Big Data

LAB-02

Timings: 11:30-2:30

Lab Protocols:

1. Carefully read and follow all instructions
2. You can search the basics of python, concepts, and syntax online
3. No evaluation would be done after Lab's timing. So, keep the track of time.
4. Do keep in mind that sharing the code, discussing it during lab or looking for online solution is highly unethical, and all actions would be considered as plagiarism.
5. Plagiarism will result in serious penalty

Task1 – Do some preprocessing steps on data. Visualize your results (10 Marks)

Part (A): (2 Marks)

Perform The following necessary tasks for data cleaning.

1. Load the csv file
2. Display last 10 rows
3. Count null values in data
4. Perform suitable action in null values (Justify it later)
5. Take care of any duplicate values
6. Remove any unnecessary feature from the data & convert string data to numbers.

Basics

Name	Usage	Comments
df.head()	Preview the first n (default=5)	Can define "n" by df.head(n
df.tail()	Preview the last n (default=5)	Can define "n" by df.tail(n)
df.sort_values()	Sort the data frame on a specific column	Can sort with multiple colur
df.columns	Display all column names	Can also set column names
df.dtypes	Display data types of the columns	Return a list of types
df.shape	Display the shape of the data frame	Return a tuple: (row_count,
df.describe()	Show basic stats of each column	Will show different stats for
s.value_counts()	Count occurrences of each value	Use df['...'] to get a column

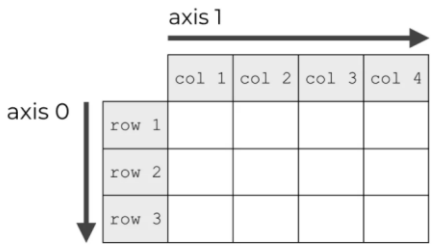
Cheat sheet

Name	Usage	Comments
s.isna()	Whether there are null values existing	Boolean
df.dropna()	Delete missing values	Can be applied on a series or
df.fillna()	Fill missing values with a certain value	Can be applied on a series or
df.drop_duplicates()	Delete all duplicated values	Can be applied on a series or

Hint:

```
df.drop(to_drop, inplace=True, axis=1)
or df.drop(columns=to_drop, inplace=True)
```

Python Axis:



Part (B): (3 Marks)

Compute PCA of the data to compute 2 dimensions.

Resource:
https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

Hint:

```
from sklearn.decomposition import PCA
pca = PCA()
components = pca.fit_transform(df)
c1=components[:,0]
c2=components[:,1]
```

Part (C): (5 Marks)

Show the results through visualization

Instructions of plotting Data: Use matplotlib's scatter plot for plotting. Use subplot functionality, make the subplot of size (6,6). Appropriate set labels to all axis and set label to complete scatter plot.

Hint

```
ax.scatter(c1, c2)
```

Task 2 – Apply Built in K mean Method (15 Marks)

Part (A): (5 Marks)

Do k-mean clustering on the data and pick k centers with hit and trial method.
Then Visualize the results by placing X at cluster centers. (Use the above Matplotlib functionality)

Part (B): (5 Marks)

Use TSNE for visualization (Note y= prediction of clusters)
https://www.datatechnotes.com/2020/11/tsne-visualization-example-in-python.html

example:

```
tsne = TSNE(n_components=2, verbose=1, random_state=123)
z = tsne.fit_transform(x)
```

```
df = pd.DataFrame()
df["y"] = y
df["comp-1"] = z[:,0]
df["comp-2"] = z[:,1]

sns.scatterplot(x="comp-1", y="comp-2", hue=df.y.tolist(),
                palette=sns.color_palette("hls", 3),
                data=df).set(title="Iris data T-SNE projection")
```

Part (C): (10 Marks)

Do k-mean clustering on the data and pick k centers with elbow Method. Custom implement, within sum of square functionality
Formula= (point- center) **2 for every point
Visualize the curve using matplotlib plotting.

Bonus – Do custom Implementation of k-mean algorithm

You can use pairwise_distacne_argmin functionality to generate pair wise distance

Hint1: (randomly sort data frame)

```
np.random.seed(42)
rndperm = np.random.permutation(df)
```


Keep Learning.
Beauty in code
comes with
experience.