EED 363: Applied Machine Learning

Topic: Voice based Gender Identification

1. Introduction:

With the speech-based recognition getting more and more powerful and used in almost in major products and available in each mobile phone, the Gender identification by voice will just add to the attraction and will be very useful to employ gender-dependent model. Further it can help in reducing computation to half for voice recognition, automation of emotion recognition, tagging and to do all this recognizing the gender is very important. With age of Internet and automation doing the repetitive work is considered useless. It's best to make the machine learn and able to reduce the load. This combined with the voice recognition can be used in Call Centre, support groups and the options keep going on.

In this project we look into how with the voice we can make machine distinguish between Male or Female voice.

2. Data Description:

The database was built using thousands of samples of male and female voices, each labeled by their gender of male or female. Each voice sample is stored as a .WAV file, which is then pre-processed for acoustic analysis using the specan function from the WarbleR R package. Specan measures 22 acoustic parameters on acoustic signals for which the start and end times are provided. The output from the pre-processed WAV files were saved into a CSV file, containing 3168 rows and 21 columns (20 columns for each feature and one label column for the classification of male or female).

Note: The features for duration and peak frequency (peakf) were removed from training. Duration refers to the length of the recording, which for training, is cut off at 20 seconds. Peakf was omitted from calculation due to time and CPU constraints in calculating the value. In this case, all records will have the same value for duration (20) and peak frequency (0).

No. of columns (X and Y)	21 (20 Input, 1 Output)
No. of rows (No. of Samples/Instances)	3168

Missing Value	0
Date of Publishing	22nd June 2016
Output (Binary Classification)	Male or Female

Table 1: Data Set Description

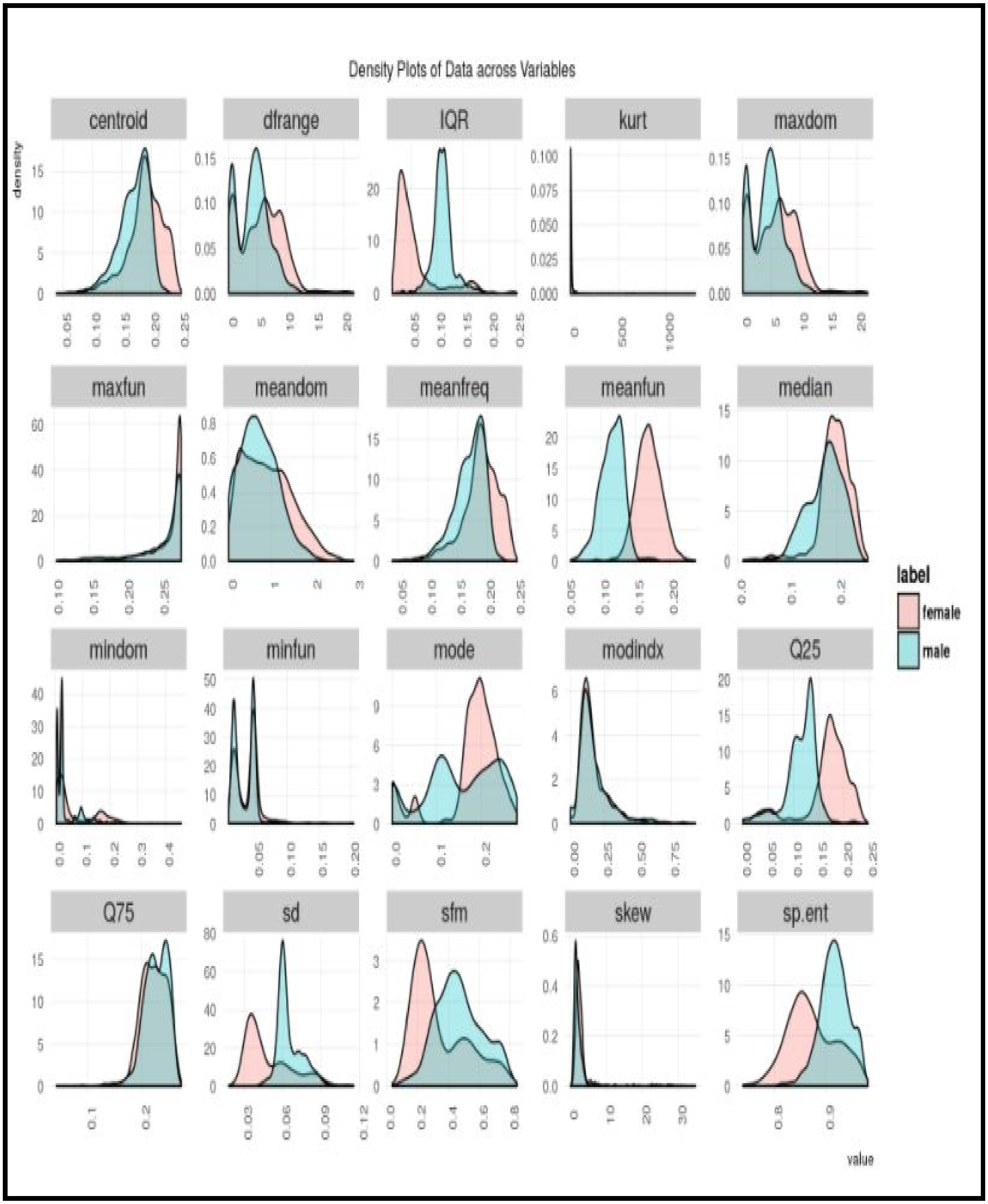
The following are the feature vectors:

SNO.	**Feature**	**Feature Information**	**UNIT**
1	meanfreq	Mean frequency	KHz
2	sd	Standard Deviation of frequency	Unit less
3	Median	Median frequency	kHz
4	Q25	First quartile	kHz
5	Q75	Third quartile	kHz
6	IQR	Interquartile range	kHz
7	skew	Skewness	Unit less
8	kurt:	Kurtosis	Unit less
9	sp.ent:	Spectral entropy	
10	sfm	Spectral flatness	
11	mode	Mode frequency	kHz
12	centroid	Frequency centroid	KHz
13	meanfun	Average of fundamental frequency measured across acoustic signal	kHz

14	minfun	Minimum fundamental frequency measured across acoustic signal	kHz
15	maxfun	Maximum fundamental frequency measured across acoustic signal	kHz
16	meandom	Average of dominant frequency measured across acoustic signal	kHz
17	mindom	Minimum of dominant frequency measured across acoustic signal	kHz
18	maxdom	Maximum of dominant frequency measured across acoustic signal	kHz
19	dfrange	Range of dominant frequency measured across acoustic signal	kHz
20	modindx	Modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range	Unit less
21	Label (OUTPUT)	Male or Female	Binary Classification

Table 2: Feature Information

Looking at Density plot of data across different features for male and female data samples.



3. Initial application of Algorithms on raw data:

The date is scaled to [0, 1] using Min-Max method. Since the data had real values 'binning' is used to get discrete values in features for Naïve Bayes. The data is subjected to Naïve Bayes(Self written script to control the number of bins and In-built), KNN (Self written and In-built with k varying between 3 and 5) and Logistic Regression (Self-written script) the result of these are present in Table 4. The data is divided in Training and test set in following manner for all 10 fold cross validation:

Training Set	2112 Instances
Test Set	1056 Instances

Table 3: Data Set division

Type of Classifier	**Source**	**Parameter**	**Accuracy**
Naïve Bayes	Self-written	4 equal bin b/w [0,1]	89.32
		8 equal bin b/w [0,1]	88.50
	MATLAB	4 equal bin	84.45
		8 equal bin	81.17
KNN	Self-written	K=3	77.08
		K=5	86.36
	MATLAB	K=3	92.81
		K=5	94.12
Logistic Regression	Self-written	Cost threshold=0.01	91.92
		Cost threshold=0.001	93.68
		Cost threshold=0.0001	94.94

Table 4: **Accuracies mentioned above are average of all 10 fold cross validation.*

Conclusion from above:

- These are just initial results for our dataset, now we will do data analysis to reduce our computational cost and hopefully increase accuracy.

For further analysis we will be using only the Self written Naïve Bayes, Logistic Regression and In-built KNN.

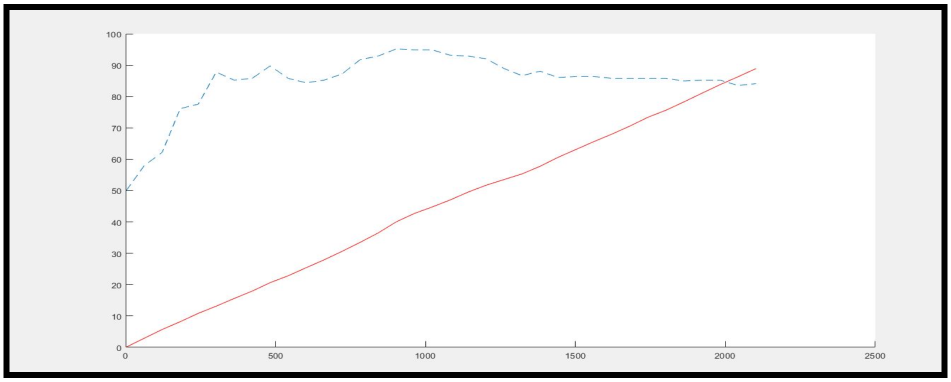
Plotting Learning Curve

It is a curve showing trend of test and training accuracies as training size is varied

Naïve Bayes

Red line indicating training accuracy

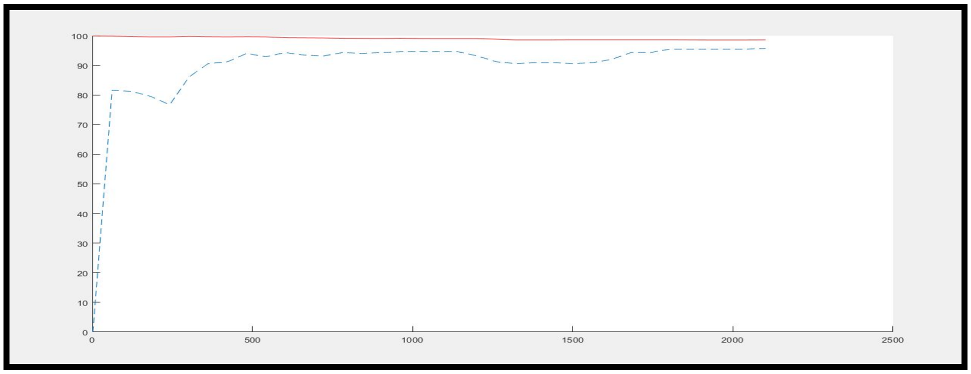
Blue(Dotted) line indicating test accuracy



KNN (K=5)

Red line indicating training accuracy

Blue(Dotted) line indicating test accuracy



4.Feature Reduction:

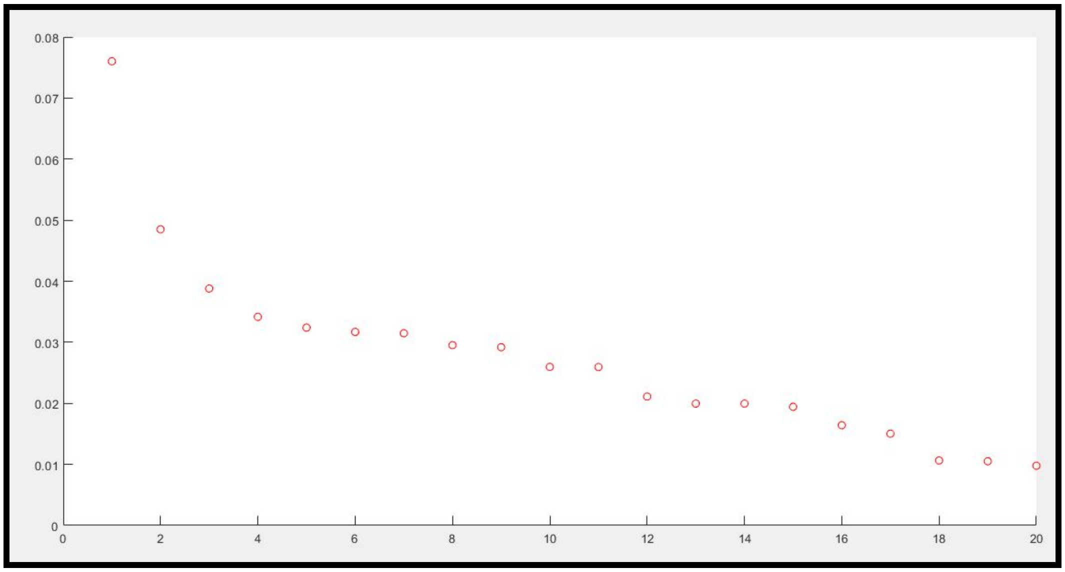
There are several reasons why we are interested in reducing dimensionality as a separate step, In most learning algorithms, the complexity depends on the number of input dimensions, ‘n’, as well as on the size of the data sample, N, and for reduced memory and computation, we are interested in reducing the dimensionality of the problem. Decreasing ‘n’ also decreases the complexity of the inference algorithm during testing.

There are two ways to go about feature reduction:

1. Selecting only High Variance Features
2. Feature Selection (Supervised)
3. Feature Extraction (Unsupervised)

In this project we aim to look at all three techniques.

4.1 Selecting features with high variances:



Taking the first 3 features [features 4,10,11] with highest variances yielded following accuracies:

Classifier	**Accuracies over 10 fold CV**
Naive Baye’s (4 bins) [Self-Written]	74.687%
KNN (K=3/K=5) [In-built]	89.289%/90.108%
Logistic Regression(Cost threshold=0.0001)	66.316%

4.2 Feature Selection:

In feature selection, we are interested in finding ‘k’ of the ‘n’ dimensions that give us the most information and we discard the other dimensions. We are going to discuss as a feature selection method.

One of the method is **Sequential Forward Selection**(SFS), we start with no features: $F = \varnothing$. At each step, for all possible , we train our model on the training set and calculate $E(F \cup \ )$ on the validation set. Then, we choose input which causes the least error.

$j = \arg\min E(F \cup \ )$

and we,

add to F if $E(F \cup \ ) < E(F)$

Accuracy of individual features:

Feature No.	Logistic Regression	Naïve Bayes	KNN(K=5)
1	52.916	64.876	57.130
2	**69.583**	**75.134**	**74.204**
3	50.587	55.483	53.882
4	**68.333**	**63.702**	**80.928**
5	50	47.329	48.087
6	**77.358**	**89.034**	**86.335**
7	49.981	52.329	56.770
8	50	51.647	53.920
9	**66.041**	**71.676**	**65.359**
10	57.5	65.066	60.691
11	49.791	62.263	61.250
12	52.916	64.876	57.130

13	**95.388**	**94.536**	**93.0114**
14	50.577	52.462	49.015
15	50	53.068	50.416
16	51.202	46.979	52.130
17	51.714	58.797	51.183
18	50.539	62.585	53.361
19	50.501	62.566	53.304
20	49.801	45.767	49.157

Table 5: Accuracy of individual features

Taking features which have an individual accuracy of greater than 60% that is feature number 13,9,6,4 and 2 and performing SFS on it.

Table 6: Performing SFS:

Feature Vectors Chosen	**Logistic Regression**	**Action Performed**
13	95.388	
13+9	93.162	Accuracy Decreases so discarding 9
13+6	95.397	Slight Increase
13+6+4	95.823	Increase
13+6+4+2	95.104	Decrease, so discarding 2

Features **13 (Meanfun), 6 (IQR)** and **4 (Q25)** can be used to get higher accuracies (**Table 6**). We can validate this by applying these features in the three algorithm applied before (Table 4)

Algorithm	**Source**	**Parameter [features (13,6,4)]**	**Accuracy(Observation)**
Naïve Bayes	Self-written	4 bins	95.710% (Significant Increase)
KNN	MATLAB	K=3/5	96.117%/96.316% (Increase)
Logistic regression	Self-written	Cost threshold=0.0001	95.823% (Increase)

Table 6: Accuracy after feature selection

Conclusion from above:

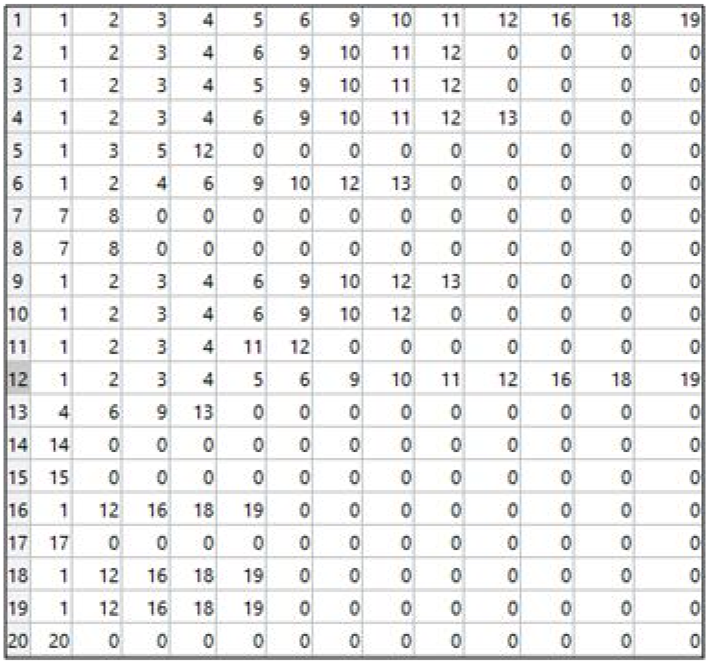
There is an increase in accuracy in all cases especially in Naïve Bayes. We can try explaining why this happens why the feature 13, 6 and 4 play such an important part in output by looking at the relationship with the output with help of Covariance and Correlation in next section.

Correlation and Covariance:

Variance tells us how a random variable varies. Covariance is how two random variable vary together. It is very much dependent on the scaling of the data. This is where correlation becomes useful — by standardizing covariance by some measure of variability in the data, it produces a quantity that has intuitive interpretations and consistent scale. Correlation value lies in the range [-1, 1]. Ideally we will require that input are highly correlated to Output -1 or 1 and input features have correlation 0 among themselves (not dependent at all).

It's clear from the above table that Features 1(Meanfreq) and 12(Centroid) are highly correlated and we can see from our dataset that they both have the same value for every sample. Implies we can decrease our computation by using only one of the two. Also feature 1 has the highest number of features with correlation coefficient greater than 0.5.

Next we found the number of features which have an absolute correlation coefficient of greater than 0.5 with each other.



1	1	2	3	4	5	6	9	10	11	12	16	18	19
2	1	2	3	4	6	9	10	11	12	0	0	0	0
3	1	2	3	4	5	9	10	11	12	0	0	0	0
4	1	2	3	4	6	9	10	11	12	13	0	0	0
5	1	3	5	12	0	0	0	0	0	0	0	0	0
6	1	2	4	6	9	10	12	13	0	0	0	0	0
7	7	8	0	0	0	0	0	0	0	0	0	0	0
8	7	8	0	0	0	0	0	0	0	0	0	0	0
9	1	2	3	4	6	9	10	12	13	0	0	0	0
10	1	2	3	4	6	9	10	12	0	0	0	0	0
11	1	2	3	4	11	12	0	0	0	0	0	0	0
12	1	2	3	4	5	6	9	10	11	12	16	18	19
13	4	6	9	13	0	0	0	0	0	0	0	0	0
14	14	0	0	0	0	0	0	0	0	0	0	0	0
15	15	0	0	0	0	0	0	0	0	0	0	0	0
16	1	12	16	18	19	0	0	0	0	0	0	0	0
17	17	0	0	0	0	0	0	0	0	0	0	0	0
18	1	12	16	18	19	0	0	0	0	0	0	0	0
19	1	12	16	18	19	0	0	0	0	0	0	0	0
20	20	0	0	0	0	0	0	0	0	0	0	0	0

From this table it is clear that features 7,8,13,14,15,17 and 20 have a correlation of greater than 0.5 with very few features (less than 4 features).

Feature	**Accuracy**				**Observation**
	NB	**KNN**		**LR**	
		K=3	**K=5**		
7,8,13,14,15,17,20 (Feature having a correlation greater than 0.5 with less than 4 features)	93.806 %	94.005 %	94.261 %	94.829%	Result with features with lowest correlation with other features

4,6,7,8,13,14,15,17,20 (Adding Features 4 and 6 as they gave good results with feature 13)	95.142%	95.483%	95.814%	95.681%	Increase in accuracy
6,7,8,13,14,15,17,20 (As 6 is correlated to fewer elements than 4)	95.833%	95.823%	96.051%	95.691%	Slight but increase in accuracy
6,8,13,14,15,17,20 (As features 7 and 8 have a correlation coefficient of 0.977 we will use only one of these features)	95.965%	95.842%	96.155%	95.587% (with 7)	
6,8/7,13,14,15,17,20 (As the remaining features are mostly independent we will remove them one at a time and record the results)	96.032%	95.700%	96.155%	95.456%	Discarding 8 (7 for LR)
	96.032%	95.492%	95.937%	95.483%	Discarding 14
	95.937%	95.833%	95.965%	95.890%	Discarding 15
	95.965	95.984%	96.2405%	95.568%	Discarding 17
	96.2405%	95.700%	96.155%	95.549%	Discarding 20

Using features that cause increase in accuracy (Algorithm dependent)	96.313 %	96.108 %	96.411 %	95.402%	With Features [6,13,15] (Best combination for NB, KNN even better than [6,13,14] and [6,13,14,15]
				95.890%	[6,7,13,14,17,20] (Best combination for LR)
Now adding all features which haven't been looked at [1,5,10,11,16/18/19] since 16,18 and 19 have a high correlation with each other.	**96.335 %**	**96.590 %**	**96.647 %**	96.089%	Adding 1 : Increase
	96.325 %	96.505 %	96.553 %	95.407%	Adding 1 and 5: Decrease
	94.905 %	96.572 %	96.004 %	**96.287%**	Adding 1 and 10: Decrease (NB &KNN) Increase (LR)
	95.890 %	96.344 %	96.553 %	96.117%	Adding 1 and 11 (&10 for LR): Decrease
	95.350 %	96.212 %	96.070 %	95.937%	Adding 1 and 16
Best Combination:	[1,6,13,15]	[1,6,13,15]	[1,6,13,15]	[1,6,7,10,13,14,17,20]	
Highest Accuracy:	**98.484 %**	**98.958 %**	**99.242 %**	**99.242%**	On one of the 10 fold cross validation

Table 7: Impact of feature selection on accuracy

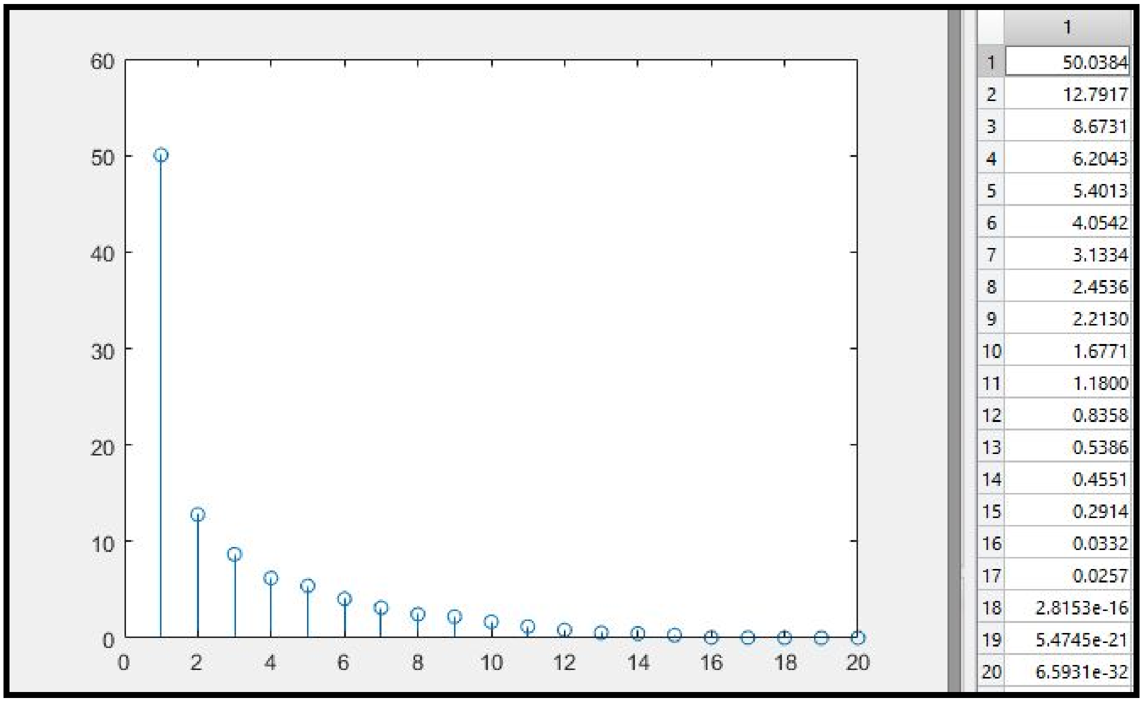
Conclusion from above: In this process we first start off by taking all the features with lowest correlation and then add/remove features in the direction of higher accuracies. We can see that Feature selection affects algorithms differently and now we have found best feature subset for our 3 of our algorithms by increasing accuracy on dataset and reduced computation by reducing number of features.

4.3 Feature extraction:

In feature extraction, we are interested in finding a new set of ‘k’ dimensions that are transformed from the original ‘n’ dimensions. One of the most widely used feature extraction method is **Principal Components Analysis (PCA).**

PCA Application:

We applied PCA on our dataset to visualize the change in accuracy of different classifiers as we increase the number of principal components included to calculate accuracy.
Below is a stem plot of average variance of Principal components over different training sets.



	1
1	50.0384
2	12.7917
3	8.6731
4	6.2043
5	5.4013
6	4.0542
7	3.1334
8	2.4536
9	2.2130
10	1.6771
11	1.1800
12	0.8358
13	0.5386
14	0.4551
15	0.2914
16	0.0332
17	0.0257
18	2.8153e-16
19	5.4745e-21
20	6.5931e-32

Variance of Principal Components and their values

Now we calculate over 10-fold cross validation average accuracy of first ‘n’ principal components by varying n from 1 to 20.

Number of PCA components used	**Accuracy of KNN [K=3]**	**Accuracy of KNN[K=5]**	**Accuracy of Logistic Regression**
1	58.172%	58.939%	64.678%
2	75.577%	76.846%	72.490%
3	77.244%	78.257%	74.649%
4	90.257%	90.814%	77.358%
5	90.823%	91.543%	88.911%
6	90.823%	91.562%	92.197%
7	91.325%	92.026%	93.276%
8	91.287%	92.064%	95.653%
9	92.187%	92.395%	95.880%
10	92.679%	93.001%	95.819%
11	92.197%	92.2064	96.089%
12	92.140%	91.931%	96.240%
13	95.520%	95.681%	96.231%
14	94.981%	95.625%	96.174%
15	95.208%	95.483%	96.117%
16	94.990%	95.113%	96.136%
17	94.725%	94.867%	96.126%
18	94.507%	94.545%	96.126%

19	94.308%	94.517%	96.098%
20	94.176%	94.346%	96.126%

5. Outliers Analysis

5.1 Using Interquartile method (Before Normalisation/Feature Scaling)

We found out the outliers in the data set using the ***interquartile method***. We considered anything outside the range from Q1-1.5IQR to Q3+1.5IQR (IQR stands for interquartile and Q1 Q3 stands for quarter). Feature number with the number of outliers present is given below.

Male Set-	**Female Set-**
Number of Outliers in feature 1 is 24	Number of Outliers in feature 1 is 16
Number of Outliers in feature 2 is 18	Number of Outliers in feature 2 is 60
Number of Outliers in feature 3 is 96	Number of Outliers in feature 3 is 188
Number of Outliers in feature 4 is 23	Number of Outliers in feature 4 is 7
Number of Outliers in feature 5 is 135	Number of Outliers in feature 5 is 197
Number of Outliers in feature 6 is 131	Number of Outliers in feature 6 is 108
Number of Outliers in feature 7 is 191	Number of Outliers in feature 7 is 151
Number of Outliers in feature 8 is 21	Number of Outliers in feature 8 is 0
Number of Outliers in feature 9 is 0	Number of Outliers in feature 9 is 0
Number of Outliers in feature 10 is 0	Number of Outliers in feature 10 is 203
Number of Outliers in feature 11 is 27	Number of Outliers in feature 11 is 58
Number of Outliers in feature 12 is 21	Number of Outliers in feature 12 is 18
Number of Outliers in feature 13 is 3	Number of Outliers in feature 13 is 33
Number of Outliers in feature 14 is 151	Number of Outliers in feature 14 is 150
Number of Outliers in feature 15 is 9	Number of Outliers in feature 15 is 2
Number of Outliers in feature 16 is 351	Number of Outliers in feature 16 is 2
Number of Outliers in feature 17 is 15	Number of Outliers in feature 17 is 13
Number of Outliers in feature 18 is 15	Number of Outliers in feature 18 is 13
Number of Outliers in feature 19 is 112	Number of Outliers in feature 19 is 110

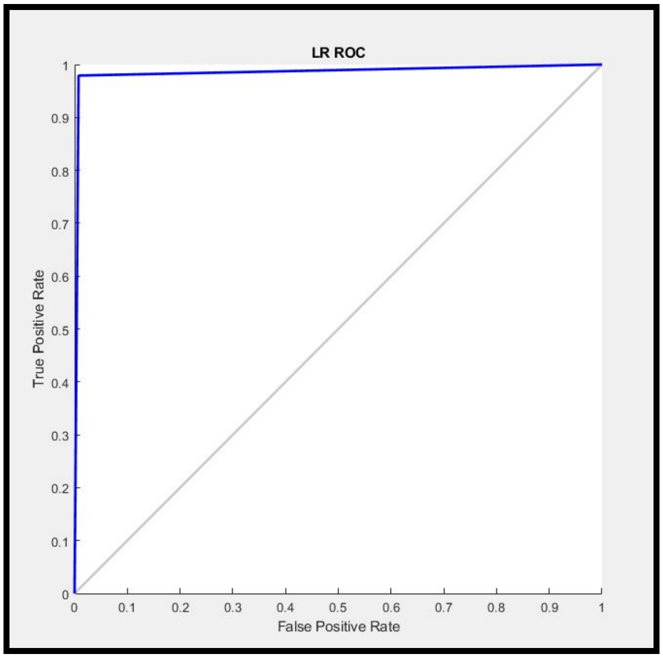
5.2. Outlier Detection using Mean and Standard Deviation

We found out mean and standard deviation values of each of the feature and considered this 21 length vector as the mean instance.Now we considered values lying within the range mean+/- 2*(standard deviation) and considered rest as outliers.
The following results were obtained:

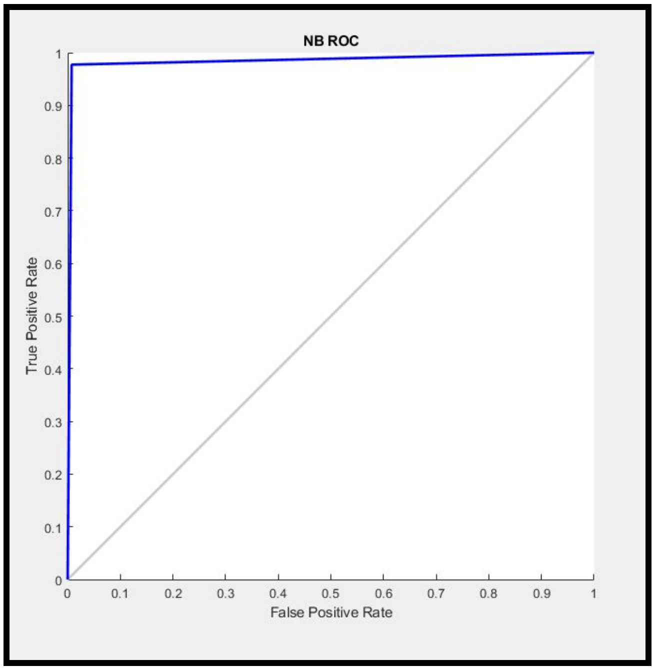
Feature	Accuracy				Observation
	NB	KNN		LR	
		K=3	K=5		
1,6,13,15-for NB and KNN 1,6,7,10,13,14,17, 20-for LR	95.86%	95.93%	95.99%	95.41%	The features were selected keeping in mind the highest accuracies for different algorithm

6. ROC Curves for different classifiers:

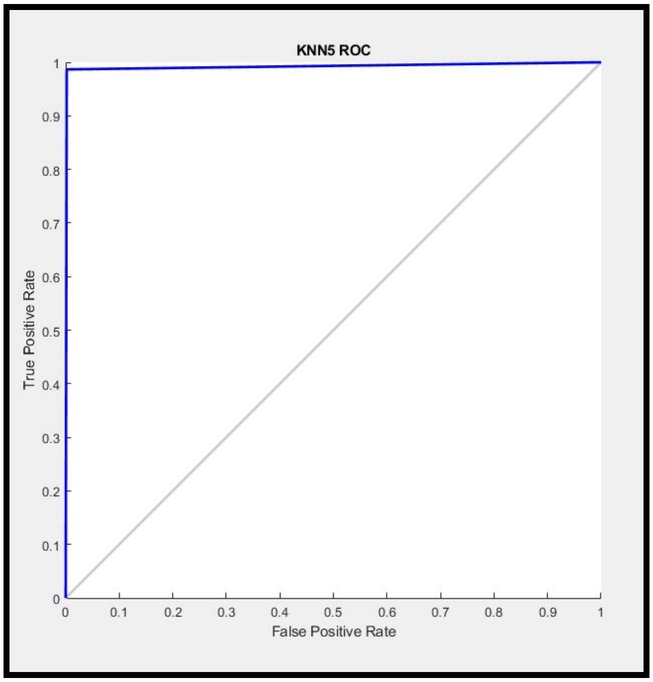
AUC: 0.9858; FPR: 0.0076 ; TPR: 0.9792



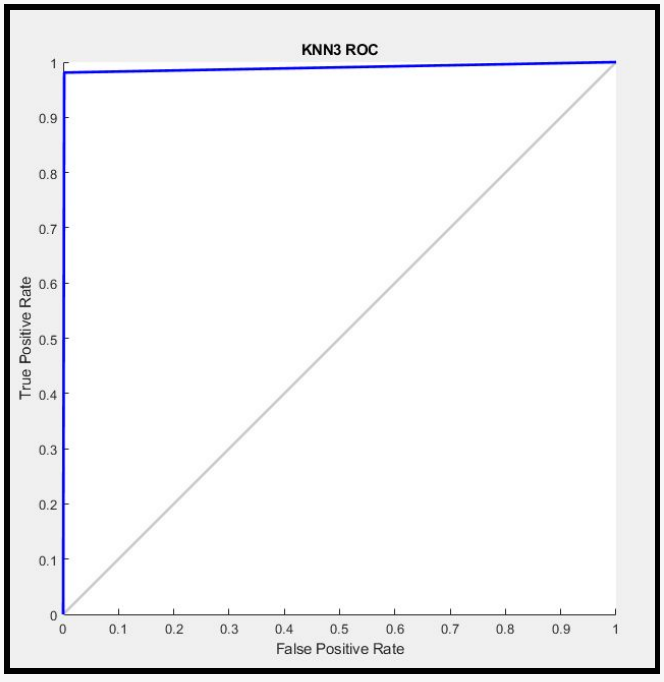
AUC: 0.9848; FPR: 0.0076 ; TPR: 0.9773



AUC: 0.9924; FPR: 0.0019 ; TPR: 0.9867



AUC: 0.9896 ; FPR: 0.0019 ; TPR: 0.9811



7. Applying SVM:

We ran our data set though standardisation ie. making zero mean and unit variance.
We divided our data into 25% training and 75% testing (random division)
On applying SVM on our original dataset we got an accuracy of **96.382%** over 10-fold cross validation while achieving a highest accuracy of **99.053%.**

Number of PCA	**Linear(%)**	**Polynomial(%)**	**RBF(%)**
1	67.3	71.3	72.47
2	82.1	79.9	83.96
3	82.4	80.5	84.47
4	83.45	80.5	88.6
5	88	89.89	93.8
6	88.76	93.3	95.07

7	94.3	95.7	96.6
8	94.3	95.45	96.7
9	97.72	97.22	97.85
10	97.85	96.6	98.1
11	97.8	96.2	97.8
12	98	96	98
13	98	96	98.1
14	98	96	98.1
15	98	96	98.1
16	98	96	98.1
17	98	96.3	98.2
18	98	96.2	98.2
19	98	95.8	98.1
20	98	95.7	98.1

We changed the **cost parameter** of SVM algorithm from 10 to 200 and found out there was no major change in the accuracy. Change of **±*0.5%*** was observed with the optimum accuracy quoted above.

Kernel Used	**Accuracy over 10-fold CV**
Default Kernel and parameters-	96.2%
Linear Kernel	96.9%
Radial Basis Function kernel	96.5%
Polynomial Function	95.2%

8. Conclusion:

In this project we looked at various classification algorithms and methods to process data in order to increase performance of our classifiers.

First we looked at Subset selection and found the subset of our original feature list which gives the highest accuracy thus in the process finding the importance of our features. Next we looked at further reducing our dimensionality but using PCA as a feature extraction method. Here we observed that choosing features with highest variation wasn't really giving us the best results as compared to subset selection. In an attempt to further refine our dataset we tried performing outlier analysis but our results didn't improve. Lastly we added SVM to our list of initial classifiers.

Classifier	**Highest Accuracy Observed**
Naive Bayes	98.484%
Logistic Regression	99.242%
KNN(K=3/K=5)	98.958%/99.242%
SVM	99.053%

Before 30th April we plan to do a better evaluation of SVM results and try and add Neural Networks and Decision Trees to our list of classifiers.

9. References:

- · http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-usingmachine-learning/
- · Introduction to Machine Learning - Ethem Alpaydin
    - a. Logistic Regression Algorithm : Pg 222
    - b. Sequential Forward Selection : Pg 111
- · https://www.kdnuggets.com/2017/02/removing-outliers-standard-deviation-python.
- https://cran.r-project.org/web/packages/warbleR/warbleR.pdf