**Hong Kong University of Science and Technology**
**COMP 4211: Machine Learning**
**Spring 2023**

**Programming Assignment 1**
Due: 6 March 2023, Monday, 11:59pm

# 1 Objectives

The objectives of this programming assignment are:

- To practise some data importing and preprocessing skills by using the `pandas` library.
- To gain a better understanding of supervised learning methods by using `scikit-learn`.
- To evaluate the performance of several supervised learning methods by conducting empirical study on a real-world dataset.

# 2 Dataset

You will use a water quality dataset provided as a ZIP file (`data.zip`). There are two `csv` data files in it. The table below summarizes the attributes of the data in each `csv` file:

| File | # of records | Has label? | # of columns |
|---|---|---|---|
| `train.csv` | 7,496 | yes | 21 |
| `test.csv` | 800 | yes | 21 |

The first 20 columns represent the features and the last column named 'is_safe' indicates whether or not the water is safe.

# 3 Major Tasks

The assignment consists of four parts and a written report:

Part 1: Use `pandas` for data importing and preprocessing.

Part 2: Use the linear regression model and feedforward neural networks for regression.

Part 3: Use the logistic regression model and feedforward neural networks for classification.

Part 4: Use `scikit-learn` to tune the hyperparameters.

Written Report: Report the results and answer some questions.

More details will be provided in the following sections. Note that **[Q*n*]** refers to a specific question (the $n$th question) that you need to answer in the written report. All the coding work should be done using Python 3.

# 4 Part 1: Data Preprocessing

In this part, you are required to preprocess the data and visualize the basic properties of the dataset. To be specific, if applicable, you need to remove the duplicates and fill in the missing values with their mean value. Note that we are not supposed to have the test data ready when performing data preprocessing on the training data. In the later questions, you are required to split the data in `train.csv` into training and validation sets. You need to perform data preprocessing first before splitting the data. After you have finished handling the above cases, visualize the correlation between every two of all the features in the training data with a heatmap.

[**Q1**] Visualize the correlation between every two of all the features with a heatmap.

# 5 Part 2: Regression

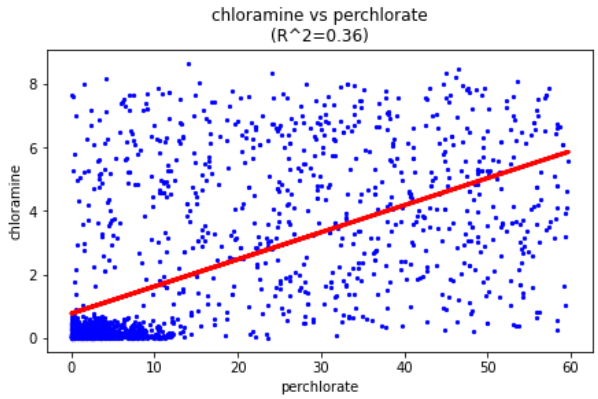
## 5.1 Linear Regression

In this task, you will build six linear regression models in the first step, where each model uses only one feature to find whether it is correlated with the attribute 'chloramine'. The six features, which should not include 'is_safe', are the ones of highest correlation to the attribute 'chloramine'. After this step, you will obtain six models for the plotting part of [**Q3**]. Then, in the second step, you will build another linear regression model to explore the relationship between linear combinations of the six selected features and 'chloramine'.

You are required to use the `train_test_split` submodule in `scikit-learn` to split the data in `train.csv`, with 80% for training and 20% for validation. You should set `random_state` = 4211 for reproducibility.

[**Q2**] After training the seven models described above using the training set, use them to make predictions on the validation set. Report the validation $R^2$ score of each of the models to evaluate the relationship between different features and 'chloramine'.

[**Q3**] Plot the regression line and the data points of the validation set for each of the first six models. For illustration, the figure below shows a plot of 'chloramine' versus the feature 'perchlorate' as well as the regression line.




[**Q4**] Report the mean squared error of each of the seven models in the validation set and compare them based on the two performance metrics, i.e., $R^2$ score and mean squared error.

[**Q5**] Build a linear regression model that uses the attribute (from among the first 20 columns

of the dataset, except the attribute 'chloramine') that has the least correlation to the attribute 'chloramine'. Plot the regression line and the data points of the validation set similar to [**Q3**]. Discuss the difference of the figures plotted in [**Q3**] and [**Q5**] that have the attributes of highest and lowest correlations, respectively.

### 5.2 Feedforward Neural Networks

In this part, you are asked to use the six features selected in Section 5.1 to train feedforward neural networks.

You need to try different number of hidden units $H \in \{1, 4, 16, 64, 128\}$ to build different three-hidden-layer neural networks. The hyperparameter `early_stopping` can be set to 'True' to avoid overfitting (default is 'False'). The other hyperparameters may just take their default values. For each hidden layer in a specific neural network model, the number of hidden units should be kept the same for simplicity. During training, you are expected to record the training time of each model. After training, evaluate your models by reporting the $R^2$ scores on the validation set. You have to report the $R^2$ score for *each value* of $H$ by plotting them using `matplotlib`.

[**Q6**] Report the model setting, training time, and performance of the neural network model for each value of $H$. You are also expected to repeat each setting three times for the same hyperparameter setting and report the mean and standard deviation of the training time and $R^2$ score for each setting.

[**Q7**] Compare the training time and $R^2$ score of the linear regression model and the best neural network model.

## 6 Part 3: Classification

In this task, you will build a logistic regression model as well as neural network classifiers to predict whether or not the water quality is safe.

You are also required to use the `train_test_split` submodule in `scikit-learn` to split the data, with 80% for training and 20% for validation. As before, we ask that you set `random_state` = 4211 for reproducibility.

### 6.1 Feature Selection

To reduce the computational cost and remove the irrelevant features, we would like to choose a subset of features for classification. You can use the feature selection module in `scikit-learn`. Use the mutual information statistics as the score for feature selection and then drop the five least important features.

[**Q8**] Report the score for each of the features.

### 6.2 Logistic Regression

Learning of the logistic regression model should use a gradient-descent algorithm by minimizing the cross-entropy loss. It requires that the step size parameter $\eta$ be specified. Try out a few values ($<1$) and choose one that leads to stable convergence. You may also decrease $\eta$ gradually during the learning process to enhance convergence. This can be done automatically in `scikit-learn` when set properly.

Use the features selected in Section 6.1 to train the model. During training, record the training time for the logistic regression model. After training, you are required to evaluate your model using accuracy and the F1 score on the validation set.

**[Q9]** Report the model setting, training time, and performance of the logistic regression model. Since the solution found may depend on the initial weight values, you are expected to repeat each setting three times and report the corresponding mean and standard deviation of the training time, accuracy, and the F1 score for each setting.

**[Q10]** Calculate the confusion matrix on the validation set with the model in **[Q9]** and report them. Give one reason why we need to examine the confusion matrix as well.

**[Q11]** Train the model on the training set and report the accuracy and F1 score on the validation set using different learning rate scheduling types (including 'sag', 'lbfgs', and 'sgd'). The `random_state` hyperparameter is again set to 4211. Other settings are the same as those mentioned in **[Q9]**.

### 6.3 Feedforward Neural Networks

Neural network classifiers generalize logistic regression by introducing one or more hidden layers. The learning algorithm for them is similar to that for logistic regression as described above. Remember to standardize the features before training and validation.

You need to try different number of hidden units $H \in \{1, 4, 16, 64, 128\}$ to build different three-hidden-layer neural networks. The hyperparameter `early_stopping` can be set to 'True' to avoid overfitting (default is 'False'). The other hyperparameters may just take their default values. During training, you are expected to record the training time of each model. After training, evaluate your models using accuracy and the F1 score on the validation set. You have to report the accuracy and F1 score for *each value* of $H$ by plotting them using `matplotlib`.

**[Q12]** Report the model setting, training time, and performance of the neural networks for each value of $H$. You are also expected to repeat each setting three times and report the mean and standard deviation of the training time, accuracy, and the F1 score for each setting.

**[Q13]** Plot the accuracy and F1 score for each value of $H$. Suggest a possible reason for the gap between the accuracy and F1 score.

**[Q14]** Compare the training time, accuracy and the F1 score of the logistic regression model and the best neural network model.

**[Q15]** Do you notice any trend when you increase the hidden layer size from 1 to 128? If so, please describe what the trend is and suggest a reason for your observation.

## 7 Part 4: Performance Enhancement

### 7.1 Hyperparameter Tuning

In this task, you need to use grid search to tune the hyperparameters of a three-hidden-layer feedforward neural network model to predict whether or not the water quality is safe. For your reference, all the hyperparameters defined in the `MLPClassifier` class in `scikit-learn` except the number of hidden layers can be tuned. Use the features selected in Section 6.1 for training and testing.

This time, you are required to evaluate your model on the test set (`test.csv`) provided. You need to import `test.csv` as a dataframe and standardize the features using the statistics you used for the training data. We assume that the test set comes from the same distribution as the training set.

You are required to use the `model_selection` submodule in `scikit-learn` to facilitate performing grid search cross validation for hyperparameter tuning. This is done by randomly sampling 80% of the training instances to train a classifier and then validating it on the remaining 20%. Five such random data splits are performed and the average over these five trials is used to estimate the generalization performance. You are expected to try at least 10 combinations of the hyperparameter setting. Set the `random_state` hyperparameter of the neural network model to 4211 for reproducibility and `early_stopping` to 'True' to avoid overfitting.

**[Q16]** Report six combinations of the hyperparameter setting.

**[Q17]** Report the three best hyperparameter settings in terms of accuracy as well as the mean and standard deviation of the validation accuracy of the five random data splits for each hyperparameter setting.

**[Q18]** Use the best model in terms of accuracy to predict the instances in the test set. Report the accuracy, F1 score and visualize the confusion matrix of the predictions on the test set.

### 7.2 Oversampling

By counting the number of records for different labels, you will notice that the training set is imbalanced. In this task, you will apply the oversampling strategy to tackle this problem. To be specific, you have to randomly oversample from the minority class and add these examples to the training set so that the two classes will become balanced in size. Set the `random_state` hyperparameter of the neural networks and oversampling to 4211 for reproducibility and `early_stopping` to 'True' to avoid overfitting. Another method to tackle this problem is to assign different weights to different classes. This can be done in `scikit-learn`.

**[Q19]** Train two logistic regression models using the two methods above (i.e., oversampling and different weights) to predict the instances in the test set (`test.csv`). Report the accuracy, F1 score and the confusion matrix of the predictions on the test set.

**[Q20]** Compare the accuracy, F1 score and the confusion matrix with those in Section 7.1 and discuss your findings.

## 8 Bonus: Comparison with Ensemble Methods (Optional)

*The questions in this part are optional and they will not be counted towards your grade for this assignment. As mentioned in class, students who do reasonably well for the bonus questions will be entitled for one day late in the submission of problem set or project later.*

In the previous questions, you solved the classification problem using two different methods, logistic regression and feedforward neural networks. In this part, you will need to compare them with ensemble methods. The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator. There are two families of ensemble methods: averaging methods and boosting methods. In averaging methods, the driving principle is to build sev-

eral estimators independently and then to average their predictions. By contrast, in boosting methods, base estimators are built sequentially and one tries to reduce the bias of the combined estimator. You can use the ensemble module in `scikit-learn`.

**[Q21]** Try to use the bagging classifier which is one of the averaging methods for the classification task. You can use logistic regression models or feedforward neural networks as the estimators. You also need to report the training time and the accuracy and F1 score on the test set. You are expected to repeat each setting three times and report the corresponding mean and standard deviation of the training time, accuracy, and the F1 score for each setting.

**[Q22]** Try to use the gradient boosting classifier which is one of the boosting methods for the classification task. You also need to report the training time and the accuracy and F1 score on the test set.

**[Q23]** Compare the performance of the two ensemble models with logistic regression and feedforward neural networks.

## 9 Report Writing

Answer **[Q1]** to **[Q20]** (**[Q1]** to **[Q23]** if you do the bonus part as well) in the report.

## 10 Some Programming Tips

As is always the case, good programming practices should be applied when coding your program. Below are some common ones but they are by no means complete:

- Using functions to structure your code clearly
- Using meaningful variable and function names to improve readability
- Using consistent styles
- Including concise but informative comments

You are recommended to take full advantage of the built-in classes of `scikit-learn` to keep your code both short and efficient. Proper use of implementation tricks often leads to speedup by orders of magnitude. Also, please be careful to choose the built-in models that are suitable for your tasks, e.g., `sklearn.linear_model.LogisticRegression` is *not* a correct choice for our logistic regression model since it does not use gradient descent.

## 11 Assignment Submission

Assignment submission should only be done electronically on the Canvas course site.

There should be two files in your submission with the following naming convention required:

1. **Report** (with filename `report.pdf`): in PDF form.
2. **Source code** (with filename `code.zip`): all necessary code, written in one or more Jupyter notebooks, compressed into a single ZIP file. The data should not be submitted to keep the file size small.

When multiple versions with the same filename are submitted, only the latest version according to the timestamp will be used for grading. Files not adhering to the naming convention above

will be ignored.

## 12 Grading Scheme

This programming assignment will be counted towards 10% of your final course grade. Note that the plus sign (+) in the last column of the table below indicates that reporting without providing the corresponding code will get zero point. The maximum scores for different tasks are shown below:

| **Grading scheme** | Code (60) | Report (+40) |
|---|---|---|
| **Part 1** | | |
| - [Q1] | 2 | +1 |
| **Part 2** | | |
| - Build the linear regression model | 3 | |
| - Compute the $R^2$ scores of the seven linear regression models + [Q2] | 2 | +2 |
| - Make prediction on the validation set + [Q3] | 3 | +2 |
| - Compute the MSE scores of the seven linear regression models + [Q4] | 2 | +2 |
| - Build the model and make prediction on the validation set + [Q5] | 2 | +1 |
| - Build the feedforward neural network model | 2 | |
| - Compute the training time and $R^2$ score for each value of $H$ in the feedforward neural network model + [Q6] | 3 | +2 |
| - Plot the $R^2$ score with different values of $H$ for the feedforward neural network model + [Q7] | 3 | +2 |
| **Part 3** | | |
| - Select the features according to mutual information + [Q8] | 3 | +2 |
| - Build the logistic regression model by adopting the gradient descent optimization algorithm | 2 | |
| - Compute the training time, accuracy, and F1 score of the logistic regression model + [Q9] | 3 | +2 |
| - Calculate the confusion matrix on the validation set + [Q10] | 3 | +2 |
| - Logistic regression with different learning rate scheduling types + [Q11] | 3 | +2 |
| - Build the feedforward neural network model | 2 | |
| - Compute the training time, accuracy, and F1 score for each value of $H$ in the feedforward neural network model + [Q12] | 3 | +3 |
| - Plot the accuracy and F1 score with different values of $H$ for the feedforward neural network model + [Q13] | 3 | +3 |
| - [Q14] | | +2 |
| - [Q15] | | +2 |
| **Part 4** | | |
| - Grid search on the feedforward neural network model for at least six combinations + [Q16] | 4 | +2 |
| - Report the three best hyperparameter settings and the validation accuracy (both mean and standard deviation) for each setting + [Q17] | 4 | +2 |
| - Report the accuracy and F1 score on the test set and visualize the confusion matrix + [Q18] | 4 | +2 |
| - Use the two methods to tackle imbalanced dataset + [Q20] | 4 | +2 |
| - [Q21] | | +2 |
| **Bonus** | | |
| - Build the averaging model + [Q21] | | |
| - Build the boosting model + [Q22] | | |
| - [Q23] | | |

Late submission will be accepted but with penalty.

The late penalty is deduction of one point (out of a maximum of 100 points) for every minute late after 11:59pm. Being late for a fraction of a minute is considered a full minute. For example,

two points will be deducted if the submission time is 00:00:34.

## 13 Academic Integrity

Please refer to the regulations for student conduct and academic integrity on this webpage: `https://registry.hkust.edu.hk/resource-library/academic-standards`.

While you may discuss with your classmates on general ideas about the assignment, your submission should be based on your own independent effort. In case you seek help from any person or reference source, you should state it clearly in your submission. Failure to do so is considered plagiarism which will lead to appropriate disciplinary actions.