Project Report
Gender Recognition by Voice
Authors: Group 4

Abstract
A typical gender recognition system can be divided into front-end system and back-end system. The task of the front-end system is to extract the gender related information from a speech signal and represents it by a set of vectors called feature. Features like power spectrum density, frequency at maximum power carry speaker information [1]. The feature is extracted using First Fourier Transform (FFT) algorithm [2]. The task of the back-end system (also called classifier) is to create a gender model to recognize the gender from his/her speech signal in recognition phase. The goal for this paper is trying to apply different potential models on voice data from male and female. By analyzing the datasets and comparing the models, we are trying to find a best model to recognize genders by voice.

Introduction
This project is classifying genders based on data of voice. As people know, male and female usually sound differently, but how does it get recognized? Why does it sound differently to human ears? After researching, factors like frequency, peak and voice range may be related to this problem.

Why is this so important? Vogt and André [2] suggested that the difference between genders helps improve automatic emotion recognition from speech. Harb and Chen [3] reported that classifying speaker's gender is an important task in the context of multimedia indexing. For experiment, the database was created to identify a voice as male or female, based upon acoustic properties of the voice and speech. The dataset consists of 3,168 recorded voice samples, collected from different male and female speakers. The voice samples are pre-processed by acoustic analysis in R using the seewave and tuneR packages, with an analyzed frequency range of 0hz-280hz.

Methods

Logistic Regression
It measures the relationship between the categorical dependent variable (DV) and one or more independent variables by estimating probabilities using a logistic function [5]. For the project, the underlying DV is called "label" which is categorical (binary) and has values male or female. The underlying distribution of the binary DV is binomial and the mean of the distribution, which is the probability of label(π), is to be modeled as a function of acoustic properties of the voice and speech such as meanfreq, sd, medan, etc. The log odds (Logit), is applied to the DV which is then expressed as a linear function of acoustic properties of the voice and speech in the following manner [5]:

$$Log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_{Mf}Meanfreq + \beta_{Sd}Sd + \beta_{M}Median + \beta_{Q25}Q25 + \beta_{Q75}Q75 + \beta_{IQR}IQR + \beta_{Skew}Skew + \beta_{Kurt}Kurt + \beta_{sp}Sp.ent + \beta_{s}Sfm + \beta_{model}Mode + \beta_{Centroid}Centroid + \beta_{P}Peakf + \beta_{Meanfun}Meanfun + \beta_{Minfun}Minfun + \beta_{Maxfun}Maxfun + \beta_{Meandom}Meandom + \beta_{Mindom}Mindom + \beta_{Maxdom}Maxdom + \beta_{Dfrange}Dfrange + \beta_{Modindx}Modindx.$$

The estimates of the β parameters of the logistic function above are obtained by the method of maximum likelihood estimation.

Linear discriminant analysis
In Linear discriminant analysis (LDA), the distribution of predictors X are modeled separately in each of the response classes, and use Bayes' theorem to flip these around into estimates for $\Pr(Y = k|X = x)$ [3].

Although LDA is popular in the case when response classes are more than two, it is still a proper model for the datasets of voice recognition. Since LDA is using Bayes' theorem for classification, the dataset of voice is suitable for using it. Because the two classes male and female data are well separated showing in Figure

1. The chart on the top is the fitted LDA model for female and the bottom one is for male.

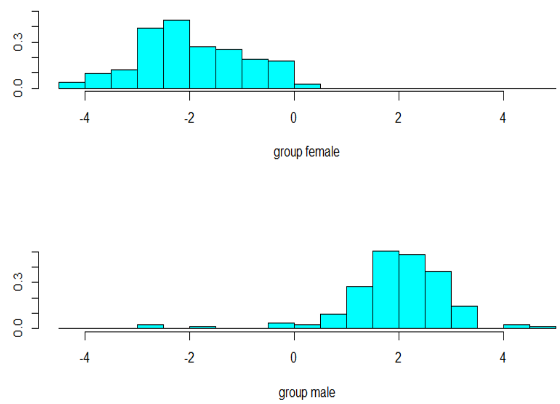


Figure 1.

In this situation, logistic regression model may become unstable, but LDA does not suffer from it. On the other hand, most of data are highly related to the gender, LDA can produce an accurate prediction without subsets selection needed.

Quadratic discriminant analysis

QDA is a special type of LDA, it models the likelihood of each class as a Gaussian distribution, then uses the posterior distributions to estimate the class for a given test point [3]. The Gaussian parameters for each class can be estimated from training points with maximum likelihood (ML) estimation. It measures two classes of male and female by a quadric surface.

K-Nearest-Neighbor

Given a positive integer K and a test observation x_0 , the KNN classifier first identifies the K points in the training data that are closest to x_0, represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0whose response values equal j:

$$\Pr(Y = j|X = x_0) = \frac{1}{K}\sum_{i \in N_0} I(y_i = j).$$

Finally, KNN applies Bayes rule and classifies the test observation x0 to the class with the largest probability [4].

Choice of K can goes from 1 to the size of dataset. As K increases, the method becomes less flexible; hence variance decreases and bias increases.

Random Forest

Using single decision tree suffers from high variance, a natural way to reduce the variance hence increase the prediction accuracy is to build many independent models and average the resulting predictions. In building random forest, each time a split in a tree is considered, a random sample of m predictors are chosen as split candidates from the full set of p predictors. Hence it decorrelates the trees by forcing each split to consider only a subset of the predictors [4].

Support Vector Machines

The concept of SVM is straightforward to use maximal margin hyperplane to separate datasets.

Figure 2.

By observing the dataset, it shows the datasets are well separated. In Figure 2, it shows female data in purple and men's in blue. Therefore, a linear kernel is probably enough to recognize them. In Figure 3, the hyperplane generated by linear kernel is separating male and female well. Moreover, the radial kernel is also performed to check how much the prediction can be improved.

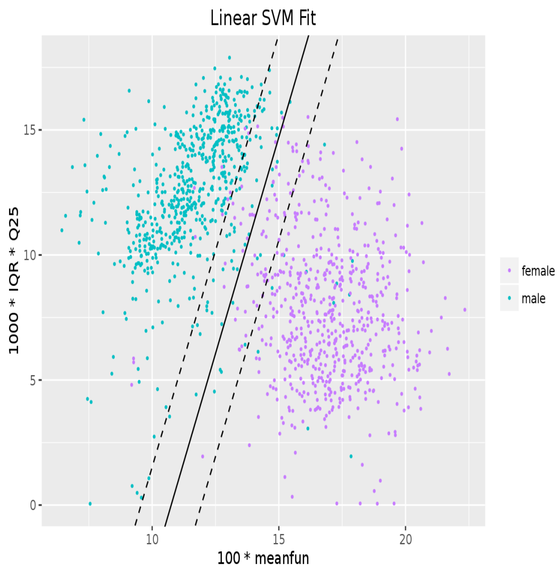


Figure 3.

Results

For the project, we divided the dataset into training subset and validation subset of equal size. For the logistic regression, first performed all variable selection on the training subset to find the predictors that have significant effect on label. Figure 4 shows results of all variable selection on the training subset using logistic regression.

```
Coefficients: (3 not defined because of singularities)
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.884e+01  1.245e+01  -1.513 0.130245
meanfreq    -2.508e+01  6.605e+01  -0.380 0.704117
sd          -4.510e+01  4.949e+01  -0.911 0.362159
median      -3.892e+00  1.846e+01  -0.211 0.833014
Q25         -6.501e+01  1.708e+01  -3.806 0.000141 ***
Q75          7.617e+01  2.946e+01   2.585 0.009728 **
IQR                 NA         NA      NA       NA
skew         2.048e-01  2.132e-01   0.961 0.336629
kurt        -8.175e-03  5.669e-03  -1.442 0.149246
sp.ent       4.737e+01  1.401e+01   3.381 0.000721 ***
sfm         -1.288e+01  3.782e+00  -3.407 0.000657 ***
mode         5.429e+00  3.070e+00   1.768 0.077022 .
centroid            NA         NA      NA       NA
meanfun     -1.735e+02  1.329e+01 -13.048  < 2e-16 ***
minfun       3.562e+01  1.668e+01   2.136 0.032661 *
maxfun       1.257e+01  9.212e+00   1.365 0.172312
meandom     -4.627e-01  6.013e-01  -0.770 0.441570
mindom       1.057e+00  2.983e+00   0.354 0.723131
maxdom       2.156e-02  9.121e-02   0.236 0.813151
dfrange             NA         NA      NA       NA
modindx     -1.170e+00  2.582e+00  -0.453 0.650497
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 4.

Then, we performed the best subset selection; however, due to large size of dataset and need to fit all $\binom{p}{k}$ models that contain exactly 20 predictors, we couldn't obtain the best subset selection result. Therefore, we chose to perform stepwise BIC to obtain the subset selection for the predictors. Figure 5 shows the result for subset selection.

```
Call:
glm(formula = label ~ Q25 + Q75 + sp.ent + sfm + meanfun + minfun,
    family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8014  -0.0345   0.0021   0.1038   4.2988

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -26.552      8.070  -3.290    0.001 **
Q25          -58.916      6.837  -8.617  < 2e-16 ***
Q75           53.546      7.321   7.314 2.60e-13 ***
sp.ent        55.907     10.614   5.267 1.38e-07 ***
sfm          -15.170      2.817  -5.385 7.25e-08 ***
meanfun     -166.507     11.992 -13.885  < 2e-16 ***
minfun        47.707     11.123   4.289 1.80e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2195.85  on 1583  degrees of freedom
Residual deviance:  272.55  on 1577  degrees of freedom
AIC: 286.55

Number of Fisher Scoring iterations: 8
```

Figure 5.

For the linear discriminant analysis (LDA), the result of LDA is so accurate. In Figure 6, by training the LDA model with 1584 observations, the ROC curve is on the top-left corner with 99.29% area under it for training subset.

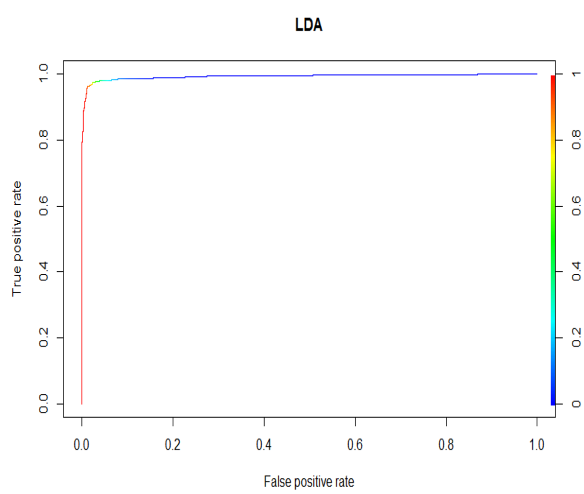


Figure 6.

Also in Figure 7, it shows 99.15% area under ROC curve of validation subset.

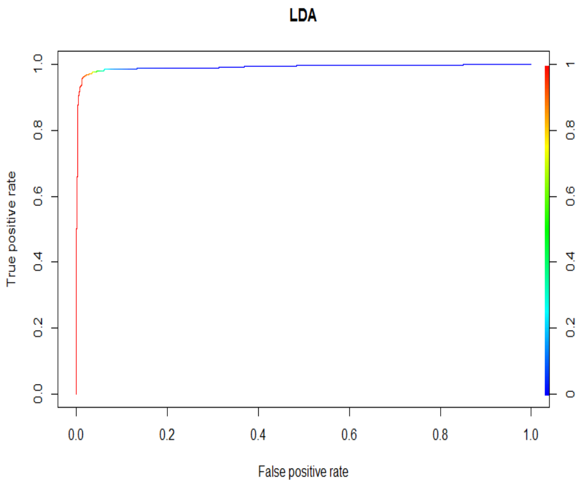


Figure 7.

As mentioned, LDA uses Bayes' theorem to flip these around into estimates for $\Pr(Y = k|X = x)$. The voice dataset for male and female are well separated, so LDA becomes one of the best choices for this situation. Furthermore, most of data in our collection are highly related to the gender recognition, and they are all numerical. Therefore, LDA fits well in this condition since it does not require any subset selection.

For Q discriminant analysis, the result of QDA is also very accurate. In Figure 8, the ROC curve has 99.35% area for training subset.

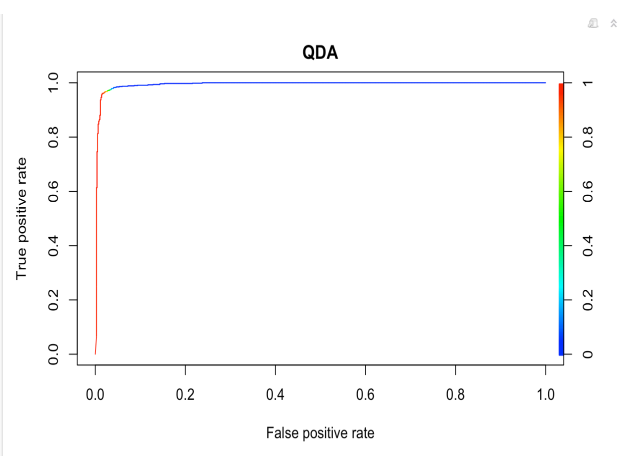


Figure 8.

Also, in Figure 9, it shows 98.98% area under ROC curve of validation subset.

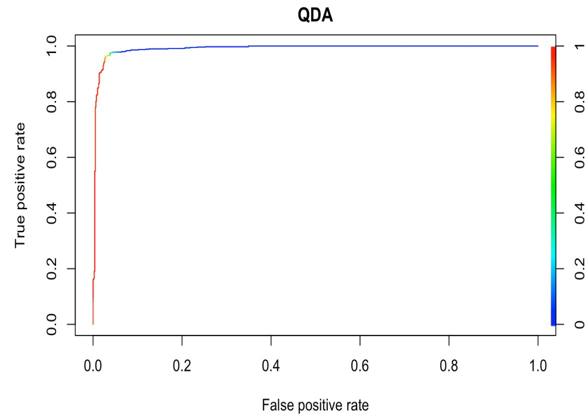


Figure 9.

Like LDA, the QDA classifier results from assuming that the observations from each class are drawn from a Gaussian distribution, and plugging estimates for the parameters into Bayes' theorem in order to perform prediction [4]. QDA is very similar as LDA, except QDA assumes that each class has its own covariance matrix [6]. Therefore, QDA and LDA have very close percentage area under ROC curve.

For KNN, we tested different K values (K=1, 3, 5). In Figure 10, it shows 67.82% area under ROC curve of validation set when K=1; with accuracy rate equals to $\frac{506+568}{1584} = 0.678.$

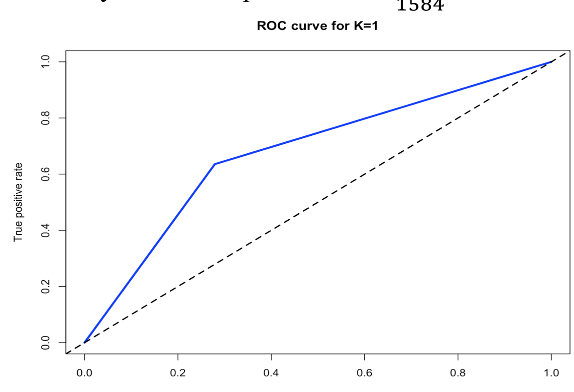


Figure 10.

In Figure 11, it shows 68.32% area under ROC curve of validation set when K=3; with accuracy rate equals to $\frac{518+564}{1584} = 0.683.$

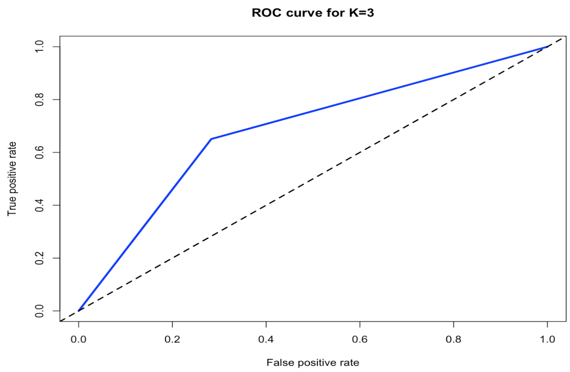


Figure 11.

And in Figure 12, it shows 68.33% area under ROC curve of validation set when K=5; with accuracy rate equals to $\frac{509+573}{1584} = 0.683$.

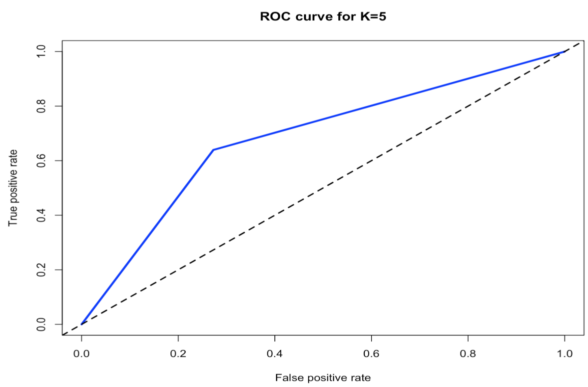


Figure 12.

As K increases, the model will become less fitted to the train subset, and eventually it will converge. When we observe the data in Figure 2, the data points are clearly divided into two parts, so K=5 will be a good stopping point.

For Random Forest, we included all predictors to build the trees, and as the tree models increased, the error rate decreased and became stable, and the accuracy is $\frac{777+771}{1584} = 0.9773$, shown in Figure 13. Figure 14 and Figure 15 shows the importance of predictors, the result is slightly different than the predictors selected by stepwise subset selection. In random forest, Q25 and sd are considered as important predictors, and minfun doesn't play an important role.

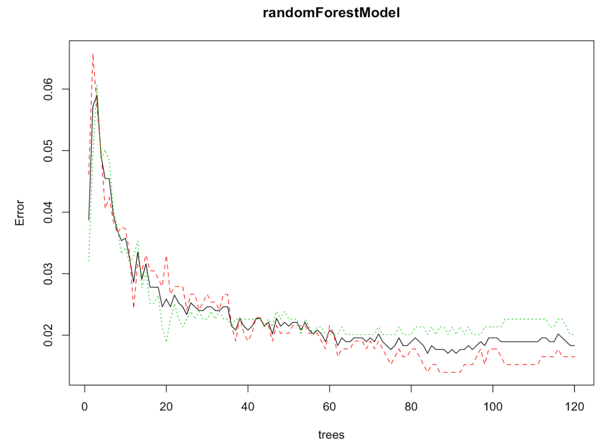


Figure 13.

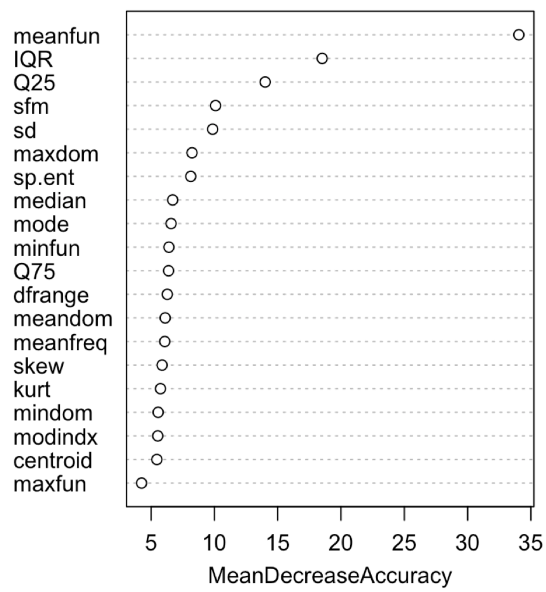


Figure 14.

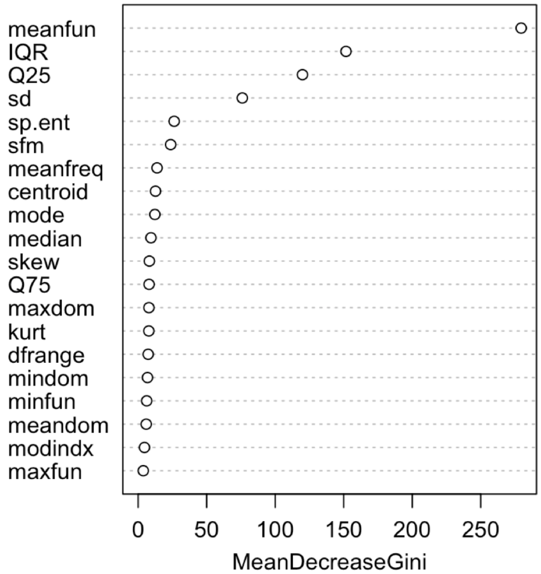


Figure 15.

SVM also does great in this problem. With simple linear kernel model, SVM can provide 97.66% accuracy on training subset with 1584 observations. Also it gives 97.29% on validation subset with the same number of observations. Moreover, radial kernel gives a slightly better prediction with 98.74% accuracy on training subset and 98.04% accuracy on validation subset. Also the ROC curve is built for SVM with radial kernel in Figure 16 and Figure 17. The area under ROC for training subset is 100% and 99.40% for validation subset.

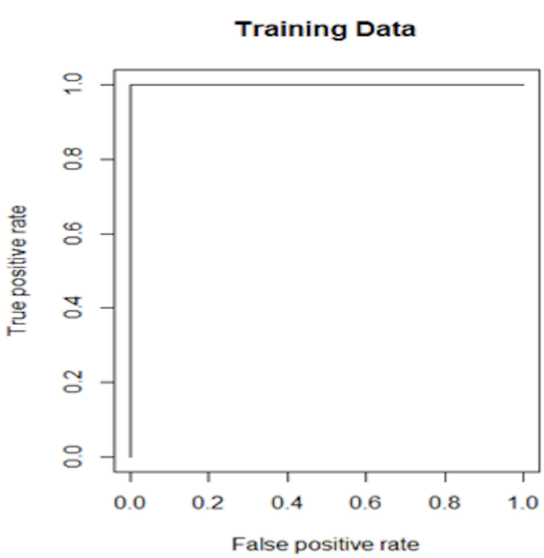


Figure 16.

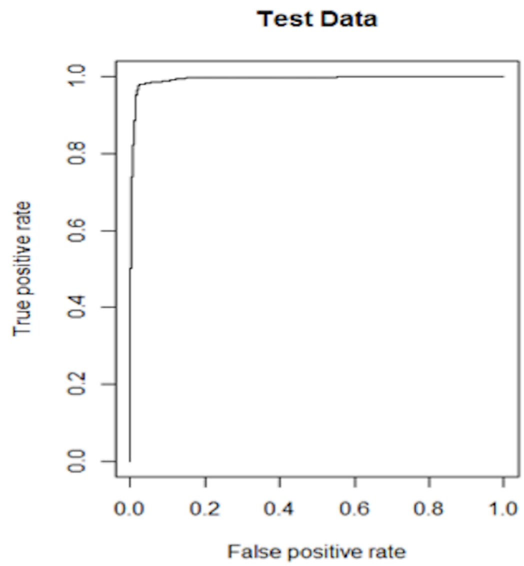


Figure 17.

Discussion

Compare different methods used in this project, the percentage area under the ROC curve for validation set for LDA, QDA, KNN, SVM are 99.15%, 99.06%, 68.26%, 99.40% respectively. SVM has the highest ROC value which means the simple linear kernel model can generate hyperplane that separates male and female well. LDA and QDA also have very high ROC value which means the response classes male and female in the dataset are well separated. KNN has the lowest ROC value which means KNN is not the good method for analyzing the dataset. The reason is simply because KNN is a non-parametric approach, and generally parametric approach such as logistic regression, LDA, QDA, random forest will outperform the non-parametric form if the parametric form that has been selected is close to the true form of function f [4]. The accuracy for random forest is 97.72% which means combining a large number of trees can often result in dramatic improvements in prediction accuracy, at the expense of some loss interpretation.

Overall, the results are pretty good for classifying the gender by voice. However, adding additional steps may improve the results such as divide the training set and the validation set unequally, or perform some data standardization for KNN because like LDA does feature scaling by design and would have no effect in performing if normalizing the dataset; however, it could have gravely affected for KNN.

References

[1] Ali, Md. Sadek. "Gender Recognition System Using Speech Signal." *International Journal of Computer Science, Engineering and Information Technology* 2.1 (2012): 1-9. Web.

[2] Steven W. Smith, Ph.D."The Scientist and Engineer's Guide to Digital Signal Processing By " *How the FFT works*. N.p., n.d. Web. 10 Apr. 2017.

[3] Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.

[4] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning.* Springer, New York, 2013.

[5] Chao-Ying Joanne Peng, Kuk Lida Lee, Gary M.Ingersoll. "An Introduction to Logistic Regression Analysis and Reporting"

[6] Jiehuan Sun, Hongyu Zhao. "The application of sparse estimation of covariance quadratic discriminant analysis".

Appendix

```
# Set up work directory
#-----------------------------------------------------------------
setwd('/Users/xinyandeng/Desktop/ML/Project')
# Read data
#-----------------------------------------------------------------
gender <- read.csv("voice.csv")

# Set the seed to produce the same result each time running the code
set.seed(123)

# 1. Data preparation and exploration
# (a) Select the training set:
#-----Partition the dataset into a training and a validation subsets of equal size-----
trainNum <- sample(x=1:nrow(gender), size=nrow(gender)/2)
train <- gender[trainNum,]
valid <- gender[-trainNum,]

# Convert the categorical variable to numeric for training set
train$label <- as.character(train$label)
train$label[which(train$label=="male")] <- "1"
train$label[which(train$label=="female")] <- "0"
train$label <- as.numeric(train$label)

# Convert the categorical variable to numeric for validation set
valid$label <- as.character(valid$label)
valid$label[which(valid$label=="male")] <- "1"
valid$label[which(valid$label=="female")] <- "0"
valid$label <- as.numeric(valid$label)

# (b) Data exploration:
# One-variable summary
summary(train)
```

```
##     meanfreq            sd             median           Q25
##  Min.   :0.0390   Min.   :0.0240   Min.   :0.011   Min.   :0.0000
##  1st Qu.:0.1640   1st Qu.:0.0420   1st Qu.:0.169   1st Qu.:0.1120
##  Median :0.1840   Median :0.0590   Median :0.189   Median :0.1400
##  Mean   :0.1804   Mean   :0.0573   Mean   :0.185   Mean   :0.1399
##  3rd Qu.:0.1990   3rd Qu.:0.0670   3rd Qu.:0.210   3rd Qu.:0.1750
##  Max.   :0.2500   Max.   :0.1150   Max.   :0.261   Max.   :0.2420
##       Q75              IQR               skew              kurt
##  Min.   :0.0430   Min.   :0.01500   Min.   : 0.590   Min.   :   2.463
##  1st Qu.:0.2080   1st Qu.:0.04200   1st Qu.: 1.654   1st Qu.:   5.691
##  Median :0.2260   Median :0.09400   Median : 2.200   Median :   8.345
##  Mean   :0.2244   Mean   :0.08445   Mean   : 3.182   Mean   :  37.954
##  3rd Qu.:0.2440   3rd Qu.:0.11400   3rd Qu.: 2.930   3rd Qu.:  13.711
##  Max.   :0.2700   Max.   :0.25200   Max.   :34.725   Max.   :1309.613
##      sp.ent           sfm              mode           centroid
##  Min.   :0.739   Min.   :0.0830   Min.   :0.0000   Min.   :0.0390
##  1st Qu.:0.861   1st Qu.:0.2600   1st Qu.:0.1160   1st Qu.:0.1640
##  Median :0.901   Median :0.3970   Median :0.1850   Median :0.1840
##  Mean   :0.895   Mean   :0.4092   Mean   :0.1637   Mean   :0.1804
```

```
##  3rd Qu.:0.928   3rd Qu.:0.5290   3rd Qu.:0.2210   3rd Qu.:0.1990
##  Max.   :0.982   Max.   :0.8430   Max.   :0.2800   Max.   :0.2500
##     meanfun           minfun            maxfun           meandom
##  Min.   :0.0560   Min.   :0.01000   Min.   :0.1030   Min.   :0.0080
##  1st Qu.:0.1170   1st Qu.:0.01800   1st Qu.:0.2500   1st Qu.:0.4340
##  Median :0.1410   Median :0.04700   Median :0.2710   Median :0.7670
##  Mean   :0.1427   Mean   :0.03694   Mean   :0.2587   Mean   :0.8305
##  3rd Qu.:0.1690   3rd Qu.:0.04800   3rd Qu.:0.2770   3rd Qu.:1.1610
##  Max.   :0.2380   Max.   :0.20400   Max.   :0.2790   Max.   :2.8050
##      mindom            maxdom          dfrange          modindx
##  Min.   :0.00500   Min.   : 0.008   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:0.00800   1st Qu.: 2.070   1st Qu.: 2.063   1st Qu.:0.1010
##  Median :0.02300   Median : 4.953   Median : 4.926   Median :0.1400
##  Mean   :0.05327   Mean   : 5.060   Mean   : 5.007   Mean   :0.1759
##  3rd Qu.:0.07800   3rd Qu.: 7.008   3rd Qu.: 6.992   3rd Qu.:0.2120
##  Max.   :0.44900   Max.   :21.867   Max.   :21.844   Max.   :0.8800
##      label
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.5025
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

```r
# Two-variable summary
round(cor(train), digits = 5)
```

```
##           meanfreq       sd   median      Q25      Q75      IQR     skew
## meanfreq  1.00000 -0.73142  0.92420  0.90655  0.74172 -0.60853 -0.34877
## sd       -0.73142  1.00000 -0.55455 -0.84371 -0.15295  0.86728  0.32786
## median    0.92420 -0.55455  1.00000  0.77132  0.72895 -0.46296 -0.27893
## Q25       0.90655 -0.84371  0.77132  1.00000  0.46548 -0.86861 -0.34301
## Q75       0.74172 -0.15295  0.72895  0.46548  1.00000  0.03410 -0.23489
## IQR      -0.60853  0.86728 -0.46296 -0.86861  0.03410  1.00000  0.25535
## skew     -0.34877  0.32786 -0.27893 -0.34301 -0.23489  0.25535  1.00000
## kurt     -0.33723  0.35432 -0.26176 -0.36922 -0.17426  0.31895  0.97709
## sp.ent   -0.58274  0.70247 -0.49123 -0.63217 -0.15319  0.62848 -0.18871
## sfm      -0.77379  0.83011 -0.65584 -0.75545 -0.36475  0.64918  0.09357
## mode      0.67918 -0.53137  0.66020  0.58074  0.47872 -0.38766 -0.44217
## centroid  1.00000 -0.73142  0.92420  0.90655  0.74172 -0.60853 -0.34877
## meanfun   0.46801 -0.46428  0.42855  0.55205  0.16014 -0.53397 -0.15987
## minfun    0.38804 -0.35375  0.33999  0.32869  0.24875 -0.23196 -0.21546
## maxfun    0.27178 -0.10386  0.24984  0.18783  0.30499 -0.04129 -0.07103
## meandom   0.55924 -0.47986  0.48341  0.47944  0.39339 -0.32116 -0.33898
## mindom    0.23118 -0.38140  0.18964  0.31130 -0.03698 -0.37235 -0.07306
## maxdom    0.53790 -0.48026  0.45915  0.47080  0.36189 -0.32919 -0.31003
## dfrange   0.53406 -0.47372  0.45600  0.46550  0.36271 -0.32274 -0.30889
## modindx  -0.19923  0.10548 -0.19890 -0.11894 -0.20290  0.02089 -0.17009
## label    -0.33580  0.46527 -0.29254 -0.51713  0.06858  0.62261  0.03191
##              kurt   sp.ent      sfm     mode centroid  meanfun   minfun
## meanfreq -0.33723 -0.58274 -0.77379  0.67918  1.00000  0.46801  0.38804
## sd        0.35432  0.70247  0.83011 -0.53137 -0.73142 -0.46428 -0.35375
## median   -0.26176 -0.49123 -0.65584  0.66020  0.92420  0.42855  0.33999
## Q25      -0.36922 -0.63217 -0.75545  0.58074  0.90655  0.55205  0.32869
## Q75      -0.17426 -0.15319 -0.36475  0.47872  0.74172  0.16014  0.24875
```

```
## IQR       0.31895  0.62848  0.64918 -0.38766 -0.60853 -0.53397 -0.23196
## skew      0.97709 -0.18871  0.09357 -0.44217 -0.34877 -0.15987 -0.21546
## kurt      1.00000 -0.12196  0.12142 -0.40916 -0.33723 -0.18791 -0.20314
## sp.ent   -0.12196  1.00000  0.86558 -0.31776 -0.58274 -0.51878 -0.30677
## sfm       0.12142  0.86558  1.00000 -0.48405 -0.77379 -0.42558 -0.36405
## mode     -0.40916 -0.31776 -0.48405  1.00000  0.67918  0.32029  0.40638
## centroid -0.33723 -0.58274 -0.77379  0.67918  1.00000  0.46801  0.38804
## meanfun  -0.18791 -0.51878 -0.42558  0.32029  0.46801  1.00000  0.34653
## minfun   -0.20314 -0.30677 -0.36405  0.40638  0.38804  0.34653  1.00000
## maxfun   -0.03333 -0.09961 -0.17070  0.18055  0.27178  0.30471  0.19536
## meandom  -0.30613 -0.29354 -0.43299  0.50901  0.55924  0.27471  0.36769
## mindom   -0.11288 -0.30953 -0.30514  0.19902  0.23118  0.16439  0.09179
## maxdom   -0.27833 -0.32126 -0.43751  0.49965  0.53790  0.28929  0.30653
## dfrange  -0.27646 -0.31592 -0.43230  0.49635  0.53406  0.28652  0.30503
## modindx  -0.20536  0.18217  0.19182 -0.17372 -0.19923 -0.05333  0.03673
## label     0.08507  0.49421  0.34973 -0.15993 -0.33580 -0.83618 -0.13712
##            maxfun  meandom   mindom   maxdom  dfrange  modindx    label
## meanfreq  0.27178  0.55924  0.23118  0.53790  0.53406 -0.19923 -0.33580
## sd       -0.10386 -0.47986 -0.38140 -0.48026 -0.47372  0.10548  0.46527
## median    0.24984  0.48341  0.18964  0.45915  0.45600 -0.19890 -0.29254
## Q25       0.18783  0.47944  0.31130  0.47080  0.46550 -0.11894 -0.51713
## Q75       0.30499  0.39339 -0.03698  0.36189  0.36271 -0.20290  0.06858
## IQR      -0.04129 -0.32116 -0.37235 -0.32919 -0.32274  0.02089  0.62261
## skew     -0.07103 -0.33898 -0.07306 -0.31003 -0.30889 -0.17009  0.03191
## kurt     -0.03333 -0.30613 -0.11288 -0.27833 -0.27646 -0.20536  0.08507
## sp.ent   -0.09961 -0.29354 -0.30953 -0.32126 -0.31592  0.18217  0.49421
## sfm      -0.17070 -0.43299 -0.30514 -0.43751 -0.43230  0.19182  0.34973
## mode      0.18055  0.50901  0.19902  0.49965  0.49635 -0.17372 -0.15993
## centroid  0.27178  0.55924  0.23118  0.53790  0.53406 -0.19923 -0.33580
## meanfun   0.30471  0.27471  0.16439  0.28929  0.28652 -0.05333 -0.83618
## minfun    0.19536  0.36769  0.09179  0.30653  0.30503  0.03673 -0.13712
## maxfun    1.00000  0.34135 -0.27725  0.35138  0.35647 -0.39000 -0.13976
## meandom   0.34135  1.00000  0.09720  0.81647  0.81515 -0.19162 -0.17552
## mindom   -0.27725  0.09720  1.00000  0.03755  0.01983  0.20918 -0.20213
## maxdom    0.35138  0.81647  0.03755  1.00000  0.99984 -0.42282 -0.19207
## dfrange   0.35647  0.81515  0.01983  0.99984  1.00000 -0.42674 -0.18859
## modindx  -0.39000 -0.19162  0.20918 -0.42282 -0.42674  1.00000  0.02570
## label    -0.13976 -0.17552 -0.20213 -0.19207 -0.18859  0.02570  1.00000
```

```r
# Missing data
sum(is.na(train))
```

```
## [1] 0
```

```r
# There is no missing data

#-------------------------- Fit logistic regression on the training set --------------------------------
glm.fit <- glm(label~., data = train, family = "binomial")
summary(glm.fit)
```

```
##
## Call:
## glm(formula = label ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
```

```
## -2.6060  -0.0310   0.0019   0.0888   4.2171
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.011e+01  1.265e+01  -1.589 0.111997
## meanfreq    -2.733e+01  6.509e+01  -0.420 0.674612
## sd          -4.686e+01  5.039e+01  -0.930 0.352371
## median      -3.945e+00  1.839e+01  -0.215 0.830144
## Q25          5.123e+02  3.436e+02   1.491 0.135973
## Q75         -5.016e+02  3.455e+02  -1.452 0.146476
## IQR          5.783e+02  3.443e+02   1.679 0.093088 .
## skew         2.310e-01  2.162e-01   1.068 0.285383
## kurt        -9.170e-03  5.748e-03  -1.595 0.110645
## sp.ent       5.079e+01  1.425e+01   3.565 0.000365 ***
## sfm         -1.356e+01  3.800e+00  -3.568 0.000360 ***
## mode         5.209e+00  3.082e+00   1.690 0.090993 .
## centroid            NA         NA      NA       NA
## meanfun     -1.750e+02  1.364e+01 -12.835  < 2e-16 ***
## minfun       3.086e+01  1.712e+01   1.803 0.071443 .
## maxfun       1.050e+01  9.402e+00   1.117 0.264203
## meandom     -4.840e-01  6.057e-01  -0.799 0.424290
## mindom      -6.133e+02  3.569e+02  -1.718 0.085751 .
## maxdom       6.146e+02  3.570e+02   1.721 0.085189 .
## dfrange     -6.146e+02  3.570e+02  -1.721 0.085195 .
## modindx     -1.537e+00  2.628e+00  -0.585 0.558635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2195.85  on 1583  degrees of freedom
## Residual deviance:  251.71  on 1564  degrees of freedom
## AIC: 291.71
##
## Number of Fisher Scoring iterations: 8
```

```r
# Perform best subset selection, and select only the predictors which have significant effect on label
# library(glmulti)
# fit.glmulti <- glmulti(label ~., data=train, level=1, method="h",
#                        crit="aic", confsetsize=5, plotty=F, report=F,
#                        fitfunction="glm", family=binomial)
# However, because of large dataset and all the combination of possibilities, best subset selection
# takes too long to obtain. Therefore, use stepwise BIC to obtain subset selection.

step.bic <- step(glm.fit, k=log(nrow(train)), trace=F)
summary(step.bic)
```

```
##
## Call:
## glm(formula = label ~ IQR + sp.ent + sfm + meanfun + minfun,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8099  -0.0352   0.0022   0.1030   4.2791
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -24.965      7.543  -3.310 0.000934 ***
## IQR           56.624      6.016   9.412  < 2e-16 ***
## sp.ent        52.527      9.249   5.679 1.35e-08 ***
## sfm          -14.001      2.265  -6.182 6.35e-10 ***
## meanfun     -166.410     11.967 -13.905  < 2e-16 ***
## minfun        46.622     11.110   4.196 2.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2195.85  on 1583  degrees of freedom
## Residual deviance:  272.48  on 1578  degrees of freedom
## AIC: 284.48
##
## Number of Fisher Scoring iterations: 8
```

```
# Keep the categorial variable
train <- gender[trainNum,]
valid <- gender[-trainNum,]


#-------------------------------- Fit LDA on training set -------------------------------------------
library(MASS)
lda.fit <- lda(label ~ ., data=train)
```

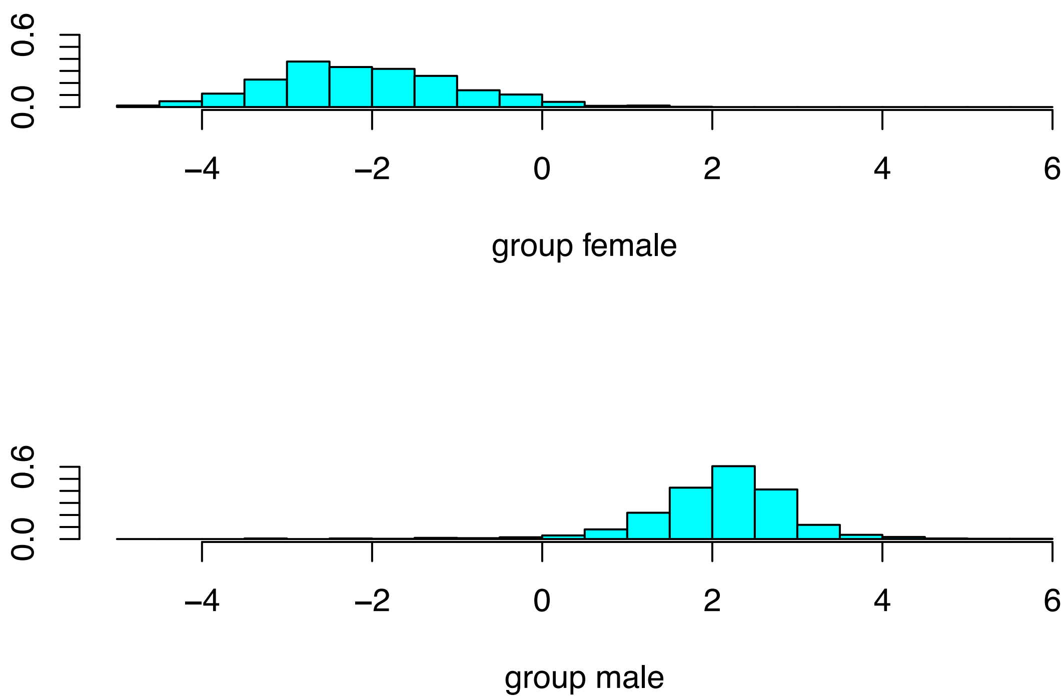
```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
lda.fit
```

```
## Call:
## lda(label ~ ., data = train)
##
## Prior probabilities of groups:
##    female      male
## 0.4974747 0.5025253
##
## Group means:
##         meanfreq         sd    median       Q25       Q75       IQR
## female 0.1905038 0.04950508 0.1956853 0.1651891 0.2226980 0.0575000
## male   0.1704020 0.06502136 0.1744058 0.1149108 0.2260013 0.1111281
##            skew     kurt    sp.ent       sfm     mode  centroid   meanfun
## female 3.042665 26.03977 0.8728756 0.3473376 0.176302 0.1905038 0.1693147
## male   3.320005 49.74817 0.9169837 0.4703957 0.151304 0.1704020 0.1163229
##            minfun    maxfun   meandom     mindom   maxdom  dfrange
## female 0.03965355 0.2628756 0.9231713 0.06599239 5.741813 5.675717
## male   0.03426256 0.2545264 0.7386658 0.04067839 4.385543 4.344750
##          modindx
## female 0.1727538
## male   0.1790176
##
## Coefficients of linear discriminants:
##                     LD1
```

```
## meanfreq  -5.187263904
## sd         4.834913064
## median    -3.770067363
## Q25       19.134513564
## Q75       -1.426809413
## IQR       33.700493722
## skew      -0.133982169
## kurt       0.001967202
## sp.ent     0.112964004
## sfm       -2.396110475
## mode       2.399651393
## centroid  -5.187263904
## meanfun  -64.328228633
## minfun    15.572361963
## maxfun     5.072622244
## meandom   -0.276834448
## mindom     0.666796625
## maxdom     0.002336375
## dfrange    0.002188687
## modindx    0.170239989
```

```r
plot(lda.fit)
```

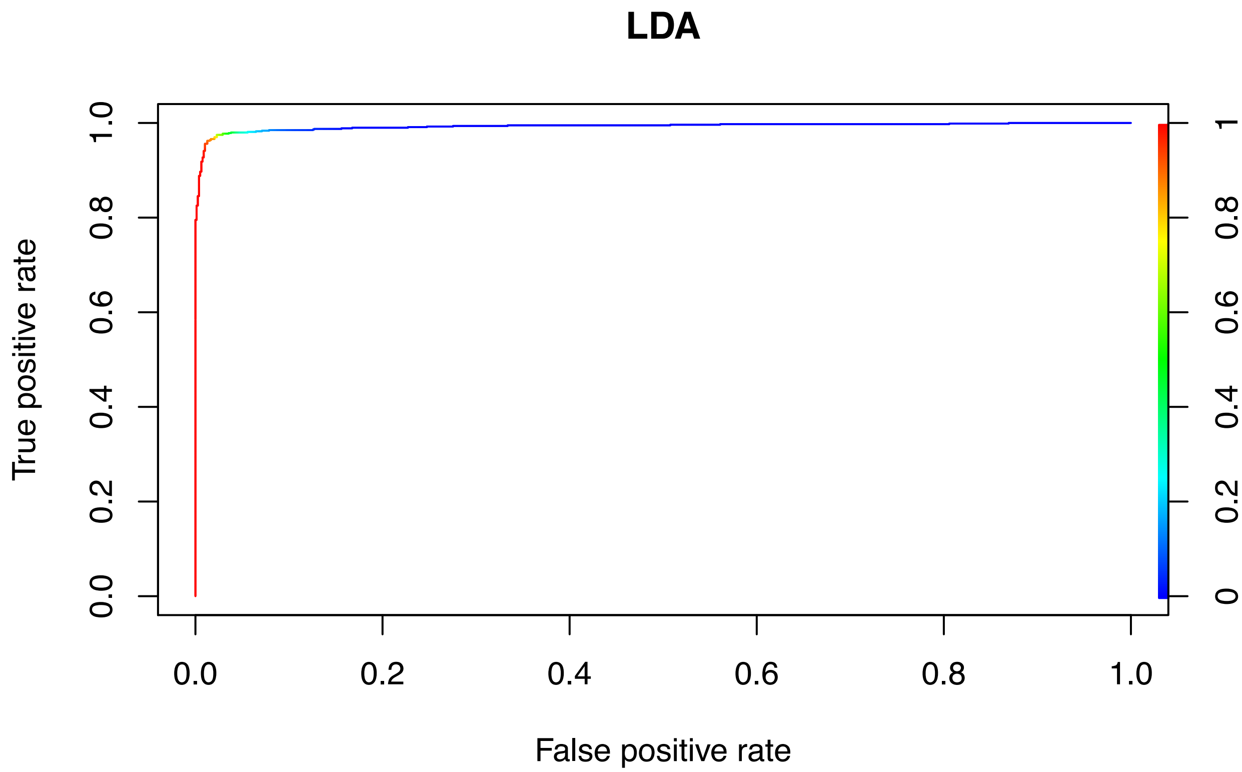



```r
# ROC
# Training set
library(ROCR)
```

```
## Loading required package: gplots

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
```

```
##     lowess
```

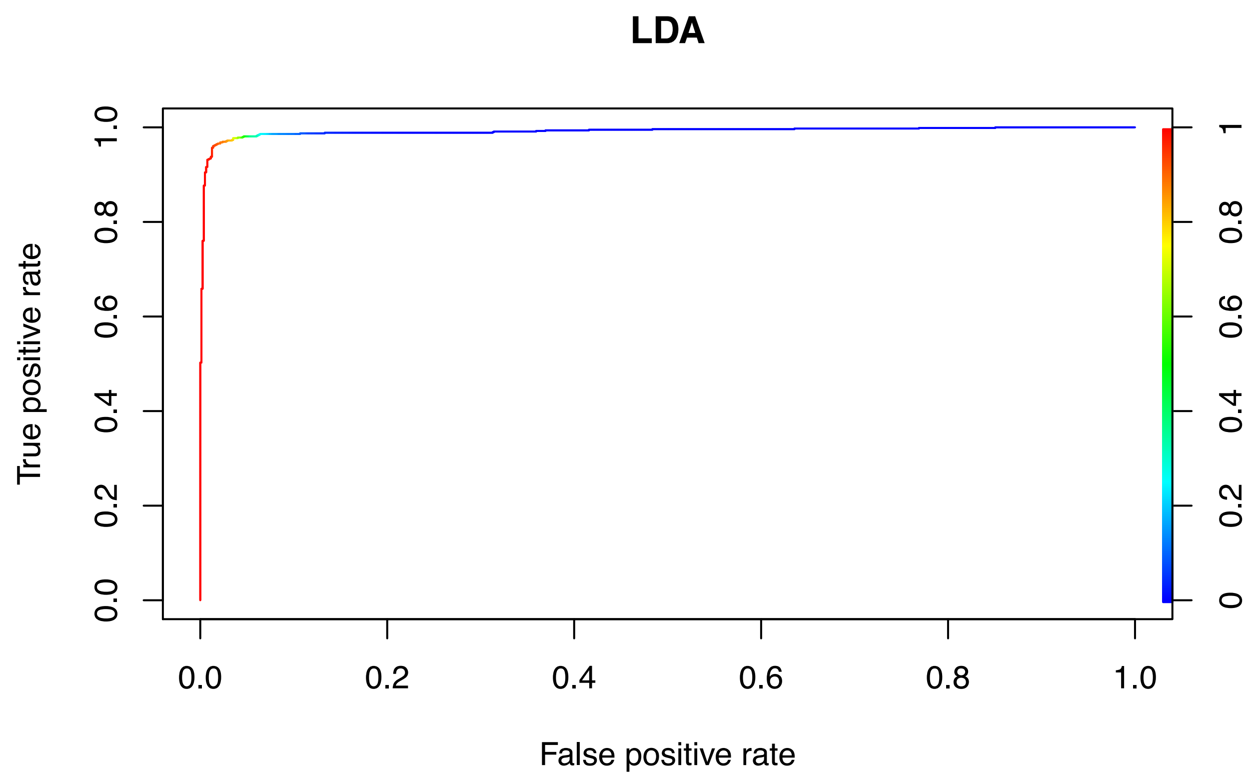
```r
scores <- predict(lda.fit, newdata= train)$posterior[,2]
pred <- prediction( scores, labels= train$label )
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize=T, main="LDA")
```




```r
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.9928561
```

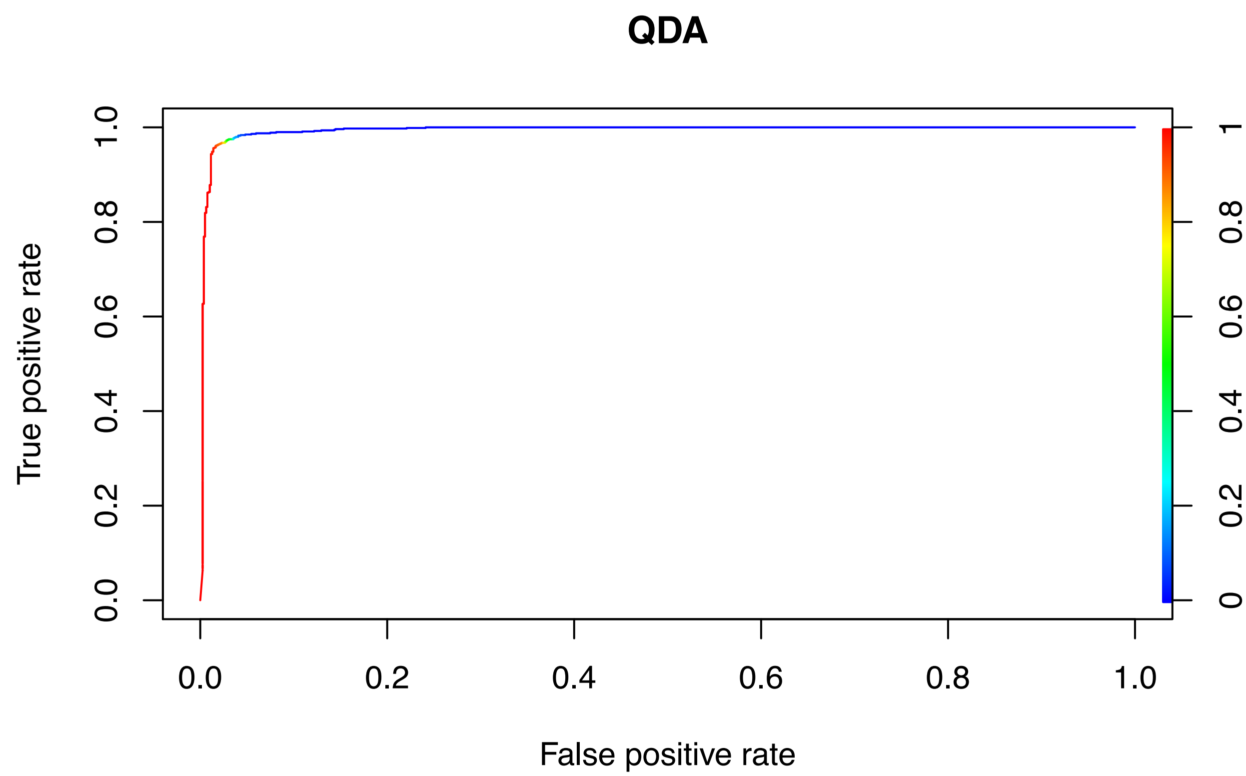
```r
# Validation set
scores <- predict(lda.fit, newdata= valid)$posterior[,2]
pred <- prediction( scores, labels= valid$label )
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize=T, main="LDA")
```




```r
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.991493
```

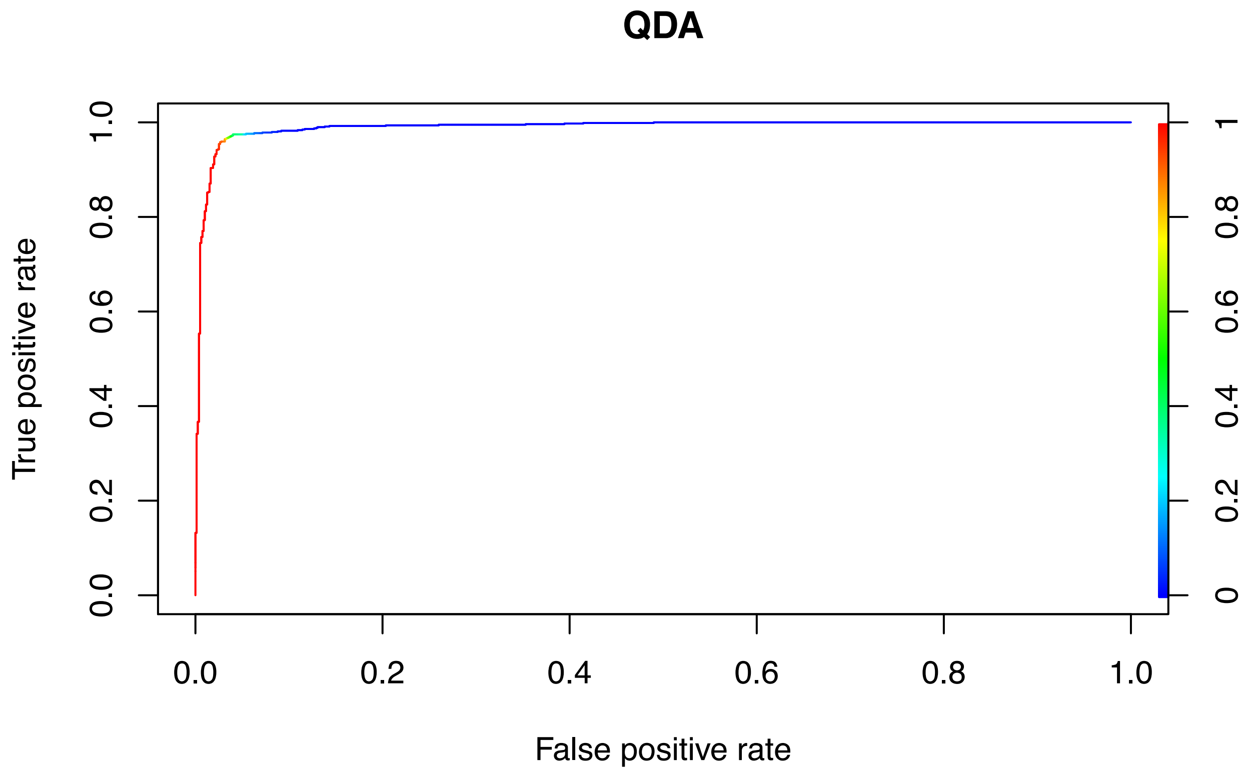
```r
#------------------------------------ Fit QDA on training set ------------------------------------------
# Based on result of summary(glm.fit), IQR, centroid, dfrange is NA, so ingore those predictors for QDA
qda.fit <- qda(label ~ meanfreq + sd + median + Q25 + Q75 + skew + kurt + sp.ent + sfm + mode + meanfun

# ROC
# Training set
library(ROCR)
scores <- predict(qda.fit, train[, c(1,2,3,4,5,7,8,9,10,11,13,14,15,16,17,18,20,21)])$posterior[,2]
pred <- prediction(scores, labels = train$label)
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize=T, main="QDA")
```




```r
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.993556
```

```r
# Validation set
scores <- predict(qda.fit, valid[, c(1,2,3,4,5,7,8,9,10,11,13,14,15,16,17,18,20,21)])$posterior[,2]
pred <- prediction( scores, labels= valid$label )
perf <- performance(pred, "tpr", "fpr")
plot(perf, colorize=T, main="QDA")
```




```
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.9898318
```

```
#----------------------------------------- KNN ---------------------------------------------
# KNN
library(class)
train_X <- train[, c(1:20)]
valid_X <- valid[, c(1:20)]
train_label <- train[,c(21)]
valid_label <- valid[,c(21)]

knn.pred1 = knn(train_X, valid_X, train_label, 1)
table(knn.pred1, valid_label)
```

```
##           valid_label
## knn.pred1 female male
##    female    506  220
##    male      290  568
```

```
mean(valid_label != knn.pred1)
```

```
## [1] 0.3219697
```

```
knn.pred3 = knn(train_X, valid_X, train_label, 3)
table(knn.pred3, valid_label)
```

```
##           valid_label
## knn.pred3 female male
##    female    518  224
##    male      278  564
```

```r
mean(valid_label != knn.pred3)
```

```
## [1] 0.3169192
```

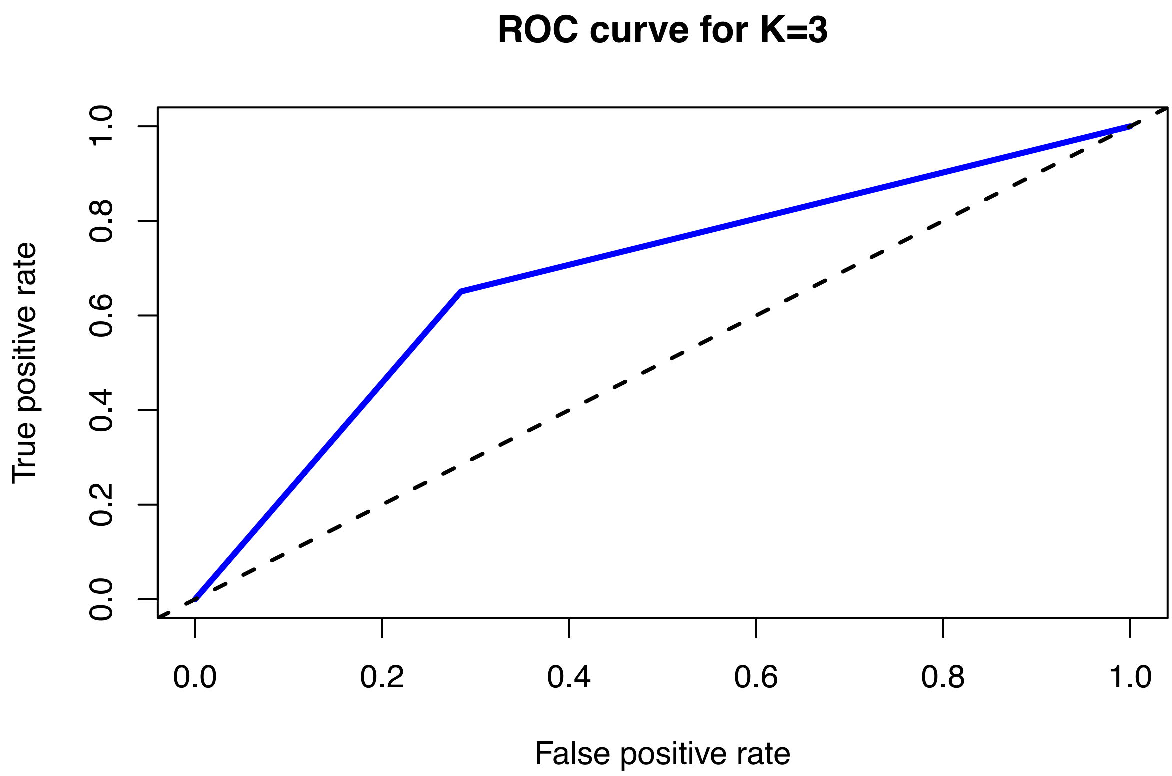
```r
knn.pred5 = knn(train_X, valid_X, train_label, 5)
table(knn.pred5, valid_label)
```

```
##            valid_label
## knn.pred5 female male
##    female    509  215
##    male      287  573
```

```r
mean(valid_label != knn.pred5)
```

```
## [1] 0.3169192
```

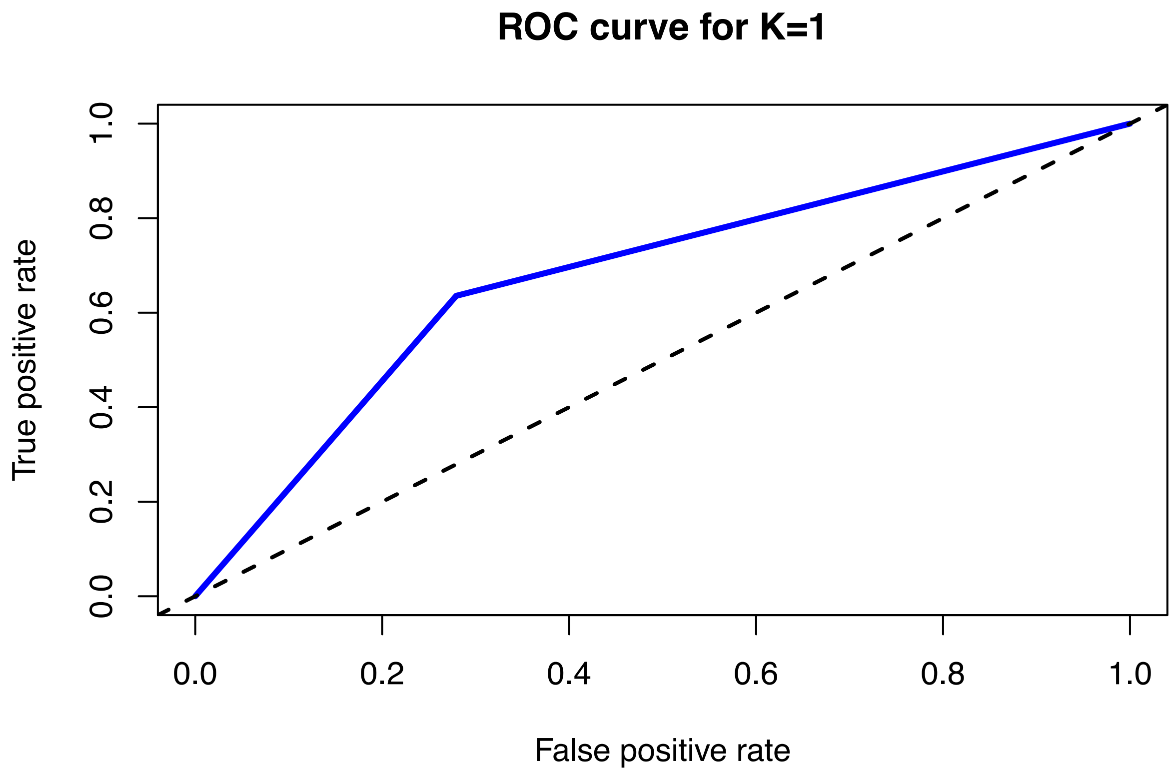
```r
# plot
# KNN = 3
library(ROCR)
pred <- ifelse(knn.pred3 == "female", 1, 0)
real <- ifelse(valid_label == "female", 1, 0)
prediction1 <- prediction(pred, real)
performance1 <- performance(prediction1, measure = "tpr", x.measure = "fpr")
plot(performance1, main = "ROC curve for K=3",col = "blue", lwd = 3)
abline(a = 0, b = 1, lwd = 2, lty = 2)
```




```r
perf.auc <- performance(prediction1, measure = "auc")
# print out the area under the curve
unlist(perf.auc@y.values)
```

```
## [1] 0.6832449
```

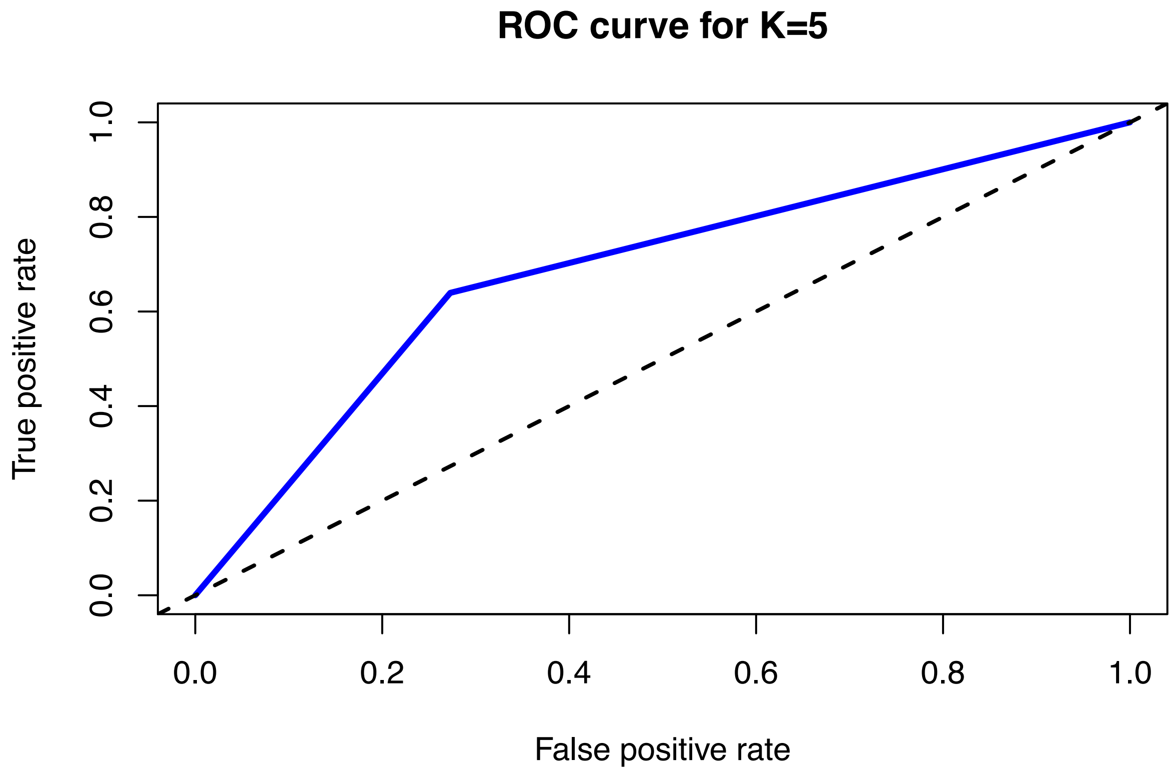
```r
# KNN = 1
pred1 <- ifelse(knn.pred1 == "female", 1, 0)
prediction11 <- prediction(pred1, real)
performance11 <- performance(prediction11, measure = "tpr", x.measure = "fpr")
plot(performance11, main = "ROC curve for K=1",col = "blue", lwd = 3)
abline(a = 0, b = 1, lwd = 2, lty = 2)
```




```r
perf.auc1 <- performance(prediction11, measure = "auc")
# print out the area under the curve
unlist(perf.auc1@y.values)
```

```
## [1] 0.6782453
```

```r
# KNN=5
pred5 <- ifelse(knn.pred5 == "female", 1, 0)
prediction5 <- prediction(pred5, real)
performance5 <- performance(prediction5, measure = "tpr", x.measure = "fpr")
plot(performance5, main = "ROC curve for K=5",col = "blue", lwd = 3)
abline(a = 0, b = 1, lwd = 2, lty = 2)
```




```
perf.auc5 <- performance(prediction5, measure = "auc")
# print out the area under the curve
unlist(perf.auc5@y.values)
```

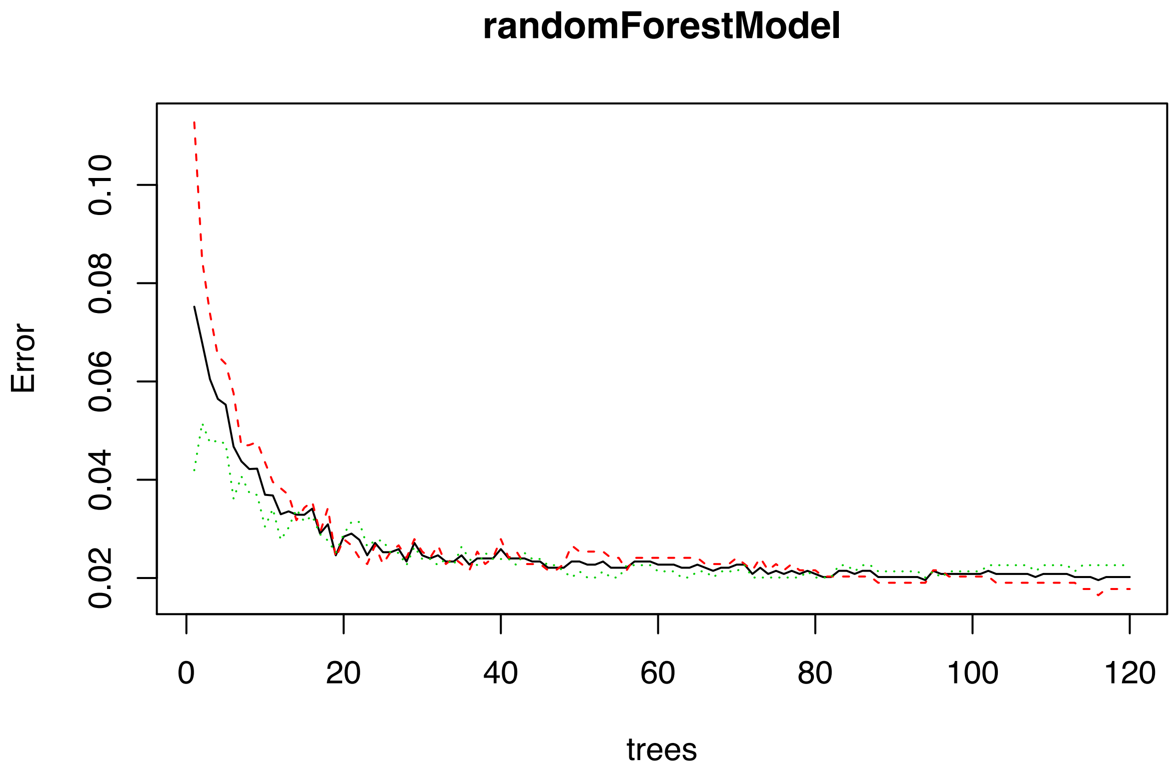
```
## [1] 0.6833023
```

```
#------------------------------------------ Random forest -----------------------------------
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

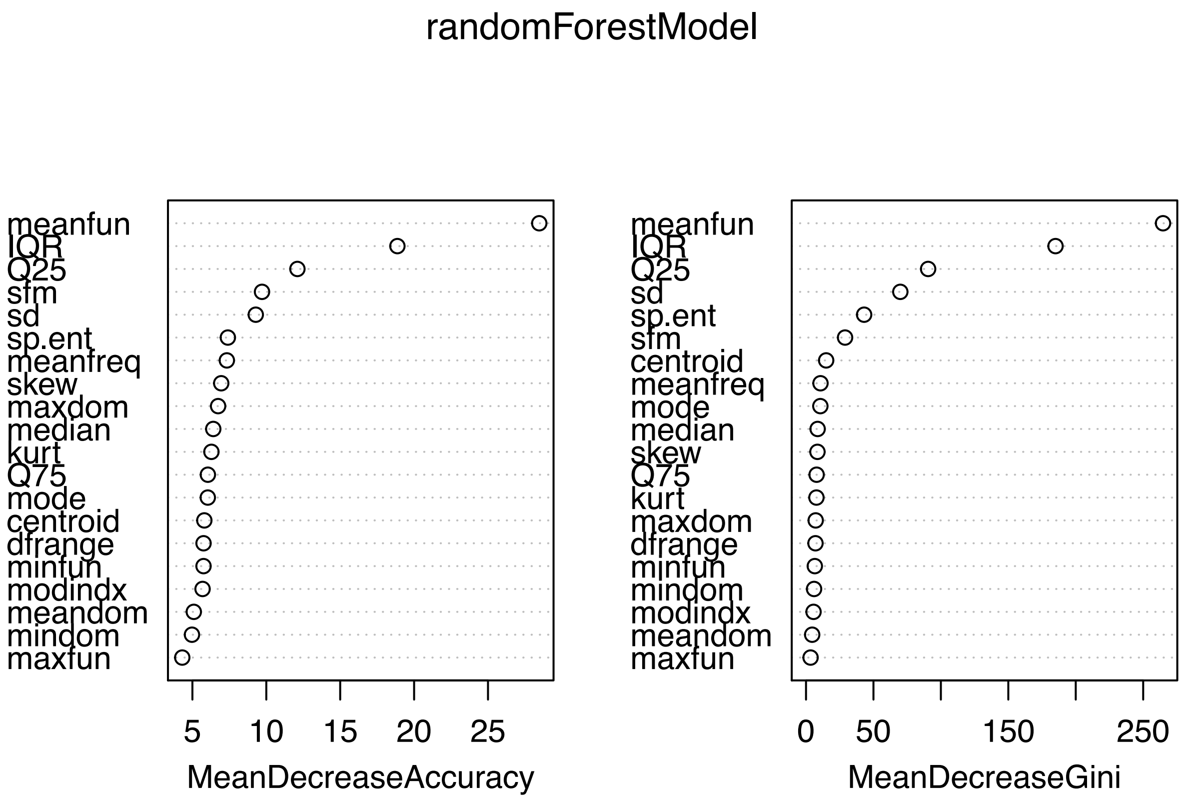
```
randomForestModel <- randomForest(label~., train, ntree=120, importance=T)
plot(randomForestModel)
```




```r
test_predict <- predict(randomForestModel, valid)
# Use for accuracy measurement
table(test_predict, valid_label)
```

```
##              valid_label
## test_predict female male
##       female    777   17
##       male       19  771
```

```r
# Select important predictors
varImpPlot(randomForestModel)
```




```
#-------------------------- Fit SVM on training set ----------------------------------
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(ggplot2)
library(e1071)
control <- trainControl(method="cv", number=12)
metric <- "Accuracy"

# Linear Kernel
model.svm <- train(label~., data=train, method="svmLinear", metric=metric, trControl=control)
```

```
## Loading required package: kernlab

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```r
prediction.svm <- predict(model.svm, train)
confusionMatrix(prediction.svm, train$label)$overall[1]
```

```
##  Accuracy
## 0.9766414
```

```r
model.svm <- train(label~., data=train, method="svmLinear", metric=metric, trControl=control)
prediction.svm <- predict(model.svm, valid)
confusionMatrix(prediction.svm, valid$label)$overall[1]
```

```
##  Accuracy
## 0.9728535
```

```r
# Radial Kernel
model.svm <- train(label~., data=train, method="svmRadial", metric=metric, trControl=control)
prediction.svm <- predict(model.svm, train)
confusionMatrix(prediction.svm, train$label)$overall[1]
```

```
##  Accuracy
## 0.9873737
```

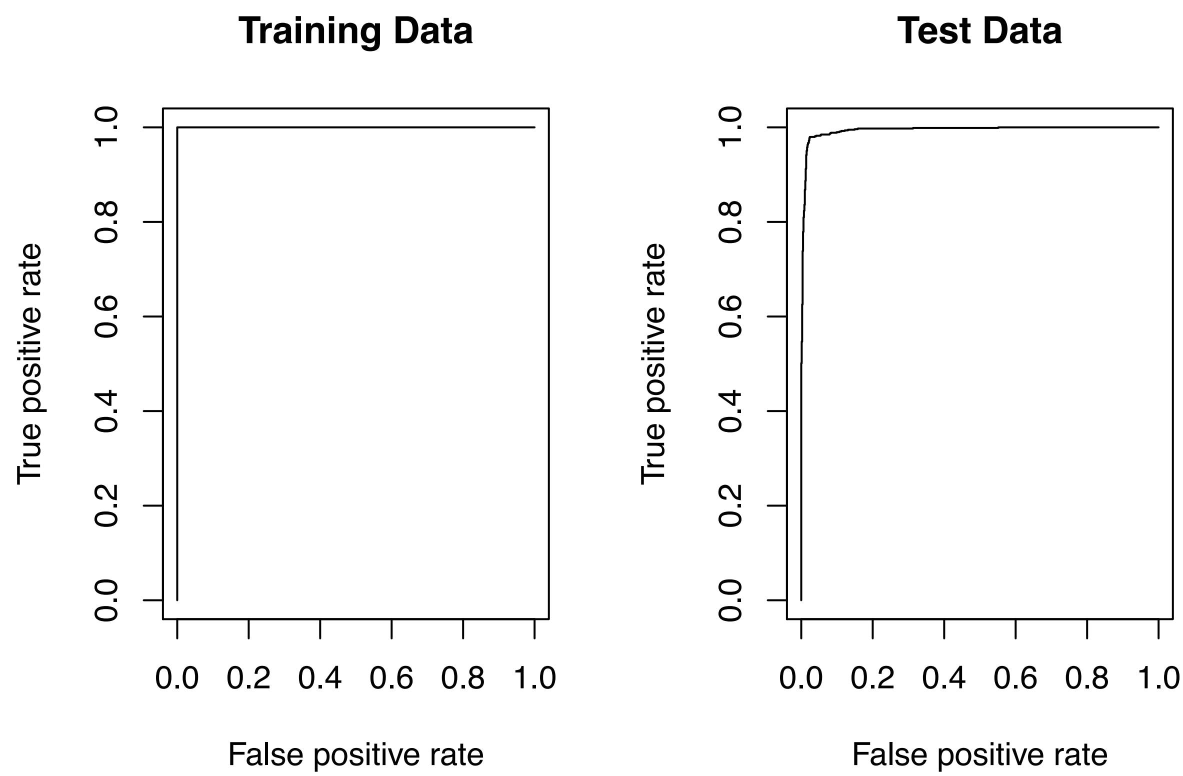
```r
model.svm <- train(label~., data=train, method="svmRadial", metric=metric, trControl=control)
prediction.svm <- predict(model.svm, valid)
confusionMatrix(prediction.svm, valid$label)$overall[1]
```

```
##  Accuracy
## 0.9804293
```

```r
# ROC
# Training set
library(ROCR)
rocplot=function(pred, truth, ...){
  predob = prediction(pred, truth)
  pref = performance(predob, "tpr", "fpr")
  plot(pref,...)
}
svmfit.opt=svm(label~., data=train, kernel="radial", gamma=1, cost=1, decision.values=T)
fitted=attributes(predict(svmfit.opt, train, decision.values=TRUE))$decision.values
par(mfrow=c(1, 2))
rocplot(fitted, train$label, main="Training Data")
pred = prediction(fitted, train$label)
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

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

```r
# Validation set
fitted=attributes(predict(svmfit.opt, valid, decision.values=TRUE))$decision.values
rocplot(fitted, valid$label, main="Test Data")
```




```
pred = prediction(fitted, valid$label)
# print out the area under the curve
unlist(attributes(performance(pred, "auc"))$y.values)
```

```
## [1] 0.9939896
```

Author	Codes	Reports
Xinyan Deng	KNN, random forest	KNN & random forest methods and result; final edit.
Qian Shen	Logistic Regression, QDA	Logistic Regression, & QDA methods and result; discussion.
Shiyu Wang	LDA, SVM	LDA, & SVM methods and result; introduction and abstract.