University of Westminster Department of Computer Science

7BUIS008W	Data Mining & Machine Learning
Module leader	Panagiotis Chountas
Unit	Coursework 1
Weighting:	50%
Qualifying mark	40%
Description	Students are expected to critically justify the use of effective and novel data mining and machine learning techniques for a specific problem domain and definitely reflect on the knowledge of how different data mining and machine learning algorithms operate in terms of their underlying design assumptions and biases for a given problem domain. Students expected to methodically analyse the output of data mining and machine learning algorithms by drawing technically appropriate and sound conclusions resulting from the application of data mining and machine learning algorithms to the given problem
Learning Outcomes Covered in this Assignment:	This assignment contributes towards the following Learning Outcomes (LOs): • LO1 critically justify the use of effective and novel data mining and machine learning techniques for Data Science applications; • LO3 critically reflect on the knowledge on how different data mining and machine learning algorithms operate and their underlying design assumptions and biases in order to select and apply an appropriate such algorithms to solve a given problem; • LO5 critically analyse the output of data mining and machine learning algorithms by drawing technically appropriate and justifiable conclusions resulting from the application of data mining and machine learning algorithms to real-world data sets
Handed Out:	25th October 2022
Due Date	29th November 2022 Submission by 13:00 hours
Expected deliverables	Submit on Blackboard a zip file containing the required documentation (either in docx or pdf format). All implemented codes should be included in your documentation together with the results/analysis.
Method of Submission:	Electronic submission on BB via a provided link close to the submission time.
Type of Feedback and Due Date:	Feedback will be provided on BB, on 16th December 2022
BCS CRITERIA MEETING IN THIS ASSIGNMENT	• **7.1.6 Use appropriate processes** • **7.1.7 Investigate and define a problem** • **7.1.8 Apply principles of supporting disciplines** • **8.1.1 Systematic understanding of knowledge of the domain with depth in particular areas** • **8.1.2 Comprehensive understanding of essential principles and practices** • **8.2.2 Tackling a significant technical problem** • **10.1.2 Comprehensive understanding of the scientific techniques**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 50 – 59%, in which case the mark will be capped at the pass mark (50%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

Coursework Description

Comparing SKLEARN Clustering Algorithms

There are a lot of clustering algorithms to choose from. The standard `sklearn` clustering suite has thirteen different clustering classes alone. So what clustering algorithms should you be using? As with every question in data science and machine learning it depends on your data. A number of those thirteen classes in `sklearn` are specialised for certain tasks. Obviously an algorithm specializing in text clustering is going to be the right choice for clustering text data. Thus, if you know enough about your data, you can narrow down on the clustering algorithm that best suits that kind of data, or the sorts of important properties your data has. But what if you don't know much about your data? If, for example, you are 'just looking' and doing some exploratory data analysis (EDA) it is not so easy to choose a specialized algorithm.

So, what algorithm is good for exploratory data analysis?

Some rules for EDA clustering

To start, lets' lay down some ground rules of what we need a good EDA clustering algorithm to do, then we can set about seeing how the algorithms available stack up.

- **Don't be wrong!**: If you are doing EDA you are trying to learn and gain intuitions about your data. In that case it is far better to get no result at all than a result that is wrong. Bad results lead to false intuitions which in turn send you down completely the wrong path. Not only do you not understand your data, you *misunderstand* your data.
- **Intuitive Parameters**: All clustering algorithms have parameters; you need some knobs to adjust things. The question is: how do you pick settings for those parameters? If you know little about your data it can be hard to determine what value or setting a parameter should have. This means parameters need to be intuitive enough that you can hopefully set them without having to know a lot about your data.
- **Stable Clusters**: If you run the algorithm twice with a different random initialization, you should expect to get roughly the same clusters back. If you vary the clustering algorithm parameters you want the clustering to change in a somewhat stable predictable fashion.
- **Performance**: Data sets are only getting bigger. Ultimately you need a clustering algorithm that can scale to large data sizes.

1. Getting set up

If we are going to compare clustering algorithms we'll need a few things; first some libraries to load and cluster the data, and second some visualisation tools so we can look at the results of clustering.

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn.cluster as cluster
import time
%matplotlib inline
sns.set_context('poster')
sns.set_color_codes()
plot_kwds = {'alpha' : 0.25, 's' : 80, 'linewidths':0}
```

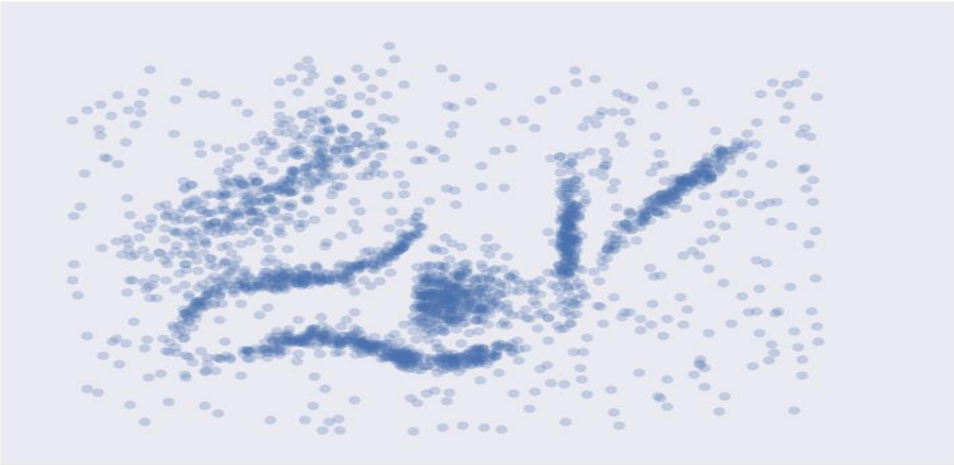
Next we need some data, download the *cluster_data.npy* from BB

This is an artificial dataset that will give clustering algorithms a challenge – some non-globular clusters, some noise etc.; the sorts of things we expect to crop up in messy real-world data. So that we can actually visualize clusterings the dataset is two dimensional; this is not something we expect from real-world data where you generally can't just visualize and see what is going on.

```
data = np.load('cluster_data.npy')
```

So let's have a look at the data and see what we have.

```
plt.scatter(data.T[0], data.T[1], c='b', **plot_kwds)
frame = plt.gca()
frame.axes.get_xaxis().set_visible(False)
frame.axes.get_yaxis().set_visible(False)
```



It's messy, but there are certainly some clusters that you can pick out by eye; determining the exact boundaries of those clusters is harder of course, but we can hope that our clustering algorithms will find at least some of those clusters. So, on to testing with SKlearn Clustering Algorithms

2. Testing Clustering Algorithms

To start let's set up a little utility function to do the clustering and plot the results for us. We can time the clustering algorithm while we're at it and add that to the plot since we do care about performance.

```
def plot_clusters(data, algorithm, args, kwds):
    start_time = time.time()
    labels = algorithm(*args, **kwds).fit_predict(data)
    end_time = time.time()
    palette = sns.color_palette('deep', np.unique(labels).max() + 1)
    colors = [palette[x] if x >= 0 else (0.0, 0.0, 0.0) for x in labels]
    plt.scatter(data.T[0], data.T[1], c=colors, **plot_kwds)
    frame = plt.gca()
    frame.axes.get_xaxis().set_visible(False)
    frame.axes.get_yaxis().set_visible(False)
    plt.title('Clusters found by {}'.format(str(algorithm.__name__)), fontsize=24)
    plt.text(-0.5, 0.7, 'Clustering took {:.2f} s'.format(end_time - start_time),
fontsize=14)
```

Before we try doing the clustering, there are some things to keep in mind as we look at the results.

- In real use cases we *can't* look at the data and realise points are not really in a cluster; we have to take the clustering algorithm at its word.
- This is a small dataset, so poor performance here bodes very badly.
- **To run KMeans for example you only have to call the** `plot_clusters(data, algorithm, args, kwds)` function in Python, executing
  plot_clusters(data, cluster.KMeans, (), {'n_clusters':6})

Tasks & Marking Scheme :

1. Set the Python environment for testing the sklearn clustering **algorithms for performing exploratory data analysis outlined in sections 1,2**.
   - Load the 'cluster_data.npy'

     [2 Marks]
   - Scatter plot the 'cluster_data.npy'

     [3 Marks]
   - Establish the plot_clusters(data, algorithm, args, kwds)little utility function to do the clustering and plot the results for us

     [5 Marks]

   [10 Marks]

2. Advice the **sklearn.cluster** and set the (algorithm, args, kwds) values for the following clustering **algorithms. You need to justify your chosen values.**
   - K-Means

     [2.5 Marks]
   - Affinity Propagation

     [3 Marks]
   - Mean Shift

     [3 Marks]
   - Spectral Clustering

     [3 Marks]
   - Agglomerative Clustering

     [2.5 Marks]
   - HDBSCAN

     [3 Marks]

   [14 Marks]

3. Test the following sklearn clustering algorithms **for performing exploratory data analysis** and plot the output.
   - K-Means

     [1 Mark]
   - Affinity Propagation

     [1 Mark]
   - Mean Shift

     [1 Mark]
   - Spectral Clustering

     [1 Mark]
   - Agglomerative Clustering

     [1 Mark]
   - HDBSCAN

     [1 Mark]

   [6 Marks]

4. Critically summarise the quality of the obtained clusters for each clustering algorithm. Your discussion should make reference to

    - the given domain;

      [3 Marks]
    - the specified (algorithm, args, kwds) values as part of the plot_clusters(data, algorithm, args, kwds) employed function;

      [7 Marks]
    - intra-cluster versus inter-cluster distance;

      [5 Marks]
    - taken scatter plots after applying each clustering algorithm.

      [5 Marks]

Present your findings for the above questions in the form of a technical report. The paper must express your own conclusions and findings. The paper size should be between [950-1250] words, excluding any references, plots. Papers violating the lower limit or exceeding the upper limit of allowable words will be subject to a penalty of 10%, (2 Marks out of 20)

[20 Marks]

.

Total [50 Marks]