**Data Science Quest I Report**

Gao Yiming

ISTD Senior

Singapore University of Technology & Design

**Instructions for the program:**



The code is written with Python 3.6.2. You will need Python 3.0 or above to run it.

I change the .data file to csv file for easy processing, so you need to put this file together with iris.csv in the same folder.

I use the data in test set for prediction. You can change it to any input with format [ [sepal length, sepal width, petal length, petal width]] (for example, [[5.7,2.9,4.2,1.3]]). The program should return the following results: Ten most similar data points and their distances to the target; The predicted class for the input data and the probability of prediction.

**Machine Learning Algorithm**: knn (k nearest neighbors).

**Parameters**: random\_state = 2; weight = ‘uniform’; n\_neighbors = 10.

**Some tests:**

I tried to change k.

When random\_state = 0 and

K = 20, accuracy = 0.9474

k = 10, accuracy = 0.9737;

K = 2, accuracy = 0.9474;

Then I change the random\_state:

Use k = 10,

Random\_state = 0, accuracy = 0.9737

Random\_state = 2, accuracy = 1.0

Random\_state = 10, accuracy = 0.9737

But the test data is too less. The test accuracy does not ensure the model will work well on large set of data. We will need more variable test data to adjust the parameters.

**Solution Extensions:**

* I can implement a function to find the optimal value of k;
* I can try other algorithms like svm or neural network to solve this problem.
* For visualization, I can develop a simple python based gui. It will show the ten most similar points given input arguments. There can also be a graph illustrating the distances between the ten points.

**References:**

<https://scikit-learn.org/dev/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>