CSCE 550 Project3

Due: November 12, 2019

1- Objective

In this assignment, we build a simple kNN model for a dataset and we explore some methods to optimize model parameters.

2- Dependencies

Python > 2.7
Scikit Learn package
Pandas toolkit (If it is needed)
Numpy toolkit (If it is needed)

3- Dataset

For this assignment, we use Wine quality dataset. More information about this data set can be found here: https://archive.ics.uci.edu/ml/datasets/Wine+Quality.
Please use the dataset uploaded on beachboard.

4- kNN project description

Part 1 – Preliminary Tasks (Simple data wrangling)

A- Make yourself familiar with data. You can read about it from link above.
B- Make yourself familiar with "train_test_split" function of Scikit. You can find some basic information here: https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
C- Identify the number of classes that exists in the dataset. (Pandas has a feature that you can collapse a column to the number of repeating items, something like "groupby" in SQL language)
D- Investigate all fields and make sure there is not an anomaly in the dataset. If you find any anomaly, make a proper decision to alleviate the problem. For example, eliminating that row or changing the current value to a default value you picked for that column carefully. If you decide to change the value, make sure your change is correct and reflects the data's integrity.

Part 2 – Building and training the kNN model

A- Split your data into training and test dataset using "`train_test_split`" function. Put test_size = 0.2 and use stratify=y.
B- Use k value from the input (in the next section, we try to find optimum k) and build your model. For example if you pick your k = 4, 3 out of 4 is the winner!
C- Two main functions that should be used here are "`KNeighborsClassifier`" for knn classifier and "`fit`" method of the

"`KNeighborsClassifier`" object for fitting your test and training datasets. This is your training step!

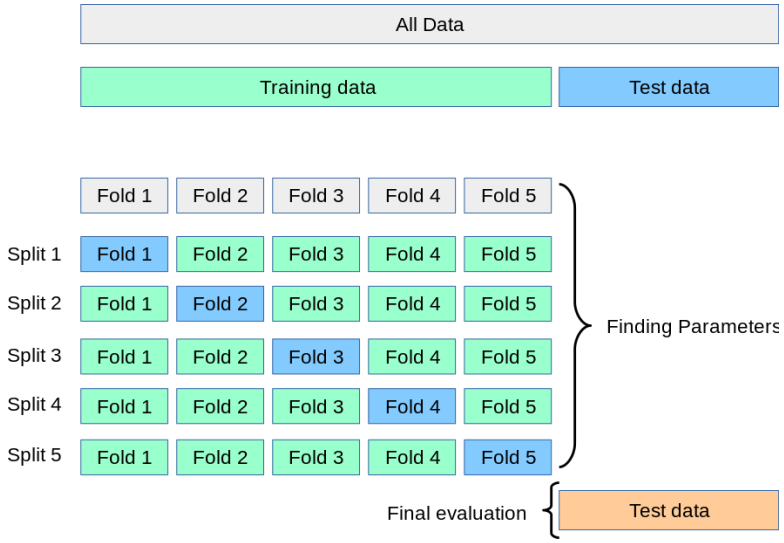
Part 3 – Testing kNN Model

A- To test your model use "predict" method of "`KNeighborsClassifier`" and then use the "score" method of "`KNeighborsClassifier`" to get the accuracy of your model.
Note: This method of testing is called holdout.

B- Report the accuracy.

Part 4 – Cross validation

A- In Cross-Validation dataset is randomly split up to T groups. Then one of the groups is considered as the test and the rest are used as the training. The process is repeated for all T groups.



For more information on cross validation in scikit please see
https://scikit-learn.org/stable/modules/cross_validation.html

B- To run cross validation, use "`cross_val_score`" function. The "cross value" parameter will be read from input.

Part 5 - Optimizing n-neighbor parameter

A- Optimizing a model parameters (hyper-tuning parameters) is a process to find optimal parameters to improve model accuracy.

B- For optimizing k value you need to use "`GridSearchCV`" function with given range of k. in this case test your model with k in the range of [1 .. 25]. For more information on "`GridSearchCV`" please see:

https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

5- What are your deliverables?

a. knn.py
   Your program should run as <knn.py> k T R1 R1
   k is initial k. T number of groups for cross validation. R1 and R2 are ranges for optimizing your k.
   Here please do not generate any chart when your program runs on terminal

b. A report file (pdf please)
   Your document contains the output of each step and necessary charts and plots.