MLPC 2025 Task 2: Data Exploration

Team LABORER

Ali Doğukan Seven **Yisong Tang** **Karel Štícha**

Dmitrii Troitskii

Contributions

Ali Doğukan Seven was responsible for Parts 1 and 6. Karel Štícha wrote Part 2 and compiled the final report. Dmitrii Troitskii wrote Parts 3 and 4. Finally, Yisong Tang completed Part 5 and prepared the presentation.

1 Labeling Function

1.1 Label Accuracy

The labeling function was evaluated on a manual subset of five classes (*Speech*, *Dog Bark*, *Siren*, *Rain*, *Car*). We first extracted the top 20 high-frequency keywords for each class and then checked whether those keywords appeared in the free-text annotations. Using this method, the labeling function achieved an accuracy of approximately 97.2%. The small number of mismatches were actually valid events described by lower-frequency keywords, which explains why they were not detected by our simple keyword check. So the labeling function is quite accurate.

1.2 Audio Features

To identify which features were the most pertinent for classification discrimination, I used ANOVA F-scores. Top features included MFCC, embeddings, contrast, flatness, and bandwidth. These features seem to capture elements like pitch and texture of sounds, which come in handy for recognizing sound types.

1.3 Feature Clusters

The higher-ranking features I gave are such sounds within a class tend to group quite nicely. The features thus help solidify a model in distinguishing types of sounds very well.

2 Data Split

2.1 Split Description

To ensure reliable model training, meaningful hyperparameter tuning, and accurate evaluation, we divided our dataset into three distinct subsets: **a training set, a validation set, and a test set**. The training set, comprising 70% of the data, was used to train the model. The validation set, which accounted for 15%, served to tune the model's hyperparameters and select the best-performing classifier. Lastly, the remaining 15% formed the test set, which was used only once to estimate the final performance.

2.2 Information Leakage

Information leakage occurs when information from outside the training dataset contributes to the training of the model. In our case, such leakage would occur from splitting the data at the frame level. Since consecutive frames are extracted

from the same audio file, they are most likely very similar, splitting at the frame level could lead to frames from the same original recording being present in both the training and test sets. This would allow the model to learn patterns specific to individual recordings, resulting in inflated performance metrics and poor generalization. To avoid this, we performed the **data split at the file level**, ensuring that all frames from a specific audio file are assigned exclusively to a single dataset. As a result, the test data truly represent unseen input.

2.3 Obtaining Unbiased Performance Estimates

To ensure an unbiased final performance estimate, we used the test set only for estimating the model's performance. The training set was used solely for model fitting, while the validation set was used exclusively for tuning hyperparameters and model selection. This procedure prevents the model from being optimized for the test data and the final performance estimate reflect the model's true ability to generalize on new, unseen data.

3 Audio Features

3.1 Selected Subset of Audio Features and Selection Process

Selected Features:

- **MFCC**: Mel-Frequency Cepstral Coefficients to capture timbral properties.
- **Embeddings**: Learned representations, possibly from a pretrained model.
- **Spectral Contrast**: Captures differences between spectral peaks and valleys.
- **Spectral Flatness**: Measures how noise-like a sound is.
- **Spectral Bandwidth**: Quantifies frequency spread.
- **Mel-Spectrogram**: A perceptually scaled time-frequency representation.

Selection Process:

- **Feature Diversity**: Combined complementary features covering timbral, spectral, and perceptual aspects of audio.
- **Empirical Evaluation**: Features were loaded from precomputed '.npz' files and concatenated before dimensionality reduction.
- **Dimensionality Handling**: PCA was used to reduce feature size while preserving 95% of the variance, which also helps prevent overfitting.

3.2 Preprocessing Techniques Applied to Audio Features

Preprocessing Steps:

- **Standardization**: All features were normalized using `StandardScaler` to have zero mean and unit variance.
- **Dimensionality Reduction**: Principal Component Analysis (PCA) was applied to compress features while retaining 95% of total variance.
- **Batch Transformation**: The same pipeline was fitted on the training set and then applied to validation and test sets to ensure consistency.

4 Evaluation

4.1 Evaluation Criterion for Comparing Hyperparameter Settings and Algorithms

Chosen Criterion:

- **F1-Score (Macro-Averaged)**: This metric was used to evaluate and compare model performance. It calculates the F1-score for each class independently and then averages them, ensuring that rare and frequent classes are treated equally.

Rationale:

- **Class Imbalance**: Many of the sound events occur infrequently, so metrics like accuracy would not reflect true performance.
- **Balanced Trade-off**: Macro-F1 balances both false positives and false negatives, which is important for multi-label tasks.
- **Cross-Validation Integration**: The F1-score was directly used as the scoring function during hyperparameter tuning via `GridSearchCV`.

4.2 Baseline and Best Possible Performance

Baseline Performance:

- **Definition**: The baseline is defined by a naive model that always predicts the absence of events (all-zero labels).
- **Computation**: Based on the mean positive rate of the validation set across all classes, the baseline accuracy is approximately $1 -$ mean positive rate.
- **Purpose**: This baseline reflects how well a model could perform by simply exploiting class imbalance — without learning meaningful patterns.

Best Possible Performance:

- **Definition**: The best performance corresponds to the highest macro F1-score obtained through model selection and tuning.
- **Estimation**: This was achieved via `GridSearchCV` and cross-validation, using various models and hyperparameters. Perfect classification is unlikely due to label noise, overlapping events and rare classes samples.

5 Experiments

5.1 Linear SVM

- **Hyperparameter:** Regularization coefficient $C \in \{0.01, 0.1, 1, 10, 100\}$.
- **Performance Curve:** Training F1 ≈ 0.7787, Validation F1 ≈ 0.7747 remain stable across C.
- **Over- vs. Under-fitting:** Train F1 $\approx$ Val F1, both well below 1.0 $\rightarrow$ slight under-fitting; no over-fitting observed.
- **Conclusion:** $C = 1$ offers a good trade-off; further tuning yields no significant gain.

5.2 Decision Tree

- **Hyperparameter:** Maximum depth max_depth $\in \{5, 10, 15, 20, \text{None}\}$.
- **Performance Curve:**
  - depth $= 5$: Train/Val F1 ≈ 0.65 (under-fitting).
  - depth $= 10$: Best validation F1 ≈ 0.674 (Train ≈ 0.807).
  - depth > 10: Train F1 increases, Val F1 decreases (over-fitting).
- **Over- vs. Under-fitting:** Depth < 10 under-fits; depth > 10 over-fits.
- **Conclusion:** Optimal depth $= 10$; deeper trees memorize noise and degrade generalization.

5.3 SGDClassifier (Logistic Regression)

- **Hyperparameter:** L2 regularization strength $\alpha \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.
- **Performance Curve:**
  - $\alpha = 10^{-5}$: Best Train F1 ≈ 0.7483, Val F1 ≈ 0.7480.
  - $\alpha \geq 10^{-4}$: Both F1 scores drop (under-fitting).
  - $\alpha = 10^{-6}$: Train F1 $>$ Val F1 (slight over-fitting).
- **Over- vs. Under-fitting:** $\alpha \gg 10^{-5}$ under-fits; $\alpha \ll 10^{-5}$ over-fits.
- **Conclusion:** $\alpha = 10^{-5}$ balances bias and variance well.

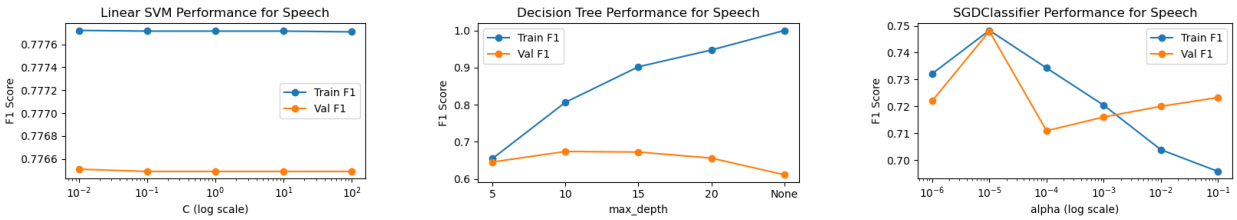

Figure 1: Performance curves for Linear SVM (left), Decision Tree (middle), and SGDClassifier (right).

5.4 Model Comparison

Model	Best Hyperparameter	Validation F1	Avg Training Speed
Linear SVM	$C = 1$	0.7748	44 s/class
SGDClassifier	$\alpha = 10^{-5}$	0.7480	5–90 s/class
Decision Tree	max_depth=10	0.6741	438 s/class

Table 1: Comparison of the three classifiers on the validation set.

5.5 Final Model Performance

The selected model (Linear SVM with $C = 1$) was evaluated on the held-out test set using six features. The overall Macro-F1 score across 58 classes is **0.5381**, reflecting average performance on frequent and rare classes.

6 Analysing Predictions

6.1 File: 451475

The model is weak in detecting some sounds while multiple events interfere. Between 00:05 and 00:08, speech is present but considered not recognized because of its lower volume. From 00:10 to 00:13, loud jackhammer sounds drown exchange horn honks-another sound.

It appears that the model tends to ignore those sounds that are quieter when there is a louder sound. There are sounds that humans simply have difficulty telling apart (e.g., jackhammer vs hammer), and some of this misclassification may be down to this.

6.2 File: 27157

There is much confusion here between speech and motor sounds. The Motor sounds may have similar waveforms to speech. Background noise could be the reason for this. The model perceives indistinct or low-pitched speech as sounds of engines or motor-sounds.

6.3 Conclusion

The classifier has problems with overlapping events and similar acoustic classes. Improvement might be possible with some postprocessing, e.g., averaging predictions over time or filtering out frames with low confidence. The model also hiddered because of some classes have very few samples. Which causing not enough pattern learning examples.