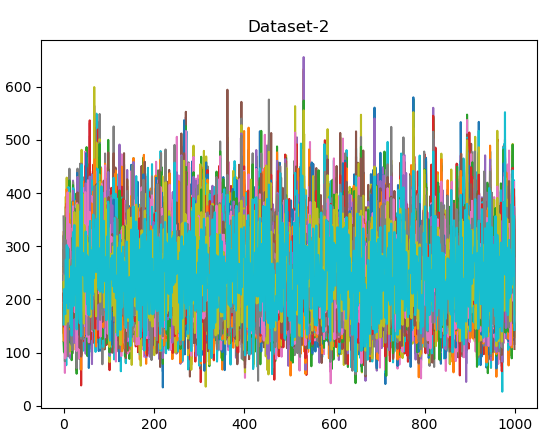
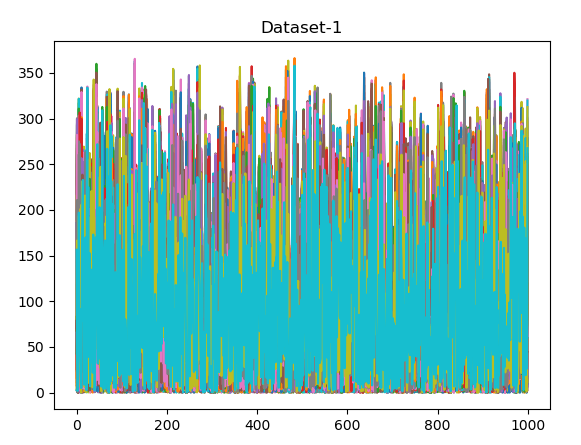
**CS 535 Introduction to Data Mining (Fall-2019) Assignment-1**

**Mohi Uddin Syed**

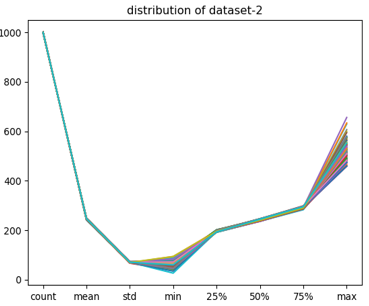
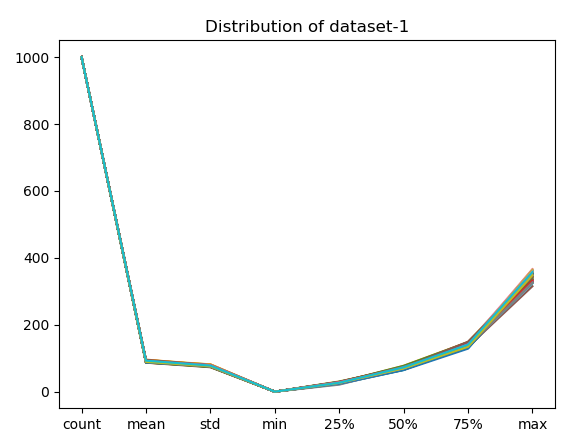
**B00766015**

1. From the following graph of both datasets, we can see that both datasets are almost equally distributed. Dataset 2 has a slight variance .

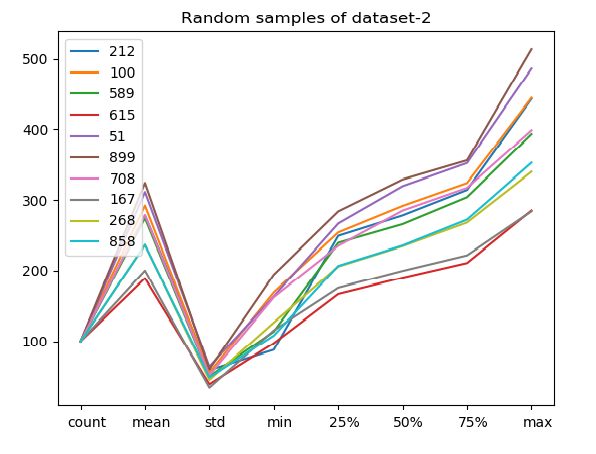
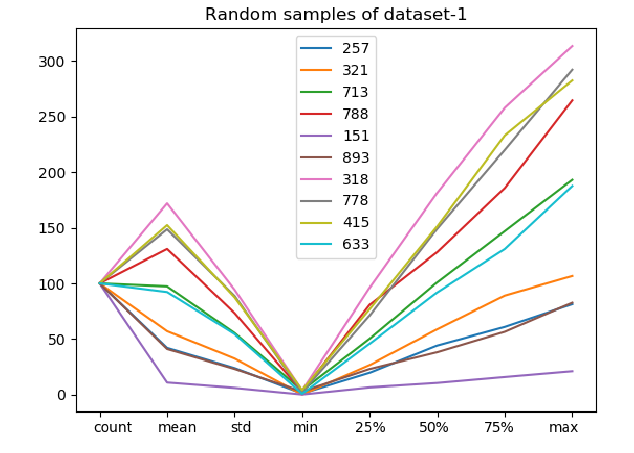
In the first two plots, X-axis represent number of rows and Y-axis represents values of the rows.



In these distributions, Y-axis represents the number of rows.



Distribution for 10 random samples for each dataset



1. **& 3.** Feature Extraction results for both datasets

For dataset-1:

To get the percentage of information spread across each component of a compressed dct frame, I summed each column and divided the sum of each column by the total sum of all columns. Since, higher the value of a dct frame cell, more information can be extracted from it that contributes towards the original dataset.

|  |  |
| --- | --- |
| **PCA** | **DCT** |
| 1. 90% of the original information can be extracted by 51 components. 2. 95% of the original information can be extracted by 59 components 3. for 85% of the information 43 components are required 4. and 68 components for 99% of information retained. | 1. Roughly 90% of the information is compressed in just the first column after applying dct, rest 99 components have the remaining 10% spread almost equally, i.e 0.1% in each column. 2. Since the remaining components have 0.1% in each component, to extract 95% of the data, 51 components would be needed |

For dataset-2:

|  |  |
| --- | --- |
| **PCA** | **DCT** |
| Similar results can be observed for dataset-2 with slight differences: for 90% of the information retained, 50 components are required, for 95% 59 components and for 85% 42 components | Dct also has similar results for dataset-2 where almost 90% of the information is compressed in just the first component and the rest spread roughly equal among others. |

**Conclusion:** If we consider both space and information retained, DCT performs better on the given data sets, because it significantly reduces the space consumption while retaining 90% of the original information. PCA performs better only when it is required to retain information close to 100%.

**4. ICA vs PCA:**

ICA is used for separation of components so that the mutual information among all the components is zero. ICA does not do compression. PCA is used to reduce the dimensionality of the dataset while retaining as much information as possible. PCA does compression.

For dataset-1:

PCA significantly reduces the dimensions of the array while retaining most of the information, for example to retain 95% of the original information, PCA reduces the dimension of the matrix to 143x59 from 1000x100, this is a huge improvement. On the other hand, even though ICA compresses and separates the matrix values to a smaller range, while reducing the columns, it still uses the same amount of rows/vectors.

If dimensionality reduction is the goal, PCA outperforms ICA by a huge margin. ICA on the other hand should be preferred for applications where finding independence among the components is the goal and dimensionality reduction is not an issue.

For dataset-2:

PCA and ICA gives similar results for dataset-2.

NOTE: I converted the two datasets into csv files(out1.csv and out2.csv) using the code in convert.py