GENDER RECOGNITION BY VOICE

ANALYTICS FOR BUSINESS INTELLIGENCE – SPRING 2017 BATCH

PROFESSOR- Dr. JAIDEEP VAIDYA

TEAM 1:

BHUWAN AGARWAL
NAVEEN DAYAKAR
KHAVYA RAMACHANDRAN
JAYADURGA NAGARAJAN

INTRODUCTION:

Determining a person's gender as male or female, based upon a sample of their voice seems to be an easy task. The human ear can easily detect the difference between a male or female voice within the first few spoken words. However, designing a computer program to do this turns out to be a bit trickier.

This report describes the design of a computer program to model acoustic analysis of voices and speech for determining gender. The model is constructed using 3,168 recorded samples of male and female voices, speech, and utterances. The samples are processed using acoustic analysis and then plotted into a dataset. The resulting program achieves 98.4% accuracy on the test set.

As intelligent systems are being implemented for authentication, hands on applications, recognition, etc., the use of an efficient classifier is mandatory. We primarily focus to classify a human voice as a male or female based on the acoustics of voice. Features like resonance, fundamental mean frequency, pitch, modulation play a vital role in identifying the human voice. The pitch of female voice is greater than that of a male voice.

MISSION:

The main analysis involves classifying the response variable into two groups as either male or female using various classifiers and then projected the classifier which has the best accuracy of classifying the dataset.

IMPORTANCE OF THIS DATASET:

The dataset was used to identify a given voice as female or male based on certain acoustic properties of a human voice. The human voice frequency ranges between 0 hz-280 hz.

Gender classification is useful in speech and speaker recognition. Better performance has been reported when gender-dependent acoustic-phonetic models are used by decreasing the word error rate of a baseline speech recognition system by 1.6%. Mobile Apps like siri, tile, etc., make use of the voice recognition for their response.

More security than traditional authentication methods- passwords, patterns, etc.

Other applications, such as HCI, passive surveillance and smart living environment.

DATASET:

The dataset that is used for the analysis of the gender recognition using voice has 20 attributes.The data attributes and the values have been analyzed using various classifiers and models. The data set was extracted by converting a .wav file using the predefined R function specan(). This resulted in the formation of the below attributes of the dataset.

It consists of various attributes:

- meanfreq: mean frequency (in kHz)
- sd: standard deviation of frequency
- median: median frequency (in kHz)
- Q25: first quantile (in kHz)
- Q75: third quantile (in kHz)
- IQR: interquantile range (in kHz)
- skew: skewness (see note in specprop description)
- kurt: kurtosis (see note in specprop description)
- sp.ent: spectral entropy
- sfm: spectral flatness
- mode: mode frequency
- centroid: frequency centroid (see specprop)
- peakf: peak frequency (frequency with highest energy)
- meanfun: average of fundamental frequency measured across acoustic signal
- minfun: minimum fundamental frequency measured across acoustic signal
- maxfun: maximum fundamental frequency measured across acoustic signal
- meandom: average of dominant frequency measured across acoustic signal
- mindom: minimum of dominant frequency measured across acoustic signal
- maxdom: maximum of dominant frequency measured across acoustic signal
- dfrange: range of dominant frequency measured across acoustic signal
- modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
- label: male or female

Using these attributes we can predict the class variable label - male/female.

We partition the dataset into two datasets- one for testing and one for training:-

DATA USED FOR TRAINING: 2376 records

DATA USED FOR TESTING: 792 records

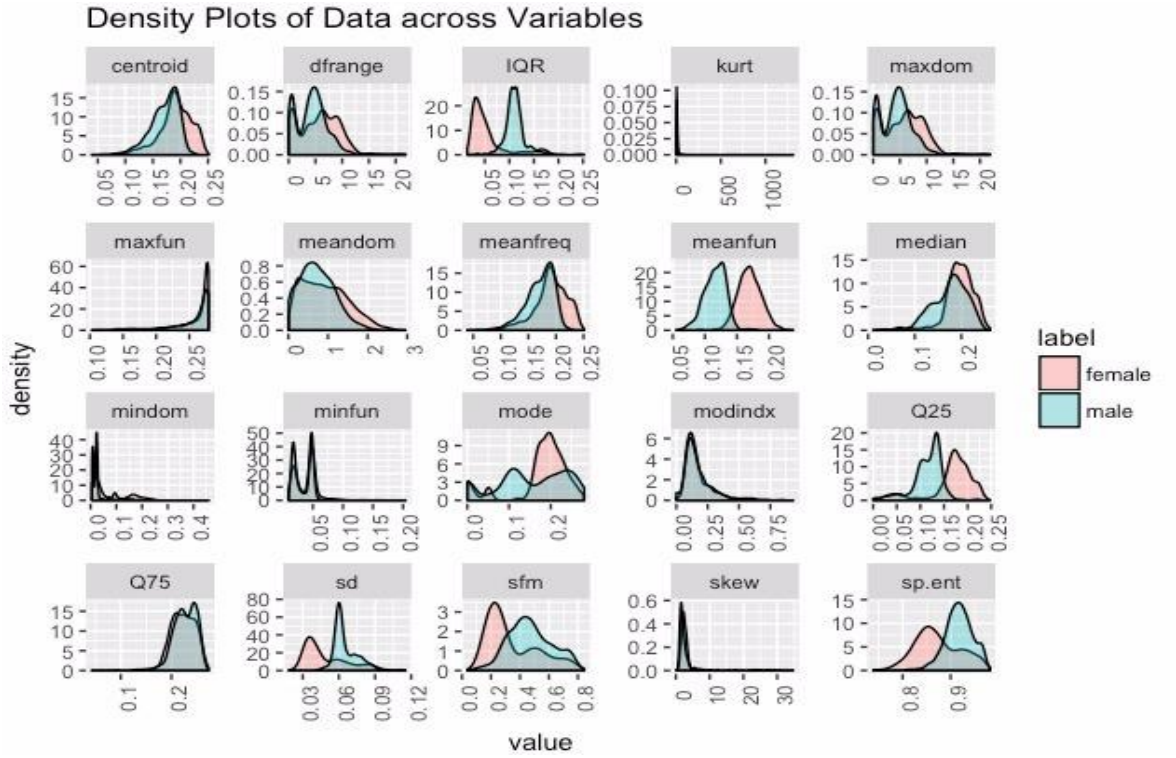
The following classifiers were used to predict the best possible classifier

- Linear discriminant analysis
- Classification and Regression trees (using R)
- KNN
- SVM
- Random Forest
- Naïve Bayes
- C5.0 decision tree

DATA PREPROCESSING:

1. There are no missing values observed in the dataset.
2. There were no duplicates in the data.
3. Dimensionality reduction: Check for correlation to remove irrelevant attributes
4. Standardization: dataset is already numeric, scaling is not required.

Following analysis was done by considering correlation and comparing it with the analysis of dataset without correlation.



The above visualization represents the overlapping of both male and female classes for each of the attributes. According to this plot the attributes meanfreq, meanfun, IQR, SD, spectral entrap are of significance. Some of the attributes have overlapping plots while other attributes are completely independent curves for male and female. The attributes that have more overlapping are of least importance.

C5.0 DST

	FEMALE(A)	MALE(A)
FEMALE(P)	386	16
MALE(P)	10	380

NB

	FEMALE(A)	MALE(A)
FEMALE(P)	350	40
MALE(P)	46	356

LDA

	FEMALE(A)	MALE(A)
FEMALE(P)	387	7
MALE(P)	9	389

RF

	FEMALE(A)	MALE(A)
FEMALE(P)	389	15
MALE(P)	7	381

SVM

	FEMALE(A)	MALE(A)
FEMALE(P)	388	13
MALE(P)	8	383

KNN

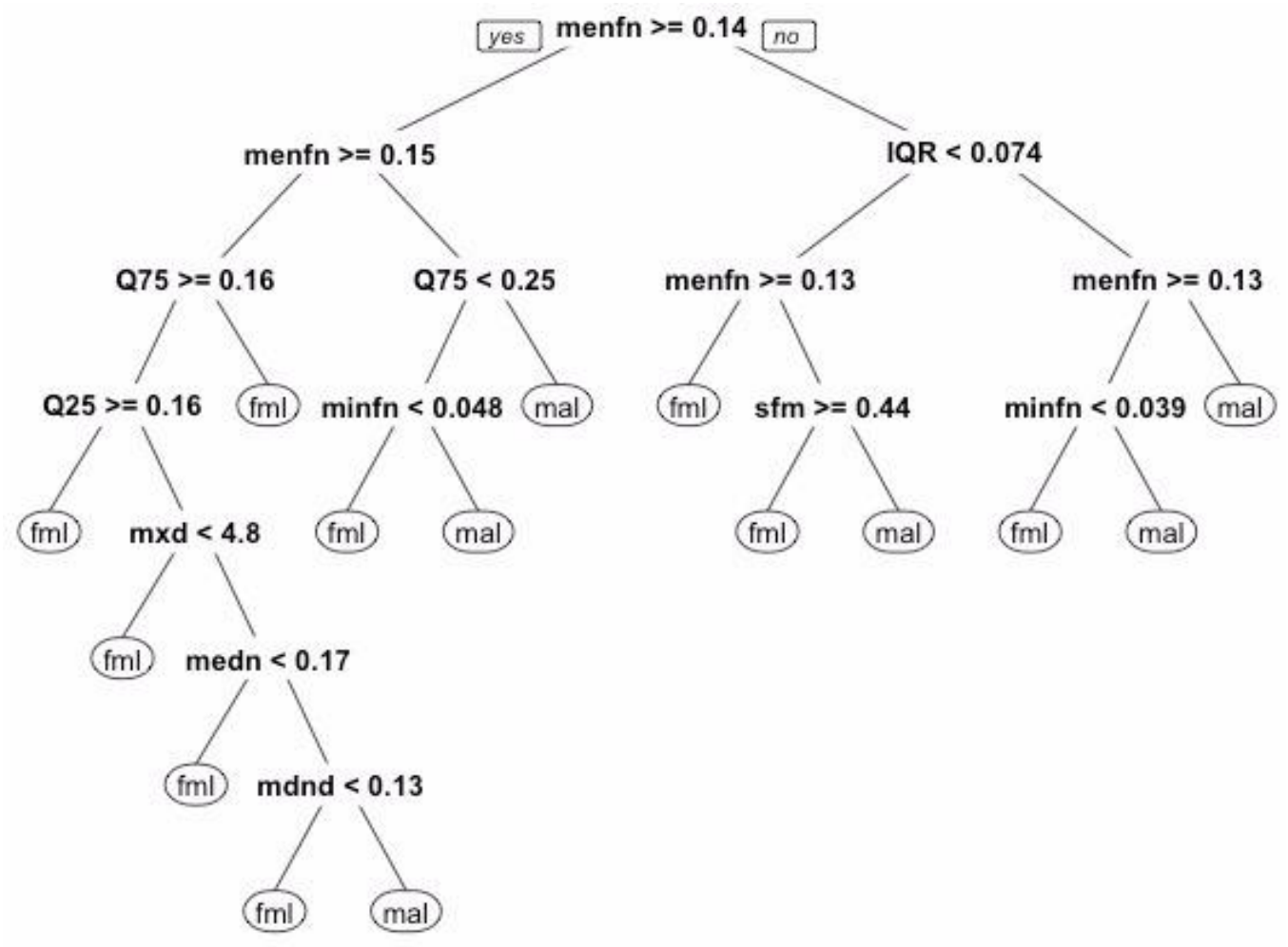
	FEMALE(A)	MALE(A)
FEMALE(P)	278	104
MALE(P)	118	292

CART

	FEMALE(A)	MALE(A)
FEMALE(P)	381	22
MALE(P)	15	374

The tree provides a clear visualization of the important attributes and how they classify a record as Male/Female.

The decision tree represents the various attributes. From the below tree representation, we are able to identify the significant attributes from the overall attributes.

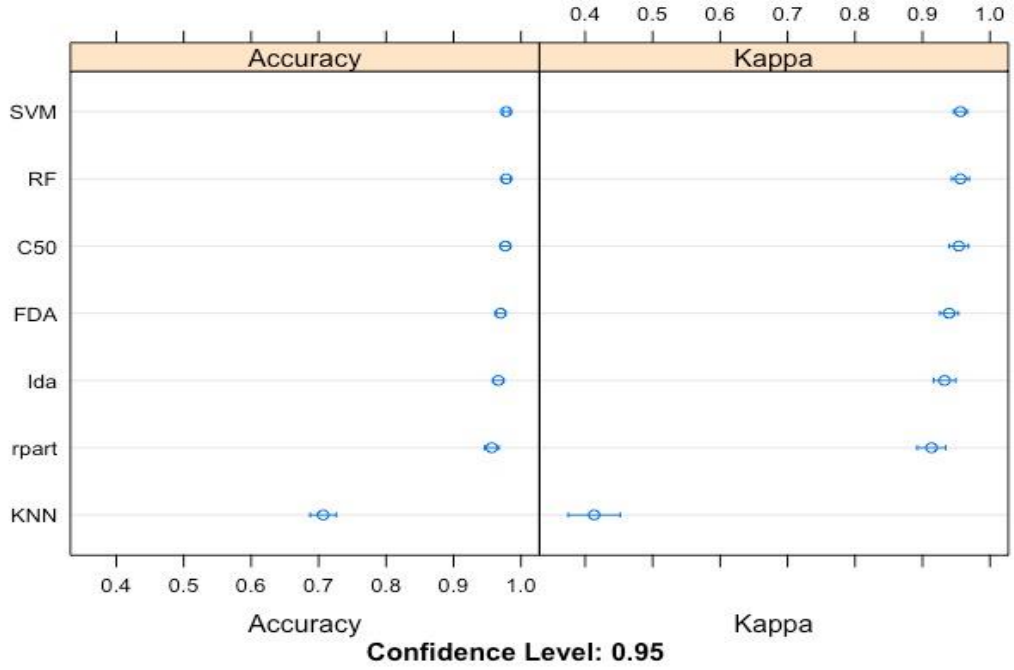


ACCURACY COMPARISON:

The above table clearly indicates the following:

- NB and KNN have low accuracy measure.
- SVM radial has the highest accuracy measure.

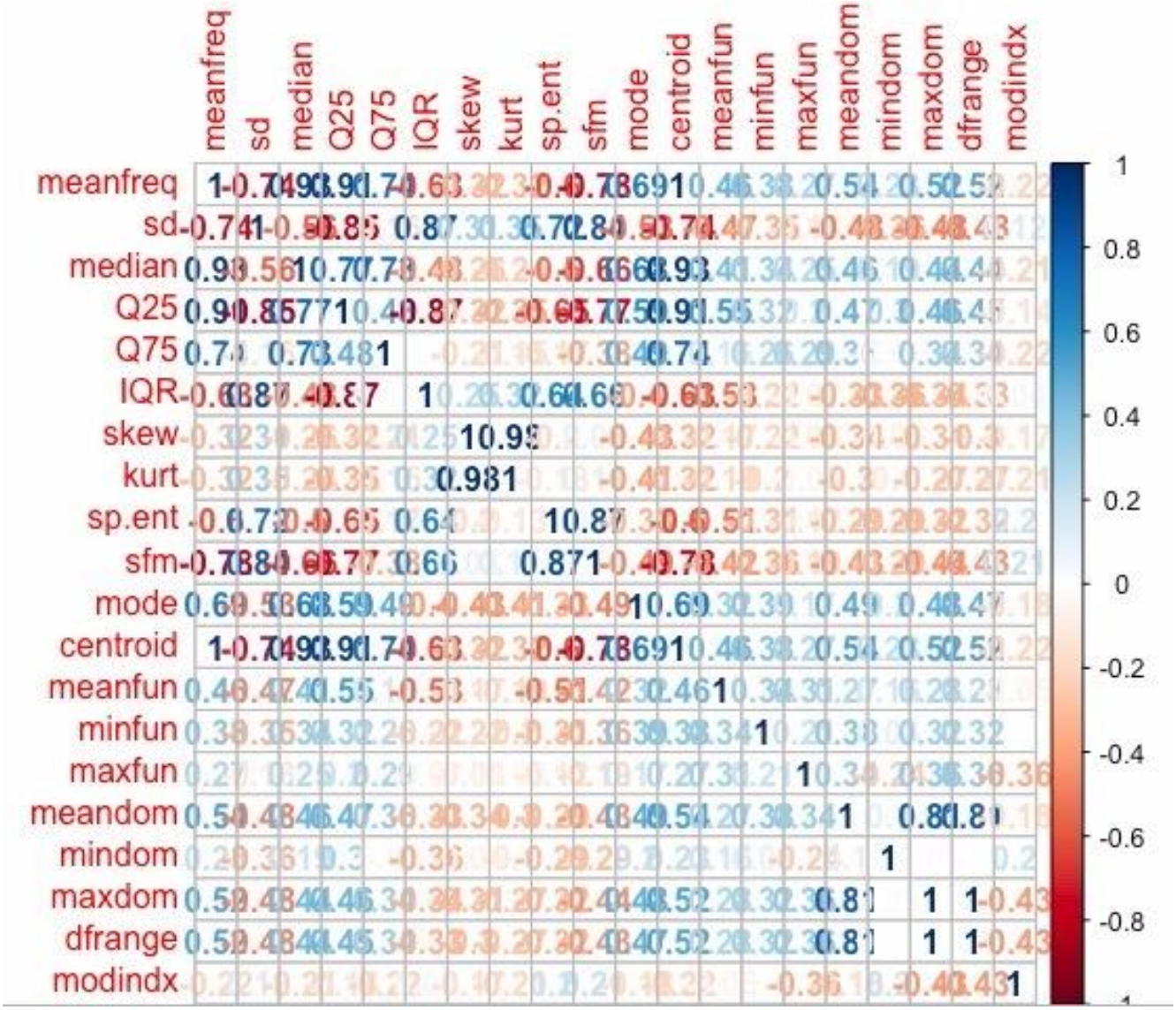
	lda	rpart	NB	C5.0t	KNN	SVMRadial
Accuracy	0.9570707	0.9532828	0.8914141	0.9671717	0.719697	0.9848485



The accuracy for SVM(Radial) classifier is 98.4% when compared to all other classifiers. So by analyzing all the above classifiers and using the confusion matrix and the accuracy measure of the prominent attributes, we select SVM(Radial) as the accurate classifier as seen in the above dot plot.

CORRELATION PLOT:

Correlation is the interaction between the attributes. We need to analyze the dataset if such an interaction should be considered or ignored and how it affects the classification. Based on this, we can concentrate on classification.



Blue - positive correlation
Red - negative correlation

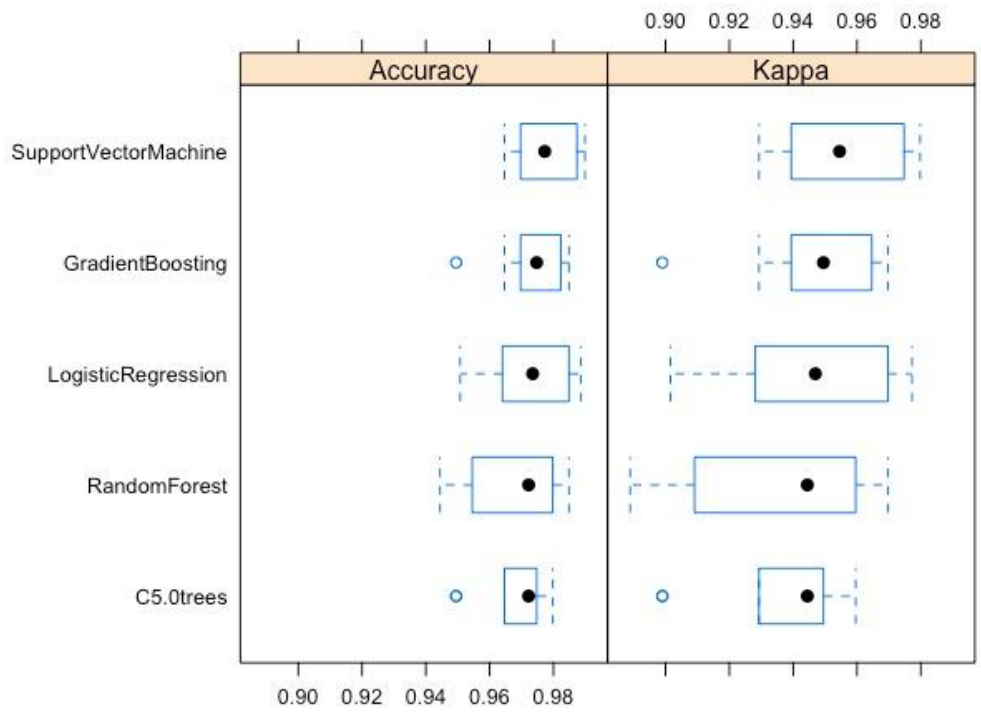
Snippet of the new dataset:

	label	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
1	male	-8.20722067	-2.1641467181	1.959474593	-5.45155625	0.930704048	-1.2019861940	-1.2391743442	-1.5554821905	1.814860868	1.124857e+00
2	male	-8.67052307	-3.8540181960	4.106559709	-5.07877296	0.411415208	-0.1201422060	-0.8800501911	-0.8998438056	0.390483419	1.804740e+00
3	male	-9.10973076	-4.5184269609	7.527063317	-3.46399606	-0.049157469	1.2191923660	-0.2681308127	-0.1872500865	0.457481639	2.178311e+00
4	male	-4.10861258	0.5622521482	-1.252939988	0.66756076	0.003348684	-0.9653865763	-0.3482840889	0.1932005818	0.575119192	1.030376e-01

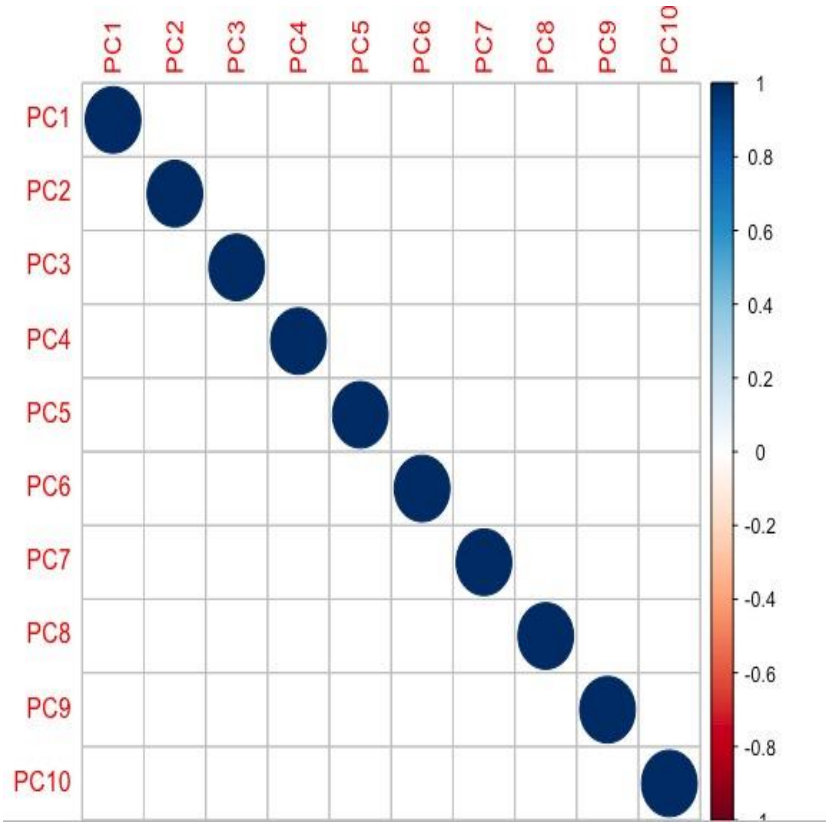
Now, we aim to analyze the classification again by removing the correlation. The motive is to find the classifier with the best accuracy. A new dataset was created where there was no correlation between attributes.

The new dataset contains 10 new attributes labeled as PC1 – PC10, in addition to the response attribute -label.

The below box plot shows the accuracy of all the classifiers used.



Correlation plot for the new dataset was plotted and we can find that there is no correlation between the PCA values. The attributes do not have any correlation with each other.

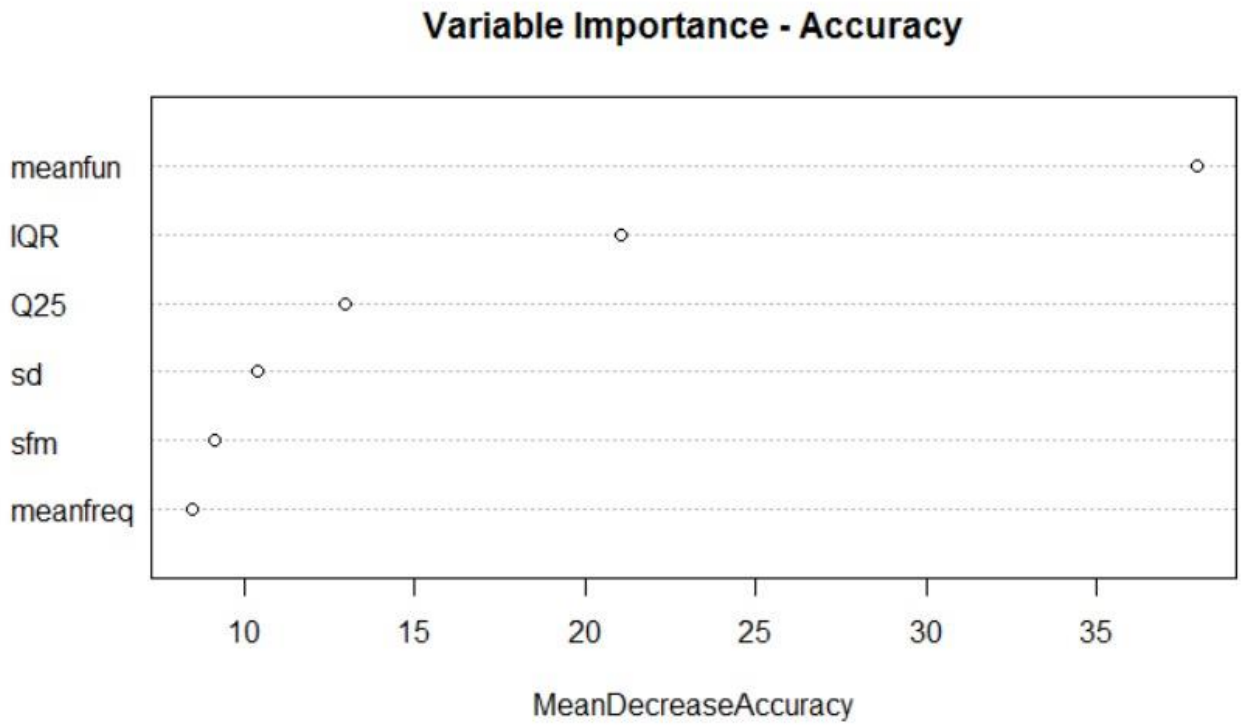


The above box plots show the accuracy for the various classifiers that have been used and we found that SVM has the highest accuracy when compared to the others. The accuracy for SVM was found out to be 97.6%

INFERENCE:

- From the above analysis we inferred that SVM is the best classifier for both the conditions as the accuracy is best in SVM classifier. Accuracy varies for all the classifiers after the correlation reduces.
- Especially SVM accuracy reduces after correlation.
- Hence original dataset with correlation is considered.
- Moving forward, SVM classification model is used for classification.

Random Forest function was used to find the most important features for classification.



The most significant attributes were plotted as above and we found that only 6 attributes were of most importance which were of most significance for classifying the data. Among the 20 attributes, it was found that only 6 of these are of most importance. Moving forward, we will use the top attributes only to predict the class labels.

The main reason to find the significant attributes is to classify the classes with least number of attribute values in hand. This is a form of dimension reduction and can make the classification process fast and efficient.

The significant attributes found are:

Mean fundamental frequency, IQR, Q25, standard deviation of frequency, spectrum flatness, and spectrum mean frequency

Using the significant attributes, we try to predict the classes for the original dataset
Using SVM Radial, the below confusion matrix and the accuracy were obtained.

	FEMALE(A)	MALE(A)
FEMALE(P)	389	7
MALE(P)	7	389

ACCURACY MEASURE: 0.9823232

With the usage of the significant attributes, the accuracy improved to 98%. The importance of finding the prominent features is, if there are missing values in any of the other attributes, they can be omitted as they are insignificant.

The most principal component is mean fundamental frequency.

CODE:

PRE-PROCESSING THE DATASET:

```
##Reading the dataset
voicedata <- read.csv("voice.csv")
View(voicedata)
library(caret)
library(kernlab)
library(adabag)
library(nnet)
library(e1071)
library(dplyr)
library(tidyr)
library(tree)
library(rpart.plot)
```

CHECK FOR MISSING VALUES:

```
> voicedata[!complete.cases(voicedata),]
 [1] meanfreq sd       median   Q25      Q75      IQR      skew     kurt     sp.ent   sfm
[11] mode     centroid meanfun  minfun   maxfun   meandom  mindom   maxdom   dfrange  modindx
[21] label
<0 rows> (or 0-length row.names)
```

OMITTING THE MISSING DATA:

```
voicedata1 <- na.omit(voicedata)
```

```
> voicedata[!complete.cases(voicedata),]
 [1] meanfreq sd       median   Q25      Q75      IQR      skew     kurt     sp.ent   sfm
[11] mode     centroid meanfun  minfun   maxfun   meandom  mindom   maxdom   dfrange  modindx
[21] label
<0 rows> (or 0-length row.names)
```

SPLITTING THE DATA INTO TEST AND TRAINING:

```
##--##--####--##--####--##--####--##--####--##--####--##--####--##--####--##--####--##--##
##Splitting the dataset
voice <- read.csv("voice.csv")
##data partition
index <- createDataPartition(voice$label, p = 0.75, list = FALSE)
test <- voicedata1[-index, ]
train <- voicedata1[index, ]

dim(test)
dim(train)
## y contains the class label which is the response attribute
x <- train[, 1:20]
y <- train[, 21]

##checking the dimensions and summary of the voice dataset
dim(voice)
str(voice)
summary(voice)
table(voice$label)
```

DENSITY PLOT OF ATTRIBUTES VERSUS LABEL:

```
## Density plot of attributes for male and female
voice %>% na.omit() %>%
  gather(type,value,1:20) %>%
  ggplot(aes(x=value,fill=label))+geom_density(alpha=0.3)+facet_wrap(~type,scales="free")
+theme(axis.text.x = element_text(angle = 90,vjust=1))
+labs(title="Density Plots of Data across Variables")
```

COLLECTING ALL SAMPLES AND COMPARING OVERALL ACCURACY:

```
# collect resamples
results <- (list(lda=lda.acc,rpart=rpart.acc, NB=nba.acc,C5.0t=c50.acc,KNN=knn.acc,SVMRadial=svm.acc))

# summarize the distributions
View(results)

model_results <- resamples(list(lda=model.lda, rpart=model.rpart,KNN=model.knn,SVM=model.svm,RF=model.rf,FDA=model.fda,C50=model.c50))
summary(model_results)
dotplot(model_results)

##overall accuracy is better for SVM
##SVM has the highest accuracy
```

ANALYSIS OF THE VARIOUS MODELS:

LDA:

```
> prediction.lda <- predict(model.lda, test)
> lda.acc=confusionMatrix(prediction.lda, test$label)$overall[1]
> confusionMatrix(prediction.lda, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    377    9
    male       19  387

               Accuracy : 0.9646
                 95% CI : (0.9493, 0.9764)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9293
 Mcnemar's Test P-Value : 0.08897

            Sensitivity : 0.9520
            Specificity : 0.9773
         Pos Pred Value : 0.9767
         Neg Pred Value : 0.9532
             Prevalence : 0.5000
         Detection Rate : 0.4760
   Detection Prevalence : 0.4874
      Balanced Accuracy : 0.9646

       'Positive' Class : female
```

RPART:

```
> model.rpart <- train(label~., data=train, method="rpart", metric=metric, trControl=control)
> prediction.rpart <- predict(model.rpart, test)
> rpart.acc=confusionMatrix(prediction.rpart, test$label)$overall[1]
> confusionMatrix(prediction.rpart, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    386   25
    male       10  371

               Accuracy : 0.9558
                 95% CI : (0.9391, 0.969)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9116
 Mcnemar's Test P-Value : 0.01796

            Sensitivity : 0.9747
            Specificity : 0.9369
         Pos Pred Value : 0.9392
         Neg Pred Value : 0.9738
             Prevalence : 0.5000
         Detection Rate : 0.4874
   Detection Prevalence : 0.5189
      Balanced Accuracy : 0.9558

       'Positive' Class : female
```

KNN:

```
> model.knn <- train(label~., data=train, method="knn", metric=metric, trControl=control)
> prediction.knn <- predict(model.knn, test)
> knn.acc=confusionMatrix(prediction.knn, test$label)$overall[1]
> confusionMatrix(prediction.knn, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    260  104
    male      136  292

               Accuracy : 0.697
                 95% CI : (0.6636, 0.7288)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.3939
 Mcnemar's Test P-Value : 0.04539

            Sensitivity : 0.6566
            Specificity : 0.7374
         Pos Pred Value : 0.7143
         Neg Pred Value : 0.6822
             Prevalence : 0.5000
         Detection Rate : 0.3283
   Detection Prevalence : 0.4596
      Balanced Accuracy : 0.6970

       'Positive' Class : female
```

SVM RADIAL:

```
> model.svm <- train(label~., data=train, method="svmRadial", metric=metric, trControl=control)
> prediction.svm <- predict(model.svm, test)
> svm.acc=confusionMatrix(prediction.svm, test$label)$overall[1]
> confusionMatrix(prediction.svm, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    385   10
    male       11  386

               Accuracy : 0.9735
                 95% CI : (0.9598, 0.9835)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.947
 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9722
            Specificity : 0.9747
         Pos Pred Value : 0.9747
         Neg Pred Value : 0.9723
             Prevalence : 0.5000
         Detection Rate : 0.4861
   Detection Prevalence : 0.4987
      Balanced Accuracy : 0.9735

       'Positive' Class : female
```

RANDOM FOREST:

```
> prediction.rf <- predict(model.rf, test)
> rf.acc=confusionMatrix(prediction.rf, test$label)$overall[1]
> confusionMatrix(prediction.rf, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    384    8
    male       12  388

               Accuracy : 0.9747
                 95% CI : (0.9613, 0.9845)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9495
 Mcnemar's Test P-Value : 0.5023

            Sensitivity : 0.9697
            Specificity : 0.9798
         Pos Pred Value : 0.9796
         Neg Pred Value : 0.9700
             Prevalence : 0.5000
         Detection Rate : 0.4848
   Detection Prevalence : 0.4949
      Balanced Accuracy : 0.9747

       'Positive' Class : female
```

FDA:

```
> model.fda <- train(label~., data=train, method="fda", metric=metric, trControl=control)
Loading required package: earth
Loading required package: plotmo
Loading required package: plotrix
Loading required package: TeachingDemos
Loading required package: mda
Loading required package: class
Loaded mda 0.4-9

> prediction.fda <- predict(model.fda, test)
> fda.acc=confusionMatrix(prediction.fda, test$label)$overall[1]
> confusionMatrix(prediction.fda, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    381    5
    male       15  391

               Accuracy : 0.9747
                 95% CI : (0.9613, 0.9845)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.9495
 Mcnemar's Test P-Value : 0.04417

            Sensitivity : 0.9621
            Specificity : 0.9874
         Pos Pred Value : 0.9870
         Neg Pred Value : 0.9631
             Prevalence : 0.5000
         Detection Rate : 0.4811
   Detection Prevalence : 0.4874
      Balanced Accuracy : 0.9747

       'Positive' Class : female
```

NAIVE BAYES:

```
> model.naiveBayes <- naiveBayes(label~., data=train, metric=metric, trControl=control)
> prediction.NB <- predict(model.naiveBayes, test)
> nba.acc=confusionMatrix(prediction.NB, test$label)$overall[1]
> confusionMatrix(prediction.NB, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    340   39
    male       56  357

               Accuracy : 0.8801
                 95% CI : (0.8554, 0.9019)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.7601
 Mcnemar's Test P-Value : 0.1007

            Sensitivity : 0.8586
            Specificity : 0.9015
         Pos Pred Value : 0.8971
         Neg Pred Value : 0.8644
             Prevalence : 0.5000
         Detection Rate : 0.4293
   Detection Prevalence : 0.4785
      Balanced Accuracy : 0.8801

       'Positive' Class : female
```

C 5.0

```
> model.c50 <- train(label~., data=train, method="C5.0", metric=metric, trControl=control)
Loading required package: C50
Loading required package: plyr
------------------------------------------------------------------------------
You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)
------------------------------------------------------------------------------

Attaching package: 'plyr'

The following objects are masked from 'package:dplyr':

    arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

> prediction.c50 <- predict(model.c50, test)
> c50.acc=confusionMatrix(prediction.c50, test$label)$overall[1]
> confusionMatrix(prediction.c50, test$label)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female    385    9
    male       11  387

               Accuracy : 0.9747
                 95% CI : (0.9613, 0.9845)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9495
 Mcnemar's Test P-Value : 0.8231

            Sensitivity : 0.9722
            Specificity : 0.9773
         Pos Pred Value : 0.9772
         Neg Pred Value : 0.9724
             Prevalence : 0.5000
         Detection Rate : 0.4861
   Detection Prevalence : 0.4975
      Balanced Accuracy : 0.9747

       'Positive' Class : female
```

SUMMARY OF ALL ABOVE CLASSIFIERS:

```
> model_results <- resamples(list(lda=model.lda, rpart=model.rpart,KNN=model.knn,SVM=model.svm,RF=model.rf,
FDA=model.fda,C50=model.c50))
> summary(model_results)

Call:
summary.resamples(object = model_results)

Models: lda, rpart, KNN, SVM, RF, FDA, C50
Number of resamples: 12

Accuracy
        Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
lda   0.9545  0.9646 0.9747 0.9714  0.9798 0.9798    0
rpart 0.9343  0.9444 0.9646 0.9609  0.9722 0.9848    0
KNN   0.6566  0.6768 0.6995 0.7066  0.7361 0.7677    0
SVM   0.9646  0.9823 0.9848 0.9827  0.9861 0.9949    0
RF    0.9646  0.9697 0.9798 0.9798  0.9899 0.9949    0
FDA   0.9545  0.9646 0.9722 0.9722  0.9760 0.9899    0
C50   0.9646  0.9785 0.9823 0.9806  0.9848 0.9899    0

Kappa
        Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
lda   0.9091  0.9293 0.9495 0.9428  0.9596 0.9596    0
rpart 0.8687  0.8889 0.9293 0.9217  0.9444 0.9697    0
KNN   0.3131  0.3535 0.3990 0.4133  0.4722 0.5354    0
SVM   0.9293  0.9646 0.9697 0.9655  0.9722 0.9899    0
RF    0.9293  0.9394 0.9596 0.9596  0.9798 0.9899    0
FDA   0.9091  0.9293 0.9444 0.9444  0.9520 0.9798    0
C50   0.9293  0.9571 0.9646 0.9613  0.9697 0.9798    0
```

PCA TO REMOVE CORRELATION:

```
### pre-process of original dataset to remove correlations
pca_Transform <- preProcess(voicedata1,method=c("scale","center","pca"))
voicedata2 <- predict(pca_Transform,voicedata1)
View(voicedata2)
head(voicedata2)
```

CORRELATION PLOT AFTER REMOVING CORRELATION:

```
### visual explorations
# correlation plot
new_Corr <- cor(voicedata2[,2:11])
corrplot(new_Corr)
```

NEW DATASET IS THE DATASET HAVING 10 ATTRIBUTES ONLY (DATASET AFTER REMOVING ALL THE CORRELATED ATTRIBUTES).

GLM APPLIED ON NEW DATASET:

```
> glm_Model <- train(
+   model_Formula,
+   data=voicedata2,
+   method="glm",
+   trControl=modelControl
+ )
> voice_Test$glmPrediction <- predict(glm_Model,newdata=voice_Test[,2:11])
> glm_Model   ### accuracy 0.9709 kappa 0.9418
Generalized Linear Model

3168 samples
  10 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2904, 2904, 2904, 2904, 2904, 2904, ...
Resampling results:

  Accuracy   Kappa
  0.9731692  0.9463384

> table(voice_Test$label,voice_Test$glmPrediction)

        female male
  female    382   14
  male        4  392
> glm.acc=confusionMatrix(voice_Test$glmPrediction,voice_Test$label)$overall[1]
> glm.acc
 Accuracy
0.9772727
```

RANDOM FOREST ON NEW DATASET:

```
> rf_Model <- train(
+   model_Formula,
+   data=voice_Train,
+   method="rf",
+   ntrees=1000,
+   trControl=modelControl
+ )
> voice_Test$rfPrediction <- predict(rf_Model,newdata=voice_Test[,2:11])
> rf_Model  ## accuracy 0.967 kappa 0.934
Random Forest

2376 samples
  10 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2178, 2178, 2178, 2178, 2178, 2178, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9696970  0.9393939
   6    0.9684343  0.9368687
  10    0.9617003  0.9234007

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 2.
> table(voice_Test$label,voice_Test$rfPrediction)

        female male
  female    387    9
  male       11  385
> RF.acc=confusionMatrix(voice_Test$rfPrediction,voice_Test$label)$overall[1]
> RF.acc
 Accuracy
0.9747475
```

SVM ON NEW DATASET:

```
> svm_Model <- train(
+   model_Formula,
+   data=voice_Train,
+   method="svmRadial",
+   trControl=modelControl
+ )
> svm_Model  ## accuracy 0.974 kappa 0.949
Support Vector Machines with Radial Basis Function Kernel

2376 samples
  10 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2178, 2178, 2178, 2178, 2178, 2178, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9760101  0.9520202
  0.50  0.9764310  0.9528620
  1.00  0.9781145  0.9562290

Tuning parameter 'sigma' was held constant at a value of 0.08443224
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.08443224 and C = 1.
> voice_Test$svmPrediction <- predict(svm_Model,newdata=voice_Test[,2:11])
> table(voice_Test$label,voice_Test$svmPrediction)

        female male
  female    391    5
  male       10  386
> svm.acc=confusionMatrix(voice_Test$svmPrediction,voice_Test$label)$overall[1]
> svm.acc
 Accuracy
0.9810606
```

GBM GRADIENT BOOSTING ON NEW DATASET:

```
Stochastic Gradient Boosting

2376 samples
  10 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2178, 2178, 2178, 2178, 2178, 2178, ...
Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa
  1                   50      0.9238215  0.8476431
  1                  100      0.9511785  0.9023569
  1                  150      0.9612795  0.9225589
  2                   50      0.9562290  0.9124579
  2                  100      0.9638047  0.9276094
  2                  150      0.9701178  0.9402357
  3                   50      0.9595960  0.9191919
  3                  100      0.9709596  0.9419192
  3                  150      0.9709596  0.9419192

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter
 'n.minobsinnode' was held constant at a value of 10
Accuracy was used to select the optimal model using  the largest value.
The final values used for the resamples(x, ...) .trees = 100, interaction.depth = 3, shrinkage = 0.1
 and n.minobsinnode = 10.
> voice_Test$gbmPrediction <- predict(gbm_Model,newdata=voice_Test[,2:11])
> table(voice_Test$label,voice_Test$gbmPrediction)

        female male
  female    385   11
  male       12  384
> gbm.acc=confusionMatrix(voice_Test$gbmPrediction,voice_Test$label)$overall[1]
> gbm.acc
 Accuracy
0.9709596
```

C 5.0 ON NEW DATASET:

```
> c50_Model <- train(
+   model_Formula,
+   data=voice_Train,
+   method="C5.0",
+   trControl=modelControl
+ )
> c50_Model  ## best performance @ accuracy 0.968 kappa 0.935
C5.0

2376 samples
  10 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2178, 2178, 2178, 2178, 2178, 2178, ...
Resampling results across tuning parameters:

  model  winnow  trials  Accuracy   Kappa
  rules  FALSE     1      0.9486532  0.8973064
  rules  FALSE    10      0.9705387  0.9410774
  rules  FALSE    20      0.9722222  0.9444444
  rules   TRUE     1      0.9511785  0.9023569
  rules   TRUE    10      0.9701178  0.9402357
  rules   TRUE    20      0.9688552  0.9377104
  tree   FALSE     1      0.9339226  0.8678451
  tree   FALSE    10      0.9663300  0.9326599
  tree   FALSE    20      0.9671717  0.9343434
  tree    TRUE     1      0.9385522  0.8771044
  tree    TRUE    10      0.9692761  0.9385522
  tree    TRUE    20      0.9684343  0.9368687

Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were trials = 20, model = rules and winnow = FALSE.
> voice_Test$c50Prediction <- predict(c50_Model,newdata=voice_Test[,2:11])
> table(voice_Test$label,voice_Test$c50Prediction)

        female male
  female    386   10
  male       10  386
> c50.acc=confusionMatrix(voice_Test$c50Prediction,voice_Test$label)$overall[1]
> c50.acc
 Accuracy
0.9747475
```

COMPARING CLASSIFIERS ON NEW DATASET:

```
### compare model performance of 4 models that have been built
model_Comparison <- resamples(
  list(
    LogisticRegression=glm_Model,
    RandomForest=rf_Model,
    SupportVectorMachine=svm_Model,
    GradientBoosting=gbm_Model,
    C5.0trees=c50_Model
  )
)

summary(model_Comparison)

## visual comparison of model performances
bwplot(model_Comparison,layout=c(2,1))
```

VARIABLE IMPORTANCE PLOT:

```
> voice_Train$svmPrediction<-predict(svm_Model,newdata=train[,1:20])
> table(test$label,voice_Test$svmPrediction)

        female male
  female    391    5
  male       10  386
> svm.acc=confusionMatrix(voice_Test$svmPrediction,test$label)$overall[1]
> svm.acc
 Accuracy
0.9810606
> set.seed(100)
> control <- trainControl(method="cv", number=12)
> metric <- "Accuracy"
> voice$label=as.factor(voice$label)
> idx=createDataPartition(voice$label,p=0.75,list=FALSE)
> train_data=voice[idx,]
> test_data=voice[-idx,]
> library(randomForest)
> index <- createDataPartition(voice$label, p = 0.75, list = FALSE)
> test <- voice[-index, ]
> train <- voice[index, ]
> x <- train[, 1:20]
> y <- train[, 21]
> table(train$label)/nrow(train)

female   male
   0.5    0.5
> table(test$label)/nrow(test)

female   male
   0.5    0.5
> set.seed(3)
> model <- randomForest(label~., train, ntree = 120, importance = T)
> plot(model)
> varImpPlot(model, sort = T, main="Variable Importance - Accuracy", n.var=6, type = 1)
```

FINAL SVM ONLY ON IMPORTANT COMPONENTS:

```
> svm_Model <- train(
+   model_Formula,
+   data=train,
+   method="svmRadial",
+   trControl=modelControl
+ )
> svm_Model  ## accuracy 0.974 kappa 0.949
Support Vector Machines with Radial Basis Function Kernel

2376 samples
   6 predictor
   2 classes: 'female', 'male'

No pre-processing
Resampling: Cross-Validated (12 fold)
Summary of sample sizes: 2178, 2178, 2178, 2178, 2178, 2178, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9726431  0.9452862
  0.50  0.9743266  0.9486532
  1.00  0.9772727  0.9545455

Tuning parameter 'sigma' was held constant at a value of 0.3909491
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.3909491 and C = 1.
> voice_Train$svmPrediction<-predict(svm_Model,newdata=train[,1:20])
> table(test$label,voice_Test$svmPrediction)

        female male
  female    391    5
  male       10  386
> svm.acc=confusionMatrix(voice_Test$svmPrediction,test$label)$overall[1]
> svm.acc
 Accuracy
0.9810606
```

CONCLUSION:

- We have identified SVM to be the best classifier for the dataset.
- The accuracy was identified to be 98.6%
- Removing correlation among the attributes actually brought down the accuracy as compared to the original dataset. However due to the reduction in accuracy, we decided to use the original dataset with correlation for classifying the various classes.
- PCA was then used to identify the principle components.
- The primary attributes that were identified for determining the class as either male or female were:
    - Mean fundamental frequency
    - IQR
    - Q25
    - standard deviation of frequency
    - spectrum flatness
    - spectrum mean frequency
- The results obtained using the above primary attributes resulted in greater accuracy of classifying the data.
- Looking at the mean fundamental frequency might be enough to accurately classify a voice. However, some male voices use a higher frequency, even though their resonance differs from female voices, and may be incorrectly classified as female.

FUTURE SCOPE:

- With growing technology, many applications that use voice recognition are being built.
- Various companies can use this technology and improve it in a way that recognizes each and every human beings voice uniquely.
- The technology can be used to bring a great deal of security in all applications.