GENDER-RECOGNITION BASED ON VOICE

CS419

Guide:
Prof. Abir De

Group:
Naman Chanduka – 190040064
Pranamya Kulkarni – 190040072
Tushar Nandy - 190020125

Contents

1. Introduction 2

2. Data Set 3

3. Analysis of Feature Space 4

3.1 Evaluation metrics 5

1. Precision 5

2. Recall 6

3. Accuracy 6

4. AUC-ROC 6

4. Algorithms and Results 7

4.1 CART model 7

4.2 Support Vector Machine 7

4.2.1 Results 8

4.3 XGBoost 9

4.3.1 Results 9

4.4 Feed Forward Neural Network 11

4.4.1 Results 12

4.5 Naive Bayes 13

4.5.1 Results 14

4.6 Logistic Regression 15

4.6.1 Results 16

4.7 Random Forest 17

4.7.1 Results 18

5. Conclusion 20

6. References 21

1.Introduction

Humans have a natural capability of identifying the difference but when it comes to computers, we need to teach it by providing inputs, methodology or different training data and make it learn. In this project, the focus is on training computers to identify the gender based on input of acoustic attributes using various Machine Learning algorithms and get the best results. The collected voice samples are obtained after preprocessing in R using the warbler package.

An important reason for voice-based gender recognition is that it can improve human-machine interaction. For example, the advertisements can be specialized based on the age and the gender of the person on the phone. It also can help identify suspects in criminal cases or at least it can minimize the number of suspects. Some other uses of this system can be applied for adaptation of waiting queue music where a different type of music can be played according to the person's age and gender. Using this age and gender recognition system, the statistics about age and gender information for a specific population can be learned.

2. Data Set

We collected our data from Kaggle. The dataset consists of 3,168 recorded voice samples, collected from 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.

The following acoustic properties of each voice are measured and included within the CSV:

- 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: interquartile 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

3. Analysis of Feature Space

A general understanding is that the frequency of male voice is lower than that of female voice. We plotted histograms for each feature, spread over the given frequency range and found that most of the plots seem to have some visible distinctions for each gender.

This fact was also confirmed by employing the 2-sided Kolmogorov-Smirnov tests on each feature. The null hypothesis is that the two samples are drawn from the same distribution. We compared the alpha values for each feature against the standard value of 0.05.

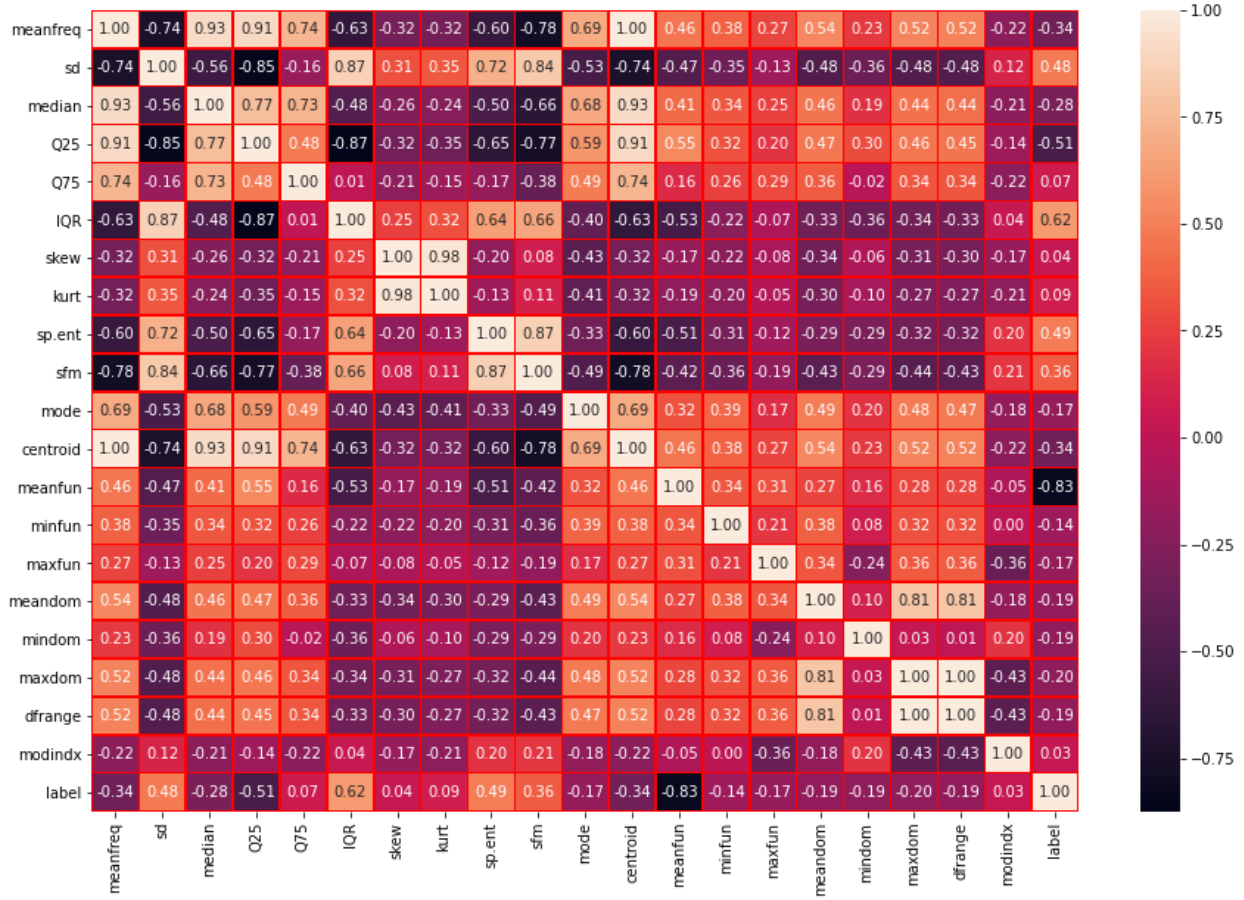
The p-values obtained (using SciPy) were:

Attribute	p-value
Mean Freq	2.62E-64
Sd	1.72E-287
Median	2.33E-49
Q25	0
Q75	1.90E-06
IQR	0
Skew	5.24E-59
Kurt	4.32E-38
Sp.Ent	1.34E-204
Sfm	3.68E-146

Attribute	p-value
Mode	1.01E-85
Centroid	2.62E-64
Meanfun	0
Minfun	2.26E-13
Maxfun	2.52E-09
Meandom	3.98E-23
Mindom	7.28E-29
Maxdom	4.11E-46
Dfrange	3.33E-43
ModIndx	0.295

The above suggests that most of the features have, indeed, not been drawn from the same distribution. This might suggest that classification algorithms should achieve reasonable accuracies.

The correlation HeatMap between the features is plotted as follows.



From the correlation heat map, we can see that the average of fundamental frequency measured across acoustic signals (given by meanfun in the dataset) is most positively correlated with the target variable i.e females tend to have a higher value for the feature meanfun than males.

From the bar plot, we can infer that if the average of the fundamental frequency is greater than 155Hz, then the probability that the target variable is female is very close to 1; and if it is less than 135Hz, then the probability that the target variable is male is very close to 1 .
Also, the modulation index does not have a solid correlation with the target variable; thus, both males and females tend to have the same modulation index.

3.1 Evaluation metrics

1. Precision

   It attempts to answer the following question:
   "What proportion of positive identifications was actually correct?"
   The formula is defined as:

   TP/(TP + FP)

2. Recall

   It attempts to answer the following question:
   "What proportion of actual positives was identified correctly?"
   The formula is defined as:

   $$TP/(TP + FN)$$

3. Accuracy

   This is simply the proportion of correctly identified labels.
   $(TP + TN)/(TP + FP + FN + TN)$

4. AUC-ROC

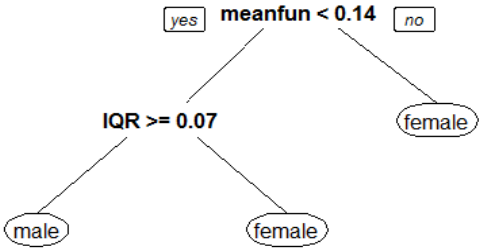
   A graph of the True Positive Rates (TPR) plotted against the False Positive Rates (FPR). It is desired that a good classifier has an area close to 1.

4. Algorithms and Results

This section presents the algorithms used for the analysis and discusses the results of each of the algorithms.

4.1 CART model

CART stands for Classification and Regression Tree. It is a series of if-else statements that aims to classify using branches. The thresholds for branching are the learnt parameters.
Previous work developed a rather simple Tree, based on 2 features only, to predict the gender.
This model, *though not used in our project*, is believed to give an accuracy of 97% on the test set.



4.2 Support Vector Machine

SVM allows transformation of feature space using the kernel trick. The choice of kernel for SVM was made during hyperparameter tuning.
From the options of 'linear', 'polynomial' and 'Radial Basis Function (or Gaussian)', RBF was the best performing kernel.

There are two important hyperparameters:

1. Regularization parameter (C)
   For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore, a simpler decision function, at the cost of training accuracy.

2. Influence parameter (gamma) for RBF
   Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters is the inverse of the radius of influence of samples selected by the model as support vectors.

4.2.1 Results

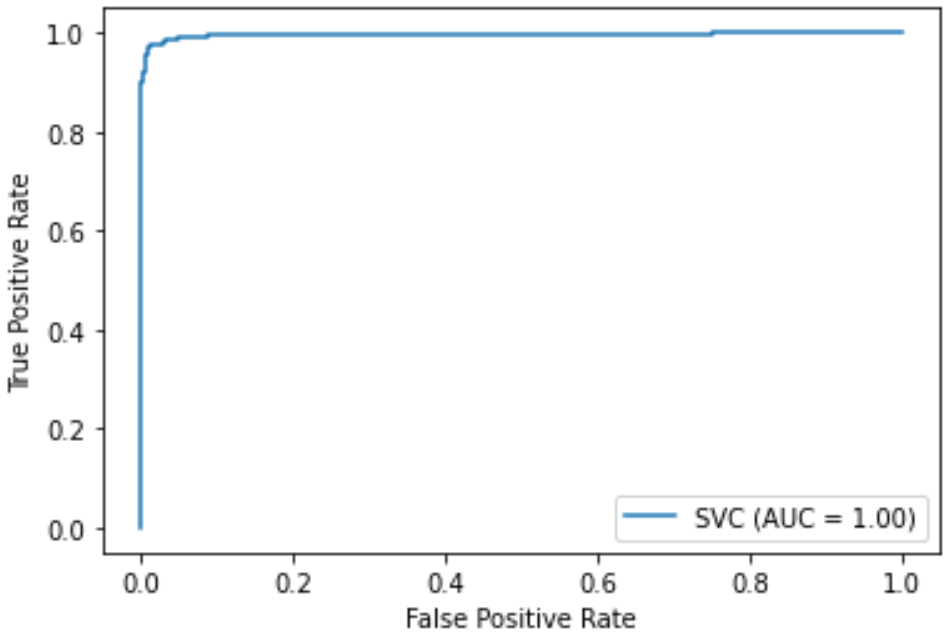
I. Best parameters:
   i. C = 1.0
   ii. Kernel: Radial Basis Function
   iii. gamma = 0.1

II. Best Accuracy:
   i. Training set: 98.58%
   ii. Test set: 97.95%

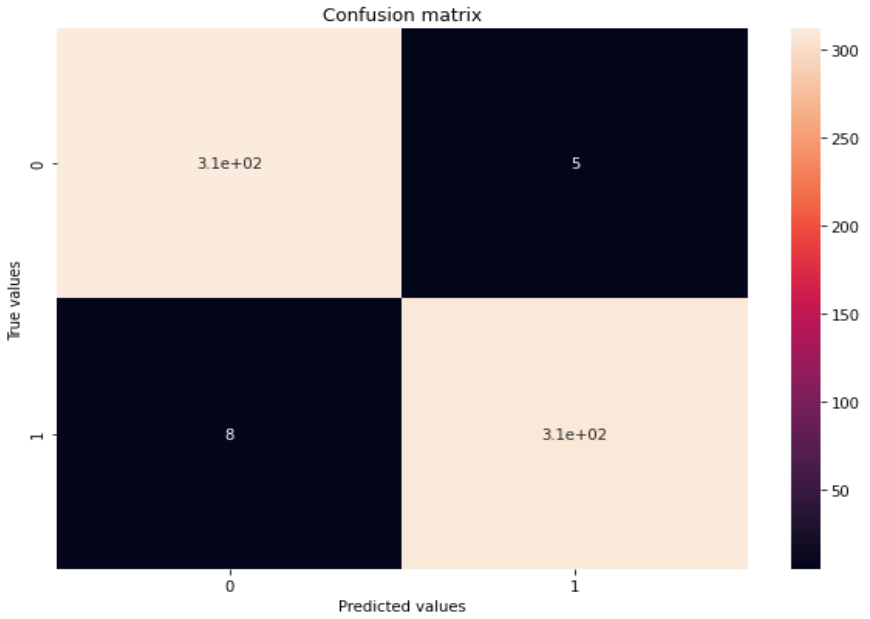
III. Precision (Reported wrt each label):
   i. Male: 98.41%
   ii. Female: 97.50%

IV. Recall
   i. Male: 97.48%
   ii. Female: 98.42%

V. AUCROC:



VI. Confusion-Matrix:



4.3 XGBoost

Given the success of this algorithm in various ML competitions, we tried this algorithm on our dataset. Gradient boosting is an ensemble method in which the successive decision trees are developed with more weightage to harder training examples so as to strengthen the weak learners.
When this Boosting is employed by utilizing the maximum amount of computing resources, it is called extreme gradient boosting.

4.3.1 Results

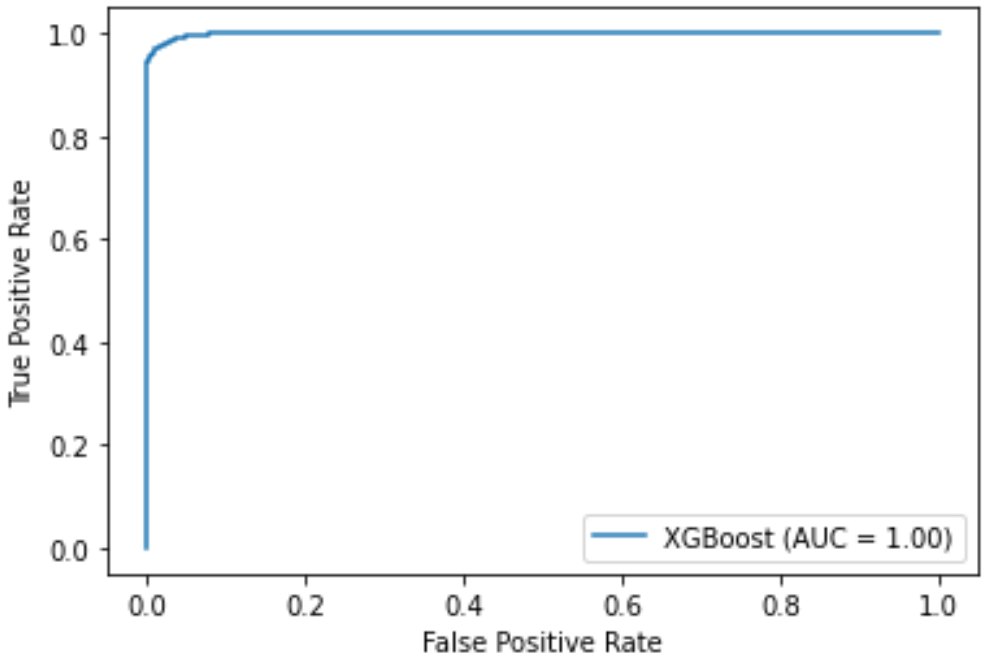
I. Best Parameters:
   i. Gamma = 0.0
      (Signifies the minimum reduction in loss required for splitting to take place)
   ii. Learning Rate = 0.1
   iii. Max Depth of tree = 4
   iv. L-2 Regularizer = 10

II. Best Accuracy:
   i. Training Set: 99.84%
   ii. Test Set: 97.95%

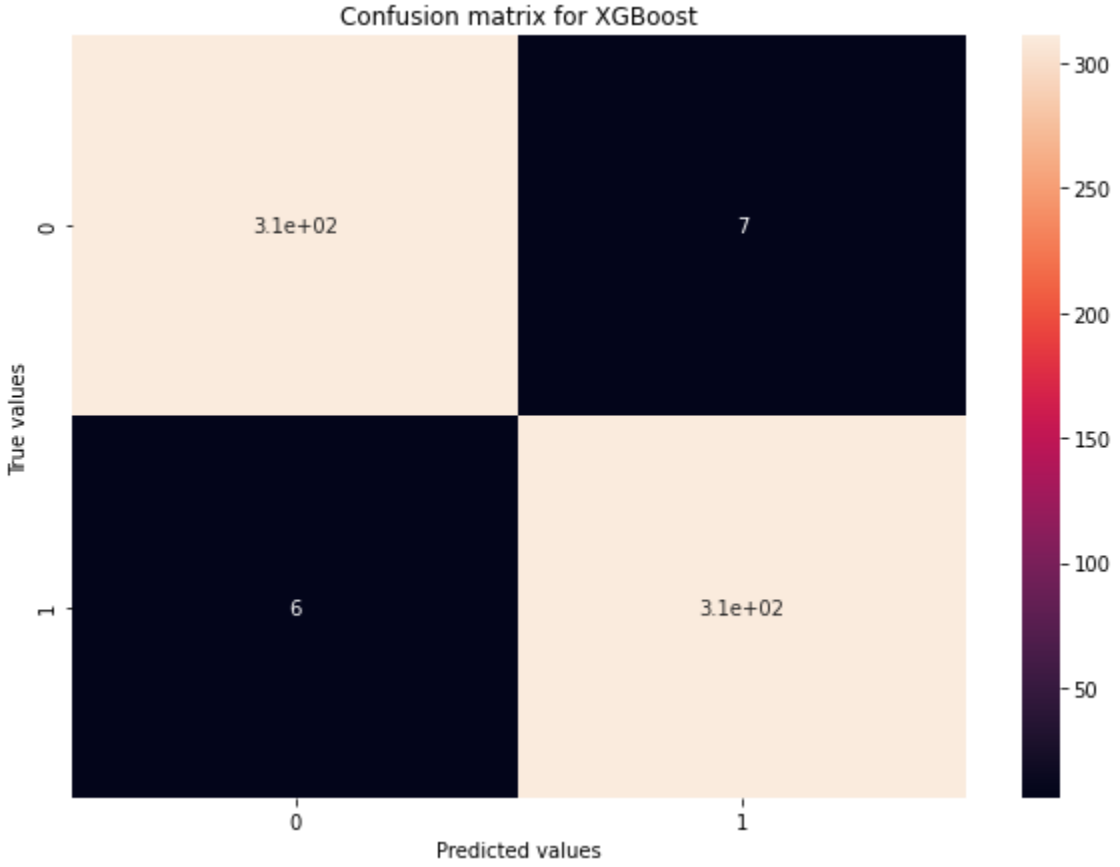
III. Precision:
   i. Male: 97.79%
   ii. Female: 98.10%

IV. Recall:
   i. Male: 98.11%
   ii. Female: 97.8%

V. AUCROC:



VI. Confusion-Matrix:



4.4 Feed Forward Neural Network

This is an implementation of a FFNN, using TensorFlow with keras, containing 1 input layer and 3 hidden layers. The activation function used in each node is ReLU.

The loss used for optimization is the Binary Cross Entropy loss, using the 'Adam' optimizer. Also, an early stopping criterion has been used with a suitable patience parameter to make sure that the optimal solution is reached. There are a lot of hyperparameters to be tuned when looking at a FFNN. For the number of hidden layers and number of nodes in each layer, generally such binary classification problems can be efficiently solved using 1 hidden layer and number of nodes roughly between the input layer and output layer. However, using trial and error, we have chosen the current combination as it gives a greater accuracy. Similarly, batch size and epochs have been decided using trial and error. Early stopping criterion is used with a patience of 25 iterations over the training loss. This prevents underfitting.

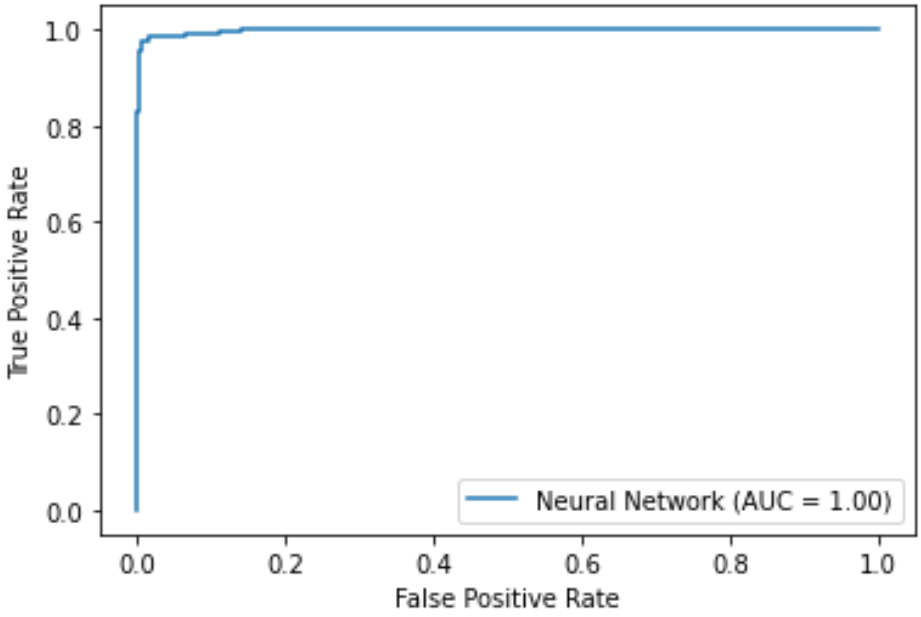
4.4.1 Results

I. Best Accuracy:
   i. Training set: 99.32%
   ii. Test set: 99.1%

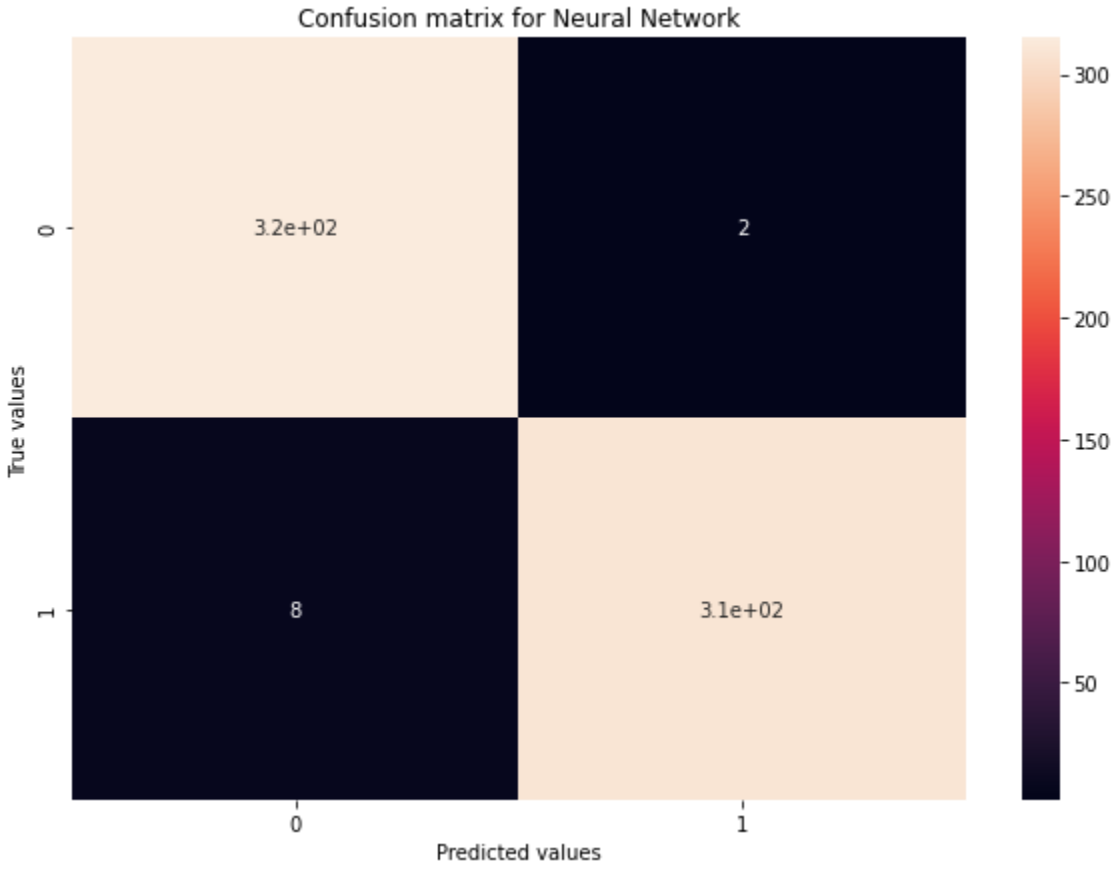
II. Precision:
   i. Male: 99.36%
   ii. Female: 97.52%

III. Recall:
   i. Male: 97.48%
   ii. Female: 99.37%

IV. AUCROC:



V. Confusion-Matrix:



This method gives the following observations:

1. The model gives roughly 98.5-99% accuracy
2. It has an AUCROC value very close to 1
3. The model misclassified only around 6-10 data points
4. It is slower to train, however gives very good accuracy

4.5 Naive Bayes

This is an implementation of the Gaussian Naïve Bayes model using the sci-kit learn library.
There are not many hyperparameters available to tune for this model, as most of the values are determined using the maximum likelihood. This method assumes conditional independence between every pair of features given (naïve assumption), and uses the Bayes theorem to find relations between features and labels. This method is extremely fast to train compared to other methods. Naïve Bayes is known to be a decent classifier, but a bad estimator.
For Gaussian Naïve Bayes, an additional assumption, that the likelihood of the features given the label is Gaussian.

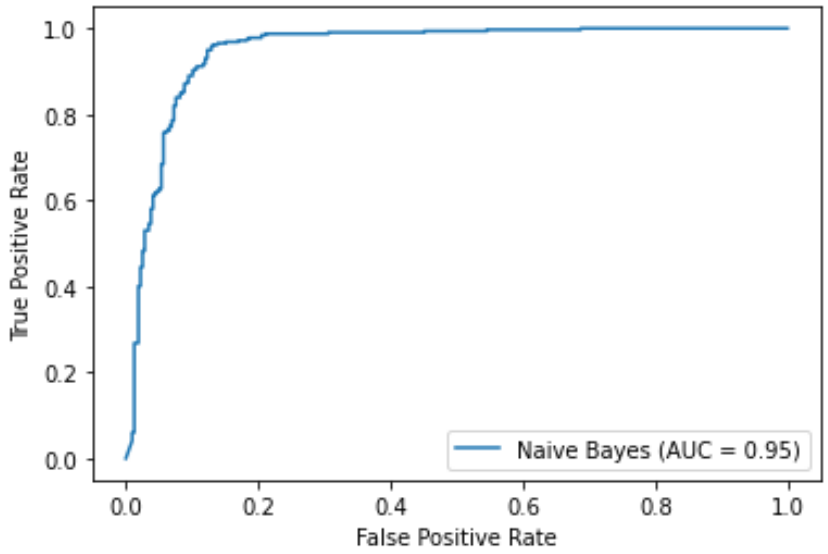
4.5.1 Results

I. Best Accuracy:
   i. Training set: 89.00%
   ii. Test set: 90.1%

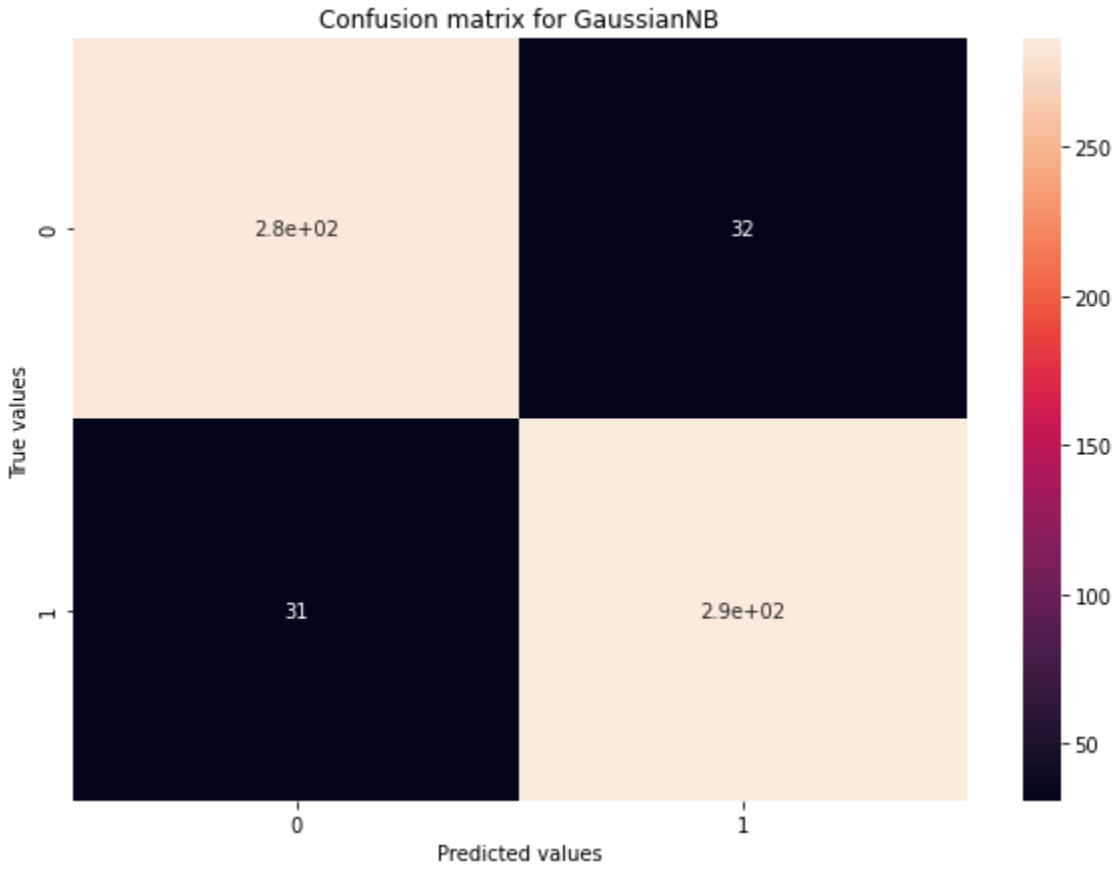
II. Precision:
   i. Male: 89.94%
   ii. Female: 90.19%

III. Recall:
   i. Male: 90.22%
   ii. Female: 89.90%

IV. AUCROC



V. Confusion Matrix



We have the following observations:

1. The model gives around 85-90% accuracy
2. Accordingly, it has a high misclassification rate, and incorrectly classifies ~70 data points
3. However, it is the fastest to train algorithm used in this project

4.6 Logistic Regression

A standard linear classifier, which assumes linearity in the data. This is implemented using the sklearn library. This is a relatively faster algorithm to train, and also achieves remarkable accuracy. The following hyperparameters were chosen for hyperparameter tuning using grid search:

- Penalty type (l1, l2) - Used to specify the norm used in the penalization
- Regularization parameter C - Inverse of regularization strength. Smaller values specified stronger regularisation.

The results are summarized in the below section.

4.6.1 Results

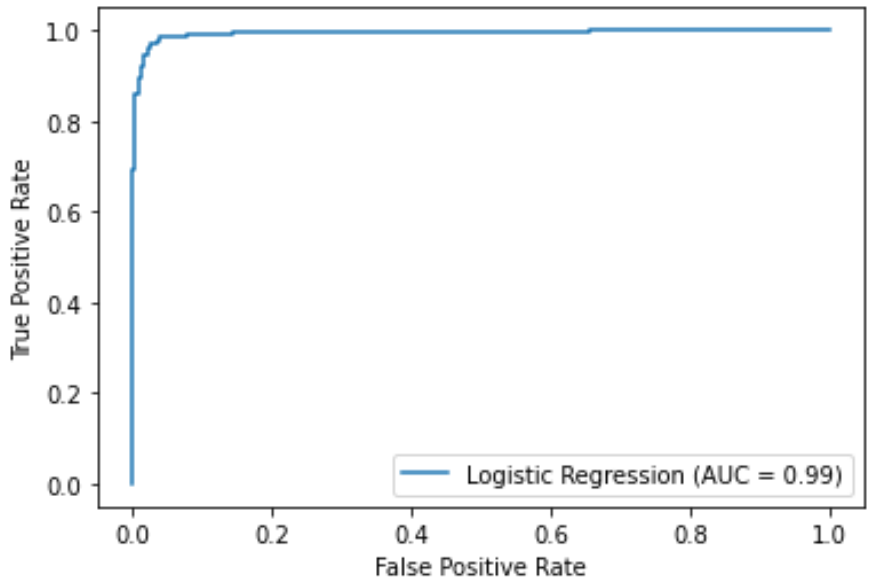
I. Best Parameters:
   i. Penalty : L2
   ii. C = 0.311

II. Best Accuracy
   i. On test set = 97

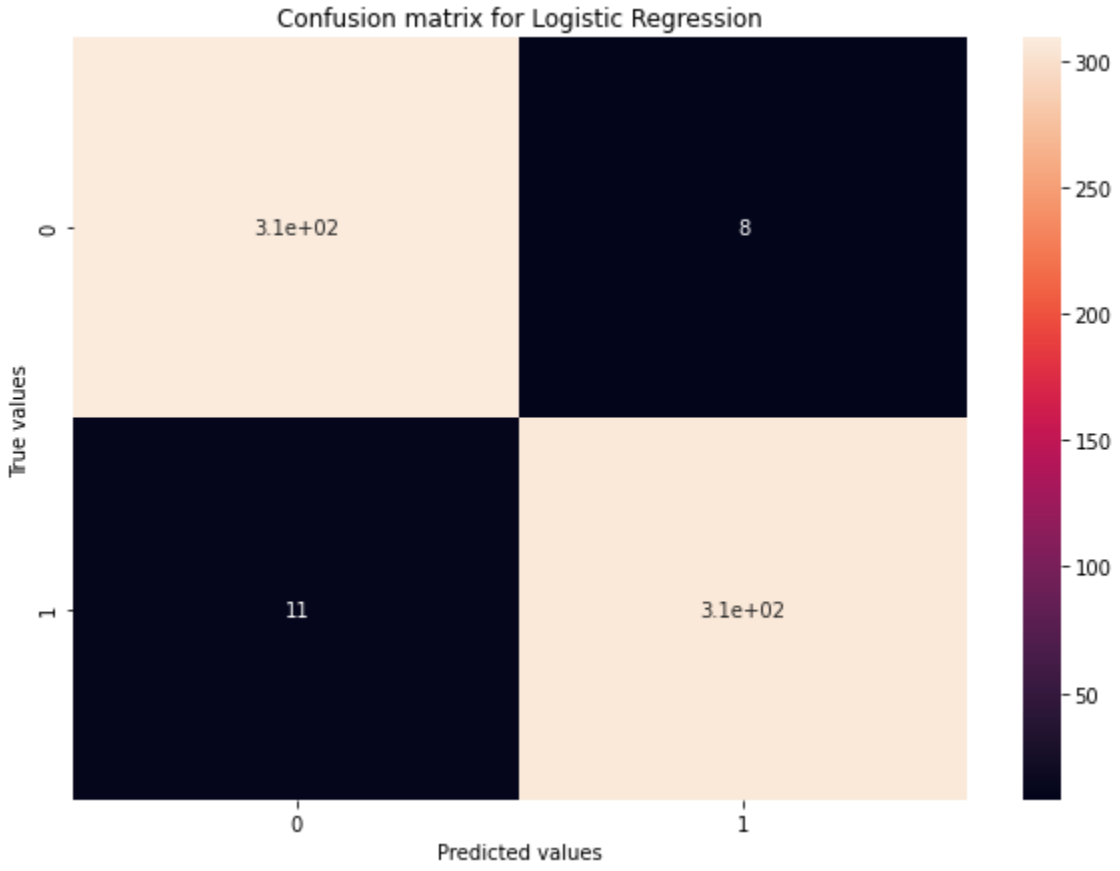
III. Precision:
   i. Male = 97.45
   ii. Female = 96.56

IV. Recall
   i. Male = 96.43
   ii. Female = 97.47

V. AUCROC



VI. Confusion Matrix



4.7 Random Forest

An ensemble model that is based on randomized decision trees. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) of the individual trees. The following hyperparameters were chosen for hyperparameter tuning using random search:

- Bootstrap (bool) - The sub-sample size is controlled with the max_samples parameter if True (default), otherwise the whole dataset is used to build each tree.
- Criterion - This function is to measure the quality of a split.
- Max depth - This is the maximum depth of the tree.
- Min_samples_leaf - This is the minimum number of samples required to be at a leaf node.
- N_estimators – It gives the number of trees in the forest.

4.7.1 Results

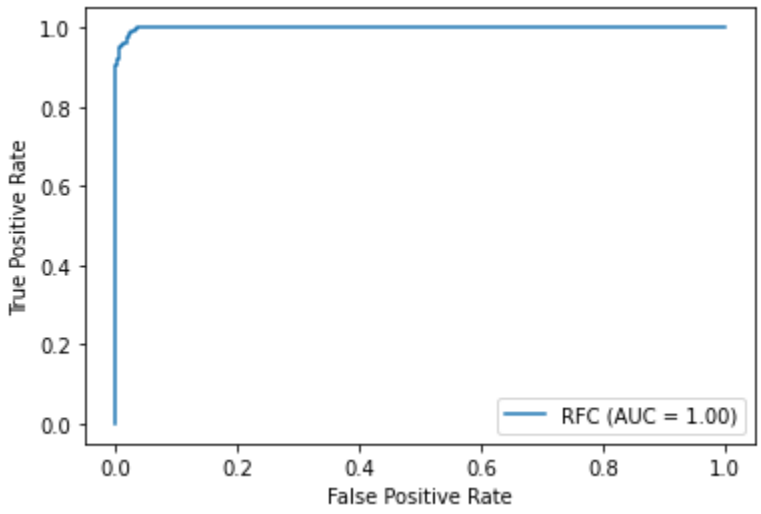
I. Best Parameters:
   i. Bootstrap = True
   ii. Criterion = Gini
   iii. Max depth = None
   iv. Min_samples_leaf = 2
   v. N_estimators = 100

II. Best Accuracy
   i. On test set = 97.79

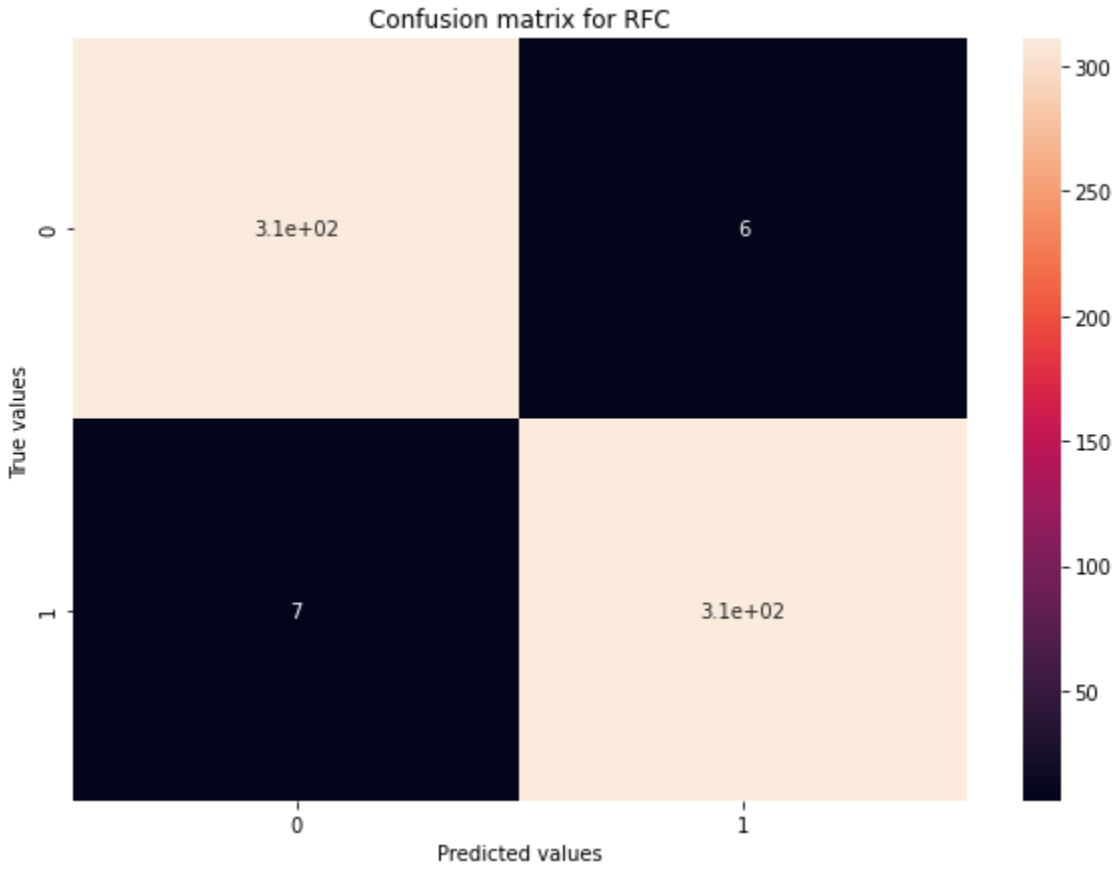
III. Precision:
   i. Male = 98.09
   ii. Female = 97.49

IV. Recall
   i. Male = 97.48
   ii. Female = 98.10

V. AUC-ROC



VI. Confusion Matrix



5. Conclusion

Algorithm	*Accuracy on Test Set*
SVM	97.95%
XGBoost	97.95%
Neural Network	99.10%
Naive Bayes	90.10%
Logistic Regression	97.00%
Random Forest Classifier	97.47%

We can see that our FFNN performs significantly better than the other models. SVM, XGBoost and Random Forest perform a little worse, however take less time to train. Naive Bayes and Logistic Regression are much faster when it comes to training the model, however Naive Bayes performs quite worse compared to all the other classifiers. Logistic Regression seems to be the perfect middle ground between time to train and accuracy, which in turn gives the most efficient model.

6. References

1. https://www.kaggle.com/primaryobjects/voicegender
2. https://drive.google.com/file/d/1Etkfb3cZBEhh8QlcIq6pgb4oq6xRSCK2/view?usp=sharing
3. https://drive.google.com/file/d/171neOuyfOq1bwQfHCZ3U7qliVp3L8Wd1/view?usp=sharing
4. http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-using-machine-learning/
5. https://scikit-learn.org/stable/index.html
6. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html#scipy.stats.ks_2samp
7. Murat H. Sazli 2006, A brief review of feed forward neural networks, Commun. Fac. Sci. Univ. Ank. Series A2-A3 V.50(1) pp 11-17
8. https://www.tensorflow.org/api_docs