Decision Models for Nutri-Score

- Master BDMA – Decision Modeling

Adrian Patricio, Joel Anil Jose

NUTRI-SCORE

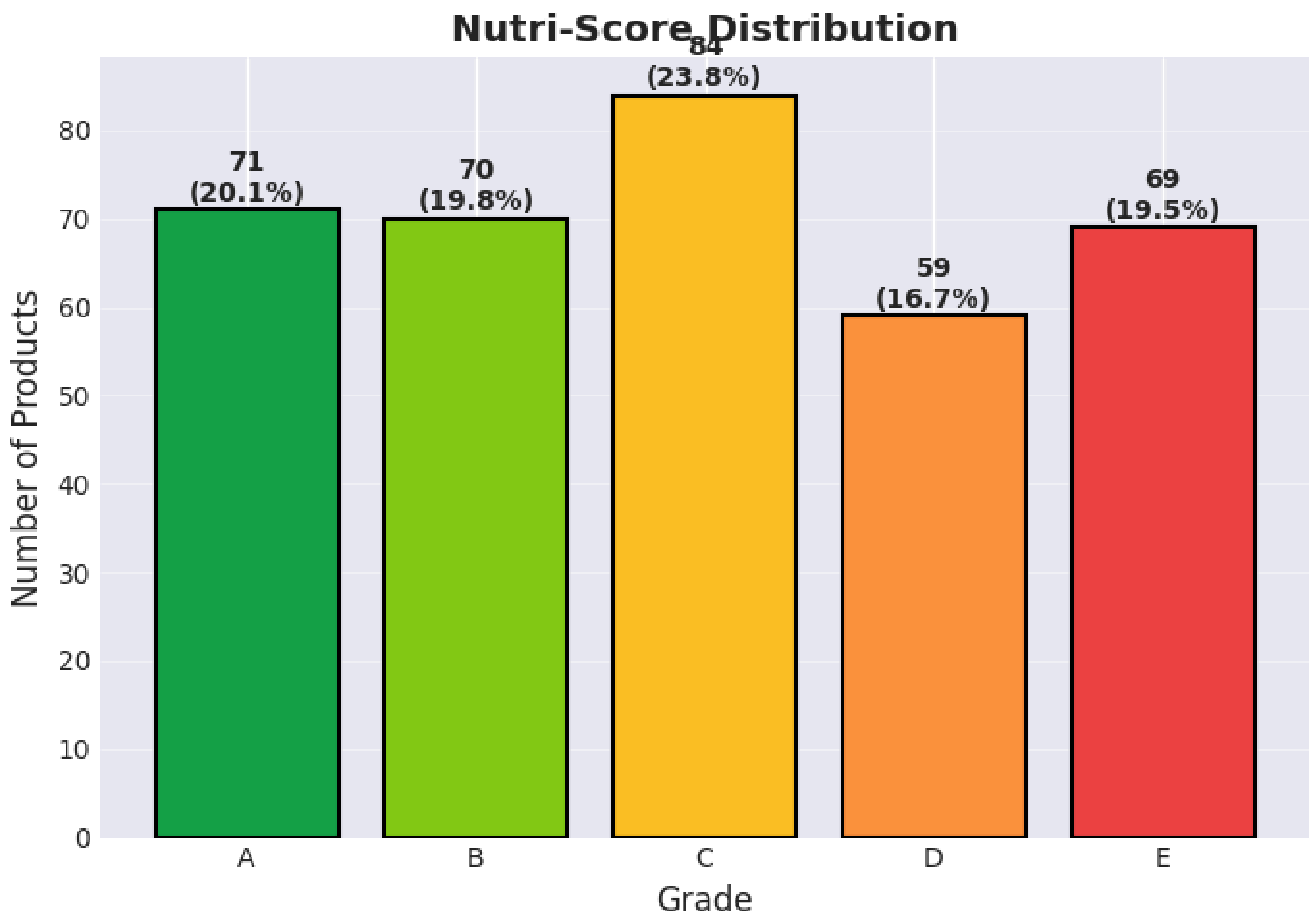
A B C D E

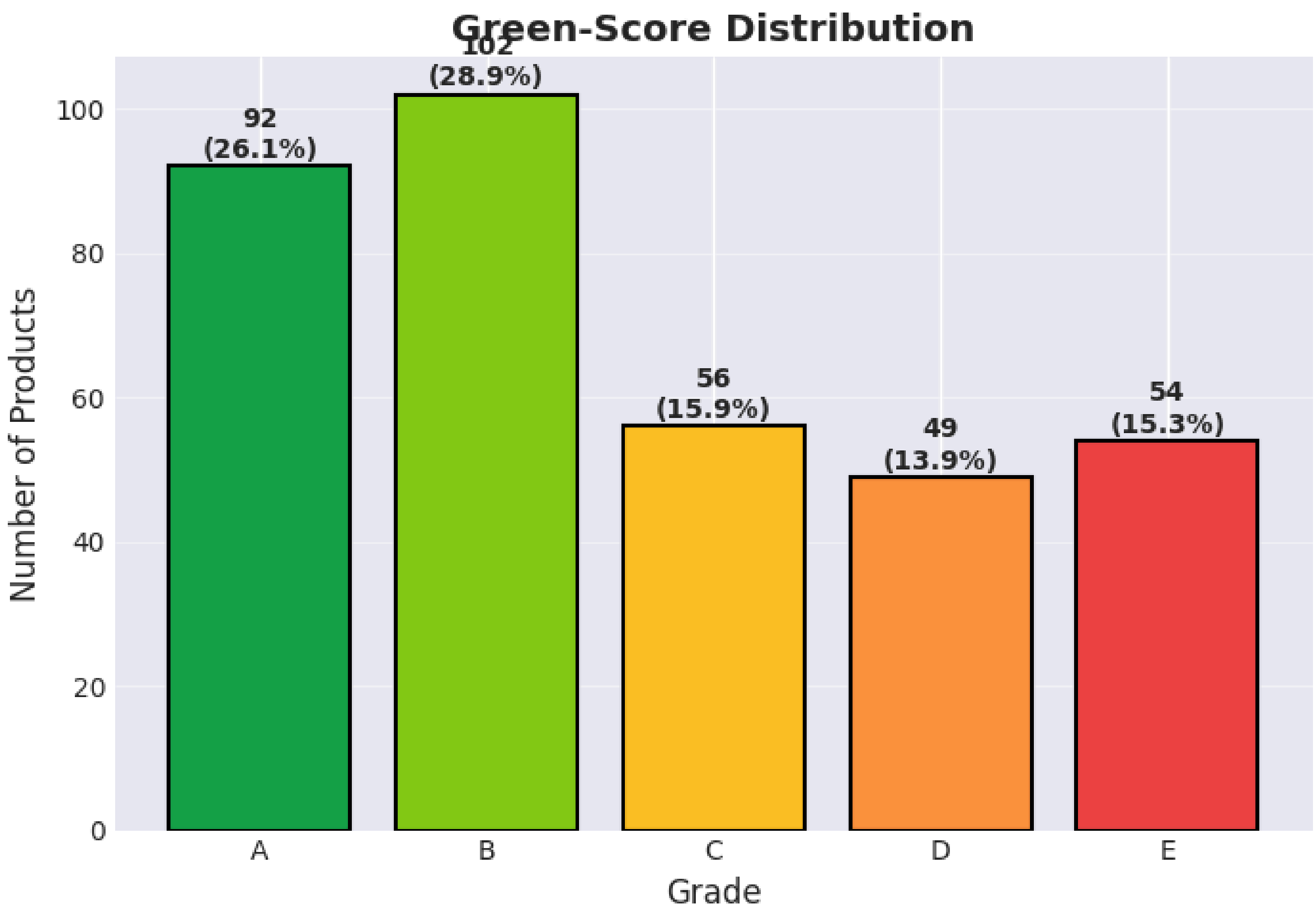
Content

- Data Collection
- Nutri-score
- ELECTRE TRI
- Weighted Sum
- Machine Learning Approaches
- Comparison

Data Collection

- **Data Source:** Open Food Facts API
- **Selected Columns:** product_name, energy_100g, saturated_fat_100g, sugars_100g, salt_100g, proteins_100g, fiber_100g, fvl_percent, green_score_value, nutri_score_value, green-score_label, nutri_score_label
- Final preprocessed dataset has **353 items**.





Nutri-Score Visualization

Technology : React.js

Inputs:

7 nutritional components

Scoring:

Negative points (0–55)

Positive points (0–17)

Nutri-Score Calculator | NutriCalc

Calculate the Nutri-Score for food products with our professional calculator

vercel.app

Input

Nutritional Values

Enter the nutritional composition per 100g

Energy

1100 kJ/100g

Saturated Fatty Acids

0.4 g/100g

Sugars

4.9 g/100g

Salt

1.2 g/100g

Fiber

7.2 g/100g

Proteins

7.7 g/100g

Fruits, Vegetables, Legumes & Nuts

3,2575757 %

Calculate Nutri-Score

Output

Your Nutri-Score

Based on the nutritional composition

2 Numeric Score

Numeric Score Range: -17 to +55

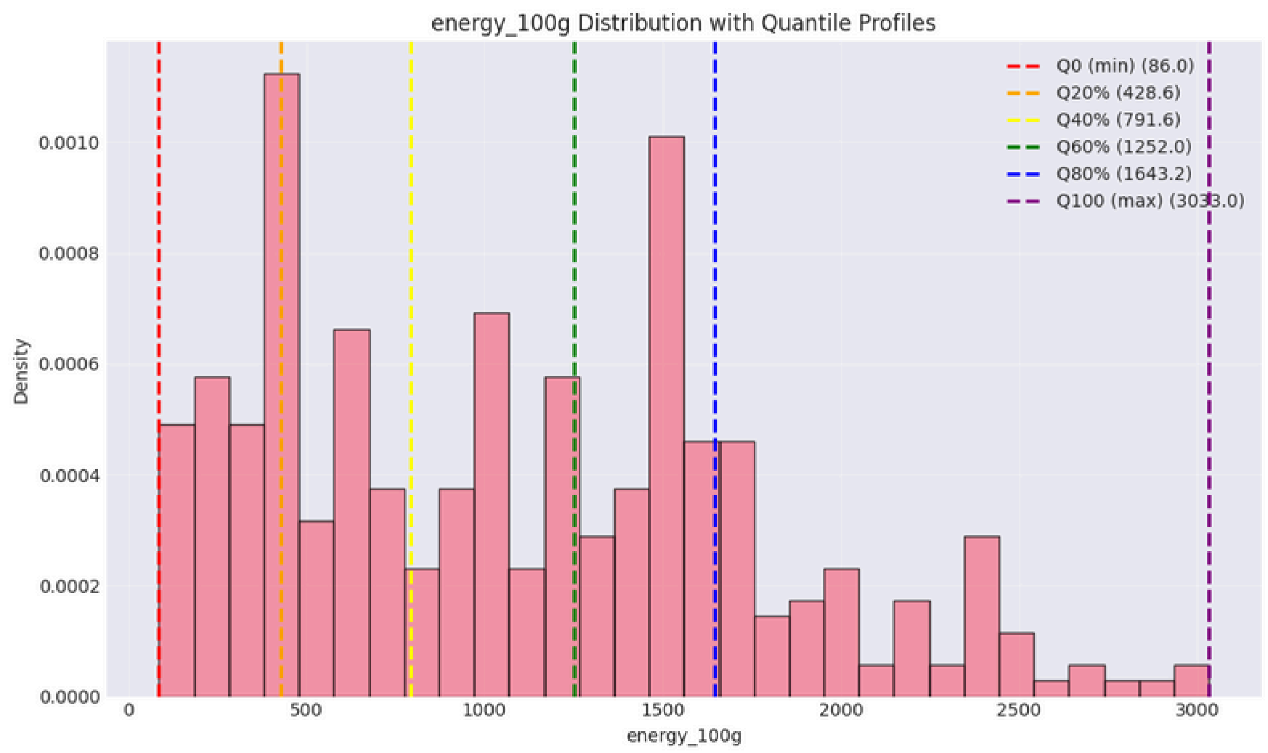
Nutri-Score Letter

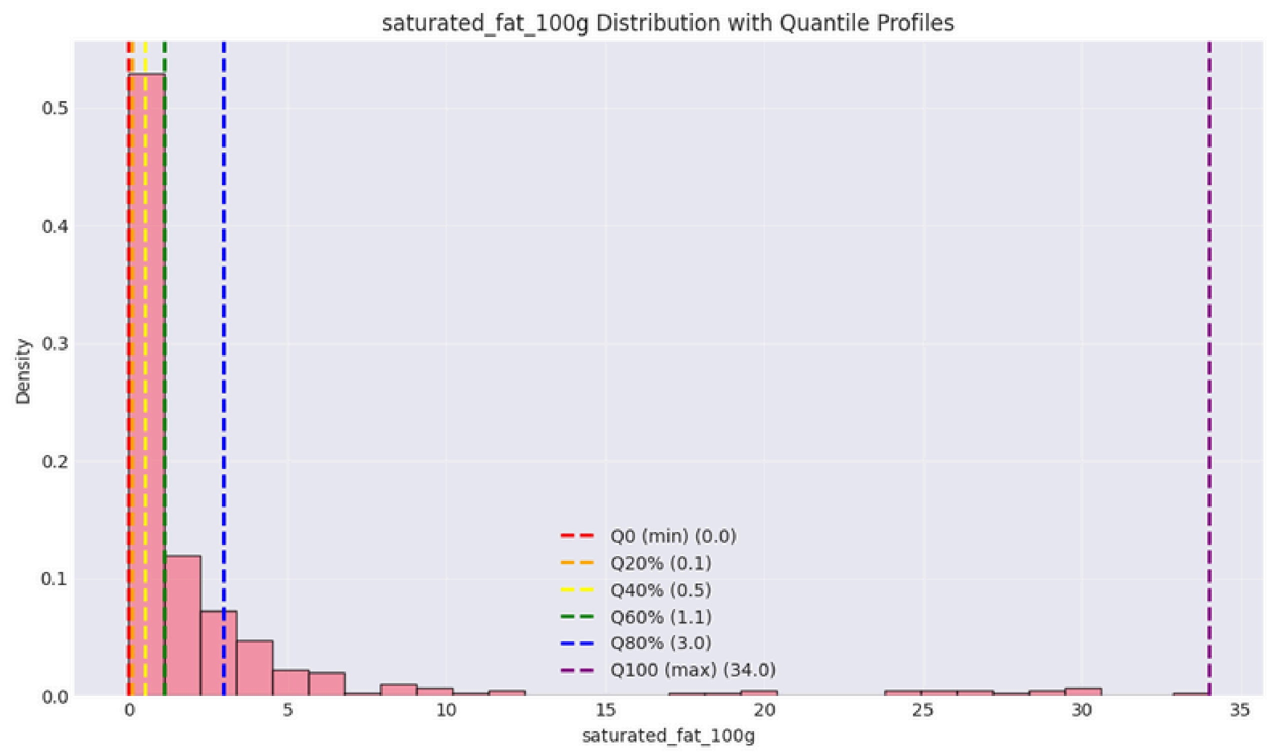
A **B** C D E

Negative Component:	9
Fiber Points:	4
Proteins Points:	3
Fruit/Veg Points (Positive Component):	0
Final Score:	**2**

ELECTRE-TRI Model: Profile and Weights Selection

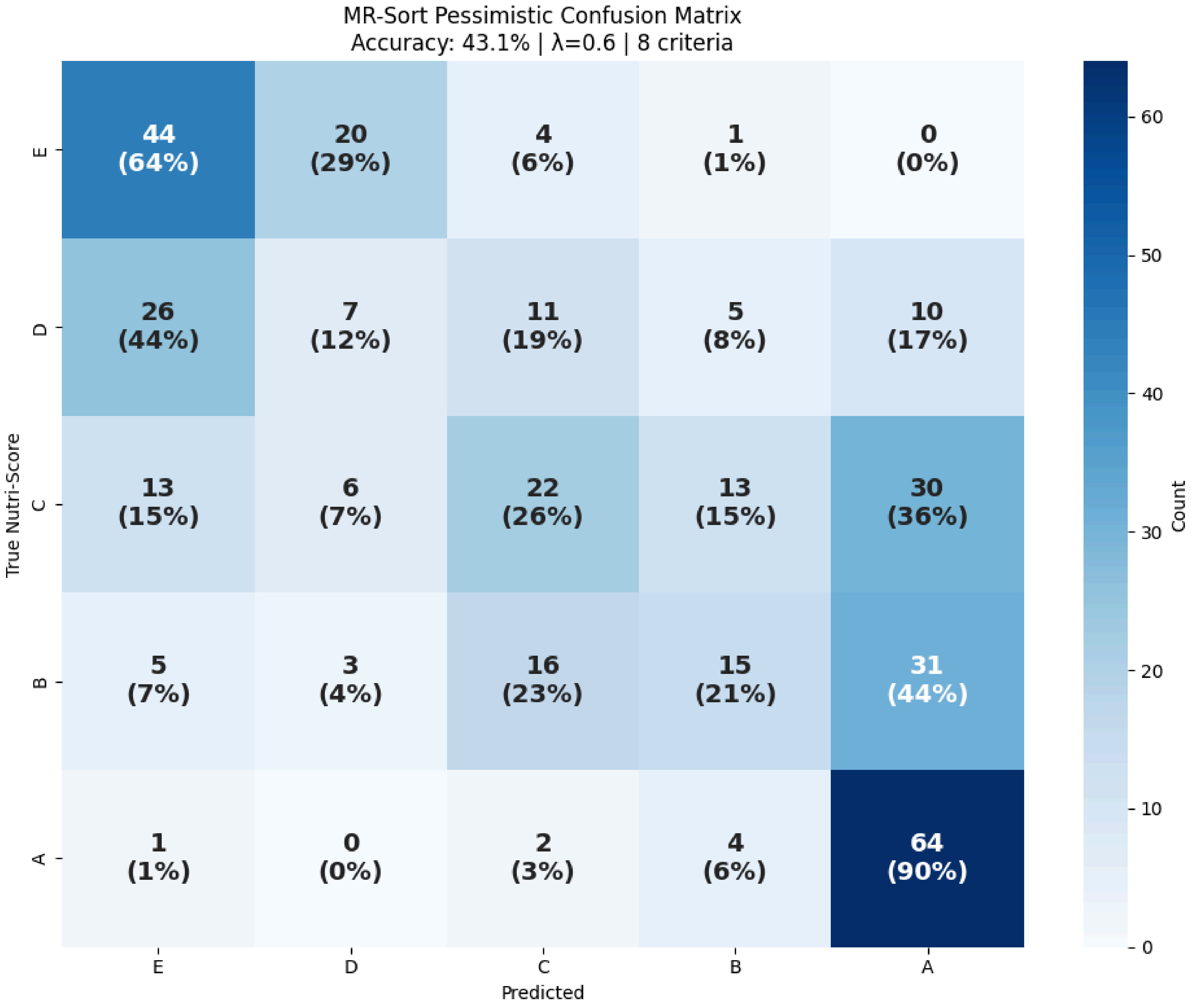
- For simplicity, MR-SORT framework was used.
- The profiles were built on equal quintiles (20% intervals)
- Weights are set equally (1/8 for each criterion)





	energy_100g	saturated_fat_100g	sugars_100g	salt_100g	proteins_100g	fiber_100g	fvl_percent	green_score_value
Lower	3033	34	98.5	17.2	0	0	0	0
E-D	1643.2	3	25.44	1.19	1.2	0	0	36
D-C	1252	1.12	4.2	0.74	5.38	0.6	0	54
C-B	791.6	0.5	2.58	0.47	8	2	2.5	70
B-A	428.6	0.1	0.9	0.02	12	4.96	34.77	78
Upper	86	0	0	0	35	17.1	100	100

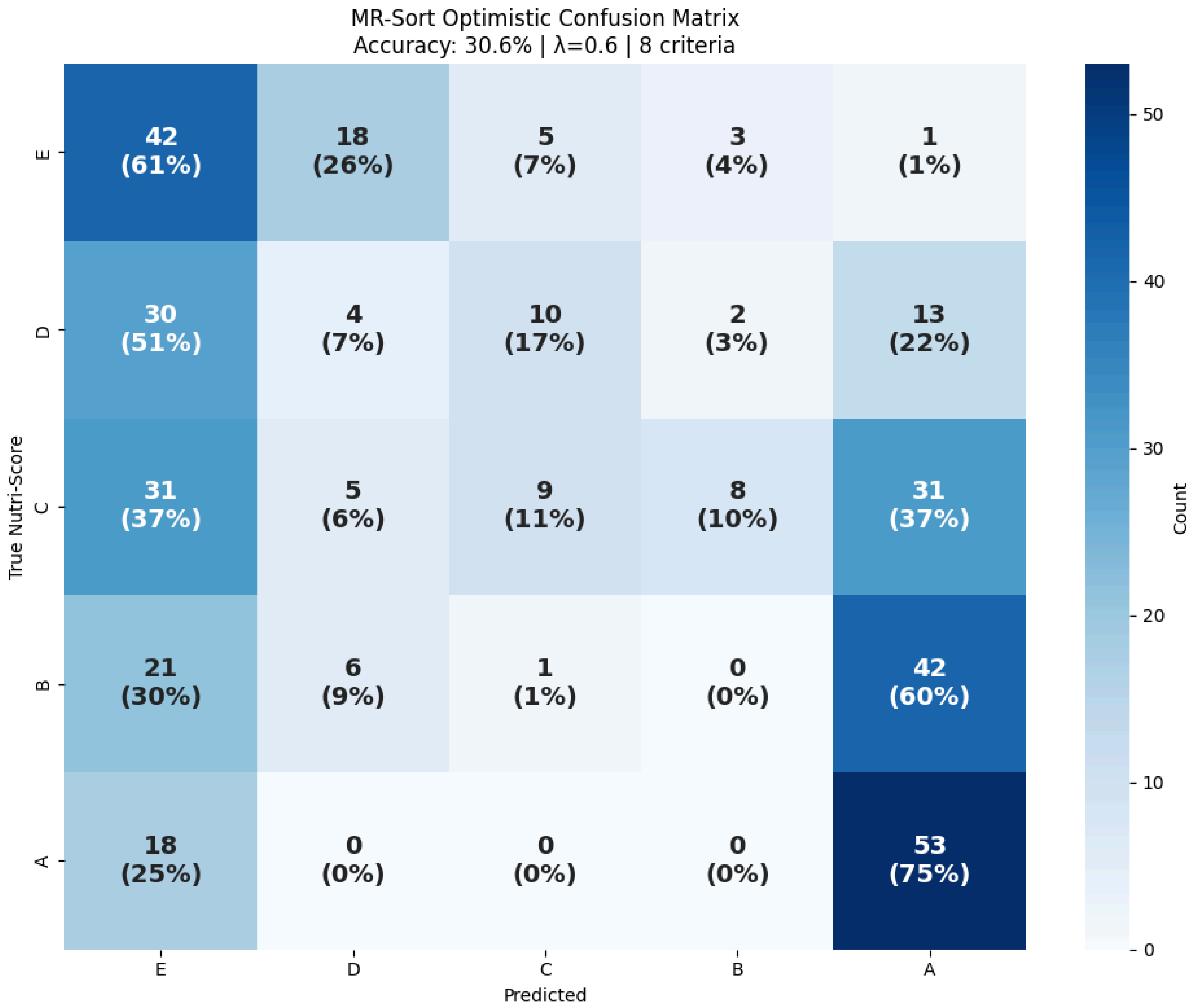
ELECTRE-TRI Model: Results (Pessimistic)




```
Per-category accuracy:
  E: 63.8% (69 samples)
  D: 11.9% (59 samples)
  C: 26.2% (84 samples)
  B: 21.4% (70 samples)
  A: 90.1% (71 samples)

              precision    recall  f1-score   support

           A       0.47      0.90      0.62        71
           B       0.39      0.21      0.28        70
           C       0.40      0.26      0.32        84
           D       0.19      0.12      0.15        59
           E       0.49      0.64      0.56        69

    accuracy                           0.43       353
   macro avg       0.39      0.43      0.38       353
weighted avg       0.40      0.43      0.39       353
```

- Same values for both 0.6 and 0.7 thresholds
- Many C/B items get pushed to A or E, indicating that profiles between B/A or D/E are not that good

ELECTRE-TRI Model: Results (Optimistic)

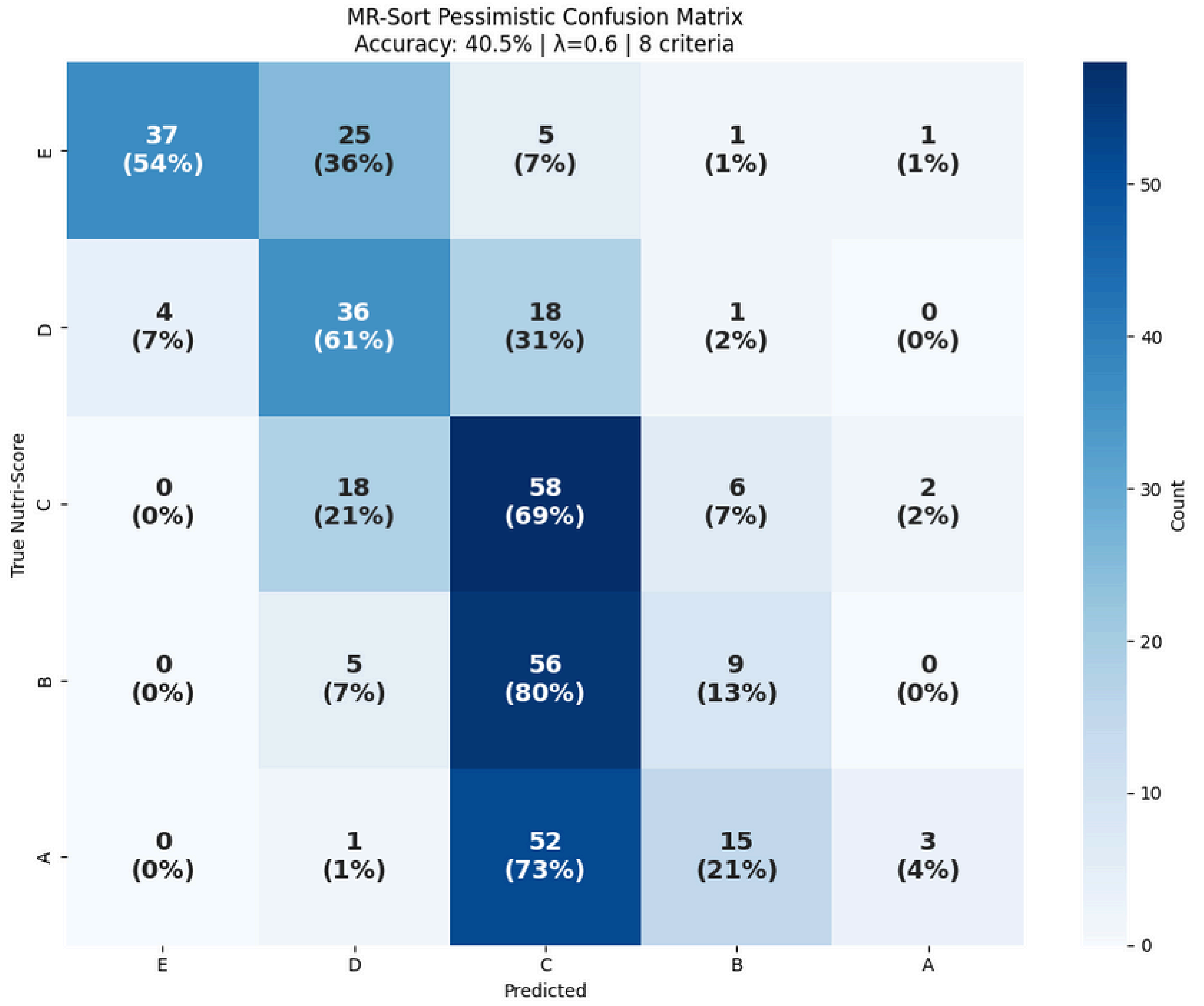


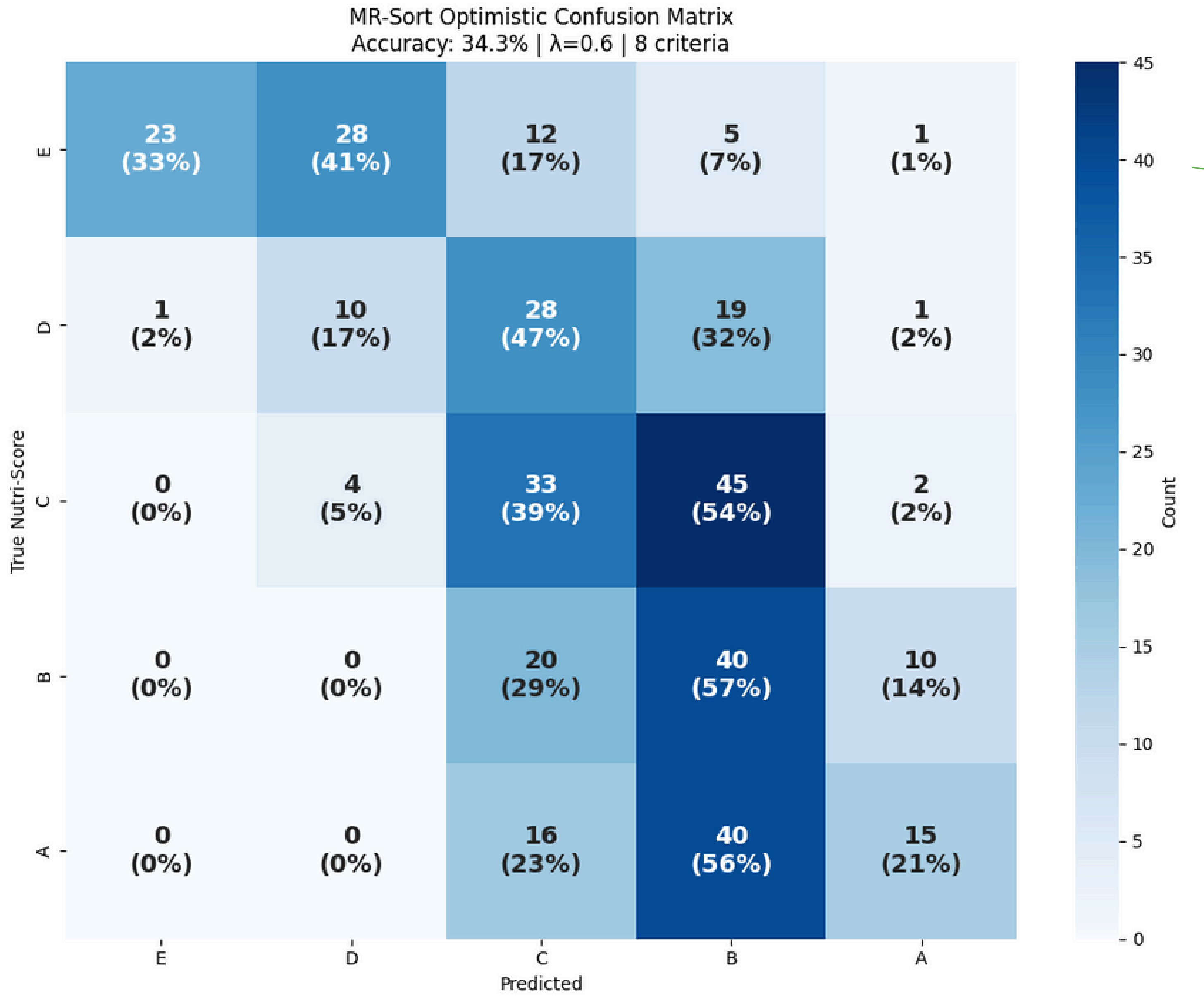

```
Per-category accuracy:
  E: 60.9% (69 samples)
  D: 6.8% (59 samples)
  C: 10.7% (84 samples)
  B: 0.0% (70 samples)
  A: 74.6% (71 samples)

              precision    recall  f1-score   support

           A       0.38      0.75      0.50        71
           B       0.00      0.00      0.00        70
           C       0.36      0.11      0.17        84
           D       0.12      0.07      0.09        59
           E       0.30      0.61      0.40        69

    accuracy                           0.31       353
   macro avg       0.23      0.31      0.23       353
weighted avg       0.24      0.31      0.23       353
```

- Same values for both 0.6 and 0.7 thresholds
- Heavily biased toward assigning A or E, with very few predictions in middle classes.

ELECTRE-TRI Model: Other Approaches

- **Entropy-Based:** Weights derived from entropy computations
- **Optimization:** Using differential evolution to find weights that minimize error
- **Correlation-Based: Weights derived from correlation with nutri-score labels**





- For pessimistic, heavily biased towards assigning C
- For optimistic, biased towards assigning B or C

Weighted Sum Model: Utility, Weights, Thresholds

- **Utility Function:** Normalization

$$u_i(x) = \frac{\max(x) - x}{\max(x) - \min(x)}$$

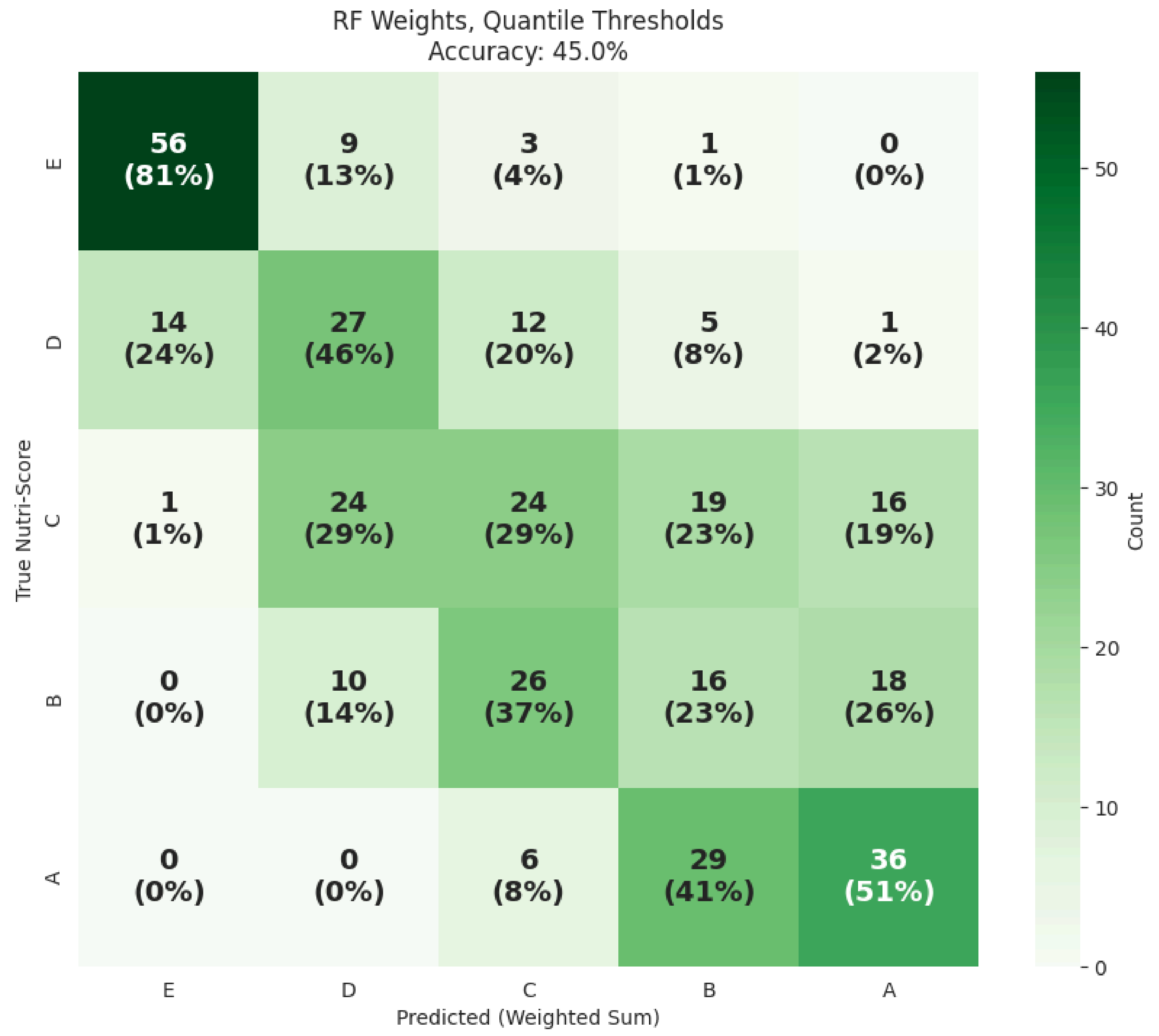
To Maximize: Proteins, Fiber, Fruit Veg Percentage, Green-Score.

$$u_i(x) = \frac{x - \min(x)}{\max(x) - \min(x)}$$

To Minimize: Energy, Saturated Fat, Sugars, Salt.

- **Weights**
  - **Correlation-Based:** Weights derived from correlation with nutri-score labels
  - **Random Forest:** Weights derived from RF feature importances
- **Thresholds**
  - **Kmeans:** Find 5 groups and sets thresholds at the midpoints between cluster centers
  - **'Quantile:** Divides the data into 5 equal-sized groups (20% each)

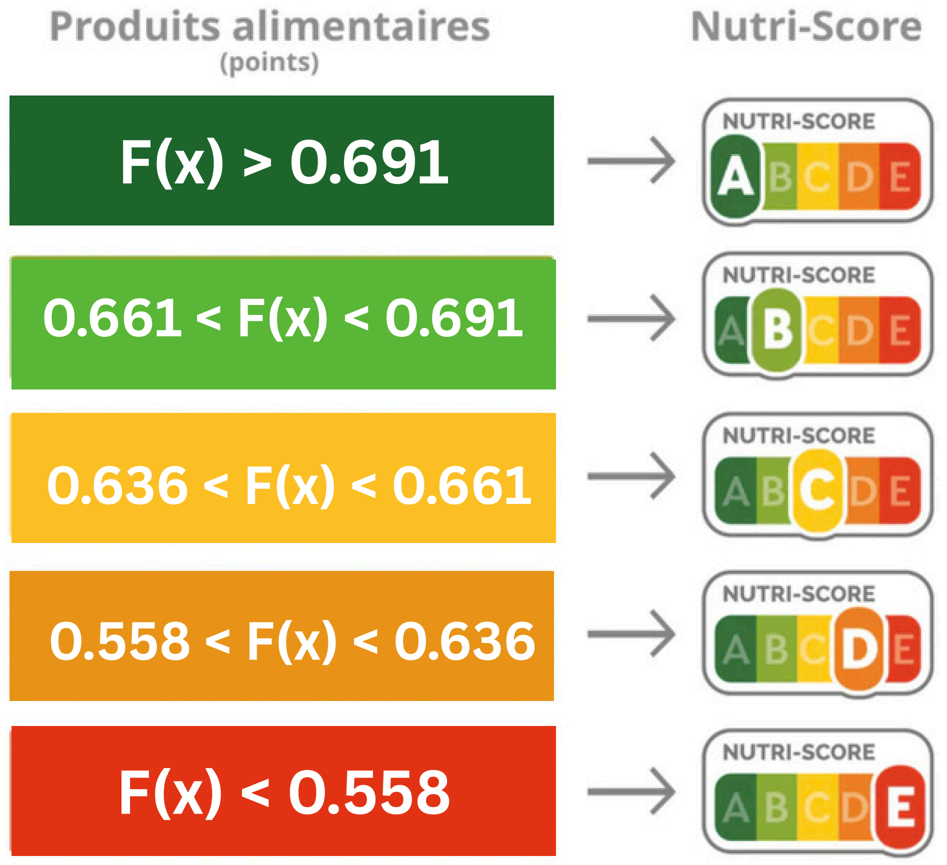
Weighted Sum Model: RF



RANDOM FOREST FEATURE IMPORTANCES

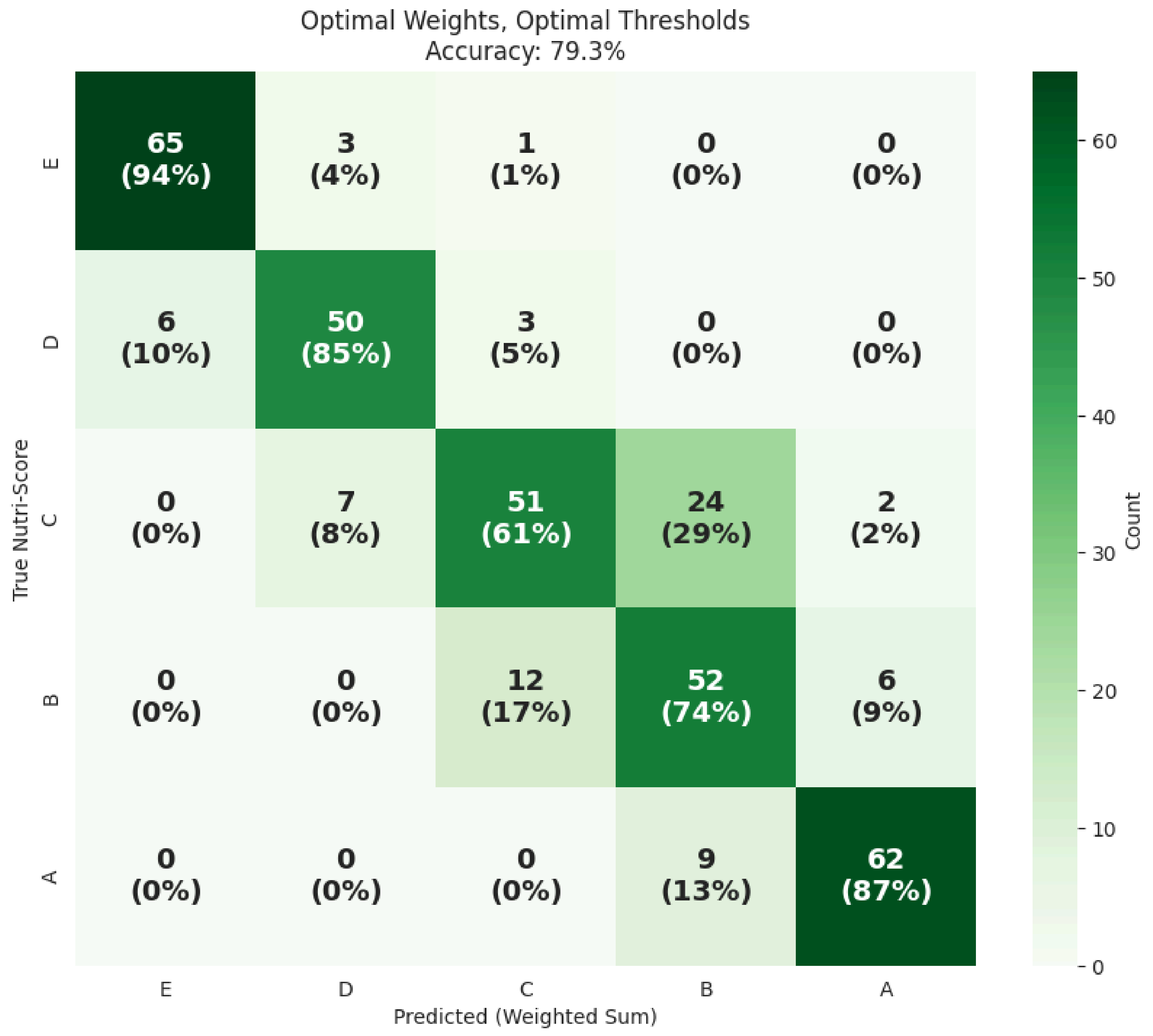
Feature Importance Ranking:

Criterion	Importance	Weight	Weight %
salt_100g	0.170357	0.170357	17.035727
sugars_100g	0.164154	0.164154	16.415438
energy_100g	0.149664	0.149664	14.966379
saturated_fat_100g	0.136880	0.136880	13.688010
proteins_100g	0.113533	0.113533	11.353349
fiber_100g	0.096741	0.096741	9.674132
fvl_percent	0.089330	0.089330	8.933029
green_score_value	0.079339	0.079339	7.933936



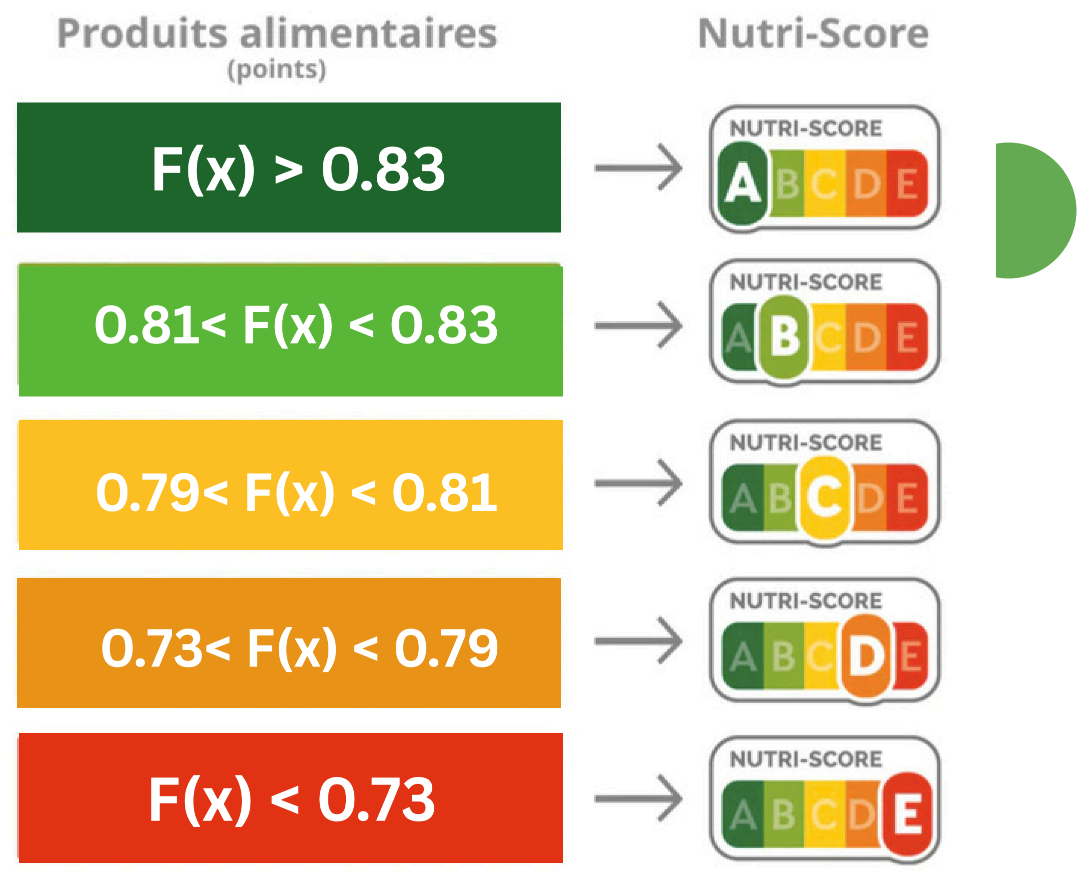
$$F(x) = 0.170357u\text{salt}(x) + 0.164154u\text{sugars}(x) + 0.149664u\text{energy}(x) + 0.136880u\text{saturated\_fat}(x) + 0.113533u\text{proteins}(x) + 0.096741u\text{fiber}(x) + 0.089330u\text{fvl\_percent}(x) + 0.079339u\text{green\_score\_value}(x)$$

Weighted Sum Model: Optimization



DIFFERENTIAL EVOLUTION DERIVED WEIGHTS

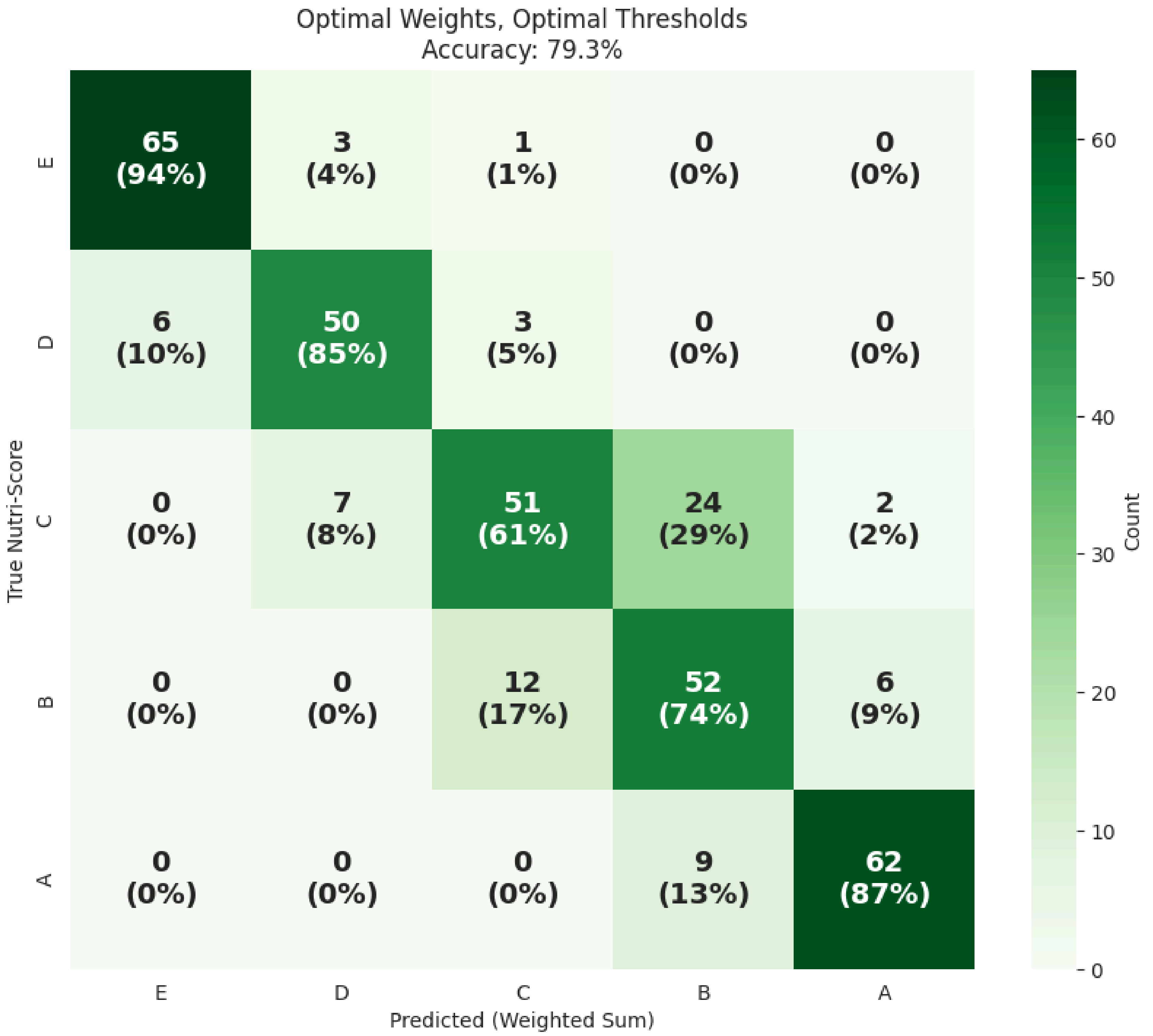
```
{'energy_100g': np.float64(0.06848415607702134),
 'saturated_fat_100g': np.float64(0.2128588362840498),
 'sugars_100g': np.float64(0.13824387475254973),
 'salt_100g': np.float64(0.4088220911907629),
 'proteins_100g': np.float64(0.07575266482185546),
 'fiber_100g': np.float64(0.07102152798817198),
 'fvl_percent': np.float64(0.017393587050708943),
 'green_score_value': np.float64(0.007423261834879819)}
```



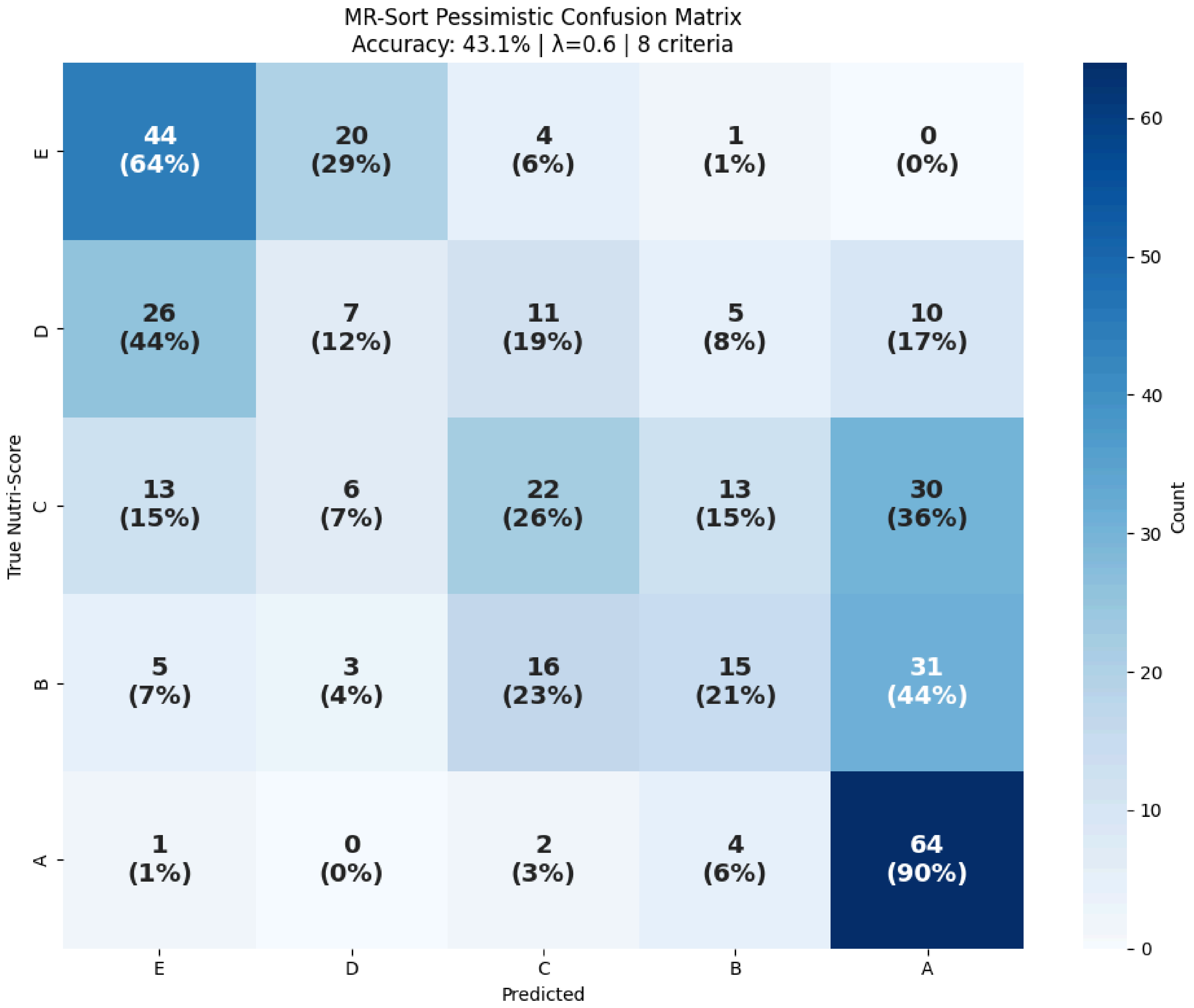
$$F(x) = 0.41u_{salt}(x) + 0.14u_{sugars}(x) + 0.07u_{energy}(x) + 0.21u_{saturated\_fat}(x) + 0.08u_{proteins}(x) + 0.07u_{fiber}(x) + 0.02u_{fvl\_percent}(x) + 0.007u_{green\_score\_value}(x)$$

Weighted Sum Model vs ELECTRE-TRI

Weighted Sum



ELECTRE-TRI



Machine Learning: Approximating the Decision Model

- **Input Features (X):** 7 Standard Nutritional Values + Green-Score Value (0–100).
- **Target (Y):** The final classification (A–E) assigned by the Weighted Sum formula.
- **Cross-Validation Strategy:** 5-fold Stratified K-Fold (ensures each fold has similar class distribution)
- **Normalization method:** Min-max scaling (0–1 range)

Machine Learning Models

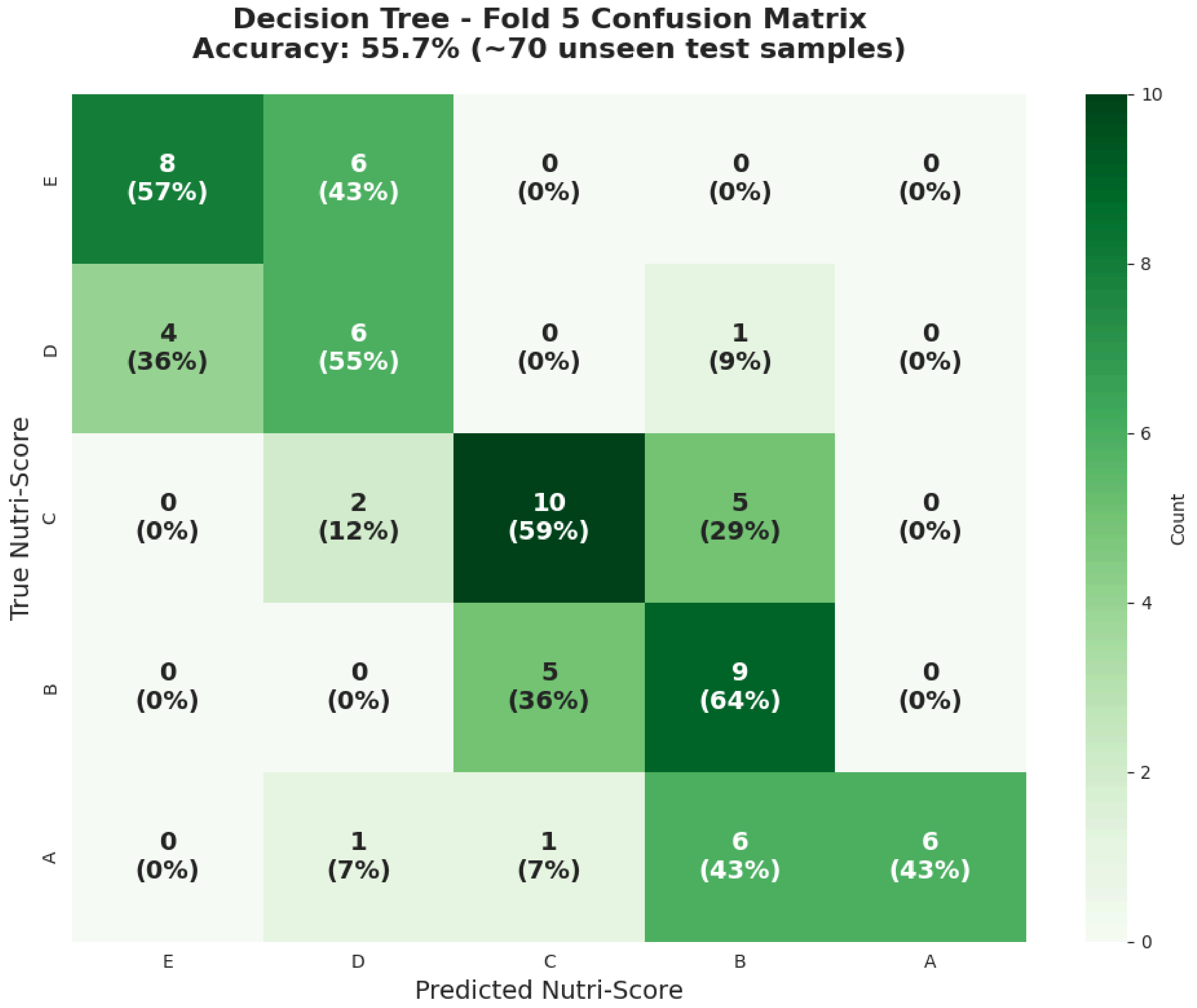
Decision Tree

Setup:

- Max depth ➔ 10, min_samples_split ➔ 10, min_samples_leaf ➔ 5

Insights:

- A single tree struggles to model transitions between Nutri-Score levels, as shown with some noticeable mispredicted classes, especially in the middle



Machine Learning Models

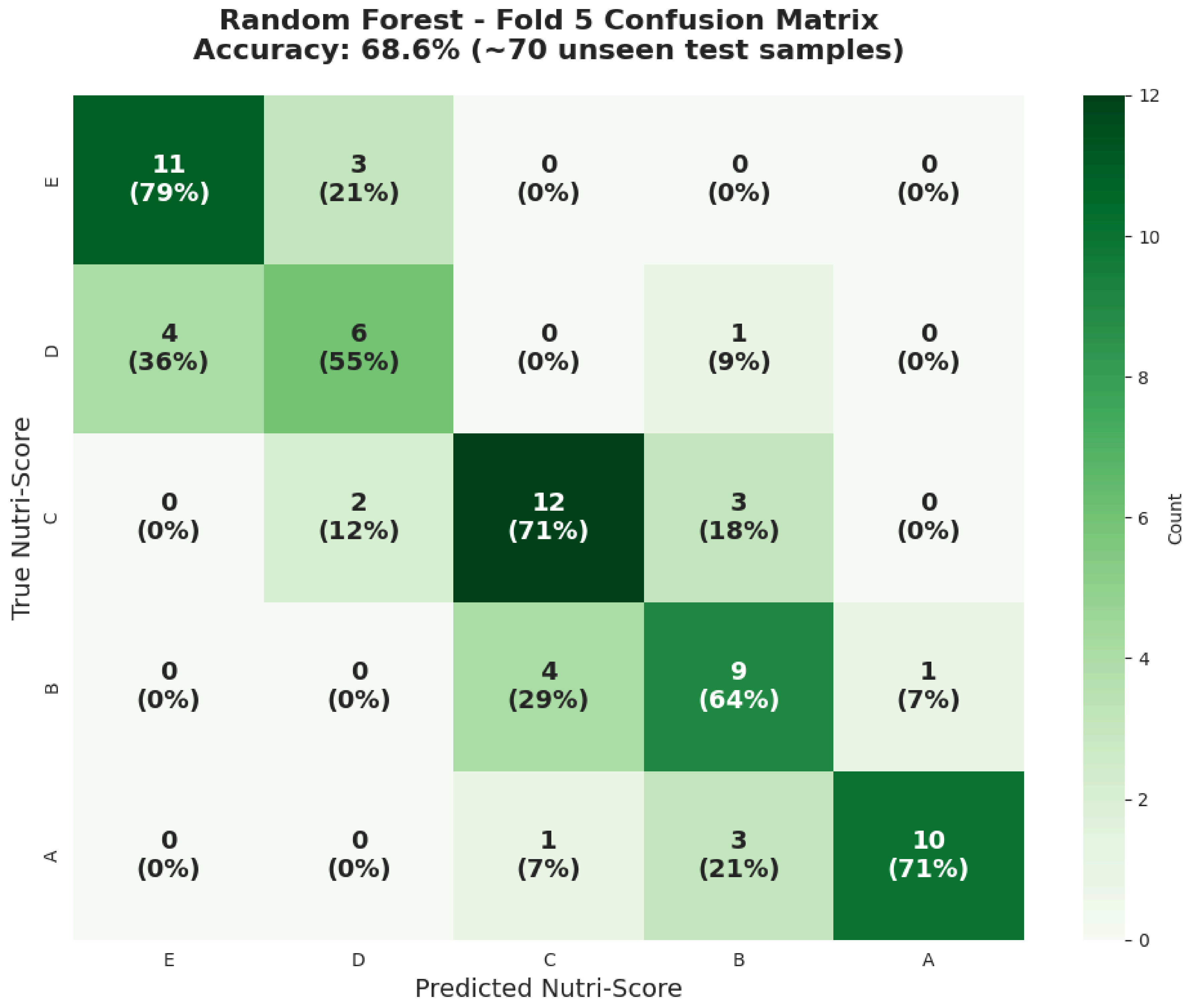
Random Forest

Setup:

- n_estimators ➔ 200, max depth ➔ 15, min_samples_split ➔ 5, min_samples_leaf ➔ 2

Insights:

- It's an improvement over Decision Tree!
- Model correctly classifies most E, C, B, A products, but D vs E and B vs C are often confused.
- Still fails to cleanly separate the middle classes



Machine Learning Models

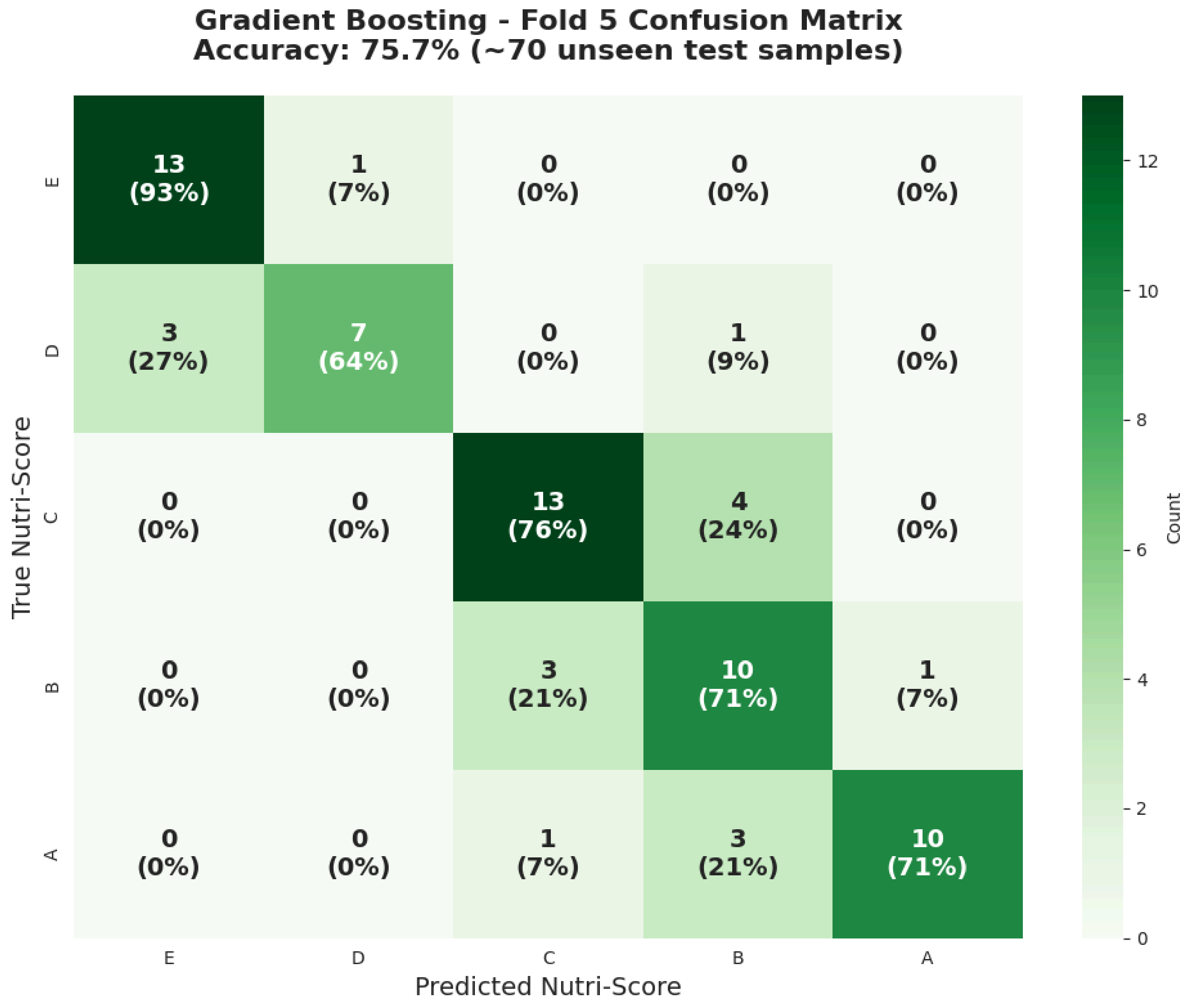
Gradient Boosting

Setup:

- n_estimators ➔ 100, learning_rate ➔ 0.1, max depth ➔ 5

Insights:

- Clear improvement over Random Forest over all classes
- There is still some ambiguity between neighbos (e.g. class D sometimes predicted as E or B)



Machine Learning Models

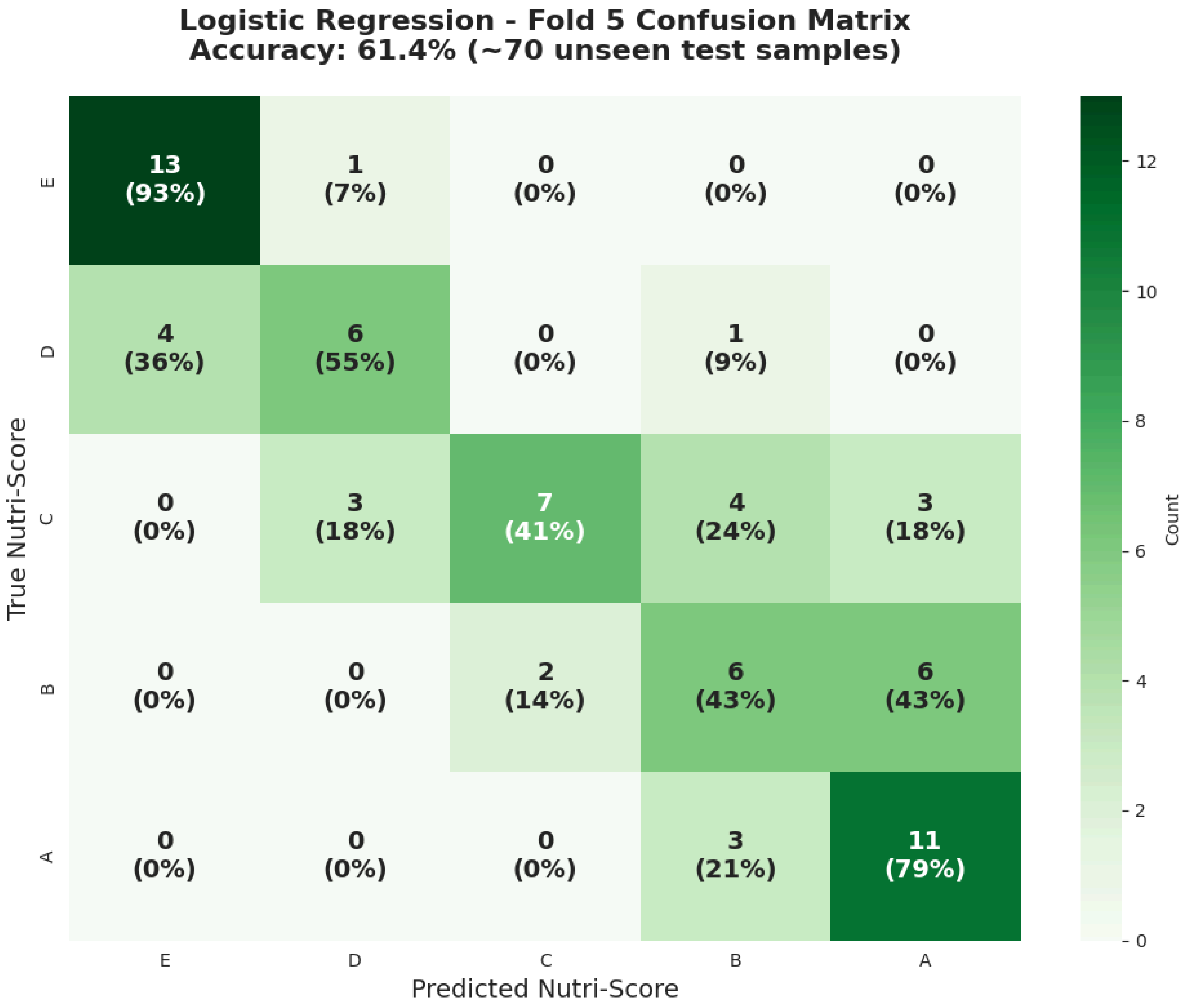
Logistic Regression

Setup:

- Solver ➔ lbfgs, C ➔ 1.0, max_iter ➔ 500

Insights:

- Predicts extreme classes E and A quite well
- Middle classes are often confused with each other and with neighboring classes
- Linear decision boundaries are not rich enough to fully separate intermediate Nutri-Score levels.



Machine Learning Models

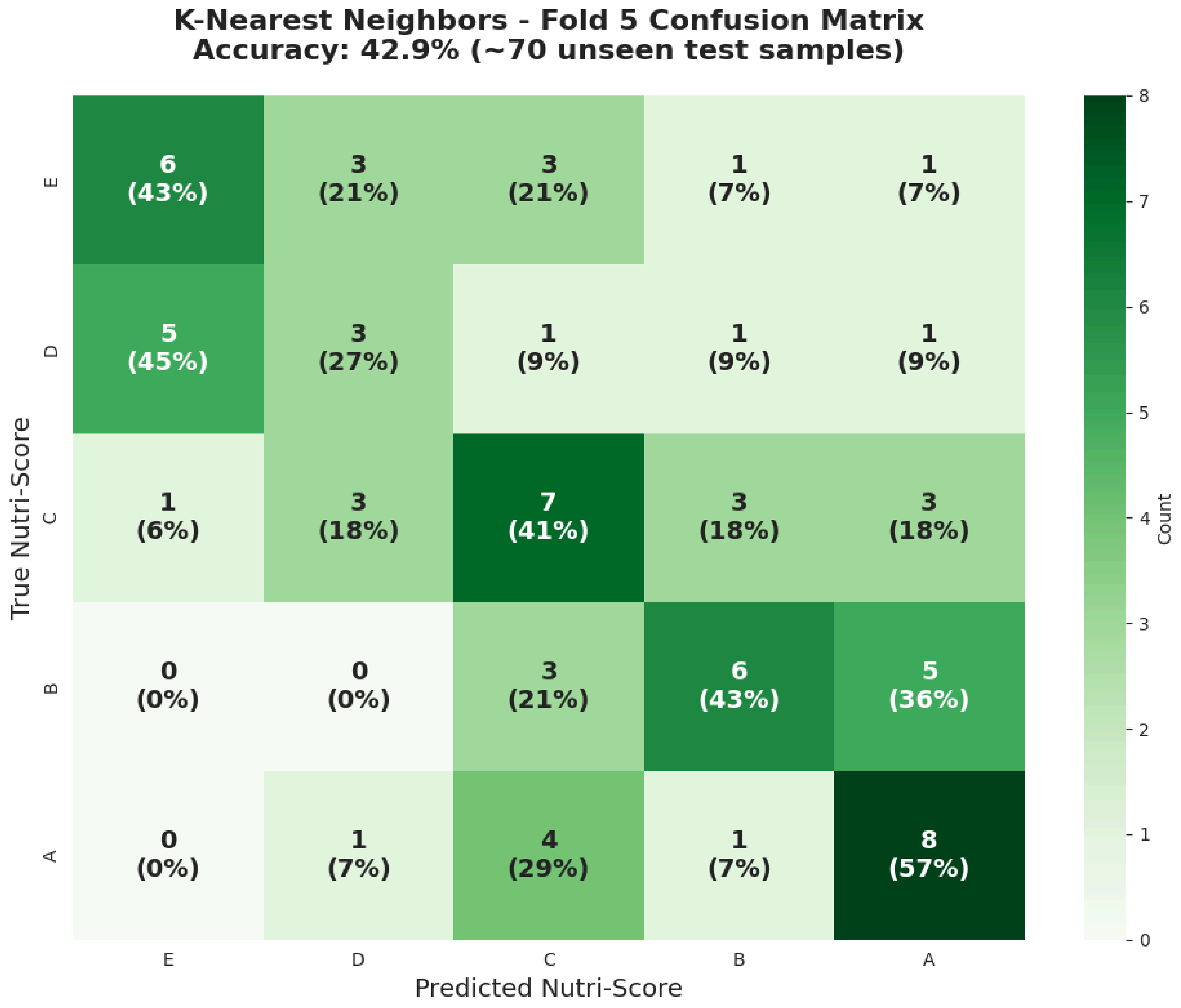
K-Nearest Neighbors

Setup:

- n_neighbors ➔ 7, weights ➔ distance, metric ➔ minkowski

Insights:

- Confusion between neighboring classes (E/D and C/B/A), showing KNN cannot clearly separate Nutri-Score levels on this data.
- Only class A is moderately well predicted; all other classes are heavily mixed, especially C and B.



Machine Learning Models

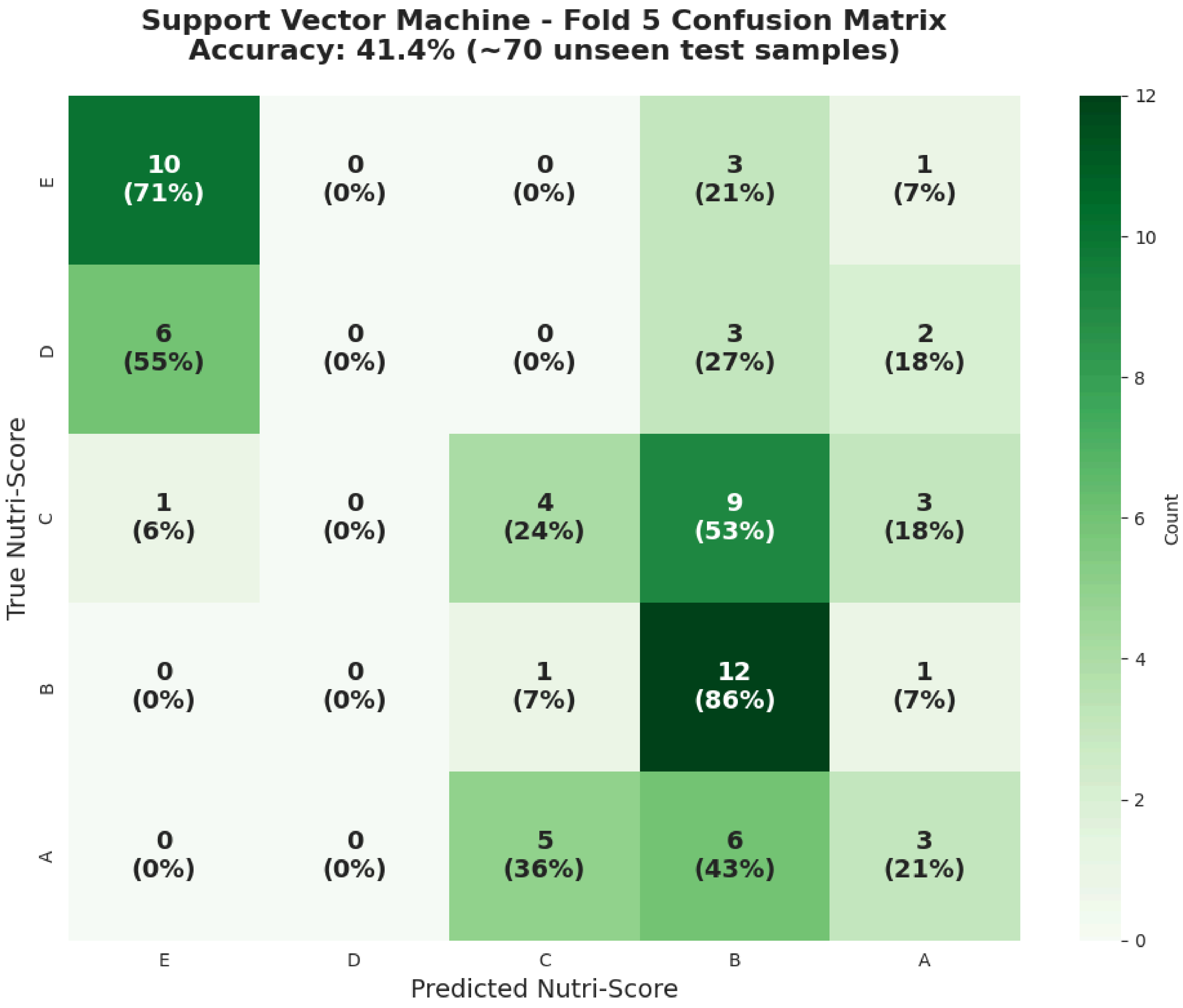
Support Vector Machine

Setup:

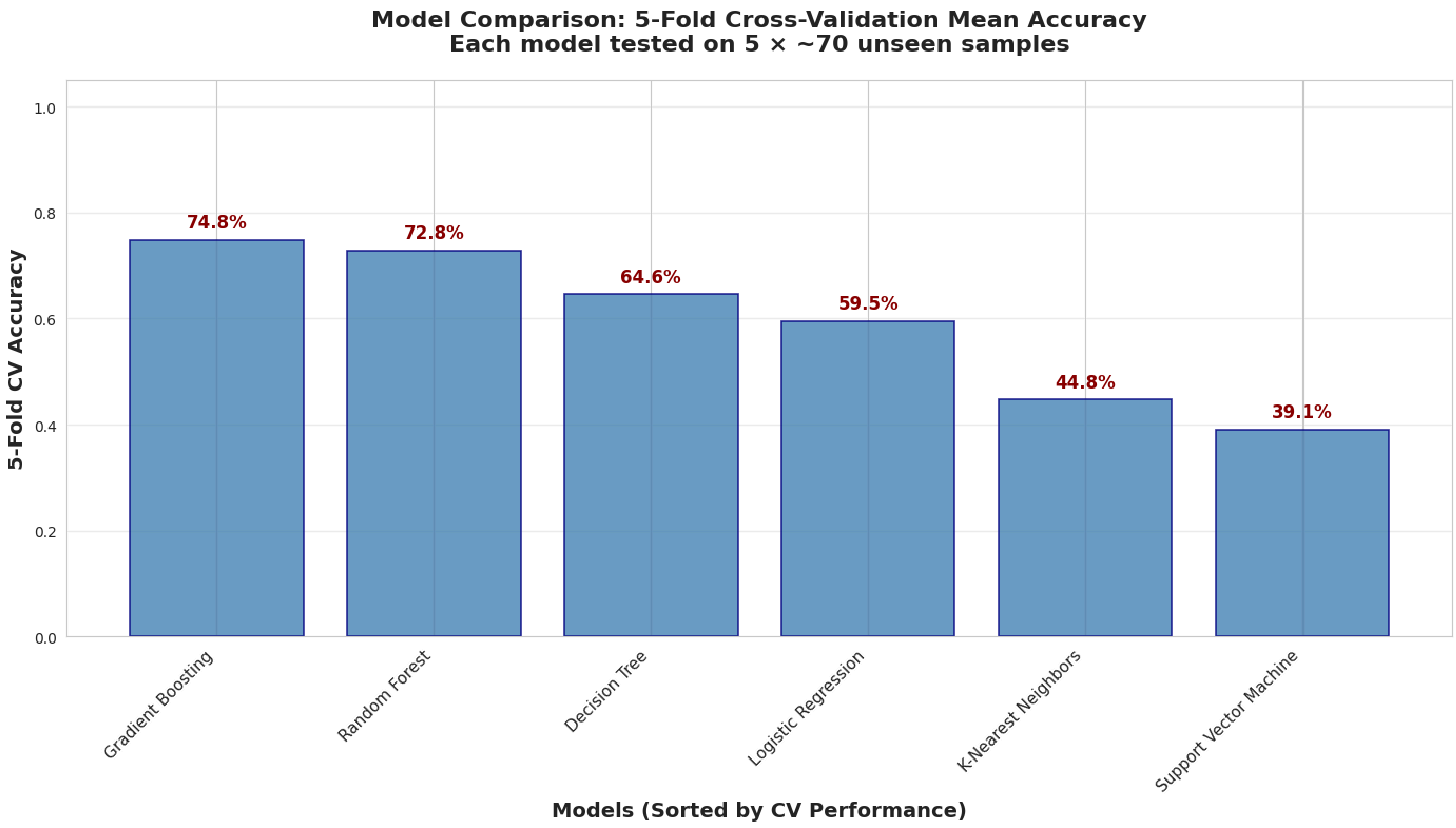
- C ➔ 10, gamma ➔ 'scale'

Insights:

- Strong confusion between neighboring classes, especially D vs E and A/C/B, so the margin does not separate middle Nutri-Score levels well.
- Model is relatively good at keeping B products as B, but A, C, and D are often misclassified into adjacent classes

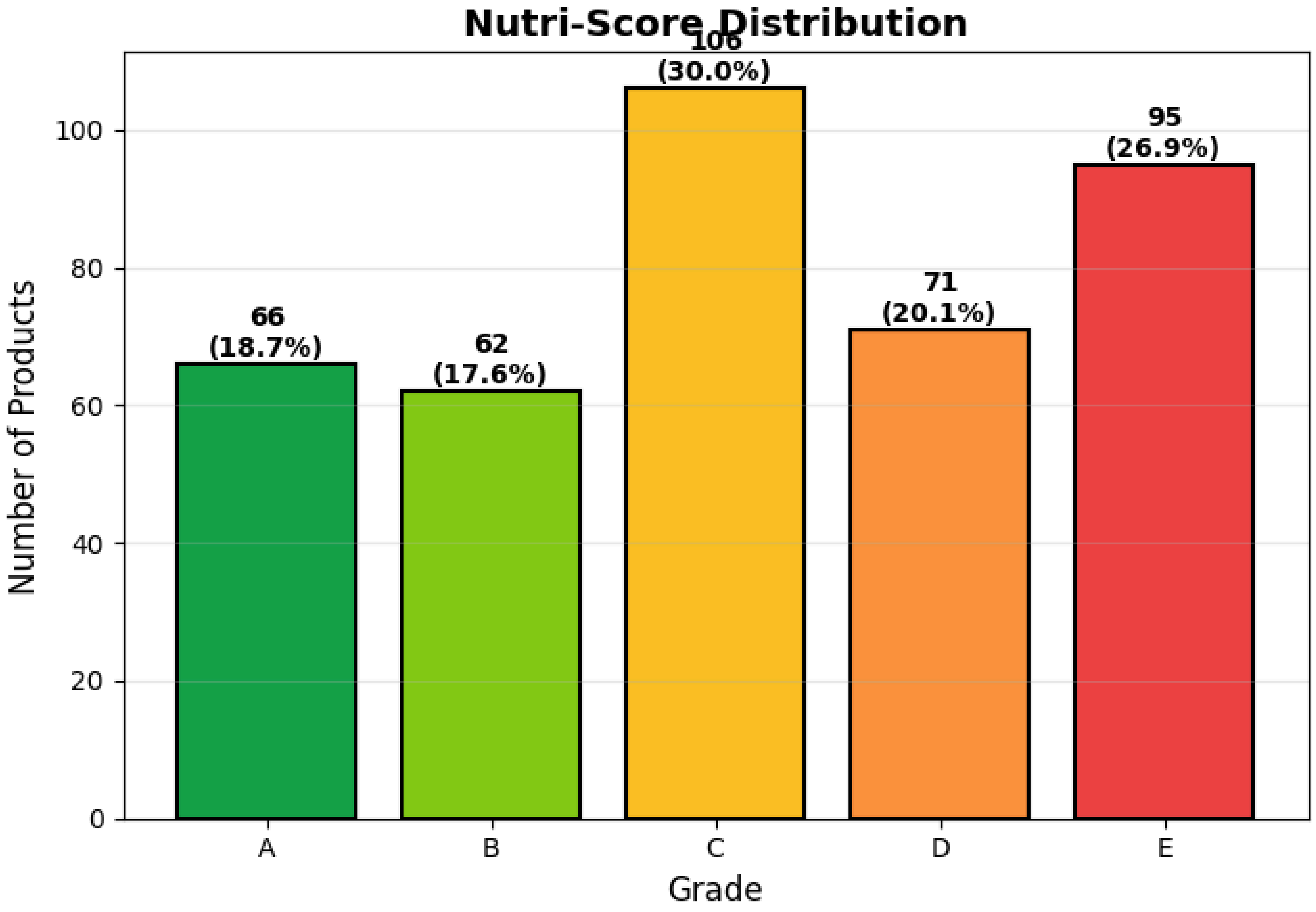


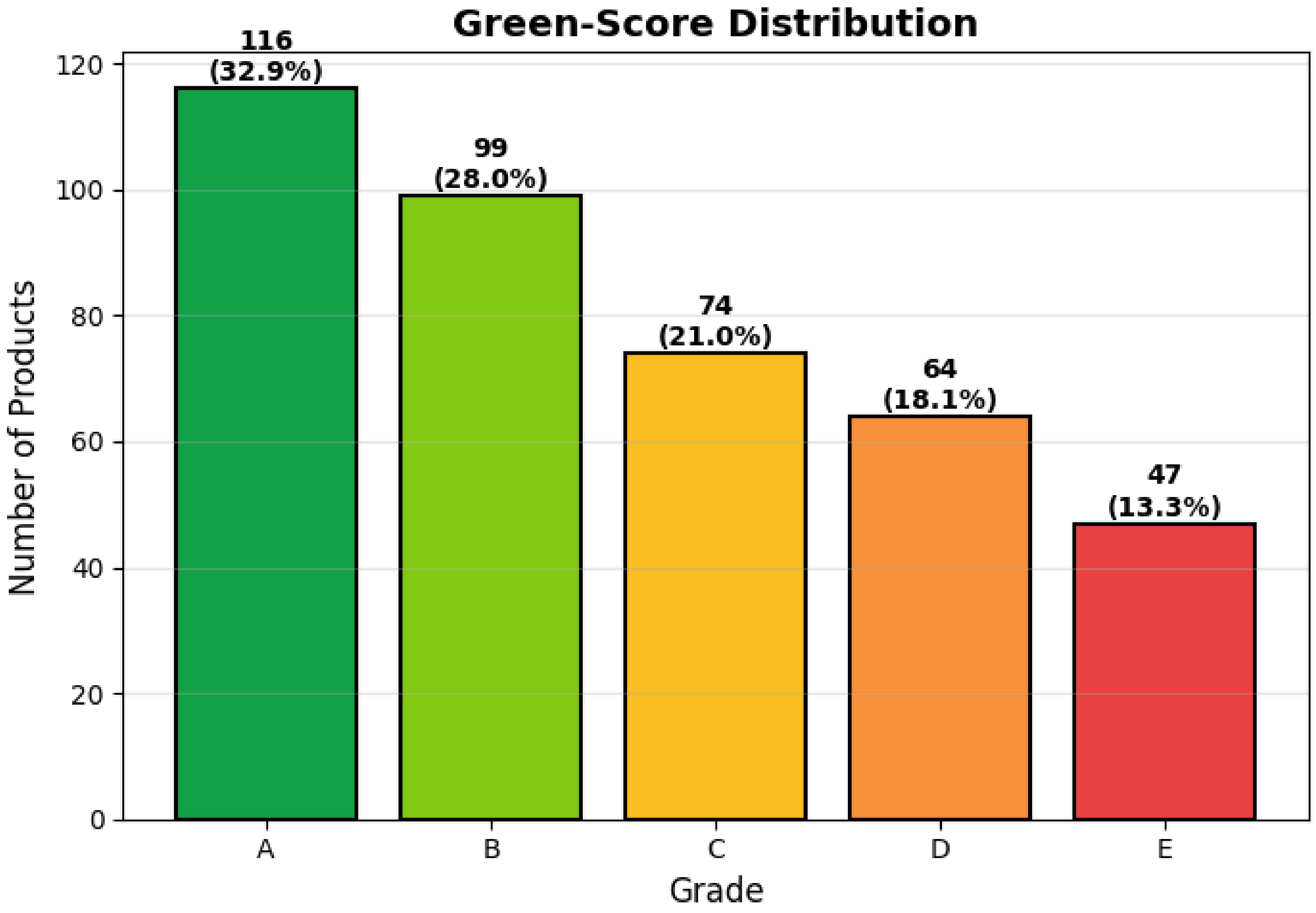
Machine Learning Models: Comparison



Who will win?!

Distribution of the data provided by the other group

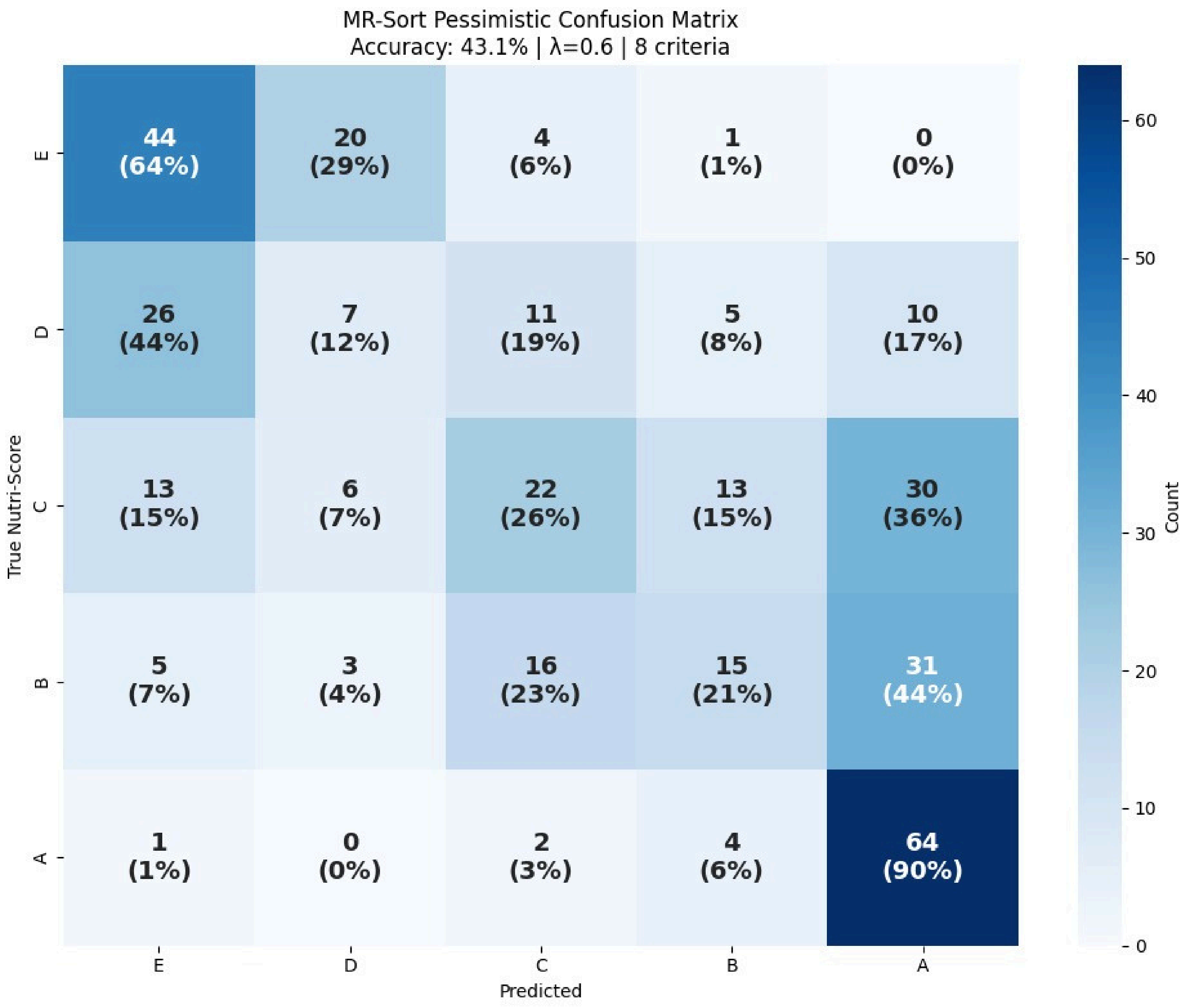




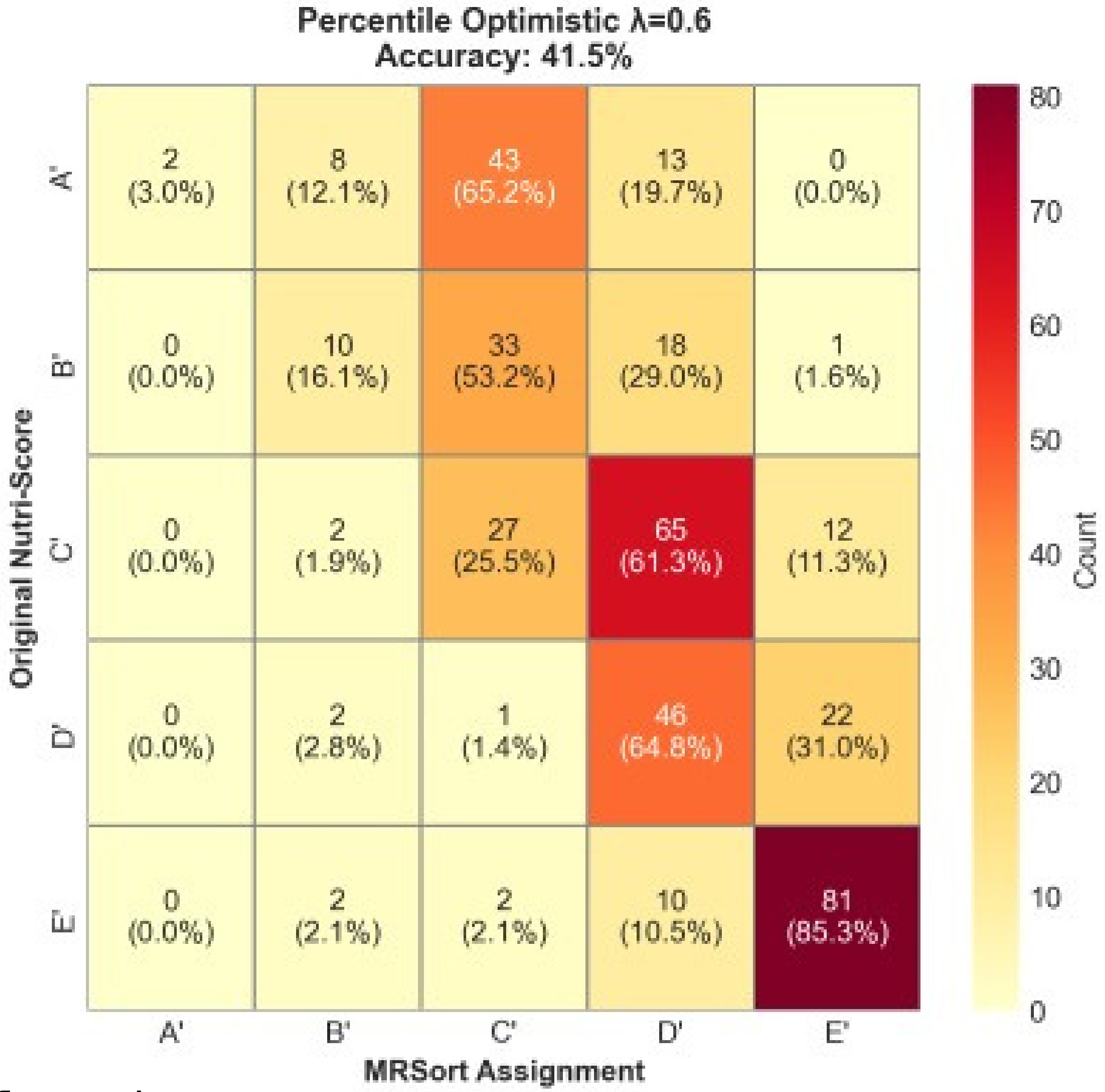
Who will win?!

<u>These are the results with the data provided by the other group</u>

Our ELECTRE-TRI Model



Other Group's ELECTRE-TRI Model



We have slightly higher accuracy overall, but noticeably, the other group performed better for E and D classes, while we did well for A and E