Data Science Take Home Assignment

Niklas Hansson

August 26, 2018

Abstract

This is my solution to the Data Science Take Home Assignment. The state problem is to classify the gender based upon sampled of speech. The assignment involves the following steps: webscraping, feature extraction, visualization, modeling and presentation. This report presents my findings and argues for my solution and also presents future improvements. The best results in terms of accuracy was achieved using random forest, 91.4% on a balanced test set. To increase performance further work should be put on preprocessing, sampling and utilizing the text features.

1 Problem

Predict a given person's gender using vocal features.

2 Approach and solution architecture

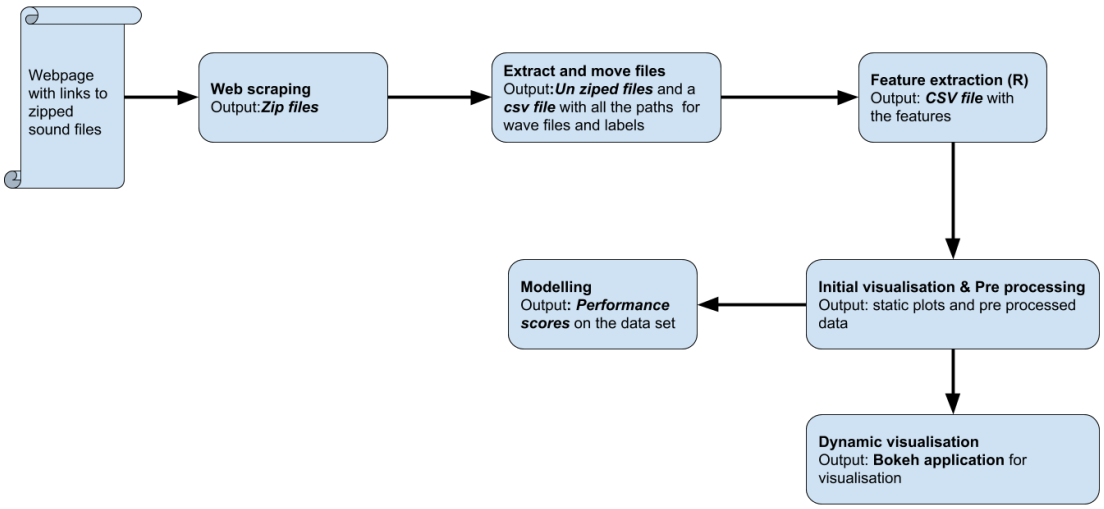

Figure 1: Figure illustration the approach to the problem.

The following main steps was identified in the problem and illustrated as seperate boxes in figure 1:

- Web scraping - automate the collection of the files.
- Prepare the files(unzip) and extract the text information before the feature extraction from wav files.
- Feature extraction from the wav files
- Visualize and understand the extracted information.

- Build models to predict the gender
- Build interactive visualization of the data

2.1 Web scraping

The structure of the website was investigated using the console in the web browser. Then a simple python script was designed in order to automate the download. The files were unpacked using python and the text features about the files were extracted. The following features were extracted from the text files:

- *Gender*
- *Age group* (called *age_range* in the code)
- *Dialect* of the speaker
- *Language*

The output from this step, where the data was downloaded and extracted, was a csv file with the following structure, see Figure 1.

path	gender	language	age_range	dialect
/home/niklas_sven_hansson/test /extracted_data/...	Male	None	Adult	British English
/home/niklas_sven_hansson/test /extracted_data/...	Female	EN	Youth	British English
/home/niklas_sven_hansson/test /extracted_data/...	Male	EN	Adult	American English

Table 1: Illustration of how the output from the web scraping

2.2 Feature extraction

In order to extract information from the raw data, several options and libraries were considered. They all had in common that the goal was to extract information about the frequencies in the voice files. Women naturally have lighter voice than men. Thus the feature extraction process was focused on extracting features related to the frequency in the voice files. Initially, python libraries where considered but in the end the best option was available in R, I ended up using *seewave* and *tuneR*. The files were processed sequentially, resulting in very low memory usage since only one file at the time were processed. However the downside was that it took quite some time to run. This could be improved if the feature process could be done for several files at once, utilizing multiprocessing. Currently, I only used one of 8 cores and thus could have reduced the computation time with $\approx x8$. This could have been done either through separating the files and running in parallel. Initially, the processing time was not known, which resulted in no parallelization. The sampling could also have been done during this phase, before the preprocessing. The following features were extracted from the wav files:

- *Mean frequency*
- *Standard deviation of the mean*
- *Median frequency*
- *Mode frequency* - The dominant frequency
- *First quartile*
- *Third quartile*
- *Interquartile range*

- *Centroid*
- *Skewness* - Measure of asymmetry
- *Kurtosis* - Measure of peakedness
- *Spectral flatness measure*
- *Spectral entropy*
- *Mean fundamental frequency*
- *Minimum fundamental frequency*
- *Max fundamental frequency*
- *Mean dominant frequency*
- *Min dominant frequency (Min dom)*
- *Max dominant frequency (Max dom)*
- *Drange* - Which was calculated as *Max dominant frequency* - *Min dominant frequency* (often the same as *Max* since *Min dominant frequency* is almost always zero in the extracted features)
- *Duration* - The length in seconds

These features were extracted after looking at the recommended features and also comparing to other open source datasets available and which features they had used while achieving very promising results, see link https://www.kaggle.com/primaryobjects/voicegender. For more information about the feature extraction, see the feature extraction script in R.

2.3 Initial visualization and preprocessing

From the heat map over the correlation between the features, see Figure 2, we can see that some of the features are very correlated with each other. One example is *Max dom* and *Drange*. The reason for the strong correlation between these two features is because *Drange* is calculated as *Max dom* minus *Min dom* and *Min dom* is almost always zero for all the observations. This could be a problem in the feature extraction for *Min dom*. The strong correlation between multiple features indicate that several features could be removed since they seem to carry the same information. Removing correlated features can improve the performance of a model both in terms of score and computational burden. No initial features where removed due to the correlation, instead recursive feature elimination where later used in order to evaluate the best number of features (and which features to use).

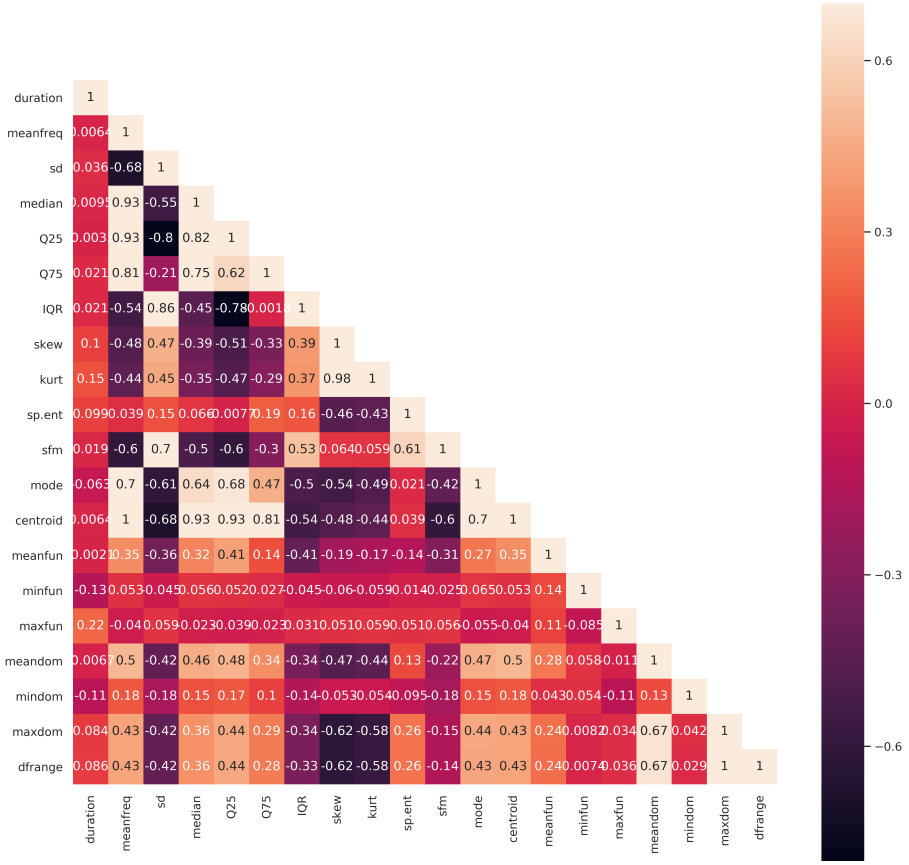

Figure 2: Heat map over the correlation for the different features.

An especially interesting distribution of the features can be seen in the Figure 3. After the initial visual observation the *Mean fundamental frequency* seemed to be the best feature for classifying the gender.

When investing the text features it was found that they were not distributed even among the genders. Some of the dialects only had observations from one of the genders and could thus lead to overfitting the model. This could be a problem if the model learned that a specific dialect always was linked to one gender. Therefore the feature *Dialects* was dropped. Information could have been gained by keeping some of the *Dialect* features were both genders were represented. Similar problem were found with the feature *Age group* and the same approach was taken. Age group could have been a very good feature since the voice change over age for humans. However, this might be a feature that is hard to collect in production.

I believe that the best way to improve the result in this problem is to further improve the feature extraction. Both by spending more time on extracting more features but also to further validate the feature extraction process.

The feature *Duration* was also dropped since it was assumed that it was not a feature that would generalize well. Depending on the application it could be that the sentences/audio files are of different length depending on interest.

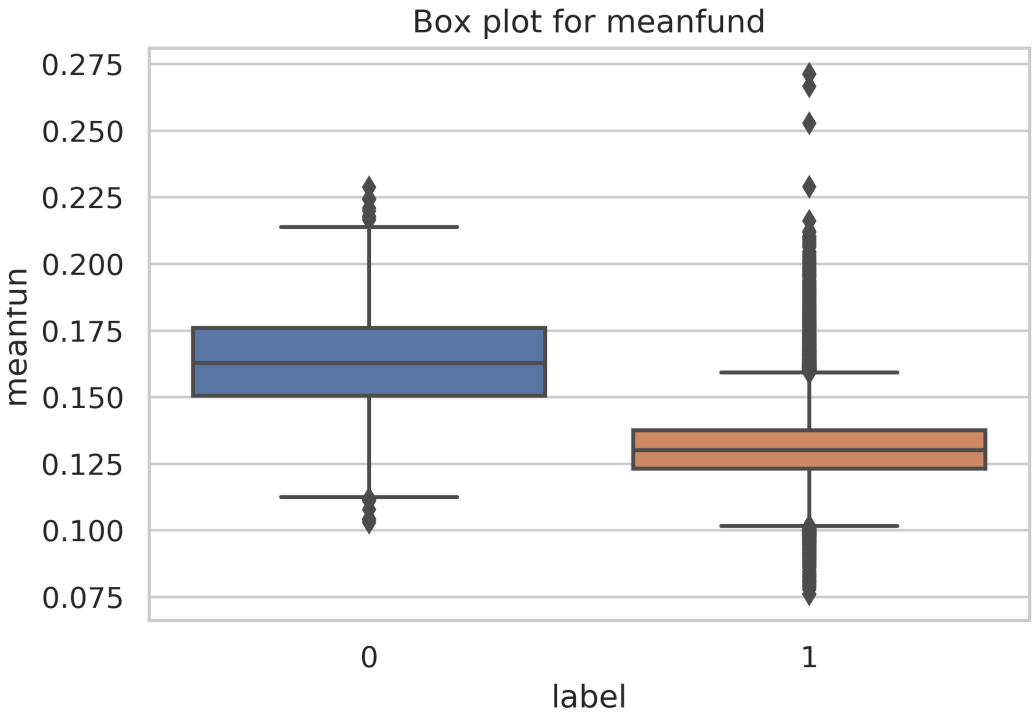

Figure 3: Box plot over the mean fundamental frequency and label. There is a separation between male and female when looking at the fundamental frequency.

Gender	Nbr of observations
Male	54677
Female	3958

Table 2: The number of observations for the different genders.

2.4 Modeling

The dataset was heavily imbalanced, see Table 2.

To handle this problem several options are possible:

- Upsampling
- Downsampling
- Using classweights for the classifier
- Adjusting the cost function

For this project there is a limited amount of time, therefore I decided to downsample the male observations. I will do this in an aggressive way by making the dataset balanced. Downsampling have the advantages of reducing the dataset (which have a cost) and thus allowing for faster iterations. The drawback is that I might loose important information. In general adding more data for a model to train on, increase the performance. Looking back this was maybe done to quickly and I could have elaborated more with different sampling rates.

The sampling resulted in a dataset with 3958 observations of each gender. The sampling process was done by selecting all female observations and then selecting male observations by random. This is also a process that could be improved together with the number of sampled examples.

The data was split in 20% for test and 80% for training. During model tuning and feature elimination, 3 fold cross-validation was used.

The following models where evaluated:

- KNN

- Logistic Regression
- Random Forest
- Gradient Boosting

These models have been selected since they are some of the most commonly used and usually perform well on classification problems were structured features have been extracted from the data. They also represent models that are easy to tune and are stable to use. Compare to neural networks, that require more tuning and are more prone to overfitting. Thus making the selected models more suitable for a quick proof of concept.

The goal is to classify the gender based upon the voice. To be fair, I will aim for being equally good on both female and male voices in terms of Recall and Precision. The reason for this is that I don't want to have a classifier that have a bias towards male or female.

The feature importance was evaluated using recursive feature elimination. This method of feature elimination could not be used for KNN in the Scikit-learn implementation. The results are presented in Table 3.

	accuracy	auc	f1_score	recall	precison
dummy	0.506313	0.490764	0.518289	0.506799	0.520566
Logistic Regression	0.878788	0.931256	0.882209	0.888752	0.875761
KNN	0.912247	0.955587	0.912853	0.899876	0.926209
Random Forest	0.914141	0.966858	0.915000	0.904821	0.925411
XGB	0.913510	0.970925	0.914428	0.904821	0.924242

Table 3: Table showing the results for the different models.

Using the feature importance and weights for logistic regression as a proxy for feature importance the *Mean fundamental frequency* where the most important feature for all of the models. For KNN the feature importance where not investigated(this is left for future work). As metric, accuracy is selected since the data is balanced. However, for later improvements of the sampling, and utilize more of the data, Recall and Precision would be better metrics. The best results were achieved with a Random Forest model with an accuracy of 91.4%. It could be noted that using only the feature *Mean fundamental frequency* for logistic regression gives an accuracy of 87.2%. The difference between KNN, Random Forest and Gradient boosting is small. To get the "true" feature importance, further work is needed since some studies have shown that the feature importance when using Random Forest might give too much importance to continuous features or high cardinality features, see http://explained.ai/rf-importance/index.html. Instead, permutation importance could be used with the cost of extra computation. Permutation importance is model agnostic.

2.5 Interactive visualization of the data

In order to run the interactive plots, the **bokeh** libary needs to be installed. **bokeh** is part of the anaconda distribution (Anaconda 3.6.5). For installation instructions see https://bokeh.pydata.org/en/latest/docs/installation.html

The interactive plots can be accessed through running the following command in the terminal from within the *Sandvik_ DataScience_ TakeHome* folder.

```
$bokeh serve --show bokeh_app
```

The web browser should then open automatically on http://localhost:5006/bokeh_app. In the repository there is also a short movie demonstrating how to start and use the interactive plots.

3 Conclusion

Relatively simple models where chosen for this investigation. This is usually to prefer for initial investigations in order to set up a baseline. Later, if the results are not good enough, investigating more complex solutions could be done. More advanced/complex models could be investigated

for this problem in order to push the result and/or decrease the amount of feature engineering. However, simpler models have advantages in terms of the possibility to explain the model and shorter time to put them in production. Depending on the problem, starting with more complex models could be reasonable, e.g working with images, to reduce the feature engineering. To further increase performance more attention should be put on the feature extraction and further attempts to use the text features. After the initial investigation I believe that this would be the most suitable step to continue on this work. Also sampling more of the data to utilize the potential of as much data as possible. Depending on the domain where the models are applied, explaining the predictions could be of interest/needed. Tools for this are **treeinterpreter** https://github.com/andosa/treeinterpreter and **Lime** https://github.com/marcotcr/lime. Extending the preprocessing to focusing more on outliers and imputing the missing text features could also improve upon the results. Predicting the gender could be used in applications for marketing purposes,identification, automated customer services.