Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

Радиотехнический факультет (РТ)

Отчёт по рубежному контролю №2

По дисциплине

«Технологии машинного обучения»

Проверил:	Выполнил:
Преподаватель кафедры ИУ-5	студент группы РТ5-61Б
Гапанюк Ю.Е.	Ануров Н.С.
Подпись: _____________	Подпись: _____________
«___» _____________ 2020 г.	«___» _____________ 2020 г.

Москва, 2020

Ануров Никита РТ5-61

Рубежный контроль №2

In [1]:
```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from IPython.display import Image
from sklearn import cluster, datasets, mixture
from sklearn.neighbors import kneighbors_graph
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import adjusted_mutual_info_score
from sklearn.metrics import homogeneity_completeness_v_measure
from sklearn.metrics import silhouette_score
from itertools import cycle, islice
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.cluster import MeanShift
from sklearn.cluster import DBSCAN
```

In [2]:
```
df=pd.read_csv('toy_dataset.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Number	City	Gender	Age	Income	Illness
0	1	Dallas	Male	41	40367.0	No
1	2	Dallas	Male	54	45084.0	No
2	3	Dallas	Male	42	52483.0	No
3	4	Dallas	Male	40	40941.0	No
4	5	Dallas	Male	46	50289.0	No

In [14]:
```
data=df[['Age','Income']]
data.head()
```

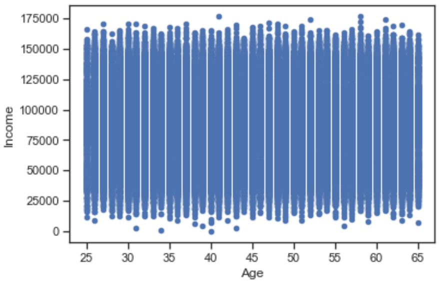
Out[14]:

	Age	Income
0	41	40367.0
1	54	45084.0
2	42	52483.0
3	40	40941.0
4	46	50289.0

In [15]:
```
data.reset_index().plot.scatter(x = 'Age', y = 'Income')
```

```
'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping will have precedence in
case its length matches with 'x' & 'y'.  Please use a 2-D array with a single row if you really want to specify the same RGB or
RGBA value for all points.
```

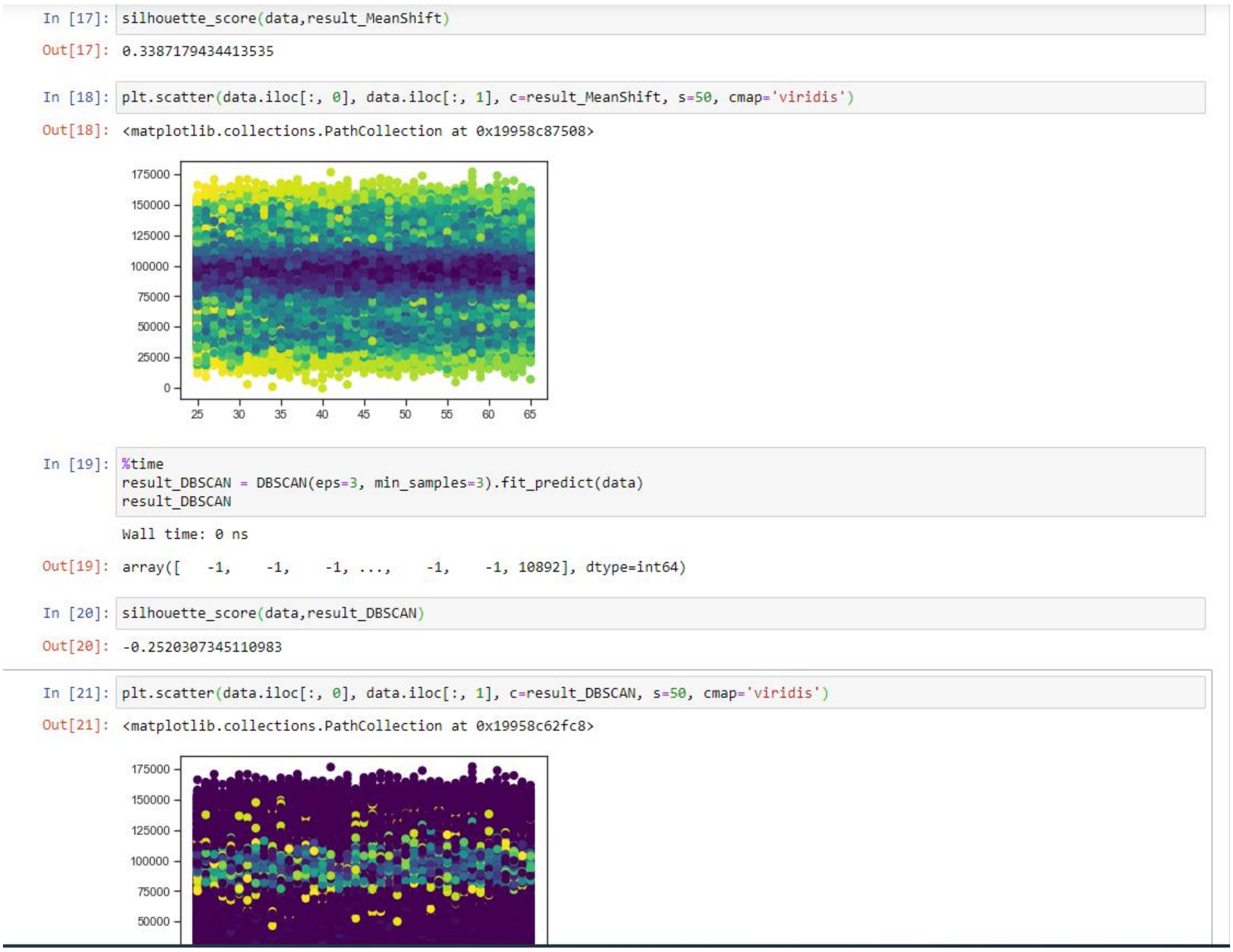
Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1995a5a1cc8>
```



In [16]:
```
%time
result_MeanShift = MeanShift(bandwidth=9).fit_predict(data)
result_MeanShift
```

```
Wall time: 0 ns
```

Out[16]:
```
array([ 9580, 11681, 16258, ...,  6818,  6818,  1499], dtype=int64)
```



```
In [17]: silhouette_score(data,result_MeanShift)

Out[17]: 0.3387179434413535

In [18]: plt.scatter(data.iloc[:, 0], data.iloc[:, 1], c=result_MeanShift, s=50, cmap='viridis')

Out[18]: <matplotlib.collections.PathCollection at 0x19958c87508>
```

```
In [19]: %time
         result_DBSCAN = DBSCAN(eps=3, min_samples=3).fit_predict(data)
         result_DBSCAN

         Wall time: 0 ns

Out[19]: array([   -1,     -1,     -1, ...,     -1,     -1, 10892], dtype=int64)

In [20]: silhouette_score(data,result_DBSCAN)

Out[20]: -0.2520307345110983

In [21]: plt.scatter(data.iloc[:, 0], data.iloc[:, 1], c=result_DBSCAN, s=50, cmap='viridis')

Out[21]: <matplotlib.collections.PathCollection at 0x19958c62fc8>
```