2024 DATA302 Techniques in Machine Learning
Assignment 3: Regression, Clustering and NNs

This assignment has 110 marks and is due at **11:59 pm, Friday, 24th May 2024**. Please submit all your answers as a single *.pdf* file including figures, tables and discussion as required, and your source code as separate files (a separate Jupyter Notebook ('.ipynb') or python file ('.py') for each of the three questions allows for a clear, organised presentation of code and outputs). Organise all your code files into a single folder, named appropriately (e.g., 'Assignment3_yourname'), and compress it before submission. Make sure you read the Assessment section at the end of the handout before writing the report. This assignment contributes **25%** to your overall course grade.

1 Linear Regression [20 marks]

This question involves using linear regression techniques to predict the fuel efficiency, measured by miles per gallon (MPG) of vehicles based on various attributes in the *Auto MPG* dataset. This analysis will help understand the influence of different vehicle characteristics such as engine size, weight, and horsepower on fuel economy. The *Auto MPG* dataset describes city-cycle fuel consumption in MPG with several car attributes such as car weight, displacement, horsepower, etc. You will use the *Auto MPG* dataset provided by seaborn.load_dataset.

Data Loading and Preprocessing

- Load the Auto MPG dataset using seaborn.load_dataset('mpg').
- Use the function train_test_split from sklearn.model_selection to perform a *80/20* split to form the training and test sets. Set the random seed to *231* for reproducibility.

Requirements:

(a) Conduct exploratory data analysis (EDA) to visualize and summarize the training set. You should include *histograms* to show distributions of variables and *scatter plots* to understand relationships between each pair of variables. *Highlight important patterns* in the report.

(b) Data Preprocessing before linear regression:

- Examine the dataset to find any *missing values*. Implement an appropriate imputation technique to manage missing data. Clearly document the method you choose for imputation.

- Examine the dataset to find any *categorical variables*. Review the encoding techniques discussed in our lectures and select the most appropriate method for each categorical variable in the dataset considering factors such as the number of categories and the ordinal nature of the data. Document the chosen encoding methods. Provide a brief justification for your choice.

(c) Construct a linear regression model to predict MPG of a vehicle using the dataset's features. Report the coefficients, the training and test performance of your model using R-squared and mean squared error (MSE).

(d) Enhance the model by applying regularisation methods including Ridge and Lasso. Compare the two enhanced models with the initial model in (c) on their training and test R-squared and MSE. Highlight important findings.

2 Clustering [25 marks]

Explore and compare the performance of K-means clustering and hierarchical clustering on a synthetic dataset to identify natural groups within the data.

Data Preparation

- use make_blobs from sklearn.datasets to generate a dataset with *four* features, *three* clusters, and *300* samples. Leave other parameters to be default values and set the random seed to *231* for reproducibility.

Requirements:

(a) Implement *K-means* clustering using sklearn.cluster.KMeans on the dataset. Determine the *best K* by evaluating the silhouette scores for various *K* values ranging from 2 to 5.

(b) Visualisation of your clusters with Principal Component Analysis (PCA): utilize PCA to reduce the dimensionality of your dataset. Specifically, project the data onto the first two principal components, which will serve as the new axes for visualisation. Construct and present a scatter plot using these two principal components. Each cluster should have a different color.

(c) Apply hierarchical clustering to the same dataset using sklearn.cluster.Agglomerative Clustering with the following linkage methods: single, complete, and average. Create dendrograms using scipy.cluster.hierarchy.dendrogram to visually represent the clusters. Ensure that each dendrogram is clearly labeled. Compare the effect of these linkage methods on creating clusters in this scenario.

(d) Discuss the advantages and disadvantages of hierarchical clustering compared with K-means clustering in this scenario. Briefly discuss the scenarios where one might be preferred over the other.

3 Neural Networks [65 marks]

This question is to show a basic understanding of neural networks by implementing a multilayer perceptron (MLP) model in PyTorch to classify handwritten digits from the Digits dataset.

The dataset contains 1,797 images of handwritten digits, each image being an 8x8 pixel grayscale image of a digit (0-9). Each image is represented as a 64-feature input vector, corresponding to the grayscale values of the pixels. As part of this question, there will be *a compulsory in-person marking for your code part* which will contribute 10 out of the total 60 marks. You will be required to demonstrate the neural network model you have developed. During the in-person marking session, you will present your code and explain your decision-making process regarding the building, and training of your neural network model.

Data Loading and Preprocessing

- Load the Digits dataset using sklearn.datasets.load_digits().
- Split the data into a training set (80%) and a testing set (20%). Set the random seed to *231* for reproducibility.
- Normalize the images by scaling the pixel values to a range of 0 to 1.
- Convert the datasets into PyTorch *tensors* and create *DataLoader* objects for both training and testing sets.

Requirments

(a) Define a neural network class by extending torch.nn.Module. The network should have one input layer, one hidden layer with 128 neurons, and one output layer. Use the ReLU activation function for the hidden layer, and the softmax activation function for the output layer. Determine the number of neurons in the input and output layers and justify your answer. Implement your neural network class accordingly.

(b) Use the torch.nn.CrossEntropyLoss for your loss function, choose an optimizer from torch.optim.SGD and torch.optim.Adam and set an appropriate learning rate, provide justifications for your choices. Train the model for 15 epochs. After each epoch, *print the training loss and accuracy*.

(c) After training, evaluate the model on the test set to measure its accuracy. Print the *test accuracy* and show *five example predictions* along with their actual labels.

(d) Discuss the impact of different learning rates on the training process. What happens if the rate is set too high or too low?

(e) Evaluate and compare the effectiveness of different activation functions including Sigmoid and Tanh in place of ReLU.

(f) Consider the network architecture, how would adding more hidden layers or changing the number of neurons in a layer affect the model's performance?

Expected Outputs (Remember to put the outputs in your report)

- **Training Output:** At the end of each training epoch, your program should display:
  - Average loss for the epoch.
  - Training accuracy for the epoch.

Example output after each epoch:

```
Epoch 1: Loss = 2.302, Accuracy = 11%
Epoch 2: Loss = 1.904, Accuracy = 32%
...
Epoch 15: Loss = 0.312, Accuracy = 90%
```

- **Testing Output:** After the model has been trained, report the overall accuracy on the test dataset. Also, include a few example images from the test set alongside their predicted and actual labels to visually demonstrate the model's performance.

  Example output:

  ```
  Test Accuracy: 88%
  ```

  Additionally, display several test images along with their predicted and actual labels to visually assess the model's performance. This can be presented in a table or as image plots with captions. For example:

  ```
  Test Image 1: Predicted Label = 3, Actual Label = 3
  Test Image 2: Predicted Label = 7, Actual Label = 7
  ...
  Test Image 5: Predicted Label = 4, Actual Label = 9
  ```

Assessment

Format: You can use any font to write the report, with a minimum of single spacing and 11 point size (hand writing is not permitted unless with approval from the lecturers). Reports are expected to be at most 8 pages that cover all the questions described above.

Late Penalty: Late submissions for assignments will be managed under the "Three Late Day Policy". You will have three automatic extension days, which can be applied to any assignments throughout the course. No formal application is required; instead, any remaining late hours will be automatically deducted when submitting assignments after the due date. You have the flexibility to use only a portion of your late day and retain the remainder for future use. Please note that these three days are for the whole course, not for each assignment. The penalty for assignments that are handed in late without prior arrangement (or use of "late days") is one grade reduction per day. Assignments that are more than one week late will not be marked. If you require any extension due to exceptional circumstances (like medical), you need to email the course coordinator.

Submission: You are required to submit a *.pdf* report and your source code files as a Jupyter notebook(.ipynb file) and/or a python code (.py) file through the web submission system from the course website *by the due time*.