Speech Phoneme Analysis and Classification

Etienne Cauchi - 5603H

Feature Extraction

I used Praat software to get the necessary formants from the data files. I extracted the first 3 formants for 5 accents, where for each accent, I chose 5 speakers per gender and for every speaker I used 3 vowel sounds. In my case, I used the words heed, head, and had. I saved this data in a csv file, which was then read into a python program.

Necessary Code

The code necessary to do the necessary analysis can be found in the file 'General.py'.

The code utilises the sklearn and pandas libraries. Pandas was used to read the csv file and store the results. Sklearn was used to split the data into a training and test set, to classify k-nearest neighbours and to create confusion matrices and calculate F1 scores.

The code is well commented, and therefore, I will only summarise the algorithm.

The data collected from Praat that had been stored in a CSV file is read into a Pandas DataFrame. The DataFrame is split into training and test sets using the `train_test_split()` method from the `sklearn.model_selection` library.

The KNN algorithm is run five times for 2 different distance metrics and for every run, the data is split randomly into training and test sets. For each split variation, the algorithm is run for different values of k (3 to 7) to identify the optimal value of k.

The performance of the KNN model is evaluated using F1 scores and confusion matrices. The results of each run are stored in a Pandas DataFrame and exported to an Excel file for further analysis.

Finally, the confusion matrices are analysed to determine which vowel phonemes were confused the most, in preparation for question 4.

What distance metric did you use? Tried any others?

I opted to answer this question before question 1 in the brief, as this decision influenced what I did in that question.

I used an Excel PivotTable to analyse the table generated by the Python script. From this, I concluded that the ideal model would be using cityblock (Manhattan) distance.

The average of the F1 scores is displayed underneath, and from this, it resulted that the average score using cityblock was over 5% more accurate than euclidean.

	cityblock	euclidean
Average of F1 score	0.6518	0.6001

How does performance change with different values of K?

I created a simple PivotTable where I filtered data to only consider cityblock distance, since that is the metric that I concluded is better, and so, it is the only metric I want to finetune.

From this table, I concluded that the best value of k is k=6, with k=7 being a close second. Choosing a good value for k is very important, as can be seen below; if for example one chooses k=5, they will have lost 5% accuracy, just for the fact of having chosen a bad value of k.

	3	4	5	6	7	Average
Average of F1 score	0.6447	0.6183	0.6059	0.6343	0.6268	0.6260

How does performance change when classification is done one data for a single gender alone, or when data from both genders are put together?

I made a copy of the original Python script and called it ‘per gender’. I altered it to split genders before any computation in order to get separate results for male and female speakers.

To examine the results, I once again used a PivotTable (displayed below) and compared the results from the 'per gender' script to the results from the 'General' script. This shows that splitting the data by gender results in a better classification for each value of k. If one were to compare fine-tuned parameters; k=5 and using cityblock as the distance metric, higher numbers will be seen, as expected, but the difference between the split by gender and the combined scores is roughly the same when a percentage change is calculated.

Split?	3	4	5	6	7	Average
Per Gender	0.6523	0.6405	0.6809	0.6458	0.6665	0.6572
Combined	0.6447	0.6183	0.6059	0.6343	0.6268	0.6260

What are the vowel-based phonemes that produce the most confusion?

```
{'IY': {'EH': 176, 'AE': 25}, 'EH': {'IY': 218, 'AE': 114}, 'AE':
{'IY': 22, 'EH': 96}}
```

I used a part of the python script to answer this question, with the results being shown above. It was apparent that:

- IY was most confused with EH by a big margin (EH: 88%, AE: 12%).
- EH was more confused with IY than AE, but by a slightly closer margin (IY: 66%, 34%).
- AE was much more confused with EH than IY (EH: 81%, IY: 22).
- IY was confused a total of 201 times, EH was confused 332 times, and AE has been confused 118 times; in percentage form this presents itself as: IY: 31%, EH: 51%, AE: 18%.
- Between IY, EH, and AE, EH was most confused by a very large margin.