Master of Applied Technologies		
Course No: COMP8831	**Machine Learning**	**Level: 8 Credits: 15**

Student Name:	**Student ID:**
Assessment Type: Assignment 2	**Weighting: 10%**
Due Date and Time: 21/04/2024, 23:59	**Total Marks: 50**

Student declaration

I confirm that:

- This is an original assessment and is entirely my own work.
- The work I am submitting for this assessment is free of plagiarism. I have read and understood the Academic Integrity Procedure here. I have also read and understood the Student Disciplinary Statue here.
- Where I have used ideas, tables, diagrams etc of other writers, I have acknowledged the source in every case.

Students Signature:	**Date:**

Assignment Aims

1. To create three supervised classification learning models using Python programming language and machine learning libraries and packages which integrate the skills and knowledge gained through the second week of the course.

2. To understand the fundamentals and applications of classification machine learning algorithms and the types of problems it can solve.

Instructions

Assignment 2 **50 Marks**

You are to implement four classification models in Python: Logistic Regression, SVM, Decision Trees and Random Forrest. There are several steps for this assignment, and each step has its own mark. The total mark is 50, and it weighs 10% of your final mark.

You should submit a *.ipynp or a *.py file (only one file) that implements the following steps through the Assignment2 Submission link in Moodle (provided under Assessments, "Assignment 2 Submission Link") before the due date. The submission file should be named as follows (depending on using .py or .ipnyp):

YourStudentID_Name_ Assignment2.ipynp or YourStudentID_Name_ Assignment2.py

Here are the steps to implement the models:

The dataset used for this assignment is a set of points generated randomly with the following specification:

- Number of Samples: 1600
- Number of Classes: 3
- Number of Features: 2 (Length and Width)

1. Generate the data using the following code:

```
# X represents the 2 input features and y represents the 3 classes.
import numpy as np
from sklearn.pipeline import Pipeline
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=1600, centers=3, n_features=2, center_box=(-4.5, 4.5),
                  random_state=21)
print(X.shape)
```

2. Set the random seed number as the last two digits on your student ID (For example, Student ID = #152435, then the random seed should be 35, Student ID = #152406, then the random seed should be 6, Student ID 152400, then 1.) (1 Mark)

3. Plot the generated points with the following code or any other method of your preference:

```
# Plot the generated blobs
plt.figure(1, figsize=(8, 6))
plt.plot(X[:, 0][y==0], X[:, 1][y==0], "ys", label="Yellow Blobs")
plt.plot(X[:, 0][y==1], X[:, 1][y==1], "g^", label="Green Blobs")
plt.plot(X[:, 0][y==2], X[:, 1][y==2], "rP", label="Red Blobs")
plt.xlabel("Length", fontsize=14)
plt.ylabel("Width", fontsize=14)
plt.legend(loc="upper left", fontsize=12)
```

4. Considering X and y, is this a regression or classification problem? (0.5 Marks)
5. Split the data into training and testing sets with the approximate ratio of 4 to 1. (You can use the **train_test_split** element from **Sklearn** package.) (1 Mark)
    - **X_train** and **y_train** for the training set.
    - **X_test** and **y_test** for the testing set.

6. Train a **Logistic Regression** model using the training set. (1.5 Marks)

7. Report the prediction performance of the trained Logistic Regression model using the testing set (X-test): (1 Mark)
    - Print the accuracy of the model in percentage.
    - Print the Confusion Matrix using **confusion_matrix** element in Sklearn.
    - Print the Classification Report using **classification_report** element in Sklearn.

    (Bonus 1) Plot the decision boundaries of the trained Logistic Regression model using the training set. (0.5 Mark)

8. Train a Linear **SVM** model using the training set. (1.5 Marks)

9. Report the prediction performance of the trained **SVM** model using the testing set: (1 Mark)
    - Print the accuracy of the model in percentage.
    - Print the Confusion Matrix.
    - Print the Classification Report.

    (Bonus 2) Plot the decision boundaries of the trained SVM model using the training set. (0.5 Mark)

10. Train a non-linear (with Kernel) **SVM** model using the training set. (1.5 Marks)

11. Report the prediction performance of the trained non-linear **SVM** model using the testing set: (1 Mark)

- Print the accuracy of the model in percentage.
- Print the Confusion Matrix.
- Print the Classification Report.

12. Train a **Random Forests** model using the training set. (1.5 Marks)

13. Report the prediction performance of the trained **Random Forests** model using the testing set: (1 Mark)
    - Print the accuracy of the model in percentage.
    - Print the Confusion Matrix.
    - Print the Classification Report

    (Bonus 3) Plot the decision boundaries of the trained Random Forests model using the training set. (0.5 Mark)

14. Compare the accuracy, classification reports and confusion matrix of the four trained models and explain, in one paragraph, why you think the results are like that. Write your answer as a comment at the end of your Python code. You do not need to submit another file for this. (3 Marks).

15. Use the iris dataset to build a **Logistic Regression** classifier to detect the Iris virginica, setosa and versicolor (3 classes) based on the petal width, petal length, sepal width and petal length features (4 Features). You need to do the following steps:
    - Load the iris dataset. (0.5 Mark)
    - Split the data into 80% training and 20% testing. (0.5 Mark)
    - Train a logistic regression model (1 Mark)
    - Print the accuracy of the model in percentage. (0.5 Mark)
    - Print the Confusion Matrix. (0.5 Mark)
    - Print the Classification Report. (0.5 Mark)

16. Use the iris dataset to build a Non-Linear **SVM** classifier to detect the Iris virginica, setosa and versicolor (3 classes) based on the petal width, petal length, sepal width and petal length features (4 Features). You need to do the following steps:
    - Load the iris dataset
    - Split the data into 80% training and 20% testing
    - Train an SVM model. (1 Mark)
    - Print the accuracy of the model in percentage. (0.5 Mark)
    - Print the Confusion Matrix. (0.5 Mark)
    - Print the Classification Report. (0.5 Mark)

17. Use the iris dataset to build a **Decision Trees** classifier to detect the Iris virginica, setosa and versicolor (3 classes) based on the petal width, petal length, sepal width and petal length features (4 Features). You need to do the following steps:
    - Load the iris dataset.
    - Split the data into 80% training and 20% testing.
    - Train a Decision Trees model. (1 Mark)
    - Print the accuracy of the model in percentage. (0.5 Mark)
    - Print the Confusion Matrix. (0.5 Mark)
    - Print the Classification Report. (0.5 Mark)
    - Print the final gini values for all final nodes (2 Marks)

18. Use the iris dataset to build a **Random Forrest** classifier to detect the Iris virginica, setosa and versicolor (3 classes) based on the petal width, petal length, sepal width and petal length features (4 Features). You need to do the following steps:
    - Load the iris dataset.
    - Split the data into 80% training and 20% testing.
    - Train a Random Forrest model. (1 Mark)
    - Print the accuracy of the model in percentage. (0.5 Mark)
    - Print the Confusion Matrix. (0.5 Mark)
    - Print the Classification Report. (0.5 Mark)

19. Download the "possum.csv" file from Moodle – Topic 2, then import it to your python environment via the following code:

```
# read the csv file from a location you have copied the file
df = pd.read_csv("/specify-your-file-location/possum.csv")

#clean up and remove any rows with missing data with
df = df.dropna()

#remove the unnecessary columns, then store the features and the label data in separate variables
X = df.drop(["case", "site", "Pop", "sex"], axis=1)
y = df["sex"]
```

    - How many samples are in the dataset? (1 Mark)
    - Print the randomness factor of the dataset. (2 Marks)
    - Split the data into 80% training and 20% testing.
    - How many features are in the datasets? (1 Mark)
    - How many classes are targeted in this question? (1 Mark)
    - Train a Logistic Regression model. (1 Marks)
    - Print the accuracy of the Logistic Regression model in percentage. (0.5 Mark)
    - Train a non-linear SVM model. (1 Marks)
    - Print the accuracy of the SVM model in percentage. (0.5 Mark)
    - Train a Random Forrest model. (1 Marks)
    - Print the accuracy of the Random Forrest model in percentage. (0.5 Mark)
    - How many trees did you use on your RF model? (1 Mark)
    - How much deep did you go in your RF model (Depth_level)? (1 mark)

- What is the **max_leaf_nodes** for your RF model? (1 Mark)

20. Which model Performs better? Why do you think it is performing better? (2 Marks)

21. To know how important each feature is in predicting a possum's sex, you can use **feature_importances_**.
    feature_importances_ reflects the importance of the features with a value between 0 and 1. The higher the value the more important it is.

    - Print the importance value of every feature in your database. (2 Marks)

22. Use the Wine Recognition dataset from scikit-learn to build a classification model that predicts the origin of wines (three classes) based on their chemical properties. Perform the following tasks:
    - Load the Dataset:

```
from sklearn.datasets import load_wine
wine = load_wine()
X = wine.data
y = wine.target
```

    - How many samples does the dataset have? (0.5 Mark)
    - How many features? (0.5 Mark)
    - Split the Data into Training and Testing Sets:

```
from sklearn.model_selection import train_test_split
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

    - Compare the accuracy of Random Forest Classifier, SVM, Decision Tree and Logistic Regression on the test set. Then write a brief discussion about their performance. Consider factors like overfitting, model complexity, and interpretability (write the code too) : (4 Marks)

Late Submission of Assignments

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline
- No grade will be awarded for an assignment that is submitted later than 48hrs after the deadline

Assignments submitted in more than 48 hours late will not be marked unless Affected Performance Consideration (APC) apply. So, it is better to submit an incomplete assignment on time.

Affected Performance Consideration

A student, who due to circumstances beyond his or her control, misses a test, final exam or an assignment deadline or considers his or her performance in a test, final exam or an assignment to have been adversely affected, should complete the Affected Performance Consideration (APC) form available from Student Central or online: https://www.unitec.ac.nz/current-students/study-support/affected-performance-consideration

Assistance to Other Students

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance

- Study Groups.
- Discussion.
- Sharing reading material.
- Testing another student's programming work using the executable code and giving them the results of that testing.

Unacceptable Assistance

- Working together on one copy of the assessment and submitting it as own work.
- Giving another student your work.
- Copying someone else's work. This includes work done by someone not on the course.
- Changing or correcting another student's work.
- Copying from books, Internet etc. and submitting it as own work.

Do you want to do the best that you can do on this assignment and improve your grades? You could:

• Talk it over with your lecturer
• Visit Student Success and Achievement for learning advice and support
• Visit the Centre for Pacific Development and Support
• Visit the Centre for Maori Development and Support