The Assignment:

Use the existing ML tools and packages to do the below tasks. **Note:** you don't need to implement the algorithms and methods from scratch, you can directly use the existing packages such as Python sklearn.

The submission

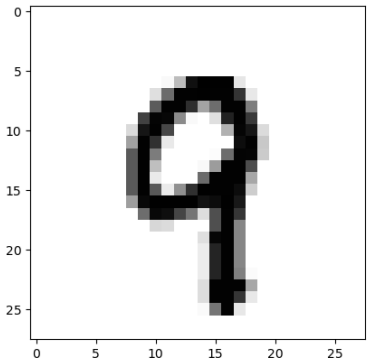
The solution should be in a word document, including the code, comments, results, tables, plots, and discussions. The submission is by email (use your name and ID as the document name). The due for submission is **12:30 PM 15/01/2023**.

Task 1: Dataset and analysis: (6 points)

In the assignment, the MINIST handwritten digits dataset will be used. The dataset consists of 70,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9. The dataset can be downloaded from Python sklearn package using this code.

```
from sklearn.datasets import fetch_openml
from matplotlib import pyplot as plt
import numpy as np

mninst = fetch_openml('mnist_784')
data   = mninst.data.to_numpy()
# The input samples as images #
dataset_images = np.reshape(data,(-1,28,28))
# The Groun truth of each sample #
y = np.array(mninst.target)
## Plot an image from the dataset ##
plt.imshow((dataset_images[350]), cmap=plt.cm.gray_r)
```



a) From each image in the dataset, extract the following features:

1) The average intensity of all pixels in the image.
2) The area of black pixels.
3) The symmetry around the x-axis.
4) The symmetry around the y-axis.

After extracting these features, the shape of the data matrix D should be (70000×4) for rows and columns respectively.

b) Calculate the correlation between feature 1 and 2. Interpret and discuss the results.

c) Using principal component analysis (PCA), visualize (i.e., by using plots) the dataset with the extracted features.

d) Randomly split the dataset (D and y) into 60% and 40% for training and testing purposes respectively.

Task 2: ML models: (9 points)

The extracted datasets (i.e., train and test) will be used to train (fit) and evaluate the following ML algorithms:

1- **Support vector machine (SVM) algorithm:**
   a. Linear SVM (soft-margin): for the value of **C**, use grid-search cross-validation to obtain the best value from the following set of values [10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001]. Use overall accuracy in the cross-validation process.
   b. SVM with RBF kernel: for the values of **C** and $\boldsymbol{\gamma}$, use grid-search cross-validation to obtain the best value from the following set of ranges (C: [10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001]; γ: [10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001]). Use overall accuracy in the cross-validation process.
   c. Using the results from both a and b parts, use the testing set to report the final evaluation result of each model; overall accuracy, and F-score as the evaluation metrics.

2- **K-nearest neighbor (KNN) algorithm:**
   a. Use grid-search cross-validation to obtain the best value **K** from the following set of values [3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25]. Use overall accuracy in the cross-validation process.
   b. Using the results from part a, use the testing set to report the final evaluation result of the KNN model; overall accuracy, and F-score as the evaluation metrics.

3- **Naive Bayes algorithm:** Fit the model by using a training dataset. Then, use the testing set to report the final evaluation result of the KNN model; overall accuracy, and F-score as the evaluation metrics.

Task 3: ML models: (5 points)

Use your results in task 2 to create useful plots and tables that can be used to compare the performance of the three algorithms. Use these plots and tables to discuss and interpret the performance of these models on this specific dataset.