МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по практическому заданию №2
по дисциплине «Машинное обучение»

Студент гр. 6304	___________________	Пискунов Я.А.
Преподаватель	___________________	Жангиров Т.Р.

Санкт-Петербург
2020

Все задания выполнены с использованием библиотек numpy и matplotlib для языка python. Код программы представлен в листинге 1.

Листинг 1. Код программы

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import KernelPCA


def kernel_func(xi, xj):
    return np.linalg.norm(xi - xj)**2


data1 = np.array([[4, 2.9], [2.5, 1], [3.5, 4], [2, 2.1]])
data2 = np.array([[8, -20], [0, -1], [10, -19], [10, -20], [2, 0]])

# Task 1
kernel_matrix = [[kernel_func(xi, xj) for xi in data1] for xj in data1]

# Task 2
mean = np.mean(data2, axis=0)
cov_matrix = np.cov(data2, rowvar=False)
eigvals, eigvecs = np.linalg.eig(cov_matrix)
inner_size = (len(data2), len(data2[0]))

data_centered = data2 - mean
project_matrix = -eigvecs[:, np.argmax(eigvals)]
first_component = np.dot(data_centered, project_matrix)

distribution = np.random.multivariate_normal(mean, cov_matrix, 1000)
plt.plot(distribution[:, 0], distribution[:, 1])

# Task 3
transformed_data = np.array(kernel_matrix)**2 + 7*np.ones((len(data1), len(data1)))
kernel_matrix_ = transformed_data@kernel_matrix@transformed_data
precomputed_data = KernelPCA(1, 'precomputed').fit_transform(kernel_matrix_)

print()
print(kernel_matrix, '\n', mean, '\n', cov_matrix, '\n', eigvals, '\n', inner_size, '\n', first_component,
      '\n', precomputed_data)
plt.show()
```

Задание 1.

Произведен расчет ядерной матрицы. Размер матрицы – 4 на 4, каждый элемент рассчитан по функции сходства. В результате получена следующая матрица:

0.00	5.86	1.46	4.64
5.85	0.00	10.00	1.46
1.46	10.00	0.00	5.86
4.64	1.46	5.86	0.00

Задание 2.

По заданным значениям данных вычислено среднее значение. Результат следующий – (6, -12).

Далее рассчитана матрица ковариации. Каждый ее элемент вычисляется следующим образом:

$$X_{ij} = M\left(\left(X_i - M(X_i)\right)\left(X_j - M\left(X_j\right)\right)\right).$$

В итоге получен следующий результат:

22.0	-47.5
-47.5	110.5

Собственные числа данной матрицы λ_1 и λ_2, полученные путем решения уравнения

$$\begin{vmatrix} 22 - \lambda & -47.5 \\ -47.5 & 110.5 - \lambda \end{vmatrix} = 0,$$

следующие – 1.33 и 131.17.

«Внутренний» размер набора данных – (5, 2).

Далее произведен расчет первого главного компонента. Для его вычисления данные центрированы относительно медианы, а также использована матрица обратных собственных векторов. В результате первый главный компонент следующий – (-8.13, 12.48, -8.01, -8.93, 12.60).

На основе значений среднего и матрицы ковариации сгенерировано нормальное распределение. Двумерная функция его плотности схематично представлена на рис. 1. Информации на данном рисунке достаточно, чтобы определить ее протяженность и ориентацию.

Задание 3.

Произведен расчет первого главного компонента для данных и ядра из первого задания. В качестве нелинейного преобразования ядра выбрано преобразование

$$f(x) = x^2 + 7.$$

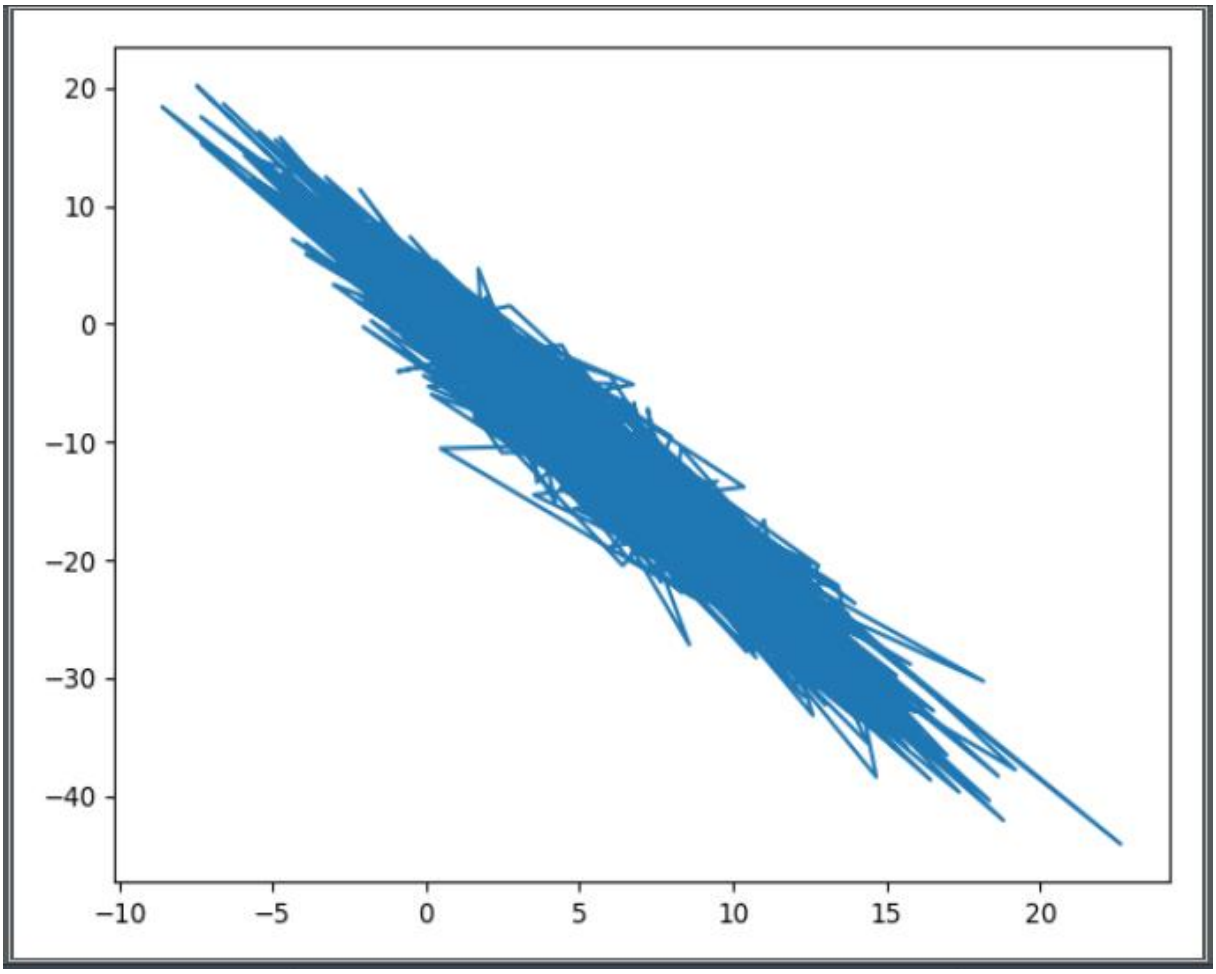

Рисунок 1 – Двумерная функция нормальной плотности

Сами вычисления первого главного компонента в этот раз произведены автоматически с использованием средств библиотеки sklearn. Результат расчета – (83.14, -83.14, -83.14, 83.14).