Project Presentation

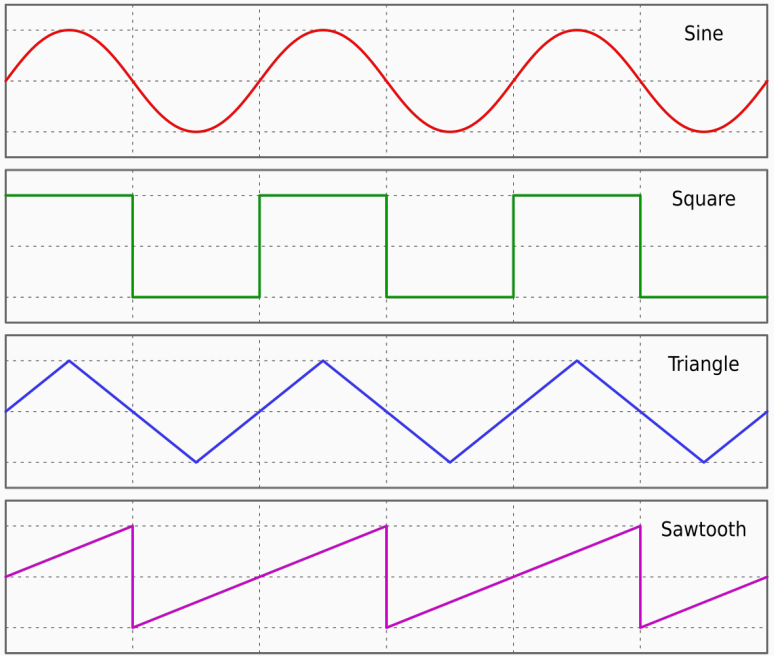
Goal:
Use a data-driven approach to identify features in electrophysiology data that help machine learning models detect synaptic IPSC events, guiding the development of an unsupervised analysis pipeline.

Why Data Science?

Combines elements of statistical techniques, computation, mathematics, and **domain knowledge** to analyze large amounts of data for the purpose of acquiring knowledge, solving problems, and **driving informed decision-making**.

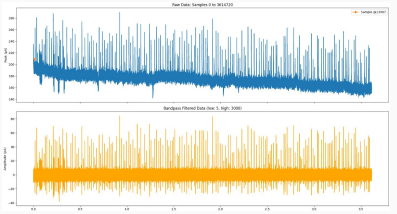
Reasoning:

- If a person can **learn** how to reliably identify synaptic events then there has to be some consistency in how the event shape presents.
- This event's shape possesses aspects or **features** that allow it to be discernible to the human eye.
- Somehow, these features within the data encode for these visually distinguishable synaptic events.

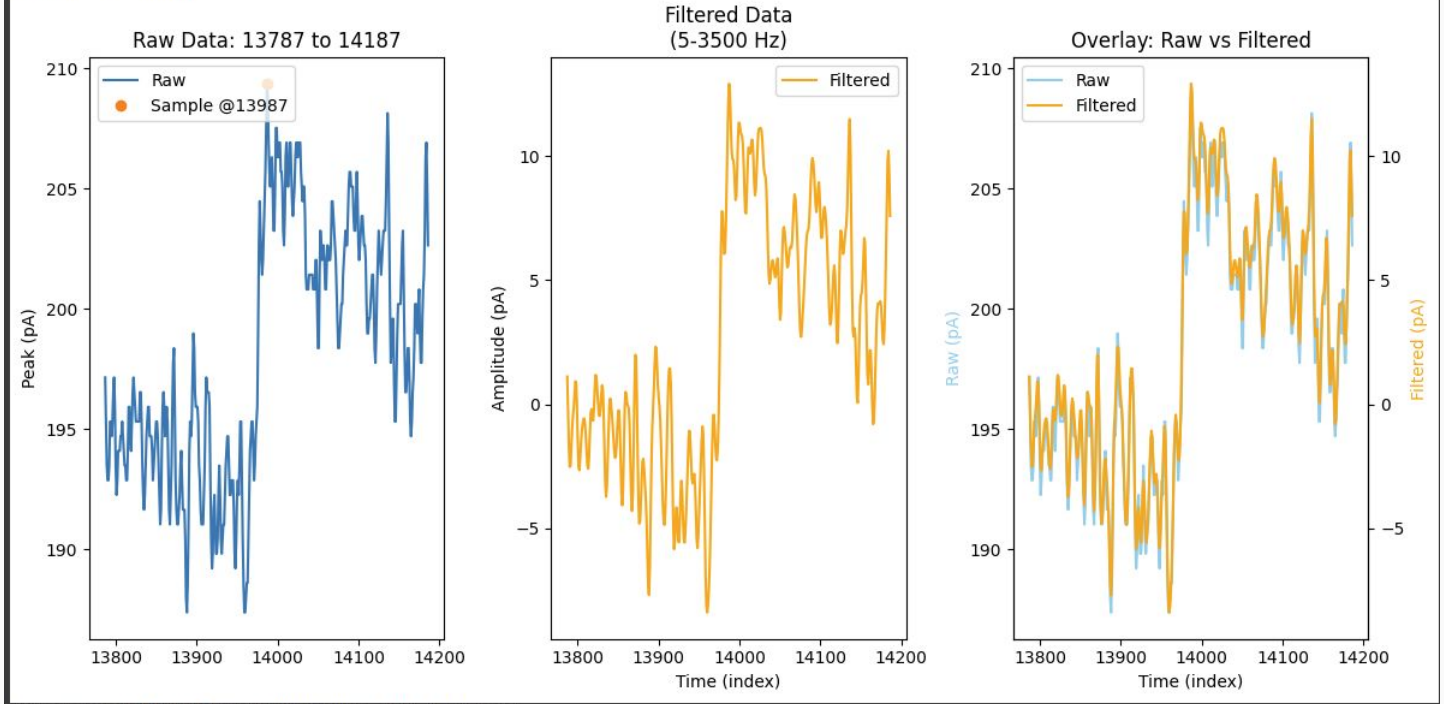


Approach Goals:
Unsupervised as possible / Not using Vision / Interpretable

Data



```
Data: (1, 3590940)
Y-Sweep: (3590940,)
```




```
sucess: /content/2024_03_06_0004_Trunc_ Cell 1 IPSC.abf
```

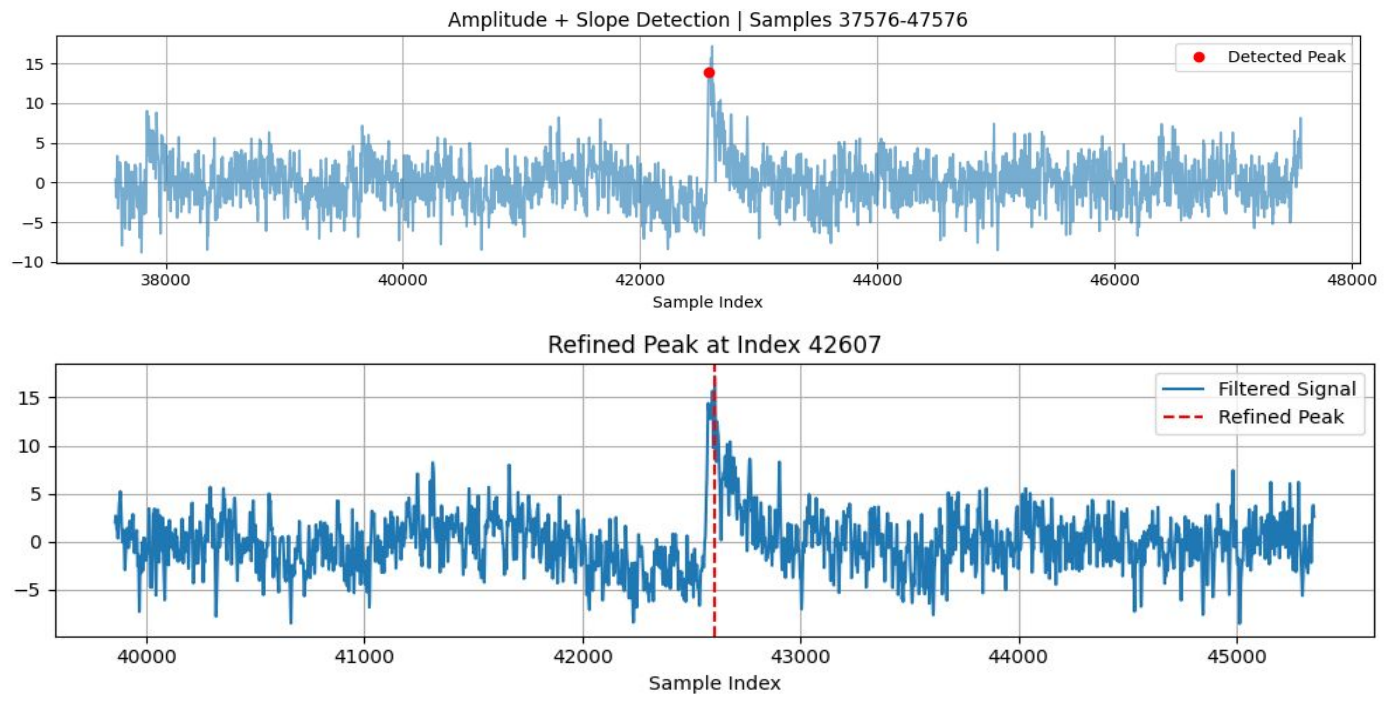
Detect & Refine

```
↳ Threshold Parameters: {'delta': 13, 'slope_thresh': 0.35, 'distance': 400}
```

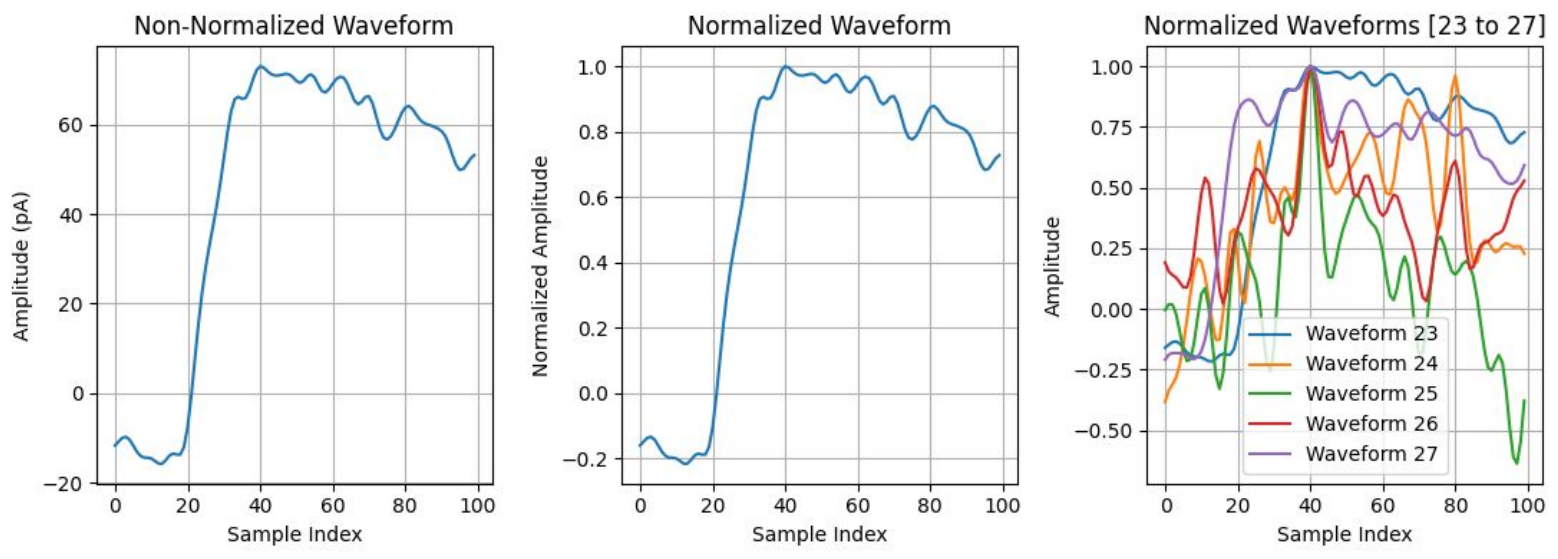
Detect initial peak candidates in the electrophysiology signal based on three simple rules:

- The amplitude must be above a threshold (delta)
- The slope between consecutive points must be steep enough (slope_thresh)
- Detected peaks are at least (distance) samples apart

🔎 Total detected peaks: 332



Pipeline Step 1



Pipeline Step 2

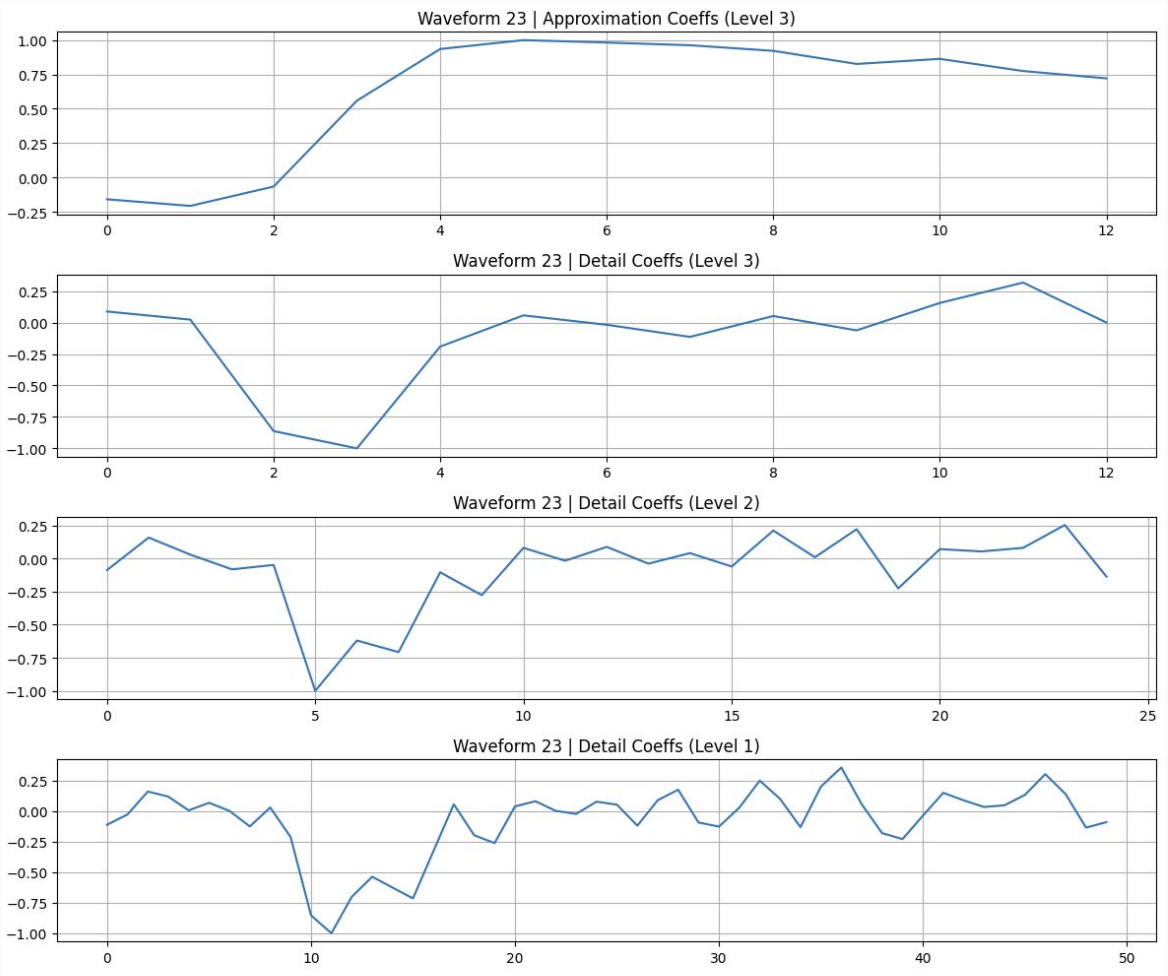
- Wavelet Decomposition

Wavelet decomposition breaks down a signal into progressively lower-frequency components at each level

Level Details:

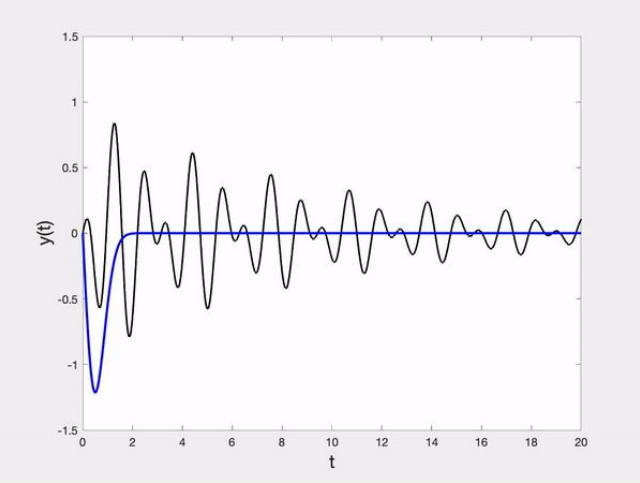
- Level 1: Highest-frequency Detail (Rapid changes (D1)
- Level 2: Next layer down in frequency (smoother than lvl1 (D2)
- Level 3: Even smoother frequency layer
- Level A3: (A3 approximate coefficient), - representing the low-frequency trend or overall shape of the waveform.

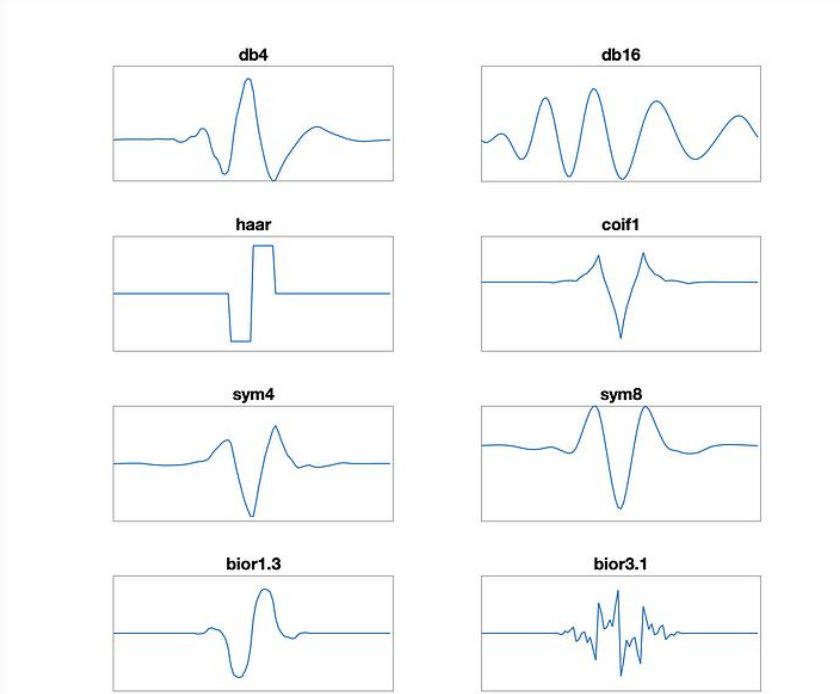
==========



Event Example Decomposed (db4)

Wavelet Transforms



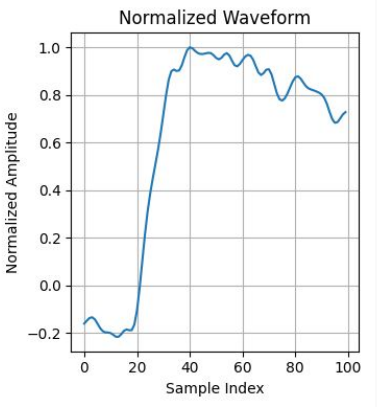


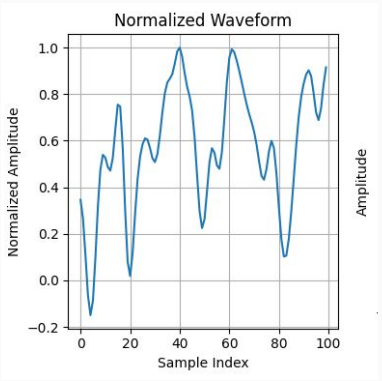
mexican hat function

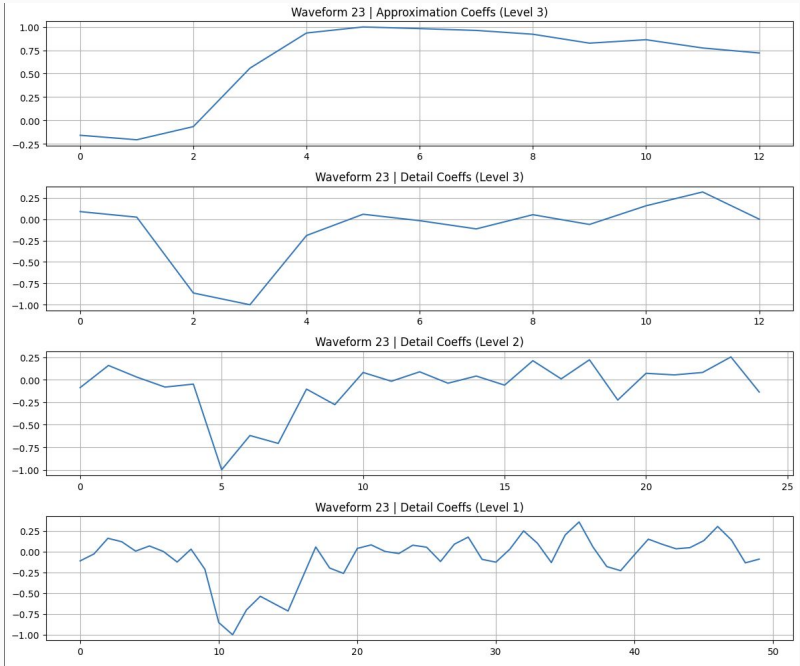
db4

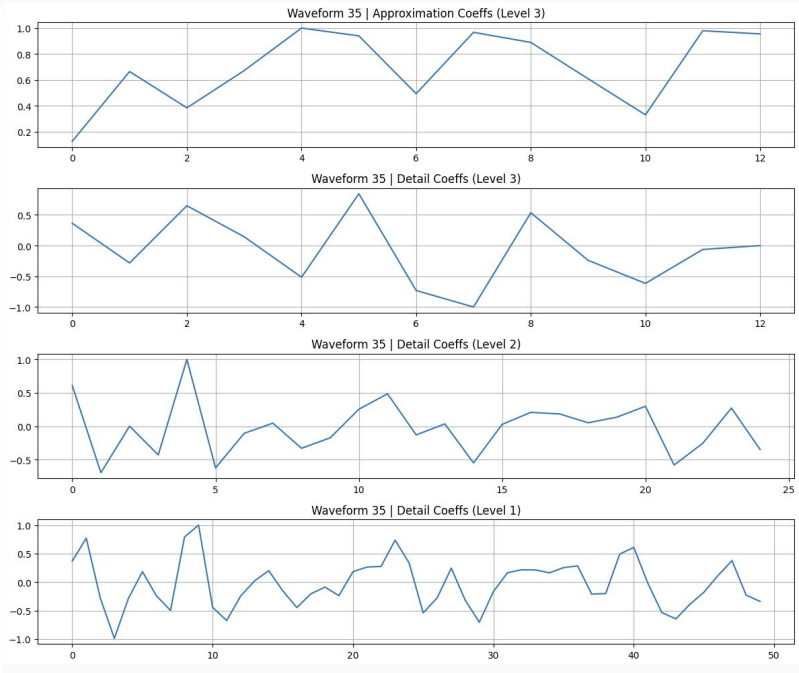
Step 2 Exploration

- Comparing Decomposition

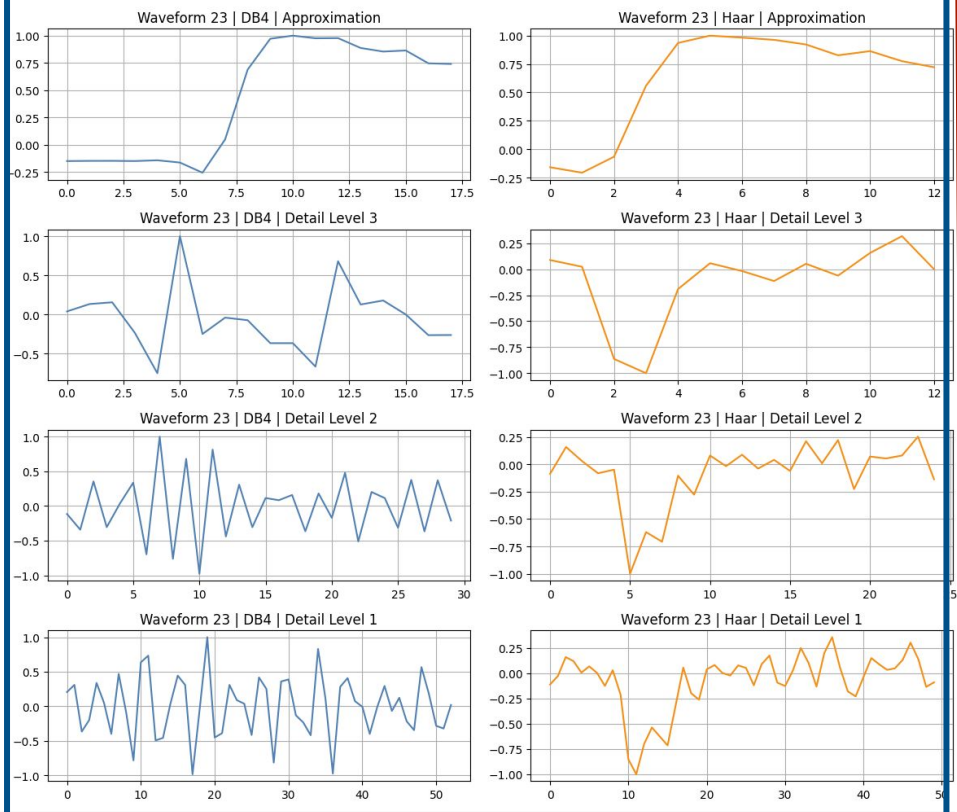




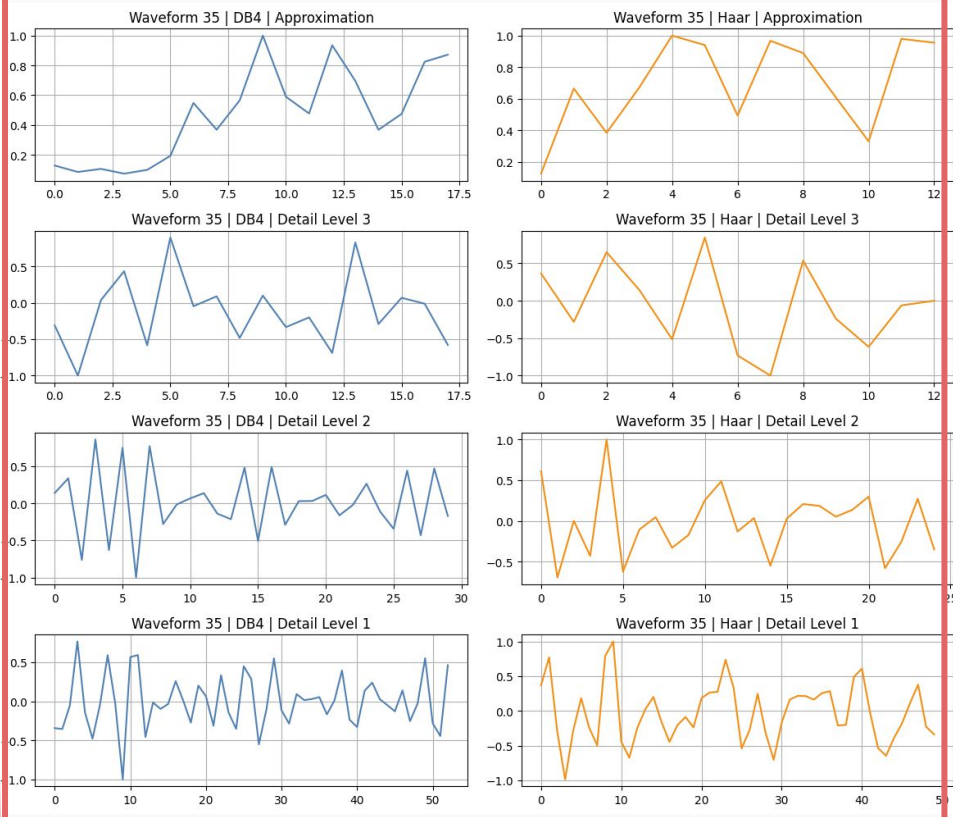




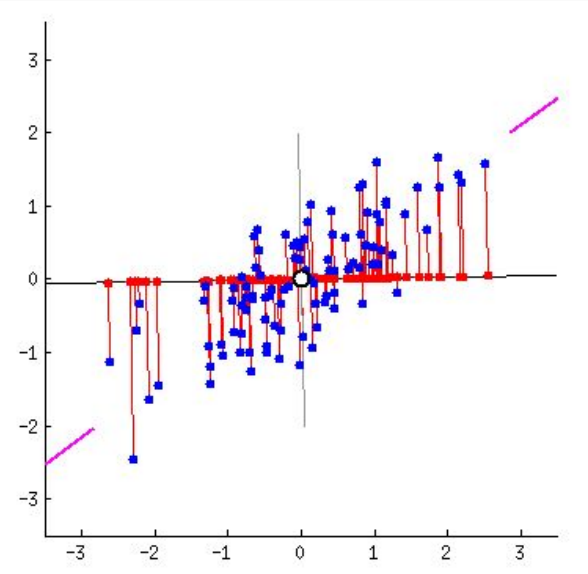
Event



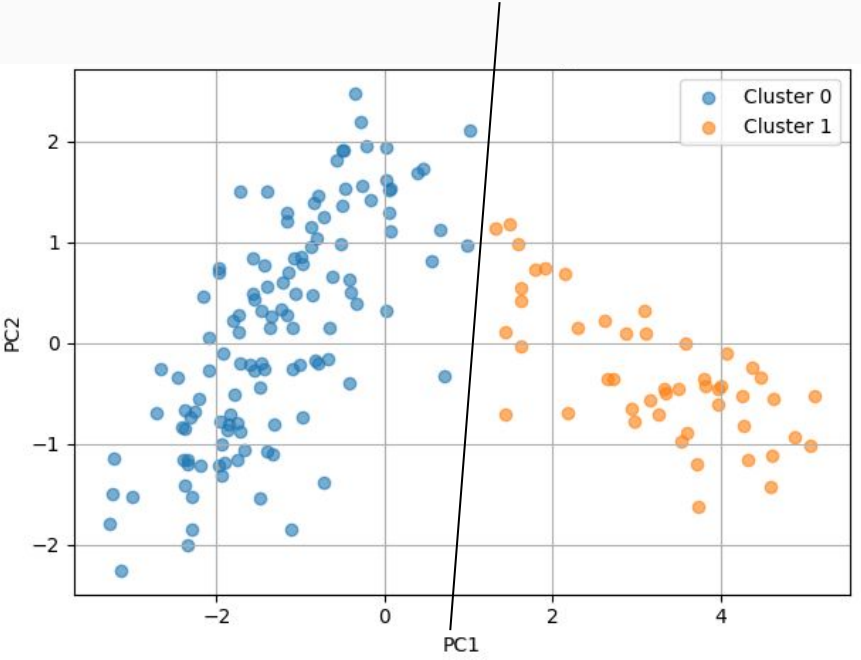
Non-Event

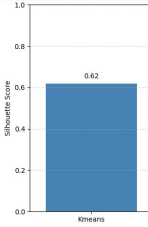


PCA



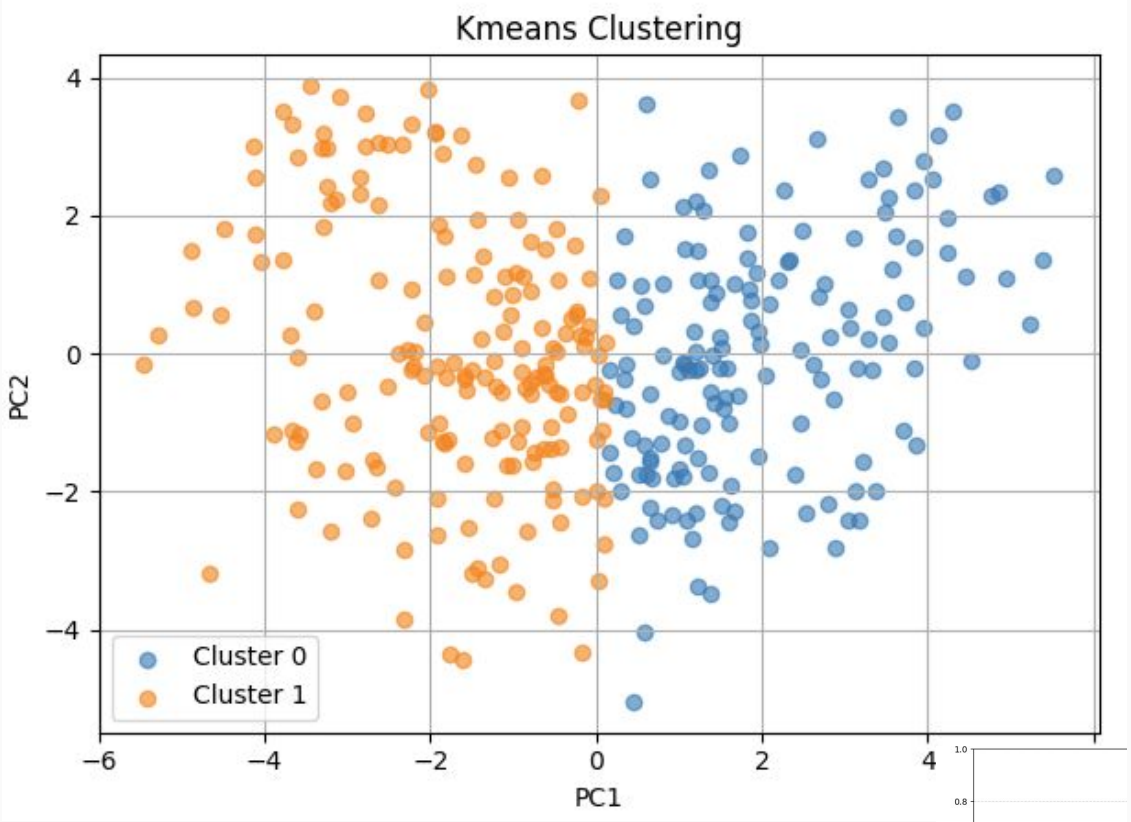
Example

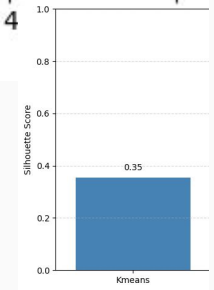




Principal Component Analysis (PCA) Dimensionality Reduction

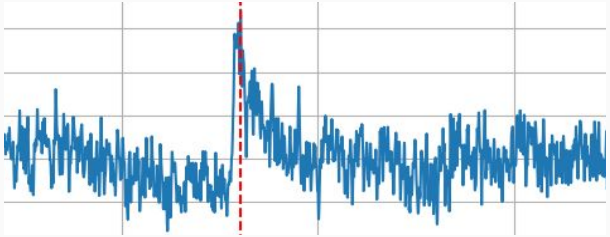
Pipeline Step 3





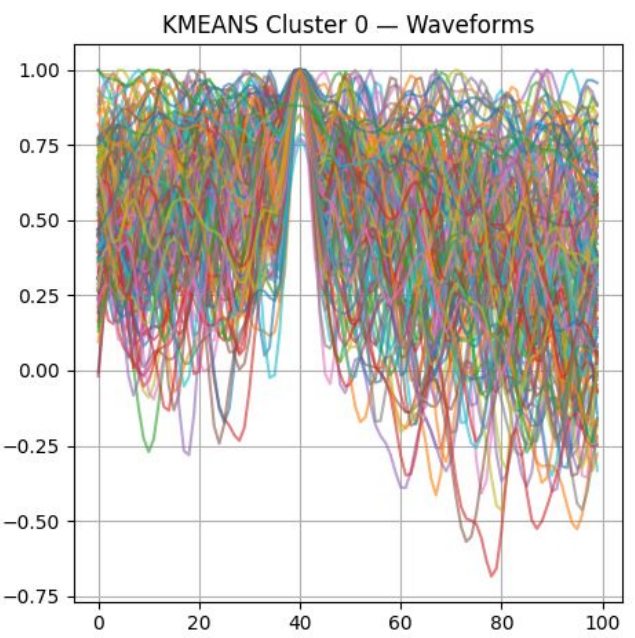
Principal Component Analysis (PCA) Dimensionality Reduction

Inspect Waveforms Per Cluster

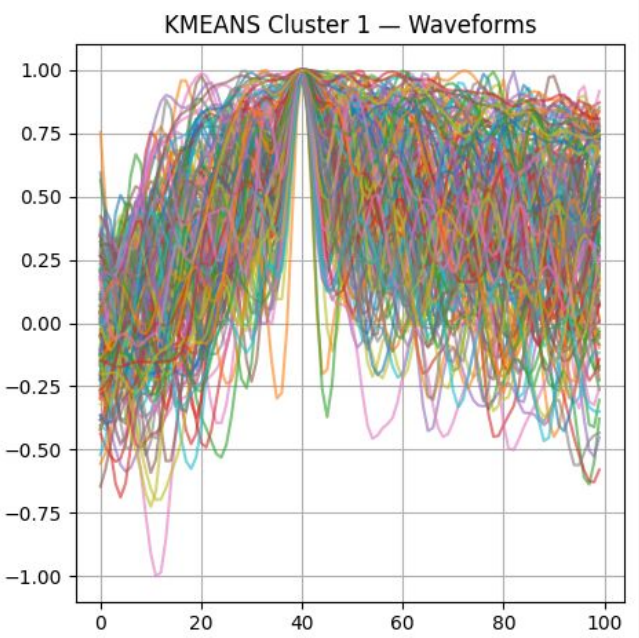


Why does this matter?

- I didn't explicitly tell it any details about what is or is not an event
- It found a potentially important detail in the structure of the data.



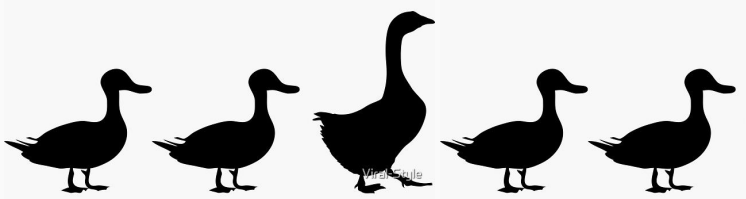
Normalized Waveforms



Structure in the data

Evaluation

Metric Name	Metric Forumla	Code	When to use
Accuracy	$\textbf{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$	`tf.keras.metrics.Accuracy()` or `sklearn.metrics.accuracy_score()`	Default metric for classification problems. Not the best for imbalanced classes.
Precision	$\textbf{Precision} = \frac{tp}{tp + fp}$	`tf.keras.metrics.Precision()` or `sklearn.metrics.precision_score()`	Higher precision leads to less false positives.
Recall	$\textbf{Recall} = \frac{tp}{tp + fn}$	`tf.keras.metrics.Recall()` or `sklearn.metrics.recall_score()`	Higher recall leads to less false negatives.
F1-score	$\textbf{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$	`sklearn.metrics.f1_score()`	Combination of precision and recall, usually a good overall metric for a classification model.
Confusion matrix	NA	Custom function or `sklearn.metrics.confusion_matrix()`	When comparing predictions to truth labels to see where model gets confused. Can be hard to use with large numbers of classes.



Threshold Detection

↓

Pipeline (1,0)

Ground Truth
Matched to Detected

↓

Precision, Recall, F1

High Precision, Low Recall: Very careful. It only calls something a Goose when it's almost certain — so when it says "Goose," it's usually right. But misses a lot of actual Geese.

High Recall, Low Precision: Very eager. It tries to find every Goose and it does. But it also wrongly labels a bunch of Ducks.

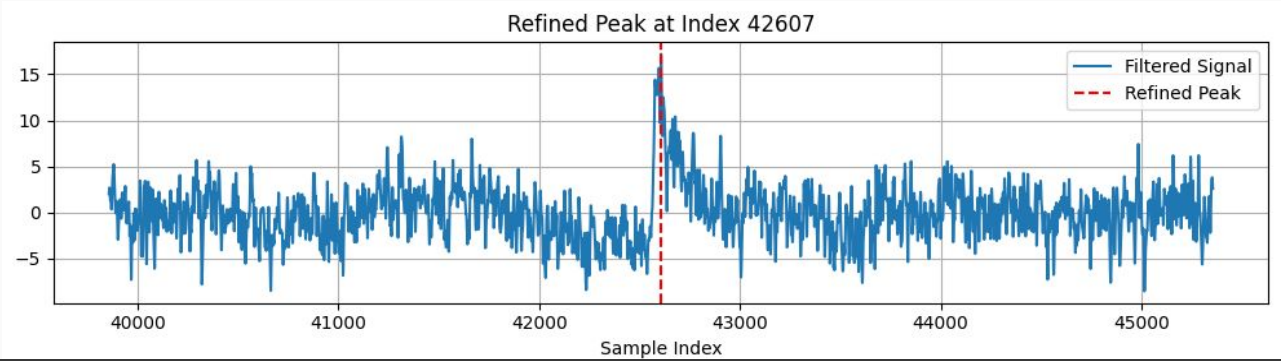
The F1 score is the balance between Precision (how many predicted Geese were actually Geese) and Recall (how many real Geese were correctly found).

Evaluate First Pass Threshold Detector

Point of weakness for the pipeline.

```
📦 Loaded 300 ground truth events
```

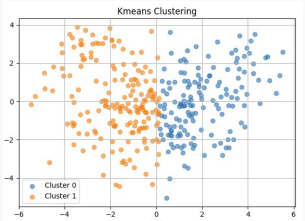
```
Total GT Events: 300 | Total Detected: 332

True Positives: 154
False Positives: 178
False Negatives: 146

Precision: 0.46
Recall: 0.51
F1 Score: 0.49
```



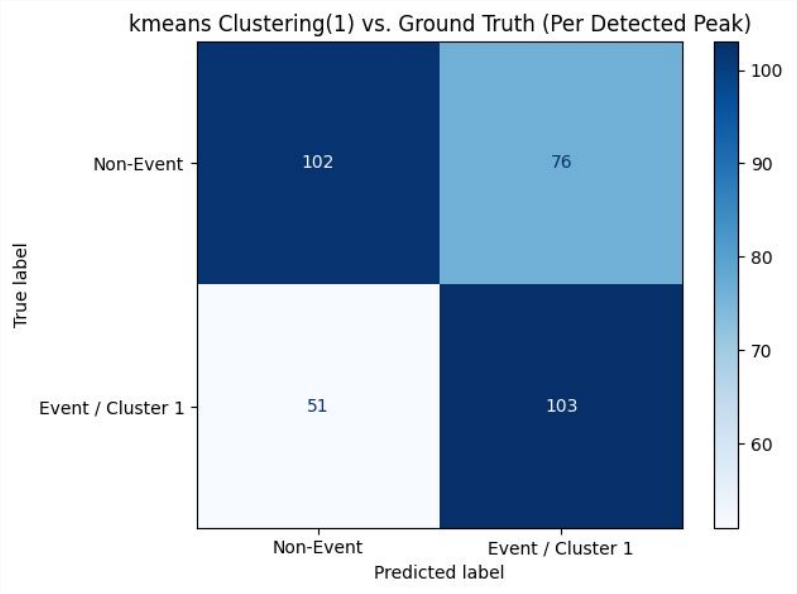

```
↬ Threshold Parameters: {'delta': 13, 'slope_thresh': 0.35, 'distance': 400}
```

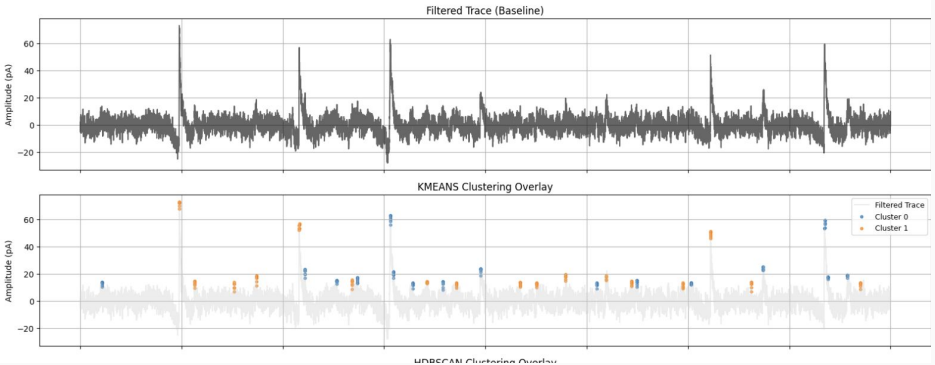
```
🚫 Undetected GT Events: 146
✅ Max # of Cluster True: 154
```

Evaluate Pipeline




```
Clustering Evaluation (Cluster 1 as 'True Event')
Cluster Method: kmeans
```






```
  - True Positives (TP): 103 (TRUE Event)
  - False Positives (FP): 76 (MISCLASSIFIED I.)
  - True Negatives (TN): 102 (TRUE Non-Event)
  - False Negatives (FN): 51 (MISCLASSIFIED II.)
```

Calculate F1 Score (20 Files)

File

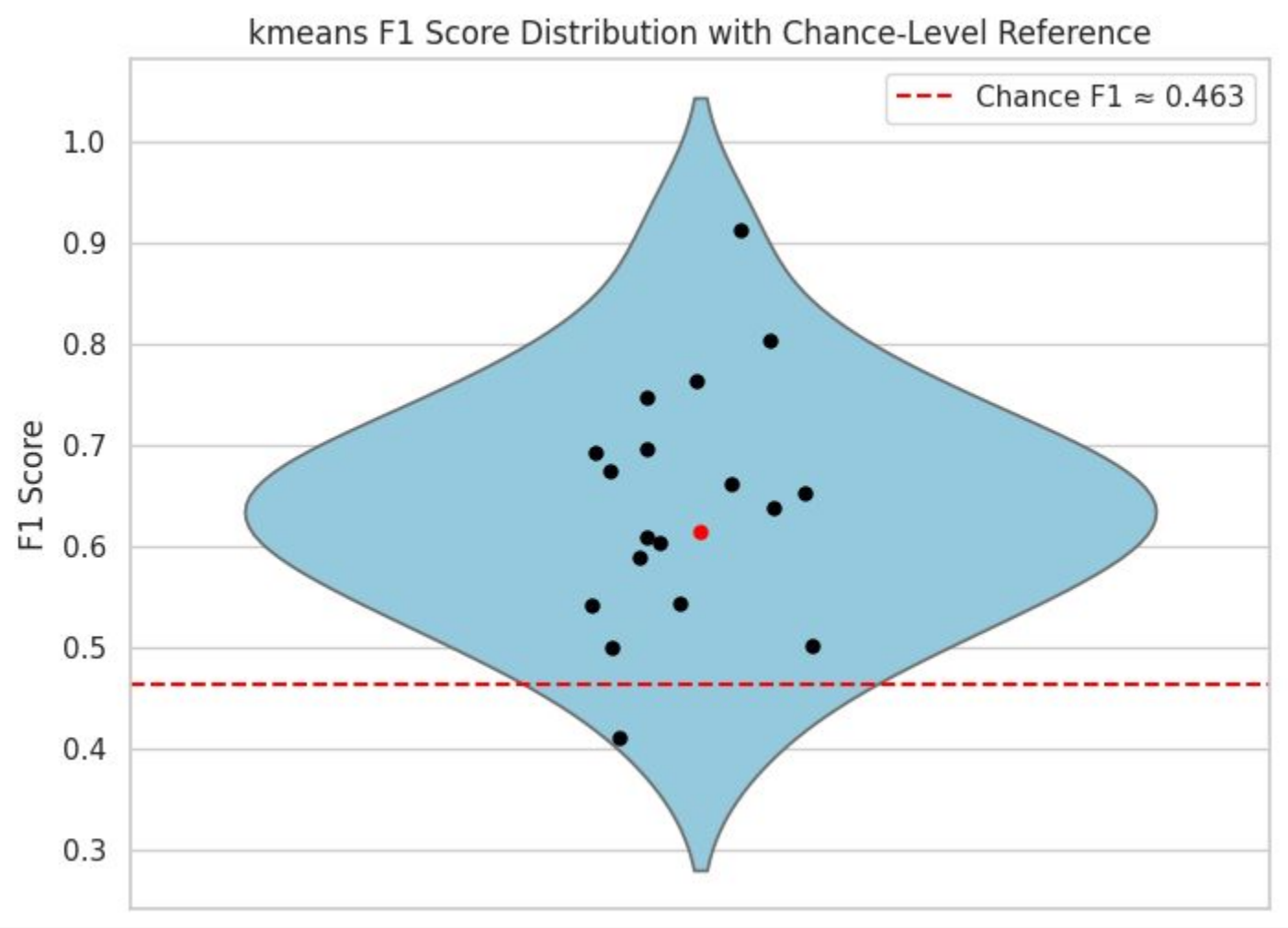
- True Positives (TP): 103 (TRUE Event)
- False Positives (FP): 76 (MISCLASSIFIED)
- True Negatives (TN): 102 (TRUE Non-Event)
- False Negatives (FN): 51 (MISCLASSIFIED)

$$\text{F1 Score} \approx 2 \cdot \frac{0.575 \cdot 0.669}{0.575 + 0.669} \approx 0.618$$

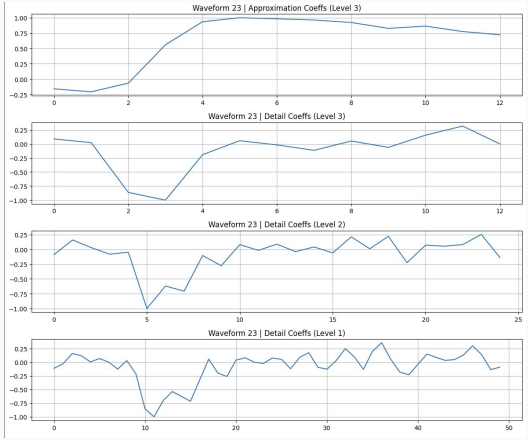
Random Chance

- Prevalence (r) = 154 / 332 = 0.464
- Random Guessing: Precision ~ Recall ~ r

Chance-Level F1 Score ~ 0.464


kmeans F1 Score Distribution with Chance-Level Reference
Chance F1 ≈ 0.463
F1 Score
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3

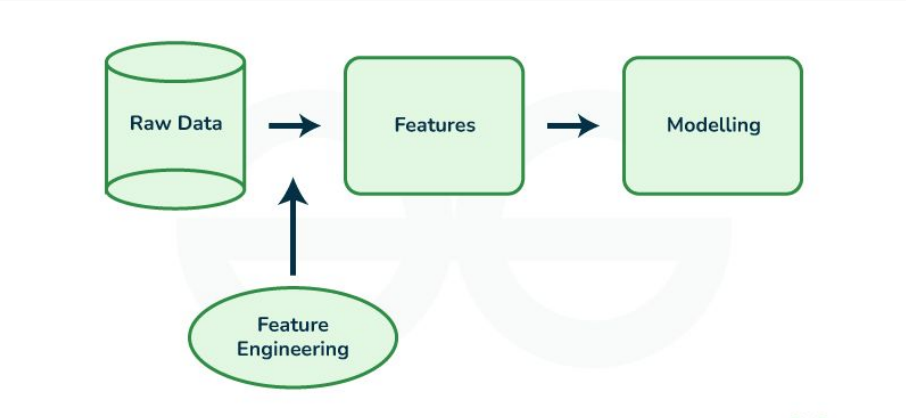
Learn With More Features



Example of some added features

Feature Name	Description
`L{i}_energy`	Total energy of that level: `np.sum(C ** 2)`
`L{i}_std`	Standard deviation: `np.std(C)`
`L{i}_max` / `L{i}_min`	Extremes: `np.max(C)` / `np.min(C)`
`L{i}_zero_crossings`	Structure complexity: `np.sum(np.diff(np.sign(C)) != 0)`
`L{i}_kurtosis`	Tail heaviness: `scipy.stats.kurtosis(C)`

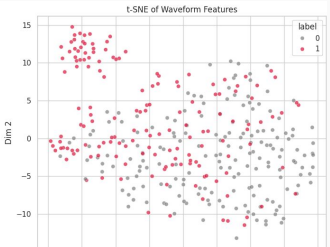
Feature Engineering

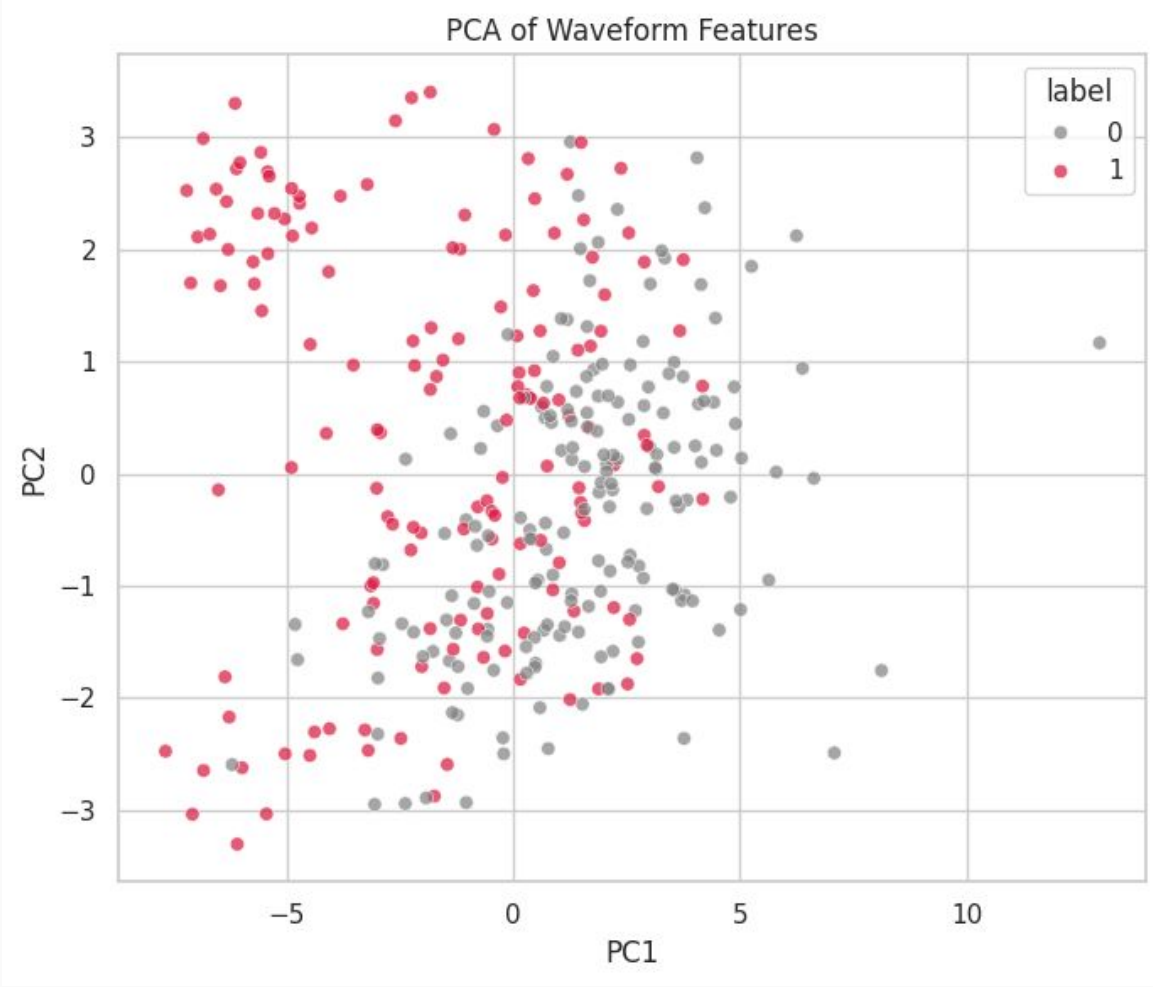


PCA

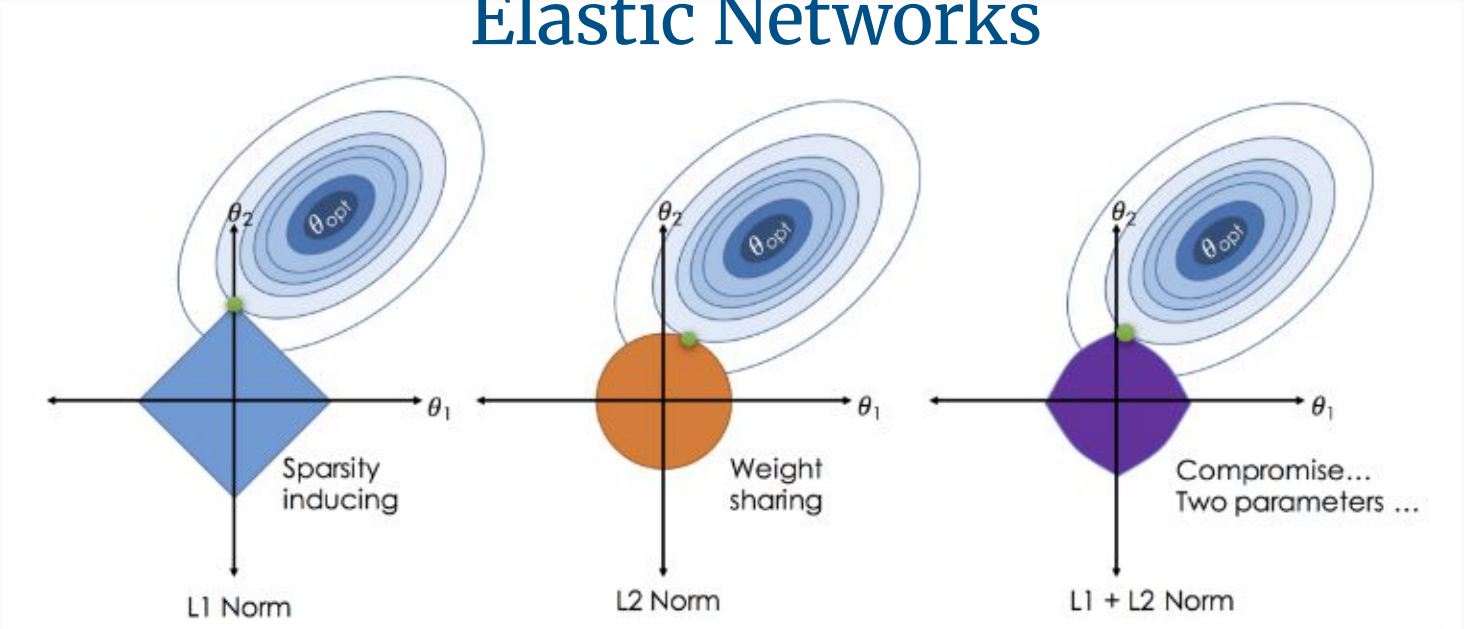
Not Much Separation

- Unstructured Data
- Redundancy in the Data
- Refine by adding only relevant features





Elastic Networks

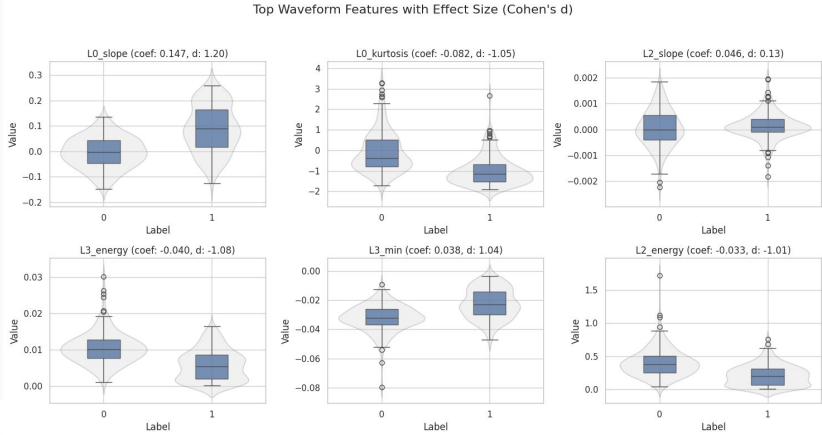


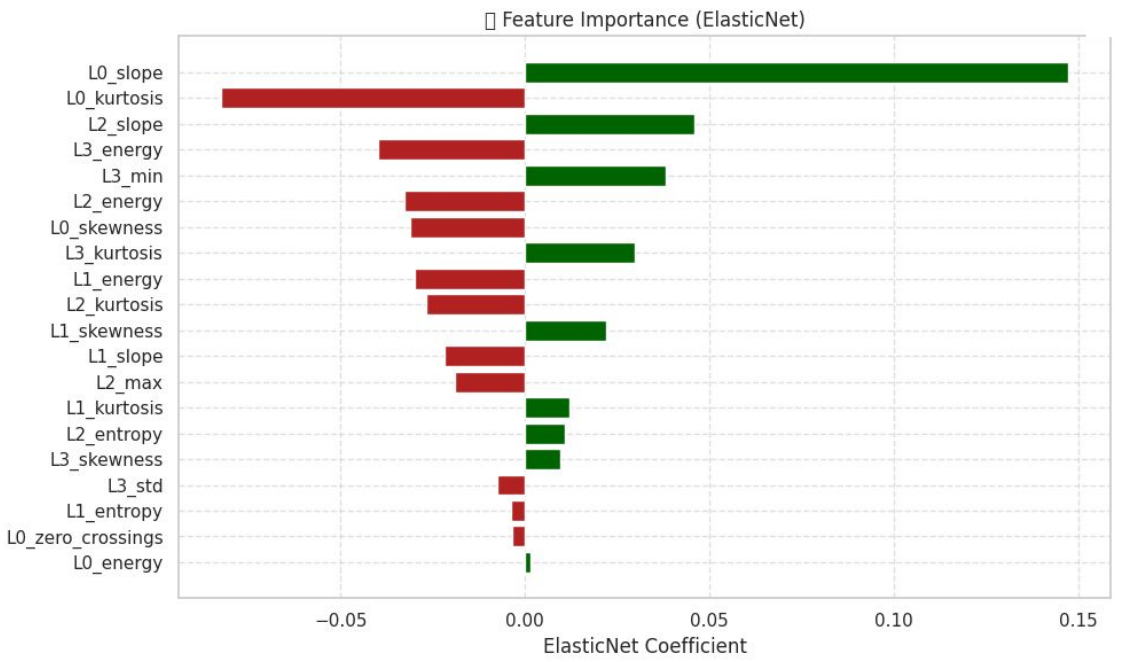
Feature + Elastic Network

1. What new features can **wavelet decomposition** and level **coefficients** give me?
2. Which of these features actually **matter** and to what **extent** for predicting the target? (event or non_event)

Elastic Network

```
L0_energy: 0.001
L0_zero_crossings: -0.003
L0_kurtosis: -0.082
L0_skewness: -0.031
L0_slope: 0.147
L1_energy: -0.029
L1_kurtosis: 0.012
L1_skewness: 0.022
L1_entropy: -0.003
L1_slope: -0.021
L2_energy: -0.032
L2_max: -0.018
L2_kurtosis: -0.026
L2_entropy: 0.010
L2_slope: 0.045
L3_energy: -0.039
L3_std: -0.007
L3_min: 0.038
L3_kurtosis: 0.029
L3_skewness: 0.009
```

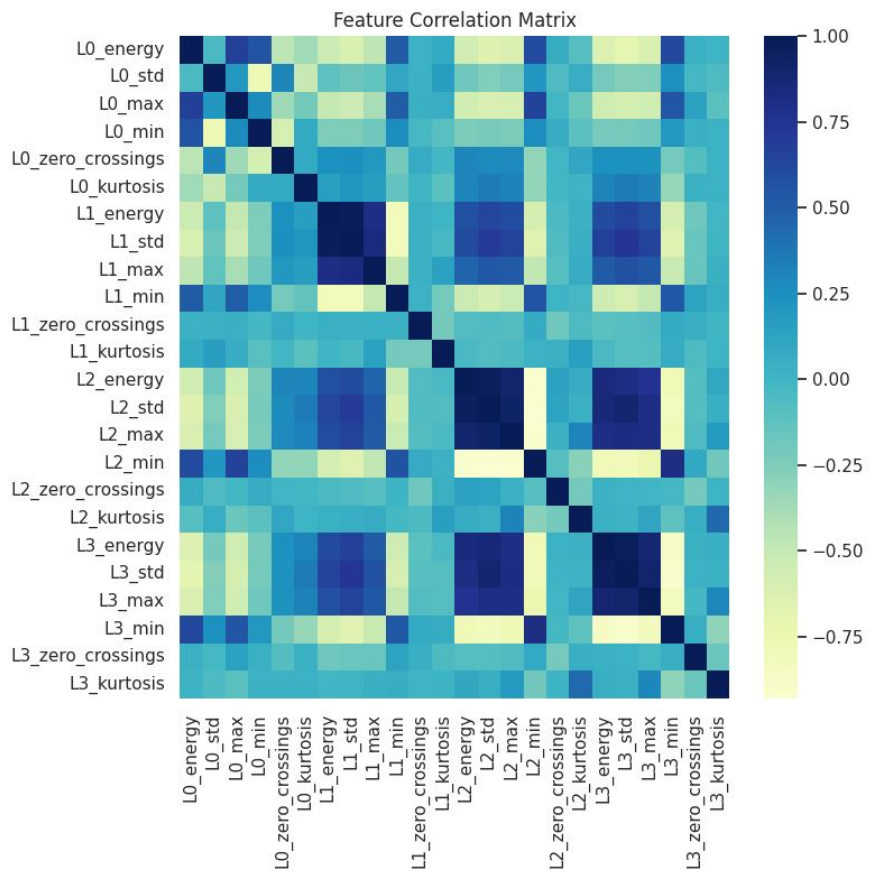


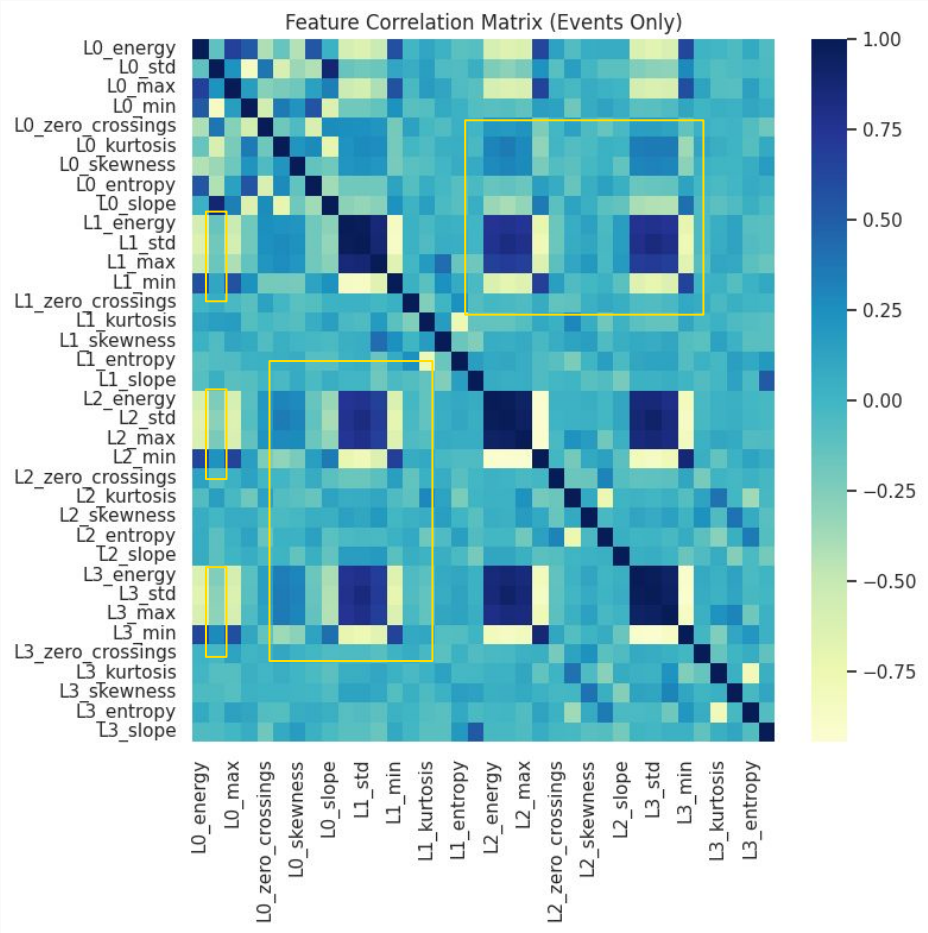


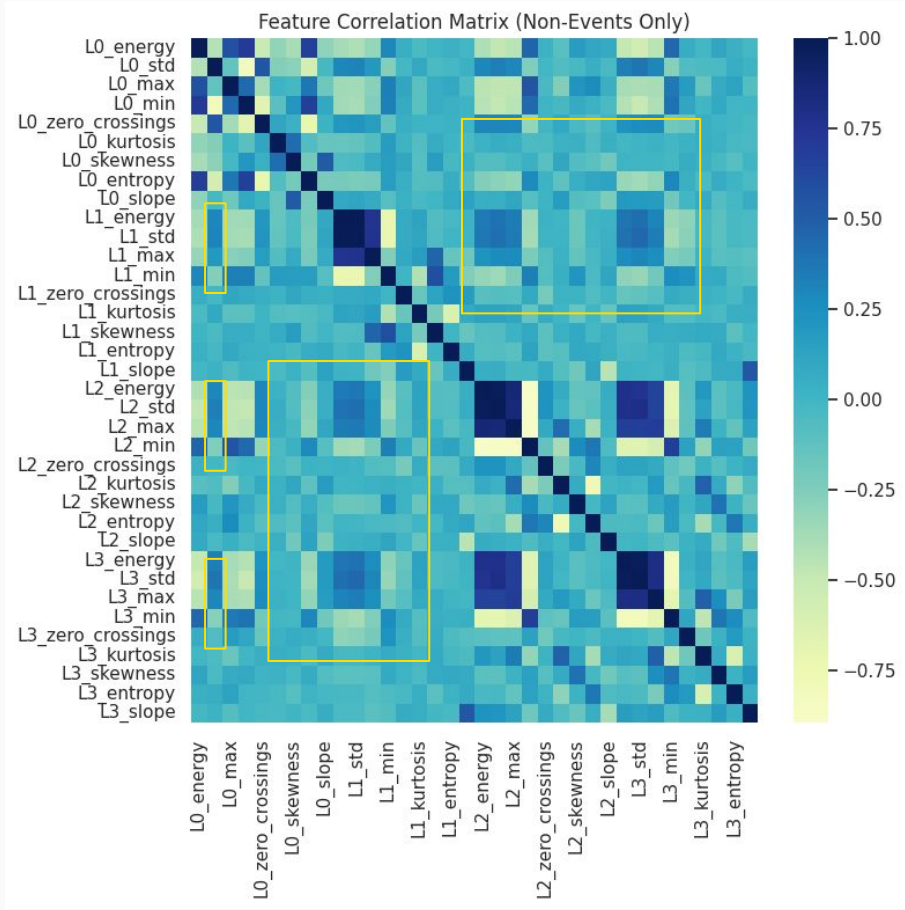
- L0_Slope:
  - Best single discriminator.
- L0_Kurtosis:
  - Highly Informative.
- L3_energy:
  - Powerful single feature. Possibly encodes for event time sharpness.
- L3_min:
  - Good signal strength. Possibly encodes negative event deflection depths.
- L2_energy
  - Meaningful drop in energy. Possibly flatter waveforms.

Collinearity Among Features

Events Vs Non-Events



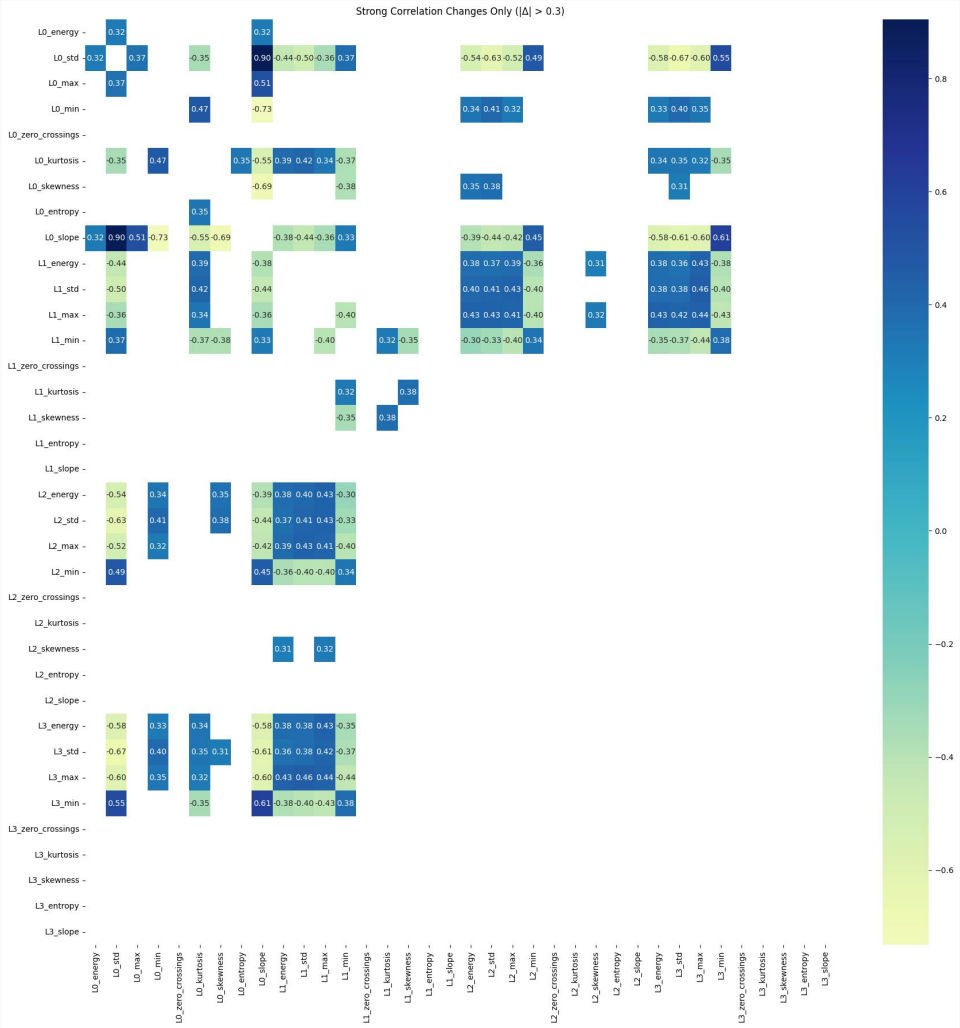




If a pair of features is highly correlated in **events** but **not in non-events**, then their **relationship** (not just values) could help classification.

This could justify exploring **feature interactions or engineered ratios** (e.g., `L0_slope / L0_energy`), especially within strongly correlated event clusters.

Features interact differently during events vs. noise.

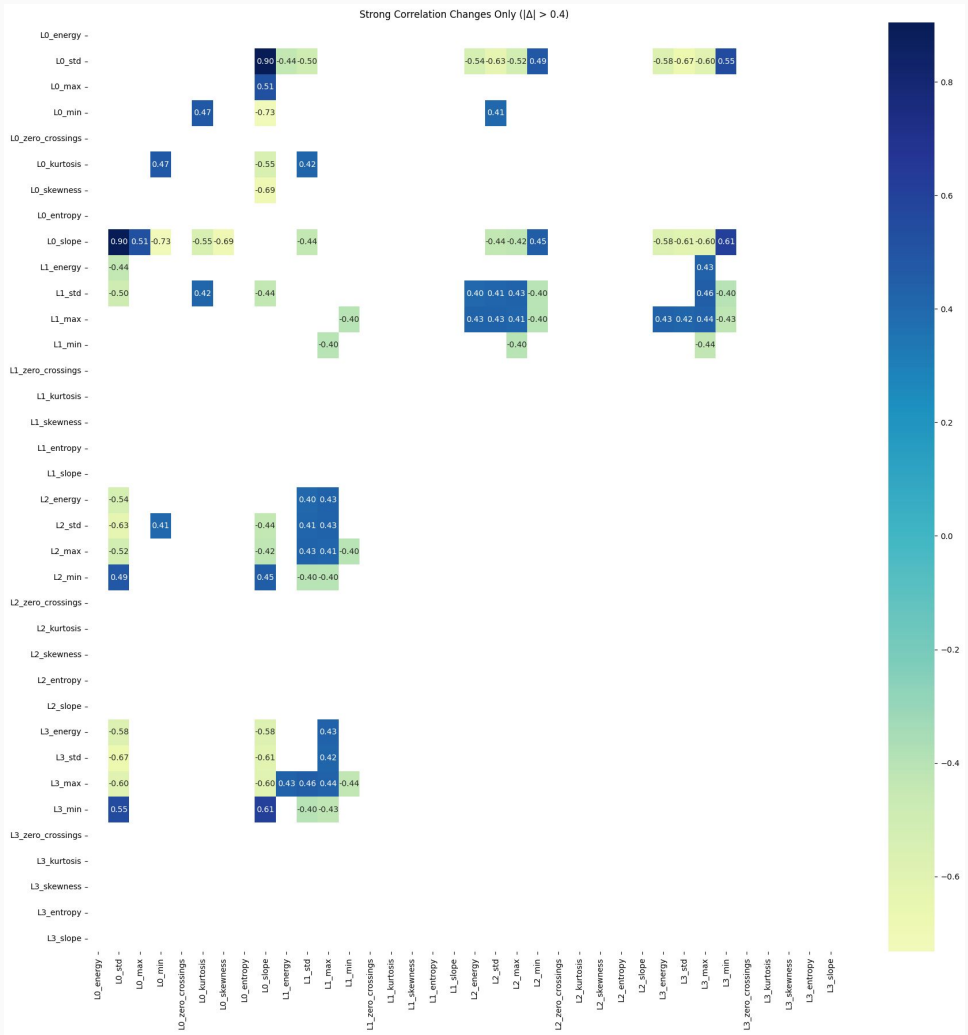


Features interact differently during events vs. noise.

<u>Colinearity within Detail Levels:</u>
L0_slope and **L0_energy** might carry overlapping or interacting information?

<u>Feature Interaction within Level:</u>
L3_min may act as a polarity/shape-sensitive component, while **L3_energy** quantifies magnitude.

<u>Across Detail Levels:</u>
fast slope indicator (**L0_slope**) with variability in a slightly slower wavelet band (**L1_std**).

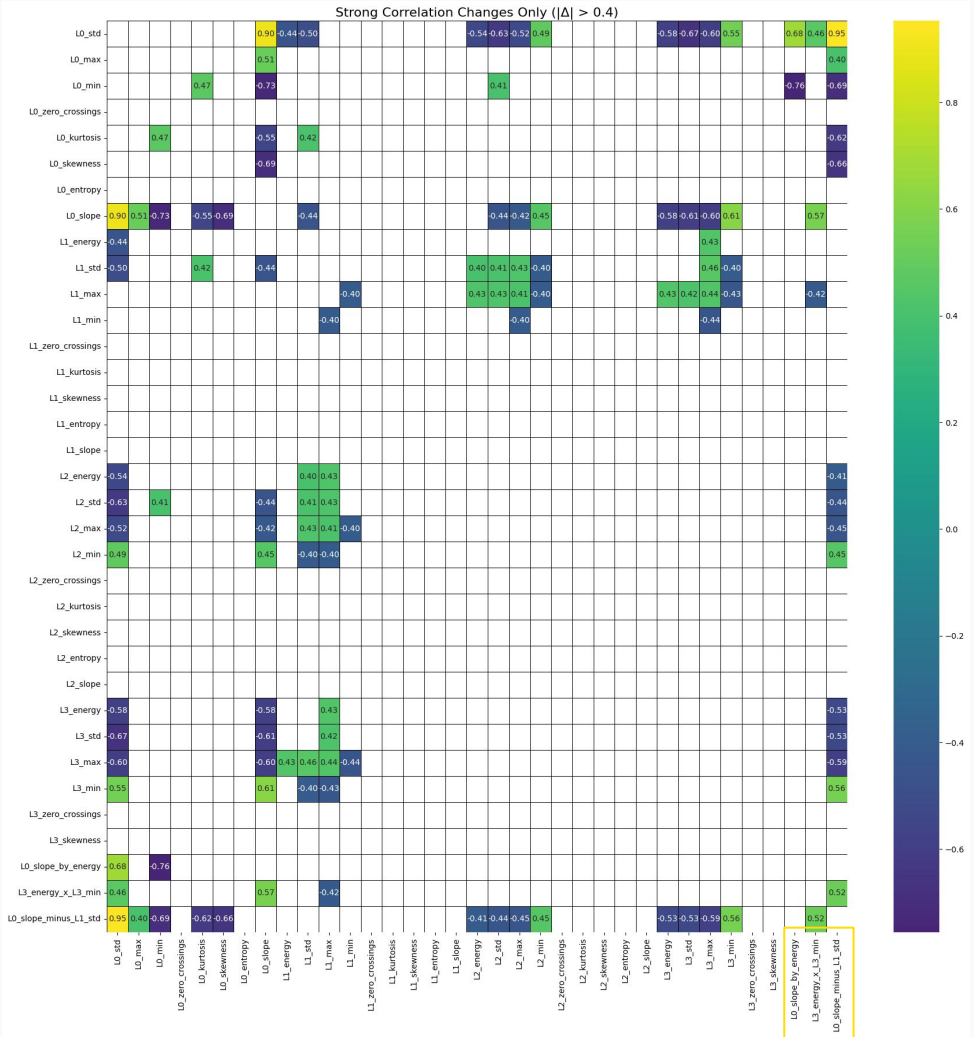


Feature Engineering

L0_slope_by_energy

L3_energy_x_L3_min

L0_slope_minus_L1_std



Feature Importance

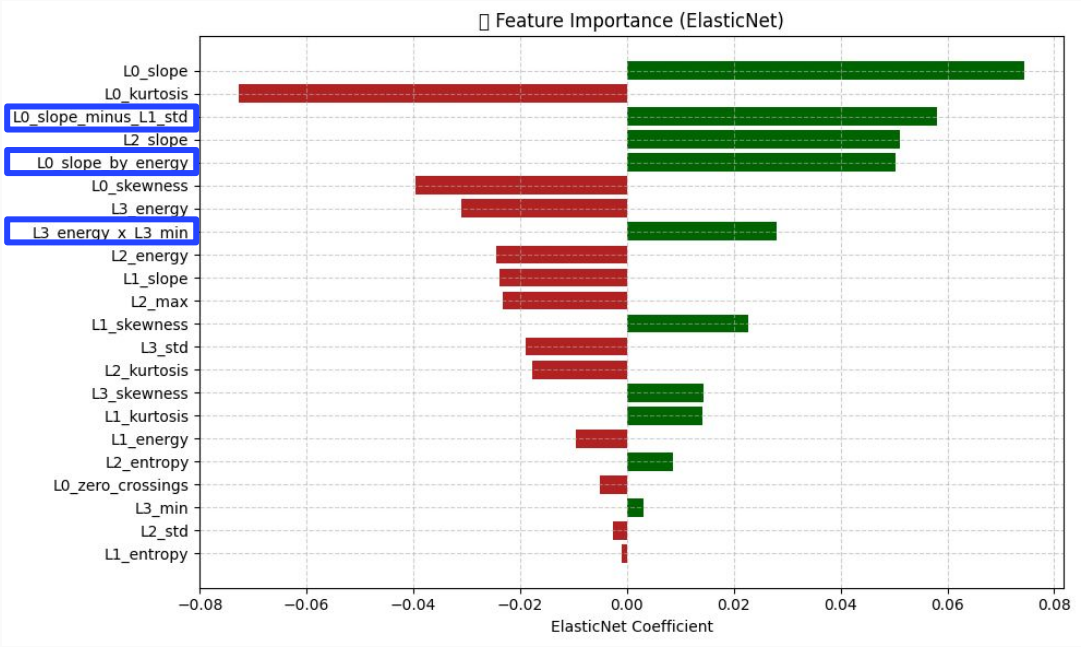
L0_zero_crossings: -0.0052
L0_kurtosis: -0.0727
L0_skewness: -0.0396
L0_slope: 0.0744

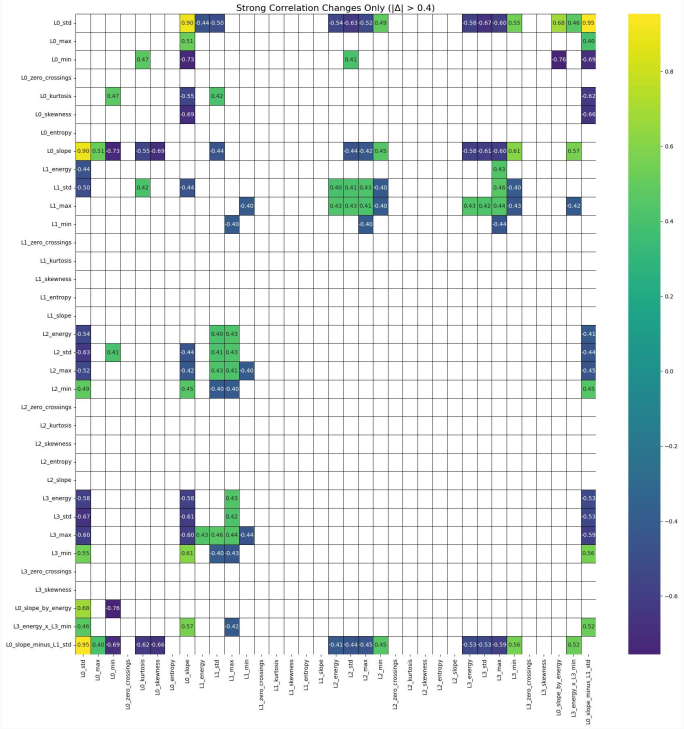
L1_energy: -0.0096
L1_kurtosis: 0.0142
L1_skewness: 0.0227
L1_slope: -0.0239

L2_energy: -0.0246
L2_std: -0.0025
L2_max: -0.0233
L2_kurtosis: -0.0178
L2_entropy: 0.0087
L2_slope: 0.0511

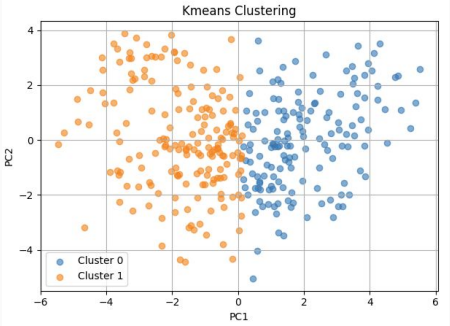
L3_energy: -0.0310
L3_std: -0.0189
L3_min: 0.0032
L3_skewness: 0.0144

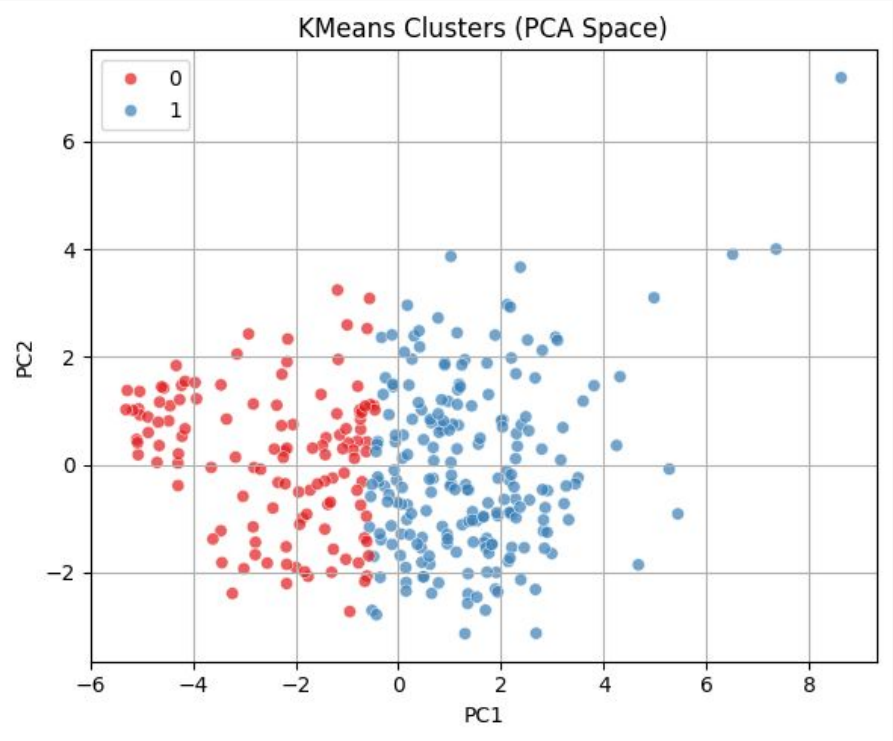
L0_slope_by_energy: 0.0503
L3_energy_x_L3_min: 0.0281
L0_slope_minus_L1_std: 0.0580

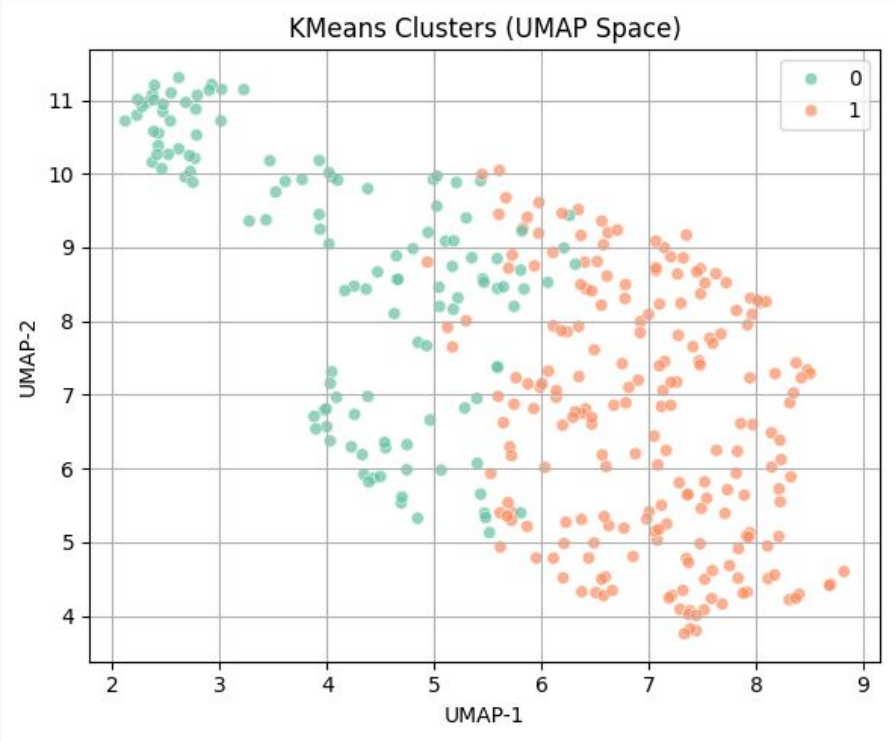




PCA separation might have benefitted from the data-approach

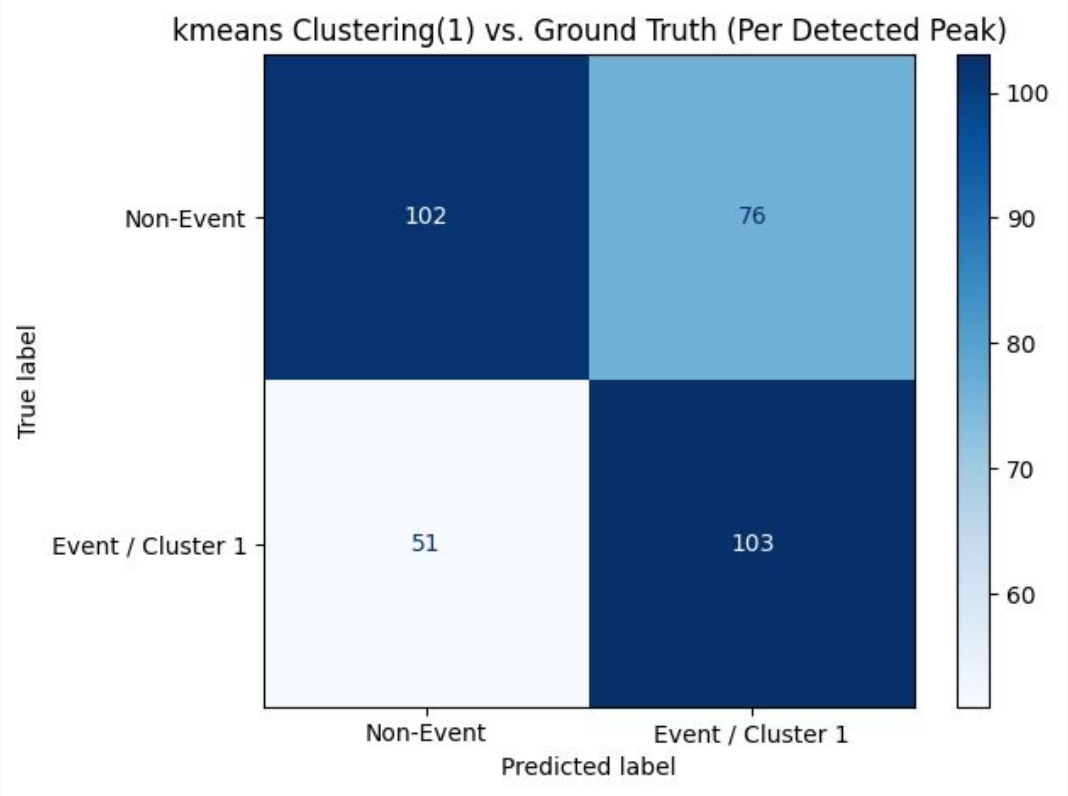






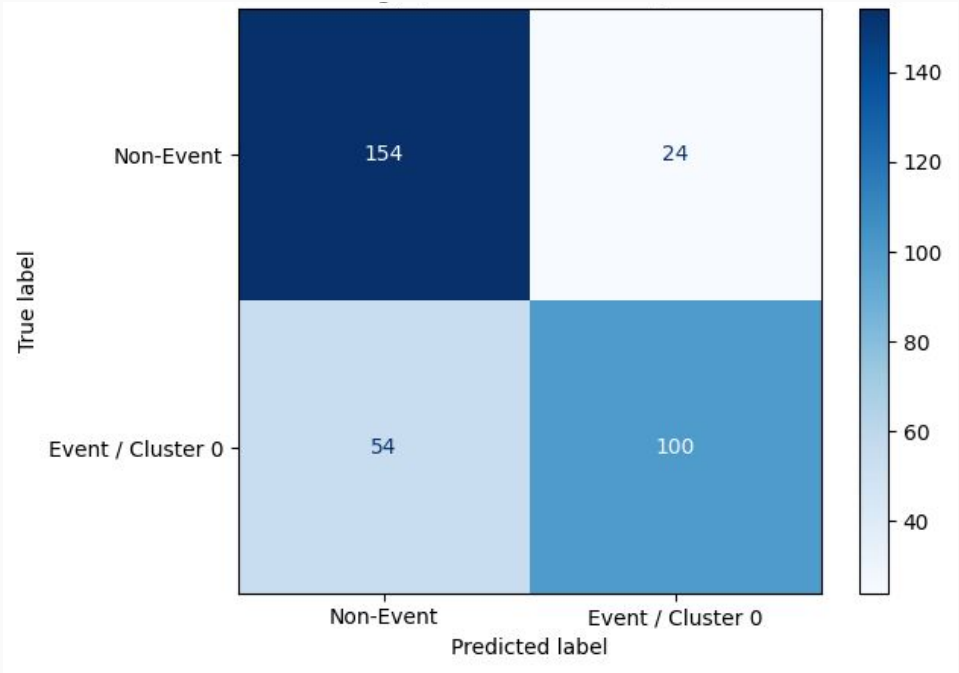
Goal:
Data → Outperform Initial Score

```
📊 Clustering Evaluation (Cluster 1 as 'True Event')
🌌 Cluster Method: kmeans

• Precision: 0.575
• Recall:    0.669
• F1 Score:  0.619

Confusion Matrix Breakdown:
  - True Positives (TP): 103
  - False Positives (FP): 76
  - True Negatives (TN): 102
  - False Negatives (FN): 51

Summary:
  ✅ Matched GT Events: 103
  🚫 Missed GT Events: 51
```



Feature Selection + Kmeans Clustering (Unsupervised)

```
🔧 Running: KMeans
==================================================
Clustering Evaluation (Cluster 0 as 'True Event')

• Precision: 0.806
• Recall:    0.649
• F1 Score:  0.719

Summary:
   ✅ Matched GT Events: 100
   🚫 Missed GT Events: 54
```



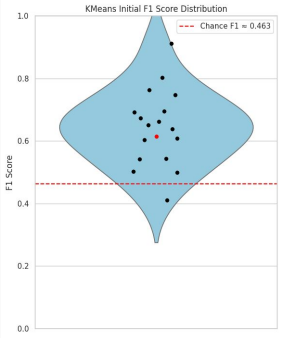
18 files later...

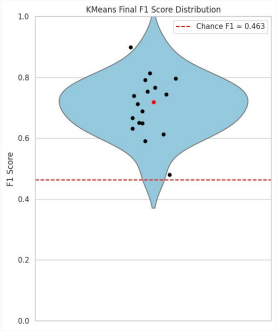
New Kmeans
f1_scores = [
0.744, 0.719, 0.614,0.480, 0.632,*0.9*, 0.74, 0.792,
0.689, 0.754, 0.713, _, 0.667, 0.591, 0.649, 0.651,
0.814, 0.767, 0.797
]

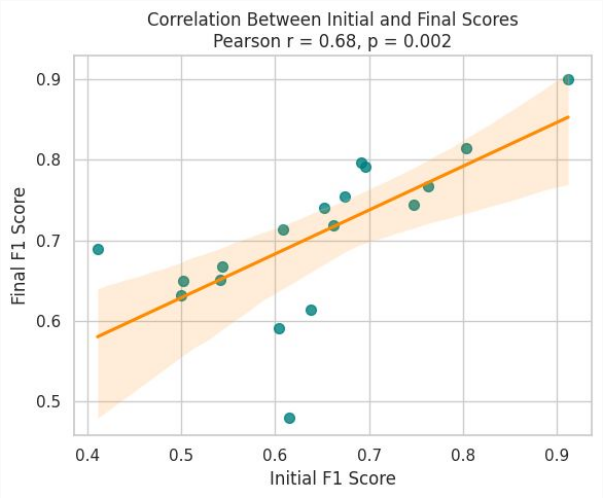
random_forest
f1_scores = [
0.883, 0.682, 0.93, 0.5, 0.560,*0.93*, 0.844, 0.667,
0.667, 0.741, 0.803, _, 0.769, 0.688, 0.567, 0.727,
0.952, 0.887, 0.827
]

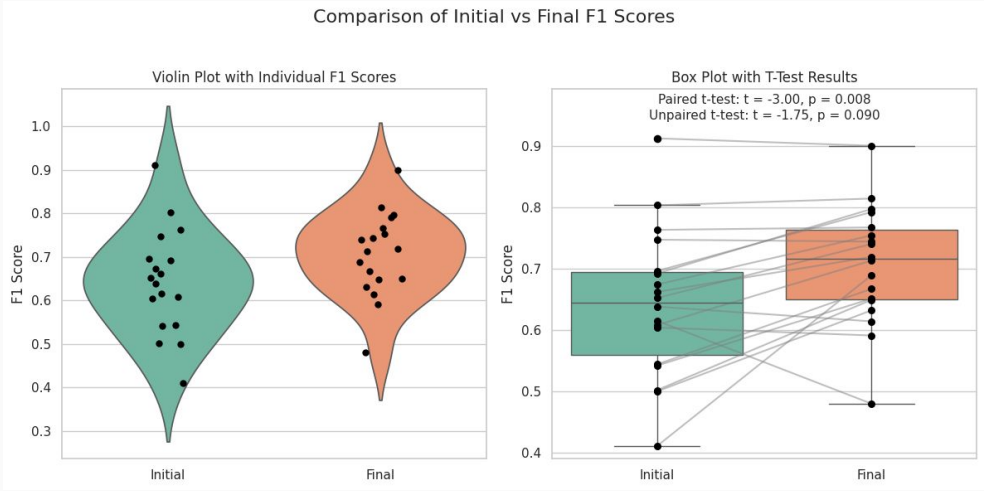
SVM
f1_scores = [
0.857, 0.742, 0.952, 0.588, 0.551, *0.889*, 0.830, 0.696,
0.667, 0.786, 0.820, _, 0.774, 0.690, 0.578, 0.734,
0.943, 0.927, 0.832
]

Increased Each F1 Score





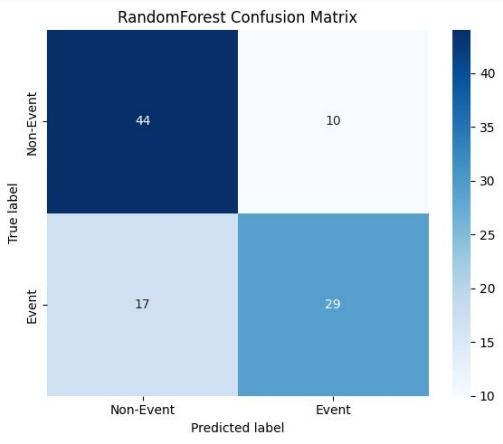


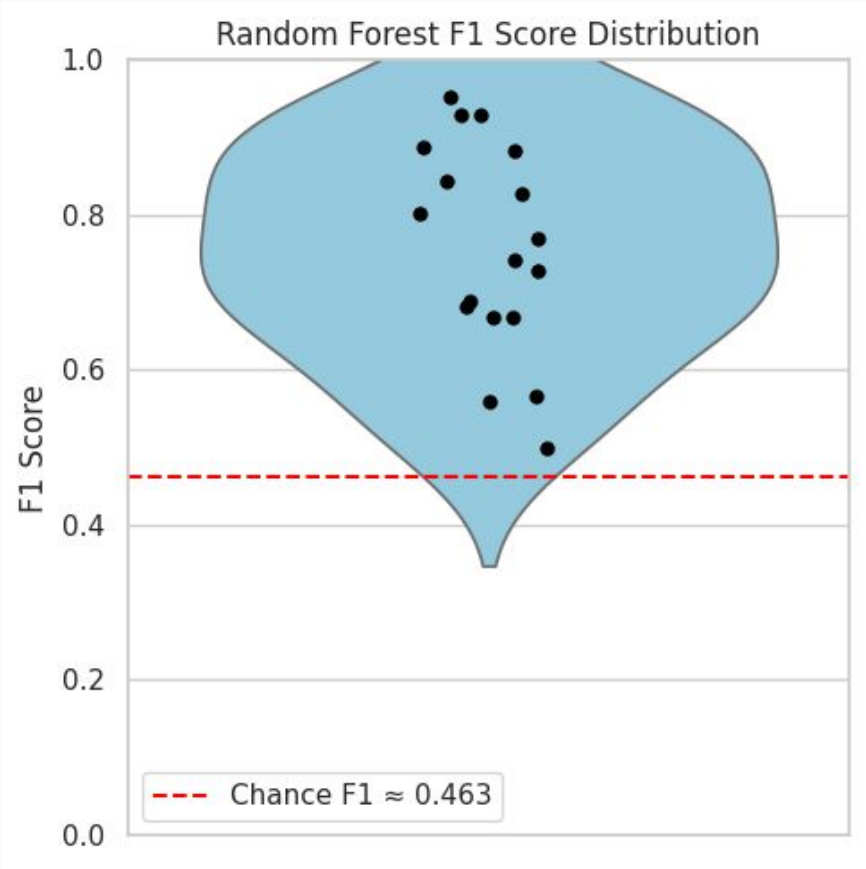


Random Forest Classifier (Supervised)

🔧 Running: RandomForest
📊 Classification Report:

	precision	recall	f1-score
0	0.721	0.815	0.765
1	0.744	0.630	0.682
accuracy			0.730
macro avg	0.732	0.723	0.724
weighted avg	0.732	0.730	0.727

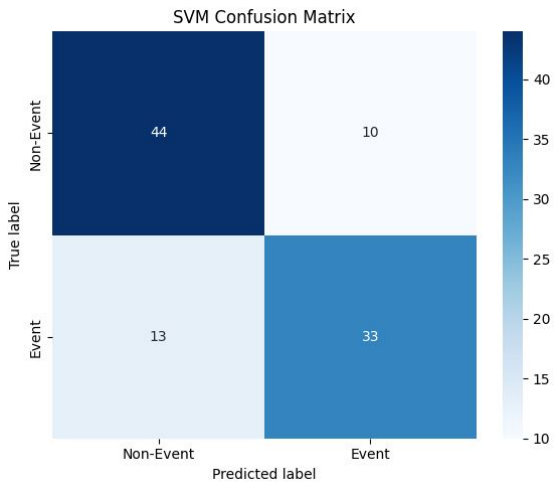


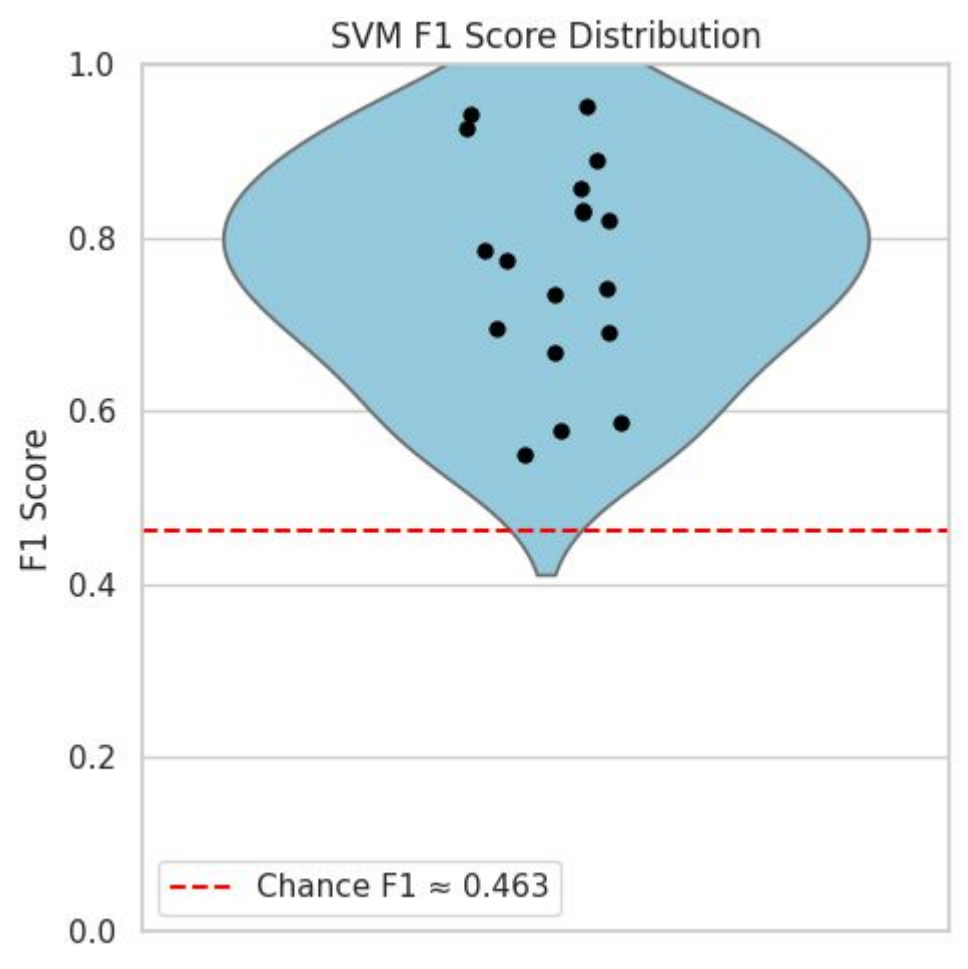


SVM Classifier (Supervised)

🔧 Running: SVM
📊 Classification Report:

	precision	recall	f1-score
0	0.772	0.815	0.793
1	0.767	0.717	0.742
accuracy			0.770
macro avg	0.770	0.766	0.767
weighted avg	0.770	0.770	0.769





Conclusions

Feature engineering **expanded** the set of inputs available for the machine learning model, providing more informative data. Elastic Net regression helped identify the most **impactful** features, which led to a measurable improvement in F1 score.

By combining Elastic Net coefficients with visual inspection of correlation heatmaps, I was able to identify and remove **redundant or collinear** features. These insights also inspired the creation of new, biologically interpretable features such as L0_slope_by_energy, L3_energy_x_L3_min, and L0_slope_minus_L1_std.

After feature engineering:

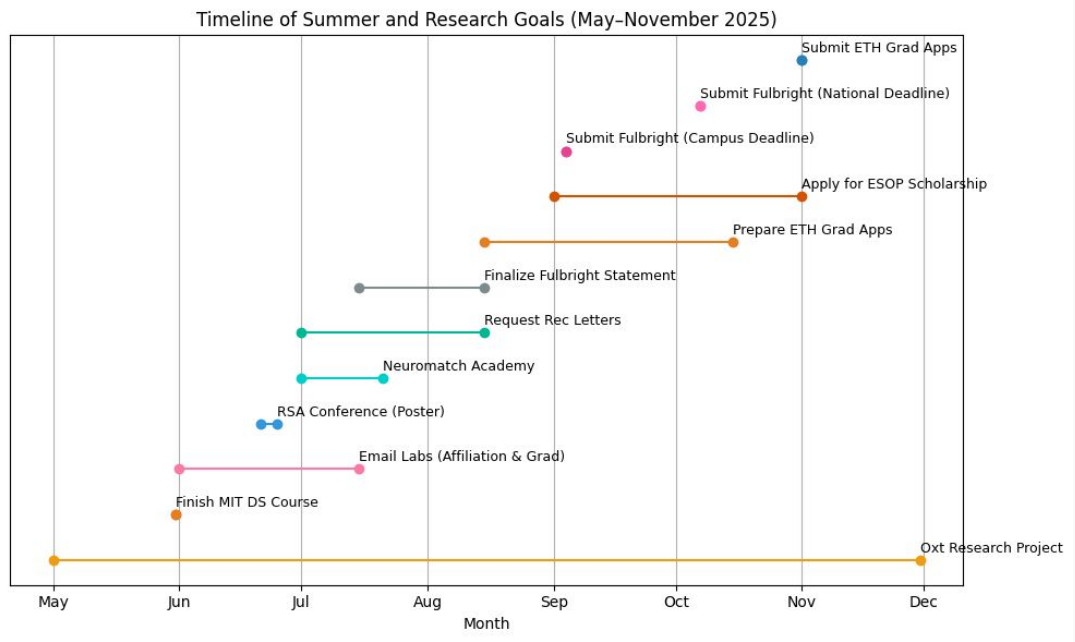
- **F1 Score improved** from **0.619** to **0.719**
- **Precision increased significantly**, reducing false positives
- **Recall remained stable**, maintaining sensitivity to true events

This improvement is valuable in the context of synaptic event detection, where high precision reduces the chance of falsely identifying noise as signal. Analysis across 18 files showed that even a simple algorithm like KMeans benefitted from this data-driven refinement, suggesting the approach adds value. While broader generalizability remains to be tested, early results are promising.

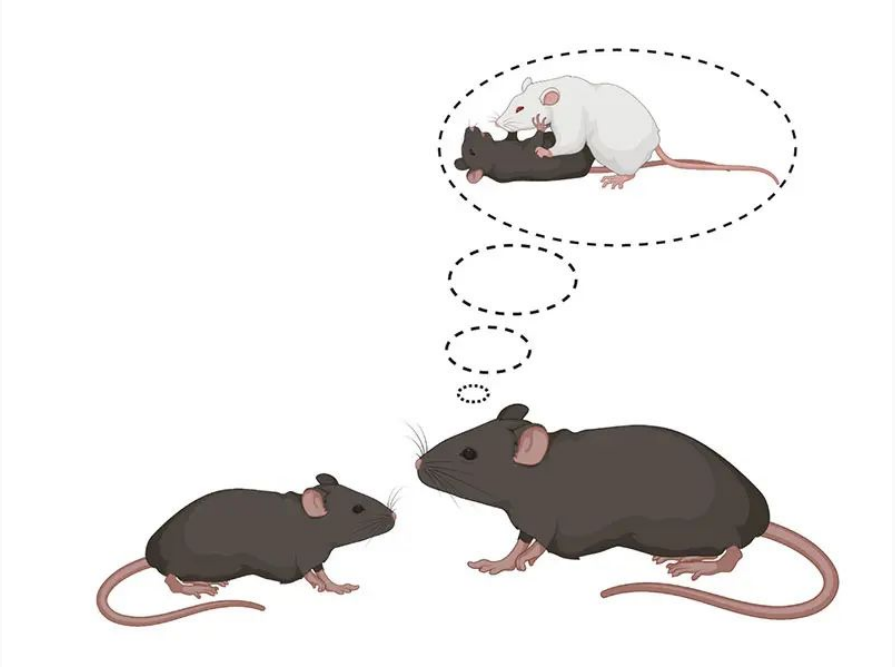
Next Steps for This Side Project

- Select features that generalize well across multiple datasets
- Explore other unsupervised methods (e.g., Random Forest, neural networks, miniML CNN reproduction)
- Extend to other types of signal datasets such as photometry data (e.g., as a Neuromatch project)

Personal Future Aims and Timeline



Social Defeat



Future projects will involve mice on mice violence.

Alcohol Drinking Data

Susceptible vs Resilient

Thanks Sparta Lab!