Лабораторная работ 8.
Кластеризация.

Цель: ознакомиться с методами кластеризации Sklearn (DBSCAN, OPTICS).
Задачи:

1. Загрузить пред обработанные данные;
2. Стандартизировать данные;
3. Провести уменьшение размерности данных;
4. Реализовать алгоритмы кластеризации;
5. Визуализировать результат.

Пример выполнения лабораторной работы.

Выведем первые пять строк из подготовленного датасета.

```
import sklearn as sk
import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import normalize
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
df = pd.read_csv('lab8.csv',encoding='cp1251')
print(df.head(5))
```

Результат:

	id	gender	age	income	spending_rating
0	1	1	19	15	39
1	2	1	21	15	81
2	3	0	20	16	6
3	4	0	23	16	77
4	5	0	31	17	40

Стандартизируем набор данных.

```
df = pd.read_csv('lab8.csv',encoding='cp1251')
scaler = StandardScaler()
x_scaled = scaler.fit_transform(df)
x_normal = normalize(x_scaled)
x_normal = pd.DataFrame(x_normal)
print(x_normal )
```

Результат:

```
            0         1         2
0   -0.824060 -0.490203  0.283948
1   -0.865764 -0.460096  0.196886
2   -0.791057 -0.603322 -0.101154
3   -0.756162 -0.597231 -0.267461
4   -0.710876 -0.694302 -0.112250
..        ...       ...       ...
193  0.800191  0.575505 -0.168787
194  0.769443  0.601467 -0.214930
195  0.701185  0.712951  0.006435
196  0.730336  0.675310  0.102788
197  0.717734  0.695519 -0.033334
```

Реализуем алгоритм анализа главных компонентов, чтобы уменьшить размерность данных для визуализации и сохраним новый набор данных.

```
df = pd.read_csv('lab8.csv',encoding='cp1251')
scaler = StandardScaler()
x_scaled = scaler.fit_transform(df)
x_normal = normalize(x_scaled)
x_normal = pd.DataFrame(x_normal)
pca = PCA(n_components=2)
x_principal = pca.fit_transform(x_normal)
x_principal = pd.DataFrame(x_principal)
x_principal.columns = ['V1', 'V2']
print(x_principal.head())
x_principal.to_csv('lab8new.csv', header=True)
```

Результат:

```
         V1        V2
0 -0.940242  0.245337
1 -0.940014  0.168566
2 -0.965465 -0.155430
3 -0.922679 -0.319759
4 -0.974309 -0.188895
```

Далее реализуем алгоритм DBSCAN и посмотрим на данные и кластеры после его реализации.

```
df = pd.read_csv('lab8new.csv',encoding='cp1251')
dbscan = DBSCAN(eps=0.036, min_samples=4).fit(df)
labels = dbscan.labels_
df['cluster'] = dbscan.labels_
print(df.tail())
```

Результат:

```
         V1        V2  cluster
0.986244 -0.115585       -1
0.987562 -0.153802       -1
1.001295  0.088696       -1
0.986184  0.174064       -1
1.003655  0.045309       -1
```

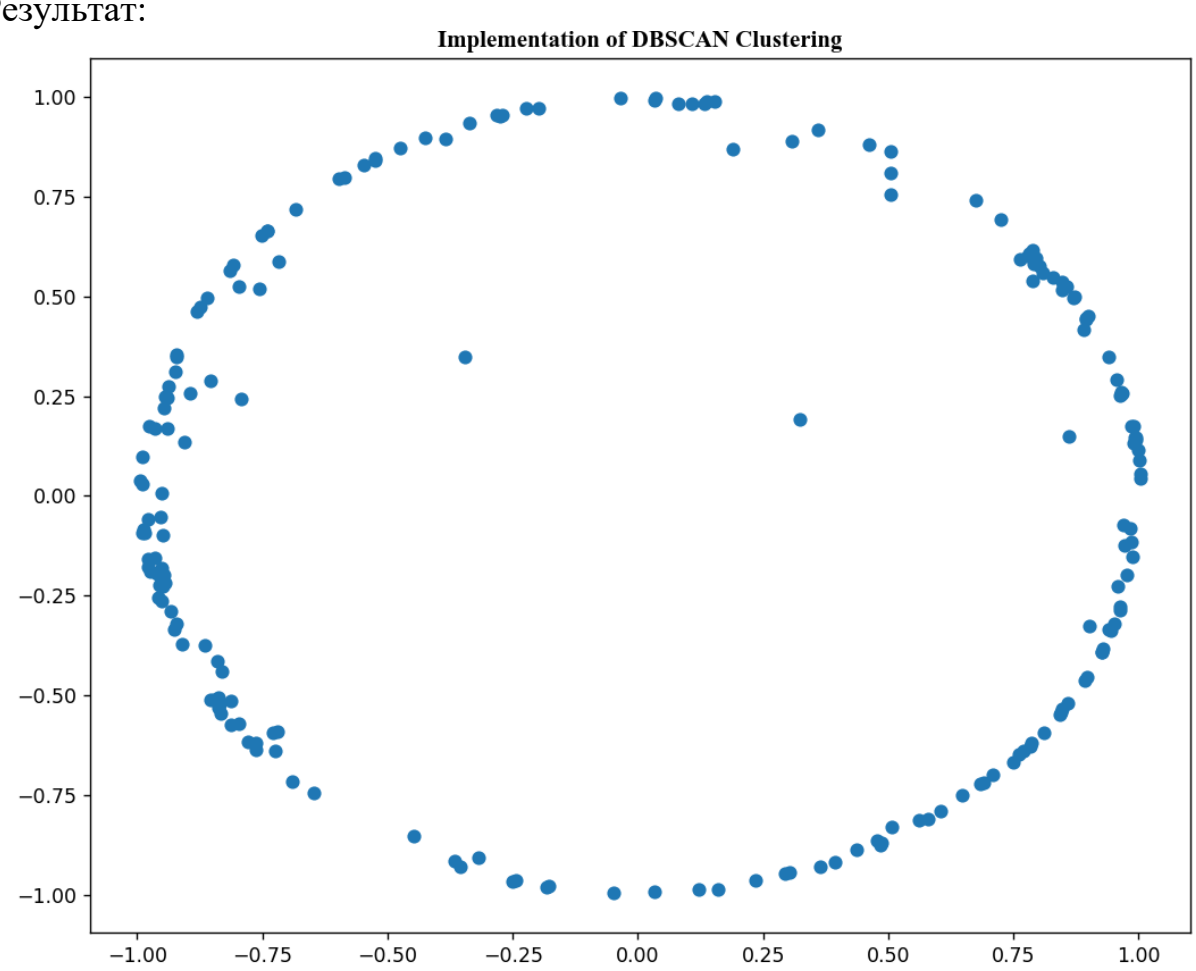
Выведем метки кластеров, количество кластеров, а также процент наблюдений, которые кластеризовать не удалось.

```
print(set(dbscan.labels_))
print(len(set(dbscan.labels_)) - 1)
print(list(dbscan.labels_).count(-1) / len(list(dbscan.labels_)))
```

Визуализируем полученные данные.

```
plt.figure(figsize=(10, 8))
plt.scatter(df['V1'], df['V2'])
plt.title("Implementation of DBSCAN Clustering", fontname="Times New Roman",fontweight="bold")
plt.show()))
```

Результат:



В ходе лабораторной работы мы познакомились с методами кластеризации Sklearn. Мы получили такие результаты… по DBSCAN и OPTICS. Данные методы отличаются….

Задание.

1. Загрузить пред обработанные данные, вывести первые 5 строк из датасета;
2. Стандартизировать данные;

3. Уменьшить размерность данных с помощью алгоритма главных компонент;
4. Реализовать методы DBSCAN и OPTICS;
5. Выделить метки кластеров, количество и процент неудач;
6. Визуализировать полученные данные по каждому методу;
7. Сделать выводы и описать различие методов.

Формат отчета.

Протокол лабораторной работы в формате PDF, который должен содержать, поэтапное выполнение всех задач с текстовым описанием ваших действий и экранными формами, отображающими данные действия.