What we can do for Food dataset:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task category | y | Possible X | Algos | Challenges | Notes |
| Classification (one-label, multi-class) | main\_category\_en | `product\_name`, `brands`, `categories`, `ingredients\_text`, `allergens`, `ingredients\_from\_palm\_oil`, `nutrition\_grade\_uk`, `nutrition\_grade\_fr`, etc. | Decision Trees, Random Forest, Gradient Boosting (e.g., XGBoost, LightGBM), or Neural Networks for more complex relationships. | (1) weird entries: fr:Succedanes-de-cafe, pl:Odwodnione-zupa, pl:Zupa (2) imbalanced data categories |  |
| Text Classification (low, med, high) | calcium\_100g | ingredients\_text | Text classification (e.g., TF-IDF + Logistic Regression), Named Entity Recognition (NER) models (e.g., spaCy, Hugging Face transformers) |  | Fish, green vegetables, diaries, bones may indicate this |
| Classification (multi-label, multi-class) | countries\_tags |  |  | (1) imbalanced dataset; (2) multi-label? |  |
| Classification (binary) | allergens | `ingredients\_text`, `ingredients\_from\_palm\_oil`, `additives`, `categories`, etc. | Classification algorithms like Logistic Regression, Support Vector Machine (SVM), Random Forest, or Neural Networks. | (1) No entry in “allergens” does not necessarily mean “no allergens”. It could mean a missing data.  (2) Based on the previous, imbalanced dataset. |  |
| Regression | nutrition-score-uk\_100g | `product\_name`, `ingredients\_text`, `categories`, `brands`, etc. | Linear Regression, Random Forest Regressor, XGBoost, or Neural Networks (e.g., MLP Regressor). | (1) validation: negative values? |  |
| Regression | energy\_100g | some nutritions etc. | Linear Regression, Random Forest Regressor, XGBoost, or Neural Networks (e.g., MLP Regressor). |  |  |
| Regression (inference) | the relationship between sugars\_100g, nutrition-score-uk\_100g |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| Clustering | products |  | K-means Clustering, DBSCAN, Hierarchical Clustering. |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**1. Prediction Tasks (Supervised Learning)**

**a) Product Classification (Categorization)**

- Goal: Predict the **product's main category**, sub-category, or nutrition grade based on the attributes of the product.

- Columns to use: `product\_name`, `brands`, `categories`, `ingredients\_text`, `allergens`, `ingredients\_from\_palm\_oil`, `nutrition\_grade\_uk`, `nutrition\_grade\_fr`, etc.

- Algorithms: Decision Trees, Random Forest, Gradient Boosting (e.g., XGBoost, LightGBM), or Neural Networks for more complex relationships.

Challenges:

(1) weird entries: fr:Succedanes-de-cafe, pl:Odwodnione-zupa, pl:Zupa

(2) imbalanced data categories

b) **Nutrition Prediction (Regression)**

- Goal: Predict numeric values for nutritional metrics such as energy content, sugars, carbohydrates, fat, protein levels based on other features.

- Columns to use: `product\_name`, `ingredients\_text`, `categories`, `brands`, etc.

- Algorithms: Linear Regression, Random Forest Regressor, XGBoost, or Neural Networks (e.g., MLP Regressor).

d) Allergen/Ingredient Risk Classification (binary classification)

- Goal: Classify products based on whether they contain specific allergens (e.g., nuts, dairy, etc.) or harmful ingredients.

- Columns to use: `allergens`, `ingredients\_text`, `ingredients\_from\_palm\_oil`, `additives`, `categories`, etc.

- Algorithms: Classification algorithms like Logistic Regression, Support Vector Machine (SVM), Random Forest, or Neural Networks.

Challenges:

(1) No entry in “allergens” does not necessarily mean “no allergens”. It could mean a missing data.

(2) Based on the previous, imbalanced dataset.

2. Clustering (Unsupervised Learning)

a) Product Segmentation

- Goal: Group products into different clusters based on similarities in features like ingredients, nutritional content, and packaging.

- Columns to use: `ingredients\_text`, `brands`, `categories`, `nutrition\_grade\_uk`, `energy\_100g`, `fat\_100g`, `carbohydrates\_100g`, etc.

- Algorithms: K-means Clustering, DBSCAN, Hierarchical Clustering.

b) Customer Segmentation

- Goal: If you have customer data or purchase data, you could segment consumers based on their purchasing behavior or preferences for certain product types.

- Columns to use: `purchase\_places`, `countries`, `cities`, `ingredients\_text`, etc.

- Algorithms: K-means, DBSCAN, or Gaussian Mixture Models.

3. Natural Language Processing (NLP)

a) Ingredient/Allergen Extraction

- Goal: Extract relevant information about ingredients, allergens, or nutritional content from the `ingredients\_text` field using text classification or named entity recognition (NER).

- Columns to use: `ingredients\_text`, `allergens`, `additives`.

- Algorithms: Text classification (e.g., TF-IDF + Logistic Regression), Named Entity Recognition (NER) models (e.g., spaCy, Hugging Face transformers).

Not use:

b) Text Generation (for automatic labeling)

- Goal: Generate product descriptions, allergen labels, or nutritional details based on structured data (like ingredients or categories).

- Columns to use: `ingredients\_text`, `categories`, `product\_name`, `brands`, etc.

- Algorithms: Pretrained language models (e.g., GPT, BERT) fine-tuned on product-specific data.

4. Anomaly Detection

a) Data Quality Issues

- Goal: Identify outliers or inconsistencies in data, such as missing values, strange nutritional information, or conflicting product details.

- Columns to use: All columns, especially those related to nutritional values, allergens, and ingredients.

- Algorithms: Isolation Forest, One-Class SVM, Autoencoders (for deep learning-based anomaly detection).

5. Feature Engineering and Data Preprocessing

- Text Preprocessing for Ingredients & Product Names: Clean and preprocess the `ingredients\_text` and `product\_name` columns using tokenization, stemming/lemmatization, and vectorization (e.g., TF-IDF or word embeddings like Word2Vec or BERT).

- Handling Missing Values: For missing values in numeric columns (e.g., nutrition facts), consider imputation strategies (mean, median, or more sophisticated models like KNN imputation).

- Encoding Categorical Variables: For columns like `categories`, `brands`, `countries`, and `origins`, apply one-hot encoding, label encoding, or use embeddings for high-cardinality features.

- Creating Interaction Features: Create interaction terms between various columns, such as between `ingredients\_text` and `nutrition\_grade\_uk`, or `categories` and `allergens`.

- Scaling/Normalization: Numeric features like `fat\_100g`, `sugar\_100g`, etc., may need scaling (e.g., MinMax scaling or Standardization) depending on the model you're using.

- Time-Based Features: For `created\_t`, `last\_modified\_t`, you can extract features such as the day of the week, month, or year, to potentially uncover patterns in product creation or updates over time.

6. Model Evaluation & Tuning

- Cross-Validation: Use cross-validation (e.g., k-fold cross-validation) to ensure the model's generalizability.

- Hyperparameter Tuning: Perform hyperparameter optimization with techniques like Grid Search or Randomized Search to find the best model parameters.

- Performance Metrics: Depending on the problem (regression vs. classification), evaluate models with metrics like Accuracy, Precision, Recall, F1-score (classification) or Mean Squared Error (MSE), R-squared (regression).

---

Example ML Task Breakdown

1. Predicting Nutrition Grade (Classification)

Target: `nutrition\_grade\_uk` (or `nutrition\_grade\_fr`)

Features: `ingredients\_text`, `product\_name`, `categories`, `brands`, `packaging`, `additives`, etc.

Algorithm: Random Forest or XGBoost

Steps:

- Clean and preprocess the text columns (e.g., `ingredients\_text`).

- Encode categorical columns (`brands`, `categories`, `countries`).

- Train and evaluate the model on a balanced dataset.

2. Clustering Products Based on Ingredients (Clustering)

Target: No specific target (unsupervised)

Features: `ingredients\_text`, `categories`, `brands`, `labels`, `allergens`, etc.

Algorithm: K-means or DBSCAN

Steps:

- Preprocess and vectorize the text data (`ingredients\_text`).

- Normalize numerical features (`energy\_100g`, `fat\_100g`, etc.).

- Apply clustering and evaluate with silhouette score or similar metrics.