Modeling

Alex Kan -lexokan

April 14, 2018

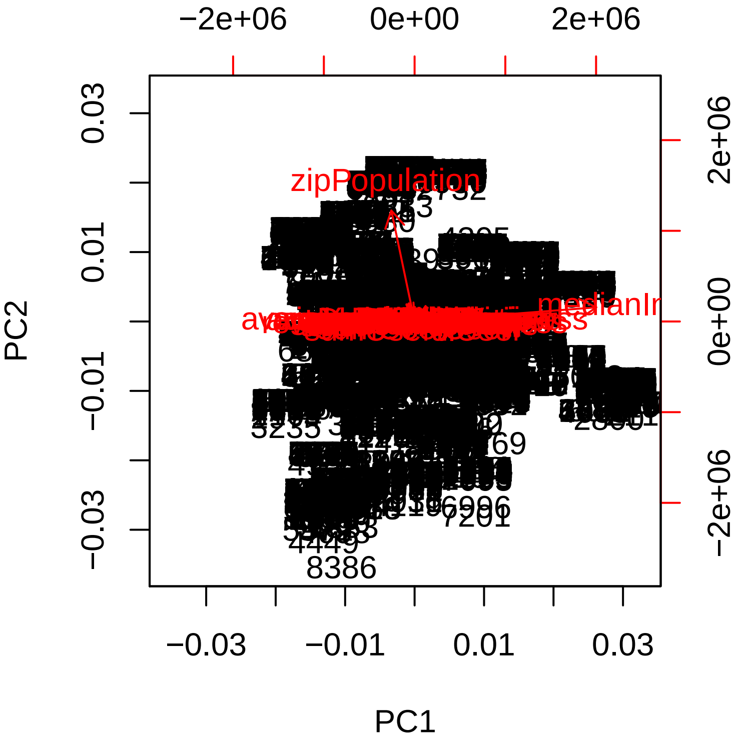
```
df <- read.csv("data/phoenixAg.csv")
temp <- df

temp <- df %>%
    dplyr::select(-c(businessID, businessName, city))
```

PCA:

```
pca <- prcomp(temp)

biplot(pca)
```



SVM:

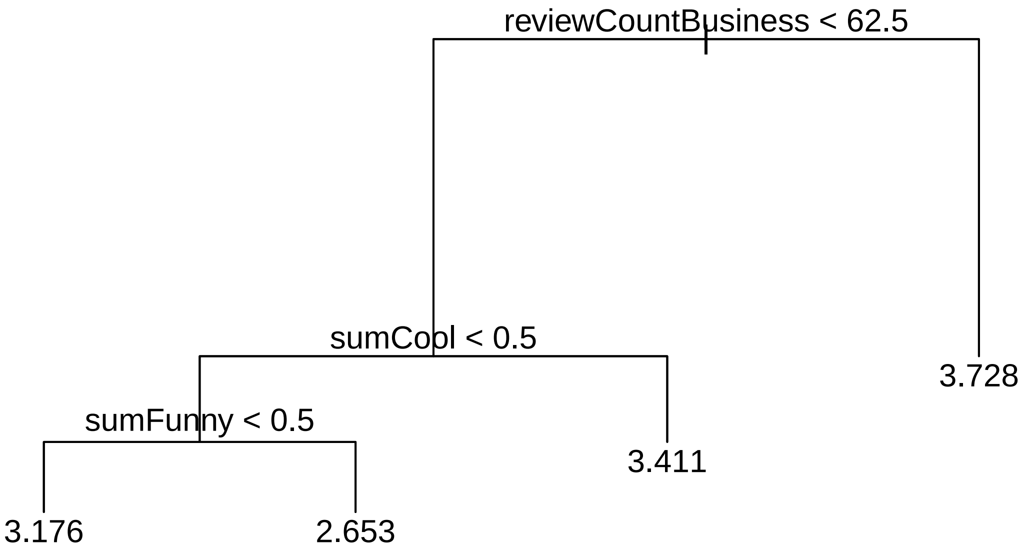
```
#svm.mod <- svm(data = temp, avgReviewStars~., kernel = "linear", scale = F, cost = .1)
```

Decision Trees:

```
tree.mod <- tree(data = temp %>%
                     dplyr::select(-c(averageReviewBusiness, avgUserPastReview)), avgReviewStars ~.)
```

```r
#plot(tree(data = temp,
#     as.formula(paste("avgReviewStars", "~",
#                      paste(colnames(temp)[8:16], collapse = "+"),
#                      sep = ""))))

plot(tree.mod)
text(tree.mod, pretty = 0)
```



Kmeans:

Logistic Regression

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis allows for non-linear (quadratic) decision boundaries, unlike Linear Discriminant Analysis. QDA require the number of predictor variables (p) to be less then the sample size (n). We are assuming predictor variables X are drawn from a multivariate Gaussian (aka normal) distribution and that the covariance matrix can be different for each class so we must estimate the covariance matrix separately for each class. However, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix is clearly untenable.

First, we will define a binary variable for averageReviewBusiness. Ratings greater than 3 are considered good ratings and ratings below 3 are bad ratings.

We will create the training and test sets(80%/20%).

Now, we will run QDA and look at the test MSE (mean squared error).

```r
az <- read.csv("data/phoenixAg.csv")
az$rate <- ifelse(az$averageReviewBusiness < 2, 0,1)

#Create Train & Test Sets
train <- sample(1:(0.80*nrow(az)),replace=FALSE)
az.train <- az[train,]
az.test <- az[-train,]
```

```r
qda.fit <- qda(rate~reviewCountBusiness, data = az.train)
qda.fit
```

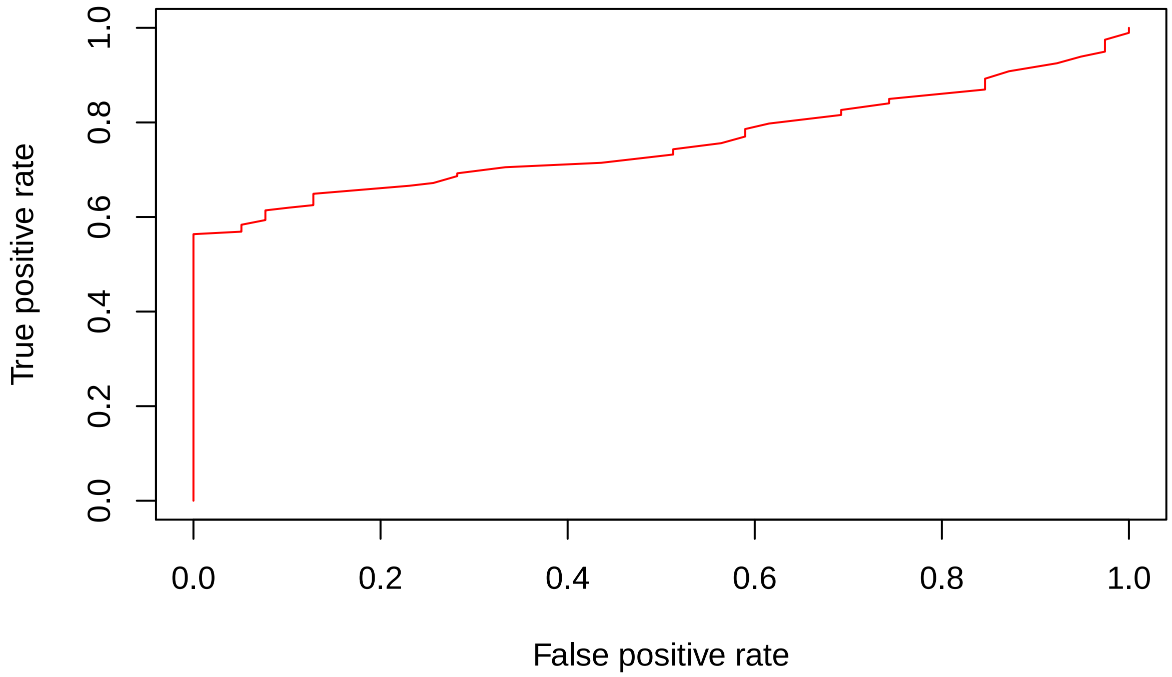
```
## Call:
## qda(rate ~ reviewCountBusiness, data = az.train)
##
## Prior probabilities of groups:
##          0          1
## 0.02302302 0.97697698
##
## Group means:
##   reviewCountBusiness
## 0            19.72671
## 1           105.09324
```

```r
#predict
qda.predict <- predict(qda.fit, newdata=az.test)
qda.class <- qda.predict$class
#Confusion matrix
table(qda.class,az.test$rate)
```

```
##
## qda.class    0    1
##         0    0    0
##         1   39 1710
```

```r
#Overall fraction of incorrect test predictions (MSE: mean squared error)
mean(qda.class != az.test$rate)
```

```
## [1] 0.02229846
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

```
## [[1]]
## [1] 0.7569051
```