Due Date November 12, 2023, 11:59pm
Late Submissions 30% per day per late deliverable
Teams You can do the assignment in teams of at most 3.
Teams must submit only 1 copy of the project via the team leader's account.

Experiments with Machine Learning

For this assignment, you will experiment with different machine learning algorithms and different data sets usingthe the `scikit-learn` library (see http://scikit-learn.org/stable/). `Scikit-learn` provides an interface to program with a variety of different algorithms and built-in datasets. There are plenty of tutorials and examples of code online.

1 The Data Sets

You will perform the same tasks with 2 different datasets:

Data Set 1: Penguins Download the Penguin dataset from Moodle. This dataset, in `csv` format, contains data regarding penguins such as their island, sex and a variety of physical features and we need to predict their species (Adelie, Gentoo, Chinstap)[1].

Data Set 2: Abalone Download the Abalone[2] dataset from Moodle. This dataset, in `csv` format, contains features of physical description of abalones (length, diameter, weights, etc) and one of 3 possible values for their sex: M (male), F (female), I (infant). Given the physical features of the abalone, the goal is to predict their sex.

2 The Experiments

For each dataset:

1. Load the dataset in Python.
   (a) for the Penguin dataset, the features island and sex are strings, therefore they need to be converted to numerical format to be fed to the MLP. To do so, experiment with 2 methods:
      i. convert these features into 1-hot vectors (also known as dummy-coded data)
      ii. convert these features into categories yourself
   (b) determine if the Abalone dataset can be used as is; otherwise convert any features using the 2 methods above.
2. Plot the percentage of the instances in each output class and store the graphic in a file called `penguin-classes.gif` / `abalone-classes.gif`. This analysis of the dataset will allow you to determine if the classes are balanced. Which metric is more appropriate to use to evaluate the performance. Be ready to discuss this at the demo.
3. Split the dataset using `train_test_split` using the default parameter values.

[1] For those you know Linux, you may appreciate these names.
[2] An abalone is a sea snail, you can read about it here.

4. Train and test 4 different classifiers:
   (a) **Base-DT:** a Decision Tree with the default parameters. Show the decision tree graphically (for the abalone dataset, you can restrict the tree depth for visualisation purposes)
   (b) **Top-DT**: a better performing Decision Tree found using a gridsearch. The gridsearch will allow you to find the best combination of hyper-parameters, as determined by the evaluation function that you have determined in step (3) above. The hyper-parameters that you will experiment with are:
      - criterion: gini or entropy
      - max_depth : 2 different values of your choice and "None"
      - min_samples_split: 3 different values of your choice

      Show the decision tree graphically (for the abalone dataset, you can restrict the tree depth for visualisation purposes)
   (c) **Base-MLP:** a Multi-Layered Perceptron with 2 hidden layers of 100+100 neurons, sigmoid/logistic as activation function, stochastic gradient descent, and default values for the rest of the parameters.
   (d) **Top-MLP:** a better performing Multi-Layered Perceptron found using grid search. For this, you need to experiment with the following hyper-parameter values:
      - activation function: `sigmoid`, `tanh` and `relu`
      - 2 network architectures of your choice: for eg 2 hidden layers with $30 + 50$ nodes, 3 hidden layers with $10 + 10 + 10$
      - solver: `adam` and stochastic gradient descent
5. For each of the 4 classifiers above 4(a), 4(b), 4(c) and 4(d), append the following information in a file called `penguin-performance.txt` / `abalone-performance.txt`: (to make it easier for the TAs, make sure that your output for each sub-question below is clearly marked in your output file, using the headings *(A), (B) ...*)
   (A) a clear separator (a sequence of hyphens or stars) and a string clearly describing the model (e.g. the model name + hyper-parameter values that you changed). In the case of Top-DT and Top-MLP, display the best hyperparameters found by the gridsearch.
   (B) the confusion matrix
   (C) the precision, recall, and F1-measure for each class
   (D) the accuracy, macro-average F1 and weighted-average F1 of the model
6. Redo steps 4 & 5, 5 times for each model and append in the performance files:
   (A) the average accuracy & the variance,
   (B) average macro-average F1 & the variance,
   (C) average weighted-average F1 & the variance.

   Does the same model give you the same performance every time? is the standard deviation high? Be ready to discuss this at the demo.

Useful methods

Below are useful methods you might consider using:

```
GridSearchCV
matplotlib.pyplot
neural_network.MLPClassifier
pandas.read_csv
pandas.get_dummies
pandas.Categorical
tree.DecisionTreeClassifier
```

3 Deliverables

The submission of the assigment will consist of 2 deliverables:

1. The code & the output files:

   - ☐ Submit all files necessary to run your code. If you used a Jupyter notebook, submit the `.ipynb` files; otherwise submit the `.py` files.
   - ☐ Submit a `readme.md` which will contain specific and complete instructions on how to run your experiments. You do not need to submit the datasets. If the instructions in your readme file do not work, are incomplete or a file is missing, you will not be given the benefit of the doubt.
   - ☐ Submit the 4 output files: `penguin-classes.gif`, `penguin-classes.txt`, and `abalone-distribution.pdf`, `abalone-performance.txt`.

2. The demo (presentation & Q/A)

   You will have to demo your mini-project for $\approx$ 15 minutes. Regardless of the demo time, you will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle. The demos will consist of 2 parts: a presentation $\approx$ 5 minutes and a Q/A part ($\approx$ 10 minutes).

   Prepare an 5-minute presentation to analyse and compare the performance of your models. The intended audience of your presentation is your TAs. Hence there is no need to explain the theory behind the models. Your presentation should focus on **your** work and the comparison of the performance of the models when the hyper-parameters are modified.

   Your presentation should contain at least the following:

   - ☐ An analysis of the initial dataset given on Moodle. If there is anything particular about these datasets that might have an impact on the performance of some models, explain it.
   - ☐ An analysis of the results of all the models with the data sets. In particular, compare and contrast the performance of each model with one another, and with the datasets. Please note that your presentation must be analytical. This means that in addition to stating the facts (e.g. the macro-F1 has this value), you should also analyse them (i.e. explain why some metric seems more appropriate than another, or why your model did not do as well as expected. Tables, graphs and contingency tables to back up your claims would be very welcome here.

   After your presentation, your TA will proceed with a question period. Each student will be asked questions on the assignment, and he/she will be required to answer the TA satisfactorily. In particular, each member should know what each parameters that you experimented with represent and their effect on the performance. Hence every member of team is expected to attend the demo.

   In addition, your TA may give you a new dataset and ask you to train or run your models on this dataset. The output file generated by your program will have to be uploaded on EAS during your demo.

4 Evaluation Scheme

Students in teams can be assigned different grades based on their individual contribution to project.

Individual grades will count for 15% and will be based on:

1. a peer-evaluation done after the submission.
2. the contribution of each student as indicated on GitHub.
3. the Q/A of each student during the demo (correct and clear answers to questions, knowledge of the program, ...).

The team grade will count for 85% and will be based on:

Code	functionality, proper use of the datasets, design, programming style, ...	45
Output files with given dataset	format, correctness and depth of discussion (`x-classes.gif`, `x-performance.txt`)	15
Demo – Presentation	depth of the analysis, clarity and conciseness, presentation, time-management, ...	15
Output with demo-dataset	correctness and format	10
Total		85

5 Submission

If you work in a team, identify one member as the team leader. The only additional responsibility of the team leader is to upload all required files (including the files at the demo) from her/his account and book the demo on the Moodle scheduler. If you work individually, by definition, you are the team leader of your one-person team.

Each deliverable is due on the date indicated below.

Deliverable	**Due Date**	**Upload as**
Submit your code, output files	see page 1	Assignment 1
Submit the output files generated at demo time	during your demo	Assignment 10 (yes! 10)

Code & Output files

- ☐ Create one zip file containing all your code, the output files for the data set on Moodle and the `README.md` file.
- ☐ Name your zip file: `472_Assignment1_ID1_ID2_ID3.zip` where `ID1` is the ID of the team leader.
- ☐ Have the team leader upload the zip file at: https://fis.encs.concordia.ca/eas/ as `Assignment1`.

Have fun!