Hong Kong University of Science and Technology
COMP 4211: Machine Learning
Fall 2021

Programming Assignment 1
Due: 30 September 2021, Thursday, 11:59pm

1 Objectives

The objectives of this programming assignment are:

- To practise some data importing and preprocessing skills by using the `pandas` library in Python.
- To acquire a better understanding of supervised learning methods by using a public-domain software package called `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 medical insurance dataset provided in the form of a ZIP file (`data.zip`). There are two `csv` data files. The following table shows the attributes of the data in each `csv` file.

File	# of records	Has label?	# of columns
`train.csv`	1,000	yes	8
`test.csv`	338	yes	8

There are features in the first six columns for each person. The 7th column named ‘charge’ shows the exact insurance charge and the last column named ‘label’ indicates whether the insurance charge is high (1) or low (0).

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 for regression.

Part 3: Use the logistic regression model and single-hidden-layer neural network model 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 nth question) that you need to answer in the written report. All the experiments are expected to be done with 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 median value. Also, for the categorical features 'sex', 'smoker' and 'region', you are asked to encode them in specific ways as described in [**Q1**] below for subsequent use. After you have finished handling the above cases, visualize the correlation between every two of the nine features in the training data with a heatmap.

[**Q1**] Map the values of the 'sex' feature as 'male' = 1 and 'female' = 0, and those of the 'smoker' feature as 'yes' = 1 and 'no' = 0. For the 'region' feature which has four possible values, one-hot encoding should be used to encode it by introducing four new columns named 'northeast', 'northwest', 'southeast' and 'southwest'. These four new columns represent the one-hot encoding of the original 'region' feature. Then, the original 'region' column will be removed from the training and testing data frames to result in 11 columns. (Hint: you may use the `OneHotEncoder` class in `sklearn.preprocessing` to perform one-hot encoding.) Then, visualize the correlation between every two of the nine features with a heatmap.

5 Part 2: Regression

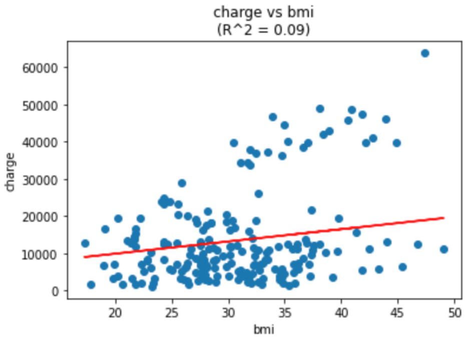
Linear regression is a basic model for regression which is expressed in the form $f(\mathbf{x}; \mathbf{w}) = w_0 + w_1 x_1 + \ldots + w_d x_d$, where $\mathbf{w}$ denotes the parameters to be learned from the training data. Note that this basic model has no hyperparameters to set.

In this task, you will build nine linear regression models in the first step, where each model uses one feature to find whether it is correlated with the attribute 'charge'. The nine features are the first nine columns in our preprocessed dataset. After this step, you will get nine models for the plotting part of [**Q3**]. Then, in the second step, you will build another linear regression model that explores the relationship between the linear combination of the nine features and 'charge'.

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**] Report the validation R^2 score of each of the first nine models to evaluate the relationship between different features and 'charge'.

[**Q3**] After training the 10 models described above 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 first nine models. For illustration, the figure below shows a plot of ‘charge’ versus the feature ‘bmi’ as well as the regression line.



6 Part 3: Classification

In this task, you will build a logistic regression model as well as neural network classifiers to predict whether a person’s medical insurance charge is high or not. You have to select the features according to some statistics and then use the selected features to complete the classification task.

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 unrelated features, we would like to choose a subset of features for classification. You can use the feature selection module in `scikit-learn`. Use three different measures (ANOVA F-value, chi-squared statistic, and mutual information) as the scores for feature selection and then drop the two least important features.

[**Q4**] Report the score for each of the nine features under three different measures. Use the scores calculated with the ANOVA F-value to perform feature selection for the following questions.

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 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.

Use the features selected in Section 6.1 above 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 F1 score[1] on the *validation set.* Remember to standardize the features before training and validation.

[**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 three times and report the corresponding mean and standard deviation of the training time, accuracy, and F1 score for each setting.

[**Q6**] Calculate the number of true positive, true negative, false positive, and false negative examples on the validation set with the last model in [**Q5**] and report them. Give one reason why we need to focus on these numbers.

6.3 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. Remember to standardize the features before training and validation.

For the single-hidden-layer neural network model, you need to try different number of hidden units $H \in \{1, 2, 8, 16, 64, 128\}$. The hyperparameter `max_iter` can be set to 500 (default is 200) and `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 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`.

[**Q7**] 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 for the same hyperparameter setting and report the mean and standard deviation of the training time, accuracy, and F1 score for each setting.

[**Q8**] Plot the accuracy and the F1 score for different values of H. Suggest a possible reason for the gap between the accuracy and the F1 score.

[**Q9**] Compare the training time, accuracy and F1 score of the logistic regression model and the best neural network model.

[**Q10**] 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 a single-hidden-layer neural network model to predict whether a person's medical insurance charge is high or not. Use the features selected in

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

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 data frame and standardize the features using the statistics you used for the training data. Remember to standardize the features in the training set as well. (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 search at least 10 combinations of the hyperparameter setting. Set the `random_state` hyperparameter of the single-hidden-layer neural network to 4211 for reproducibility and `early_stopping` to 'True' to avoid overfitting.

[**Q11**] Report 10 combinations of the hyperparameter setting.

[**Q12**] 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.

[**Q13**] Use the best model in terms of accuracy 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.

7.2 Comparison of Classification Methods

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.

[**Q14**] List one disadvantage of each of the two classification methods.

[**Q15**] For logistic regression, train a model on the training set and report the accuracy and F1 score on the test set using different learning rate scheduling types (`constant`, `optimal`, `invscaling`). The `random_state` hyperparameer is again set to 4211. Other settings are the same as those mentioned in Section 6.2.

[**Q16**] Train a single-hidden-layer neural network with 64 hidden units on the training set and report the accuracy and F1 score on the test set using different activation functions (`logistic`, `tanh`, `relu`). The `random_state` hyperparameter is again set to 4211. Other settings are the same as those mentioned in Section 6.3.

[**Q17**] For each of the activation functions used in [**Q16**], list at least one advantage of it.

8 Report Writing

Answer [**Q1**] to [**Q17**] in the report.

9 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 our logistic regression model since it does not use gradient descent.

10 Assignment Submission

Assignment submission should only be done electronically in 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, preferably in the form of 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.

11 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 score of the nine linear regression models + [Q2]	2	+3
- Make prediction on the validation set + [Q3]	3	+3
Part 3		
- Calculate the feature scores using different statistics and select the important features using ANOVA F-value scores + [Q4]	2	+2
- 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	+2
- Calculate TP, TN, FP, FN on the validation set + [Q6]	3	+3
- 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 + [Q7]	3	+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
Part 4		
- Grid search on the single-hidden-layer neural network model for at least 10 combinations + [Q11]	6	+2
- Report the three best hyperparameter settings and the validation accuracy (both mean and standard deviation) for each setting + [Q12]	6	+2
- Report the accuracy and F1 score on the test set and visualize the confusion matrix + [Q13]	4	+4
- [Q14]		2
- Logistic regression with different learning rate scheduling types + [Q15]	4	+2
- Neural networks with different activation functions + [Q16]	4	+2
- [Q17]		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.

At most one NQA coupon may be used to entitle you to submit this assignment late for one day without grade penalty. You must download the file named NQA.pdf from Canvas and submit it by the original deadline to show that you want one coupon to be used.

12 Academic Integrity

Please refer to the regulations for student conduct and academic integrity on this webpage: https://acadreg.ust.hk/generalreg.

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.