WEEK- 7: Supervised learning-Classification.

Session No.2

Classification Algorithm in Machine Learning

As we know, the Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

What is classification?

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation intoa number of classes or groups. Such as, **Yes or No, 0 or 1, Spam or Not Spam, cat or dog,** etc.Classes can be called as targets/labels or categories.

Unlike regression, the output variable of Classification is a category, not a value, such as"Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labeled input data, which means it contains input with the corresponding output.

Types of classification

1. **Binary Classification:**

   Binary is a type of problem in classification in machine learning that has only two possible outcomes. For example, yes or no, true or false, spam or not spam, etc. Some common binary classification algorithms are logistic regression, decision trees, simple bayes, and support vector machines**.**

2. **Multi-Class Classification:**

   Multi-class is a type of classification problem with more than two outcomes and does not have the concept of normal and abnormal outcomes. Here each outcome is assigned to onlyone label. For example, classifying images, classifying species, and categorizing faces, among others. Some common multi-class algorithms are choice trees, progressive boosting,nearest k neighbors, and rough forest.

2. **Multi-Label Classification:**

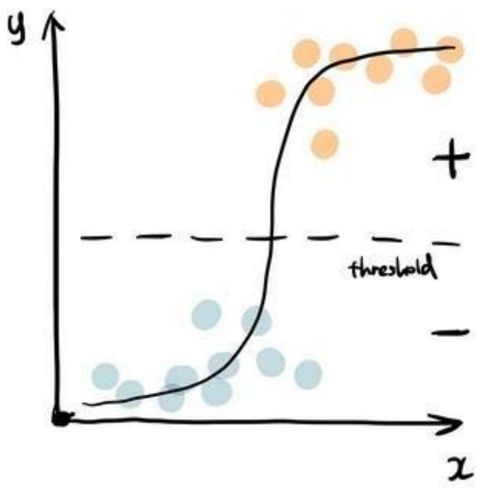
Multi-label is a type of classification problem that may have more than one class label assigned to the data. Here the model will have multiple outcomes. For example, a book or amovie can be categorized into multiple genres, or an image can have multiple objects. Somecommon multi-label algorithms are multi-label decision trees, multi-label gradient boosting,and multi-label random forests.

Imbalanced Classification:

Most machine learning algorithms assume equal data distribution. When the data distribution is not equal, it leads to imbalance. An imbalanced classification problem is a classification problem where the distribution of the dataset is skewed or biased. This method employs specialized techniques to change the composition of data samples. Some examples of imbalanced classification are spam filtering, disease screening, and fraud detection.

Classification Models

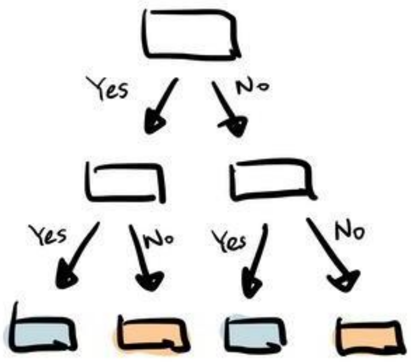
1. Logistic Regression



Logistics regression uses sigmoid function above to return the probability of a label. It is widelyused when the classification problem is binary — true or false, win or lose, positive or negative The sigmoid function generates a probability output. By comparing the probability with a pre- defined threshold, the object is assigned to a label accordingly.

```
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(X_train, y_train)
y_pred = reg.predict(X_test)
```

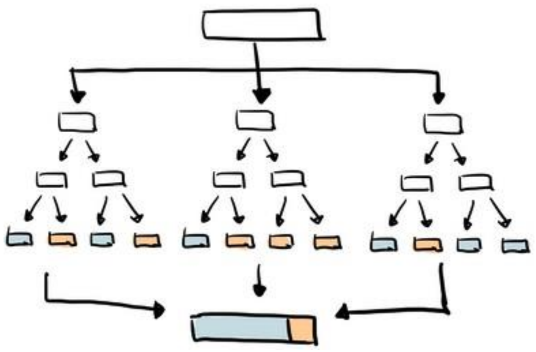
2.Decision Tree



Decision tree builds tree branches in a hierarchy approach and each branch can be considered asan if-else statement. The branches develop by partitioning the dataset into subsets based on mostimportant features. Final classification happens at the leaves of the decision tree.

```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)
y_pred = dtc.predict(X_test)
```

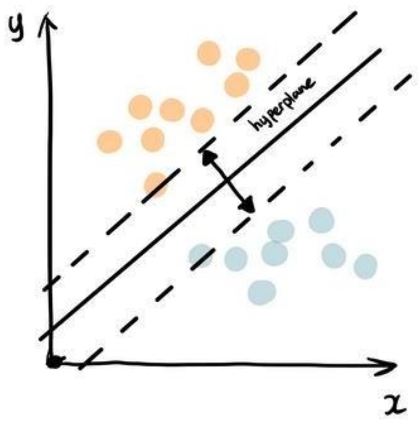
3.Random Forest



As the name suggest, random forest is a collection of decision trees. It is a common type of ensemble methods which aggregate results from multiple predictors. Random forest additionallyutilizes bagging technique that allows each tree trained on a random sampling of original datasetand takes the majority vote from trees. Compared to decision tree, it has better generalization butless interpretable, because of more layers added to the model.

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
```

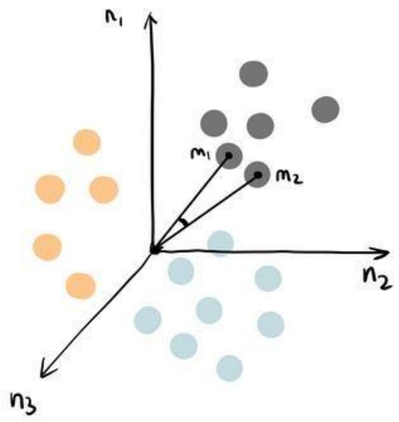
Support Vector Machine (SVM)



Support vector machine finds the best way to classify the data based on the position in relationto a border between positive class and negative class. This border is known as the hyperplane which maximize the distance between data points from different classes. Similar to decision treeand random forest, support vector machine can be used in both classification and regression, SVC (support vector classifier) is for classification problem.

```
from sklearn.svm import SVC
svc = SVC()
svc.fit(X_train, y_train)
y_pred = svc.predict(X_test)
```

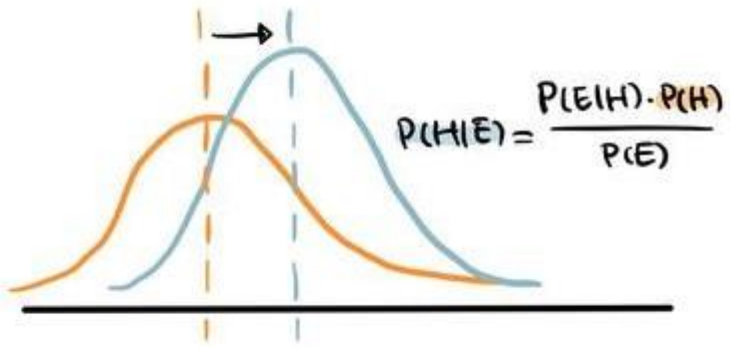
4.K-Nearest Neighbour (KNN)



You can think of k nearest neighbour algorithm as representing each data point in a n dimensional space — which is defined by n features. And it calculates the distance between one point to another, then assign the label of unobserved data based on the labels of nearest observed data points. KNN can also be used for building recommendation systems.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

5.Naive Bayes



Naive Bayes is based on Bayes' Theorem — an approach to calculate conditional probability based on prior knowledge, and the naive assumption that each feature is independent to each other. The biggest advantage of Naive Bayes is that, while most machine learning algorithms rely on large amount of training data, it performs relatively well even when the training data sizeis small. Gaussian Naive Bayes is a type of Naive Bayes classifier that follows the normal distribution.

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
```