Hong Kong University of Science and Technology
COMP 4211: Machine Learning
Spring 2020

Programming Assignment 1
Due: 26 March 2020, Thursday, 11:59pm

1 Objectives

The objectives of this programming assignment are:

1. To practise some data importing and preprocessing skills by using the `pandas` library in Python.

2. To acquire a better understanding of supervised learning methods by using a public-domain software package called `scikit-learn`.

3. To evaluate the performance of several supervised learning methods by conducting empirical study on a Pokemon dataset.

2 Dataset

You will use a Pokemon dataset provided in the form of a ZIP file (`data.zip`). The following table shows the number of features and the number of records in each of the `csv` data files.

Data file	#features	#records
`pokemon.csv`	13	800
`battles.csv`	3	40,000
`q4_test.csv`	3	10,000

In `pokemon.csv`, each record is a Pokemon with its ID indicated in the column with label '#'. The IDs are used to refer to the Pokemons in `battles.csv` and `q4_test.csv`, where each record corresponds to a battle between two Pokemons.

3 Major Tasks

The assignment requires you to do the following:

1. To learn to use `pandas` for data importing and preprocessing.

2. To learn to use the linear regression model for regression.

3. To learn to use the logistic regression model for classification.

4. To learn to use the single-hidden-layer neural network model for classification.

5. To conduct empirical study using different supervised learning methods.

6. To answer several questions.

More details will be provided in the following subsections. Note that [Qn] refers to a specific question (the nth question) that you need to answer in the written report.

3.1 Task 1: Calculating the Win Rate

In this task, you need to calculate the win rate of each Pokemon using the battle history. You first import `pokemon.csv` and `battles.csv` as `pandas` data frames. Then, you create a new column called 'Win Rate' in the data frame created from `pokemon.csv`. The win rate of Pokemon i is defined as follows:

$$\text{WinRate}_i = \frac{\#\text{battles that Pokemon } i \text{ wins}}{\#\text{battles that Pokemon } i \text{ is involved}}$$

For example, if Pokemon 1 wins 26 times in 104 battles, its win rate will be $\frac{26}{104} = 0.25$.

[Q1] When calculating the win rates of the Pokemons, you may notice that some of them have not participated in any battle. Explain how you deal with them.

3.2 Task 2: Finding the Most Correlated Feature using Linear Regression

Linear regression is a basic model for regression which is expressed in the form $f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_d x_d$, where $\mathbf{w}$ denotes the parameters to be learned from the data. Note that this basic model has no hyperparameters to set.

In this task, you will build six linear regression models, where each model uses one numerical feature to predict the win rate generated in Task 1 above. The six numerical features are 'HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def', and 'Speed'.

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

[Q2] Report the validation R^2 score of each model to evaluate its prediction performance.

[Q3] After training the models with the training set, use them to make prediction on the validation set. Then, plot the regression line and the data points of the validation set for each of the six models. For illustration, Figure 1 shows a plot of the win rate versus the feature 'HP' and the regression line.

[Q4] By looking at the regression lines of the six plots, find the feature that is most correlated to the win rate. Explain how you find it.

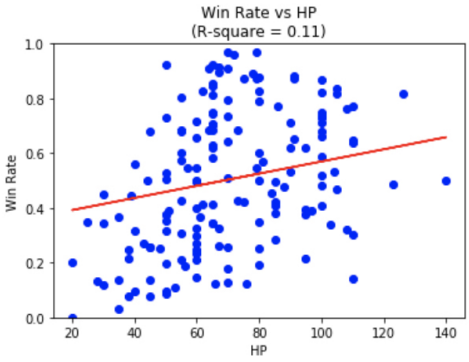

Figure 1: Win rate versus HP

3.3 Task 3: Legendary Pokemon Classification using Logistic Regression and Single-hidden-layer Neural Networks

In this task, you will build a logistic regression model as well as neural network classifiers to predict whether a Pokemon is legendary or not. You should use the following 10 features for your models: 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def', 'Speed', 'Generation', and 'Has Gender'. One-hot encoding may be used for the categorical features.

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.

3.3.1 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. This can be done automatically in `scikit-learn` when set properly.

During training, record the training time for the logistic model. After training, you are required to evaluate your model using both accuracy and the F1 score[1] on the *validation set*.

[Q5] 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 multiple times (e.g., three times) for the same hyperparameter setting and report the mean and standard deviation of the training time, accuracy, and F1 score for each setting.

3.3.2 Single-hidden-layer 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.

[1]The F1 score is the harmonic mean of precision and sensitivity. You can find this metric in `sklearn.metrics`.

For the single-hidden-layer neural network model, you need to try different number of hidden units $H \in \{1, 2, 4, 8, 16, 32, 64\}$. The hyperparameter `max_iter` can be set to 500 (default is 200). The other hyperparameters may just take their default values. During training, you are expected to record the training time of the models. After training, evaluate your models using the accuracy and the F1 score on the *validation set*. You have to report the accuracy and the F1 score for *each value* of H by plotting them using `matplotlib`.

[Q6] 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 multiple times for the same hyperparameter setting and report the mean and standard deviation of the training time, accuracy, and F1 score for each setting.

[Q7] Compare the training time, accuracy and F1 score of the logistic regression model and the best neural network model.

[Q8] Plot the accuracy and the F1 score for different values of H.

[Q9] Do you notice any trend when you increase the hidden layer size from 1 to 64? If so, please describe what the trend is.

[Q10] Referring to your experiment results, comment on the gap between accuracy and the F1 score? Suggest a reason for this observation.

3.4 Task 4: Predicting the Winners in the Pokemon Battles

In this task, you need to use grid search to tune a single-hidden-layer neural network model to predict the winner of a battle between two Pokemons, which are referred to as the first and second Pokemons. Before training, you have to organize the data frames from `Pokemon.csv` and `battles.csv` so that the model takes the features of both Pokemons as input and predicts whether the first Pokemon wins using a binary label, with values 1 and 0 indicating that the first Pokemon wins and loses, respectively.

This time, you are required to evaluate your model on the test set `q4_test.csv` provided. You need to import `q4_test.csv` as a data frame and use the data to define input features for the model like what you did for the training data.

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 search at least 10 combinations of the hyperparameter setting.

[Q11] Report 10 combinations of the hyperparameter setting.

[Q12] Report the three best hyperparameter settings as well as the mean and standard deviation of the validation accuracy of the five random data splits for each hyperparameter setting.

[Q13] Use the best model to predict the instances in the test set (`q4_test.csv`). Report the accuracy.

[**Q14**] Print the confusion matrix of the predictions on the test set.

3.5 Report Writing

Answer [**Q1**] to [**Q14**] in the report.

4 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 program clearly
- Using meaningful variable and function names to improve readability
- Using consistent styles
- Including concise but informative comments

For `scikit-learn` in particular, you are recommended to take full advantage of the built-in classes which can keep your program both short and efficient. Proper use of implementation tricks often leads to speedup by orders of magnitude. 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 Task 3 since it does not use gradient descent.

5 Assignment Submission

Assignment submission should only be done electronically using the Course Assignment Submission System (CASS):

`https://cssystem.cse.ust.hk/UGuides/cass/student.html`

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

1. **Report** (with filename `<StudentID>_report`): preferably in PDF format.
2. **Source code and a README file** (with filename `<StudentID>_code`): all necessary code for running your program as well as a brief user guide for the TA to run the programs easily to verify your results, all compressed into a single ZIP or RAR 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.

6 Grading Scheme

This programming assignment will be counted towards 15% 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)
Task 1		
- Calculate win rate	6	
- [Q1]		2
Task 2		
- Build the linear regression model	5	
- Compute the R^2 score of the 6 linear regression models + [Q2]	2	+3
- Plot the regression line and the data points for each of the 6 linear regression models + [Q3]	6	+3
- [Q4]		3
Task 3		
- Build the logistic regression model by adopting the gradient descent optimization algorithm	6	
- Compute the training time, accuracy, and F1 score of the logistic regression model + [Q5]	3	+4
- Build the single-hidden-layer neural network model	6	
- Compute the training time, accuracy, and F1 score for each value of H in the single-hidden-layer neural network model + [Q6]	3	+4
- [Q7]		3
- Plot the accuracy and F1 score with different values of H for the single-hidden-layer neural network model + [Q8]	3	+3
- [Q9]		2
- [Q10]		2
Task 4		
- Grid search on the single-hidden-layer neural network model for at least 10 combinations + [Q11]	8	+4
- Report the 3 best hyperparameter settings and the validation accuracy (both mean and standard deviation) for each setting + [Q12]	6	+3
- Report the accuracy on the test set + [Q13]	4	+2
- Record and visualize the confusion matrix on the test set + [Q14]	2	+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 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.

7 Academic Integrity

Please read carefully the relevant web pages linked from the course website.

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.