Read Me

Command Format:

For KNN with Random-Subsampling:
python driver_Random_Sampling.py <numberOfTrials> <k> <file-name> <feature-count>

For KNN with 5-fold cross validation:
python driver_kFold_Sampling.py <k> <file-name> <feature-count>

Decision boundary for Iris dataset :
python plot_iris.py <file-name>

1. Iris Data:

Command for,
1NN with Random subsampling: *python driver_Random_Sampling.py 10 1 iris.data 4*
3NN with Random subsampling: *python driver_Random_Sampling.py 10 3 iris.data 4*
1NN with 5-fold cross validation: *python driver_kFold_Sampling.py 1 iris.data 4*
3NN with 5-fold cross validation: *python driver_kFold_Sampling.py 3 iris.data 4*

2. Wine Data:

Command for,
1NN with Random subsampling: *python driver_Random_Sampling.py 10 1 wine.data 13*
3NN with Random subsampling: *python driver_Random_Sampling.py 10 3 wine.data 13*
1NN with 5-fold cross validation: *python driver_kFold_Sampling.py 1 wine.data 13*
3NN with 5-fold cross validation: *python driver_kFold_Sampling.py 3 wine.data 13*

3. Tic-Tac-Toe End game:

First create a modified data-set by replaing the characters/strings with integers using the following command:

python tic-tac-toe_preproess.py

The above command assumes that the tic-tac-toe data-set is present in the same file as that of the script.
After the exeution of the command, a file named tic-tac-toe.mod.data will be created in the same directory.
Now command for,
1NN with Random subsampling: *python driver_Random_Sampling.py 10 1 tic-tac-toe.mod.data 8*
3NN with Random subsampling: *python driver_Random_Sampling.py 10 3 tic-tac-toe.mod.data 8*
1NN with 5-fold cross validation: *python driver_kFold_Sampling.py 1 tic-tac-toe.mod.data 8*
3NN with 5-fold cross validation: *python driver_kFold_Sampling.py 3 tic-tac-toe.mod.data 8*