Anly501 NB and SVM

Yangyi Li

```
# we set align to be the label
# we want to see if any other elements affect the align
marvel <- read.csv("marvel.csv")
names(marvel) <- tolower(names(marvel))

head(marvel)
```

```
##   page_id                                    name
## 1    1678             spider-man (peter parker)
## 2    7139       captain america (steven rogers)
## 3   64786 wolverine (james \\"logan\\" howlett)
## 4    1868   iron man (anthony \\"tony\\" stark)
## 5    2460                    thor (thor odinson)
## 6    2458            benjamin grimm (earth-616)
##                                   urlslug               id              align
## 1             \\/spider-man_(peter_parker)  secret identity    good characters
## 2       \\/captain_america_(steven_rogers)  public identity    good characters
## 3 \\/wolverine_(james_%22logan%22_howlett)  public identity neutral characters
## 4   \\/iron_man_(anthony_%22tony%22_stark)  public identity    good characters
## 5                   \\/thor_(thor_odinson) no dual identity    good characters
## 6            \\/benjamin_grimm_(earth-616)  public identity    good characters
##          eye       hair             sex gsm             alive appearances
## 1 hazel eyes brown hair male characters     living characters        4043
## 2  blue eyes white hair male characters     living characters        3360
## 3  blue eyes black hair male characters     living characters        3061
## 4  blue eyes black hair male characters     living characters        2961
## 5  blue eyes blond hair male characters     living characters        2258
## 6  blue eyes    no hair male characters     living characters        2255
##   first.appearance year
## 1           aug-62 1962
## 2           mar-41 1941
## 3           oct-74 1974
## 4           mar-63 1963
## 5           nov-50 1950
## 6           nov-61 1961
```

```
marvel <- marvel %>% select(id, align, sex, alive, appearances, year)
marvel <- marvel %>% select(align, everything())
marvel <- subset(marvel, align!="")
marvel <- subset(marvel, id!="")
marvel <- subset(marvel, sex!="")
marvel <- marvel %>% filter(sex == "male characters"|
           sex == "female characters")
marvel <- marvel %>% drop_na()
colnames(marvel)[1] <- "label"
marvel <- marvel[1:1000,]
head(marvel)
```

```
##                label               id             sex              alive
## 1    good characters  secret identity male characters living characters
## 2    good characters  public identity male characters living characters
## 3 neutral characters  public identity male characters living characters
## 4    good characters  public identity male characters living characters
## 5    good characters no dual identity male characters living characters
## 6    good characters  public identity male characters living characters
##   appearances year
## 1        4043 1962
## 2        3360 1941
## 3        3061 1974
## 4        2961 1963
## 5        2258 1950
## 6        2255 1961
```

```
str(marvel)
```

```
## 'data.frame':    1000 obs. of  6 variables:
##  $ label      : chr  "good characters" "good characters" "neutral characters" "good c
haracters" ...
##  $ id         : chr  "secret identity" "public identity" "public identity" "public id
entity" ...
##  $ sex        : chr  "male characters" "male characters" "male characters" "male char
acters" ...
##  $ alive      : chr  "living characters" "living characters" "living characters" "liv
ing characters" ...
##  $ appearances: int  4043 3360 3061 2961 2258 2255 2072 2017 1955 1934 ...
##  $ year       : int  1962 1941 1974 1963 1950 1961 1961 1962 1963 1961 ...
```

```
## If necessary - correct data types
marvel$label <- as.factor(marvel$label)
marvel$id <- as.factor(marvel$id)
marvel$sex <- as.factor(marvel$sex)
marvel$alive <- as.factor(marvel$alive)
```

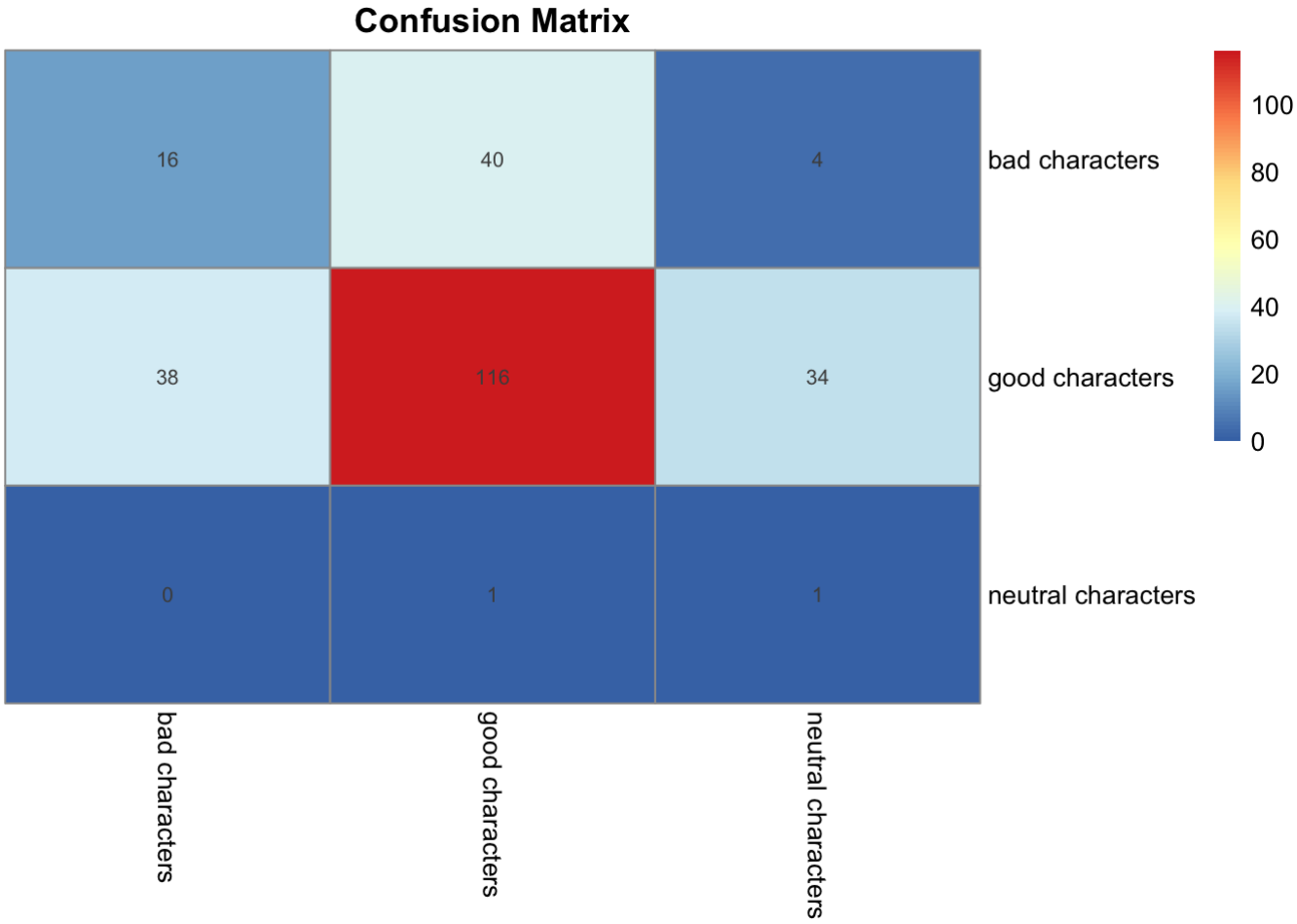
```
DataSize=nrow(marvel)
TrainingSet_Size<-floor(DataSize*(3/4))
TestSet_Size <- DataSize - TrainingSet_Size

MyTrainSample <- sample(nrow(marvel),TrainingSet_Size,replace=FALSE)
MyTrainingSET <- marvel[MyTrainSample,]
MyTestSET <- marvel[-MyTrainSample,]

train_label <- MyTrainingSET$label
test_label <- MyTestSET$label

old_set <- MyTrainingSET
MyTrainingSET<-MyTrainingSET[ , -which(names(MyTrainingSET) %in% c("label"))]
head(MyTrainingSET)
```

```
##                    id               sex               alive appearances year
## 784  public identity female characters deceased characters          43 1964
## 572  public identity   male characters   living characters          63 1964
## 825  secret identity   male characters deceased characters          41 1970
## 124 no dual identity   male characters   living characters         296 1973
## 267  secret identity   male characters   living characters         147 1966
## 284 no dual identity female characters   living characters         140 1977
```

```
SVM_fit_P <- svm(label~., data=old_set,
               kernel="polynomial", cost=.05,
               scale=FALSE)
print(SVM_fit_P)
```

```
##
## Call:
## svm(formula = label ~ ., data = old_set, kernel = "polynomial", cost = 0.05,
##     scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  0.05
##      degree:  3
##      coef.0:  0
##
## Number of Support Vectors:  441
```

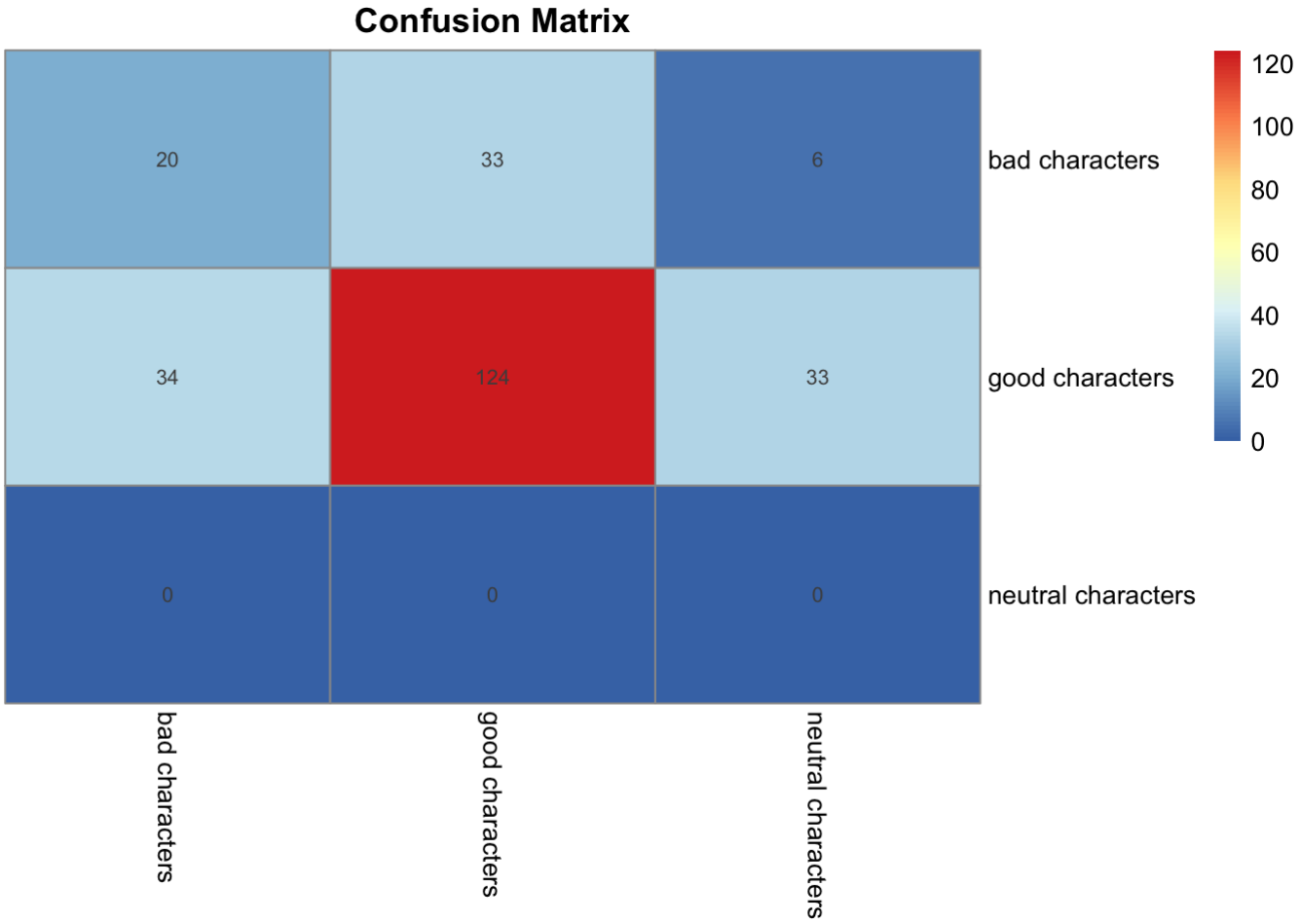
```
pred_P <- predict(SVM_fit_P, MyTestSET, type="class")
Ptable <- table(pred_P, test_label)
pheatmap(Ptable,cluster_cols=F,cluster_rows=F,display_numbers=T,number_format = "%.f",ma
in='Confusion Matrix')
```




```
SVM_fit_P <- svm(label~., data=old_set,
                 kernel="linear", cost=.1,
                 scale=FALSE)
print(SVM_fit_P)
```

```
##
## Call:
## svm(formula = label ~ ., data = old_set, kernel = "linear", cost = 0.1,
##     scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  598
```

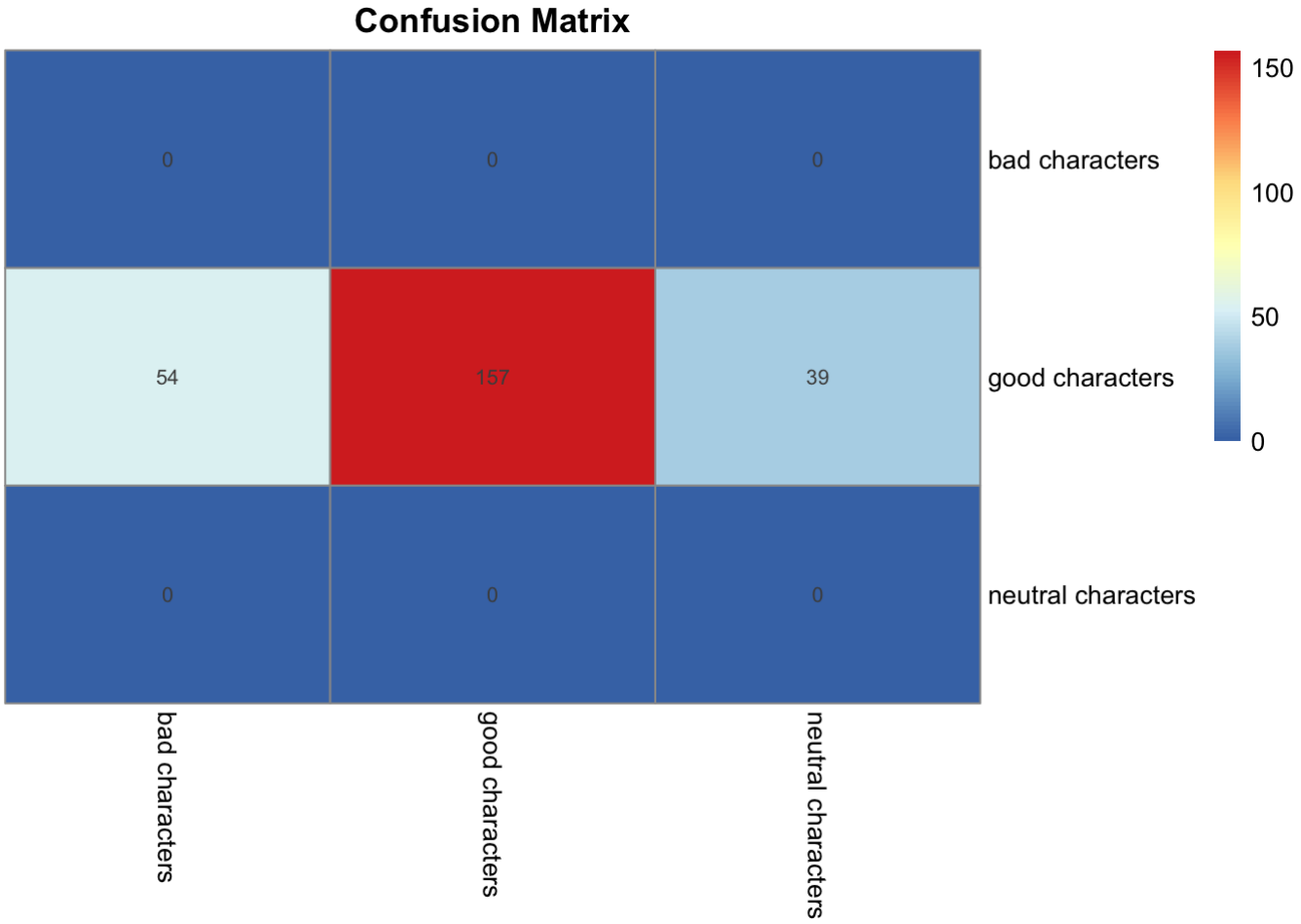
```
pred_P <- predict(SVM_fit_P, MyTestSET, type="class")
Ptable <- table(pred_P, test_label)
pheatmap(Ptable,cluster_cols=F,cluster_rows=F,display_numbers=T,number_format = "%.f",ma
in='Confusion Matrix')
```




```
SVM_fit_P <- svm(label~., data=old_set,
                 kernel="radial", cost=.3,
                 scale=FALSE)
print(SVM_fit_P)
```

```
##
## Call:
## svm(formula = label ~ ., data = old_set, kernel = "radial", cost = 0.3,
##     scale = FALSE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  0.3
##
## Number of Support Vectors:  740
```

```
pred_P <- predict(SVM_fit_P, MyTestSET, type="class")
Ptable <- table(pred_P, test_label)
pheatmap(Ptable,cluster_cols=F,cluster_rows=F,display_numbers=T,number_format = "%.f",ma
in='Confusion Matrix')
```


Confusion Matrix
0
0
0
54
157
39
0
0
0
bad characters
good characters
neutral characters
150
100
50
0
bad characters
good characters
neutral characters