Assignment 4

In this assignment, you will have the chance to practice building a Support Vector Machine models. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.

Below is an example of how a linear SVM model can be trained using the caret package. In this example, we use “PimaIndiansDiabetes” dataset that is part of the mlbench library. This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

The dataset consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

Pregnancies:Number of times pregnant

Glucose:Plasma glucose concentration a 2 hours in an oral glucose tolerance test

BloodPressure:Diastolic blood pressure (mm Hg)

SkinThickness: Triceps skin fold thickness (mm)

Insulin: 2-Hour serum insulin (mu U/ml)

BMI: Body mass index (weight in kg/(height in m)^2)

Age:Age (years)

Diabetes : The target (outcome) variable with positive and negative levels.

library(mlbench)

## Warning: package 'mlbench' was built under R version 3.4.4

#install using the following command if the library is not installed  
#install.packages('mlbench')  
library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.4

# This will load the PimaIndiansDiabetes dataset  
data(PimaIndiansDiabetes)  
  
#check the levels  
levels(PimaIndiansDiabetes$diabetes)

## [1] "neg" "pos"

set.seed(2)  
intrain <- createDataPartition(y = PimaIndiansDiabetes$diabetes, p= 0.7, list = FALSE)  
training <- PimaIndiansDiabetes[intrain,]  
testing <- PimaIndiansDiabetes[-intrain,]  
  
svm.model<-train(diabetes~.,data=training,method='svmLinear')  
svm.model

## Support Vector Machines with Linear Kernel   
##   
## 538 samples  
## 8 predictor  
## 2 classes: 'neg', 'pos'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 538, 538, 538, 538, 538, 538, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.7578508 0.4412716  
##   
## Tuning parameter 'C' was held constant at a value of 1

# You can define the exact values of mtry parameters to be tested by caret  
#Lets say we want to check 1, 2,3 and 9 as values for C  
# Forgot what C was? it's the cost parameter of the SVM algorithm   
  
Grid\_Serach <- expand.grid(.C=c(1,2,3,9))  
  
svm.model2<-train(diabetes~.,  
 data=training,  
 method='svmLinear',  
 tuneGrid=Grid\_Serach)  
  
svm.model2

## Support Vector Machines with Linear Kernel   
##   
## 538 samples  
## 8 predictor  
## 2 classes: 'neg', 'pos'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 538, 538, 538, 538, 538, 538, ...   
## Resampling results across tuning parameters:  
##   
## C Accuracy Kappa   
## 1 0.7579101 0.4387906  
## 2 0.7583099 0.4399528  
## 3 0.7581202 0.4393908  
## 9 0.7579151 0.4389981  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was C = 2.

pred\_class <-predict(svm.model2,testing,type="raw")  
head(pred\_class)

## [1] neg pos neg pos neg pos  
## Levels: neg pos

**Your Task**

Your task is to use the above example to build a random forest classification model to predict the type of a breast tumor (benign or malignant).

The data can be loaded using the mlbench library using

library(mlbench)  
data(BreastCancer)

A data frame with 699 observations on 11 variables, one being a character variable, 9 being ordered or nominal, and 1 target class.

[,1] Id Sample code number

[,2] Cl.thickness Clump Thickness

[,3] Cell.size Uniformity of Cell Size

[,4] Cell.shape Uniformity of Cell Shape

[,5] Marg.adhesion Marginal Adhesion

[,6] Epith.c.size Single Epithelial Cell Size

[,7] Bare.nuclei Bare Nuclei

[,8] Bl.cromatin Bland Chromatin

[,9] Normal.nucleoli Normal Nucleoli

[,10] Mitoses Mitoses

[,11] Class Class

Submission:

Submit a pdf of your r markdown with the results shown in the file. Similar to my example above (you can alternatively include screen shots- however your submission should be one single pdf). In your submission:

1. Try examine the following values for the C parameter 1,2.5, 6.25

2.5 was the best value of C

({'C': 2.5}, 0.954455443247712)

1. Show the confusion matrix of predictions on the test data (in the above example, I have the raw (class) predictions. You need to compare them with the ground truth values (i.e. testing$Class). Hint: table( ) function

