Classification of Audio Embeddings

Subject: Introduction to Machine Learning

Supervised by: Bùi Duy Đăng, Nguyễn Thanh Tình

Location: Thành phố Hồ Chí Minh, Việt Nam

Trường Đại học Khoa Học Tự Nhiên, tp Hồ Chí Minh

Prepared by: Group 30
Date: May 25, 2025

1 Group Information

This project was completed by the group **Group 30**. The team members are listed below:

Name	**Student ID**
Bùi Kim Phúc	21120112
Lê Hoàng Sơn	21120127

Bảng 1: Group Members

2 About the Project

The objective of this project is to develop a machine learning model to classify audio clips based on the presence of turkey sounds. The input data consists of audio embeddings, each represented as a matrix of shape $[10, 128]$, where 10 is the number of frames and 128 is the dimensionality of each frame's feature vector. These embeddings are extracted from audio clips and provided in a JSON file (`train.json`). The task is a binary classification problem, where the model predicts whether an audio clip contains turkey sounds (`is_turkey = 1`) or not (`is_turkey = 0`). The classification is performed using a logistic regression model implemented with Scikit-learn, leveraging the flattened embeddings (size 1280) as input features. Finally, the model's performance is evaluated on a test set, and the results are submitted in a CSV file (`submission.csv`) for Kaggle competition.

3 Data Preprocessing

The dataset was sourced from a JSON file (`train.json`) containing audio embeddings and binary labels. The preprocessing steps included:

- Loading the JSON data into a pandas DataFrame.
- Extracting `audio_embedding` (shape [10, 128]) and `is_turkey` (binary labels: 0 or 1).
- Flattening each audio embedding to a 1D array of size 1280 (10 × 128).
- Padding or truncating embeddings to ensure a consistent size.
- The data contains a total of 1195 samples and is splitted into 836 training samples (70%), 179 validation samples (15%), and 180 test samples (15%) with shuffling to ensure randomness.

4 Model Description

The models used for this project are a logistic regression classifier and a random forest classifier, both implemented using Scikit-learn. Key details for each model are as follows:

- **Logistic Regression**:

  - **Algorithm**: Logistic regression with the `liblinear` solver, suitable for small datasets.
  - **Input Features**: Flattened audio embeddings of size 1280 (10 frames × 128 dimensions).
  - **Output**: Binary classification (0 or 1) for the `is_turkey` label.
  - **Hyperparameters**: C=1.0, max_iter=1000, random_state=42.

- **Random Forest**:
  - **Algorithm**: Random forest classifier, an ensemble method using multiple decision trees.
  - **Input Features**: Flattened audio embeddings of size 1280 (10 frames × 128 dimensions).
  - **Output**: Binary classification (0 or 1) for the `is_turkey` label.
  - **Hyperparameters**: n_estimators=100, max_depth=None, random_state=42.

5 Platform

The project was developed and executed on the following platform:

- **Google Colab**: The primary development environment, providing a cloud-based Jupyter notebook interface with GPU support for efficient model training and evaluation.
- **Google Drive**: Used for storing and accessing the dataset (`train.json`) and saving the submission file (`submission.csv`).
- **Libraries**: Scikit-learn for model implementation, pandas for data processing, NumPy for numerical operations, and other standard Python libraries.
- **Latex**: The report is formatted using LaTeX for professional presentation.

6 Training

The training process involved the following steps for both the logistic regression and random forest models:

- **Data Loading**: The JSON dataset was loaded using pandas and processed to extract features and labels.
- **Data Splitting**: The dataset was split into:
  - Training set: 70% of the data, including 836 samples.
  - Validation set: 15% of the data, including 179 samples.
  - Test set: 15% of the data, including 180 samples.

  Splitting was performed using Scikit-learn's `train_test_split` with a random state of 42 for reproducibility.

- **Model Training**:
  - **Logistic Regression**: The model was trained on the training set using the `fit` method with the `liblinear` solver.
  - **Random Forest**: The model was trained on the training set using the `fit` method with an ensemble of decision trees.
- **Validation**: Both models were evaluated on the validation set to tune hyperparameters and assess performance.

7 Evaluation Metrics

The model's performance was evaluated using the following metrics:

- **Accuracy**: The proportion of correct predictions on the validation and test sets.

8 Accuracy Report

The performance of both models is reported below:

- **Logistic Regression**:
  - **Validation Accuracy**: 0.9497.
  - **Test Accuracy**: 0.9167.
- **Random Forest**:
  - **Validation Accuracy**: 0.8939.
  - **Test Accuracy**: 0.9000.

9 Submission

Submission files include:

- `submission.csv`: Our best result achieves a test accuracy of 93.347% using the logistic regression model on Kaggle.
- `Turkey30_Scikit_learn.ipynb`: The Jupyter notebook containing the complete code for data preprocessing, model training, evaluation, and submission generation.
- `This report`: The LaTeX report detailing the project, methodology, and results.

10 Conclusion

In this project, we successfully implemented a machine learning model to classify audio embeddings for the presence of turkey sounds. The logistic regression model achieved a validation accuracy of 94.97% and a test accuracy of 91.67%, while the random forest model achieved a validation accuracy of 89.39% and a test accuracy of 90.00%.

Although these models are quite simple, they demonstrate the feasibility of using audio embeddings for classification tasks.

The logistic regression model outperformed the random forest model in terms of accuracy, indicating that a linear approach was effective for this dataset. Future work could explore more complex models like deep learning Neural Networks for potentially better performance.

11 References

- Scikit-learn Documentation: `https://scikit-learn.org/stable/`
- Pandas Documentation: `https://pandas.pydata.org/`
- Google Colab: `https://colab.research.google.com/`
- Kaggle competition page:
  `https://www.kaggle.com/competitions/introduction-to-machine-learning-project-cq-`

12 AI Usage

The project utilized AI tools including GPT-4, Grok, and Claude Sonnet 3.7 for various tasks such as:

- Code syntax support and debugging.
- Report helping and formatting in LaTeX assistance.

We acknowledge the use of AI tools in enhancing our productivity and code quality, while ensuring that the core logic and implementation were developed by the team.

Chat conversations with AI can be found here:

- `https://grok.com/share/bGVnYWN5_161d9f2e-a240-48ec-a2ac-be01d0881371`