FACULTY OF ENGINEERING

ACE6233 ASSIGNMENT

Machine Learning and Deep Learning

Trimester March 2024

Objectives:

- Use k-Means **clustering** algorithm for customer segmentation
- Apply Principal Component Analysis (PCA) to **reduce** the **dimensionality** of data used in a classification problem
- Develop a Convolutional Neural Network (**CNN**) for classification purposes

Introduction

Unsupervised learning is a type of machine learning where the model is not given labeled training data. Instead, the model is given only input data and is left to discover the underlying structure or patterns in the data on its own. This contrasts with supervised learning, where the model is given both input data and corresponding labeled output data. Some common unsupervised learning techniques include clustering and dimensionality reduction.

Task 1: Customer Segmentation

This task explores the combined power of unsupervised and supervised learning for customer segmentation. Firstly, k-Means clustering is applied to uncover natural groupings within a dataset of customer information. Subsequently, the cluster assignments are used as labels for the data to validate the performance of a Logistic Regression model using 5-fold cross validation.

1. Download the dataset from the following link:

https://raw.githubusercontent.com/wooihaw/datasets/main/shopping_data.csv

2. Go to Google Colab and create a new Jupyter Notebook. Name it as **Assignment_1_Clustering**.
3. Load the dataset and store it in a Pandas DataFrame.
4. Preview the first five rows of the dataset. What are the features available in this dataset?
5. Check whether there are any missing values; if there are any missing values, handle them by either removing or imputing them.
6. Store only column 2 to 4 into *X* and apply StandardScaler() to scale *X*.
7. Use the Silhouette Score method on the scaled *X* to find the optimal number of clusters, which should be between 2 and 10
8. Use the optimal number of clusters found in Step (7) to fit k-Means clustering to the scaled *X*.
9. Store the labels assigned by k-Means clustering into *y*.
10. Write necessary Python code to validate the performance of a Logistic Regression model using 5-fold cross validation.
11. Record the performance obtained in Step (10).

A total of **4 marks** are given to:

1. Correct optimal number of clusters

2. Correct implementation of k-Means clustering

3. Correct implementation of a Logistic Regression model

4. Quality of report presentation

One mark each.

Task 2: Dimensionality Reduction

This task aims to compare the performance of eight different machine learning models in classifying faulty steel plates. Each model is trained on a dataset containing information about steel plates categorized into seven fault types. To optimize model performance while minimizing computational cost, Principal Component Analysis (PCA) is used to reduce the dataset's dimensionality by half. The models' performance before and after dimensionality reduction is then compared to evaluate the effectiveness of this technique in this specific context.

1. Download the dataset from the following link:

https://raw.githubusercontent.com/wooihaw/datasets/main/steel_faults.csv

2. Go to Google Colab and create a new Jupyter Notebook. Name it as **Assignment_2_PCA**.
3. Load the dataset and store it in a Pandas DataFrame.
4. Preview the first five rows of the dataset. How many features are there in this dataset?
5. Check whether there are any missing values; if there are any missing values, handle them by either removing or imputing them.
6. Separate the dataset into features (X), targets (y) and apply standard scaling to the features.
7. Write the necessary Python codes to train and validate k-NN, Logistic Regression (LR), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Gradient Boosting Trees (GBT), and Multi-layer Perceptron (MLP) classifiers by using 5-fold cross validation.
8. Record the performance of all models trained in Step (7). Which model is the best performing model for this dataset?
9. Now, use Principal Component Analysis (PCA) to reduce the number of features by half (round down to the nearest lower integer).
10. Repeat Steps (7) and (8) for the reduced features obtained in Step (9).
11. Is there any difference in terms of the performance of the models with the new features?

A total of **4 marks** are given to:

1. Correct implementations of k-NN, LR, GNB, and SVM classifiers

2. Correct implementations of DT, RF, GBT, and MLP classifiers

3. Correct implementation of PCA

4. Performance comparison before and after PCA implementation

One mark each.

Task 3: Deep Learning

Develop a Convolutional Neural Network (CNN) that can classify different types of flowers based on some input images. By training the model on a diverse dataset encompassing various types of flowers, our aim is to create a reliable and accurate tool for flower classification.

1. The **eight_flowers.zip** dataset has been shared in the MMU OneDrive. Place the zip file in the root folder of your Google Drive.
2. Go to Google Colab and create a new Jupyter Notebook. Name it as **Assignment_3_CNN**.
3. Load the dataset by running the following codes in the notebook:

```
# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive/')
```

```
# Extract dataset from Google Drive
!cd /content/
!unzip /content/drive/MyDrive/eight_flowers.zip > /dev/null
```

4. Build a CNN that can predict different types of flowers in the dataset with the accuracy of at least 90%. You can begin from the CNN used in the lab experiment AIS2, and then modify it.

A total of **7 marks** are given to:

Correct implementation of a basic CNN with clear explanation [2 marks]

Modifications and improvements of the CNN with clear explanation [2 marks]

Results, comparisons, and discussions [2 marks]

Quality of report presentation [1 mark]

Mode of Submission

1. Assignment carries a total of **15 marks**.
2. Each student is required to submit the following items:
   i. An individual report with the following contents:
      - Results of ALL the tasks
      - Discussions of the results
      - One conclusion for each task
   ii. Jupyter Notebook file(s) with all the Python codes for all tasks, with appropriate comments/annotations.
3. Compress all the files into a single file (ZIP or RAR file).
4. Submit it at eBwise.