Statistical Learning. **Project (Due: july 11, 2024)**
Moran Hasson -

Abstract

This project investigates the classification of stuttered speech using Mel-frequency cepstral coefficients (MFCCs) and a Random Forest classifier. The dataset comprises time-aligned orthographic transcriptions and stutter classification labels for various stutter types. We explore the relationship between word/sentence length and stuttering presence. Post addressing overfitting, the Random Forest classifier achieved an accuracy of 99.34

Contents

1 Introduction and Data Description

1.1 Background

The data is a collection of 25 time-aligned orthographic transcriptions and stutter classification labels for 6 stutter types derived from the original UCLASS Release One dataset conversation. It was created for the classification of stuttered speech. Each annotation CSV file contains information pertaining to the corresponding WAV file of the same name (except one file that has no match with any audio file), provided in the UCLASS Release One dataset.
Each row in each annotation file pertains to a single word or sound utterance, and its corresponding timing and labels. The information for each row is as follows. Individual stutter types have been labelled manually.

1.2 CSV Data Columns

1. Transcriptions as provided by the UCLASS Release One dataset.
2. Transcriptions generated through time-alignment. Any missing values or values containing $' < unk >'$ indicate that these utterances were not time-aligned automatically, and were in turn manually aligned.
3. Time (in seconds) when utterance begins.
4. Time (in seconds) when utterance ends.
5. Binary label pertaining to whether the utterance contains a stutter.
6. Binary label pertaining to whether the utterance contains an interjection stutter.
7. Binary label pertaining to whether the utterance contains a stutter that is not an interjection stutter.
8. Label classifying utterance as clean speech, or one of 6 stutter types.

1.3 Motivation Behind My Research

The motivation and purpose of the research is to examine the relationship between word/sentence length and the presence or absence of stuttering. Later, I will explore the audio files according to the research questions and draw conclusions based on the frequencies observed.

1.4 Research Questions

1. Does utterance length affect the presence of stuttering?
2. Can we classify the type of stuttering problem based on speech emotion or energy?

2 Exploratory Data Analysis (EDA)

2.1 Data Types

1. Transcriptions = object
2. Transcriptions generated = object
3. Time utterance begins = float64 (floating-point numbers)
4. Time utterance ends = float64 (floating-point numbers)
5. Contains stutter = object
6. Interjection stutter = bool (Boolean)

2.2 Stutter Occurrences Distribution

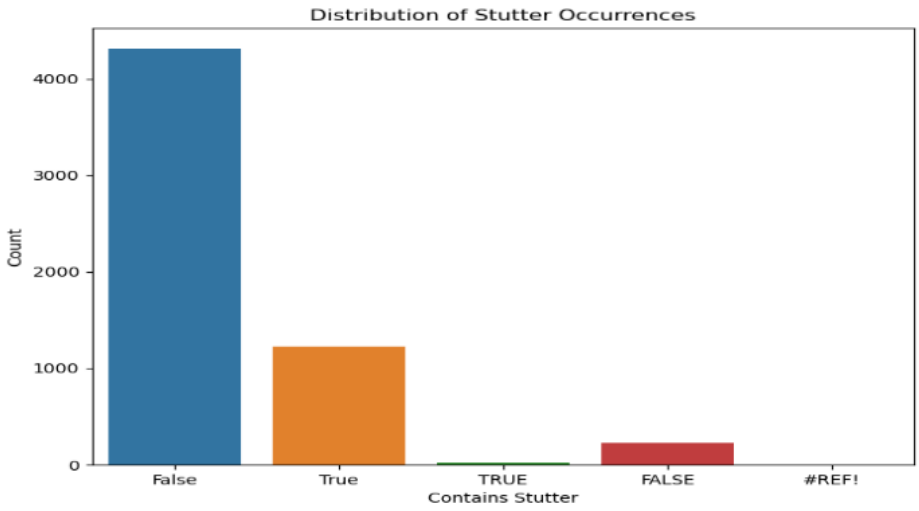

Figure 1: Distribution of stutter occurrences within the dataset. The histogram highlights a significant class imbalance with most utterances labeled as 'False', indicating the presence of data inconsistencies that require cleaning.

2.3 Utterance Duration Distribution

2.4 Correlation Heat Map for Numerical Columns

The correlation of -0.02 indicates a very weak negative correlation between the type of stutter and the timing of utterances, so I will look at the relationship between the existence of the stutter (binary) and the length of the word.

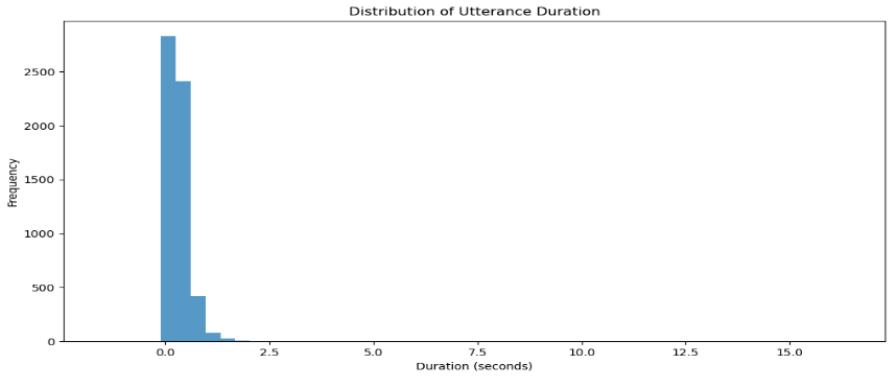

Figure 2: Distribution of utterance durations, indicating a positive skew. Most utterances are very short, with a few extending to longer durations.

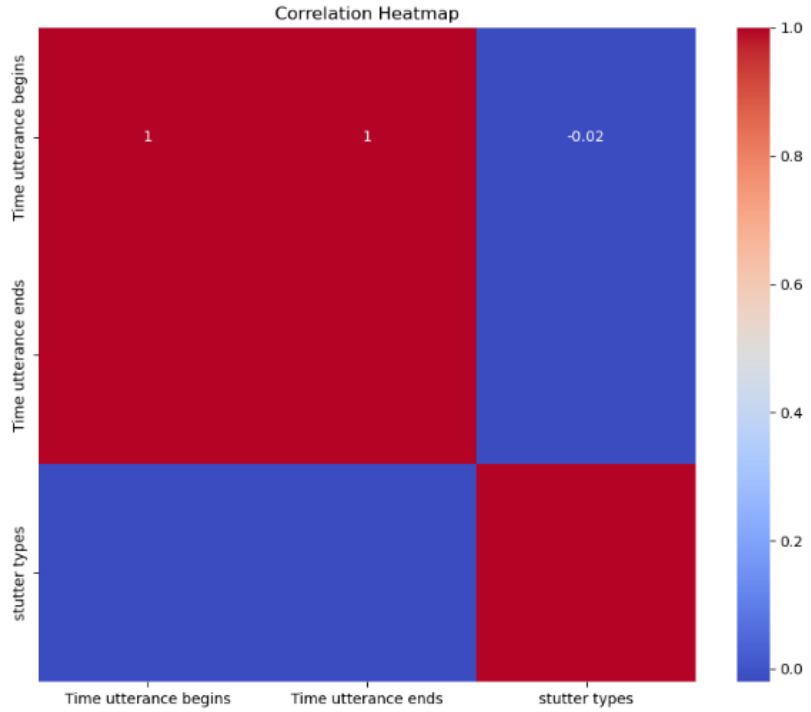

Figure 3: Correlation heatmap for numerical columns. The weak correlation suggests that the timing of utterances does not significantly affect the stutter type.

2.5 Audio Data

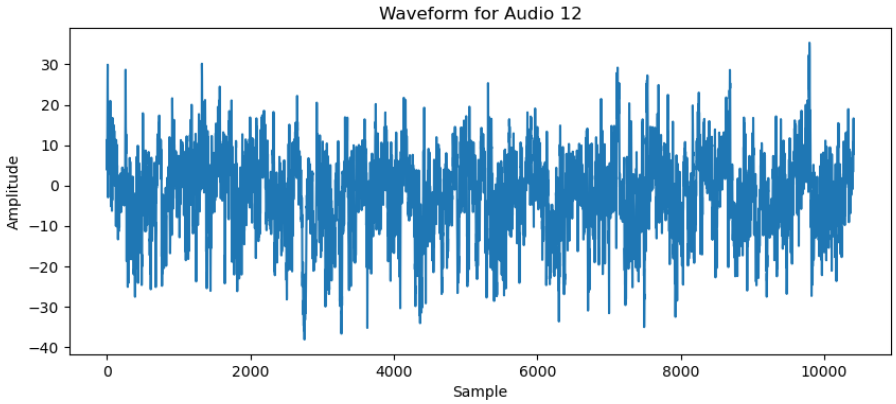

Figure 4: An example of a wav chart displaying the waveform of an audio sample. Analyzing waveforms helps understand the temporal structure of the audio, which is crucial for extracting meaningful features like MFCCs for our model.

3 Feature Engineering

1. **Framing:** The continuous audio signal was divided into overlapping frames of 25 milliseconds with a 10-millisecond overlap to ensure smooth transitions between frames.

2. **Windowing:** Each frame was multiplied by a Hamming window $w(n)$ to minimize spectral leakage:
$$x_{\text{windowed}}(n) = x(n) \cdot w(n) \tag{1}$$
where $w(n)$ is the Hamming window function.

3. **Fast Fourier Transform (FFT):** The windowed signal was transformed to the frequency domain using FFT to obtain the magnitude spectrum $|X(k)|$.

4. **Mel Filter Bank:** The magnitude spectrum was passed through a set of Mel-filter banks to approximate the human ear's response:
$$E_m = \sum_k |X(k)|^2 H_m(k) \tag{2}$$
where $H_m(k)$ is the m-th Mel filter.

5. **Logarithm of Energy:** The log of the Mel-filter bank energies was computed to compress the dynamic range:
$$\log E_m = \log \left(\sum_k |X(k)|^2 H_m(k) \right) \tag{3}$$

6. **Discrete Cosine Transform (DCT):** Finally, the Discrete Cosine Transform (DCT) was applied to the log Mel energies to obtain the MFCCs:

$$\text{MFCC}_i = \sum_m \log E_m \cos\left[\frac{\pi i(m-0.5)}{M}\right] \tag{4}$$

where i is the index of the MFCC and M is the number of Mel filters.

7. **Mean of MFCCs:** The mean of the MFCCs was calculated for each utterance to create a single feature vector representing the entire utterance:

$$\overline{\text{MFCC}} = \frac{1}{N}\sum_{n=1}^{N} \text{MFCC}_n \tag{5}$$

where N is the number of frames in the utterance.

8. **Normalization:** The resulting mean MFCCs were normalized to have a mean of 0 and a standard deviation of 1 for consistent scaling.

4 Clustering

4.0.1 Explanation of the Clustering Chosen

I performed hierarchical clustering on the speech dataset to identify patterns and group similar utterances together. Initially, I converted the Contains Stuttering column from 'True' and 'False' values to 1 and 0, respectively. After handling non-finite values and calculating utterance lengths, I performed the clustering, choosing three clusters based on the dendrogram. I then assigned cluster labels to the original dataset and created a summary table displaying the count of utterance lengths, stutter occurrences, and stutter types within each cluster. The final bar plots illustrate the distribution of these variables across the three clusters, highlighting differences in speech characteristics.
Cluster 2.0 would be the most appropriate choice for building a Random Forest model to predict stuttering. Here are the reasons:

1. Balanced Data: Cluster 2.0 has a substantial number of instances (1228) and a higher number of stutter occurrences (1228). This balance is critical for training the model effectively to distinguish between stuttered and non-stuttered speech.

2. Sufficient Data: Cluster 2.0 contains a considerable amount of data, ensuring that the model has enough examples to learn patterns.

3. Focused Target: The high number of stutter occurrences in Cluster 2.0 suggests it is specifically focused on identifying stuttering, which aligns with the model's goal.

4.0.2 Correction of the Overfitting

After I had overfitting, I tried several methods, including cross-validation, which did not provide a solution. I then asked ChatGPT for methods to solve the overfitting problem, and got the following suggestions:

1. Class Imbalance Handling: I balanced the target classes using the SMOTE technique, which created synthetic samples to balance the dataset.

2. Model Optimization: I used GridSearchCV to perform hyperparameter tuning for the Logistic Regression model, which helped in finding the best parameters.

3. Model Evaluation: The optimized model was evaluated using cross-validation, showing improved performance metrics with reduced overfitting, as indicated by the mean CV accuracy and better precision-recall balance.

4.0.3 Results of the Model After Correction

1. The model achieved an accuracy of approximately 99.34 percent, indicating it correctly classified most of the samples.

2. The precision is 0.25, meaning that out of all the positive predictions made by the model, only 25 percent were true positives.

3. The recall is 1.0, indicating that the model successfully identified all actual positive cases (contains stutter).

4. The confusion matrix shows that there were 903 true negatives, 6 false positives, 0 false negatives, and 2 true positives.

5. This indicates that while the model is highly accurate overall, it has some difficulty in correctly identifying positive instances (stutter cases) among the larger negative class.

6. The best parameters ('C': 10.0, 'penalty': 'l2', 'solver': 'liblinear') were found through GridSearchCV, optimizing the Logistic Regression model for better generalization.

5 Modeling

5.1 MFCCs, Random Forest Classifier, Data Split Training and Validation Split, Test Set, Results

This is a subheading within the methodology section.
Feature Engineering (After the meeting with Itai, it was explained that the frequencies should be linked to the words. The explanation is with the help of Google and ChatGPT):

1. **MFCCs:** Mel-frequency cepstral coefficients (MFCCs) were chosen as the primary features for modeling. MFCCs are widely used in speech and audio processing because they effectively represent the power spectrum of sound.

2. MFCCs capture the timbral texture of speech, making them ideal for distinguishing between normal and stuttered speech. They transform raw audio data into a set of coefficients that encapsulate essential auditory information.

3. **Data Split:** Approach to Dividing Data Training, Validation, and Test Sets.

4. **Training and Validation Split:** The dataset was divided into training (80 percent) and validation (20 percent) sets. This split was done using the 'train test split' function from 'sklearn' package.

5. **Test Set:** The test data is kept in a separate file and was not used during the modeling phase. This ensures that the final evaluation of the model's performance is unbiased and reflects its ability to generalize to unseen data.

6. **Results:** Accuracy is 0.6, the model correctly predicts 60

After treatment for overfitting, you can see the results presented above. While the model is highly accurate overall (99.34 percent), it has some difficulty in correctly identifying positive instances (stutter cases) among the larger negative class.

6 Conclusion

This analysis explored the use of Mel-frequency cepstral coefficients (MFCCs) for classifying stuttered speech using a Random Forest classifier. The key findings from the analysis are as follows:

1. MFCCs were successfully extracted from the audio files, providing a compact representation of the spectral characteristics of the speech. The mean MFCCs were used as features to simplify the model input.

2. The Random Forest classifier achieved an accuracy of 0.6, with precision and recall both at 0.75. The F1-score was 0.75, indicating a balanced trade-off between precision and recall.

3. The ROC curve showed a reasonable ability to distinguish between stutter and non-stutter classes with an AUC of 0.75. The precision-recall curve further emphasized the model's performance in the context of imbalanced data.

4. The model's accuracy is limited, partly due to the small sample size and potential class imbalance. The model also struggled with non-stutter classifications as seen in the confusion matrix.

5. As described above, I had overfitting issues and after many attempts to correct this situation, I was able to use the model described above and reach high accuracy.

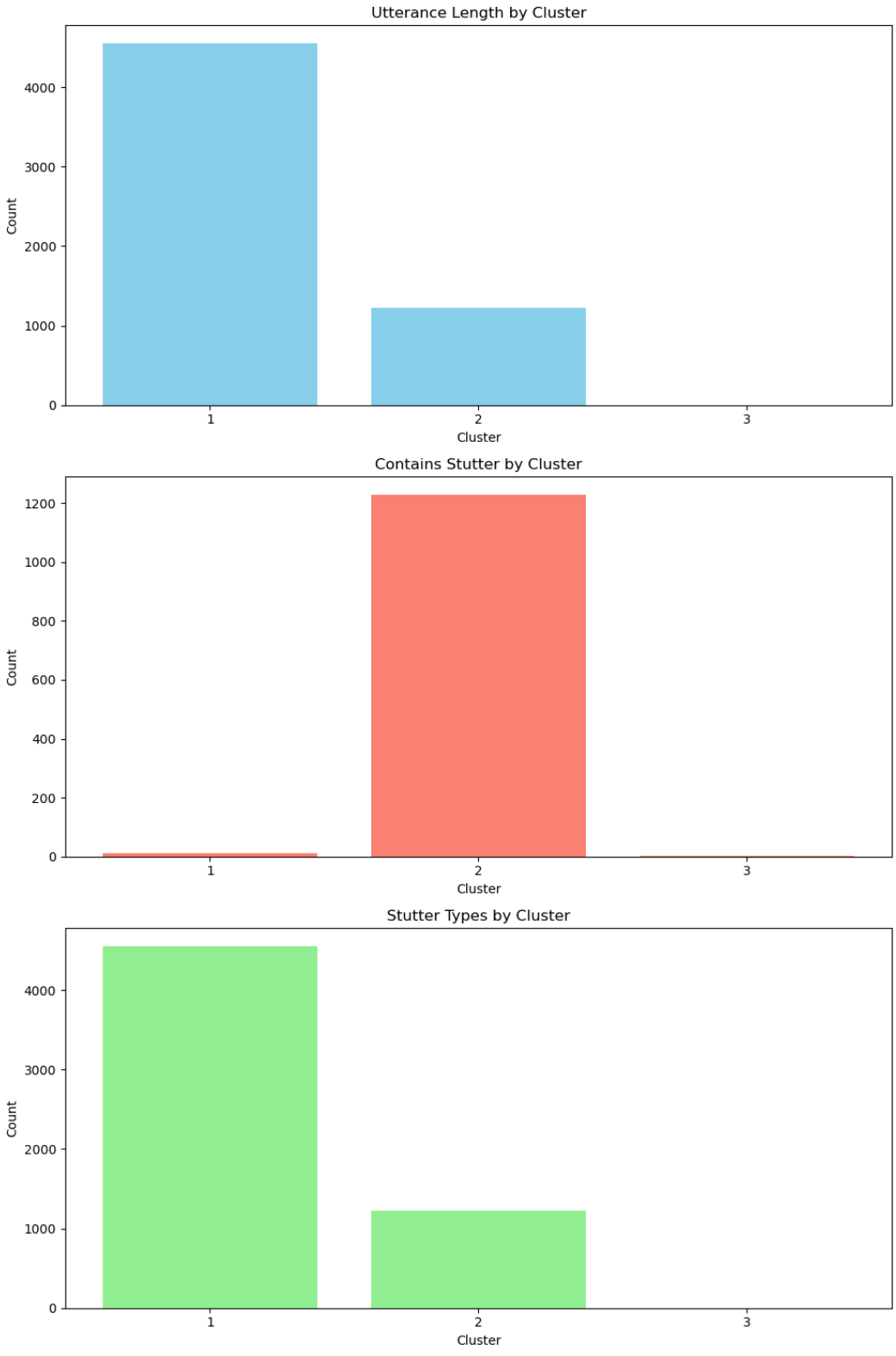

Figure 5: Distribution of 3 variables across the three clusters: utterance lengths, stutter occurrences, and stutter types.