Data Mining Project

Total points: 100

Objective

The US food supply is dominated by ultra-processed foods. The purpose of this project is to implement **binary classifiers to identify food items that are not ultra-processed** (positive class, 1). It is recommended to use Python. If you wish to use another language, your team needs to agree, and you also need to let the instructor know.

Problem Definition

Goal: Build binary classifiers that identify food items that are not ultra-processed.

- *Positive class (1):* non–ultra-processed foods (labels = 0, 1, and 2, i.e. everything except ultra-processed)
- *Negative class (0)*: ultra-processed (label = 3).

Dataset

The dataset includes a number of food items, and some features that describe them. At this point, you will be provided with one file which you can use however you see fit to build your classifiers. At a later point, we will also share a test file, that has identical structure as the first file.
The columns/features are the following:

1. original_ID
2. name
3. store
4. food category
5. brand
6. **f_FPro_class** -> class label; takes values [0,1,2,3], where 0 corresponds to raw products, 3 corresponds to ultra-processed foods.
7. price
8. price per cal
9. package_weight
10. Protein
11. Total Fat
12. Carbohydrate
13. Sugars, total
14. Fiber, total dietary
15. Calcium
16. Iron
17. Sodium
18. Cholesterol
19. Fatty acids, total saturated

1. Exploratory Data Analysis (EDA)

First, you need to start with EDA. The purpose is to understand the type of data features available and their characteristics.

1. 1 Data Structure

Inspect data types, the values that each attribute might take, and any missing values. Identify categorical vs. numerical features.

1.2 Target Distribution

Analyze the distribution of f_FPro_class and your binary mapping (Ultra processed vs rest). Check for class imbalance.

1.3 Descriptive Statistics

Explore the distribution of the attribute values for each feature. Explore, overall and by binary class label, the summary statistics of each feature. You may use boxplots or histograms to show their distributions.

2. Data Preprocessing

Feel free to process the data as you see fit to build a reliable and accurate prediction model.

2.1 Target Construction

Convert f_FPro_class → binary (Ultra processed or not).

2.2 Feature Creation & Selection

Drop any unnecessary features. Deal with any missing values. If needed, encode any high cardinality categorical features or extract text features (e.g., bag-of-words). Use one-hot encoding as needed. Normalize features as needed.

2.3 Handling Class Imbalance

Explore ways to handle class imbalances (oversampling, undersampling, SMOTE, etc.)

3. Classification models

You need to build the following four models:

(1) First, you need to come up with a simple **baseline classifier** that would serve as the baseline for your comparison. You can decide what example this will be. The important thing to remember here is that this **baseline model** will not actually "learn" a model. You can base your decisions on average, popularity, randomness, etc.

(2) Build a **Decision Tree** model.
(3) Build a **Random Forest** model.
(4) Build any other model of your choice.

Model Selection & Evaluation

Split the data into three (stratified) parts: training, validation, and test sets. Set the random seed to your TeamID, in order to replicate your results as needed. Identify the key hyperparameters for each model. For each hyperparameter, try at least 3 values. Build the models with the training set, perform model selection (hyperparameter tuning) with the validation set, and model evaluation (comparison) with the test set **using the F1 score**.

Evaluation metrics

Report test Accuracy, Precision, Recall, F1 score, and AUC for the best combination of hyperparameters for each classifier. Also, report F1-score and AUC for training/validation/test set.

4. Outlier Detection

Use clustering to identify foods' Nutritional Profiles in the **training set**. Use the Elbow Method to plot the number of clusters (x-axis) and the Within-Cluster Sum of Squares (y-axis) to select the number of clusters. Rank the foods with respect to the distance to their centroids. Remove the top 5% data points with the highest distances.
Rerun the four classifiers with the updated training set and examine how removing the "outlier-like" points affects the performance of the model in the test set.

Notes

- All filenames submitted or generated by your code should be starting with "`t09`", where the two digits correspond to the team ID.
- Your code should have clear comments to help us understand your logic and flow.
- You can use the scikit-learn package to build your models.
- Please, list all the sources that you will use. You can even use LLMs, but only if you properly mention how you used them. Also, keep in mind, that in the end of the day, each member of the team will be responsible for their code.
- When you are comparing models, the metrics should be computed over the same data, i.e., train/validation/test data should be the same across models. So, remember to properly use seeds (and the "`random_state`" on the models) and verify that the data splits are the same.
- Your code should be reproducible.
- If you are not following the guidelines, we might have to subtract points from your submission.