Elec 378 2025 Final Project Description

Designed by the Spring 2025 TAs

1 Introduction

The final project gives you a "hands-on" opportunity to apply the techniques you've learned/will learn in machine learning! The project is intended to not only give you an opportunity to demonstrate your ability to apply the course material, but also to strongly encourage you to explore new concepts and techniques. This year, your project is to classify emotional states from short speech recordings.

2 Dataset

The dataset includes around 13,800 .wav recordings labeled by emotion: Angry, Happy, Sad, Neutral, Fearful, Disgusted, and Surprised. Note that classes are imbalanced: for example, there are only 592 recordings for Surprise but 2167 recordings for Angry. Each audio file of a certain emotion class is in the corrensponding folder of that class name. The dataset is split into 80% training data, 10% testing data that is publicly available, and 10% private testing data. **You will only be graded on the 10% of private testing data. However when tuning your model and submitting on Kaggle you will only be able to see your performance on the public testing data.** After the Kaggle competition ends, your models will be evaluated on the private testing data. If you overfit your models on the public testing data, your ranking on the leader board will likely suffer. Keep this in mind and be sure to avoid overfitting!

3 Data Science Process

A good report outlines the entire "data science pipeline," which typically consists of the steps described below. You should follow these steps in order and write about each one in detail in your report.

3.1 Data Exploration

Before doing anything else, you first need to get familiar with your data! You should understand the dimensions and format of the dataset, listen to several individual observations from each category, and experiment with various ways of storing the audio files. This would also be a great time to split your data into training, validation, and test sets. For your training data, please create a csv file that matches the audio files to their corresponding emotion labels. This will help you when you test out your validation set.

3.2 Feature Extraction

In addition to (or instead of) using raw audio files as input to your classifier, you should extract features of each file. What are some features of an audio clip? Which ones might be useful for speech emotion classification?

- Provide a description of every feature you extracted during the project. Make sure to describe what information each feature gives you about the audio. This will help you determine which features are needed and which are irrelevant.
- Compare different sets of features and justify your final selection. Discuss why certain features were included or discarded. A certain combination of features may give you the most information about the audio clips.

3.3 Model Selection

You are required to try at least three different model families for this classification task, including one not covered during class. Justify why you chose to try each model. Compare and contrast their performances. Rather than simply stating in your report which model had the highest accuracy, discuss the strengths and weaknesses of each one. Provide visualizations to support your answers.

In your report, make sure to:

- Quantitatively compare model performance across training, validation, and test sets (public/private)
- Discuss overfitting trends observed across models and datasets. (See more below for overfitting)
- For each model, include key design decisions (e.g., number of neighbors in KNN, tree depth in Random Forest, architecture in MLP, etc.) and model-specific tuning strategies

3.4 Overfitting Considerations

A crucial part of your machine learning pipeline is understanding and mitigating overfitting. Overfitting occurs when your model learns to perform well on training data but fails to generalize to unseen data. You should reflect on the relationship between the number of data points, number of features, and the number of model parameters. A high number of features or complex models with many parameters (e.g., deep neural networks) can easily overfit especially when data is limited or imbalanced.

In your report, include a discussion of:

- The number of features and whether dimensionality reduction (e.g., PCA) or feature selection was considered.
- The number of parameters in your final models.
- Whether the size of the training set supports your model's complexity.
- Strategies used to reduce overfitting (e.g., regularization, dropout, early stopping, data augmentation).

3.5 Complete Pipeline

Describe the entire pipeline, from preprocessing and feature extraction to model selection and hyperparameter tuning, that you chose to submit. Justify why you think this pipeline will generalize well to the unseen data in the private Kaggle dataset.

Include:

- A clearly labeled diagram that illustrates the full pipeline
- A description of how each step (preprocessing, segmentation, feature extraction, model training, hyperparameter tuning) was iteratively refined
- Any changes made late in the process that led to better generalization

3.6 Conclusions

Summarize your final results on the public dataset, providing visualizations where appropriate. Analyze why you believe you saw the results that you did. Specifically, why did the features you chose help distinguish between emotions? Why did your chosen model outperform others? Reflect on what part of the pipeline had the biggest impact (was it feature selection? model tuning?).

Make sure to analyze what performance gaps (e.g., public vs. private test scores) revealed about generalization or overfitting. How well did your models generalize given the overfitting techniques you implemented? If your Kaggle ranking drops by a lot after the competition closes, please explain what you missed in reducing overfitting.

Lastly, reflect on the process. What did you learn about the data science process that surprised you? What would you do differently in the future?

4 Rules and Expectations

- You are not allowed to hard code the true labels of the audio files in the test set in any way.
- You must use and write in your report on at least THREE different models.
- At least one of your models must be one not introduced in class.
- At least one of your models must be non-neural network based.
- You are allowed to use machine learning toolboxes and deep learning packages.
- You are NOT allowed to use pre-trained models.
- You are NOT allowed to use outside data that isn't provided on our Kaggle competition.
- You are NOT allowed to copy another group's code.
- You are, however, allowed to reference external resources, provided you can explain the code and cite it in your report.

5 Grading

- 10% - Kaggle leaderboard accuracy on the private test data.
- 40% - Technical understanding of new models and methods used.
- 50% - General report quality.

6 Report Submission

Report must be submitted as a PDF, along with any supporting code.

7 Tips from the TAs on potentially useful things to consider

- Consider how you can break the provided data up into even more training data. We already have quite a few audio files, but how can you get more training data to work with?
- If your models take very long to train and you want to do quick prototyping with many different models, consider how you can drop certain data temporarily for quick training models. Is all of the data needed to prototye with every kind of model?
- What if you convert the data from time series to visual data? (Think spectrograms).
- TRY TO REDUCE OVERFITTING

GOOD LUCK HAVE FUN!