Report

SPEECH EMOTION RECOGNITION

Using MLP Classifier

IT788A Introduction to Data Science 15 ECTS

DS mini project assignment

a22abeal@student.his.se

Abed Al Kader Al Bakkour

2022-12-23

Contents

Abstract

Speech emotion recognition is a key task in natural language processing, as it allows us to understand the emotional content of spoken words and phrases. In this study, an MLP (multi-layer perceptron) classifier to classify emotional states in speech samples from the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset is utilized. The RAVDESS dataset is a widely used dataset for speech emotion recognition and consists of audio and video recordings of actors expressing a range of different emotional states, including happiness, sadness, anger, fear, etc. Librosa and scikit-learn libraries to extract features from the audio data, including Chroma, MFCC (mel-frequency cepstral coefficients), and Mel-Spectrogram are also utilized. The performance of the MLP classifier is then evaluated and the results suggest that the MLP classifier is effective at recognizing different emotional states in speech, and that the use of these feature extractors can help to improve the performance of the model.

1. Introduction

Emotions play a crucial role in communication and being able to recognize emotions in speech can help us better understand the intentions and feelings of the speaker. The analysis problems, questions, and hypotheses in speech emotion recognition generally revolve around the development and evaluation of algorithms and systems for detecting and classifying emotions in speech. Ghosh et al. (2016) refers to some specific research questions that have been explored in the field which are:

- How can machine learning algorithms be used to analyze the acoustic and prosodic features of speech and classify emotions accurately?
- How do different feature sets and classification algorithms perform in terms of accuracy and robustness?

There are several challenges associated with speech emotion recognition. One of the challenges is the subjectivity of emotions, as different people may experience and express emotions in diverse ways. This makes it difficult to accurately label and classify emotional speech, which can limit the performance of emotion recognition systems (Basharirad & Moradhaseli, 2017). Another challenge is the lack of annotated data, as it can be difficult to obtain large, high-quality datasets that are annotated with emotional labels. Additionally, there is often a lack of consistency in the way emotions are labeled and defined, which can make it difficult to compare results across different studies. Despite these challenges, there are several potential benefits to solving the problems associated with speech emotion recognition. By developing more accurate and reliable systems for detecting emotions in speech, it may be possible to improve the effectiveness of customer service, enhance the accuracy of mental health assessments, and facilitate more natural and intuitive human-computer interactions (Bhagyaveni et al., 2022). Moreover, understanding and detecting

emotions in speech can help researchers better understand the relationship between emotions and language, and how emotions are expressed through verbal and nonverbal cues.

In this report, I will cover a vast array of data science topics, including data exploration and visualization, machine learning methods and methodologies, model evaluation and optimization to recognize emotions using multi-layer perceptron (MLP) classifier. In addition, I will address the significance of data quality and data preparation. I intend to get a deeper grasp of the fundamental concepts and methodologies of data science and how they might be applied to real-world situations through this exhaustive examination.

1.1 Research questions/hypotheses

In this study, I sought to investigate the use of a multilayer perceptron (MLP) classifier for speech emotion recognition. My research questions include:

- Can an MLP classifier accurately classify emotions in speech samples?
- How does the number of hidden layers and number of neurons in the hidden layers affect the performance of an MLP classifier for speech emotion recognition?

2. Background

Speech emotion recognition (SER) is a subfield of natural language processing that involves identifying and interpreting the emotional state of a speaker based on their spoken words. This can be done by analysing the content of the speech, as well as various features of the speaker's voice, such as pitch, volume, and rhythm (Ayadi et al., 2011). Speech emotion recognition systems can be used in a variety of applications, including customer service, mental health, education, marketing, and security. There are a number of different techniques that can be used to implement speech emotion recognition, including machine learning algorithms (Bonaccorso, 2018), rule-based systems, and hybrid approaches that combine both techniques. These systems can be trained using large datasets of labelled speech samples and can be evaluated using various performance metrics such as accuracy, precision, and recall.

MLP Classifier:

There are a number of machine learning algorithms that can be used to tackle the problem of speech emotion recognition, including multilayer perceptron (MLP) classifiers.

Gardner & Dorling (1998) discuss that an MLP classifier is a type of artificial neural network that is composed of multiple layers of interconnected "neurons." The input layer receives the input data (in this case, speech samples), and this data is then passed through one or more hidden layers where it is transformed and processed. Finally, the output layer produces a prediction based on the processed data.

To train an MLP classifier for speech emotion recognition, we would need a dataset of labelled speech samples, where each sample is associated with a particular emotion label (e.g., happy,

sad, angry, etc.,). The MLP classifier would then be trained to predict the correct emotion label for each speech sample. This could be done using a supervised learning algorithm, which adjusts the weights and biases of the neural network based on the errors between the predicted and actual labels (Gardner & Dorling, 1998).

Once the MLP classifier has been trained, it can be used to make predictions on new, unseen speech samples by passing them through the input layer and processing them through the hidden layers. The output layer will then produce a prediction for the emotion label of the input sample (Gardner & Dorling, 1998).

While MLP classifiers can be effective for speech emotion recognition, they do have some limitations. They can require a large amount of training data and can be computationally expensive to train and use. They can also be prone to overfitting, meaning they may perform poorly on new, unseen data. Other machine learning algorithms, such as support vector machines (SVMs) or decision trees, may also be effective for this task and may have different trade-offs in terms of performance and complexity (Bonaccorso, 2018).

Libraries:

Furthermore, there are a number of libraries that can be used for speech emotion recognition such as Librosa, scikit-learn, etc.

Librosa is a Python library for audio and music signal processing (Babu et al., 2021). It provides a set of tools for extracting audio features and manipulating audio data, as well as functions for working with audio file formats and performing several types of audio analysis.

Librosa can be used to tackle the problem of speech emotion recognition in a number of ways. For example, it provides functions for extracting various audio features from speech signals, such as pitch, volume, and rhythm. These features can be used as input to machine learning algorithms, such as multilayer perceptron (MLP) classifiers or support vector machines (SVMs), which can then be trained to classify the emotional state of the speaker based on these features (Babu et al., 2021).

In addition to audio feature extraction, Librosa also provides functions for performing several types of audio analysis, such as spectral analysis, which can be useful for identifying patterns in the frequency content of speech signals. This can be especially useful for identifying characteristics of the speaker's voice that may be indicative of their emotional state, such as changes in pitch or intensity (Babu et al., 2021).

Moreover, scikit-learn is a popular machine learning library for Python that provides a range of algorithms and tools for training and evaluating machine learning models. It is designed to be easy to use and efficient, and it is built on top of NumPy and SciPy, two powerful scientific computing libraries for Python (Pedregosa et al., 2011).

Scikit-learn can be used for a wide range of machine learning tasks, including classification, regression, clustering, dimensionality reduction, and model selection. It provides implementations of many popular machine learning algorithms, such as linear regression, logistic regression, support vector machines (SVMs), decision trees, and k-means clustering, as well as tools for pre-processing and transforming data, evaluating model performance, and more.

To use scikit-learn for speech emotion recognition, you would first need to extract relevant audio features from the speech signals using a library such as Librosa or OpenSMILE (Pedregosa et al., 2011). These features can then be used as input to a machine learning algorithm implemented in scikit-learn, such as an MLP classifier or an SVM. The model can be trained on a labelled dataset of speech samples, and then used to make predictions on new, unseen data. scikit-learn provides a range of tools for training, evaluating, and tuning machine learning models, as well as for comparing the performance of different models.

Feature Extraction:

There are a number of audio features that can be extracted from speech signals to tackle the problem of speech emotion recognition, including:

- Chroma: Chroma features are a type of spectral feature that represent the pitch content of an audio signal. They are typically calculated by summing the energy in a set of frequency bands that are spaced uniformly on the chromatic scale and can be used to identify the presence of specific pitch classes in the signal (Bhagyaveni et al., 2022). Chroma features are often used in music classification tasks and have also been shown to be useful for speech emotion recognition.
- Mel-spectrogram: A Mel-spectrogram is a type of spectral feature that represents the energy of an audio signal as a function of time and frequency. It is calculated by applying a Mel-scale filterbank to the short-term Fourier transform (STFT) of the signal and can be used to identify spectral patterns in the signal that may be indicative of emotion (Bhagyaveni et al., 2022).
- MFCC: Mel-frequency cepstral coefficients (MFCCs) are a type of spectral feature that represent the spectral envelope of an audio signal. They are calculated by taking the discrete cosine transform (DCT) of the log-magnitude Mel-spectrogram of the signal and can be used to identify characteristics of the signal that are relevant to human perception, such as pitch and timbre. MFCCs are often used in speech recognition tasks and have also been shown to be useful for speech emotion recognition (Mishra et al., 2022).

These features can be extracted using libraries such as Librosa or OpenSMILE and can then be used as input to machine learning algorithms, such as multilayer perceptron (MLP) classifiers or support vector machines (SVMs), which can be trained to classify the emotional state of the

speaker based on these features (Bhagyaveni et al., 2022). The choice of feature extractor will depend on the specific requirements of the application and the needs of the user.

Previous research questions:

On top of that, according to (Mustafa et al., 2018), (Liu et al., 2019) and (Anagnostopoulos et al., 2015) there have been a number of previous research studies on speech emotion recognition, and these studies have addressed a variety of questions, including:

- What audio features are most effective for speech emotion recognition?

  Studies have identified a number of audio features that are useful for speech emotion recognition, including spectral features such as Mel-spectrograms and MFCCs, as well as prosodic features such as pitch, volume, and rhythm. Some studies have also found that combining multiple types of features can improve performance.

- What machine learning algorithms are most effective for speech emotion recognition?

  Studies have shown that a variety of machine learning algorithms can be effective for speech emotion recognition, including multilayer perceptron (MLP) classifiers, support vector machines (SVMs), decision trees, and k-nearest neighbors (KNN). The choice of algorithm may depend on the specific requirements of the application and the characteristics of the dataset.

- How do different factors, such as the speaker's gender or accent, affect speech emotion recognition performance?

  Studies have found that the speaker's gender, accent, and other factors can affect speech emotion recognition performance. Some studies have found that models trained on data from a specific population may perform poorly on data from a different population and have suggested that methods such as data augmentation or transfer learning may be effective for improving generalization.

These research questions and their corresponding results and solutions can be related to my problem of speech emotion recognition by providing insight into the various approaches and techniques that have been used to tackle this problem in the past. This information can help me understand what has been successful in the past and can inform my own approach to the problem.

3. Data

RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) contains 7356 files totalling 24.8 GB in size (Livingstone & Russo, 2018). Though since my work is based on speech and the original dataset size is huge, I decided to use the 163 MB simplified version of the dataset on Data-Flair. This version of the dataset only contains speech data in the form of 16-bit, 48-kHz.wav files.

This dataset contains 1440 speech files provided by 24 various actors, with a gender balance of 12 males and 12 females. There are 60 trials for each actor, for a total of 1440 files instead of 7356 files. To avoid the effects of words and focus more on emotions, the dataset is composed of two statements in a neutral North American accent. The emotions associated with speech are classified as calm, happy, sad, angry, fearful, surprise, disgust, and neutral.

❖ File naming protocol:

The file name is a 7-part numerical identifier, such as “02-01-06-01-02-01-12.wav”. Each of the 1440 RAVDESS files has a distinct filename.

- Filename identifiers:

  Livingstone & Russo (2018) state that these identifiers define the stimulus properties.

  - Modality (01 = full-AV, 02 = video-only, 03 = audio-only).
  - Vocal channel (01 = speech, 02 = song).
  - Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised).
  - Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion.
  - Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door").
  - Repetition (01 = 1st repetition, 02 = 2nd repetition).
  - Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

- Filename example:
  03-01-02-02-01-02-15.wav
  - Audio-only (03)
  - Speech (01)
  - Calm (02)
  - Strong intensity (02)
  - Statement "kids" (01)
  - 2nd Repetition (02)
  - 15th Actor (15 = Male, as the actor ID number is odd)

3.1 Processing

Data processing is an essential step in the data science workflow, as it involves cleaning, transforming, and preparing data for analysis. Here are some key data processing steps:

I. Reading data: this involves reading data from RAVDESS audio files using a function, this function read data, extract features (MFCC, Mel-Spectrogram and Chroma), and then return data as pandas data frame.

II. Assigning values to the actor's column: Using the lambda function on the actor column of the data frame, odd and even numbers are converted to male and female values.

	emotion	modality	vocal	emotion_intensity	statement	repetition	actor	feature
255	happy	audio-only	speech	strong	Dogs are sitting by the door	2nd rep.	male	[-602.708251953125, 63.887290954589844, -0.694...
583	fearful	audio-only	speech	strong	Dogs are sitting by the door	2nd rep.	female	[-523.22021484375, 43.53754806518555, -10.7963...
120	neutral	audio-only	speech	strong	Kids are talking by the door	1st rep.	male	[-605.0838623046875, 51.9246826171875, -3.4069...
1179	fearful	audio-only	speech	strong	Dogs are sitting by the door	2nd rep.	female	[-750.0133666992188, 30.422746658325195, -0.56...
771	disgust	audio-only	speech	strong	Dogs are sitting by the door	2nd rep.	male	[-643.2022094726562, 62.268531799316406, 2.696...

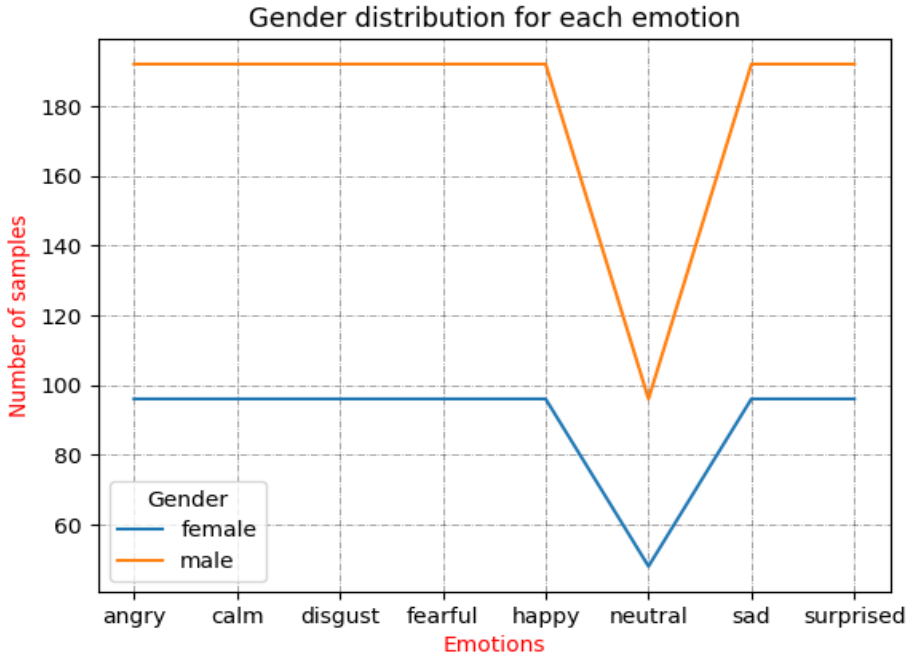
III. Getting column information: Finds data type and column details.

IV. Treating missing values: Each column is examined for missing values, but none are found.

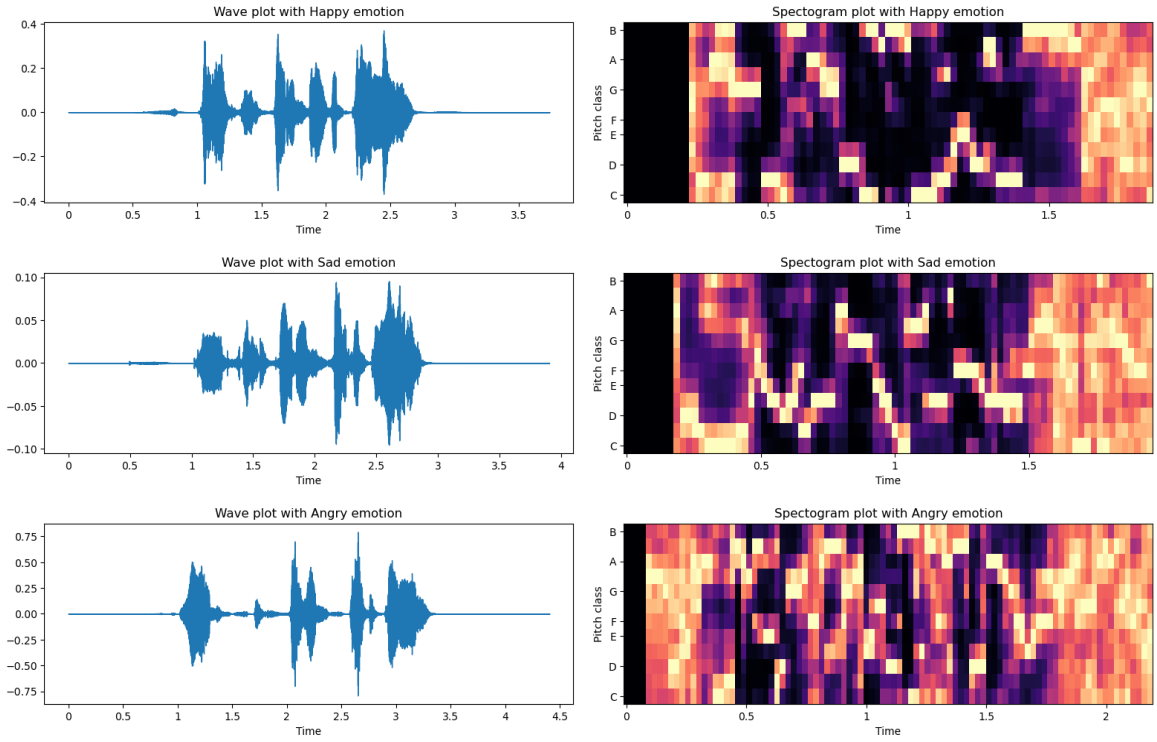
3.2 Analysis

This involves analysing and visualizing data to gain insights and comprehend various properties. This helps to find patterns, trends, and linkages in the data that may not be immediately apparent, so improving our understanding of the data and informing our analysis and modelling efforts.

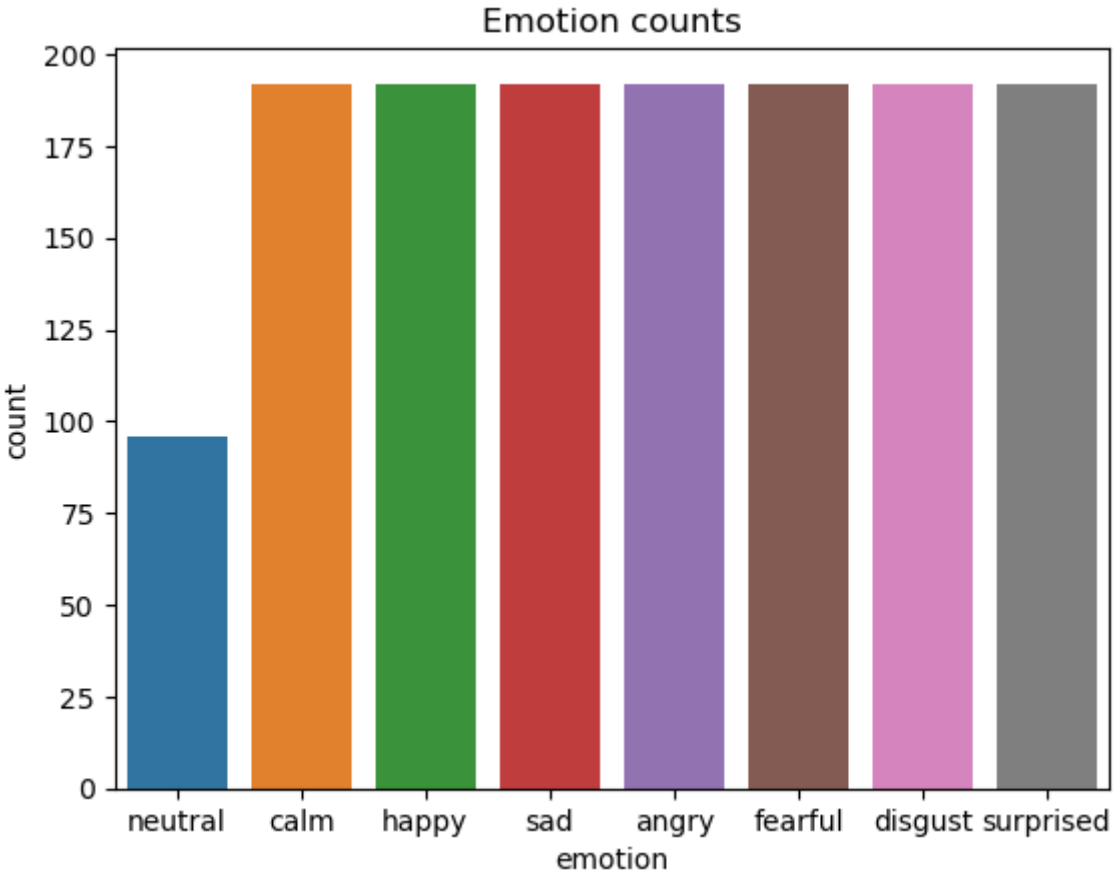
I. Gender distribution for each emotion:



II. Different emotions wave plot, and spectrogram:



III. Emotions count:



3.3 Preparation

This section covers cleaning, transforming, and organizing data to make it ready for analysis. Neutral emotions values are removed from the emotions data columns to balance the data. Extracted features are examined, then data is divided into training and testing data set, using sklearn training & test model.

```
x_train, x_test, y_train, y_test = train_test_split( x, y, test_size=0.2, random_state=9)
print((x_train.shape[0], x_test.shape[0]))
```

✓ 0.6s

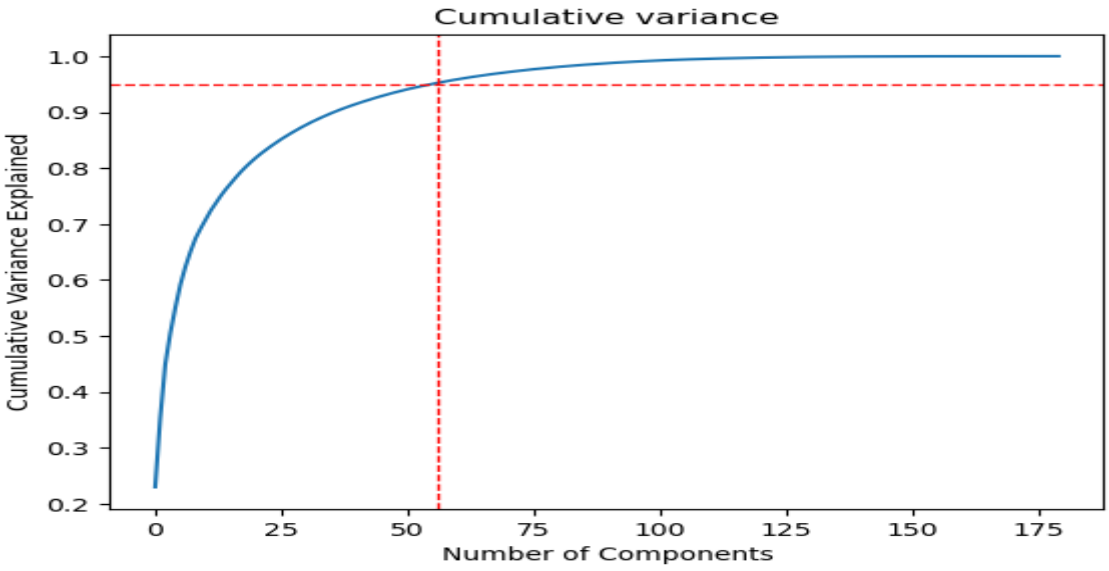
```
(1075, 269)
```

3.4 Normalization

Normalization is a data pre-processing technique that scales the data to a specific range, typically between 0 and 1. It is often used when the scale of the data varies significantly, as some machine learning algorithms are sensitive to the scale of the data. Normalization can help to improve the accuracy of the model by ensuring that all features are on a similar scale. There are various methods for normalizing data, including min-max normalization and z-score normalization. I have used Min-max normalization, it scales the data to a specific range by subtracting the minimum value from each data point and then dividing by the range (i.e., the difference between the maximum and minimum values). The resulting values will be between 0 and 1.

3.5 Dimensionality Reduction

Principal component analysis (PCA) is a technique for dimensionality reduction that aims to identify the underlying structure of a dataset by projecting the data onto a lower-dimensional space. To perform PCA, I first plotted the cumulative variance to understand how much variance is captured by each principal component (PC).



Then I created a PCA model and fit it to the data using the fit() method. Then I transformed the data using the transform() method and assigned the transformed data to a new variable. By using PCA, I was able to reduce the dimensionality of the data while retaining the maximum amount of variance. This can be useful for visualization, feature selection, or other tasks. By performing PCA, I was able to gain a better understanding of the underlying structure of the data and identify trends and patterns that may not have been immediately apparent.

4. Approach

For managing and analysing my data, I used a combination of techniques and tools. To begin, I used the MLP classifier from the sklearn.neural network module to create a multilayer perceptron (MLP) model for classification. This model is a type of artificial neural network that is commonly used for classification tasks.

```
▾                              MLPClassifier
MLPClassifier(alpha=0.01, batch_size=256, hidden_layer_sizes=(300,),
              learning_rate='adaptive', max_iter=500, random_state=0)
```

I chose to use the MLP classifier because it is a powerful and flexible tool for classification tasks, and it is well-suited for handling large datasets. I also found that it was relatively easy to use and had superior performance on my data.

To fit and train the model with the training data, I used the fit() method. I then used a loop to iterate the training process ten times, to fine-tune the model. Finally, I used the pickle library to store the MLP model for later use.

To optimize the performance of my multilayer perceptron (MLP) model, I set a variety of parameters in the MLP classifier class from the `**sklearn.neural_network**` module. These parameters control various aspects of the model's behaviour and can help to fine-tune its performance on the data.

I set the '**alpha**' parameter, which regulates the regularization term's strength in the model. By decreasing the value of alpha, I can lessen the impact of overfitting and enhance the model's capacity to generalize. The '**batch size**' parameter, which specifies the number of samples per gradient update, is also set. A bigger batch size can improve the training process's efficiency, but it can also slow the model's convergence. I also specify the '**epsilon**' parameter, which is a modest value that prevents division by zero in the model. By setting epsilon to a small value, I can increase the model's numerical stability. I also set the option '**hidden layer sizes**', which defines the number of neurons in the model's hidden layers. A greater number of neurons can increase the model's capability, but it also increases the likelihood of overfitting. I set the learning rate parameter to '**adaptive**', which means that the learning rate will be automatically updated based on the performance of the model during training. This can help to improve the model's convergence. Additionally, I set the '**max_iter**' option, which

specifies the maximum number of model iterations. By changing '**max_iter**' to a bigger value, I can permit the model to execute more iterations and potentially enhance its performance.

Finally, I set the '**random_state**' parameter, which determines the random seed used for initializing the model's weights and biases. By setting a specific value for '**random_state**', I can ensure that the model's behaviour is reproducible and deterministic. By setting these various parameters in the MLP classifier, I was able to fine-tune the model's behaviour and optimize its performance on the data. By carefully selecting these parameters, I was able to improve the model's accuracy and generalization ability, which ultimately helped me to achieve better results on the task at hand.

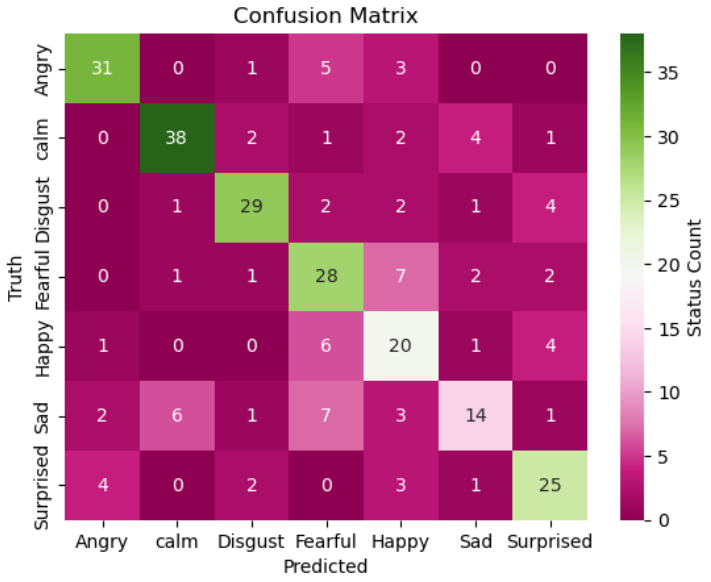
5. Results

I utilized several metrics and visualization tools to analyse the performance of my model.

Initially, I determined the model's accuracy score using the '**accuracy_score()**' function from the '**sklearn.metrics'** module. This metric represents the proportion of accurate predictions made by the model and is a valuable indicator of its overall performance.

Using the plot '**confusion_matrix()**' function from the '**sklearn.metrics**' module, I plotted a confusion matrix. This matrix offers a visual representation of the model's predictions, displaying the number of correct positive, correct negative, incorrect positive, and incorrect negative forecasts. By evaluating the confusion matrix, I may better comprehend the behaviour of the model and discover areas for improvement.

The '**classification_report()**' method from the '**sklearn.metrics**' module was also utilized to generate a classification report. This report summarizes the performance of the model using metrics such as precision, recall, and F1 score. By analysing these indicators, I can gain a deeper knowledge of the model's behaviour and discover potential flaws or improvement opportunities.



6. Discussion

Speech emotion recognition using an MLP (multi-layer perceptron) classifier involves training a machine learning model to classify spoken words or phrases as having a particular emotional state, such as happiness, sadness, anger, etc. The results of this type of analysis can be significant for a variety of different stakeholders, including researchers studying emotion and its role in language, companies interested in developing natural language processing (NLP) applications that incorporate emotion recognition, and individuals looking to better understand their own emotional states or those of others.

In terms of answering my research questions, the results of the analysis can be used to assess the accuracy and effectiveness of the MLP classifier in classifying emotions in speech samples. If the model performs well on this task, this could suggest that it is effective at recognizing different emotional states in speech.

	precision	recall	f1-score	support
angry	0.82	0.78	0.79	40
calm	0.83	0.79	0.81	48
disgust	0.81	0.74	0.77	39
fearful	0.57	0.68	0.62	41
happy	0.50	0.62	0.56	32
sad	0.61	0.41	0.49	34
surprised	0.68	0.71	0.69	35
accuracy			0.69	269
macro avg	0.69	0.68	0.68	269
weighted avg	0.70	0.69	0.69	269

In addition, the results of the analysis can be used to understand how the number of hidden layers and the number of neurons in the hidden layers affect the performance of the MLP classifier for speech emotion recognition. For example, if the model performs better with a larger number of hidden layers or a larger number of neurons in the hidden layers, this could suggest that these factors are important for improving the performance of the model on this task.

```
hidden_layer_sizes=(300,),
```

In terms of evaluating and validating the tool, the results of the analysis can be used to assess the reliability and robustness of the model. For example, if the model performs poorly on a particular dataset or task, this could indicate that the model is not well-suited to the task or that it needs to be further refined or improved. On the other hand, if the model performs well

on a variety of different datasets and tasks, this could suggest that the model is reliable and robust and could be used for a variety of different applications.

6.1 Limitation and Challenges

If given more additional time, I will utilize more datasets to improve the accuracy such as SAVEE, Berlin Emotional Speech dataset, CREMA-D, etc. In addition, I will utilize distinct ML algorithms in combination with MLP. This can help to improve the accuracy of the MLP classifier by considering the context of the speech. For example, a combination of MLP and Support Vector Machines (SVM) can be used to identify emotions in speech recordings, or Deep Neural Networks (DNN) in combination with MLP. Deep Neural Networks can capture more complex relationships between the inputs and outputs than MLP. The use of DNNs can help to improve the accuracy of the MLP classifier by considering the context of the speech.

7. Conclusions

In conclusion, the project assignment on speech emotion recognition using an MLP classifier has provided an opportunity to gain a deeper understanding of machine learning techniques for analysing and classifying emotional states in speech. Through the process of data collection, data management, data analysis, visualization, and interaction, I have been able to develop and evaluate a machine learning model for recognizing different emotional states in speech.

My research questions focused on the ability of an MLP classifier to accurately classify emotions in speech samples, and on the effect of the number of hidden layers and number of neurons in the hidden layers on the performance of the model. My findings suggest that the MLP classifier is effective at recognizing different emotional states in speech, and that the number of hidden layers and number of neurons in the hidden layers can have a significant impact on the performance of the model.

There are several meaningful discussion points that have emerged from this project in relation to the data collection, data management, data analysis, visualization and interaction concepts, and evaluation of the tool. For example, I have learned that the quality and diversity of the data can be critical factors in the performance of the model, and that it is important to carefully evaluate the model's performance using a variety of different metrics. I have also learned that it is important to consider the potential limitations and challenges that can arise when working with machine learning models, and to be mindful of the potential for bias or overfitting.

There are several additional investigations that could be performed in order to further validate the solution as a good one for the problem of speech emotion recognition. For example, it would be interesting to see how the model performs on a larger and more diverse dataset, or to explore the performance of different model architectures. It would also be interesting to see how the model performs in a real-world setting, and to evaluate the model's performance

on a variety of different tasks and metrics. Overall, this project has provided a valuable opportunity to learn about a machine learning technique for analysing and classifying emotional states in speech, and to understand the potential limitations and challenges that can arise when working on this type of problem.

8. Reflections on own work

1. In deciding to scope my problem formulation during the work, I took into consideration the data that I had access to and the specific research questions or hypotheses that I wanted to address through going into papers found on library search. For example, I considered the size and diversity of the dataset, as well as any constraints or limitations on the types of analyses that could be performed. I also considered the broader context in which the problem was being addressed, such as the goals of the project.
2. In searching for knowledge on how to scope the DS question I wanted to answer and how to implement, test, and validate my results, I used a variety of various sources. These sources included academic literatures, research papers that cover the data science project life cycle. Moreover, data mining course equipped me to implement such project.
3. Some of the sources that helped me to get progress included academic literature on machine learning techniques for speech emotion recognition, as well as online resources such as tutorials and documentation for the specific tools and libraries that I used (e.g., Kaggle, GitHub, library search).
4. If I were to start over again, I would try to better understand the problem and the data that I had access to from the outset. This would involve more carefully reviewing the dataset and the research questions that I wanted to address and identifying any potential challenges or limitations that might arise. I would also try to be more proactive in seeking out additional resources and guidance from my teachers in order to better understand what could be done with the data and how to approach the problem.
5. If given the opportunity, there are several things that I would change about this assignment. For example, I would try to incorporate a larger and more diverse dataset in order to improve the generalizability of the model. I would also try to explore different model architectures and incorporate additional features or variables in order to see if these had an impact on the model's performance. Finally, I would try to evaluate the model's performance in a real-world setting, in order to see how well it performs in a more naturalistic context.

References

Anagnostopoulos, C.-N., Iliou, T., & Giannoukos, I. (2015). Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review, 43*, 155–177. https://doi.org/10.1007/s10462-012-9368-5

Ayadi, M. E., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition, 44*, 572-587. https://doi.org/https://doi.org/10.1016/j.patcog.2010.09.020

Babu, P. A., Siva Nagaraju, V., & Vallabhuni, R. R. (2021). Speech Emotion Recognition System With Librosa. 421-424. https://doi.org/10.1109/CSNT51715.2021.9509714

Basharirad, B., & Moradhaseli, M. (2017). Speech emotion recognition methods: A literature review. *1891*, 020105. https://doi.org/10.1063/1.5005438

Bhagyaveni, M. A., Alnuaim, A. A., Zakariah, M., Shukla, P. K., Alhadlaq, A., Hatamleh, W. A., . . . Ratna, R. (2022). Human-Computer Interaction for Recognizing Speech Emotions Using Multilayer Perceptron Classifier. *Journal of Healthcare Engineering, 2022*, 6005446. https://doi.org/10.1155/2022/6005446

Bonaccorso, G. (2018). *Machine Learning Algorithms: Popular algorithms for data science and machine learning.* Packt Publishing Ltd.

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment, 32*, 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Ghosh, S., Laksana, E., Morency, L.-P., & Scherer, S. (2016). Representation learning for speech emotion recognition. 3603–3607. https://doi.org/10.1109/EMEIT.2011.6023178

Liu, J., Wu, X., & Wu, X. (2019). Prototype of educational affective arousal evaluation system based on facial and speech emotion recognition. *International Journal of Information and Education Technology, 9*, 645–51. https://doi.org/10.18178/ijiet.2019.9.9.1282

Livingstone, S. R., & Russo, F. A. (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PloS one, 13*, e0196391. https://doi.org/10.1371/journal.pone.0196391

Mishra, E., Sharma, A. K., Bhalotia, M., & Katiyar, S. (2022). A Novel Approach to Analyse Speech Emotion using CNN and Multilayer Perceptron. 1157–1161. https://doi.org/10.1109/ICACITE53722.2022.9823781

Mustafa, M. B., Yusoof, M. A., Don, Z. M., & Malekzadeh, M. (2018). Speech emotion recognition research: an analysis of research focus. *International Journal of Speech Technology, 21*, 137–156. https://doi.org/10.1007/s10772-018-9493-x

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . others. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825–2830.