CS6301.012 Advance Computational Methods for Data Science

Assignment 10

Data Set Used:
LMR_LEVEE

Team Members:

Aditya Mahajan Net Id: axm156630
Arnav Sharma Net Id: axs144130
Divyanshu Paliwal Net Id: dxp151630
Nipun Agarwal Net Id: nxa150830

LMR LEVEE DATASET

```
> library(e1071)
> data<- mmr_levee
> names(data) <- c("Fail","Year", "River","Sedi","Borrow", "Meander","Channel
","Floodway","Cons", "Land", "Veg","Sinu","Dred", "Reve")
> str(data)
'data.frame':  70 obs. of  14 variables:
 $ Fail    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Year    : int  1880 1908 1908 1908 1908 1908 1908 1948 1948 1948 ...
 $ River   : num  188 190 174 147 143 ...
 $ Sedi    : int  0 1 0 1 0 0 0 0 0 1 ...
 $ Borrow  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Meander : int  2 1 1 1 4 4 1 1 1 3 ...
 $ Channel : num  2513 1271 920 1115 1032 ...
 $ Floodway: num  6991 4344 3396 2105 2513 ...
 $ Cons    : num  1 3.058 1.001 0.949 0.958 ...
 $ Land    : int  1 3 3 1 4 3 2 4 4 4 ...
 $ Veg     : num  0 183 2411 0 305 ...
 $ Sinu    : num  1.231 1.24 0.994 1.107 0.997 ...
 $ Dred    : int  0 0 19354 34968 46540 0 0 218751 218751 0 ...
 $ Reve    : int  0 0 0 0 0 0 0 0 0 0 ...
> summary(data)
      Fail             Year          River              Sedi             Borrow
       Meander          Channel
 Min.   :0.0   Min.   :1880   Min.   :  2.70   Min.   :0.0000   Min.   :0.000
0   Min.   :1.000   Min.   :   7.78
 1st Qu.:0.0   1st Qu.:1948   1st Qu.: 35.25   1st Qu.:0.0000   1st Qu.:0.000
0   1st Qu.:1.000   1st Qu.: 783.95
 Median :0.5   Median :1948   Median : 99.92   Median :1.0000   Median :0.000
0   Median :3.000   Median :1006.59
 Mean   :0.5   Mean   :1952   Mean   : 94.08   Mean   :0.5571   Mean   :0.157
1   Mean   :2.643   Mean   :1011.96
 3rd Qu.:1.0   3rd Qu.:1986   3rd Qu.:146.90   3rd Qu.:1.0000   3rd Qu.:0.000
0   3rd Qu.:4.000   3rd Qu.:1157.48
 Max.   :1.0   Max.   :1998   Max.   :190.00   Max.   :1.0000   Max.   :1.000
0   Max.   :4.000   Max.   :2512.91
    Floodway           Cons             Land            Veg              Sinu
           Dred             Reve
 Min.   : 134.8   Min.   :0.0778   Min.   :1.000   Min.   :   0.0   Min.   :0
.8944   Min.   :     0   Min.   :0.00000
 1st Qu.:2158.3   1st Qu.:0.2320   1st Qu.:3.000   1st Qu.: 186.3   1st Qu.:1
.0135   1st Qu.:     0   1st Qu.:0.00000
 Median :2512.1   Median :0.4622   Median :4.000   Median : 464.5   Median :1
.1103   Median : 34398   Median :0.00000
 Mean   :2847.0   Mean   :0.6556   Mean   :3.343   Mean   : 776.4   Mean   :1
.2073   Mean   : 96618   Mean   :0.01429
 3rd Qu.:3395.1   3rd Qu.:0.9669   3rd Qu.:4.000   3rd Qu.:1000.9   3rd Qu.:1
.2182   3rd Qu.:121606   3rd Qu.:0.00000
 Max.   :7030.7   Max.   :3.0576   Max.   :4.000   Max.   :2993.4   Max.   :2
.5775   Max.   :872528   Max.   :1.00000
> set.seed(1)

#removing revetement variable as it is 0 for all but 1 observations
> data= data[, -c(14)]
> data= data[, -c(1)]
```

```
> x=data

#creating the data frame with y as response variable
> dat=data.frame(x=x, y=as.factor(y))
> smp_size <- floor(0.6 * nrow(dat))
> train_ind=sample(seq_len(nrow(dat)), size = smp_size)
> data.train <- dat[train_ind, ]
> data.test <- dat[-train_ind, ]
#fitting the support vector classifier
> svmfit=svm(y~., data=data.train, kernel="linear", cost=10,scale=FALSE)

WARNING: reaching max number of iterations

#this does not work as we have more  than 2 predictors
> plot(svmfit, dat)
Error in plot.svm(svmfit, dat) : missing formula.

#the support vectors are found using index on the fit
> svmfit$index
 [1] 11 12 27 35  6  9 13 17 18 29 30 40

> summary(svmfit)

Call:
svm(formula = y ~ ., data = data.train, kernel = "linear", cost = 10, scale =
 FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  10
      gamma:  0.08333333

Number of Support Vectors:  12

 ( 4 8 )


Number of Classes:  2

Levels:
 0 1

> svmfit=svm(y~., data=data.train, kernel="linear", cost=0.1,scale=FALSE)

WARNING: reaching max number of iterations
> svmfit$index
 [1] 11 12 19 23 27 35  6  9 13 29 30 40

#using 10 fold cross validation to find the best value for cost
> tune.out=tune(svm,y~.,data=dat,kernel="linear",ranges=list(cost=c(0.001, 0.
01, 0.1, 1,5,10,100)))
> summary(tune.out)

Parameter tuning of ‘svm’:
```

```
- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.3714286

- Detailed performance results:
   cost     error dispersion
1 1e-03 0.6571429 0.09988656
2 1e-02 0.6142857 0.15133570
3 1e-01 0.3714286 0.16768397
4 1e+00 0.4285714 0.15058465
5 5e+00 0.4142857 0.15721499
6 1e+01 0.4142857 0.15721499
7 1e+02 0.4142857 0.15721499

> bestmod=tune.out$best.model
> summary(bestmod)

Call:
best.tune(method = svm, train.x = y ~ ., data = dat, ranges = list(cost = c(0
.001, 0.01, 0.1, 1, 5, 10, 100)),
    kernel = "linear")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.1
      gamma:  0.08333333

Number of Support Vectors:  56

 ( 27 29 )


Number of Classes:  2

Levels:
 0 1

#using the model to predict class for the test dataset
> ypred=predict(bestmod,data.test)
> table(predict=ypred, truth=data.test$y)
       truth
predict 0 1
      0 5 7
      1 7 9
```

The error in classification using the best model by using support vector classifier techniques is 50% which suggests it might not be the best of the method to classify this dataset.