Computational Intelligence & Adversarial Machine Learning:

Assignment #3: Genetic & Evolutionary Feature Selection (GEFeS)

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Assignment #3

(Due 11/1/2018)

Assignment #3

- Given the baselines for an Linear Support Vector Machine (SVM), a Radial-Based Support Vector Machine (RBSVM) and a Backpropagation Neural Network select one of these three to improve upon the baseline accuracy using the following GEFeS Methods (to evolve a population of Feature Masks):
  - A Steady-State GA
  - An Elitist GA
  - An Estimation of Distribution Algorithm
- Your three GEFeS methods should use the following parameters:
  - Population Size = 25
  - Crossover = Uniform
  - Mutation Rate = 0.05
  - Number of Evaluations = 5000
  - Evaluation Function = Accuracy
- You should run your GEFeS methods 10 times and record the average and best results.

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Assignment #3

(Due 11/1/2017)

Assignment #3 (Grading)

- [30pts] Developing the three GEFeS methods
- [15pts] For the a table of the performances of the three methods recording the average and best performance of each method
- [15pts] For a plot of the Accuracy Curves with respect to the best performances of your three methods
- [10pts] For any innovation for improving GEFeS methods (one innovation per GEFeS method)
- [30pts] Develop the paper.

Assignment #3

(Due 11/1/2017)

Assignment #3 (cont.)

- Write a paper using IEEE or AAAI format documenting your work:
  I. Title
  II. Authors
  III. Abstract
  IV. Introduction
  V. Methodology
  VI. Experiment
  VII. Results
  VIII. Breakdown of the Work
  IX. References

Assignment #3

(Due 11/1/2018)

```
import Data_Utils
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler, normalize
from sklearn import svm
from sklearn.neural_network import MLPClassifier
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.model_selection import StratifiedKFold
from sklearn import preprocessing
import numpy as np
from sklearn.model_selection import cross_val_score

CU_X, Y = Data_Utils.Get_Casis_CUDataset()

rbfsvm = svm.SVC()
lsvm = svm.LinearSVC()
mlp = MLPClassifier(max_iter=2000)

skf = StratifiedKFold(n_splits=4, shuffle=True, random_state=0)
fold_accuracy = []

scaler = StandardScaler()
tfidf = TfidfTransformer(norm=None)
dense = Data_Utils.DenseTransformer()

for train, test in skf.split(CU_X, Y):
    #train split
    CU_train_data = CU_X[train]
    train_labels = Y[train]

    #test split
    CU_eval_data = CU_X[test]
    eval_labels = Y[test]

    # tf-idf
    tfidf.fit(CU_train_data)
    CU_train_data = dense.transform(tfidf.transform(CU_train_data))
    CU_eval_data = dense.transform(tfidf.transform(CU_eval_data))

    # standardization
    scaler.fit(CU_train_data)
    CU_train_data = scaler.transform(CU_train_data)
    CU_eval_data = scaler.transform(CU_eval_data)

    # normalization
    CU_train_data = normalize(CU_train_data)
    CU_eval_data = normalize(CU_eval_data)

    train_data =  CU_train_data
    eval_data = CU_eval_data

    # evaluation
    rbfsvm.fit(train_data, train_labels)
    lsvm.fit(train_data, train_labels)
    mlp.fit(train_data, train_labels)

    rbfsvm_acc = rbfsvm.score(eval_data, eval_labels)
    lsvm_acc = lsvm.score(eval_data, eval_labels)
    mlp_acc = mlp.score(eval_data, eval_labels)

    fold_accuracy.append((lsvm_acc, rbfsvm_acc, mlp_acc))

print(np.mean(fold_accuracy, axis = 0))
```

Basline Accuracies = [0.73 0.63 0.67]

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING

Have a Great Day!!!

AUBURN UNIVERSITY
SAMUEL GINN COLLEGE OF ENGINEERING