Faculty of Engineering and Technology
Computer Science Department

Artificial intelligence
ENCS 3340

Project 2 report
Machine Learning for Classification

Prepared by:
Abdel Rahman Shahen 1211753
Mahmoud Awad 1212677

Instructor:
Dr. Aziz Qaroush

Section: 3
Date: 6/20/2024

Speaker Accent Recognition Dataset:

The Speaker Accent Recognition Dataset contains recordings of speech samples from different speakers, labeled by their respective languages. Each instance in the dataset represents a set of extracted features from a speech sample, and the task is to classify these samples based on the language of the speaker.

❖ **Attributes:**

- **The parameters ($x1$, $x2$, …, $x12$):** that describe sounds typically refer to the 12 Mel-frequency cepstral coefficients (MFCCs). These coefficients are widely used in speech and audio processing to capture the short-term power spectrum of sound. Here is a brief description of each parameter:
  1. **$x1$(MFCC1):** Represents the overall energy in the signal, often associated with the loudness or volume of the sound.
  2. **$x2$(MFCC2):** Captures the slope of the spectrum, related to the timbre or quality of the sound.
  3. **$x3$(MFCC3):** Reflects the shape of the spectrum, providing information about the formant frequencies.
  4. **$x4$(MFCC4):** Further refines the spectral shape and captures additional details about the formants.
  5. **$x5$(MFCC5):** Continues to describe the spectral envelope, refining the characterization of the sound.
  6. **$x6$(MFCC6):** Adds more detail to the spectral shape, focusing on higher-order characteristics.
  7. **$x7$(MFCC7):** Captures finer details of the spectral envelope, often associated with subtle changes in timbre.
  8. **$x8$(MFCC8):** Further refines the spectral shape, focusing on higher frequencies.
  9. **$x9$(MFCC9):** Adds more detail to the higher frequency components of the spectrum.
  10. **$x10$(MFCC10):** Continues to describe the spectral envelope with a focus on higher frequencies.
  11. **$x11$(MFCC11):** Captures even finer details of the spectral shape, often associated with very subtle characteristics of the sound.
  12. **$x12$(MFCC12):** Further refines the spectral shape, providing a comprehensive representation of the sound's characteristics.

  These coefficients are derived from the logarithm of the power spectrum of the sound signal, followed by a discrete cosine transform (DCT). They are essential in various applications, such as speech recognition, speaker identification, and audio classification, as they effectively represent the key features of the sound.

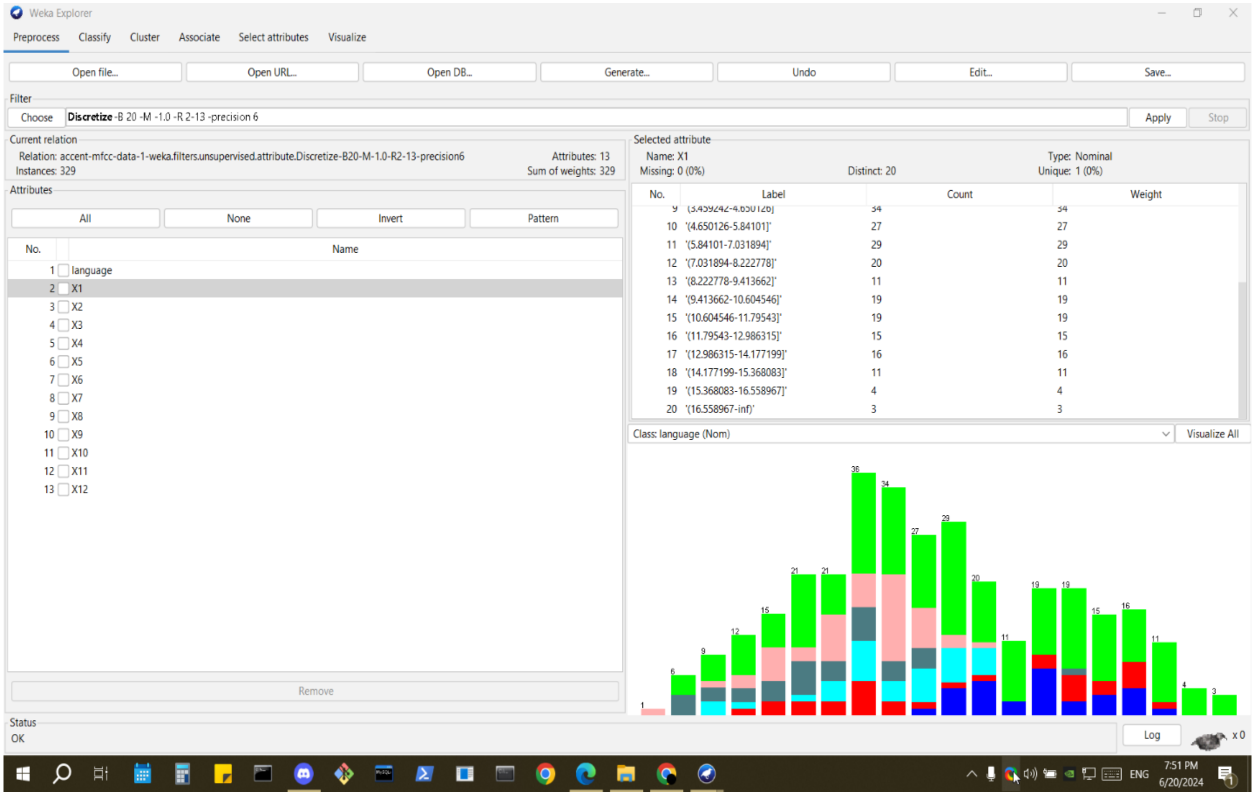
- **Language:** The classification representing the language of the speaker. It is a nominal attribute with the following possible values:
  ES: Spanish
  FR: French
  GE: German
  IT: Italian
  UK: British English
  US: American English

1. Decision Tree:

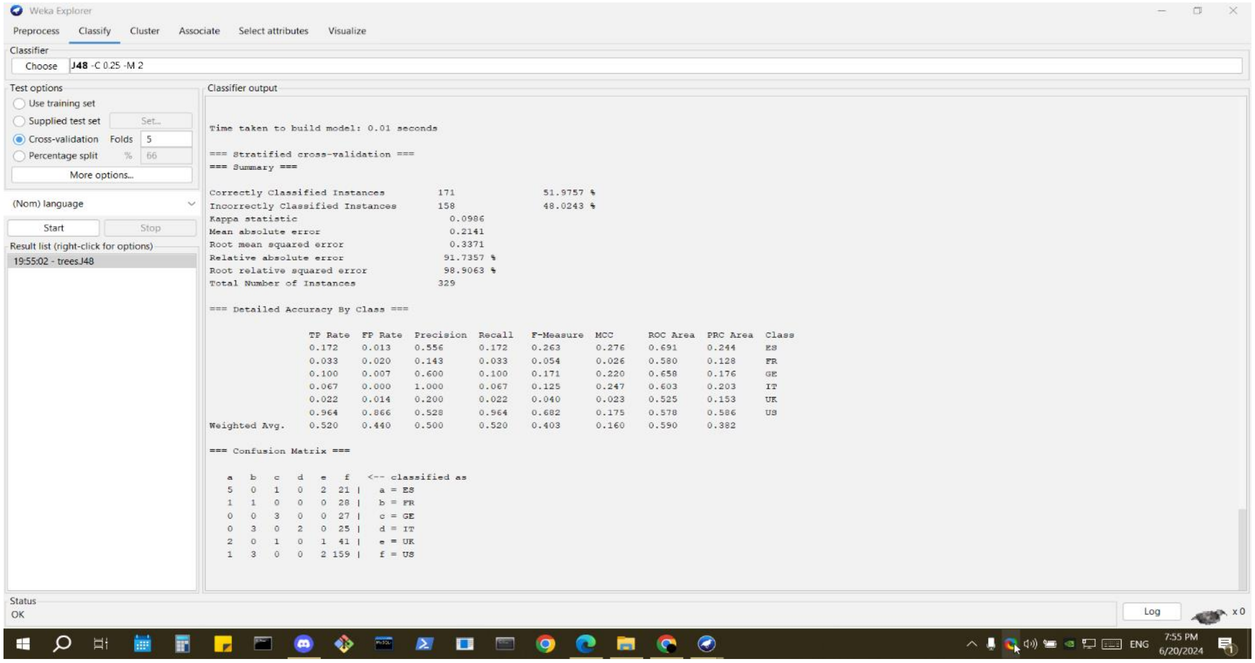
Test 1:

- Discretized all continuous attributes (X1-X12) to 20 Bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
5	0	1	0	2	21 |	a = ES
1	1	0	0	0	28 |	b = FR
0	0	3	0	0	27 |	c = GE
0	3	0	2	0	25 |	d = IT
2	0	1	0	1	41 |	e = UK
1	3	0	0	2	159 |	f = US

Accuracy: 51.9757 %

Precision: 0.500

Recall: 0.520

F1-score: 0.5098

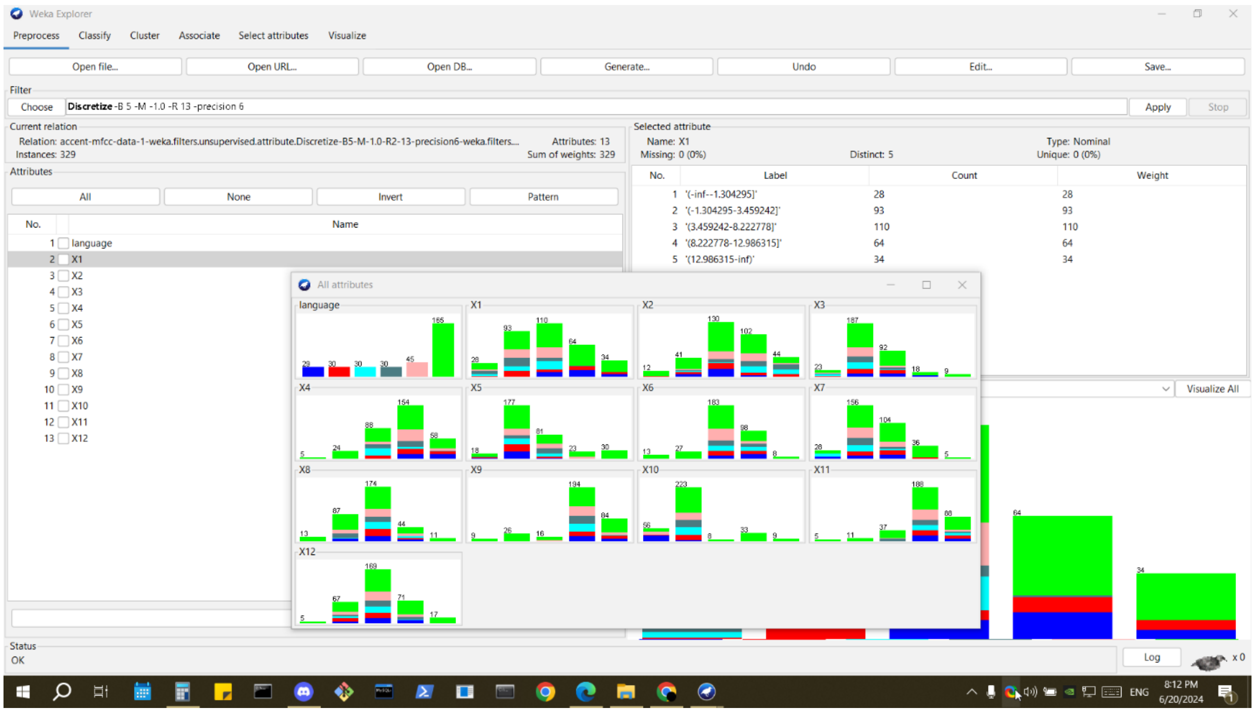
➢ Results Discussion:

Low accuracy (51.98%). Poor classification, especially for classes 'a' and 'f'.

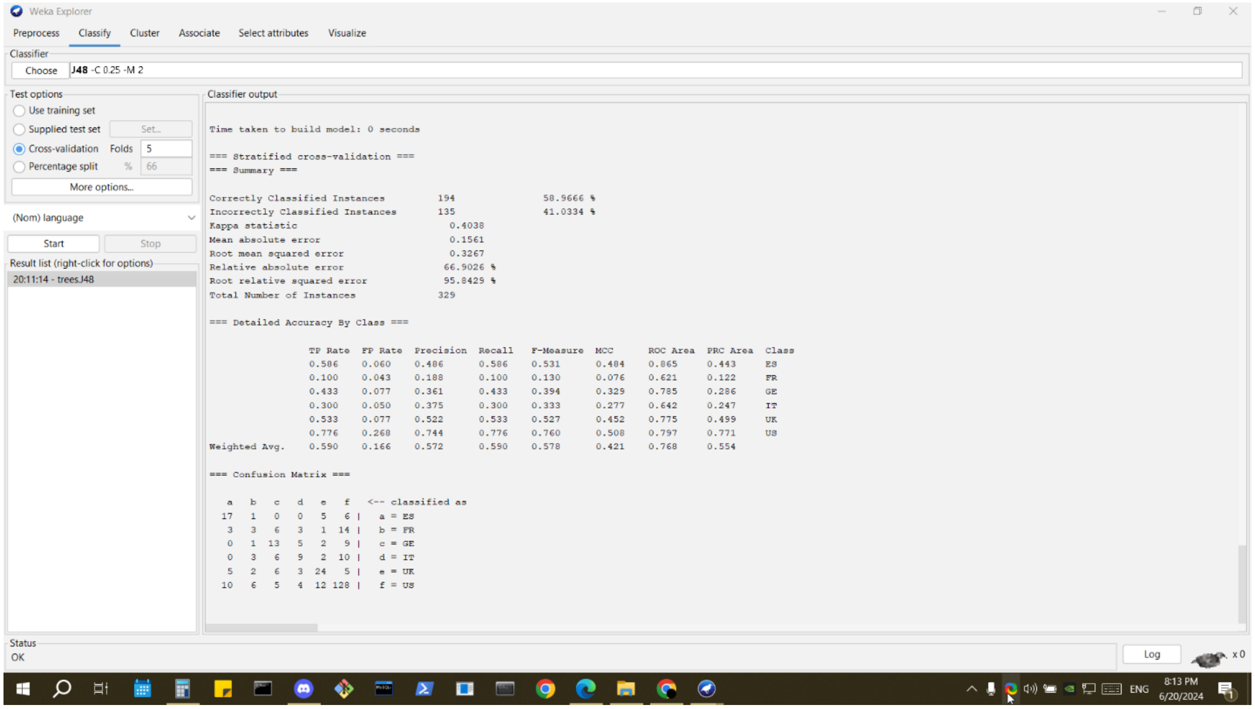
Test 2:

- Discretized all continuous attributes (X1-X12) to 5 Bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

17 1 0 0 5 6 | a = ES

3 3 6 3 1 14 | b = FR

0 1 13 5 2 9 | c = GE

0 3 6 9 2 10 | d = IT

5 2 6 3 24 5 | e = UK

10 6 5 4 12 128 | f = US

Accuracy: 58.9666 %

Precision: 0.572

Recall: 0.590

F1-score: 0.58086

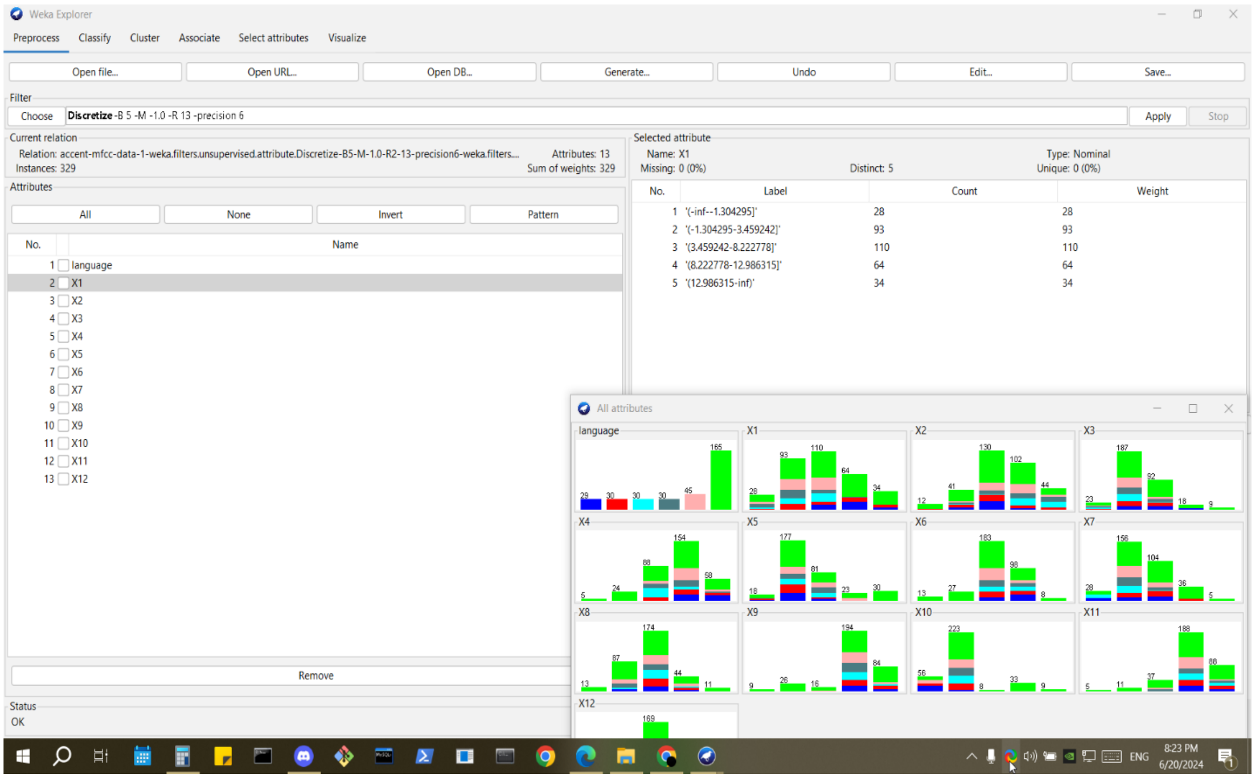
➢ Results Discussion:

Slight improvement (58.97%) with better balance across classes.

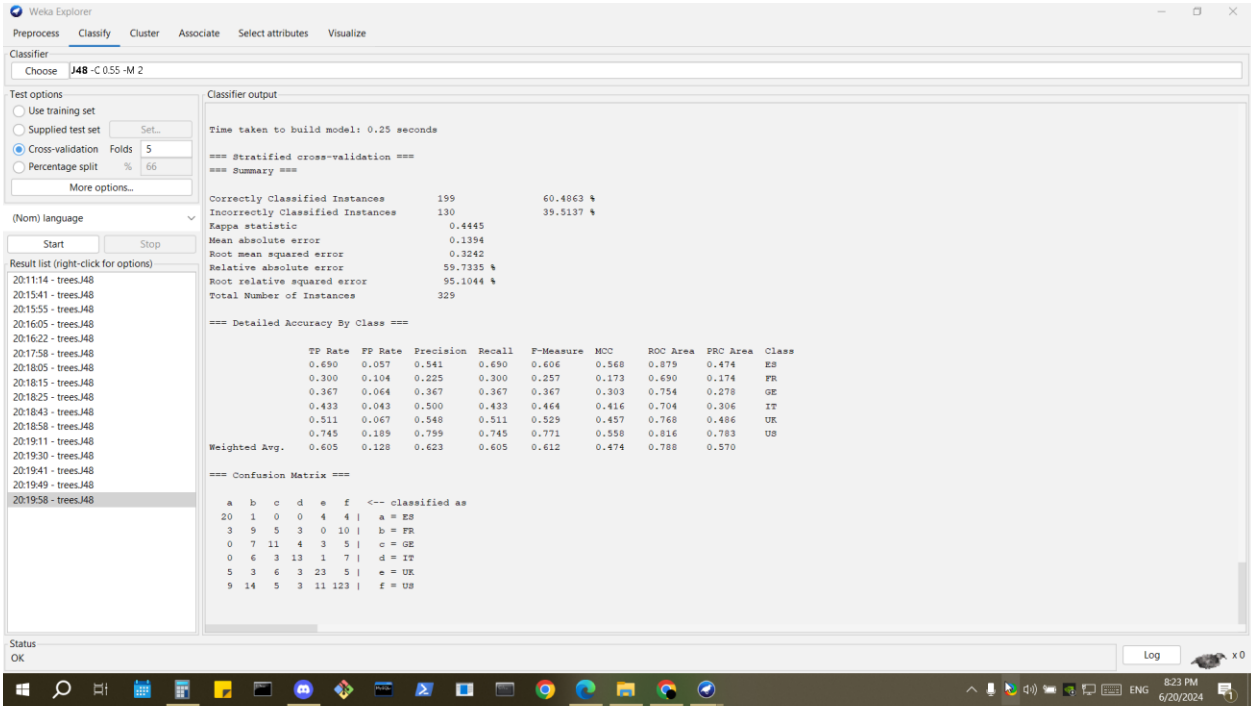
Test 3:

- Discretized all continuous attributes (X1-X12) to 5 Bins.
- Test is done on 5-fold cross validation.
- Hyper-parameter: (confidence factor) is sat to (0.55) according to many tests done to find a good value for it.

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

20 1 0 0 4 4 | a = ES

3 9 5 3 0 10 | b = FR

0 7 11 4 3 5 | c = GE

0 6 3 13 1 7 | d = IT

5 3 6 3 23 5 | e = UK

9 14 5 3 11 123 | f = US

Accuracy: 60.4868 %

Precision: 0.623

Recall: 0.605

F1-score: 0.613868

➢ Results Discussion:

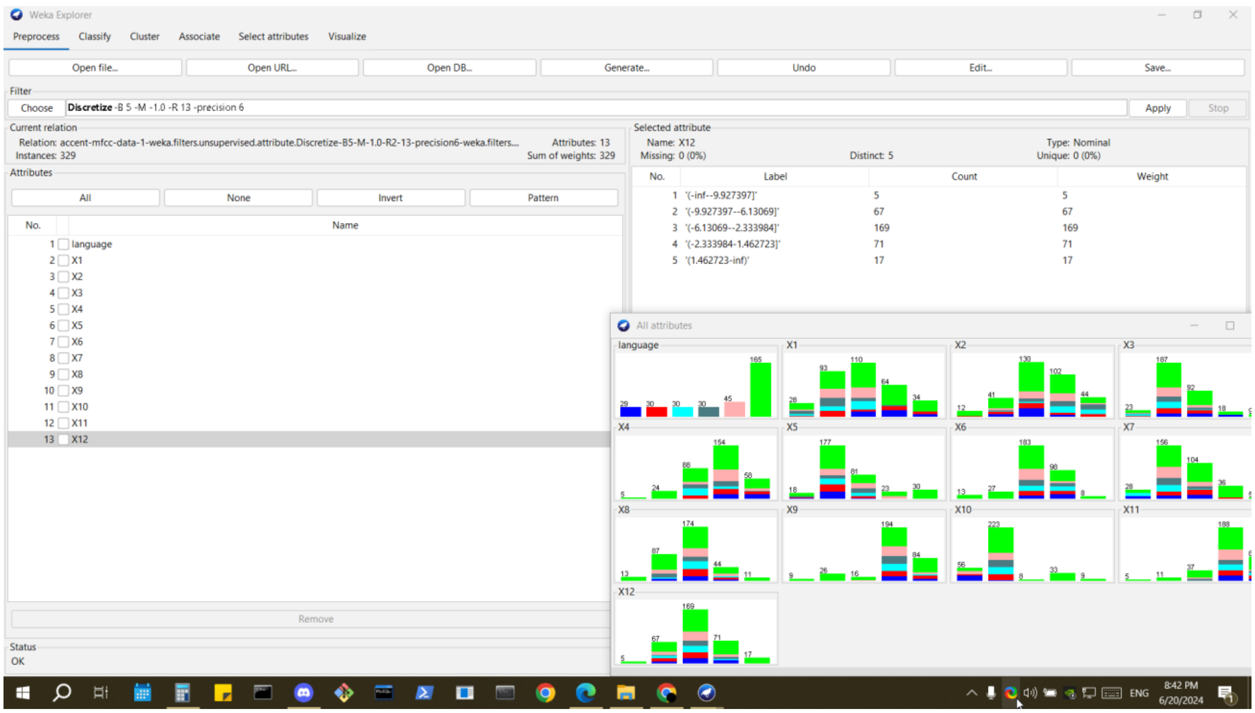
Best accuracy (60.49%) among tests with improved precision and recall.

2. Naive Bias:

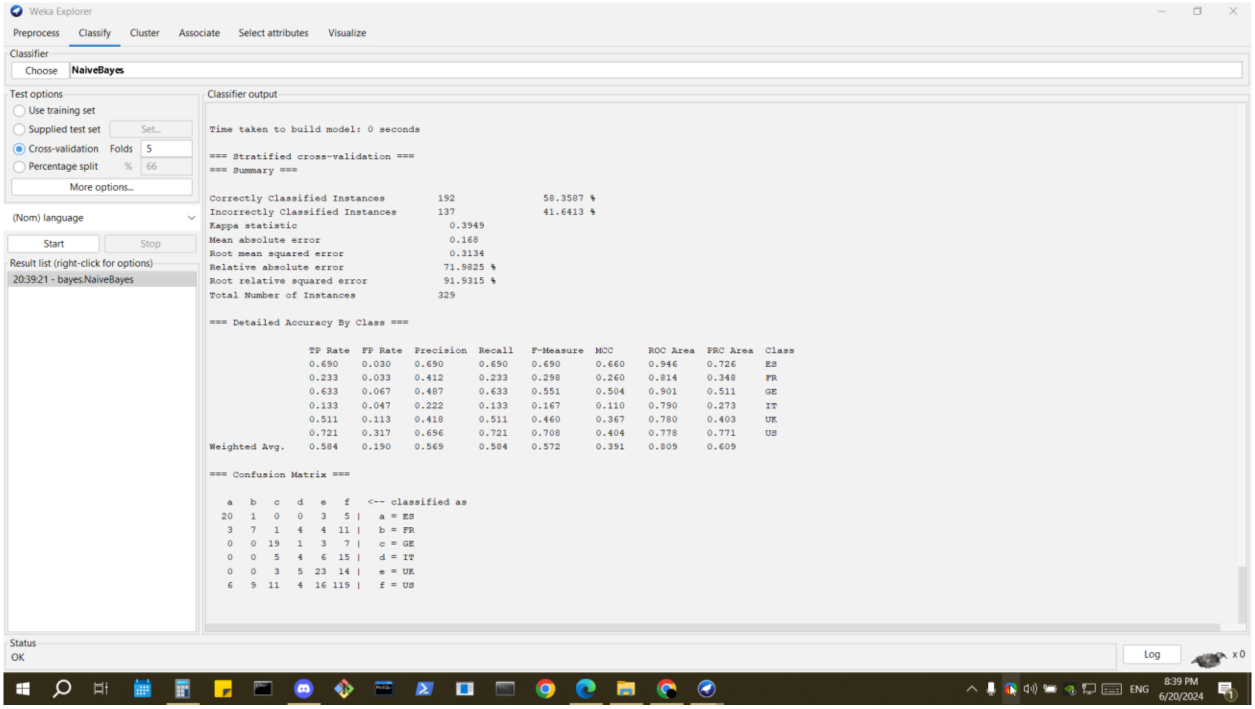
Test 1:

- Discretized all continuous attributes (X1-X12) to 20 bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

20 1 0 0 3 5 | a = ES

3 7 1 4 4 11 | b = FR

0 0 19 1 3 7 | c = GE

0 0 5 4 6 15 | d = IT

0 0 3 5 23 14 | e = UK

6 9 11 4 16 119 | f = US

Accuracy: 58.3587 %

Precision: 0.569

Recall: 0.584

F1-score: 0.5764

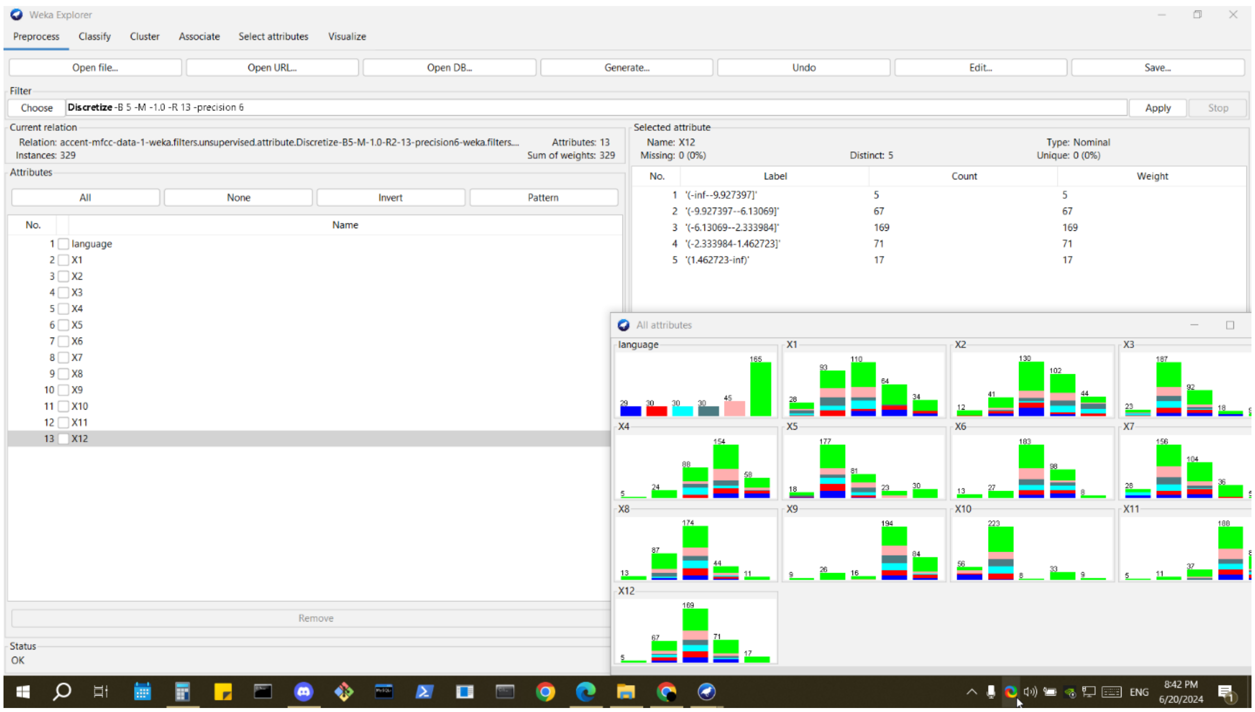
➢ Results Discussion:

Moderate accuracy (58.36%), fair precision, and recall.

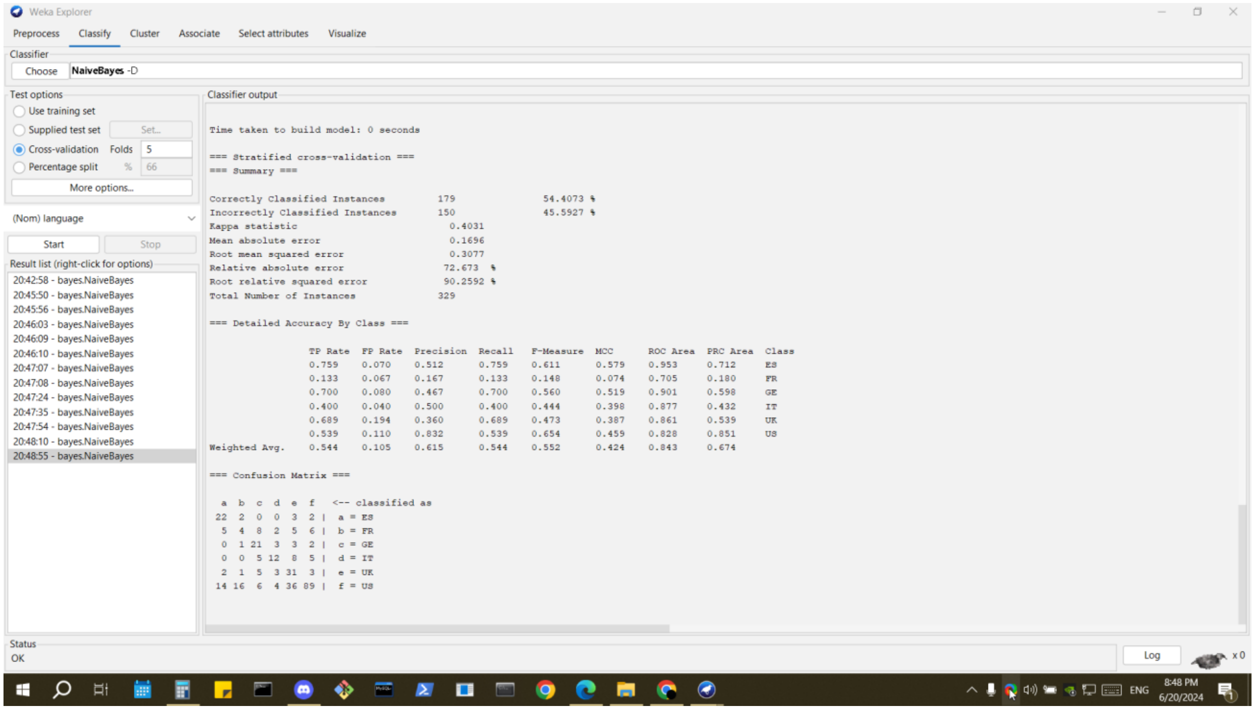
Test 2:

- Discretized all continuous attributes (X1-X12) to 5 bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➤ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

22 2 0 0 3 2 | a = ES

5 4 8 2 5 6 | b = FR

0 1 21 3 3 2 | c = GE

0 0 5 12 8 5 | d = IT

2 1 5 3 31 3 | e = UK

14 16 6 4 36 89 | f = US

Accuracy: 54.4073%

Precision: 0.615

Recall: 0.544

F1-score: 0.522

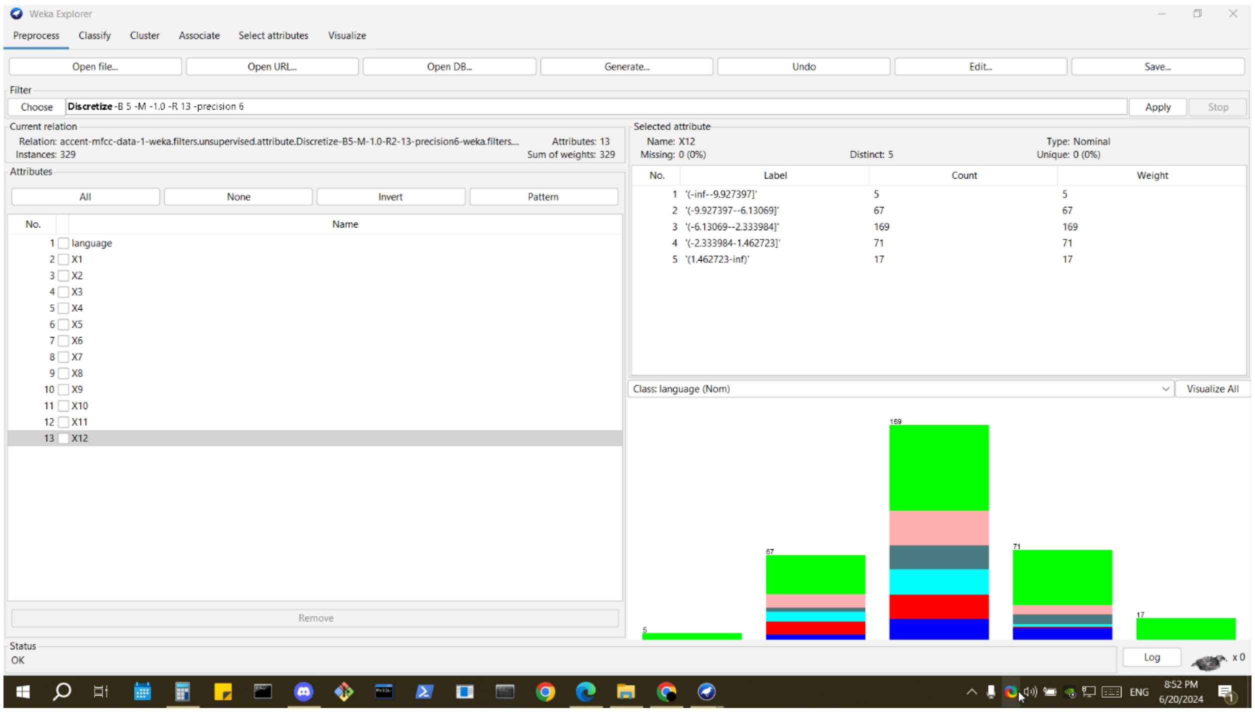
➢ Results Discussion:

Reduced accuracy (54.41%). Balanced performance with a lower F1-score.

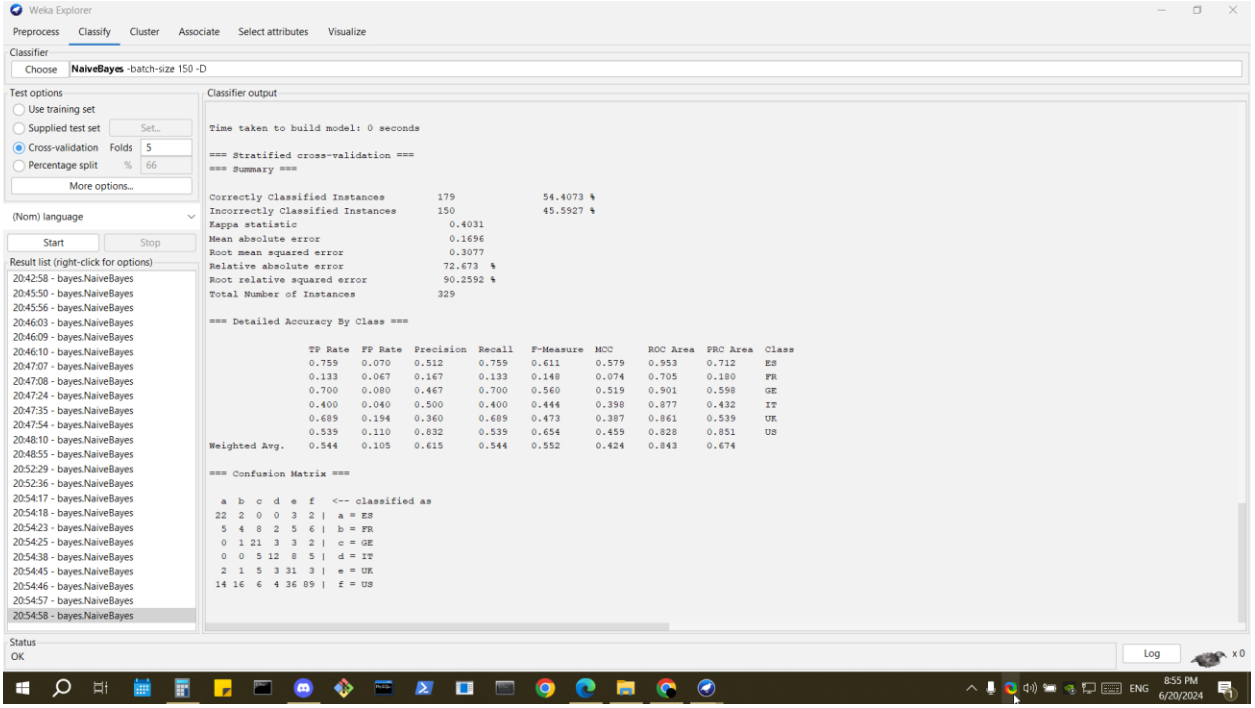
Test 3:

- Discretized all continuous attributes (X1-X12) to 5 bins.
- Test is done on 5-fold cross validation.
- Hyper-parameter: (batch size) is sat to (150) according to many tests done to find a good value for it.

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

22 2 0 0 3 2 | a = ES

5 4 8 2 5 6 | b = FR

0 1 21 3 3 2 | c = GE

0 0 5 12 8 5 | d = IT

2 1 5 3 31 3 | e = UK

14 16 6 4 36 89 | f = US

Accuracy: 54.4073%

Precision: 0.615

Recall: 0.544

F1-score: 0.522

➢ Results Discussion:

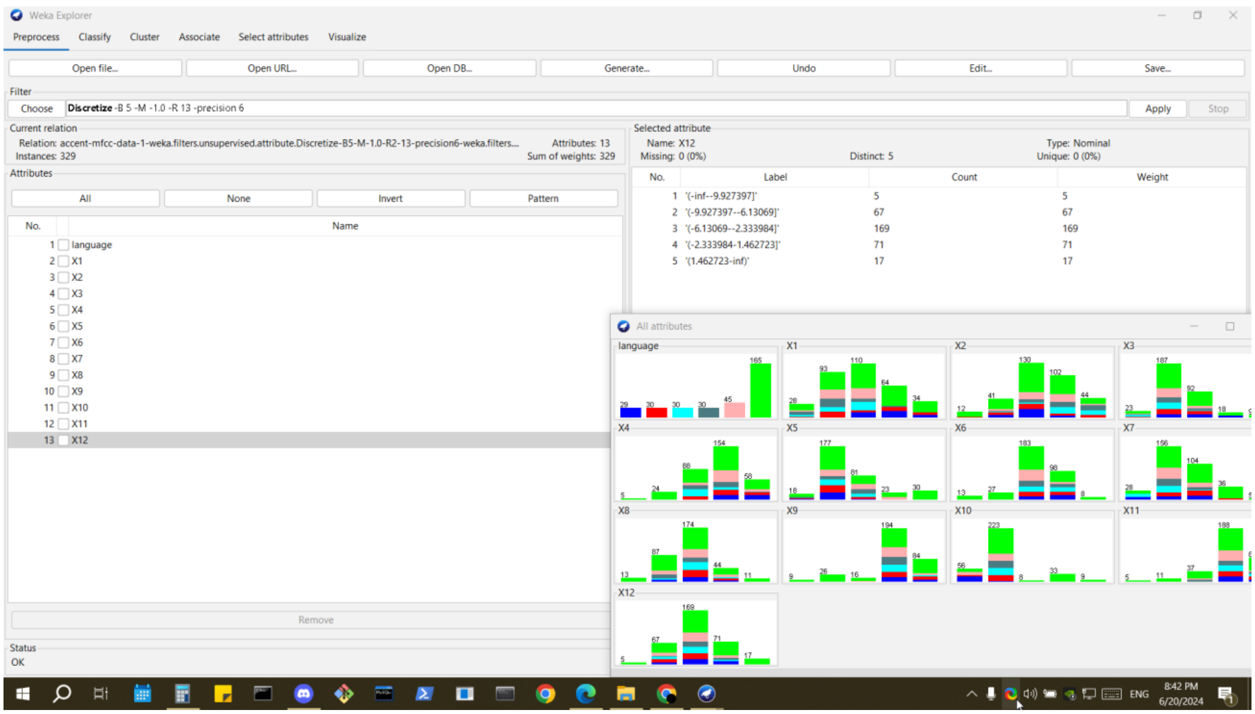
Identical results to Test 2, indicating batch size adjustment had no effect.

3. MLP

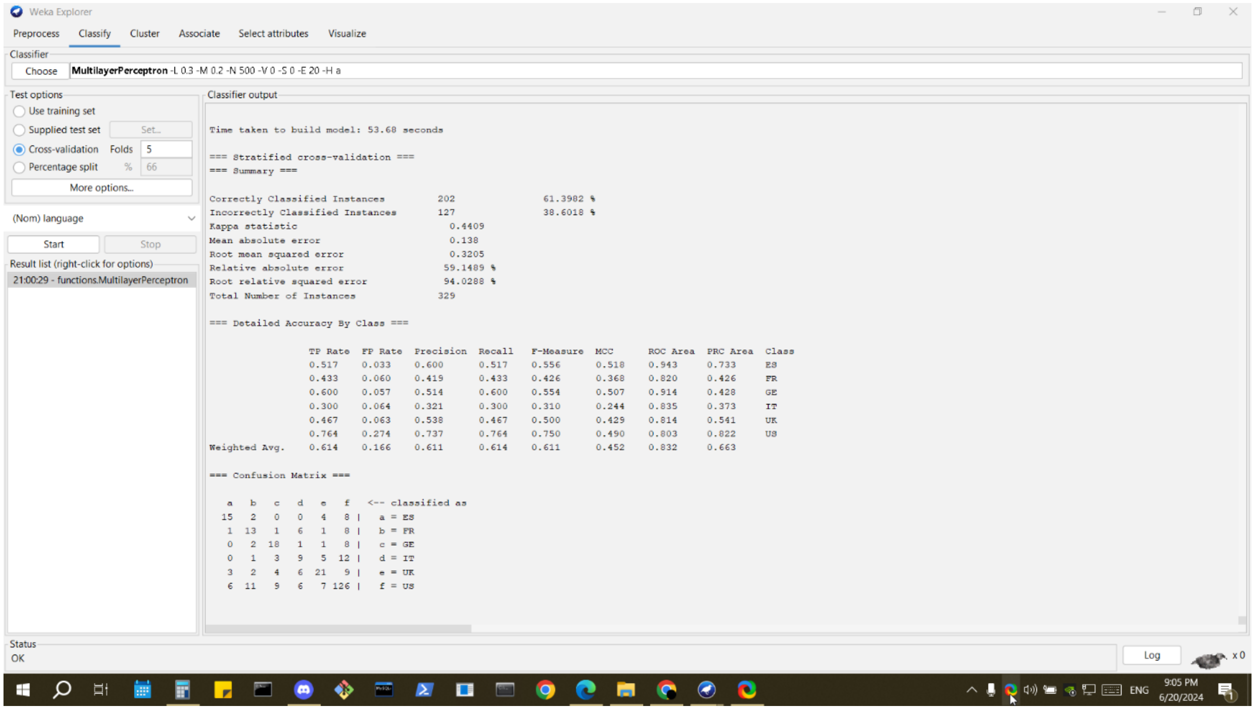
Test 1:

- Discretized all continuous attributes (X1-X12) to 20 bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

15 2 0 0 4 8 | a = ES

1 13 1 6 1 8 | b = FR

0 2 18 1 1 8 | c = GE

0 1 3 9 5 12 | d = IT

3 2 4 6 21 9 | e = UK

6 11 9 6 7 126 | f = US

Accuracy: 61.3982 %

Precision: 0.611

Recall: 0.614

F1-score: 0.611

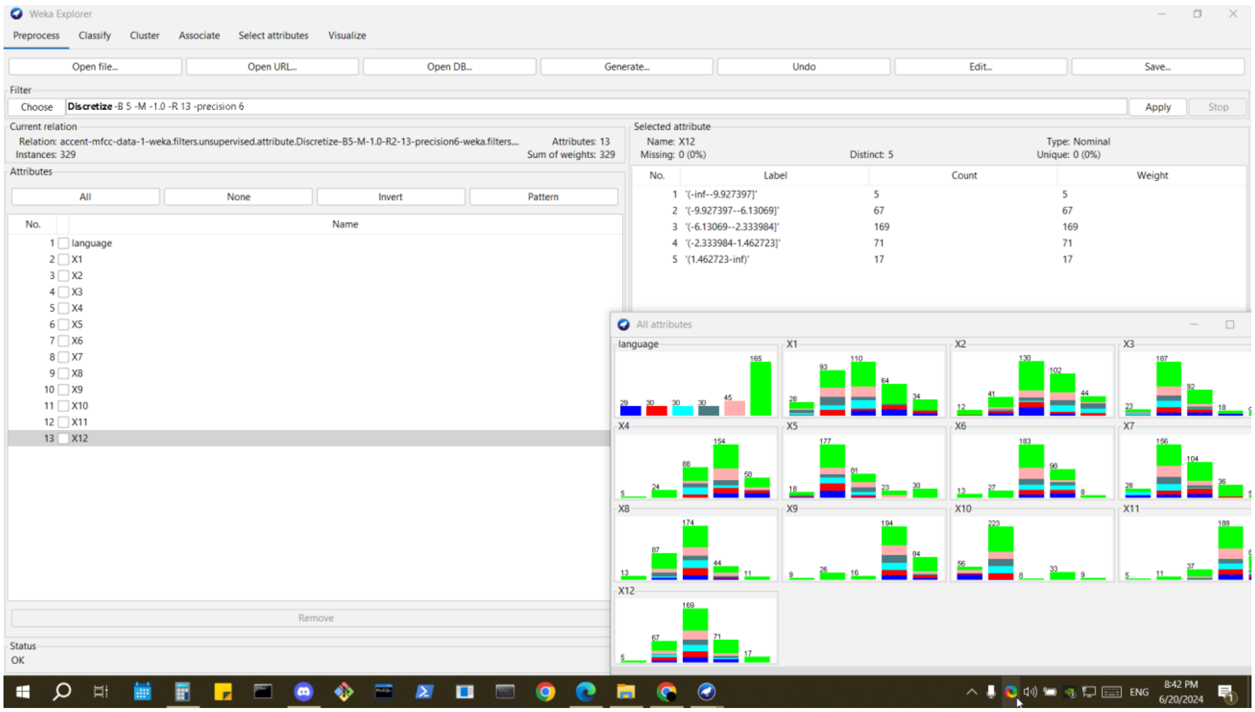
➢ Results Discussion:

Good accuracy (61.40%), balanced precision, and recall.

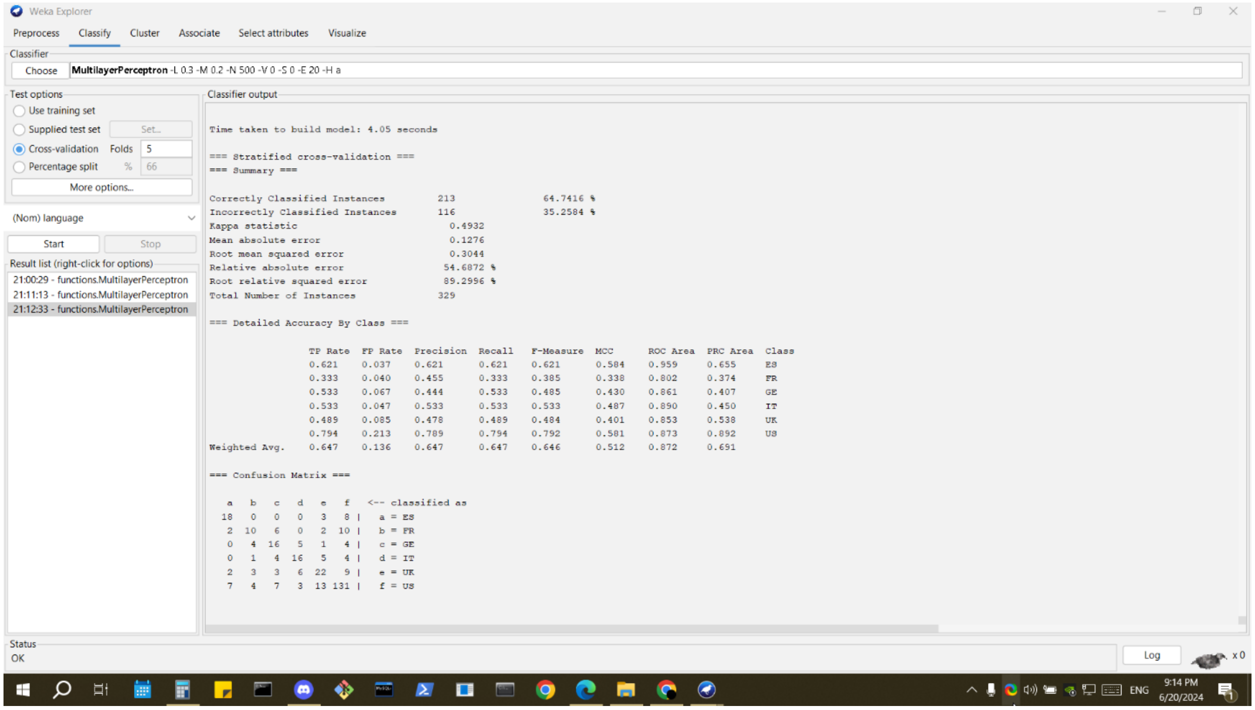
Test 2:

- Discretized all continuous attributes (X1-X12) to 5 bins.
- Test is done on 5-fold cross validation.
- No hyper-parameter is changed (default).

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

18 0 0 0 3 8 | a = ES

2 10 6 0 2 10 | b = FR

0 4 16 5 1 4 | c = GE

0 1 4 16 5 4 | d = IT

2 3 3 6 22 9 | e = UK

7 4 7 3 13 131 | f = US

Accuracy: 64.7416 %

Precision: 0.647

Recall: 0.647

F1-score: 0.646

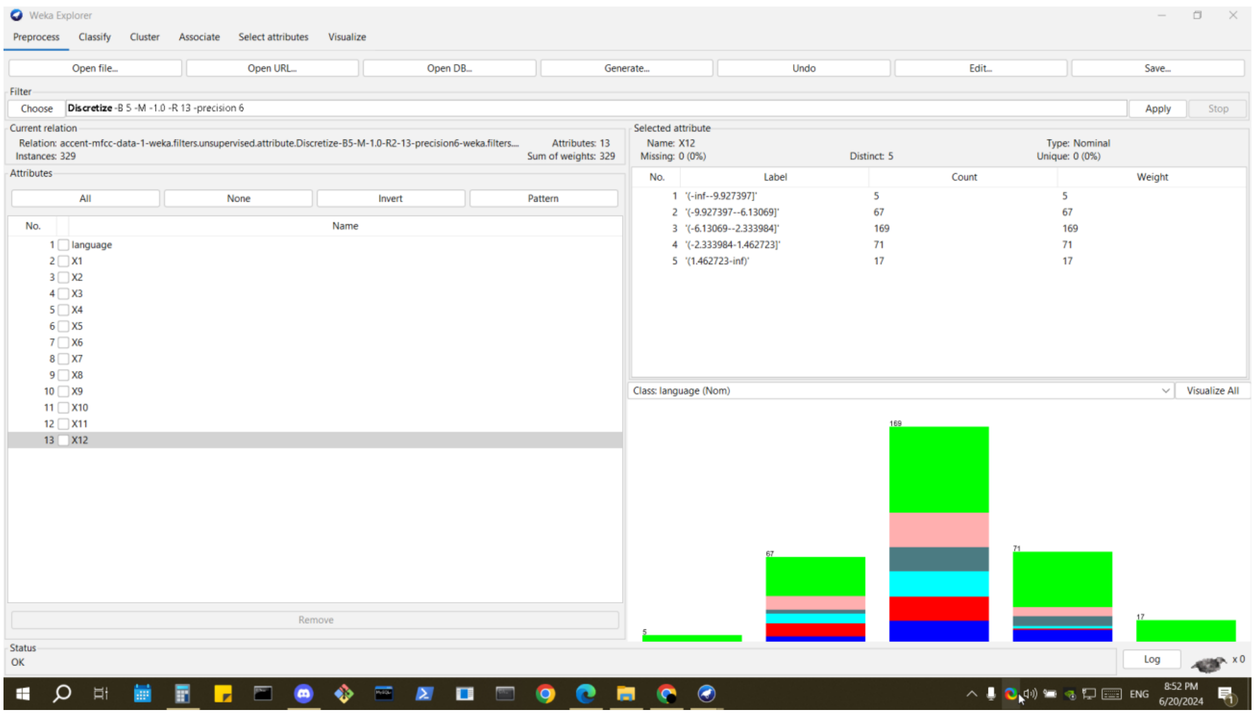
➢ Results Discussion:

Improved accuracy (64.74%) with consistent classification across classes.

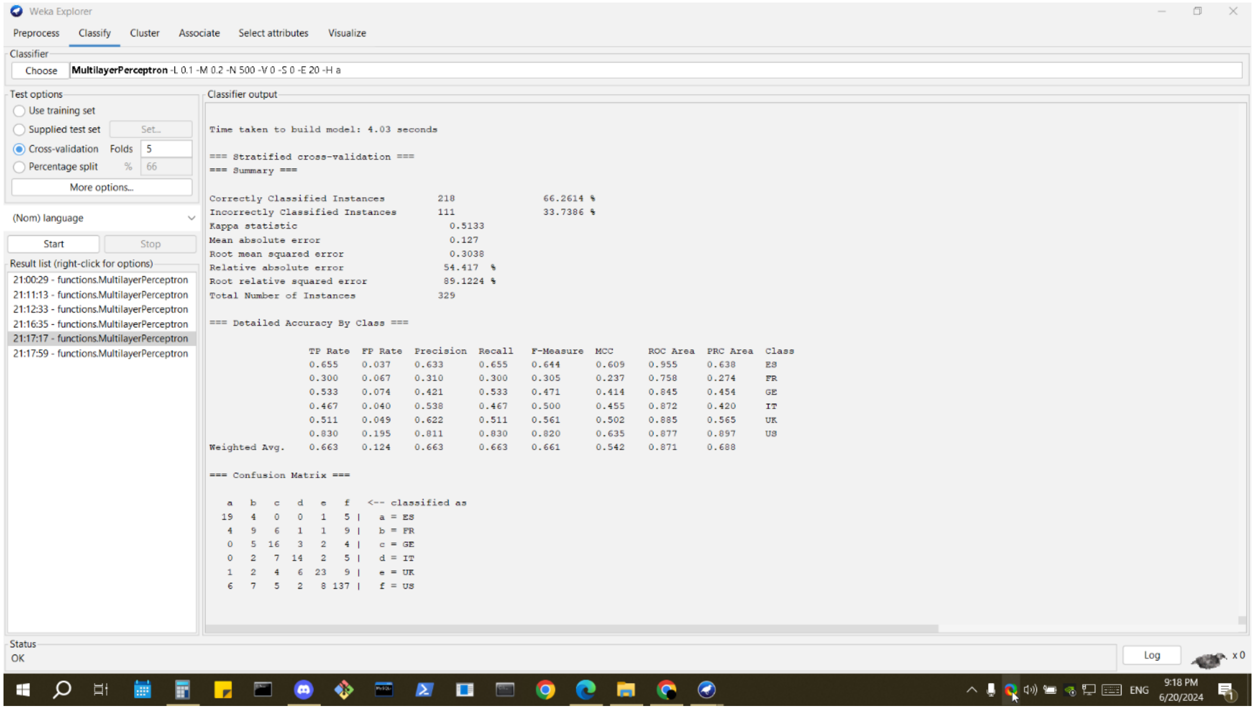
Test 3:

- Discretized all continuous attributes (X1-X12) to 5 bins.
- Test is done on 5-fold cross validation.
- Hyper-parameter: (learning rate) is sat to (0.2) according to many tests done to find a good value for it.

➢ Preprocessing:



➢ Classification:



➢ Result:

=== Confusion Matrix ===

a b c d e f <-- classified as

19 4 0 0 1 5 | a = ES

4 9 6 1 1 9 | b = FR

0 5 16 3 2 4 | c = GE

0 2 7 14 2 5 | d = IT

1 2 4 6 23 9 | e = UK

6 7 5 2 8 137 | f = US

Accuracy: 66.2614 %

Precision: 0.663

Recall: 0.663

F1-score: 0.661

➢ Results Discussion:

Best performance (66.26%) with high precision and recall.

❖ Changed Hyper Parameters:

- **Confidence Factor in Decision Tree:**

  The confidence factor in a decision tree controls the extent of pruning during the training process. A lower confidence factor leads to more aggressive pruning, reducing overfitting but possibly underfitting the data. A higher confidence factor results in less pruning, capturing more data details but increasing the risk of overfitting.

- **Batch Size in Naïve**:
  Bayes Batch size in Naive Bayes, though not commonly referenced, can affect the efficiency of processing large datasets. Using larger batch sizes can make the training process more efficient by reducing the number of updates to model parameters, while smaller batch sizes can lead to more frequent updates, which might be beneficial for capturing data variability but could be computationally expensive.
  - when changing the batch size does not affect the learning process, this means current training setup and data are well-balanced, leading to consistent model performance.

- **Learning Rate in MLP:**
  The learning rate in a Multi-Layer Perceptron (MLP) determines the size of the steps taken during gradient descent optimization. A high learning rate can speed up learning but may cause the model to overshoot minima, leading to instability. A low learning rate ensures stable convergence but can make the training process slow and prone to getting stuck in local minima.

❖ Summary Comparison

- **Decision Tree:**

  Best accuracy with 5 bins and confidence factor adjustment (Test 3: 60.49%).

- **Naive Bayes:**

  Moderate performance, best with 20 bins (Test 1: 58.36%).

- **MLP:**

  Overall highest accuracy and balanced performance (Test 3: 66.26%).

In summary, MLP outperforms both Decision Tree and Naive Bayes, particularly with hyper-parameter tuning. Decision Tree shows improvement with discretization and confidence factor adjustments, while Naive Bayes performs moderately well but shows limited sensitivity to hyper-parameter changes.