Homework 1

Muh Alif Ahsanul Islam

5/18/2019

Question 2.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a classification model would be appropriate. List some (up to 5) predictors that you might use.

Answer:

Classifying manufacturing product into acceptable and not-acceptable.
After a product is produced it is necessary to do quality control process. Products that has defects, or do not meet requirements must be inspected. To automate the process machine learning models can be used.
Predictors:
1. Geometrical shape
2. Color
3. Mass

Question 2.2

The files credit_card_data.txt (without headers) and credit_card_data-headers.txt (with headers) contain a dataset with 654 data points, 6 continuous and 4 binary predictor variables. It has anonymized credit card applications with a binary response variable (last column) indicating if the application was positive or negative. The dataset is the “Credit Approval Data Set” from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Credit+Approval) without the categorical variables and without data points that have missing values.

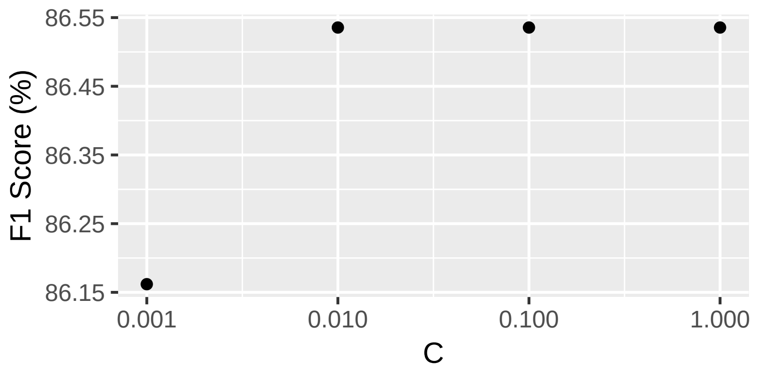
Question 2.2.1

Using the support vector machine function ksvm contained in the R package kernlab, find a good classifier for this data. Show the equation of your classifier, and how well it classifies the data points in the full data set.

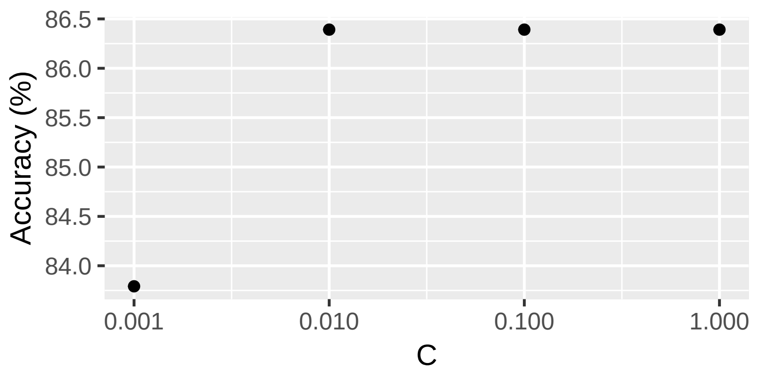
```r
setwd('/home/alifahsanul/Documents/analytics_modelling')
data = read.table('./hw1/week_1_data-summer/data 2.2/credit_card_data-headers.txt',
                  header=TRUE)
y_col_name = 'R1'
cols = c(y_col_name)
data[cols] = lapply(data[cols], factor)
y_col_ind = grep(y_col_name, colnames(data))
X = data[,-y_col_ind]
y = data[,y_col_ind]
X = as.matrix(X)
y = (y)
c_params = c(0.001, 0.01, 0.1, 1, 10, 100, 1000)
c_params = c(0.001, 0.01, 0.1, 1)
accuracy_list = c()
model_list = c()
conf_matrix_list = c()
f1_score_list = c()
for (c_i in c_params){
  cat(sprintf('\n---------------------------\n'))
  cat(sprintf('C: %.6f\n', c_i))
  model = ksvm(X, y, type='C-svc', kernel='vanilladot', C=c_i, scaled=TRUE)
  y_pred = predict(model, X)
  conf_matr = confusionMatrix(y_pred, y)
  accuracy = conf_matr[['overall']][['Accuracy']] * 100
  f1_score = conf_matr[['byClass']][['F1']] * 100
  cat(sprintf('Accuracy: %.2f %%\tF1 Score: %.2f %%\n', accuracy, f1_score))
  model_list = c(model_list, model)
  accuracy_list = c(accuracy_list, accuracy)
  f1_score_list = c(f1_score_list, f1_score)
  conf_matrix_list = c(conf_matrix_list, conf_matr)
}
```

```
##
## ---------------------------
## C: 0.001000
##  Setting default kernel parameters
## Accuracy: 83.79 %    F1 Score: 86.16 %
##
## ---------------------------
## C: 0.010000
##  Setting default kernel parameters
## Accuracy: 86.39 %    F1 Score: 86.54 %
##
## ---------------------------
## C: 0.100000
##  Setting default kernel parameters
## Accuracy: 86.39 %    F1 Score: 86.54 %
##
## ---------------------------
## C: 1.000000
##  Setting default kernel parameters
## Accuracy: 86.39 %    F1 Score: 86.54 %
```

```
qplot(c_params, f1_score_list, log='x') +
  labs(x='C', y='F1 Score (%)')
```



```
qplot(c_params, accuracy_list, log='x') +
  labs(x='C', y='Accuracy (%)')
```



```r
best_model_ind = which.max(f1_score_list)
best_model = model_list[[best_model_ind]]
best_c_params = c_params[best_model_ind]
best_f1_score = f1_score_list[best_model_ind]
cat(sprintf('Best model with higest F1 Score of %.2f%% is model with C = %.6f\n',
            best_f1_score, best_c_params))
```

```
## Best model with higest F1 Score of 86.54% is model with C = 0.010000
```

```r
a = colSums(best_model@xmatrix[[1]] * best_model@coef[[1]])
a0 = -best_model@b
model_col_name = names(a)
equation = ''
for (i in seq_along(model_col_name)){
  coef = a[i]
  name = model_col_name[i]
  equation = paste(equation, sprintf('%.2e * %s + ', coef, name), sep='')
}
equation = paste0(paste(equation, sprintf('%.2e = 0',a0)))
equation = break_string(equation, '+', 70)
cat(sprintf('Best model equation is:\n%s\n', equation))
```

```
## Best model equation is:
## -1.50e-04 * A1  + -1.48e-03 * A2 + 1.41e-03 * A3 + 7.29e-03 * A8 +
##  9.92e-01 * A9 + -4.47e-03 * A10 + 7.15e-03 * A11 + -5.47e-04 * A12 + -1.69e-03 * A14 +
##  1.05e-01 * A15 +  8.20e-02 = 0
```