[School Logo Placeholder]

Classification of Audio Embeddings Using Logistic Regression

Subject: Machine Learning Project

Date: May 25, 2025

Supervised by: [Teacher Name(s)]

Location: [City, Country]

[School Name]

Prepared by: [Group Name]

1 Group Information

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

Name	Student ID
Bùi Kim Phúc	21120112
Mary Johnson	20210456
Alex Williams	20210789

Table 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.

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.
- Splitting the data into training (70%), validation (15%), and test (15%) sets using stratified splitting to maintain class distribution.

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**: [Insert hyperparameters here, e.g., C=1.0, max_iter=100].
- **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**: [Insert hyperparameters here, e.g., n_estimators=100, max_depth=None, random_state=42].

5 Platform

The project was developed and executed on the following platform:

- **Google Colab**: insert here
- **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.
  - Validation set: 15% of the data.
  - Test set: 15% of the data.

  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**: [Insert validation accuracy here, e.g., 0.XXXX].
  - **Test Accuracy**: [Insert test accuracy here, e.g., 0.XXXX].
- **Random Forest**:
  - **Validation Accuracy**: [0.8939].
  - **Test Accuracy**: [0.9000].

9 Submission File

(`submission.csv`)

10 Conclusion

[Insert your conclusion here, summarizing the project outcomes, model performance, challenges faced, and potential improvements.]

11 References

- Scikit-learn Documentation: `https://scikit-learn.org/stable/`
- Pandas Documentation: `https://pandas.pydata.org/`
- Google Colab: `https://colab.research.google.com/`