CS 584 Final - Fall 2017

Jake Whalen

December 11, 2017

# Executive Summary

This final project was developed for the GMU CS 584 Spring lecture course at Engility. The goal was to choose an interesting data set and apply the data mining techniques learned in class to the data. There was an additional goal to go above an beyond what was done in class. This was accomplished in several ways during the completion of this project.

The data selected was from [Kaggle](https://www.kaggle.com/primaryobjects/voicegender) and contains over 3,000 pre-processed audio samples. Each sample contains a label for the gender of the voice in the clip. The data was stored in a CSV file and then loaded into both R and Python. R was used for early exploratory data analysis. The data was plotted and run through statistical tests to better understand the underlying distributions. Python was used for applying the data mining techniques learnined in class. The [scikit-learn](http://scikit-learn.org/stable/) package was used for al of this work. It was used previously for all three homeworks so the application of the package was fast and easy for the analyst. The results of the models in python were exported into Excel files and then imported into R for reporting. THe final presentation and report were written in R Markdown for instant reproducibility. Making a slight change to the model parameters would automatically update the results displayed in the presentation and report.

The final results were encouraging. The worst model acheived over 80% classification accuracy while the best reached 98%. All models did better classifying Male voices. The better models acheived higher female classification accuracy. In the end it was a difficult choice between 3 different models. The best model did not acheive the highest rank in all metrics used for evaluation but it was conistently in the top 3 models for every metric.

# Motivation

Using machine learning techniques to understand human audio is a trending topic. Look no further then the in home personal assistants being sold by companies like Apple (Siri), Google (Google Home), and Amazon (Alexa). Answering a question posed by a human however can be quite difficult and would require massive amounts of data and a powerful computer to process and train models. This project reduces the scope to a more managelable level.

This project attempts to develop methods that can classify the gender of a speaker. While not as impressive as responding to a question with an answer this project will introduce how different features of a voice can help group the voice into one or more classes. These features could be used for more then gender classification. This project could potentially be extended to classify the speakers mood or age group. Overall this project was motivated by a curiosity to see how well a model could interpret a speakers voice.

# Data

The data was downloaded from [Kaggle](https://www.kaggle.com/primaryobjects/voicegender) and stored in a CSV file. The data consisted of more then 3,000 gender labelled records. Each audio clip was pre-processed in R and the transformed into 20 features by the original poster. The features included measures such as mean frequency, skewness, spectral entropy and the average of the fundamental frequency measured across the acoustic signal. The data values all consisted of floating point values. The features had ranges from 0 up to 1,300 and thus might require som transformations when passed into a model. There were no missing values which allowed th analyst to concentrate on the models and there results rather then preparing the data.

# Approach

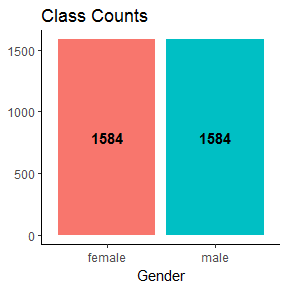
Three main steps were used to take the raw data and produce models that would accuractely classify a voice by the gender. First the method known as exploratory data analysis (EDA) was used to investigate the underlying data distributions. Second the data was passed to various algorithms and trained models to then make predictions on a test set of data. Third, and finally, the model results were compared using a variety of metrics and a best overall model was chosen.

# EDA

The first step in building a gender classifier was to inspect the data. This was acheived mostly through data visualization with a few statistical tests. The main idea of a visualization is to gain insight into the data. Rather then view the raw numbers in a table, a visualization can quickly tell a story.

#### Classes

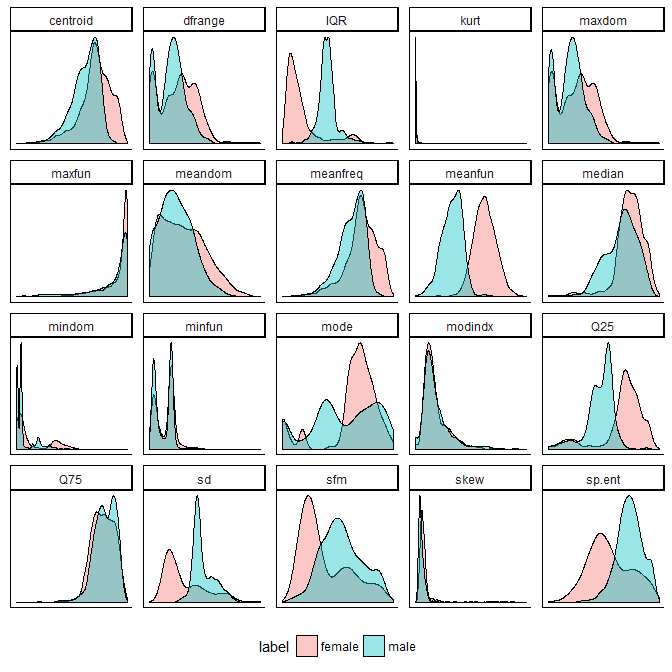
The first attribute in the data that was inspected was the label. This feature is what every model built in step 2 will try to determine and thus may well be the most important feature in the data. In this project the label consists of two values, male and female. The graph below shows the count of each label.



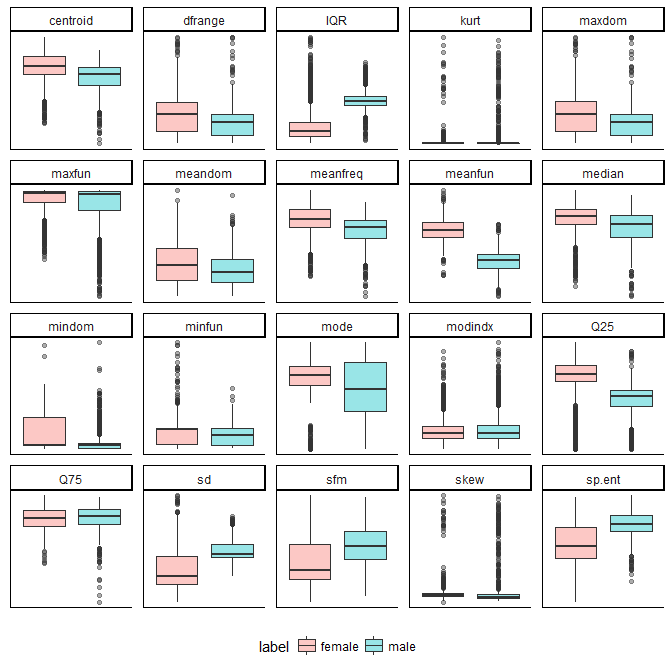
What is important to note is that the classes are evenly distributed. There are an equal number of male anf female voice recordings in the dataset. Thus a 'dumb classifier' that predicts male every time would be correct 50% of the time. In step 2 this will be used as the threshold to measure wether a model is of any use or not. A model that acheives less then 50% gender classification accuracy should be discarded.

#### Visualize

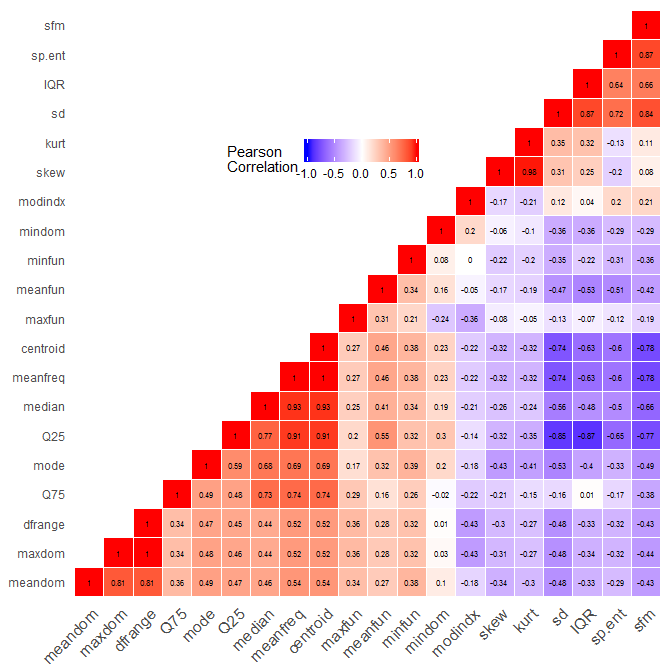
###### Distributions



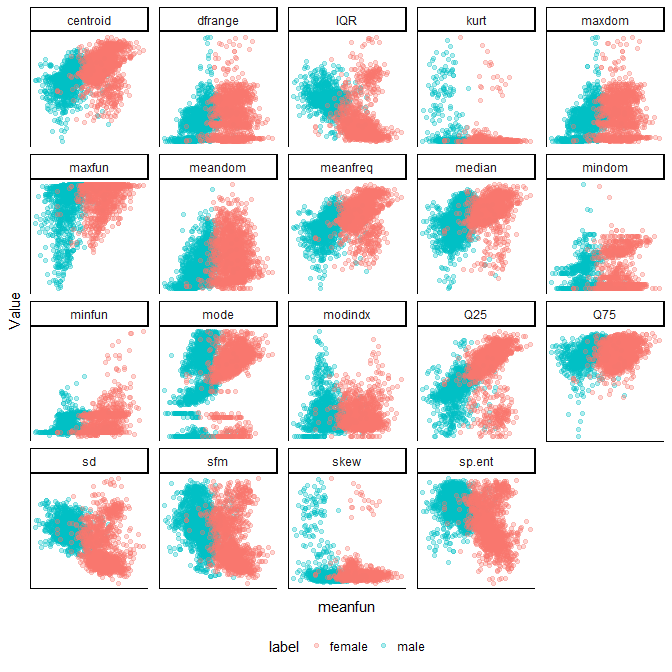
###### Boxplot



###### Heatmap



###### Scatter Plot



Also discuss 3D

#### Statistical Testing

## Feature T.Value df P.Value CI.Lower CI.Upper Female Male  
## 1 meanfreq 20.168 3112.4 0.0000000 0.018 0.023 0.191 0.171  
## 2 sd -30.748 2365.9 0.0000000 -0.017 -0.015 0.049 0.065  
## 3 median 16.661 3096.4 0.0000000 0.017 0.024 0.196 0.175  
## 4 Q25 33.490 2700.8 0.0000000 0.046 0.054 0.165 0.116  
## 5 Q75 -3.773 3161.1 0.0001642 -0.005 -0.001 0.223 0.226  
## 6 IQR -44.337 2264.3 0.0000000 -0.056 -0.050 0.058 0.111  
## 7 skew -2.062 2597.2 0.0392808 -0.699 0.078 2.985 3.295  
## 8 kurt -4.925 2588.8 0.0000009 -35.840 -11.213 24.805 48.332  
## 9 sp.ent -31.675 2622.8 0.0000000 -0.048 -0.041 0.873 0.917  
## 10 sfm -21.539 3070.4 0.0000000 -0.142 -0.112 0.345 0.472  
## 11 mode 9.811 3019.7 0.0000000 0.020 0.033 0.179 0.152  
## 12 centroid 20.168 3112.4 0.0000000 0.018 0.023 0.191 0.171  
## 13 meanfun 85.022 3149.8 0.0000000 0.052 0.056 0.170 0.116  
## 14 minfun 7.764 2878.6 0.0000000 0.004 0.007 0.039 0.034  
## 15 maxfun 9.499 2586.7 0.0000000 0.007 0.013 0.264 0.254  
## 16 meandom 10.953 2977.2 0.0000000 0.153 0.248 0.930 0.729  
## 17 mindom 11.185 2780.7 0.0000000 0.019 0.030 0.065 0.040  
## 18 maxdom 11.226 2986.3 0.0000000 1.061 1.694 5.736 4.358  
## 19 dfrange 11.021 2985.7 0.0000000 1.037 1.669 5.671 4.318  
## 20 modindx -1.734 3058.5 0.0830348 -0.018 0.004 0.170 0.177

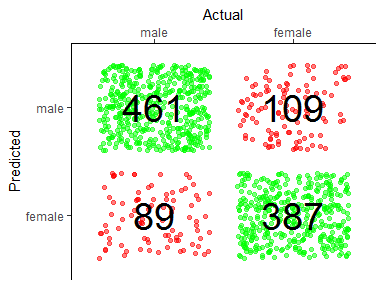
# Classification

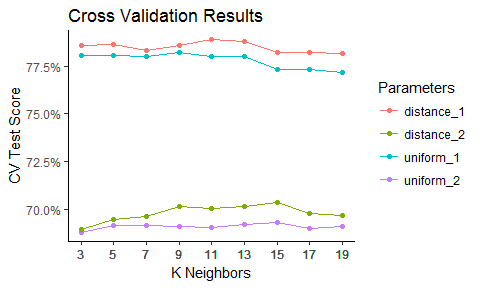
#### Machine Learning

###### Python

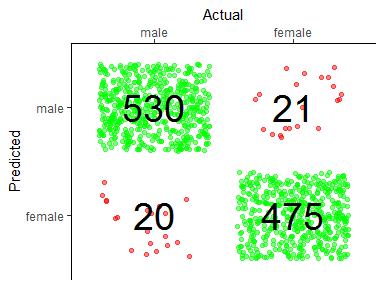
###### Scikit Learn

#### K-Nearest Neighbors

Confusion Matrix 

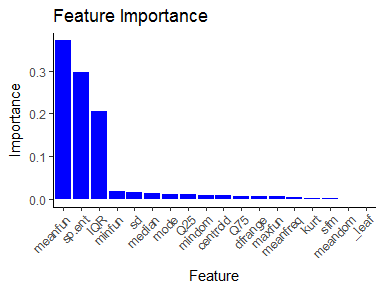
Cross Validation Results 

#### Decision Tree

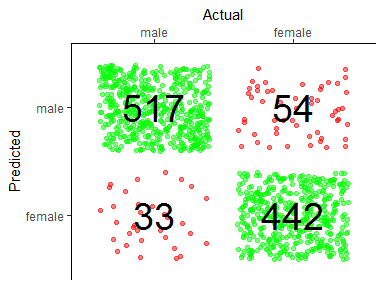
Confusion Matrix 

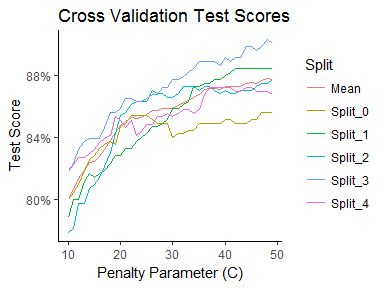
Feature Importance

## Warning: Removed 1 rows containing missing values (position\_stack).

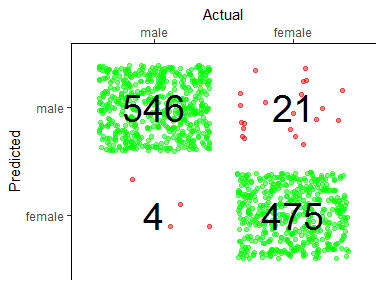


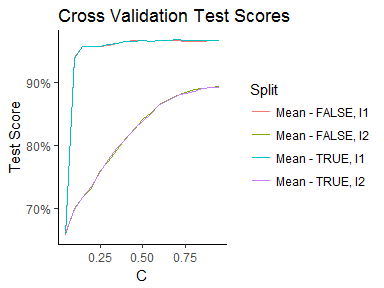
#### Support Vector Machine

Confusion Matrix 

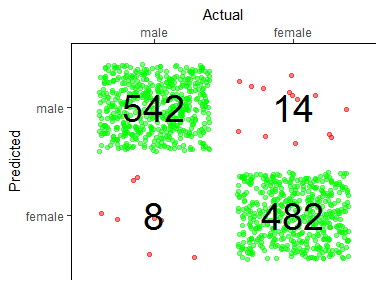


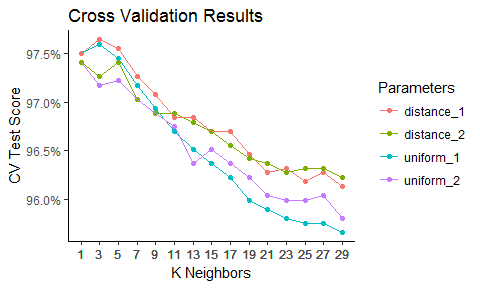
#### Logistic Regression

Confusion Matrix 

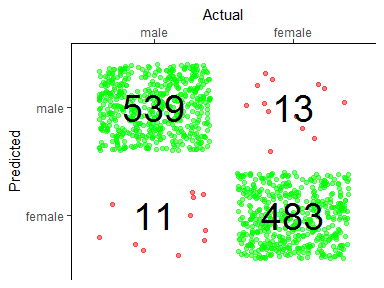


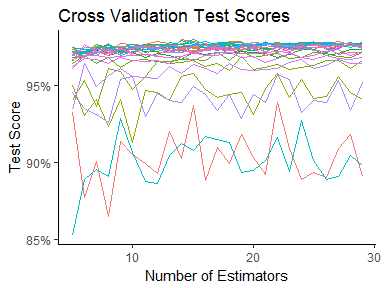
#### K-Nearest Neighbors (PCA)

Confusion Matrix 

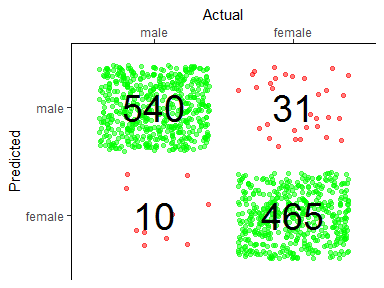
Cross Validation Results 

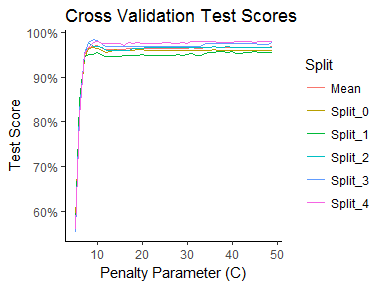
#### Random Forest

Confusion Matrix 

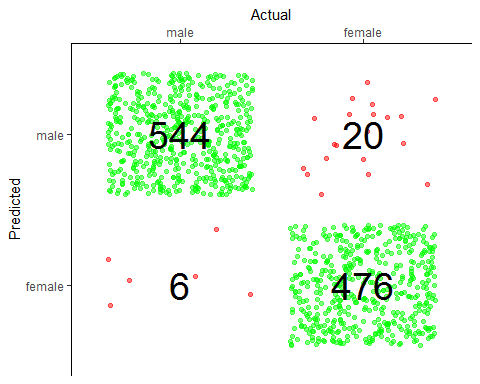


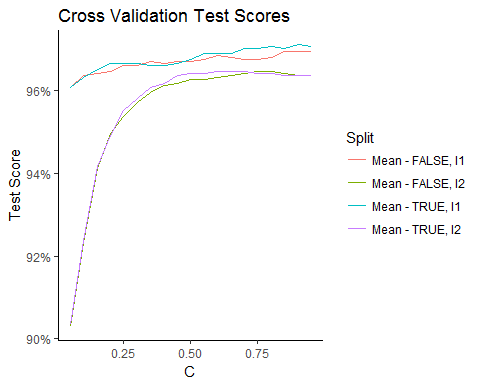
#### Support Vector Machine (PCA)

Confusion Matrix 



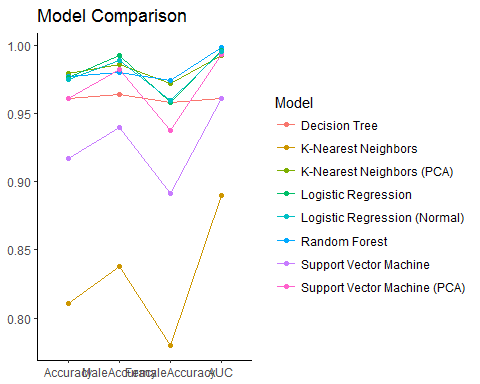
#### Logistic Regression (Normalized)

Confusion Matrix 

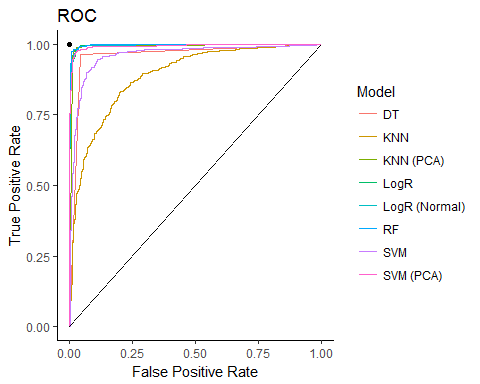


# Conclusions

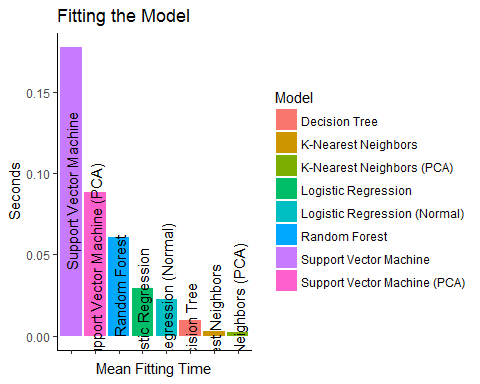
#### Criteria



#### ROC



#### Fitting Times



#### Scoring Times

