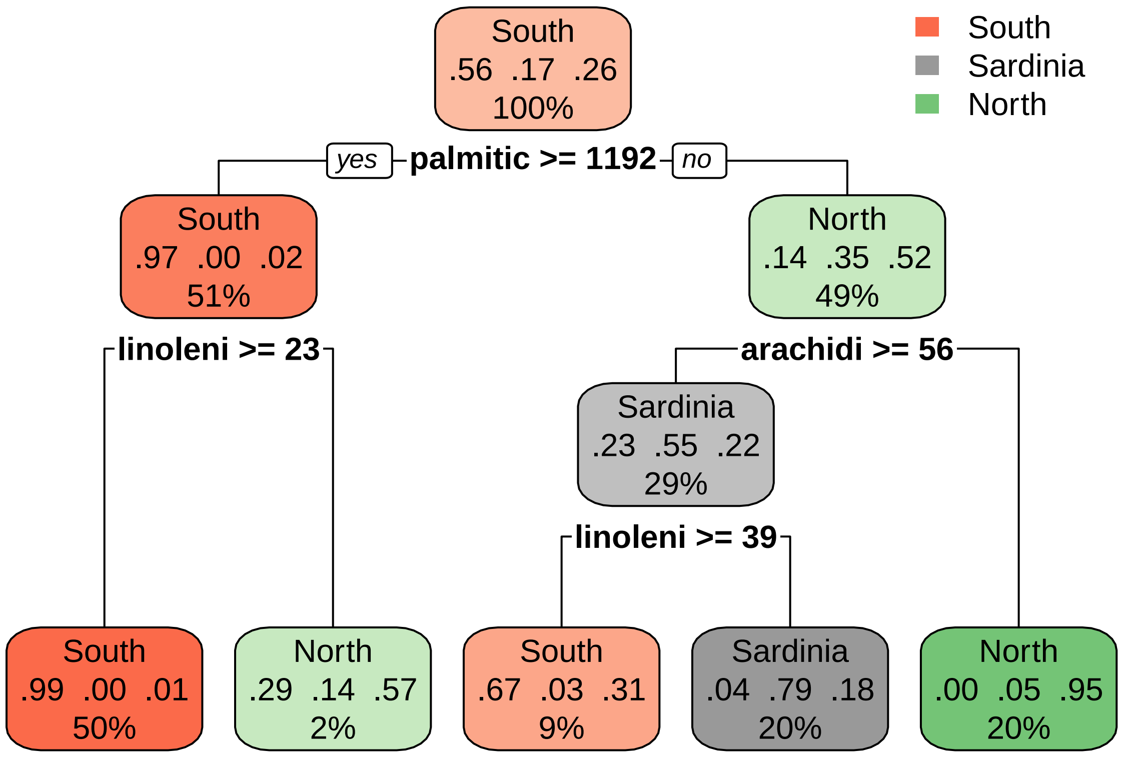
We will use `rpart` to build a regression tree for this classification problem. Later, we will use random forests.

0. set.seed(2011001)
1. Run use the caret function to train rpart with the following formula: `region~.`. In caret, use the rpart2 method. Remember to specify 10-fold cv in the `trControl` (do not set repeats).
2. You will find the best model is the list element `finalModel` in the result of train.
3. Visualize this best model using `rpart.plot`. Be prepared to describe this model in the questions that follow.
4. Predict on the TEST data. Examine accuracy and confusion matrices for this run.

```r
# step 0: set seed
set.seed(2011001)

# step 1: train regression tree model using caret
fit3 <- train(region ~ ., data = training, method = "rpart2",
                 trControl = trainControl(method = "cv", number = 10))
fit3
```

```
## CART
##
## 402 samples
##   5 predictor
##   3 classes: 'South', 'Sardinia', 'North'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 362, 361, 363, 363, 361, 362, ...
## Resampling results across tuning parameters:
##
##   maxdepth  Accuracy   Kappa
##   1         0.7462179  0.5626182
##   2         0.8406270  0.7378988
##   3         0.8952095  0.8160154
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was maxdepth = 3.
```

```r
# step 2: extract best model
best_model3 <- fit3$finalModel

# step 3: visualize best model
rpart.plot(best_model3)
```



```r
# step 4: predict on TEST data
predict3 <- predict(best_model3, newdata = testing, type = "class")

confusionMatrix(predict3, testing$region)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction South Sardinia North
##   South       87        2     6
##   Sardinia     4       27     9
##   North        5        0    30
##
## Overall Statistics
##
##                Accuracy : 0.8471
##                  95% CI : (0.784, 0.8976)
##     No Information Rate : 0.5647
##     P-Value [Acc > NIR] : 3.125e-15
##
##                   Kappa : 0.7407
##
##  Mcnemar's Test P-Value : 0.02074
##
## Statistics by Class:
##
##                      Class: South Class: Sardinia Class: North
## Sensitivity                0.9062          0.9310       0.6667
## Specificity                0.8919          0.9078       0.9600
## Pos Pred Value             0.9158          0.6750       0.8571
## Neg Pred Value              0.8800          0.9846       0.8889
## Prevalence                  0.5647          0.1706       0.2647
## Detection Rate              0.5118          0.1588       0.1765
## Detection Prevalence        0.5588          0.2353       0.2059
## Balanced Accuracy           0.8991          0.9194       0.8133
```

Questions:

- Which region was hardest to classify? I.e., for which r is P(Prediction=r|Region=r) smallest (use TEST data results)? **The North region was the hardest to classify. This is evidenced by its lowest sensitivity score, which is a measure of how well the classifier is able to correctly identify positive cases. The sensitivity score is determined by the ratio of true positives to true positives plus false negatives. A low sensitivity score suggests that the classifier had difficulty accurately identifying oil from the North region.**
- Our classifier tells us our oil is from region r. For which region r will we be least sure that the oil is truly from that region (use test data performance)? **The region that we will be least sure that the oil is truly from is Sardinia. Sardinia region has the lowest Positive Predictive Value (PPV). PPV is a measure of how often the classifier is correct when it predicts a positive outcome. It is calculated as the ratio of true positive predictions to the total number of positive predictions made by the classifier, and a lower PPV value indicates that the classifier is less accurate in identifying oil from the Sardinia region.**
- Which acid seems the most important (predictive) feature (look at tree from training)? **palmitic**
- Given palmitic=1000, palmitol=100, stearic=200, linoleni=40, arachidi=60, what region would you say the oil is from? **South**
- What is the probability that you are wrong about your answer to the last question (based on the tree and training data)? **0.34**

Problem: randomForest

We will use `randomForest` to build a regression tree for this classification problem.

0. set.seed(2011001)
1. Run use the caret function to train rpart with the following formula: `region~.`. In caret, use the rf method. Remember to specify 10-fold cv in the `trControl` (do not set repeats). Be sure to set `importance=TRUE` so that you can do (3), below. THIS MAY BE A BIT SLOW. BE PATIENT.
2. You will find the best model is the list element `finalModel` in the result of train.
3. `treesize` is a function that takes a randomForest object and return the number of nodes in each tree in the ensemble. Compute the average tree size.
4. Use the `varImpPlot` function to determine which variables are most predictive of each of the three regions (so three plots). Use parameter `type=1`, which gives the Mean Decrease Accuracy (%). Be prepared to discuss.
5. Predict on the TEST data. Examine accuracy and confusion matrices for this run.

```r
# step 0: set seed
set.seed(2011001)

# step 1: train random forest model using caret
fit4 <- train(region ~ ., data = training, method = "rf",
              trControl = trainControl(method = "cv", number = 10), importance = TRUE)
fit4
```

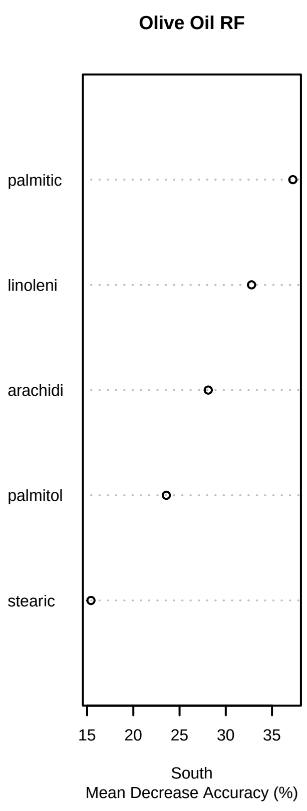
```
## Random Forest
##
## 402 samples
##   5 predictor
##   3 classes: 'South', 'Sardinia', 'North'
```

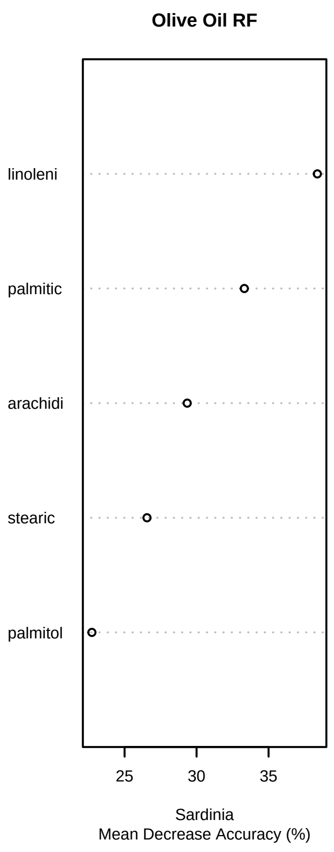
```
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 362, 361, 363, 363, 361, 362, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9647983  0.9400497
##   3     0.9599203  0.9315611
##   5     0.9400328  0.8984753
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

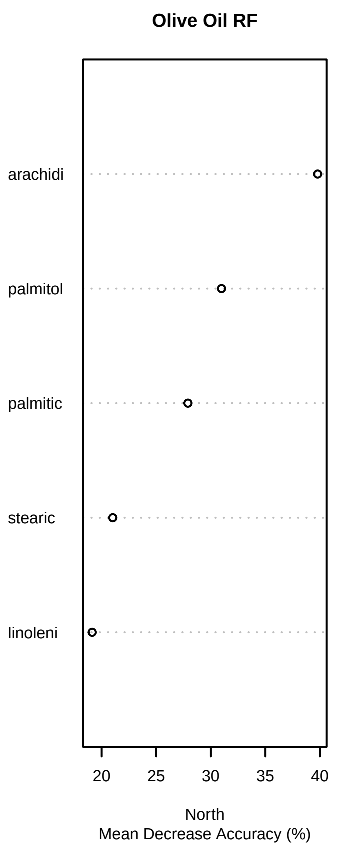
```
# step 2: extract best model
best_model4 <- fit4$finalModel

# step 3: compute average tree size
treesize(best_model4) %>%
        mean()
```

```
## [1] 27.724
```

```
# step 4: determine most predictive variables
par(mfrow=c(1,3))
for (i in 1:3) {
  varImpPlot(best_model4, type = 1, class = best_model4$classes[i],
             main = "Olive Oil RF", sub = "Mean Decrease Accuracy (%)",
             cex = .5)
}
```







```
# step 5: predict on TEST data
predict4 <- predict(best_model4, newdata = testing)
confusionMatrix(predict4, testing$region)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction South Sardinia North
##   South       94        1     3
##   Sardinia     1       27     4
##   North        1        1    38
##
## Overall Statistics
##
##                Accuracy : 0.9353
##                  95% CI : (0.8872, 0.9673)
##     No Information Rate : 0.5647
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8885
##
##  Mcnemar's Test P-Value : 0.4235
##
## Statistics by Class:
##
##                      Class: South Class: Sardinia Class: North
## Sensitivity                0.9792          0.9310       0.8444
## Specificity                  0.9459           0.9645       0.9840
## Pos Pred Value               0.9592           0.8437       0.9500
## Neg Pred Value               0.9722           0.9855       0.9462
## Prevalence                   0.5647           0.1706       0.2647
## Detection Rate               0.5529           0.1588       0.2235
## Detection Prevalence         0.5765           0.1882       0.2353
## Balanced Accuracy            0.9626           0.9478       0.9142
```

Questions:

- What is the overall accuracy of this method, and how much larger is it than that for rpart? **The overall accuracy of this method is 0.9353. This method's accuracy is 0.0882 larger than that of rpart, which is 0.8471.**
- Based on the plots, performance tables and other calculations, what accounts for this great improvement? **The random forest model performs better than the rpart model because it combines the predictions of multiple decision trees, which reduces overfitting and improves overall accuracy. Additionally, random forest also has a built-in feature selection, which can further improve performance by identifying and using only the most important features. This can be seen by comparing the performance of the two models, as seen in the confusion matrix where the random forest model successfully predicts 94 cases in the South region, while the rpart model successfully predicts 87 cases in the South region. Additionally, for North region, the random forest model successfully predicts 38 cases, while the rpart model predicts 30 cases.**
- Which acid seems the most important for South (look at importance plots)? **palmitic**
- Is the most important feature (for two of the three regions) the one deemed most predictive by rpart? **For RF, the palmitic acid appears to be the most important feature for the South, linolenic acid for Sardinia, and arachidic acid for the North. Therefore, for the South region, the palmitic feature matches the feature used on the first cut point for rpart.**
- In this random forests approach, is there a single tree that you can plug the values: palmitic=1000, palmitol=100, stearic=200, linoleni=40, arachidi=60 into to predict region? If not, how does the prediction get made? **No, random forest uses an ensemble of decision trees to make predictions. The algorithm applies the input values to each of the decision trees in the ensemble and aggregates the predictions by taking the majority vote.**
- What is the average number of nodes in the trees in your best model's ensemble of trees? **27.724**