CITS5508 Machine Learning
Semester 1, 2022

Mid-Semester Test

Worth: 15%. Due: 11:59pm, Friday 1$^{\text{st}}$ April 2022

The mid-semester test this year is a take-home assignment. For each question, your answer should be around one page long (using the standard 11 or 12 points *Times New Roman* (or similar) font). Your assignment MUST be properly typed (You can use Microsoft Words, LaTeX, markdown cells in Jupyter-Notebook, or any word processing application) and converted to pdf. That is, your submitted assignment must be in pdf (Portable Document Format). Hand-drawn diagrams are fine and recommended to be used; however, they must be photographed or scanned and included in the document, NOT as separate PNG or JPG files. You should ensure that your photographs are clear (i.e., not out-of-focus, under-exposed, or over-exposed).

Please number your answer to each question and each sub-part of the question clearly. To make it easier to mark your assignment, please provide your answers in the order of the question numbers.

The total mark of the assignment is 50, which will be scaled to 15% of the total assessment.

Name your assignment as **mst.pdf** and submit it to **cssubmit** (**https://secure.csse.uwa.edu.au/run/cssubmit**) before the due date and time shown above. You can submit your file multiple times. Only the latest version will be marked. Late submissions will incur penaty, as described in https://ipoint.uwa.edu.au/app/answers/detail/a_id/2711/~/consequences-for-late-assignment-submission.

You should attempt the assignment by yourself. Collusion with other students is considered to be serious academic misconduct and can cause you to be suspended or expelled from the unit. Please see https://www.uwa.edu.au/students/my-course/student-conduct for more details.

Question 1 (10 marks)

Suppose that you have been given a training set and a test set for a complex, multiclass classification problem and that your machine learning library only has the Support Vector Machine classifier available. Describe all the steps that you would take to train and evaluate this classifier for your problem.

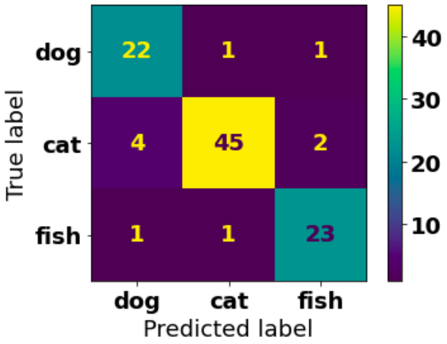
Question 2 (10 marks)

In binary classification, precision and recall are computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP, FP, and FN denote, respectively, the numbers of true positives, false positives, and false negatives. Given below is the confusion matrix from a classifier on a 3-class classification problem:



The classes are three common types of domestic pets: *dog*, *cat*, and *fish*.

(i) Describe how *average precision* and *average recall*† can be computed from a given $n \times n$ confusion matrix. Include appropriate diagrams if they help to clarify your description. (7 marks)

(ii) Supply Python code (including the *import* statement for any required libraries) for computing the average precision and average recall from the 3×3 confusion matrix shown in the diagram above. Your results need to be accurate up to the second decimal place only. Note: you will need to hard-code the elements of the confusion matrix in your code. (3 marks)

(†We want the straight average (not the weighted average) precision and recall)

Question 3 (10 marks)

The web-page below:

https://archive.ics.uci.edu/ml/datasets/Algerian+Forest+Fires+Dataset++

describes a forest fire dataset collected in Algeria in 2012. To make the data cleaning task easier, a slightly modified data file called **modified-Algerian_forest_fires_dataset.csv** is supplied on LMS for you. The dataset has 12 columns. The last column, with the heading *Classes*, contains the classes for binary classification. Our objective is to predict the *Classes* column using some other columns in the data. As this is a binary classification problem, you should inspect the *Classes* column to make sure that there are only 2 distinct values. The modified *csv* file still has a few problems. Some of the problems can be spotted if you open the file in a text editor or using *Excel*. However, all these problems can be easily fixed if you specify appropriate arguments when you read the data in using the `pandas.read_csv` function.

Describe the data cleaning process and data preparation process (starting from loading the data from the *csv* file) that you would carry out to the supplied *csv* file before a suitable machine learning classifier can be applied. You should also describe what features would be suitable to use for the classification task and how you would extract `X_train`, `y_train`, `X_test`, and `y_test`† for the training and test sets. You should supply some Python code together with your description.

(†`X_train` and `y_train` are, respectively, the feature matrix and the class label vector for the training set. `X_test` and `y_test` are similar entities for the test set.)

Question 4 (10 marks)

(i) Give an example for each of the following:

a) binary classification, (1 mark)

b) multiclass classification, (1 mark)

c) multilabel classification, and (1 mark)

d) multioutput multiclass classification. (1 mark)

Your examples MUST NOT be the same as any examples that have been mentioned in the lecture notes or in the textbook. Where relevant, you should state the number of classes and number of labels in your examples. Each of your example should include a brief description about the features and the class names.

(ii) Go through each line of the Python code below and explain what it does and what the code tries to achieve. (6 marks)

```
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from copy import deepcopy

sgd = SGDClassifier(max_iter=1, tol=1e-2, warm_start=True,
                    class_weight=’balanced’, n_jobs=-1,
                    learning_rate=’constant’, eta0=1e-2)

max_val_accuracy = 0
best_epoch = None
best_sgd = None

for epoch in range(100):
    sgd.fit(X_train, y_train)
    y_pred = sgd.predict(X_val)
    accuracy = accuracy_score(y_val, y_pred)
    if accuracy > max_val_accuracy:
        max_val_accuracy = accuracy
        best_epoch = epoch
        best_sgd = deepcopy(sgd)

print(’Best epoch =’, best_epoch)
y_pred = best_sgd.predict(X_test)
print(’Accuracy on the test set is’, accuracy_score(y_test, y_pred))
print(’The confusion matrix is:’)
print(confusion_matrix(y_test, y_pred))
```

You may assume that the data has been appropriately split into a training set (`X_train` and `y_train`), a validation set (`X_val` and `y_val`), and a test set (`X_test` and `y_test`).

Question 5 (10 marks)

(i) *Ridge regression*, *Lasso regression*, and *Elastic Net* are regularisation functions that can be added to your cost function to help overcome the overfitting problem. They all involve one or two regularisation coefficients (referred to as α and r in the textbook), which are hyperparameters that need to be optimally determined. Comment on the problems when these coefficients are too small or too large. (3 marks)

(ii) Suppose that you have implemented two Support Vector Machine regressors using a *polynomial* kernel and an *radial basis function* (RBF) kernel respectively. Suppose that you set the hyperparameter r (corresponding to the `coef0` in the `SVC` class in the Scikit-learn library) to 0. So the kernels effectively become:

Polynomial kernel of degree d: $K(\mathbf{a}, \mathbf{b}) = (\gamma \mathbf{a}^\top \mathbf{b})^d$
Gaussian RBF: $K(\mathbf{a}, \mathbf{b}) = \exp\left(-\gamma \|\mathbf{a} - \mathbf{b}\|^2\right)$

If you experience an underfitting issue in both regressors, how would you adjust the hyperparameters d and γ? Explain your answer. (4 marks)

(iii) Both the `StandardScaler` and `MinMaxScaler` classes supply the `fit`, `fit_transform`, and `transform` methods for feature scaling. Which of these functions would be suitable to use for

a) the training set?

b) the test set?

Explain your answers. (3 marks)