Assignment 1 – Classification using Scikit-learn

Assessment weighting:

This assignment is worth **15% of the total marks** for this module.

Due date:

This assignment is due by **23:59 on Sunday 20th October 2024**.

Please be aware that late submissions will not be accepted, except for cases with serious extenuating circumstances (e.g., illness).

Submission instructions:

Your report should be uploaded as a .pdf to the Turnitin link marked ‘Submit Assignment 1 Report’ provided on Canvas, with the naming convention CT4101_A1_lastname_firstname.pdf. Your report should be a separate document from your code submission and not contained e.g. within a Jupyter Lab notebook / .ipynb or other type of file.

Your code should be neatly laid out and formatted and commented appropriately. Submitted code samples (including comments) should adhere to the conventions in the PEP 8 Style Guide for Python Code (https://www.python.org/dev/peps/pep-0008/).

Your code should be submitted to the link marked ‘Submit Assignment 1 Code’ provided on Canvas in a single .zip folder (**NOT** in a .rar, tar.gz, .7z etc.), with the naming convention CT4101_A1_lastname_firstname_code.zip. Code submissions can be in the form of a standard Python module or modules (file extension .py), or in the form of a Jupyter Lab notebook (file extension .ipynb).

Submissions via email will not be accepted under any circumstances, you should upload your work on Canvas only. Failure to adhere to these submission instructions may lead to penalties being applied.

Plagiarism and copying:

Plagiarism is passing off the work of another person as one’s own.

While you are allowed to collaborate with your classmates and review online and print resources for high-level problem solving and background research, you are each expected to complete this assignment **individually**. This means that every sentence in your report submission should be written by **you alone**. You may reuse or modify **short snippets** of code from the lecture notes or online resources for your code submission, but the overall application structure and all code comments should be your own original work. Please see the University of Galway Code of Practice for Dealing with Plagiarism for further information on plagiarism: https://www.universityofgalway.ie/plagiarism/

Plagiarism is a serious academic offence and may lead to a loss of some or all marks and/or disciplinary proceedings if it is detected in any of your submissions. All submissions will be processed using automatic plagiarism detection software. Students who facilitate others to copy their work are also subject to plagiarism sanctions (including loss of marks), so you should not share your assignment solutions with classmates.

You may find it helpful to consult online guidelines for help with referencing:
https://libguides.library.nuigalway.ie/c.php?g=681111&p=4858348

Assignment description (25 marks max.)

The goal of this assignment is to learn the basics of using the scikit-learn package to develop machine learning (ML) models for a classification task. To complete this assignment you must write a Python application (extension .py) or a Jupyter notebook (extension .ipynb) that trains two ML models for classification and explores the impact of different hyperparameter values on the accuracy achieved by your models. You must also prepare a short .pdf report (5 pages max.) discussing your findings.

A dataset called **wildfires** has been supplied on Canvas. This dataset has already been split into separate training and test sets: **wildfire_training.csv** and **wildfire_test.csv**. The data is supplied in comma separated values format. Each row describes one instance in the dataset. The attributes are in columns in the following order: fire, year, temp, humidity, rainfall, drought_code, buildup_index, day, month, wind_speed. The goal of your classification models is to predict the value of the target attribute **fire** (which may be one of two classes: yes or no) based on the values of other attributes.

You are required to select any two classification algorithms (apart from *k*-NN) from the scikit-learn package, and then apply them to this dataset to train predictive models. Please note that selection of the *k*-nearest neighbours algorithm (scikit-learn class `KNeighborsClassifier`) is **not allowed** as we have covered a full worked example on it in lectures (**k-NN_hyperparameters.ipynb** – you may wish to consult this example on Canvas for hints on how to approach this assignment). Please note that it is not necessary to implement any ML algorithms yourself, you should use the algorithm implementations available in scikit-learn.

A list of available scikit-learn classifier implementations can be found at:
https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

The models that you develop should be trained on the supplied training set only. You should report accuracy results for each model on both the training set and the test set independently. You should present results for each model using the default hyperparameter settings in scikit-learn, and results achieved when attempting to tune two of the available hyperparameters for each model. For each model, you may choose any two available hyperparameters to tune. You should not use any automated hyperparameter tuning methods (e.g., the scikit-learn `GridSearchCV` class) for this assignment.

Ensure that you acknowledge all your sources of information using appropriate references when writing the report. All of the report should be written **in your own words** – do not copy text directly from online or print sources as this will lead to mark deductions for plagiarism.

The required sections in the report (5 pages max.) are described below:

1. **Description of algorithms (2 x 5 marks)**: Clearly describe each of your chosen scikit-learn algorithm implementations in turn, paying special attention to discussing the two hyperparameters that you have chosen to tune for each algorithm. You should write a maximum of 1 page per algorithm.
2. **Model training and evaluation (2 x 5 marks):** For each of your chosen algorithms, you should discuss the results achieved with the default settings, and also discuss on the results you achieved after trying out a selection of different values for the two selected hyperparameters. You should summarise the accuracy results on the training and test data in an appropriate format (e.g., in graphs or tables). You should write a maximum of 1 page per algorithm.
3. **Conclusion (5 marks):** Briefly sum up your key findings, including e.g., which model performed best, whether the achieved accuracy of one of the models is more sensitive to hyperparameter values than the other, and your recommended hyperparameter values for each algorithm based on your findings. You should write a maximum of 1 page for this section.

For each of the sections above, marks out of 5 will be awarded based on the following scale:

5 – exceptional, 4 – very good, 3 – average, 2 – passable, 1 – incomplete, 0 – section missing