CITS5508 Machine Learning
Semester 1, 2021

Mid-Semester Test

Worth: 10%. Due: 11:59pm, Friday 26th March 2021

The mid-semester test this year is a take-home assignment. For each question, your answer should be around half to one page long (using the standard 11 or 12 points *Times New Roman* (or similar) font). Your assignment MUST be properly typed (using Microsoft Words, LaTeX, 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 name your submitted file as **mst.pdf**. Please number your answer to each question and each subpart 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 10% of the total assessment.

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 are given a complex, multiclass classification problem and that your machine learning library only has the Support Vector classifier. Describe all the steps that you would take to train this classifier for your problem.

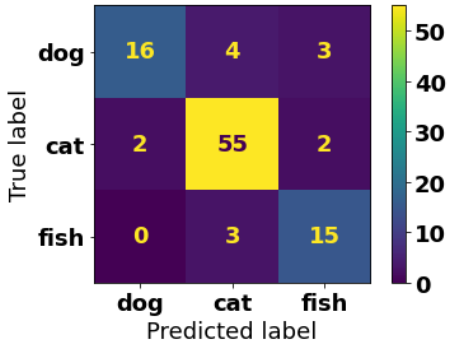
Question 2 (10 marks)

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 the *average precision* and *average recall* can be computed from the matrix. Include appropriate diagrams in your description. (8 marks)

(ii) What are the average precision and average recall of this classifier? (2 marks)

Note: we want the straight average (not the weighted average) precision and recall. You can include some Python code or just use a calculator to show your calculation. Your results only need to be accurate up to the first decimal place.

Question 3 (10 marks)

Given in the webpage below:

https://archive.ics.uci.edu/ml/datasets/Auto+MPG

is a dataset `auto-mpg.data` for regression. The dataset contains 398 instances and 9 columns. Our objective is to predict the fuel efficiency, i.e., the numbers of miles travelled per gallon of fuel (*miles per gallon*, abbreviated as *mpg* in the spreadsheet), using (all or some of) the remaining columns. Describe the data cleaning and data preparation process you would carry out before a suitable machine learning regressor can be applied. Your data cleaning and data preparation process should be specific to the dataset itself. You can write some Python code to read[1] and inspect the dataset and visualise the features but the code should not be included in your answer. To keep your answer to one page long, visualisation should be limited to one single figure (which may contain subplots).

Question 4 (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 classifiers using the *polynomial* kernel and *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 overfitting issue in both classifiers, how would you adjust the hyperparameters d and γ? Explain your answer. (4 marks)

(iii) Suppose that you need to develop a machine learning algorithm for a fire alarm system to alert people on fire incidents. For the safety of the people living in the area, you want your fire alarm system to be extremely sensitive so that it won't miss any true fire incidents. Comment on

[1] Use the `read_csv` function of `pandas` with `sep='\s+'`

a) the trade-off between the precision and recall values for your ML algorithm, (1 mark)

b) how you would like the confusion matrix to look like, and (1 mark)

c) the decision threshold that you would set in your algorithm. (1 mark)

Hint: See Figures 3-3 and 3-4 in the textbook.

Question 5 (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.

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

```
from sklearn.linear_model import SGDRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

sgd = SGDRegressor(max_iter=1, tol=-np.infty, warm_start=True,
                   penalty=None, learning_rate='constant', eta0=0.005)

min_val_err = np.float("inf")
best_epoch = None
best_sgd = None

for epoch in range(500):
    sgd.fit(X_train, y_train)
    y_pred = sgd.predict(X_val)
    err = mean_squared_error(y_val, y_pred)
    if err < min_val_err:
        min_val_err = err
        best_epoch = epoch
        best_sgd = sgd

print('Best epoch =', best_epoch)
y_pred = best_sgd.predict(X_val)
print('Mean squared error of the best model is',
      mean_squared_error(y_val, y_pred))

plt.plot(X_val, y_val, 'ro', X_val, y_pred, 'bx')
plt.legend(['ground truth', 'prediction'])
plt.show()
```

You may assume that

- the data has been appropriately split into the training set (`X_train` and `y_train`) and validation set (`X_val` and `y_val`);
- both `X_train` and `X_val` have feature dimension equal to 1.

Hint: You will need to look up the Scikit-learn library for the various functions in the code.