Documentation
Machine Learning Project
Project No (1)

Eng: Mai Mokhtar
Date: 16/5/2022

Support Vector Machine (SVM):

It is considered a linear algorithm, any idea whose science is to separate two types of data through a line on the plane.

Hyperplane:

A decision boundary that differentiates the two classes in SVM. A data point falling on either side of the hyperplane can be attributed to different classes. The dimension of the hyperplane depends on the number of input features in the dataset.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Applications:

1-Classification

2-Regression

3-Oultier detection

4- Clustering

Why SVM?

-You have < 1000 data points (rows).
-Avoid premature optimization.
So we don't need to use the deep learning (complex model) for not so complex data.

Types of SVM:

-Linear SVM:

used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

-Non-Linear SVM:

used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

SVM's way to find the best line:

According to the SVM algorithm we find the points closest to the line from both the classes. These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane.

Why we used SVM:

-Because our label is binary classifier so that SVM predict the decision boundary good.

-Because the our data < 1000 and not complex.

-we used SVM with RBF kernel to guaranty correct separation if the data is non linear.

Advantages of Deep learning:

-Feature generation automation.

-Works well with unstructured data.

-Better self-Learning capabilities.

-Supports parallel and distributed algorithms.

-Cost effectiveness.

-Advanced analytics.

-Scalability.

Advantages of Multi Layer Perceptron (MLP):

- Capability to learn non-linear models.
- Capability to learn models in real-time (on-line learning).

Why the accuracy of MLP not high?

Because it needs features and data very big and it is sensitive to feature scaling.

What does the perceptron and why it's important?

- Perception is the sensory experience of the world. It involves both recognizing environmental stimuli and actions in response to these stimuli. Through the perceptual process, we gain information about the properties and elements of the environment that are critical to our survival.

- It is a good model in which we can give weight to the feature used and work on as much of the feature as we want.

- And we it in hyperparameter {'per__max_iter': 100}

What is the k nearest neighbor?

- The abbreviation KNN stands for "K-Nearest Neighbour". It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problem statements. The number of nearest neighbours to a new unknown variable that has to be predicted or classified is denoted by the symbol 'K'.

The advantage of KNN or why you we use it:

- The advantage of nearest-neighbor classification is its simplicity. There are only two choices a user must make: (1) the number of neighbors, k and (2) the distance metric to be used. Common choices of distance metrics include Euclidean distance, Mahalanobis distance, and city-block distance.

- And we use it in hyperparameter.

- { 'knn__n_neighbors': 30 }

What is the decision tree?

- Decision tree is a very specific type of probability tree that enables you to make a decision about some kind of process.

- It is a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves .

- And in our project we use it in hyper perimeter {'dt__max_leaf_nodes': 10}.