Deep Leaning
Project

PROJECT REPORT

- 2022 / 2023 -

Written By

Hadil Helali
Soulaima Kahla
Raoua Trimech
Mouhamed Zied Brahmi

Github link

PROJECT STRUCTURE

01

data preprocessing

involves cleaning, formatting, and transforming the data into a suitable format for the model

02

data augmentation

to increase the size and diversity of a training dataset

03

data training and testing

splitting the dataset into training and testing then applying the ML Classifiers

04

model evaluation & comparison

involves calculating the accuracy of each model and then comparing them

01

DATA PROCESSING

DATASET

initially the dataset contained the following columns :

- id
- Region
- Area
- palmitic (*)
- palmitoleic (*)
- stearic (*)
- oleic (*)
- linoleic (*)
- linolenic (*)
- arachidic (*)
- eicosenoic (*)
- other

with **572 samples**

These fields (*) are all **fatty acids** typically found in olive oil :

Palmitic acid: This is a saturated fatty acid, meaning it has no double bonds in its carbon chain. It typically accounts for around 7.5-20% of the total fatty acid content in olive oil.

Palmitoleic acid: This is a monounsaturated fatty acid, accounting for around 0.3-3.5% of the total fatty acid content in olive oil.

Stearic acid: This is also a saturated fatty acid, accounting for around 0.5-5% of the total fatty acid content in olive oil.

Oleic acid: This is the most abundant fatty acid in olive oil, accounting for approximately 55-83% of the total fatty acid content. Oleic acid is a monounsaturated fatty acid, meaning it has one double bond in its carbon chain.

Linoleic acid: This is a polyunsaturated fatty acid, meaning it has two or more double bonds in its carbon chain. It typically accounts for around 3-21% of the total fatty acid content in olive oil.

Linolenic acid: This is another polyunsaturated fatty acid, accounting for around 0.2-1.5% of the total fatty acid content in olive oil.

Arachidic acid: This is a saturated fatty acid, accounting for less than 1% of the total fatty acid content in olive oil.

Eicosenoic acid: This is a monounsaturated fatty acid, accounting for less than 1% of the total fatty acid content in olive oil.

The oils are samples taken from **three Italian regions** varying number of areas within each region. The regions and their areas are recorded as shown in the following table:

Region	Area
North	North-Apulia, South-Apulia, Calabria, Sicily
South	East-Liguria, West-Liguria, Umbria
Sardinia	Coastal-Sardinia, Inland-Sardinia

DATA PROCESSING

After doing some research , we found out that there are **two types of oil** in these regions :

- **Extra virgin oil**
- **Organic extra virgin oil**

respectively in the regions :

North-Apulia ------ **Extra virgin olive oil**
South-Apulia ----- **Extra virgin olive oil**
Calabria ---------- **Organic extra virgin olive oil**
Sicily ---------- **Organic extra virgin olive oil**
Inland-Sardinia --- **Extra virgin olive oil**
Coast-Sardinia ---- **Extra virgin olive oil**
Umbria ---------- **Extra virgin olive oil**
East-Liguria -------- **Extra virgin olive oil**
West-Liguria ------- **Extra virgin olive oil**

For this reason , we'll change the **regions** with the **two types** that we identified which will be **our target in the dataset.**

We're also going to drop the unnecessary fields like **id , area** and **other** columns

02

DATA AUGMENTATION

Giving that the data offered for **Organic extra virgin olive oil** is far less than the other type , we can opt for a data augmentation to better enhance the data for that we use the **Gaussian Mixture algorithm*** with **half the number of samples of the Organic extra virgin olive oil**

***Gaussian mixture** model is a type of probabilistic model used in machine learning for clustering and density estimation. It assumes that the data is generated from a mixture of Gaussian distributions, where each Gaussian represents a distinct cluster or component of the data.

The model is defined by a set of parameters, including the number of components, the mean and covariance matrix of each component, and the weights of each component (i.e., the proportion of the data that belongs to each component). These parameters are learned from the data using an algorithm such as Expectation-Maximization (EM).

Initial dataset size	**572**
dataset size after data augmentation	**618**

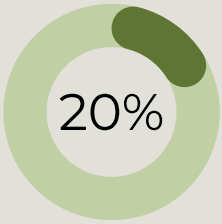
03

DATA TRAINING AND TESTING

DATA SPLITS

Now that our dataset is preprocessed and ready , we can split it into a **training set** and a **testing set**

Training set



Testing set

ML CLASSIFIERS

The models that we're going to use are the following :

- Naïve Bayesian
- Nearest Neighbors (-NN)
- Linear Discriminate Analysis (LDA)
- Decision Tree
- Artificial Neural Networks (ANN)
- Support Vector Machine (SVM)

and for each model , we'll calculate its **accuracy**

NAÏVE BAYESIAN

Naïve Bayes is a type of **probabilistic machine learning algorithm** used for classification and prediction tasks. It is based on the **Bayes' theorem of probability,** which provides a way to calculate the conditional probability of a hypothesis given some evidence.

In Naïve Bayes, the algorithm makes the assumption of independence between the features of the data. This assumption is called the "naïve" assumption, since it is often not true in real-world data. However, this simplifying assumption allows the algorithm to work efficiently with large datasets and high-dimensional feature spaces.

ACCURACY SCORE

0.9193548387096774

The algorithm calculates the probabilities of each class label given the features of the data, using the Bayes' theorem. It assumes that the features are conditionally independent given the class label, which means that the presence or absence of one feature does not affect the probability of another feature being present.

NEAREST NEIGHBORS (-NN)

The k-Nearest Neighbors (k-NN) algorithm is a **non-parametric machine learning algorithm** used for both **classification** and **regression** tasks. It is a type of instance-based learning, where the algorithm stores the training data instead of learning a model from it.
The k-NN algorithm works by finding the k closest data points to a given test data point in the feature space, based on a distance metric such as Euclidean distance or cosine similarity. The class or value of the test data point is then predicted based on the majority class or mean value of the k nearest neighbors. In other words, the algorithm assigns the test data point to the class or value that is most common among its k nearest neighbors.

To get the best k for the best precision , we choose an interval for k then use cross-validation to evaluate each k value

In this case , we choose **k=1**

ACCURACY SCORE

0.967741935483871

LINEAR DISCRIMINATE ANALYSIS (LDA)

Linear Discriminant Analysis (LDA) is a technique used in machine learning and statistics to **find a linear combination of features** that characterizes or separates two or more classes of objects or events. LDA is often used as a dimensionality reduction technique in data preprocessing or feature extraction, as well as a **supervised learning algorithm for classification**.In our case we'll use it as an algorithm for classification

ACCURACY SCORE

0.9193548387096774

DECISION TREE

A decision tree is a type of supervised learning algorithm used for classification and regression tasks. It is a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

ACCURACY SCORE

0.9354838709677419

TIn a decision tree, each internal node represents a feature or attribute of the data, and each leaf node represents a class label or a value for a regression task. The tree is constructed by recursively partitioning the data based on the values of the features, in a way that maximizes the information gain or minimizes the impurity at each split.

ARTIFICIAL NEURAL NETWORKS (ANN)

Artificial Neural Networks (ANNs) are a type of machine learning algorithm inspired by the structure and function of the human brain. ANNs consist of interconnected processing nodes or neurons that work together to learn and make predictions from input data.

The basic building block of an ANN is a neuron, which takes one or more inputs, applies a weight to each input, adds them together, and applies an activation function to produce an output. The weights and biases of the neurons are learned from the training data using an optimization algorithm, such as gradient descent or backpropagation.

To avoid overfitting for our dataset se tend to choose a smaller number of hidden layers (in our case **70**) and we choose the **"activation" function logistic** which refers to **1/(1+exp(-x))**

ACCURACY SCORE

0.9032258064516129

SUPPORT VECTOR MACHINE (SVM)

Support Vector Machines (SVMs) are a type of supervised learning algorithm used for classification, regression, and outlier detection tasks. SVMs aim to find a hyperplane that separates the data into different classes or groups, in a way that maximizes the margin between the hyperplane and the closest data points, known as support vectors.

ACCURACY SCORE

0.9274193548387096

04

MODEL EVALUATION & COMPARISON

COMPARAISON

Here is a summary of all **the precisions for each algorithm**

	Algorithm	precision
0	Naive bayes	0.919355
1	Nearest Neighbors	0.967742
2	LDA	0.919355
3	Decision Tree	0.935484
4	ANN	0.903226
5	SVM	0.927419