# 19AIE205
# Python for Machine Learning

# FOOD CLASSIFIER BASED ON NUTRIENT LEVELS

**TEAM-7 End Sem Report**

| | |
|---|---|
| Karthik Kanisettypalli | BL.EN.U4AIE20025 |
| Samirit Saha | BL.EN.U4AIE20058 |
| B Srivathsan | BL.EN.U4AIE20006 |

Course Instructor:

**Dr. Thangam S.**

# *CONTENTS*

# Table Of Contents:

# List of figures:

# 2. ABSTRACT

The given project aims to build a classifier that will identify which category a particular food item belongs to among the specific categories of foods, such as dairy products, fruits and vegetables, meat, and so on, based on the various levels of nutrients that it contains, which includes the level of proteins, carbohydrates, fats, vitamins, fiber content, and so on which is present in each of these food items. To classify the food based on the content of its' various nutrients, diverse kinds of classification algorithms can be used., and in our project we have made use of the following algorithms: K Nearest Neighbors (KNN) algorithm, Naïve Bayes (NB) classification algorithm, Logistic Regression algorithm. The KNN algorithm, which is a supervised learning algorithm, operates by finding a similarity between the new data point and the available cases which it was introduced to earlier, and then puts the new case in a category which has the most similarity to the categories that are available to it. Naïve Bayes classification is also a supervised learning algorithm that is based on Bayes' theorem, and is probabilistic in nature, which means that it makes its' prediction based on the probability of an object being in a particular category. The Logistic Regression algorithm (LR) is a supervised learning algorithm which can predict the outcome of a variable being in a particular category by giving a probabilistic value which lies between 0 and 1, after fitting an S-shaped logistic function with the data given. Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression, wherein the distance between the different clusters of categorized data is maximized so that a new dataset can be easily grouped into one of the existing categories. Decision Tree algorithm is a machine learning feature that allows us to perform both classification as well as prediction after representing the prediction of the value in a tree-shaped flowchart where each internal node represents a test which has been taken on the attribute of a dataset. The Random Forest algorithm is a classifier comprising multiple decision trees which operate within different subsets of the given dataset. Finally, we will have an analysis of the seven algorithms and infer which one provides the best food classifier model.

# 3. INTRODUCTION

For a healthy lifestyle, it is necessary that a balanced diet should be consumed on a regular basis. It is a known fact that one type of food cannot provide all the essential nutrients that are required for the smooth and proper functioning of the human body. For example, one cannot retrieve all the important nutrients which one's body requires from only a diet of dairy-based foods, such as milk, curd, and butter, nor can he do the same from only a diet of meat-based products. In the end, only a balanced intake of all kinds of foods will provide the person of all essential nutrients, ranging from proteins, fats, carbohydrates, to the various kinds of vitamins and minerals, which shall help his body run efficiently and protect oneself from various diseases. Therefore, the idea behind the creation of this model is to classify the various foods which a person eats based on the level of different nutrients which are present within it, so that the person can have an idea if whether his daily diet contains all kinds of food items which can provide him the different nutrients which his body needs, and thus ensure that the person is aware of whether his daily consumption is from a balanced diet or not. Eventually, this model can help in ensuring the good health of an individual since a balanced diet is a crucial step to a healthy body and mind.

# 4. THEORY AND CONCEPT

## A. ALGORITHMS USED

### 1.K NEAREST NEIGHBORS ALGORITHMS (KNN)

The K Nearest Neighbors or KNN Algorithm is a supervised learning algorithm that can be used for both the purposes of classification as well as regression. Some of its' applications include the filling up of missing values and the resampling of datasets. As its' name implies, it considers the data points which are nearest to the data point being analyzed, to predict the class into which the data point can be placed. A few important features of the KNN algorithm are as follows:

1.It follows instance-based learning: Unlike in normal model-based learning algorithms where the weights from the training data are used by the algorithm for learning to predict the output, the KNN algorithm makes use of the entire instance of training data for learning so that the algorithm can learn to predict the output.

2.It is a **lazy learner**: The KNN model doesn't learn from the training data prior to the implementation of the model, in fact the learning process is postponed until the prediction of the output is requested for a new instance.

3.It is **non-parametric**: For a KNN algorithm, there is no mapping function which is predefined in the model.

Now let us go through the steps of the KNN algorithm one by one:

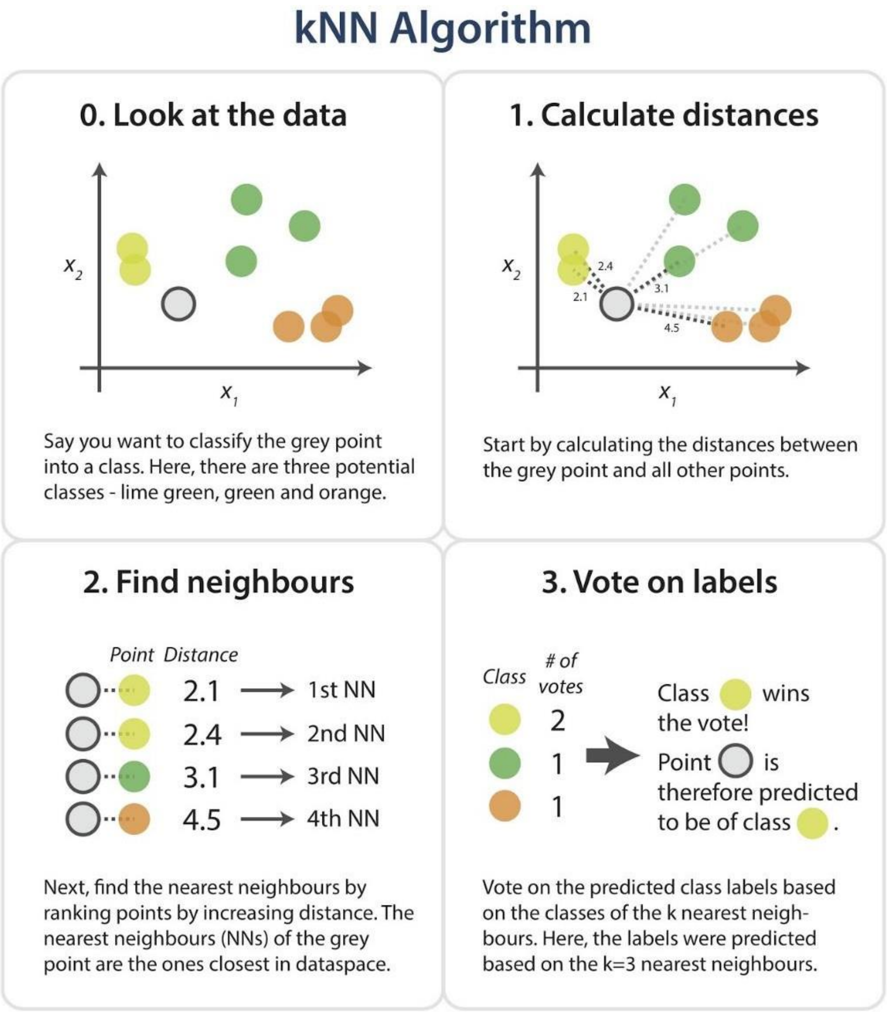
1.Selecting the value of K of the neighbors.

2.Calculating the Euclidean distance of the K number of neighbors.

3.Taking the K number of nearest Neighbors based on their Euclidean distances.

4.Among the selected k neighbors, counting the number of data points which are present in each category.

5.Assigning the new data points to the category for which the neighbor's number is maximum.



kNN Algorithm

0. Look at the data

Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances

Start by calculating the distances between the grey point and all other points.

2. Find neighbours

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

**FIG 1: KNN Algorithm**

## 2. NAIVE BAYES CLASSIFICATION ALGORITHM (NB)

The Naïve Bayes Classification Algorithm (NB) is a supervised learning algorithm that makes use of the Bayes theorem for solving problems that are related to classification. Its' application is usually for high dimensional classification such as text classification, spam classification, and so on. By nature, the NB algorithm is a probabilistic classifier, and this implies that it's; predictions are based on a probability distribution function which it creates based on the Bayes theorem. Among the various classification algorithms, the NB algorithm is known to make quick classifications.

The ‘Naive’ word in the NB classification refers to the fact that this algorithm assumes that the occurrence of a certain feature in a data point is independent of the occurrence of other features of that data point. This assumption made may or may not be true.

The Bayes’ theorem upon which the NB classification is used to determine the probability of an assumption with prior knowledge, its’ formula is:

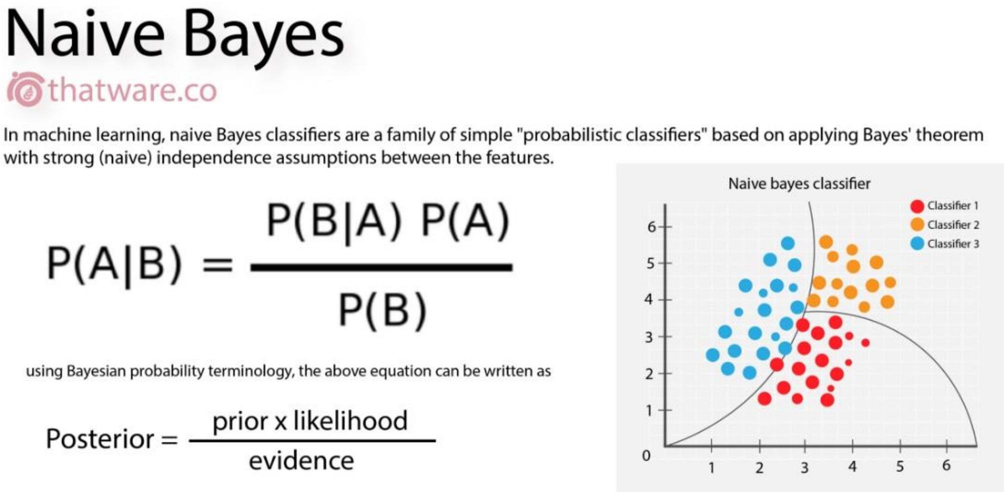
$$\mathrm{P(A|B)}= \frac{P(B|A)P(A)}{P(B)}$$

**In this case,**
**P(A|B) is Posterior probability**: Probability of assumption A on the observed event B being true.
**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of an assumption is true.
**P(A) is Prior Probability**: Probability of the assumption made being true before observing the evidence.
**P(B) is Marginal Probability**: Probability of the evidence being true.

The steps of the Naïve Bayes Classification Algorithm is as follows:

1. Convert the given dataset into frequency tables.
2. Generating a likelihood table after finding the probabilities of the given features, which can be done after the creation of a probability distribution function.

3. Making use of the Bayes theorem to calculate the posterior probability, **P(A|B).**



Naive Bayes

thatware.co

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

**FIG 2: Naive Bayes Algorithm**

## 3. LOGISTIC REGRESSION CLASSIFICATION ALGORITHM (LogR)

The Logistic Regression Algorithm predicts the output of a variable that is categorically dependent. This means that the LR algorithm can make a categorical or discrete prediction. For example, the outcome can be either Yes or No, 1 or 0, Spam or Not Spam, and so on. However, usually the answer is given as a probabilistic value that lies in between 1 or 0, Yes or No, and so on.

Despite the name being 'Logistic Regression', this algorithm is utilized more for the purpose of solving classification problems. While many concepts of the logistic regression are like its' counterpart, linear regression, a few fundamental differences between the two algorithms are as follows:

1. In Logistic Regression, instead of fitting a line like we do in the case of Linear Regression, we make use of an S-shaped curve, which is the geometric representation of the logistic function.
2. The curve from the logistic function is sued to determine the likelihood of an event to be possible, such as the event of finding a spam mail in the case of a logistic regression-based spam mail classifier algorithm, and so on.

A logistic function, also known as a sigmoid function, is a function that maps the predicted value to the probability of that predicted value occurring. It can map any real value to another value from the range of 0 to 1.

Another important feature in logistic regression is the concept of the ‘threshold value’ which can define the probability of an event being either possible (represented by ‘1’) or being impossible (represented by ‘0’). Values which are above the threshold value are assigned the value 1, and the ones below the threshold value are assigned the value 0.

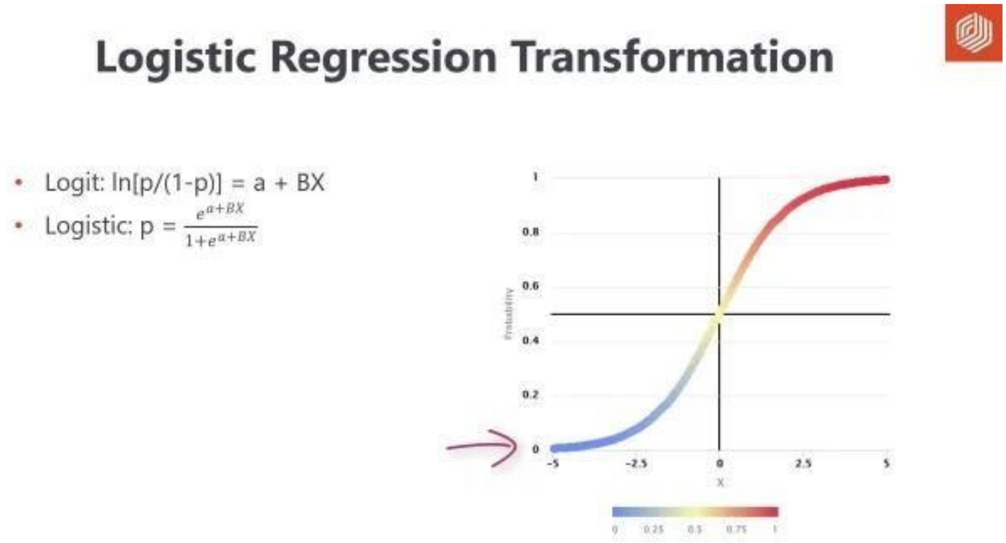
Logistic Regression can be of multiple types, such as Binary Logistic Regression (where there can be only two types of dependent variables such as ‘Spam’ or ‘Not Spam’), Multinomial Logistic Regression (where there can be more than two types of dependent variables, such as ‘Dairy Product’, ‘Dessert’, ‘Meat Product’, and so on) and Ordinal Logistic Regression (where there can be 3 or more ordered variables, such as ‘1st place’,’2nd place’, and so on).

The equation used in Logistic Regression is as follows:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The steps of LogR are as follows:

1. Create the dataset.
2. Map the logistic curve, or sigmoid curve and create a threshold value based on the sigmoid curve drawn.
3. Assign the datasets into the respective categories based on their comparison with the threshold value.



Logistic Regression Transformation

- Logit: ln[p/(1-p)] = a + BX
- Logistic: p = $\frac{e^{a+BX}}{1+e^{a+BX}}$

**FIG 3: Logistic Regression Algorithm**

## 4. SUPPORT VECTOR MACHINE ALGORITHM (SVM)

Support Vector Machine (SVM) algorithm is a supervised machine algorithm, which is utilized, as mentioned earlier, for both classification and regression purposes. In the SVM algorithm, each data item is plotted in an n-dimensional space (where n is the number of features which define the dataset), with each value of n serving as the value of a particular coordinate. Following the creation of the coordinates, the classification using SVM is performed by finding a hyper-plane (or hyper-planes, if there are more than 2 categories to be created) that can differentiate between the categories that have been created.

Therefore, given a set of training examples, wherein each example is marked to be belonging to a definite category, the SVM model can learn from these training examples, thus building a model which is able to assign each new data point one of the existing categories.

In addition to performing linear classification, SVMs can efficiently perform using a feature known as the 'kernel trick', wherein the inputs are implicitly mapped onto feature spaces of higher dimensions.

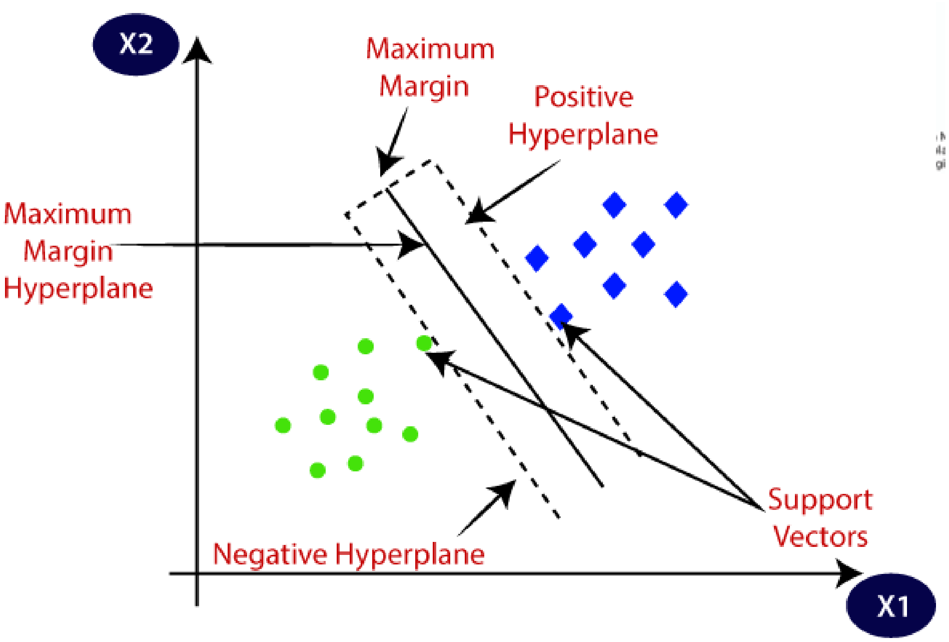


**FIG 4: SVM Algorithm**

## 5. DECISION TREE ALGORITHM

Decision tree is a supervised machine learning algorithm where the classification is done by continuously splitting the data according to a certain condition or parameter. The tree consists of two main components which are the decision nodes and the leaves. The leaves essentially represent the final outcomes, while the decision nodes are the places where the data is split during classification.

A decision tree can be of two types:

1. A classification tree (where the decision being made is a 'Yes/No' type of decision): A classification tree deals with discrete data, which can be split into either of two states, thus creating a binary tree-like structure. An example could be a tree which is made to determine whether a person has come to office. It could split the data based on certain parameters like: "Did he wake up at the right time?", "Did he take the right bus route?" and so on. The splitting of data in these cases would be based on whether the answer to the different questions is yes or no.
2. A regression tree: These kinds of trees deal with more continuous datasets. Here, instead of the data being classified into either of two classes, the number of classes into which the data can be classified increases. An example of this could include classifiers for topics with more than two categories. These include weather classifiers for example.

The general steps for creating a decision tree are:

1. Create root node for the tree
2. If all examples are positive, return leaf node 'positive'

3. Else if all examples are negative, return leaf node ‘negative’
4. Calculate the entropy of current state H(S), the formula for H(S) is: $$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$
5. For each attribute, calculate the entropy with respect to the attribute ‘x’ that is denoted by H (S, x). Entropy, which in this case is also called as Shannon Entropy, which is denoted by H(S) for a finite set S, is the measure of the amount of uncertainty or randomness in data.
6. Select the attribute which has maximum value of IG (S, x). Information gain (IG) is also called in this case as Kullback-Leibler divergence and is denoted by IG (S, A) for a set S is the effective change in entropy after deciding on a particular attribute A. It is used to measure the relative change in entropy with respect to the independent variables. $IG(S, A) = H(S) - H(S, A)$
7. Remove the attribute that offers highest IG from the set of attributes
8. Repeat until all attributes are exhausted, or the decision tree has all leaf nodes.

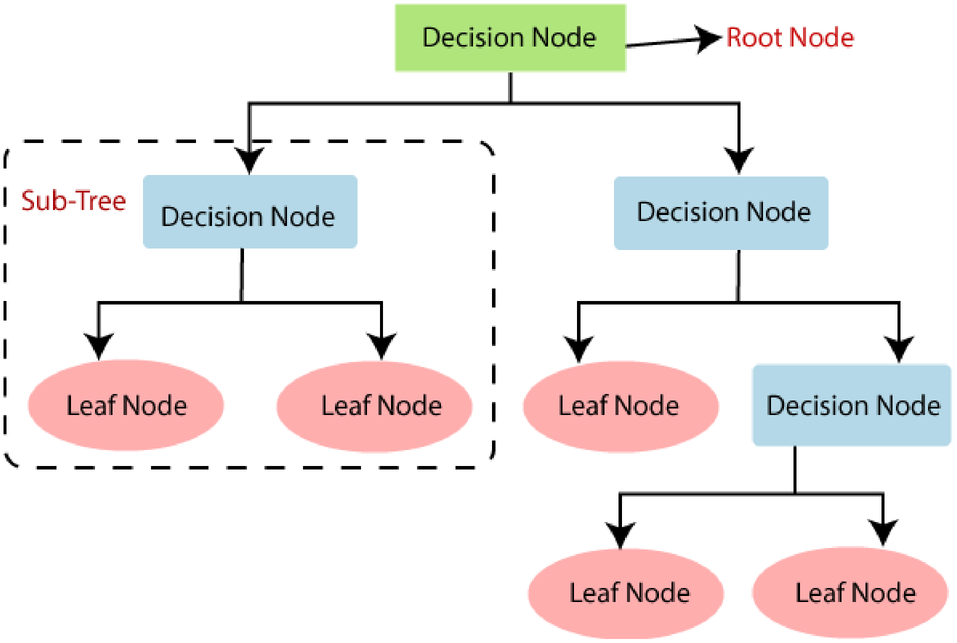


**FIG 5: Decision Tree**

## 6.Random Forest

Random forest algorithm uses supervised machine learning algorithm to predict values and in itself thus forms an improved Supervised learning algorithm with higher accuracy. Using the maximum values predicted from the different trees the data will be classified. This helps us prevent overfitting and work with high dimensional data.

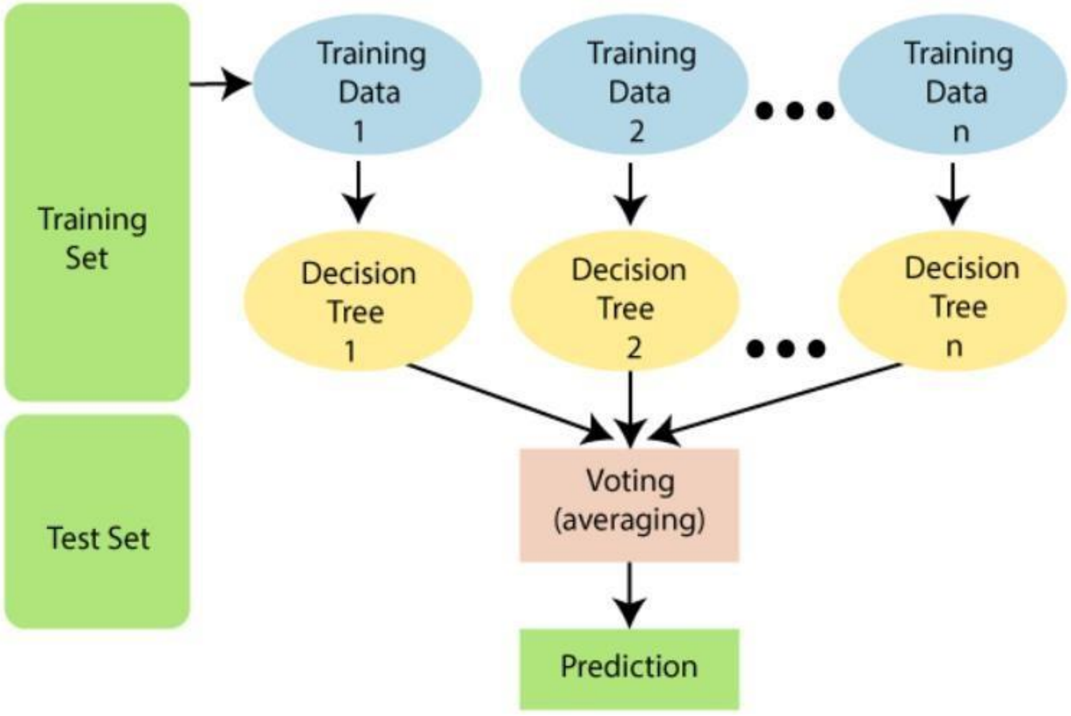


**FIG 6: Random Forest**

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

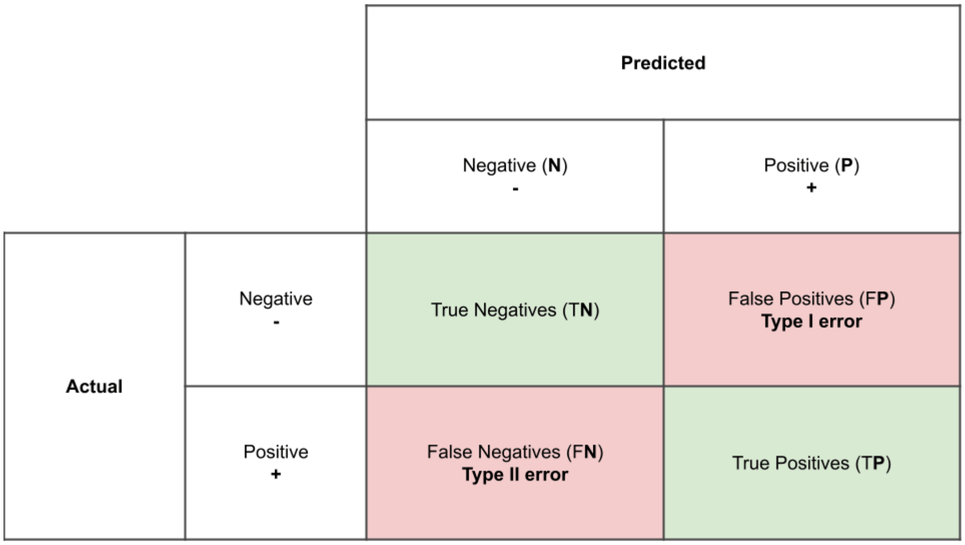
**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

## B. IMPORTANT PARAMETERS FOR EFFICIENCY OF CLASSIFER: ACCURACY, PRECISION, RECALL, F1 SCORE, AND CONFUSION MATRIX

Accuracy, precision, recall, and F1 score are four important metric values that are closely tied to the concept of the confusion matrix.

1. The **confusion matrix** is a tabular layout which provides information the performance of a classifier in machine learning, which might be a supervised or an unsupervised learning algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in the predicted version of that class.



| | | | |
|---|---|---|---|
| | | Predicted | |
| | | Negative (**N**)<br>- | Positive (**P**)<br>+ |
| **Actual** | Negative<br>- | True Negatives (T**N**) | False Positives (F**P**)<br>**Type I error** |
| | Positive<br>+ | False Negatives (F**N**)<br>**Type II error** | True Positives (T**P**) |

**Fig 7: Confusion matrix**

A confusion matrix has four entries,

I. **True Negatives (TN)**: Entries which are classified as negative and are also negative in reality.

II. **False Positives (FP)**: Entries which are classified as positive but are negative in reality.

III. **False Negatives (FN)**: Entries which are classified as negative but are positive in reality.

IV. **True Positives (TP)**: Entries which are classified as positive and are also positive in reality.

2. **Accuracy** is one of the most common statistical measures that are calculated to check the efficiency of a classifier, even though it is most effective only in the instance of a symmetric dataset, which means that the dataset doesn't have one particular type of data in majority, for example, an unsymmetric dataset might have too many samples taken from a particular location instead from a wider variety of places. The accuracy is calculated as the total number of instances that have correctly been labelled positive and negative, divided by the total number of instances in the dataset. Its' formula is:

   $$\text{Accuracy}= TP+TN/(TP+TN+FP+FN)$$

3. **Precision** tells us out of how many instances which have been predicted as 'positive', our classifier has found instances that are actually positive. It is especially important in cases where we might want to reduce the instances of false positives. For example, in a spam mail classifier, we might want to reduce the number of times, the spam mail classifier falsely classifies spam mail as positive, which here stands for them not being spam. Its' formula is:

   $$\text{Precision}= TP/TP+FP$$

4. **Recall** tells us how out of how many instances that are actually positive in our datasets, our classifier has identified instances that are positive. Its' formula is:

Recall= TP/(TP+FN)

5. The **F1 score** is a function involving both precision and recall. It is considered to be quite useful when we strive to maintain a balance between the precision and recall values when there is a large number of false negatives generated by the classifier, resulting in an uneven class distribution. (False negatives mean that the classifier classifies the instance as negative, when in reality it is positive.)

F1= (2 x precision x recall)/ (precision + recall)

# 5.IMPLEMENTATION

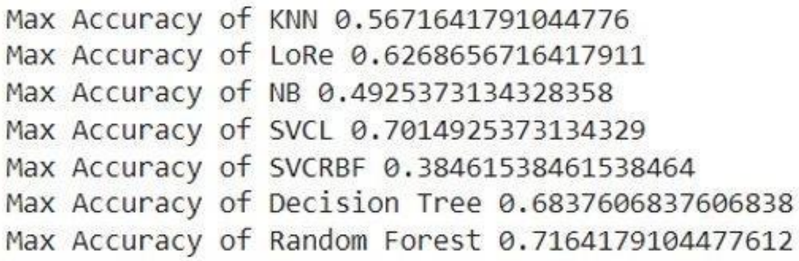
We Have implemented the following project on Python programming language using ipnyb file script.
The steps are as follows:

1. Import all required modules(pandas,mlxtend(plotting),pyplot,machine learning modules KNN,SVC etc)
2. Using pandas dataframe read the csv file nutrients.csv
3. Replace the values of string class variables of food types to integer class variables of upto 9
4. Construct arrays ac pc and rc to store accuracy,precision score and recall score for the 7 implemented models.
5. Using for loop change size of training and test dataset to compare and find the best size of training and test data set.
6. Allot dataset based on graph plot observation of accuracy,precision,recall score.
7. Choose the train data set size with priority to accuracy,precision and recall in decending order.
8. Plot confusion matrix for the 7 methods in most accurate set.
9. Plot decision regions and observe the behaviour of models.
10. Conclude the best model for the application by comparison.

Note: The Application of food prediction using nutritional values is used in forensics,Diet plan,qualitiative analysis to ensure safety etc.

# 6. RESULTS

## *A. Max Accuracy and Max Recall Scores of different classifiers:*



```
Max Accuracy of KNN 0.5671641791044776
Max Accuracy of LoRe 0.6268656716417911
Max Accuracy of NB 0.4925373134328358
Max Accuracy of SVCL 0.7014925373134329
Max Accuracy of SVCRBF 0.38461538461538464
Max Accuracy of Decision Tree 0.6837606837606838
Max Accuracy of Random Forest 0.7164179104477612
```

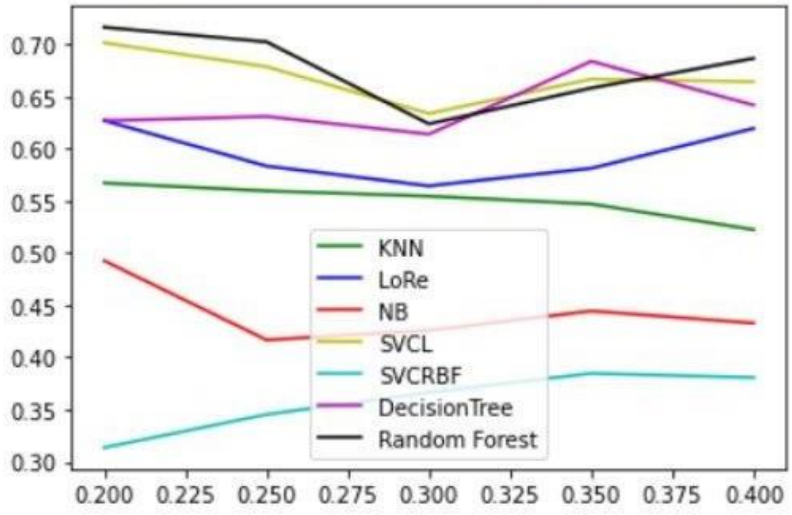
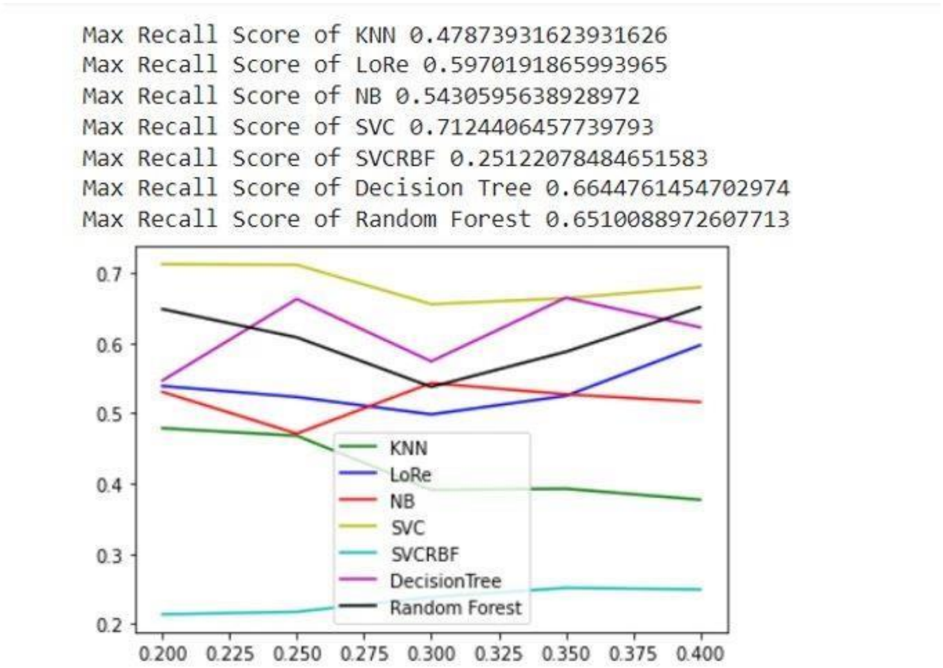
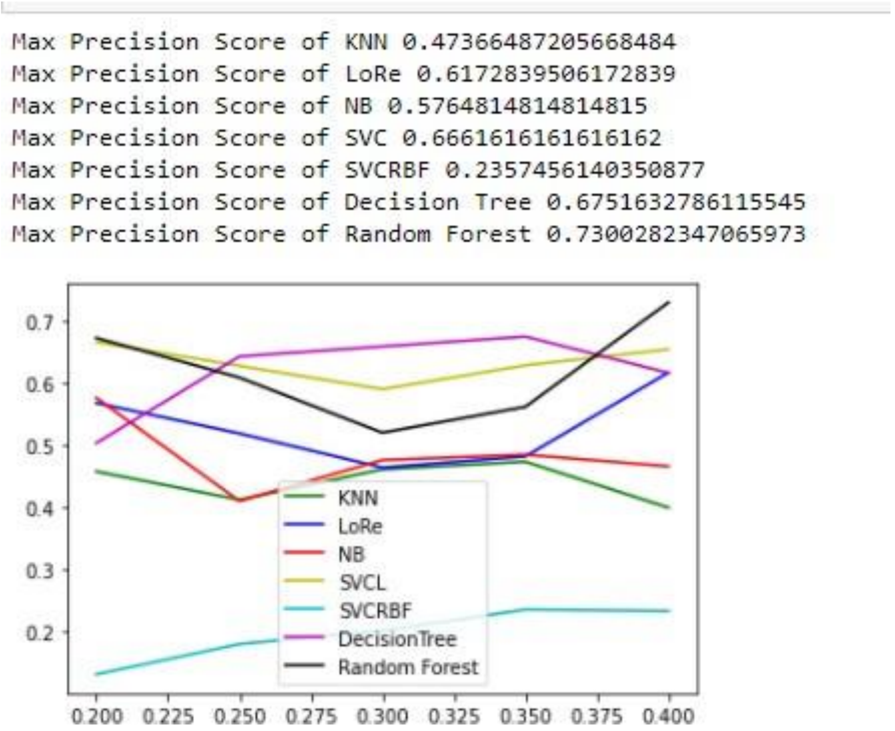


**FIG 8: Max Accuracy Scores**



```
Max Recall Score of KNN 0.47873931623931626
Max Recall Score of LoRe 0.5970191865993965
Max Recall Score of NB 0.5430595638928972
Max Recall Score of SVC 0.7124406457739793
Max Recall Score of SVCRBF 0.25122078484651583
Max Recall Score of Decision Tree 0.6644761454702974
Max Recall Score of Random Forest 0.6510088972607713
```

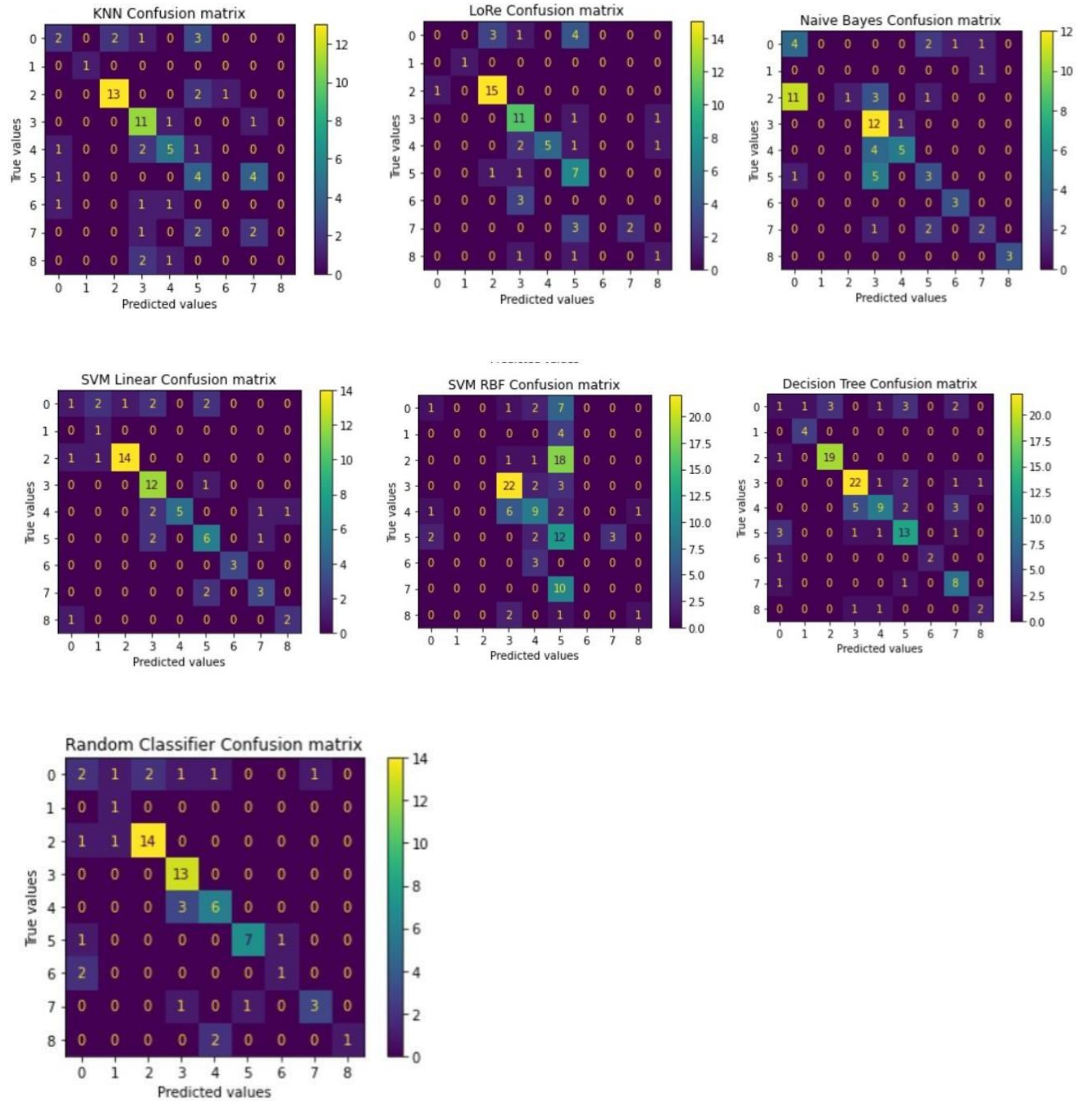
**FIG 9: Max Recall Scores**



```
Max Precision Score of KNN 0.47366487205668484
Max Precision Score of LoRe 0.6172839506172839
Max Precision Score of NB 0.5764814814814815
Max Precision Score of SVC 0.6661616161616162
Max Precision Score of SVCRBF 0.2357456140350877
Max Precision Score of Decision Tree 0.6751632786115545
Max Precision Score of Random Forest 0.7300282347065973
```

**FIG 10: Max Precision Scores**

*Confusion Matrices of different classifiers:*



KNN Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 0 | 2 | 1 | 0 | 3 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 13 | 0 | 0 | 2 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 11 | 1 | 0 | 0 | 1 | 0 |
| 4 | 1 | 0 | 0 | 2 | 5 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 | 4 | 0 | 4 | 0 |
| 6 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 0 |
| 8 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 |

LoRe Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 1 | 0 | 4 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 11 | 0 | 1 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 2 | 5 | 1 | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | 1 | 0 | 7 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 2 | 0 |
| 8 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

Naive Bayes Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 11 | 0 | 1 | 3 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 12 | 1 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 4 | 5 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 5 | 0 | 3 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

SVM Linear Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 1 | 2 | 0 | 2 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 14 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 12 | 0 | 1 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 2 | 5 | 0 | 0 | 1 | 1 |
| 5 | 0 | 0 | 0 | 2 | 0 | 6 | 0 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |

SVM RBF Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 2 | 7 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 1 | 18 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 22 | 2 | 3 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 6 | 9 | 2 | 0 | 0 | 1 |
| 5 | 2 | 0 | 0 | 0 | 2 | 12 | 0 | 3 | 0 |
| 6 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 1 |

Decision Tree Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 3 | 0 | 1 | 3 | 0 | 2 | 0 |
| 1 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 19 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 22 | 1 | 2 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 5 | 9 | 2 | 0 | 3 | 0 |
| 5 | 3 | 0 | 0 | 1 | 1 | 13 | 0 | 1 | 0 |
| 6 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 8 | 0 |
| 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 2 |

Random Classifier Confusion matrix

|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 1 | 2 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 14 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 13 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 3 | 6 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 | 7 | 1 | 0 | 0 |
| 6 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 3 | 0 |
| 8 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 |

**FIG 11:** ***Confusion Matrices of different classifiers***

*C. Plots of different classifiers:*

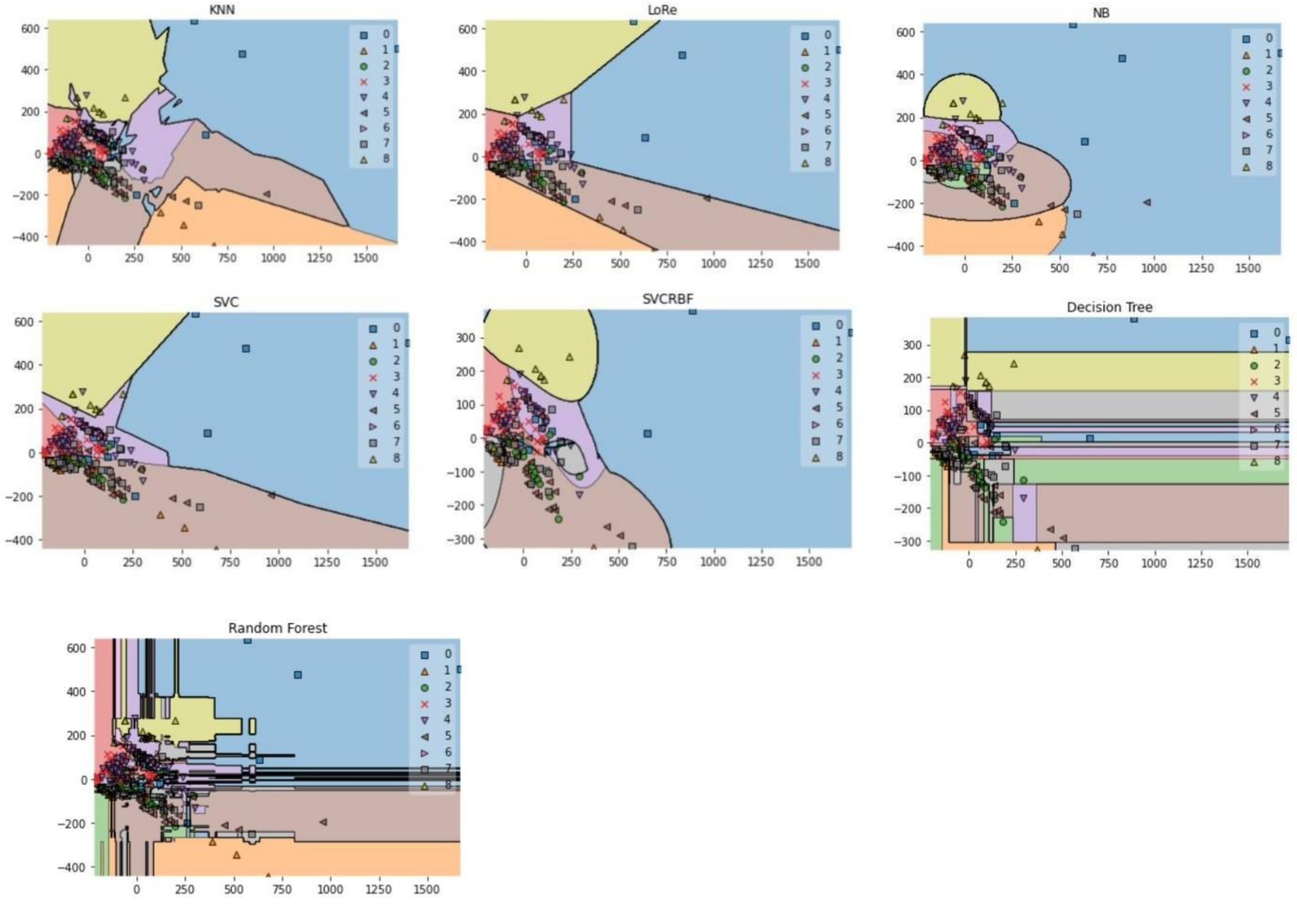


**FIG 12:** ***Plots of different classifiers***

# 7. COMPARISON

The following table will be presenting the accuracy, precision score and recall score of the 7 methods in a tabular form for analysis

**Table 1. Maximum values**

| ***S.no*** | ***Method*** | ***Accuracy*** | ***Precision Score*** | ***Recall*** |
|---|---|---|---|---|
| *1* | *KNN* | *0.56716* | *0.47366* | *0.47873* |
| *2* | *Logistic regression* | *0.62686* | *0.61728* | *0.59701* |
| *3* | *Naïve Bayes* | *0.49253* | *0.57648* | *0.54305* |
| *4* | *SVM linear* | *0.70149* | *0.66616* | *0.71244* |
| *5* | *SVM RBF* | *0.38461* | *0.23574* | *0.21522* |
| *6* | *Decision Tree* | *0.68376* | *0.67516* | *0.66447* |
| *7* | *Random Forest* | *0.71641* | *0.73002* | *0.65100* |

To decide the best model, we are going to assign a priority to the values of the maximum accuracy precision and recall as 3,2,1 as for this project in a multiclass model the precision score and accuracy is not as valid as accuracy in calculating the best model. As this field involves applications of very low impact a class decision won't impact the performance of the whole model the accuracy is assigned 3.

Formula= (Accuracy*3+precision score*2+recall score*1)/5KNN=0.5901

Logistic Regression=0.7305

Naïve Bayes=0.6459

*SVM linear=0.8342*

*SVM RBF=0.3004*

*Decision Tree=0.8055*

*Random Forest=0.8259*

# 8. CONCLUSION

We compared 7 different machine learning algorithms for classifying data that is supervised learning algorithms. The algorithms used were Logistic regression, K nearest Neighbors, Gaussian Naïve Bayes, SVM linear, SVM radial basis function, which is used as a default in python programming, Decision Tree, Random Forest. We have plotted graphs and used arrays to evaluate the best performing parameters for the Models. Using the optimized parameter, we predict and display the confusion matrices and the visualization of 9 class supervised learning and predict the values in a numerical fashion. By the comparison values we obtained we conclude that the order of model effectiveness is SVM Linear with a score of 0.8342, Random Forest 0.8259, Decision Tree 0.8055, Logistic Regression 0.7305 Naïve Bayes 0.6459, KNN 0.5901 and SVMRBF 0.3004.

# 9. REFERENCES

*[1] kaggle kernels output niharika41298/food-nutrition-analysis-eda -p /path/to/dest*

*[2]* ***Food vs Non-Food Classification,****Francesco Ragusa,Valeria Tomaselli,Antonino Furnari,Sebastiano Battiato,Giovanni M. Farinella, MADiMa '16: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, October2016, https://doi.org/10.1145/2986035.2986041*