CSE-6363 Machine Learning, Fall 2024

Programming Assignment 1

For this programming assignment you will implement the KNN algorithm from scratch and the functions to evaluate it with a k-fold cross validation (also from scratch). You can use the code in the following tutorial to get started and get ideas for your implementation of the KNN algorithm but please, enhance it as much as you can (there are many things you can do to enhance it such as those mentioned at the end of the tutorial):

https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/

Another nice extensions that you can implement to get better grade is:

- Kernel Nearest Neighbors
- An Improved Multilabel k-Nearest Neighbor Algorithm Based on Value and Weight
- Extensions of the K nearest neighbor methods for classification problems
- Other interesting extensions that you can find…

and you can also follow this tutorial to implement your KNN cross validation:

https://machinelearningmastery.com/k-fold-cross-validation/ (Links to an external site.)

Get the accuracy of your KNN algorithm using 10-Fold cross validation on the following datasets from the UCI-Machine Learning Repository and compare your accuracy with that obtained with KNN from Scikit-Learn (see NOTE 1, one comparison using hypothesis testing for each dataset between your KNN implementation and Scikit-Learn KNN implementation).

You might need to do some data preparation on the datasets to be used by KNN.

- Hayes-Roth Dataset (Links to an external site.)
- Car Evaluation Dataset (Links to an external site.)
- Breast Cancer Dataset (Links to an external site.)

Submit a report in "Microsoft Word" format (.doc or .docx) including the results of your comparisons and references to the tutorials that you used (remember that you borrowed some ideas/code from those tutorials or papers). You also need to submit your code so that our TA can test it and grade it. You should be able to explain how KNN and k-FCV work at a very detailed level and answer questions regarding your implementation to your TA.

NOTES:

1. The accuracy comparison must be a hypothesis testing as described in the respective class.
2. Remember that in case your algorithm cannot deal with a specific data type, you have ways to deal with this problem as mentioned in class (data pre-processing). If you have not had that class, please ask your TA of Professor.

3. Do your own work! I am giving you the reference to the KNN code and to the k-fold cross validation code in python to get you started and see how it works, but you must do your own implementation.
4. Your work will also be graded according to its originality, so I recommend you working on one of the extensions mentioned at the end of the tutorial (or a different one).
5. **DO NOT JUST COPY THE CODE FROM THE TUTORIAL.**