**Dear all students,**

**Follow the instructions carefully, please:**

* Kindly you can find the following 3 questions you have whole **130 minutes** to answer the questions and **10 minutes of upload** time (not extendable).
* For the questions that you need to answer in the text, add your answer in this world file and upload it into Moodle. For the questions that you need to write code, upload your Python file into Moodle. Rename both files to your English name or student ID.
* You can use Slides and class code examples only.
* No need to mention using the Internet or communicating during the exam is completely prohibited.
  + Cell phones must be in the pocket **all the time**, not on the table (you need to make sure you don’t miss your file due to unexpected reset and save it into a flash drive).

**Good Luck.**

--------------------------------------------------------------------------------------------------

English Name:

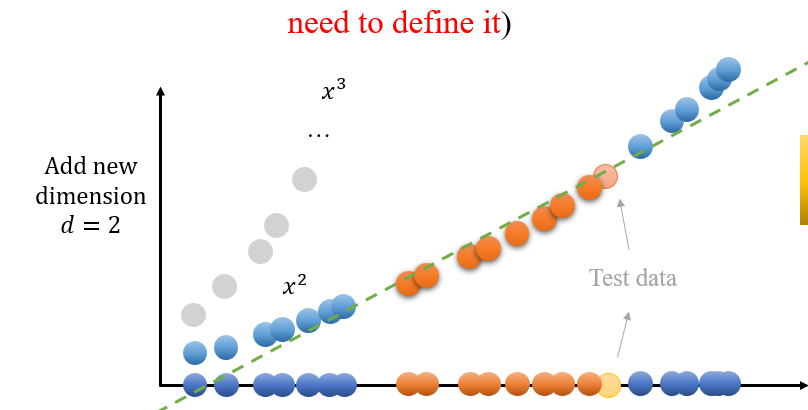
Chinese name:

Student ID:

--------------------------------------------------------------------------------------------------

**Questions:**

1. (a) What does the green dashed line in the following picture in SVM algorithm demonstrates, and what approach can generate this? Explain very short and clear.



1. Does Ensemble learning can perform better than one chosen ML algorithm for a problem? If yes how and why?

Answer:

1. (a) Why we need to write a function like Sigmoid function for Logistic Regression? Why it is necessary?
2. (b) Consider you have a problem dataset to classify data and you have 85,000 samples which 200 of them are labeled already. Explain clearly how you solve this problem? And which DL model is good option to solve this problem? Explain the general flow, support your choice, and why not the other ones.

Answer:

1. Implement a simple ANN algorithm to train following generated dataset for spam detection and report evaluation metrics for Cross-validation (Accuracy, Precision, Recall, and F1-Score, and confusion matrix) and increase models’ accuracy without overfitting. You can use libraries to train your network.

Note: data is in X, and labels are in Y variables. Make sure that your network is not underfit and overfit.

import numpy as np

num\_samples = 100

# Initialize the dataset

X = np.zeros((num\_samples, 6))

Y = np.zeros(num\_samples)

# Define patterns for spam and ham

# Feature 0: Number of links (1 for many, 0 for few)

# Feature 1: Use of special characters (1 for many, 0 for few)

# Feature 2: Length of message (1 for long, 0 for short)

# Feature 3: Number of capital letters (1 for many, 0 for few)

# Feature 4: Presence of certain keywords (1 for present, 0 for absent)

# Feature 5: Sender reputation (1 for unknown, 0 for known)

# Generate spam samples

num\_spam = num\_samples // 2

for i in range(num\_spam):

    X[i] = [

        np.random.choice([1, 0], p=[0.7, 0.3]),  # Likely many links

        np.random.choice([1, 0], p=[0.8, 0.2]),  # Likely many special characters

        np.random.choice([1, 0], p=[0.6, 0.4]),  # Likely long messages

        np.random.choice([1, 0], p=[0.6, 0.4]),  # Likely many capital letters

        np.random.choice([1, 0], p=[0.9, 0.1]),  # Keywords likely present

        np.random.choice([1, 0], p=[0.8, 0.2])   # Likely unknown sender

    ]

    Y[i] = 1  # Label as spam

# Generate ham samples

for i in range(num\_spam, num\_samples):

    X[i] = [

        np.random.choice([1, 0], p=[0.3, 0.7]),  # Likely few links

        np.random.choice([1, 0], p=[0.2, 0.8]),  # Likely few special characters

        np.random.choice([1, 0], p=[0.4, 0.6]),  # Likely short messages

        np.random.choice([1, 0], p=[0.4, 0.6]),  # Likely few capital letters

        np.random.choice([1, 0], p=[0.1, 0.9]),  # Keywords likely absent

        np.random.choice([1, 0], p=[0.2, 0.8])   # Likely known sender

    ]

    Y[i] = 0  # Label as ham

print("X:")

print(X)

print("Y:")

print(Y)

print("Shape of X:", X.shape)

print("Shape of Y:", Y.shape)

**Important Note:** Attach the Python code into Moodle (do not copy code into Word document and do not compress files as .zip or .rar files).