­­­­­­­­

Accent Recognition: Application of Classification, Unsupervised Learning, and Hybrid Data Analysis

202177000

July 2022

ME-315

Table of Contents

[1. Introduction 3](#_Toc108112266)

[2. Data 3](#_Toc108112267)

[2.1 Response Variable 3](#_Toc108112268)

[2.2 Features 3](#_Toc108112269)

[3. Analysis 6](#_Toc108112270)

[3.1 Supervised Learning 6](#_Toc108112271)

[3.1.1 Discriminant Analysis 6](#_Toc108112272)

[3.1.2 Multinomial Regression 6](#_Toc108112273)

[3.1.3 k-Nearest Neighbors 6](#_Toc108112274)

[3.1.4 Bagging 7](#_Toc108112275)

[3.1.5 Random Forest 7](#_Toc108112276)

[3.1.6 Results 8](#_Toc108112277)

[3.2 Unsupervised Learning 9](#_Toc108112278)

[3.2.1 K-Means Clustering 9](#_Toc108112279)

[3.2.2 Principal Component Analysis 10](#_Toc108112280)

[3.3 Hybrid Analysis 11](#_Toc108112281)

[4. Conclusion 12](#_Toc108112282)

[4.1 Future Work 13](#_Toc108112283)

# Introduction

The field of audio and signal processing is one in which machine learning is a powerful tool. Using machine learning, researchers can provide insights into the patterns of large sets of complicated data and draw conclusions which would otherwise be hidden from researchers due to the complexity of the data. This paper will analyze a specific set of audio data and try to determine if supervised, unsupervised, and a hybrid of both types of machine learning can provide insights into this data.

The data in question will be pre-processed audio data meant for classifying participant accents into one of six categories based on their auditory pronunciation of English words. The data was gathered for and published in a paper by Ernest Fokoue and Zichen Ma. The raw audio data was processed into numerical scores using a signal processing method to calculate a signal’s MFCC or Mel-Frequency Cepstral Coefficient. The goal of Fokoue’s study was to try and use these MFCC’s to differentiate speaker accents from 6 categories- American, English, German, Italian, French, and Spanish, into two binary categories- American, and Non-American. He performed three methods of analysis for this binary classification problem (Discriminant analysis, support vector machines, and k-nearest neighbors) and determined that the k-nearest neighbors algorithm is the most accurate in identifying accents based on MFCC’s for this dataset.

This study will attempt to expand the analysis performed by Fokoue and Ma by using more classification methods to try and identify accent by MFCCs. Another expansion of the original study will be in the attempt to classify more than just the binary response of American vs non-American. This study will make use of multiple methods of machine learning spanning unsupervised and supervised techniques as well as a combination of the two. Like in the original study, we will also be evaluating the data with discriminant analysis (QDA and LDA) and k-nearest neighbors (kNN). In the end, the analysis will either support the conclusion that kNN has the highest accuracy rate or find another method between cluster analysis, factor analysis, multinomial regression, bagging, and random forests which is more effective at classification/identification on what is likely a more difficult classification task.

# 2. Data

## 2.1 Response Variable

The dataset collected by Fokoue and Ma contains 329 instances of processed audio signal. Each instance represents 1s of a spoken English word. Over the course of the experiment, 15 words were spoken by 22 participants. One of the instances for one of the Spanish speakers is not present in the published data for unspecified reasons. For the sake of simplicity and drawing conclusions, this incongruity will be essentially ignored. Of the participants: 11 were male and 11 were female; 11 were American, 3 from the UK, 2 German, 2 French, 2 Italian, and 2 Spanish. The language for the accent was treated as the response variable. This variable was predicted both as a 6-part categorical variable and a binary variable to indicate American or Non-American.

## 2.2 Features

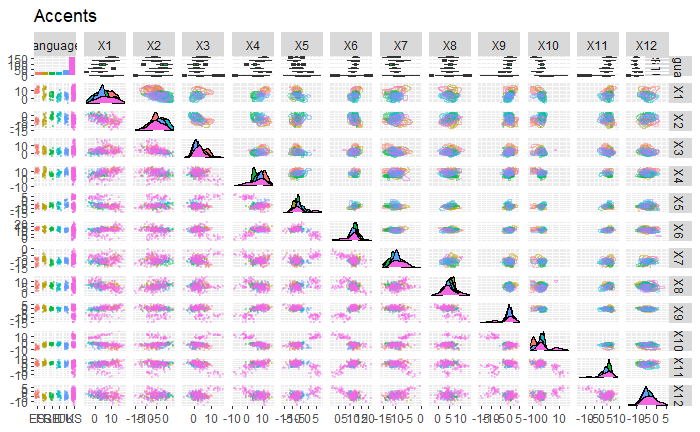
The features of this dataset were the MFCCs or the Mel-Frequency Cepstral Coefficients. According to Fokoue and Ma, “The main idea of MFCC is to transform the signal from time domain to frequency domain and to map the transformed signal in hertz onto Mel-scale due to the fact that 1 kHz is a threshold of humans’ hearing ability. Human ears are less sensitive to sound with frequency above this threshold. The calculation of MFCCs includes the following steps:

* Pre-emphasis filtering;
* Take the absolute value of the short time Fourier transformation using windowing;
* Warp to auditory frequency scale (Mel-scale);
* Take the discrete cosine transformation of the log-auditory-spectrum;
* Return the first q MFCCs.”

The dataset contains 12 different MFCC features for each of the 329 signals. The features are not strongly correlated with one another and their distributions can be visualized with figures 1, 2, and 3. As seen in figures 1 and 2, the individual MFCCs (X1, X2, …., X12) are not strongly correlated with each other but the means show that there is reason to suspect some difference between groups of signals by accent. The means are all positive but varying and the spread of each group is also fairly different. This variation among groups of interest may present interesting results in the unsupervised section of the analysis.

**Figure 1.** *Correlation Matrix for All Features of Dataset*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **X1** | **X2** | **X3** | **X4** | **X5** | **X6** | **X7** | **X8** | **X9** | **X10** | **X11** | **X12** |
| **X1** | 1 | -0.52 | 0.11 | 0.47 | -0.42 | 0.37 | 0.08 | -0.47 | 0.39 | -0.36 | 0.16 | -0.24 |
| **X2** | -0.52 | 1 | -0.48 | -0.13 | 0.09 | -0.11 | 0.05 | 0.35 | -0.11 | 0.08 | -0.01 | 0.02 |
| **X3** | 0.11 | -0.48 | 1 | -0.3 | 0.33 | -0.32 | 0.13 | 0.04 | -0.3 | 0.37 | -0.6 | 0.17 |
| **X4** | 0.47 | -0.13 | -0.3 | 1 | -0.52 | 0.59 | -0.28 | -0.32 | 0.67 | -0.74 | 0.56 | -0.44 |
| **X5** | -0.42 | 0.09 | 0.33 | -0.52 | 1 | -0.6 | 0.41 | 0.44 | -0.77 | 0.74 | -0.53 | 0.44 |
| **X6** | 0.37 | -0.11 | -0.32 | 0.59 | -0.6 | 1 | -0.51 | -0.4 | 0.71 | -0.71 | 0.63 | -0.54 |
| **X7** | 0.08 | 0.05 | 0.13 | -0.28 | 0.41 | -0.51 | 1 | 0.22 | -0.53 | 0.53 | -0.41 | 0.45 |
| **X8** | -0.47 | 0.35 | 0.04 | -0.32 | 0.44 | -0.4 | 0.22 | 1 | -0.52 | 0.3 | -0.03 | 0.06 |
| **X9** | 0.39 | -0.11 | -0.3 | 0.67 | -0.77 | 0.71 | -0.53 | -0.52 | 1 | -0.76 | 0.5 | -0.41 |
| **X10** | -0.36 | 0.08 | 0.37 | -0.74 | 0.74 | -0.71 | 0.53 | 0.3 | -0.76 | 1 | -0.67 | 0.44 |
| **X11** | 0.16 | -0.01 | -0.6 | 0.56 | -0.53 | 0.63 | -0.41 | -0.03 | 0.5 | -0.67 | 1 | -0.39 |
| **X12** | -0.24 | 0.02 | 0.17 | -0.44 | 0.44 | -0.54 | 0.45 | 0.06 | -0.41 | 0.44 | -0.39 | 1 |

**Figure 2.** *Visual plot of entire dataset using ggpairs*

Note. Scatterplot in bottom left quadrant, histograms along diagonal and boxplots along the uppermost horizontal layer. Plot is very busy and lower axis is difficult to read but the overall point is more to show the general shape of the data with the visuals.

**Figure 3***. Table showing the center and spread of the MFCCs for each grouping of participants*

|  |  |  |  |
| --- | --- | --- | --- |
| **Accent** | **Number** | **Mean** | **Standard deviation** |
| Spanish | 29 | 9.60069505575862 | 5.94479727245324 |
| French | 30 | 6.98044954436667 | 13.7513720315757 |
| German | 30 | 2.6561074344 | 5.03121434054137 |
| Italian | 30 | 0.5516317259 | 8.60941209142909 |
| UK | 45 | 0.334636833644444 | 7.011279061399 |
| US | 165 | 8.21177939830303 | 6.73641825929222 |

*Note*. Mean and standard deviation columns are the result of taking the arithmetic mean and standard deviation of every MFCC feature for every element in each accent language group.

# 3. Analysis

## 3.1 Supervised Learning

The analysis will begin with an attempt to classify and model the data using supervised learning techniques like linear discriminant analysis, quadratic discriminant analysis, multinomial regression, k-nearest neighbors, cross-validated trees, bagging, and random-forests. We will compare accuracies as the main metric for determining the best model. For this section, many training and test ratios were explored and the results across them were similar but the data presented (when not cross-validated by nature of the modeling technique) is from a test size of 50 and training size of 279. For a summary of the best method by test subset size, see figure 8 in the results. (see section 3.1.6).

### 3.1.1 Discriminant Analysis

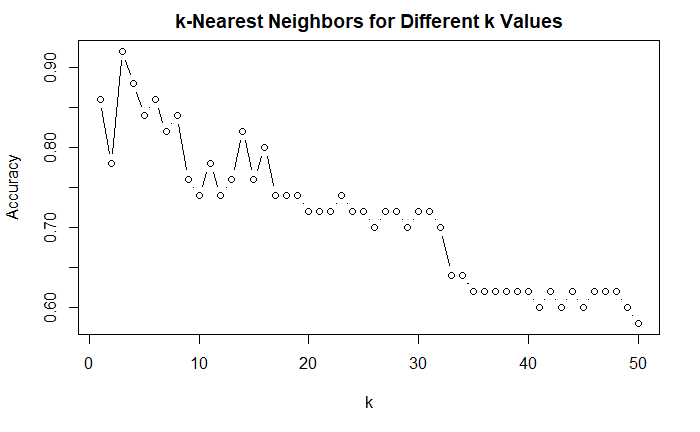
A linear discriminant analysis was performed using the “lda” function in R. The model was constructed by fitting to the training data and the accuracy attained from its application to the test data can be seen in the results in figure 8. (see section 3.1.6) The same applies for the quadratic discriminant analysis which was performed using the “qda” function.

### 3.1.2 Multinomial Regression

A multinomial regression model was fitted to the training data and the accuracy attained from its application to the test data can be seen in the results in figure 8. (see section 3.1.6)

### 3.1.3 k-Nearest Neighbors

The k-Nearest Neighbors (kNN) method was then fitted using the training subset and then run with different parameters to gain the best image of this method on the model. As seen in Figure 4, the lower values of k perform significantly better than the higher values and even achieve an accuracy close to 90% on the test data. This strong performance corroborates the research performed by Fokoue and Ma.

**Figure 4**. *Test accuracy for Different k Values*

### 3.1.4 Bagging

Bagging, or bootstrapping aggregation, was then used as a precursor to random forest. This method was performed using the varying tree numbers but the default n = 500 trees proved to be sufficient in getting the highest accuracy as running more trees only increased runtime for no decrease in accuracy for up to 30,000 trees. The resulting confusion matrix can be seen below in Figure 5.

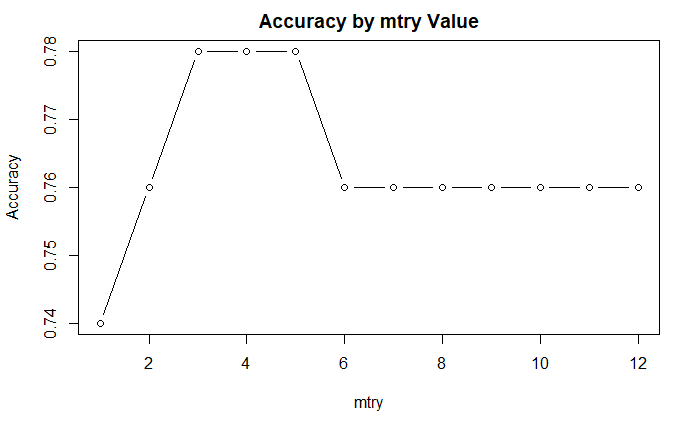
**Figure 5.** *Confusion Matrix for Bagging with n=500*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ES** | **FR** | **GE** | **IT** | **UK** | **US** | **Class Error** |
| **ES** | 18 | 0 | 0 | 0 | 0 | 6 | 0.25 |
| **FR** | 1 | 16 | 0 | 0 | 1 | 9 | 0.407407407407407 |
| **GE** | 0 | 0 | 19 | 1 | 0 | 7 | 0.296296296296296 |
| **IT** | 0 | 0 | 3 | 15 | 0 | 8 | 0.423076923076923 |
| **UK** | 0 | 0 | 0 | 1 | 22 | 13 | 0.388888888888889 |
| **US** | 1 | 1 | 3 | 1 | 5 | 128 | 0.079136690647482 |

### 3.1.5 Random Forest

The random forest method is often effective at improving the accuracy of a model because of how it takes samples of both features and instances to maximize the forms of the information processed. The random forest algorithm performs slightly better than the bagging algorithm, illustrated in Figure 6.

**Figure 6.** *Random Forests Checking `m` Features at Each Step*



*Note*. The last node for mtry = 12 is equivalent to the bagging algorithm (see section 3.1.4)

**Figure 7**. *Confusion Matrix for Random Forest (m = 4)*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ES** | **FR** | **GE** | **IT** | **UK** | **US** | **Class Error** |
| **ES** | 17 | 0 | 0 | 0 | 0 | 7 | 0.291666666666667 |
| **FR** | 0 | 19 | 0 | 0 | 0 | 8 | 0.296296296296296 |
| **GE** | 0 | 0 | 16 | 1 | 0 | 10 | 0.407407407407407 |
| **IT** | 0 | 0 | 3 | 16 | 0 | 7 | 0.384615384615385 |
| **UK** | 1 | 1 | 0 | 2 | 20 | 12 | 0.444444444444444 |
| **US** | 1 | 0 | 1 | 0 | 7 | 130 | 0.064748201438849 |

### 3.1.6 Results

First, we should compare the accuracies of the varying methods applied so far.

**Figure 8.** *Accuracy Chart*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Method** | LDA | QDA | multinomial | KNN (k=1) | KNN (k=3) | KNN (k=10) | best KNN | bagging | best random forest |
| **Accuracy** | 0.72 | 0.82 | 0.72 | 0.86 | **0.92** | 0.74 | **0.92** | 0.76 | 0.78 |

*Note*. The chart can be broken down into 4 main categories, represented above in colors. The blue corresponding to discriminant analysis, brown for multinomial regression, green for k-Nearest neighbors, and yellow for bootstrapping based methods.

As shown in the accuracy chart, the k-Nearest Neighbors method was indeed the best at predicting the accent of a signal based on the MFCCs. This result could be due to many latent and underlying factors. One explanation for this is that a lack of linear or quadratic trend leads the discriminant analysis methods to have no advantage in predicting accent. The kNN performs surprisingly well when compared to bagging and random forest which are powerful and generally effective machine learning tools. Another potential explanation for the failure of bootstrapping methods to outperform kNN is the fact that this test/training split is 50/279 and there might not be enough data points in the training set to prepare the bootstrapping algorithm to predict on the test set. Now let us take a look at the best performing method for varying ratios of test to training data.

**Figure 9.** *Model Performance by Test Set Size*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Set Size** | **Method** | LDA | QDA | multinomial | KNN (k=1) | KNN (k=3) | KNN (k=10) | best KNN | | bagging | best random forest |
| 15 | **Accuracy on Test Data** | 0.87 | 0.87 | 0.72 | 0.86 | **0.92** | 0.74 | **0.92** | 0.80 | | 0.80 |
| 30 | 0.80 | 0.83 | 0.80 | 0.83 | **0.90** | 0.73 | **0.90** | 0.83 | | 0.87 |
| 50 | 0.72 | 0.82 | 0.72 | 0.86 | **0.92** | 0.74 | **0.92** | 0.76 | | 0.78 |
| 100 | 0.69 | 0.75 | 0.72 | **0.84** | 0.81 | 0.75 | **0.84** | 0.72 | | 0.75 |
| 150 | 0.69 | ----- | 0.62 | **0.81** | 0.75 | 0.65 | **0.81** | 0.72 | | 0.75 |
| 200 | 0.70 | ----- | 0.64 | **0.81** | 0.74 | 0.68 | **0.81** | 0.69 | | 0.71 |

*Note*. QDA fails to have enough values in the training set in order for it to work past a certain point. This point occurs somewhere between 179 and 229 values in this case.

Upon further analysis, we see that kNN consistently outperforms all other modeling and classification methods with respect to this dataset. For all examined sizes of the training or test data, the k-Nearest Neighbors algorithm proves to predict above the 80% accuracy level while other methods show about a 10% drop in accuracy from the low (15-50) test size range to the higher (100-200) test size range. This drop is intuitive to machine learning processes as they gain more information and fit data better when they are given more time to train on sample cases before being applied to test data.

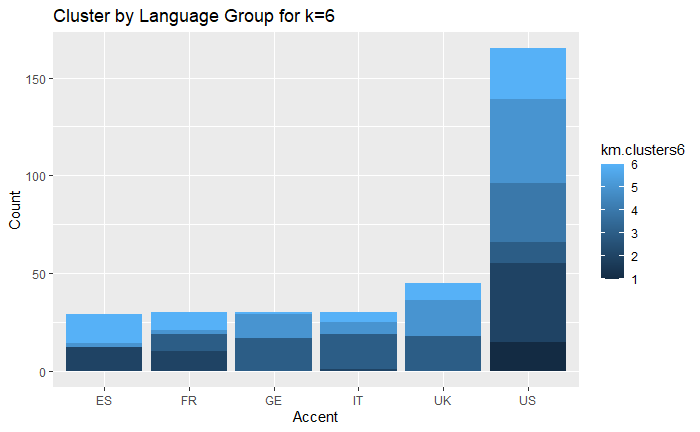
## 3.2 Unsupervised Learning

This section will attempt to add another dimension to the analysis by using unsupervised machine learning methods such as k-means clustering and principal component analysis so try and shed new light on the data.

### 3.2.1 K-Means Clustering

We can use k-means clustering to try and group the data without knowing the accent category that each instance falls under. Given each speaker has 15 instances of signal data associated with them, we can hope that clustering will be effective at identifying instances with common traits. We will perform two different k-mean clustering methods for two different goals. First, we can try and separate the data into 6 distinct groups and hopefully these groups can be mapped onto the distribution of accents that we know are present. Figure 10 shows the results of this clustering on the entire dataset.

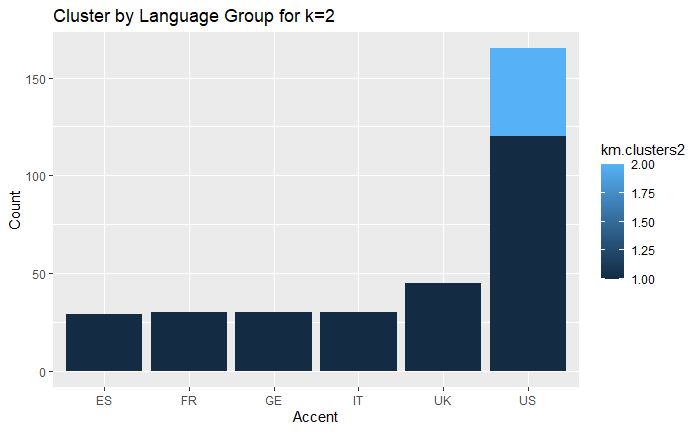
**Figure 10.** *Stacked Bar Plot of 6 Clusters by Group*



This image shows a somewhat disappointing clustering pattern when applying K-means to the accent signal data. Each accent group has more than one cluster represented inside of it and the 165 US signals are practically evenly distributed among the 6 clusters. There was an obvious attempt made by the algorithm to cluster related accent groups given that most groups were only split among a few clusters. However, based on the chart as a whole, we are inclined to conclude that the clustering method using 6 cluster groups is not very helpful in assessing our data.

One potential way for the clustering algorithm to redeem itself is to apply it to the data whilst only separating into two distinct clusters. Maybe this way the algorithm will be able to split the data into American accents and non-American accents. Though before applying this procedure, it does seem unlikely given the 6-group clustering’s poor performance within the American group shown by Figure 10. Here is the same table but with only 2 clusters:

**Table 11.** *Stacked Bar Plot of 6 Clusters by Group*



This time we see that k-means has successfully put every single non-American into one cluster. Unfortunately, it was only able to do so by juicing the size of one cluster until it had 284 instances while the second cluster only had 45. In a way, this is better performance than before but we are still forced to conclude that k-means clustering is not a very viable option for this specific set of data.

### 3.2.2 Principal Component Analysis

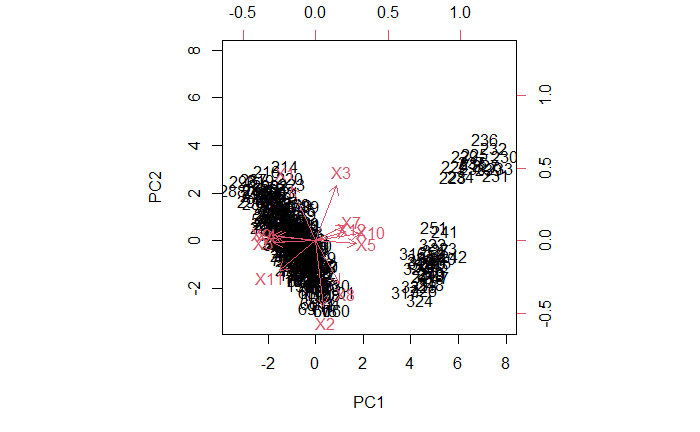
The goal of applying PCA to the dataset is to reduce the dimensions of our data whilst still covering a bulk of the information contained within the data. We can derive principle components which are linear combinations of the values of the MFCCs and potentially simplify our model. Another potential benefit to PCA is that this new set of variables will be more independent than the original set as the disproportionate dimension loss to information loss is due to a targeting of unnecessary, dependent information within the data.

**Figure 12.** *Importance of Principal Components*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | **PC9** | **PC10** | **PC11** | **PC12** |
| **Standard deviation** | 2.343 | 1.427 | 1.021 | 0.943 | 0.803 | 0.701 | 0.657 | 0.568 | 0.474 | 0.425 | 0.38 | 0.30 |
| **Proportion of Variance** | 0.45 | 0.16 | 0.086 | 0.074 | 0.053 | 0.041 | 0.036 | 0.026 | 0.018 | 0.015 | 0.01 | 0.007 |
| **Cumulative Proportion** | 0.45 | 0.62 | 0.714 | 0.788 | 0.842 | 0.883 | 0.919 | 0.946 | 0.965 | 0.980 | 0.99 | 1.000 |

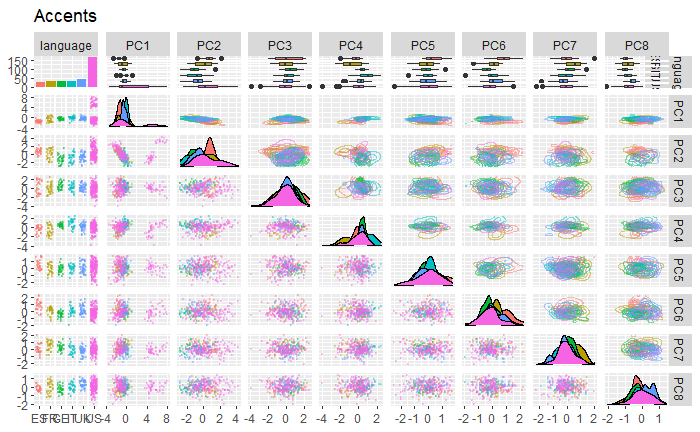
With the goal being to reduce the dimensions of our features, we would like to remove as many components as possible whilst still covering as large a cumulative portion of the variance as possible. It looks like the PCA will only explain 95% of the variance at around 8 of the most influential principal components. This would be a dimension reduction from 12 but a substantial one. In figure 13 we can see the potential benefit of this method in identifying clusters of data points.

**Figure 13.** *Accents Data Visualized by First Two Principal Components*



## 3.3 Hybrid Analysis

The goal of this section is to combine the unsupervised method of PCA with the same supervised learning classification methods (see section 3.1) to try and better predict accent from MFCCs. The idea is that the dimension reduction of using a subset of principal components may simplify the model and help improve accuracy by accounting for most of the variance in the data whilst providing a more independent set of features upon which we build our models. We will use the first eight principal components as they still explain ~95% of the variance in the data. We will perform the same analysis from section 3 and show the two most important plots: the data visualized and the final comparison of method accuracy.

**Figure 14.** *Visualization of Principal Component Adjusted Data*

**Figure 15**. *Final Accuracy Chart*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Machine Learning Method** | LDA | QDA | Multinomial | KNN (k=1) | KNN (k=3) | KNN (k=10) | Best KNN | Bagging | Best Random Forest |
| **Classification**  **Accuracy** | 0.62 | **0.82** | 0.62 | 0.74 | 0.76 | 0.74 | 0.78 | 0.80 | 0.78 |

The accuracy chart indicates that overall, classification accuracies were lower for the PCA data, with Quadratic Discriminant Analysis being the only model that did not experience a decrease in accuracy. The K-Nearest-Neighbor method suffered a significant decrease in accuracy, while multinomial and LDA continued to underperform. The hybrid analysis method did not enhance accuracy; however, it maintained accuracy in the QDA case and simplified the model by using fewer features. The lack of improvement in the predictive performance of our hybrid model suggests that the information loss during the PCA process was not compensated for by the increased resistance to multicollinearity provided by the greater independence of features. This is not entirely unexpected, considering that the original MFCCs did not exhibit a strong correlation with one another during our initial testing.

# 4. Conclusion

The analysis performed in the original paper by Fokoue and Ma was expanded in multiple ways by this report. Their classification problem was one of binomial logistic regression and they concluded that k-Nearest Neighbors was the most accurate method at predicting and classifying audio data in the form of MFCCs when compared to two other methods. This paper has come to a similar conclusion but tested against many other methods of machine learning. We also see that a similar accuracy can be achieved with kNN on a 6-fold classification problem rather than a binomial one. Fokoue and Ma observed an 85% average accuracy when using 12 MFCC features and kNN. We saw a 92% accuracy at best and a similar average accuracy for kNN in general. Their prediction was for a binomial variable of American vs non-American so it is interesting to see that a similar accuracy can be achieved with question that requires even more precision in prediction.

Another expansion of the work done by Fokoue and Ma is the inclusion of unsupervised learning methods in this report. While the clustering and PCA methods did not yield significantly more accurate models, they did still tell us a little more about our data. Additionally, their failure to predict or classify the data as well as kNN serves to, yet again, solidify k-Nearest Neighbors as a very strong method of classification. Even in a dataset with more than ten features and across binomial and multinomial predictive goals, kNN proves its value in comparison with many competitive and effective modeling techniques.

## 4.1 Future Work

The work performed on this dataset has been expanded by this paper but there are still many ways in which the data could be analyzed. One idea would be to attempt other methods of unsupervised learning and then try and incorporate these methods with the regression performed by supervised techniques to make a more effective hybrid analysis. One initial idea would be to try and find the best clustering method and then make an interaction term for the cluster or clusters in order to test if this interaction would improve the predictive capability of the supervised learning techniques. The clusters described in this research would likely not help much in this department but it is not entirely clear.

The conclusion that k-nearest neighbors is the best model at making predictions based on this data set with these constraints, but it would be interesting to test this method on similar scenarios. You could run a similar experiment with a different method of signal to scale conversion. You could also run kNN on a dataset without Americans present or perhaps with other participants/demographics and see if it still outperforms other methods of prediction. There are of course many other machine learning techniques which might be helpful in accent recognition and categorization. This is a very exciting application of machine learning and it has quite a lot of potential to grow in accuracy and complexity.

References

Fokoue, E. (2020). UCI Machine Learning Repository

[https://archive.ics.uci.edu/ml/datasets/Speaker+Accent+Recognition#](https://archive.ics.uci.edu/ml/datasets/Speaker+Accent+Recognition). Irvine, CA: University of California, School of Information and Computer Science.

Ma, Zichen, & Fokoué, Ernest. (2014). A comparison of classifiers in performing speaker accent

recognition using MFCCs. *Open Journal of Statistics,* *4*, 258–266.