NutriClass: Food Classification using Nutritional data

Introduction:

The aim of this project is to provide end-users insights of the given dataset of nutrition information for decision making on food classification.

Data set analysis:

The data set provided contains a single spreadsheet. The spreadsheet is totally of 31,700 rows and 16 columns. While most of the columns are numerical variables, few categorical variables are also seen. The datatypes of the rows and columns as present in the are listed in Table 1.

Upon initial analysis, of the 31,325 values are present and 375 missing values are found on each numerical variables columns, evidently a need for handling missing numerical values.

Statistical analysis reveal that central tendencies(mean, median) of most of the columns. The spread and standard deviation is large for almost every feature, indicating evident variation between the foods. The gap between Max values and Min values are high indicating strong skew/outliners within.

The diversity in features points a need for well planned feature scaling, pre-processing and possible outliner handling.

Data Preprocessing:

Handling of Missing values: Missing ‘NaN’ values in numerical column are identified and filled with the median of each respective column.

Outliers capping: For each column with outliers, the 1st percentile value and the 99th value are identified. Any value above or below are capped at the respective percentile,, thus eliminating the outliers without altering the sample size.

Duplicates removals: All rows with identical values on all columns are identified and removed. A total of 313 rows are removed.

Normalisation: Standardisation of values so that techniques like PCA., works efficiently. Rescales all the numerical features so that they have a mean of 0 and standard deviation of 1.

Principle component analysis: All numerical columns are 2D PCA and plotted in the scatterplot and analysed. The results have both distinct cluster and overlapping clusters, suggesting reduction of the number of features for effective training and test of models. Another PCA for modelling is done with n_components=10. This step is done after encoding that standardised text features to numerical values.

Encoding: Conversion of all the text-based categories into numerics. LabelEncoder is used for ‘Food_Name’ and one hot encoding for every other text categorical columns.

Model selection and prediction results :

The prepared data is run on the different models (Logistic Regression,Decision tree, Random forest, KNN, SVM, XGBoost and Gradient Boosting), in order to evaluate each model.

This is done by looping a dictionary named ‘models’ that acts as a lineup of all the different algorithms. Each entry corresponds a different model. The code iterates through every entry(thus the model), by training and testing every model on the same train test spilt thus enabling direct comparison of performance. The data set is spilt into the following standard for the purpose

‘Train shape: (25109, 19) Test shape: (6278, 19) ‘

Once all the models are ran successfully, the results are saved in a separate folder for further analysis. Statical visualisations are also given for each model. This included a classification report with accuracy, precision, recall and F1-scores. A heat map of the confusion matrix of each model is also plotted for easier interpretation.

Model Evaluation and Insights :

All the results stored are synthesised into high-level summary tables and visualisation for easy comparison, looping through all the classification reports to extract all the metrics. Finally a summary table, a heat map and a bar chart are created for much deeper and more granular comparative analysis. Based on the analysis, the models are ranked accord to their performance Table 2.

Overall analysis shows that Support Vector Machine(SVM) is the best performing model for this food classification task, demonstrating the highest accuracy and best balance of precision and recall.

The models Gradient Boosting, XGBoost and Random forest care not very far behind, proving to be highly effective alternatives.

While the Decision Tree model is the weakest performer, primary due to confusion between similar food classes.

The F1-score of each food category is extracted to plot a heat map. This map shows the ability of each model to correctly identify each specific class. This the better the F1-score the better the model predicts.

The models performance across food categories reveals that the most challenging prediction is between the food types ‘Burger’ and ‘Pizza’. Decision Tree struggled on this with a F1-score of 0.96 on both the classes.There was also a minor confusion between the fruit classes ‘Apple’, ‘Banana’,., where Logistic Regression shows a slight weakness.

On the aspects of model simplicity or training speed Gradient Boosting and XGBoost are excellent only marginally behind SVM in overall performance.

All the models are extremely strong, showing that the features are highly predictive for this dataset.The differences, while small, indicate a clear tiering:

SVM ≈ Gradient Boosting ≈ XGBoost ≈ Random Forest > KNN > Logistic Regression > Decision Tree.

Hence, for maximum accuracy SVM, Gradient boosting, XCBoost are optimal. Random forest is more feature oriented and suitable for large-scale data. While Logistic Regression and Decision tree can be used as a quick check because of their simplicity. A model-to-task comparison is given in Table 3.

Recommended improvements :

Hyperparameter tuning: Usage of GridsearchCV or RandomizedSearchCV can be used in SVM, XGBoost and Gradient Boosting for optimal performance.

Feature engineering tweaks: In-order to gain more domain knowledge-driven features interactions between nutrients and unsupervised clustering can be used.

Minority balance: Methods like SMOTE or class weighting can be used to pay more attention to minority classes in the dataset.

Along with the suggested improvisations, techniques like ensembling to combine top performing models and effective Post-processing can be added to improvise the performance to end users desired results.

Conclusion :

In this project, we successfully developed a robust machine learning system to classify food items based on their nutritional and preparation attributes.

The resulting classification system can be effectively integrated into smart dietary applications, nutritionist tools, educational platforms, and food logging or grocery apps—enabling automated, accurate, and scalable food identification to enhance user experience and support dietary management.

Tables used :

Columns	Datatype	Type
Calories	Float	Numerical
Protein	Float	Numerical
Fat	Float	Numerical
Carbs	Float	Numerical
Sugar	Float	Numerical
Fiber	Float	Numerical
Sodium	Float	Numerical
Cholesterol	Float	Numerical
Glycemic_Index	Float	Numerical
Water_Content	Float	Numerical
Serving_Size	Float	Numerical
Meal_type	Category	Categorical
Preparation_method	Category	Categorical
Is_Vegan	Boolean	Categorical
Is_Gluten_free	Boolean	Categorical
Food_Name	String	Text

Table 1

Model	Accuracy	Weighted precision	Weighted recall	Weighted F1 score
SVM	0.994425	0.994451	0.994425	0.994428
Gradient Boosting	0.993469	0.993489	0.993469	0.993469
XGBoost	0.992832	0.992837	0.992832	0.992831
Random Forest	0.992195	0.992204	0.992195	0.992198
Logistic regression	0.991080	0.991096	0.991080	0.991083
KNN	0.990602	0.990630	0.990602	0.990601
Decision Tree	0.985983	0.986020	0.985983	0.985988

Table 2

Model	Suitable tasks
SVM	Maximum accuracy, clean/medium-size data
XGBoost/ Gradient Boosting	Complex, tabular data, robust to noise/ outliers, production
Random forest	Quick results, good interpretability, large features
KNN	Data with well-separated clusters, few samples
Logistic Regression	Fast, interpretable, mostly linear separability
Decision tree	EDA, feature importance, need for decision paths

Table 3

System architecture :