COMP 2019 Assignment 2 – Machine Learning

Please submit your solution via LEARNONLINE. Submission instructions are given at the end of this assignment.

This assessment is due on **Sunday, 16 June 2019, 11:55 PM.**
This assessment is worth 20% of the total marks.

In this assignment you will aim to identify which hand gesture is being performed based on recorded Electromyography (EMG) data. You will perform a number of machine learning tasks, including training a classifier, assessing its output, and optimising its performance. You will document your findings in a written report. Write concise explanations; approximately one paragraph per task will be sufficient.

Download the data file for this assignment from the course website (*file EMG.zip*). The archive contains the data file in CSV format, and some python code that you may use to visualise a decision tree model.

Before starting this assignment, ensure that you have a good understanding of the Python programming language, the Jupyter Python notebook environment, and an overall understanding of machine learning training and evaluation methods using the scikit-learn python library (Practical 3). You will need a working Python 3.x system with the Jupyter Notebook environment and the 'sklearn' package installed.

Documentation that you may find useful:

- Python: https://www.python.org/doc/
- Jupyter: https://jupyter-notebook.readthedocs.io/en/stable/
- Scikit-learn: http://scikit-learn.org/stable/
- Numpy: https://docs.scipy.org/doc/
- Pandas: https://pandas.pydata.org/ (optional, for reading the data file)

Preparation

Create a Jupyter notebook and set the random state based on your student ID.

```
import numpy as np
np.random.seed(1234) # use your StudentID in place of 1234.
```

Include this this code as the preamble to each of your questions in the Jupyter notebook.

Then, load the data. Use

```
import numpy as np
data = np.loadtxt(‘EMG.csv’,skiprows=1,delimiter=’,’, dtype=np.int)
```

to load the data. Type this code into the notebook. You will get a syntax error if you copy and paste from this document. Students familiar with the Pandas library may use that to load and explore the data instead.

Familiarise yourself with the data. There are 65 columns and 11678 rows. The first 64 columns represent the predictors, and the 65th column represents the target label. The 64 predictors are organised in 8 blocks, where each block corresponds to Electromyography (EMG) data obtained at the same time instant. There are 8 time instants, 0,…,7. In each block there are readings from 8 sensors (S1,…,S8). Hence, the column titled “S2_3” contains sensor readings taken from the second sensor, S2, at the fourth time instant.

The last column, titled Target, represents the gesture that was performed while taking the sensor readings. There are four gestures, each encoded as an integer in the range {0,…,3}.

Explore the distribution of data in each column.

Question 1: Baseline

What performance can we expect from this simple model?

Choose an appropriate measure to evaluate the classifier.
Select among Accuracy, F_1-measure, Precision, Recall, or ROC curve.
Justify your selection.

Note that you will need to use the same measure for all tasks in this Assignment.

Use a confusion matrix and/or classification report to support your analysis.

Question 2: Nearest Neighbour

Train a k Nearest Neighbour classifier (KNeighborsClassifier) to predict Target.

Use the Euclidean distance, 5 neighbours, and uniform weighting for the classifier. This should be the default offered by sklearn for this classifier.

Ensure that you follow correct training and evaluation procedures.

1. Assess how well the classifier performs on the prediction task.
2. What performance can we expect from the trained model if we used sensor data acquired from additional subjects as input?

Question 3: Decision Tree

Train a DecisionTreeClassifier to predict Target. Use the default parameter values for the classifier (i.e. don't specify your own values).

Ensure that you follow correct training and evaluation procedures.

1. Assess how well the classifier performs on the prediction task.
2. What performance can we expect from the trained model if we used sensor data acquired from additional subjects as input?

If you wish to visualise the decision tree you can use function print_dt provided in dtutils.py in the Assignment 2 zip archive:

```
import dtutils
dtutils.print_dt(tree, feature_names=flabels)
```

where tree refers to the trained decision tree model, and flabels is a list of features names (columns) in the data. This function prints a hierarchical representation of the tree where nodes deeper in the tree are indented further. For internal nodes, the children are shown. For leaf nodes, the class label associated with the node is shown, as well as the frequency of each class among the samples associated with the node (in square brackets).

Question 4: Diagnosis

Does the Decision Tree model suffer from overfitting or underfitting? Justify why/why not.

If the model exhibits overfitting or underfitting, revise your training procedure to remedy the problem, and re-evaluate the improved model. The DecisionTreeClassifier has a number of parameters that you can consider for tuning the model:

- max_depth: maximum depth of the tree
- *min_samples_split*: minimum number of samples required to split an internal node in the tree
- *max_leaf_nodes*: maximum number of leaf nodes in the tree
- *min_samples_leaf*: minimum number of samples per leaf nodes

Question 5: Recommendation

Which of the models you trained should be selected for the prediction task?

a) Assume that all errors made are equally severe.

b) Assume that correctly recognising the gesture 2 is three times more important than any other gesture. Would your recommendation change? Justify your argumentation based on the results obtained in the previous questions.

Task 6: Report

Write a concise report showing your analysis for Question 1-5.

Demonstrate that you have followed appropriate training and evaluation procedures, and justify your conclusions with relevant evidence from the evaluation output.

Where there are alternatives (e.g. measures, procedures, models, conclusions), demonstrate that you have considered all relevant alternatives and justify why the selected alternative is appropriate.

Ensure that the report is professionally presented and self-contained.

Do not include the python code in your report, and select relevant output from your program for use in justifications and discussion. Do not copy and paste the entire output into the report. The Jupyter notebook containing your code and complete output will be submitted as a separate deliverable.

Submission Instructions

Submit a single zip archive containing the following:

- **emg.ipynb**: the Jupyter Notebook file.
- **emg.html**: the HTML version of emg.ipynb showing the notebook including all output. Create this by selecting File>Download as>HTML after having run all cells in the Jupyter notebook.
- **emg.pdf**: the report as specified in Task 6 (i.e. your answers to questions 1-5)

Marking Scheme

Question	Marks
Q1: Baseline Appropriate measure selected and justified Correct evaluation & analysis	10
Q2: k Nearest Neighbour Correct training procedure applied Correct evaluation procedure applied Correct conclusion & analysis	15
Q3: Decision Tree Correct training procedure applied Correct evaluation procedure applied Correct conclusion & analysis	15
Q4: Diagnosis Correct diagnosis Correct revised training and evaluation procedure applied	30
Q5: Recommendation Correct recommendations Recommendations justified by evaluation results	20
Task 6: Report format Well-structured report Professional presentation	10
Jupyter notebook Random state set based on Student ID at the start of each question Executes correctly when using Run All Uses only packages/code mentioned in this assignment Copy saved as HTML format submitted Matches the contents of the report	Deductions apply