**Project 1**

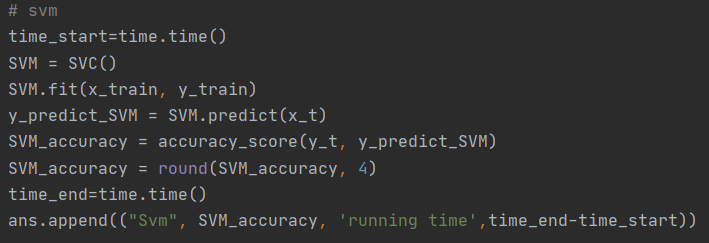
1. **Introduction**

The program provides an algorithm to analyze the winners of an LOL game by obtaining information about the games, such as First Dragon, First Tower, first Blood, etc., and the program will show the accuracy of its predictions.This program needs a time library to record the time spent on each classifier, panda library to read CSV files, sklearn.classifier libraries to train the classifier and predict the result and sklearn.metrics library to calculate the accuracy. When running this program, we will use the training set to train this four classifiers, namely Decision tree, K-NN, SVM and MLP. These classifiers are then used to analyze the test set and determine the winners. Finally, the precision is calculated by comparing with the actual results.

1. **Algorithms**
   * 1. Decision tree

Decision tree algorithm includes two stages: in the training stage, it is necessary to use certain standards and rules to split the training sample set into several subsets, and then to split each subset with the same rules. This process recursively stops until each subset contains only samples belonging to the same class. During the training process, each segmentation node needs to save the attribute number of the classification. In the test phase, the test sample is identified from the root node to see which child node the sample belongs to, and the same recursively executes until the sample is divided into leaf nodes, at which point the sample belongs to the category of the current leaf node.

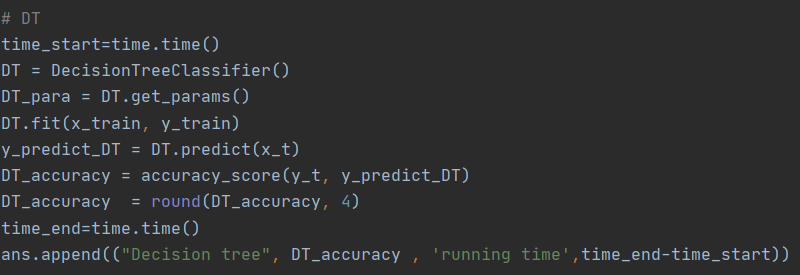
Parameter set: default



* + 1. K-NN

After inputting the new data without labels, each feature of the new data is compared with the corresponding feature of the data in the sample set, and then the algorithm extracts the classification label with the most similar data (nearest neighbor) in the sample set.

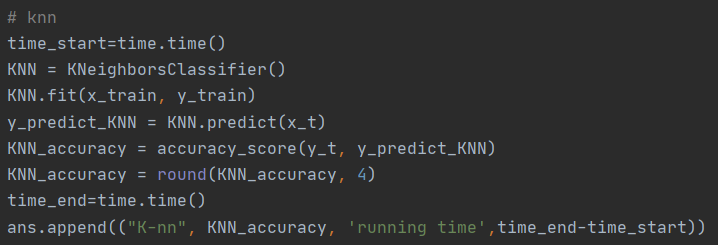
Parameter set: default



* + 1. SVM

The basic model of SVM is to find the optimal separation hyperplane in the feature space to maximize the positive and negative sample spacing on the training set.

Parameter set: default



* + 1. MLP

Neurons are arranged in layers. A neuron is connected to all neurons in next layer. Neurons may have different activation functions or no activation function.

Two training methods: 

* + - 1. Batch training 

One update, all samples 

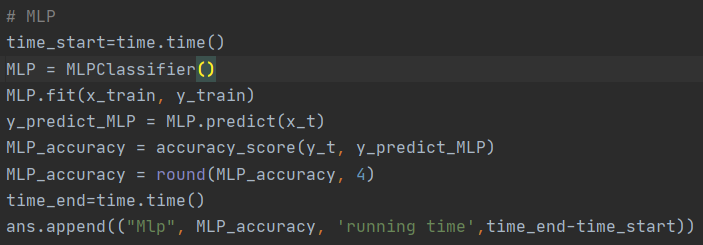
Sample sequence will not affect 

* + - 1. Stochastic training 

One update, one sample 

Samples can be chosen randomly to avoid the influence of the sample sequence

Parameter set: default



1. **Requirements**

We need to install the following libraries in Python to make sure the program runs.

import pandas as pd

import time

import train\_test\_split

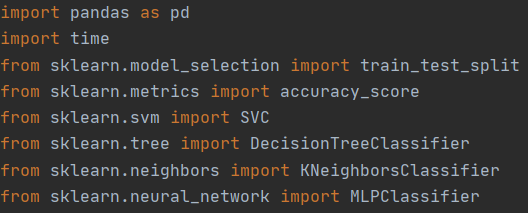
import accuracy\_score

import SVC

import DecisionTreeClassifier

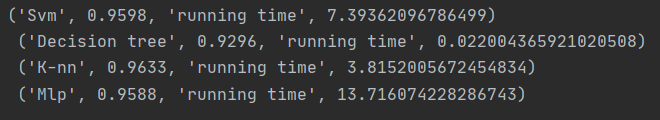
import KNeighborsClassifier

import MLPClassifier



1. **Result**

|  |  |  |
| --- | --- | --- |
| Classifier | Accuracy | Running time |
| SVM | 0.9598 | 7.393620 |
| Decision tree | 0.9296 | 0.022004 |
| K-NN | 0.9633 | 3.815200 |
| MLP | 0.9588 | 13,716074 |



1. **Comparison and Discussion**

We can clearly find that the accuracy of these four classifiers is very close. Their accuracy is very high and there is only a little difference. The reason for this situation may be that I used the default parameters, which led to the classification method of classifiers is too simple, so their accuracy is very high and close. On the other hand, we can find that although the accuracy of each classifier is very close, the time taken varies greatly, the Decision tree takes less than a second to get the results, while MLP takes nearly 14 seconds. If we have enough time, we can also set each parameter separately to get a more accurate result.