CONTINUOUS ASSESSMENT

PYTHON DATA VISUALIZATION EXERCISE

Contents

[1 Description 3](#_Toc38389741)

[2. Aim 3](#_Toc38389742)

[3. Justification 3](#_Toc38389743)

[3.1 Choice of Visualization Technique 3](#_Toc38389744)

[4. Methodology 4](#_Toc38389745)

[4.1 Implementation 4](#_Toc38389746)

[5. Testing and Data Exploration 6](#_Toc38389747)

[5.1 Result Analysis 9](#_Toc38389748)

[6. Conclusion 9](#_Toc38389749)

# 1 Description

The given dataset is called “Clients.csv”, the data has 9 columns and 43,193 rows. Each row in the dataset represent a client which is describe by 9 variables. The variables and their properties are shown in the table below:

|  |  |  |
| --- | --- | --- |
| S/N | Variable Name | Description |
| 1 | age | (numeric) |
| 2 | job | type of job (categorical: "admin.", "unemployed", "management", "housemaid",  "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services") |
| 3 | marital | marital status (categorical: "married", "divorced", "single"; note: "divorced" means  divorced or widowed) |
| 4 | education | (categorical: "secondary", "primary", "tertiary") |
| 5 | default | has credit in default? (binary: "yes", "no") |
| 6 | balance | average yearly balance, in euros (numeric) |
| 7 | housing | has housing loan? (binary: "yes", "no") |
| 8 | personal | has personal loan? (binary: "yes", "no") |
| 9 | term | has term deposit? (binary: "yes", "no") |

# 2. Aim

The aim of this exercise is to Visualize natural groupings or clusters in the given datasets using dimensionality-reduction & clustering algorithms. The goal is to interpret groupings to extract meaningful insights from the data. The T-Distributed Stochastic Neighbor Embedding technique (T-NSE) is used to implement the visualization.

# 3. Justification

## 3.1 Choice of Visualization Technique

To produce the best visualization the T-distributed stochastic neighbor embedding visualization technique is used, this is due to its ability to accurately represent multidimensional data in 2-dimensional space with minimal loss of data. Unlike the Principal Component Analysis technique, the T-SNE method can present the natural clustering of a complex data without placing overlapping data.

Advantages of the T-SNE dimensionality reduction technique

1. Handles Non-Linear Data Efficiently
2. Preserves Local and Global Structure

These advantages make the T-SNE the most suitable technique for the data because of its complex structure.

# 4. Methodology

All methods and steps implemented in this exercise were carefully examined and justified. The steps taken in the implementation of the python code are explained in the following section.

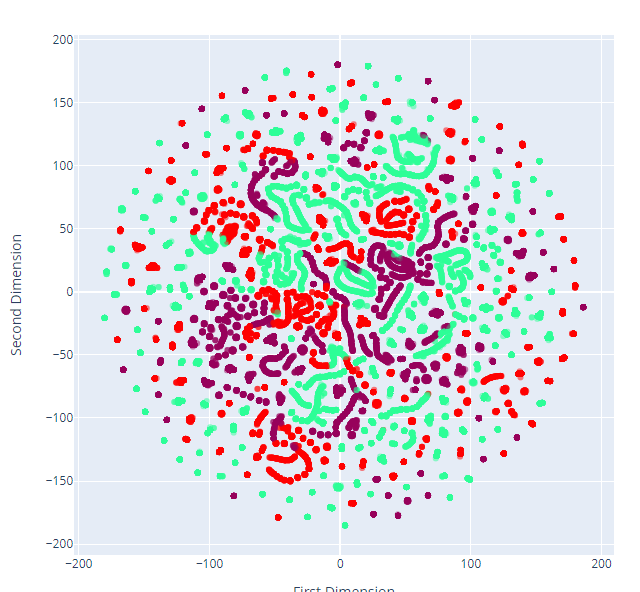
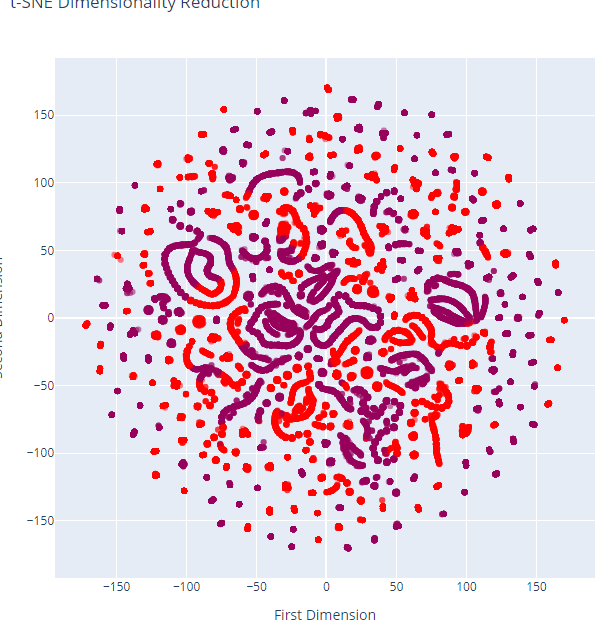
## 4.1 Implementation

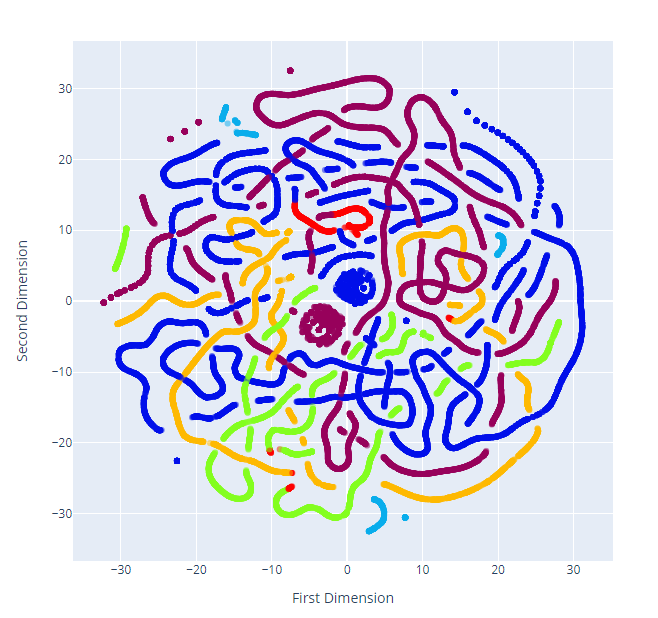
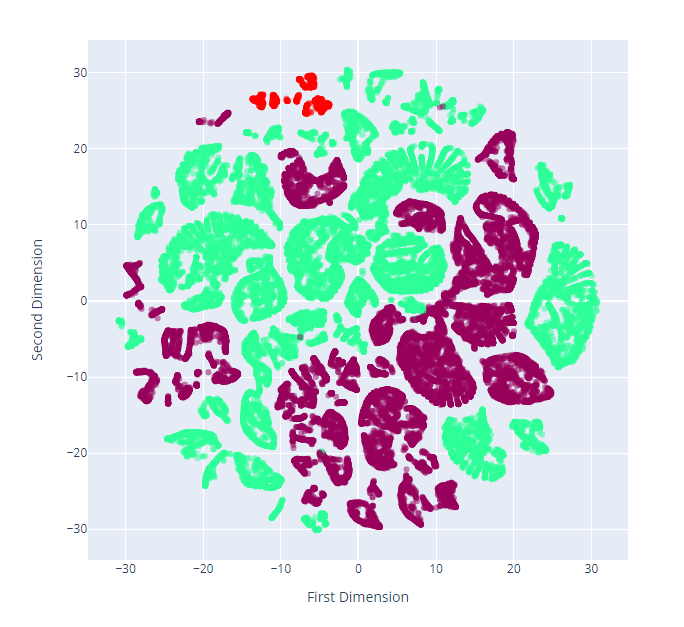
1. **Importation of Library**: Specific python data science library were selected to enable analysis and visualization of data. The libraries used include:
   1. Pandas: This is one the most powerful and basic python data science libraries, it enables reading of dataset such as the comma separated values (CSV) used in this exercise. It also enables other operation such as saving dataframes, data conversion among others.
   2. Numpy: This library enables easy Mathematical operations and manipulations of multidimensional arrays.
   3. ScikitLearn: This library features numerous functions and classes that supports machine learning and data analytics techniques such as regression and classification.
   4. Matplotlib: This library enables data visualization; it leverages power python GUI libraries to display plots which enables us to understand pattern within out dataset. It can be used to make plots ranging from simple line plots to complex 3D plots.
   5. Plotly: This is very powerful data visualization tool. Its advantage over the matplotlib is its ability to enable interaction with the plot which is essential for the T-SNE plot.
2. **Data acquisition and Cleaning**: After importing the library, the pandas library is used to read the “csv” file and convert it to a data frame. Next the summary of the data is generated using commands such as dataset.head(), dataset.info() and dataset.describe() among others. These gives us an insight into the structure of the data, revealing, Means, Standard Deviation, Minimum and Maximum values among other. Any anomaly in the data is cleaned at this stage.
3. **Data Encoding**: To make data processing easier, it is necessary to convert categorical data such as the containing a “yes” and “no” values to numerical values like “1” and “0” to enable numerical operations to be performed on them. The Age and Balance were also encoded to make analysis easier.
4. **Data Correlation**: Data encoding is followed by generating correlation diagram which helps us to determine if there are any correlation among different columns in the dataset, this will helps us to decide whether to remove a column or not. Columns with absolute correlation values greater than 0.5 are considered to be high, and such columns are discovered to have causative relationship, then one of the columns can be removed to prevent data redundancy.
5. **Splitting Data into Subsets:** To allow exploratory analytics, the columns are split into subsets so as to reveal existing natural pattern and clusters within the data. A particular may or may not generate clear patterns, this is dependent on the combination of columns that make up the subset.
6. **Generating Elbow Plot:** To get an idea of how many clusters to expect in the dataset, it is necessary to generate an elbow diagram. The elbows diagram generates an inertia plot, and the point on the plot that has the greatest change in slope indicates the likely number of clusters within the data. The observed number from the elbow plot is inputted in the int K-means function which is later used to label the TSNE-plot.
7. **Generating the T-SNE plot**: The effectives of the T-SNE plot in displaying clusters depends on two hyperparameters which are the perplexity value and step value (number of iteration), the values selected for this exercise are 50 and 1000 respectively.

The labels generated from the K-mean in the previous step is fed to the T-SNE plot, this is used to label the plot. Plotly was used to generate and store the plot file in HTML format

# 5. Testing and Data Exploration

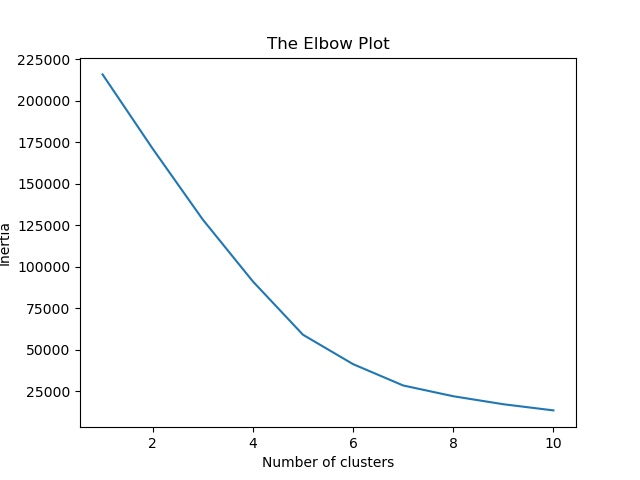
The data was split into 3 subsets. Initially, the splitting was based on personal and financial data but the result obtained were not showing any patterns. Below are some of the results obtained with the parameters used.



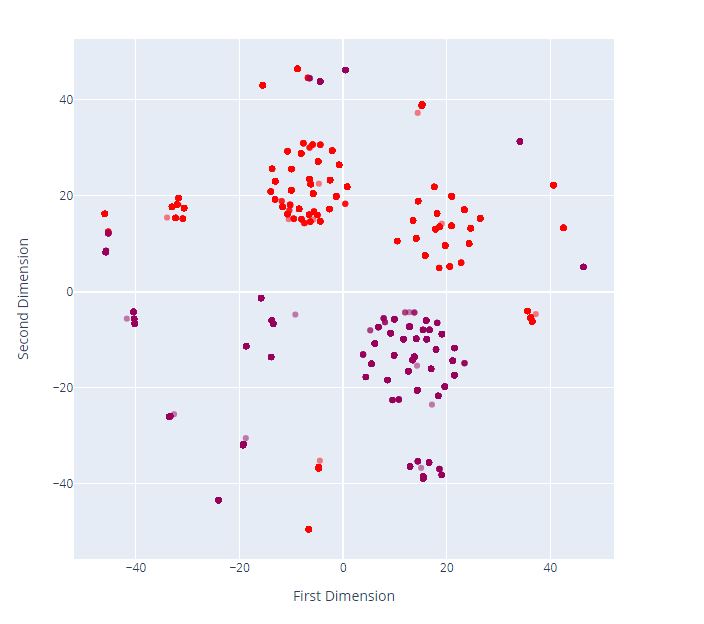
*Different results obtained while testing*

After exploring the data set with different combination of perplexities, iterations and subsets, the only combination that produced a fairly good cluster was a combination of columns with discreet values which include; age, term, default, balance, housing and personal. Below is the snapshot of the elbow diagram:



*Elbow Diagram*

Although, the elbow indicated the number of clusters should be 5 or thereabout, the actual cluster number that produced fairly good visualization was 2 - even though multiple sub-clusters were identified. The perplexity and iteration values that were used for the plot are 2 and 1000 respectively. The result of the visualization is show in the following figure:



*Final Result of TSNE Plot*

The plot above is the result of the TSNE plot of the data set, two major clusters were identified, they colored in the red and purple. The purple color indicates where Housing loan value is 1 while red indicates where client has no housing loan, which is denoted by 0. There some notable sub-clusters within each cluster. Breakdown of clusters and sub-clusters are shown in the table below:

## 5.1 Result Analysis

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S/N | Major Cluster | Sub Cluster | Housing | Default | Balance | Personal | Term | Age | color |
| 1 | No Housing  Loan | 1 | 0 | 0 | 1 | 0 | 0 | 1 | Red |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 0 | 1 | 1 | 1 | 1 |
| 2 | Housing Loan | 1 | 1 | 0 | 1 | 0 | 0 | 1 | Purple |
|  | 2 | 1 | 0 | 1 | 1 | 0 | 0 |
|  | 3 | 1 | 0 | 1 | 0 | 0 | 0 |
|  | 4 | 1 | 0 | 1 | 1 | 0 | 1 |
|  | 5 | 1 | 0 | 1 | 0 | 1 | 1 |
|  | 6 | 1 | 1 | 1 | 1 | 0 | 1 |

*Result Table*

# 6. Conclusion

The dataset is a complex one with weak correlations between columns. After several combinations of subsets, perplexity and iteration steps, we were able to reveal a weak clustering in the data, the line of distinction between the clusters is housing loan which produced two major clusters. Each cluster has about six different sub-clusters which further divides the data into about 12 sub-clusters.