

```

1  import warnings
2  warnings.filterwarnings('ignore')
3
4  # Importing all required packages
5  import numpy as np
6  import pandas as pd
7
8  # Data viz lib
9  import matplotlib.pyplot as plt
10 import seaborn as sns
11 %matplotlib inline
12 from matplotlib.pyplot import xticks

```

```
1 bank = pd.read_csv('/content/bankmarketing.csv')
```

```
1 bank.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_month
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	1
1	57	services	married	high.school	unknown	no	no	telephone	may	1
2	37	services	married	high.school	no	yes	no	telephone	may	1
3	40	admin.	married	basic.6y	no	no	no	telephone	may	1
4	56	services	married	high.school	no	no	yes	telephone	may	1

5 rows × 21 columns

```

1 bank_cust = bank[['age', 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'day_of_month']]
2 bank_cust.head()

```

	age	job	marital	education	default	housing	loan	contact	month	day_of_month
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	1
1	57	services	married	high.school	unknown	no	no	telephone	may	1
2	37	services	married	high.school	no	yes	no	telephone	may	1
3	40	admin.	married	basic.6y	no	no	no	telephone	may	1
4	56	services	married	high.school	no	no	yes	telephone	may	1

```

1 bank_cust['age_bin'] = pd.cut(bank_cust['age'], [0, 20, 30, 40, 50, 60, 70, 80, 90, 100],
2                               labels=['0-20', '20-30', '30-40', '40-50', '50-60', '60-70', '70-80', '80-90', '90-100'])
3 bank_cust = bank_cust.drop('age', axis = 1)

```

```
1 bank_cust.head()
```

	job	marital	education	default	housing	loan	contact	month	day_of_week
0	housemaid	married	basic.4y	no	no	no	telephone	may	me
1	services	married	high.school	unknown	no	no	telephone	may	me
2	services	married	high.school	no	yes	no	telephone	may	me
3	admin.	married	basic.6y	no	no	no	telephone	may	me
4	services	married	high.school	no	no	yes	telephone	may	me

```

1 from sklearn import preprocessing
2 le = preprocessing.LabelEncoder()
3 bank_cust = bank_cust.apply(le.fit_transform)
4 bank_cust.head()

```

	job	marital	education	default	housing	loan	contact	month	day_of_week	pout
0	3	1	0	0	0	0	1	6	1	
1	7	1	3	1	0	0	1	6	1	
2	7	1	3	0	2	0	1	6	1	
3	0	1	1	0	0	0	1	6	1	
4	7	1	3	0	0	2	1	6	1	

```
1 bank_cust_copy = bank_cust.copy()
```

```
1 pip install kmodes
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/>

Collecting kmodes

Downloading kmodes-0.12.2-py2.py3-none-any.whl (20 kB)

Requirement already satisfied: numpy>=1.10.4 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: scikit-learn>=0.22.0 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: scipy>=0.13.3 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages

Installing collected packages: kmodes

Successfully installed kmodes-0.12.2

```
1 from kmodes.kmodes import KModes
```

```

1 km_cao = KModes(n_clusters=2, init = "Cao", n_init = 1, verbose=1)
2 fitClusters_cao = km_cao.fit_predict(bank_cust)

```

```

Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 5322, cost: 192203.0
Run 1, iteration: 2/100, moves: 1160, cost: 192203.0

```

```
1 fitClusters_cao
```

```
array([1, 1, 0, ..., 0, 1, 0], dtype=uint16)
```

```
1 clusterCentroidsDf = pd.DataFrame(km_cao.cluster_centroids_)
2 clusterCentroidsDf.columns = bank_cust.columns
```

```
1 clusterCentroidsDf
```

	job	marital	education	default	housing	loan	contact	month	day_of_week	pout
0	0	1	6	0	2	0	0	6	2	
1	1	1	3	0	0	0	1	6	0	

```
1 km_huang = KModes(n_clusters=2, init = "Huang", n_init = 1, verbose=1)
2 fitClusters_huang = km_huang.fit_predict(bank_cust)
```

```
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 3724, cost: 195568.0
```

```
1 fitClusters_huang
```

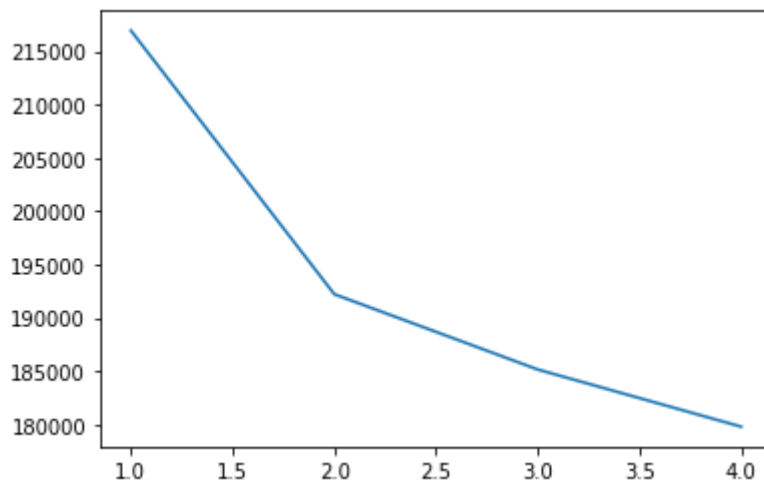
```
array([1, 1, 1, ..., 0, 0, 0], dtype=uint16)
```

```
1 cost = []
2 for num_clusters in list(range(1,5)):
3     kmode = KModes(n_clusters=num_clusters, init = "Cao", n_init = 1, verbose=1)
4     kmode.fit_predict(bank_cust)
5     cost.append(kmode.cost_)
```

```
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 0, cost: 216952.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 5322, cost: 192203.0
Run 1, iteration: 2/100, moves: 1160, cost: 192203.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 4993, cost: 185138.0
Run 1, iteration: 2/100, moves: 1368, cost: 185138.0
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 6186, cost: 179774.0
Run 1, iteration: 2/100, moves: 1395, cost: 179774.0
```

```
1 y = np.array([i for i in range(1,5,1)])
2 plt.plot(y,cost)
```

[<matplotlib.lines.Line2D at 0x7fca833f53d0>]



```
1 km_cao = KModes(n_clusters=2, init = "Cao", n_init = 1, verbose=1)
2 fitClusters_cao = km_cao.fit_predict(bank_cust)
```

```
Init: initializing centroids
Init: initializing clusters
Starting iterations...
Run 1, iteration: 1/100, moves: 5322, cost: 192203.0
Run 1, iteration: 2/100, moves: 1160, cost: 192203.0
```

```
1 fitClusters_cao
```

```
array([1, 1, 0, ..., 0, 1, 0], dtype=uint16)
```

```
1 bank_cust = bank_cust_copy.reset_index()
```

```
1 clustersDf = pd.DataFrame(fitClusters_cao)
2 clustersDf.columns = ['cluster_predicted']
3 combinedDf = pd.concat([bank_cust, clustersDf], axis = 1).reset_index()
4 combinedDf = combinedDf.drop(['index', 'level_0'], axis = 1)
```

```
1 combinedDf.head()
```

	job	marital	education	default	housing	loan	contact	month	day_of_week	pout
0	3	1	0	0	0	0	1	6	1	
1	7	1	3	1	0	0	1	6	1	
2	7	1	3	0	2	0	1	6	1	
3	0	1	1	0	0	0	1	6	1	
4	7	1	3	0	0	2	1	6	1	

