Hong Kong University of Science and Technology
COMP 4211: Machine Learning
Spring 2025

Programming Assignment 1
Due: 28 February 2025, Friday, 11:59pm

1 Objectives

The objectives of this programming assignment are:

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

2 Dataset

You will use an anonymous dataset provided as a ZIP file (data.zip). There is one CSV data file: data.csv. The `'reg_target'` column of the CSV file is the **regression target**, while the `'class_target'` column is the **classification target**. By default, feature columns are **ordered** starting from `c1`.

3 Major Tasks

The assignment consists of four parts (plus an optional bonus part) and a written report:

Part 1: Use `pandas`, `scipy`, `matplotlib`, or `seaborn` for data importing and exploration.

Part 2: Use `scikit-learn` for data preprocessing.

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

Part 4: Use the logistic regression model and feedforward neural networks for classification.

Bonus: Use `scikit-learn` to tune the hyperparameters and validate the data preprocessing.

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 nth question) that you need to answer in the written report. All the coding work should be done using Python 3.

4 Part 1: Data Exploration and Preparation

Load the provided data.csv file using `pandas`. Write code to perform an initial data exploration to understand the basic statistics about all the attributes and target variables. Besides running the code to explore this dataset, you need to answer the questions [**Q1**] to [**Q3**]. Your report should include:

[Q1] Dataset Overview.

- **Size of the Dataset:** Total number of samples and features.
- **Missing Values:** Report **all** feature columns with missing values and the proportion of missing values for them.
- **Potential Impact:** Briefly discuss how **each** feature column containing missing values might affect model performance and data preprocessing.

[Q2] Feature Distribution.

- **Numerical Features:** Identify which ones are discrete or continuous. Summarize the distribution (i.e., mean, median, standard deviation, etc.) of **all** numerical features, and visualize the distribution of the **first-in-order** numerical features using a box plot.
- **Categorical Features:** Identify which ones are binary, nominal, or ordinal. Summarize the count of categories for **all** categorical feature, and visualize the distribution of the **first-in-order** categorical feature using bar plots.

[Q3] Correlation Analysis.

- **Feature Correlation:** Visualize the correlation between every two features (numerical features only, plus the regression and classification targets) with a heatmap. You can use `matplotlib` and `seaborn` to help you.
- **Insights:** Highlight any strong or weak correlations that might influence feature selection or necessitate feature engineering.

5 Part 2: Data Preprocessing Techniques

Preprocessing is an indispensable step in the machine learning pipeline, serving as the foundation upon which the performance of your models is built. It encompasses a range of techniques to transform raw data into a more suitable and effective format for modeling. Proper preprocessing can significantly enhance model accuracy, efficiency, and interpretability by addressing missing values, inconsistent data formats, and irrelevant or redundant features.

In this task, you will apply a variety of advanced preprocessing techniques from `scikit-learn` to your dataset. You are expected to understand the rationale behind each technique and its impact on the data to predict the effect on subsequent model performance. In addition to the required APIs, you can use and report on any other preprocessing and normalization modules in `scikit-learn`. You will have a chance to test these conjectures in the optional bonus part.

[Q4] Handling Missing Values: Remove features with too high missing proportion (e.g., $\geq 70\%$). Then, apply `SimpleImputer` with proper imputation strategies (e.g., mean, median, most_frequent, and constant) to handle different missing values. Briefly discuss which imputation strategies should be used for what scenario with examples from the dataset.

[Q5] Normalization and Standardization: Normalize/standardize different types of numerical features using `StandardScaler`, `MinMaxScaler`, or `RobustScaler`. You need to report the **first-in-order** numerical feature of the first 10 samples **before and after processing**. Briefly discuss the difference between these techniques and when to use each with examples.

[**Q6**] **Encoding Categorical Variables:** Remove categorical features that more than 70% of samples have unique categories. Then, utilize `OneHotEncoder` and `OrdinalEncoder` to encode different types of categorical variables. For certain ordinal feature(s), a direct application of the OrdinalEncoder may lead to a suboptimal ranking between categories. You need to handle this by adjusting the `categories` settings within the `OrdinalEncoder`. You need to report the **first-in-order** categorical feature column of the first 10 samples **before and after processing**. Briefly explain when `OrdinalEncoder` is preferred over `OneHotEncoder` with examples.

[**Q7**] **Feature Engineering:** In the dataset, feature column `c9` stands for the full name of a person. This identifier is useless to our task in its original format. However, titles (including Mr, Mrs, Master, and Miss) within people's names may contain potentially helpful information. Following this insight, please write code to extract the four titles from people's names and replace `c9` in your dataset with four new and meaningful features, each of which represents the presence of a title in the name. Show and justify your creation.

6 Part 3: Regression

6.1 Linear Regression

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

In this task, you will first build multiple linear regression models, where each model uses only one feature to determine its correlation with the regression target (the second last column `reg_target`). In this stage, you only need to focus on the numerical and ordinal categorical features you have identified.

Then, in the second and third steps, you will build two linear regression models using all the features you used in the first step to explore their combined relationship with the regression target.

[**Q8**] After training multiple models described above using the training set, use them to make predictions on the validation set. Report the R^2 score and mean squared error of **each** model on the validation set to evaluate the relationship between different features and the regression target. Then, use those selected features together to build a **single**, combined linear regression model and report the R^2 score and the mean squared error on the validation set.

[**Q9**] How can a nominal categorical feature with more than 2 possible categories be formulated as the independent variable of a linear model? In addition to selecting all the features you used in [**Q8**], please add a nominal categorical feature with more than 2 possible categories (and smaller than 10) to build another linear regression model. Report the R^2 score and the mean squared error of this combined model on the validation set.

6.2 Feedforward Neural Networks

In this part, you are asked to use the features used in [**Q8**] to train feedforward neural networks.

You need to try different numbers of hidden units $H \in \{1, 8, 32, 128\}$ to build different three-hidden-layer neural networks. The hyperparameter `early_stopping` can be set to 'True' to

avoid overfitting (default is 'False'). 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`. Note that for different model settings the best hyper-parameters may not be the same, so you shall experiment with different hyper-parameters to get reasonable results. You may run multiple times to get the best results for a specific model setting.

[Q10] Report the model setting, hyper-parameters, training time, and performance of the neural network model for each value of H.

[Q11] Compare the training time and R^2 score of the linear regression model and the best neural network model.

[Q12] Do you notice any trend of R^2 and MSE score when you increase the hidden layer size from 1 to 128? Explain the difference between the two metrics in evaluating regression models.

7 Part 4: Classification

In this task, you will build a logistic regression model as well as neural network classifiers to predict whether or not the classification target `'class_target'` is 1. 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. Note that for different settings the best hyper-parameters may not be the same, so you shall experiment with different hyper-parameters to get reasonable results.

7.1 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 η be specified. Try out a few values (>1) and choose one that leads to stable convergence. You may also decrease η gradually during the learning process to enhance convergence. When set properly, this can be done automatically in `scikit-learn`. Use the features used in **[Q8]** 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. Please be careful to choose the built-in models that are suitable for your tasks, i.e., the default setting of `sklearn.linear_model.LogisticRegression` is *not* a correct choice for our logistic regression model since it does not use gradient descent.

[Q13] Report the model setting, training time, accuracy, and the F1 score of the logistic regression model.

[Q14] Plot the ROC curve calculated on the validation set with the model in **[Q13]** and report the AUC value.

7.2 Feedforward Neural Networks

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

You need to try different numbers of hidden units $H \in \{1, 8, 32, 128\}$ to build different three-hidden-layer neural networks. The hyperparameter `early_stopping` can be set to 'True' to avoid overfitting (default is 'False'). 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`. You may run multiple times to get the best results for a specific model setting.

[Q15] Report the model setting, hyper-parameters, 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.

[Q16] Compare the training time, accuracy, and the F1 score of the logistic regression model and the best neural network model.

[Q17] Plot the ROC curve calculated on the validation set with the hidden layer size of 1 and the hidden layer size of 128. Give one reason why we need to examine the ROC curve as well.

8 Bonus: Performance Enhancement

The questions in this section are optional and designed to provide additional scoring opportunities and help students explore more on enhancing model performance. The maximum possible score for this assignment remains 100 points. If you lose points in Parts 1 to 4, the bonus points from this section can help recover your score up to the maximum.

8.1 Preprocessing Validation

In this task, you will build a logistic regression model as well as neural network classifiers to predict whether or not the classification target `'class_target'` is 1. 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. This section aims to demonstrate the impact of different preprocessing techniques on model performance by constructing and evaluating multiple pipelines. This task is designed to provide hands-on experience with `sklearn`'s `Pipeline` and `ColumnTransformer` utilities, enabling efficient experimentation with various preprocessing strategies. You could use `Pipeline` to create a sequence of preprocessing steps for numerical and categorical data separately. Then, `ColumnTransformer` could be used to apply the respective pipelines to your dataset's numerical and categorical columns. Your task is to evaluate how different preprocessing combinations affect the performance of three-hidden-layer neural networks to predict whether or not the `'class_target'` is 1.

[Q18] Combination A: For numerical features, apply `SimpleImputer` with mean strategy for imputation. For categorical features, use `OneHotEncoder` for encoding. Normalize **all** features with `StandardScaler`. Do not apply feature selection and feature engineering. Validate this combination with a neural network model.

[Q19] Combination B: For numerical features, use `StandardScaler` for normalization and `SimpleImputer` with zero strategy for imputation. For categorical features, encode ordinal features with `OrdinalEncoder` and remaining with `OneHotEncoder`. Do not apply feature selection and feature engineering. Validate this combination with a neural network model.

[**Q20**] **Combination C:** Create a custom combination where you choose a different preprocessing technique for numerical and categorical features based on your hypothesis of what might work best. This could also involve custom encoders, any appropriate feature selection, or engineering techniques. Validate this combination with a neural network model.

8.2 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 target `'class_target'` is 1. All the hyperparameters defined in the `MLPClassifier` class in `scikit-learn` except the number of hidden layers can be tuned. Use one of the preprocessing techniques in [**Q18-20**] and features selected in [**Q8**] for training and testing.

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.

[**Q21**] Report five combinations of the hyperparameter settings and highlight the best hyperparameter setting in terms of validation accuracy. You also need to report the mean and standard deviation of the validation accuracy for the five random data splits for each hyperparameter setting. Finally, you need to evaluate the model with the best hyperparameter setting on the testing set and report the testing accuracy.

9 Report Writing

Answer [**Q1**] to [**Q17**] ([**Q1**] to [**Q21**] 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_{student_id}.pdf`): in PDF form.

2. **Source code** (with filename `code_{student_id}.zip`): all necessary code, written in one or more **Jupyter Notebooks** (showing **all the execution outputs**), 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.

Grading scheme	Code (40)	Report (60)
Part 1: Data Exploration & Preparation	**6**	**14**
- Summarize dataset statistics [Q1]		2
- Discuss missing values and impact on learning [Q1]		2
- Interpret numerical and categorical feature distributions [Q2]		5
- Discuss correlation insights and feature selection implications [Q3]		2
- Provide relevant summaries and visualizations [Q2, Q3]	6	3
Part 2: Data Preprocessing	**13**	**12**
- Handle missing values using different imputation strategies and show [Q4]	3	1
- Explain the impact of imputation strategies with examples [Q4]		2
- Normalize/standardize numerical features and show [Q5]	3	1
- Compare normalization techniques and justify choices with examples [Q5]		2
- Encode categorical variables correctly and show [Q6]	3	1
- Discuss when to use Ordinal vs. OneHot encoding with examples [Q6]		2
- Apply feature engineering on column c9 [Q7]	4	
- Show and justify feature engineering choices [Q7]		1
- Provide sample before/after tables for transformations [Q7]		2
Part 3: Regression	**11**	**17**
- Train linear regression models with individual features [Q8]	3	
- Train a linear regression model using multiple features [Q8]	2	
- Compare R^2 and MSE for linear regression models [Q8]		4
- Train a linear regression model with additional nominal [Q9]	2	
- Discuss the impact of adding categorical features to regression [Q9]		3
- Implement and evaluate feed-forward neural networks for regression [Q10]	4	
- Report the model setting, hyper-parameters, training time, and performance of the neural network model [Q10]		4
- Compare and Interpret performance trends across different model settings [Q11]		4
- Provide visualizations and plots for performance metrics [Q12]		2
Part 4: Classification	**10**	**17**
- Build the logistic regression model [Q13]	4	
- Report model setting, hyper-parameters, training time, accuracy, and F1 score [Q13]		5
- Plot the ROC curve and report the AUC value [Q14]	1	2
- Build the neural network classifier model [Q15]	4	
- Report model setting, hyper-parameters, training time, accuracy, and F1 score for each value of H [Q15]		5
- Compare training time, accuracy, and F1 score of logistic regression vs. neural network [Q16]		3
- Plot ROC curve for different hidden layer sizes and discuss [Q17]	1	2
Bonus	**4**	**6**
- Combination A: Preprocessing pipeline and validation [Q18]	1	1
- Combination B: Preprocessing pipeline and validation [Q19]	1	1
- Combination C: Custom preprocessing pipeline and validation [Q20]	1	2
- Hyperparameter tuning using grid search [Q21]	1	2

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 hour late after 11:59pm with no more than two days (48 hours). Being late for a fraction of an hour is considered a full hour. For example, two points will be deducted if the submission time is

01:23:34.

12 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.