**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА САПР**

**ОТЧЕТ**

**по Лабораторной работе № 1**

**«ГЕНЕРАЦИЯ МОДЕЛЬНЫХ НАБОРОВ**

**ДАННЫХ»**

**по дисциплине «Автоматизация схематического проектирования»**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 0302 |  | Сухарев Л. А.. |
| Преподаватель |  | Боброва Ю. О. |

Санкт-Петербург

2023

**Цель:** получение навыков работы с numpy-массивами и написание функций на языке Python на примере генерации массивов произвольно распределенных данных.

Для изучения различных классификаторов, их свойств и особенностей, создадим модельные данные, форму распределения и смешанность которых можно регулировать вручную.

Наборы формируются так, чтобы первый набор удовлетворял условиям применения текущего метода классификации/кластеризации, второй – нет.

Используемые модули: numpy, matplotlib

1. **ОСНОВЫНЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ**

В работе необходимо сгенерировать два массива с элементами различных классов по 1000 объектов в каждом в 4-х мерном пространстве признаков, а также массив с метками класса для них. Сначала мы создадим переменные, распределение по нормальному закону с незначительно различными средними и дисперсиями. Эта форма распределения характерна для случайных величин, подверженных большому числу случайных факторов и очень часто встречается в биологии и медицине. После этого мы создадим переменную, содержащую метки класса – логические нули и единицы. После этого нужно будет разделить данные на обучающую и тестовую подвыборки и визуализируем результаты, чтобы оценить пересекаемость классов. По распределению данных в виде гистограмм и диаграммы рассеяния можно будет сделать выводы о возможности их последующего анализа и необходимости коррекции свойств распределения.

1. **КОД ПРОГРАММЫ. ПОЯСНЕНИЕ ЭТАПОВ**

Для начала импортируем необходимые библиотеки и зададим средние значения mu и стандартное отклонения sigma для каждого из классов (Рис.1)

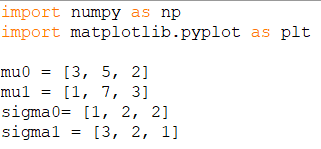


Рис. 1 — Начало работы

После этого создадим переменные, соответствующие классам. Удобнее сделать это в цикле for, создавая столбец и конкатенируя (стыкуя) его с предыдущими. (Рис. 2)

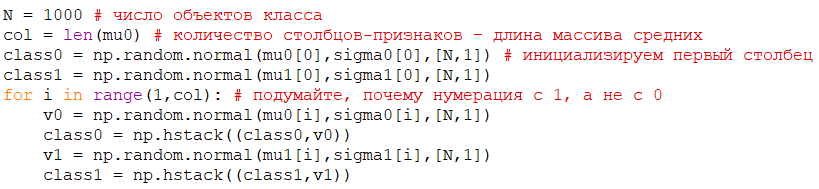


Рис. 2 — Инициализация значений

Создадим переменную, содержащую метки класса – логические 0 и 1. Сведем в единую переменную Х объекты двух классов, а также метки двух классов в переменную Y. (Рис. 3)

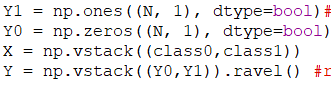


Рис. 3 — Создание меток классов

Разделим данные на две части – это будут обучающая и тестовые подвыборки. Пропорции 70/30. Перед этим перемешаем («пошафлем») данные так, чтобы объекты из разных классов перемешались. Обратим внимание, что меняя местами элементы массива с данными X, необходимо точно также перемешать метки Y, чтобы не утратить соответствие между объектами и метками. (Рис. 4)

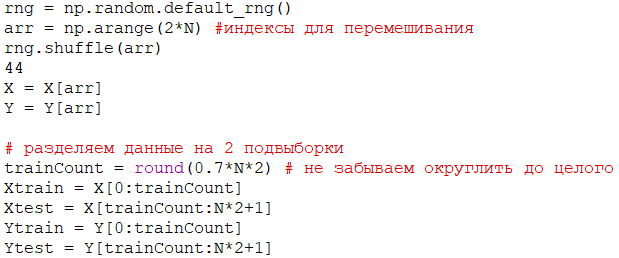


Рис. 4 — Разделение на тестовую и обучающую выборки

Визуализируйте результаты, чтобы оценить пересекаемость классов. Это можно сделать, используя гистограммы и диаграммы рассеяния (скатерограммы), средствами модуля matplotlib. Не забудем подписать название и оси графиков. (Рис. 5)

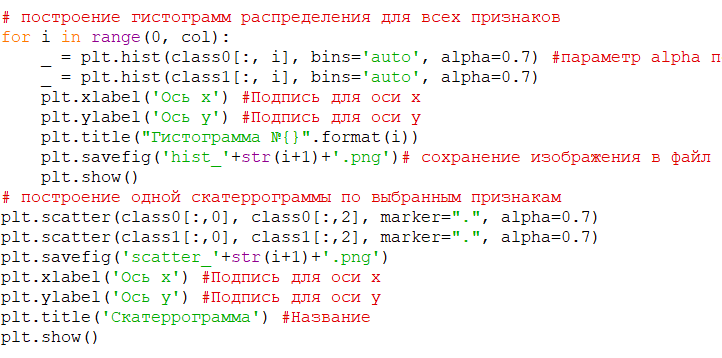


Рис. 5 — Код для визуализации результатов

Вынесем генерацию данных из основного скрипта. Это позволит использовать ее в других проектах и обеспечит повторное использование кода без копирования. Перепишем основной скрипт так, чтобы вместо генерации данных в напрямую в этом скрипте, скрипт вызывал созданную функцию генерации данных. (Рис. 6)

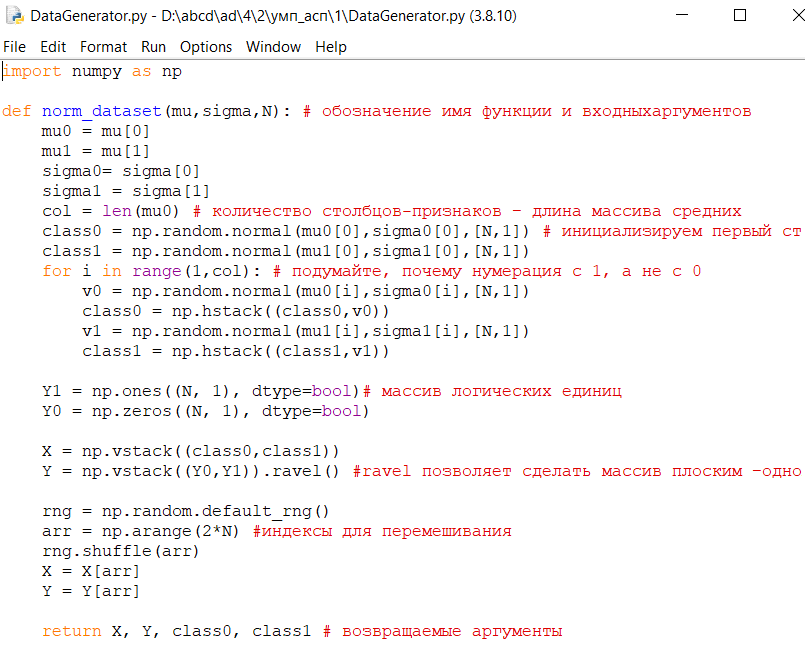
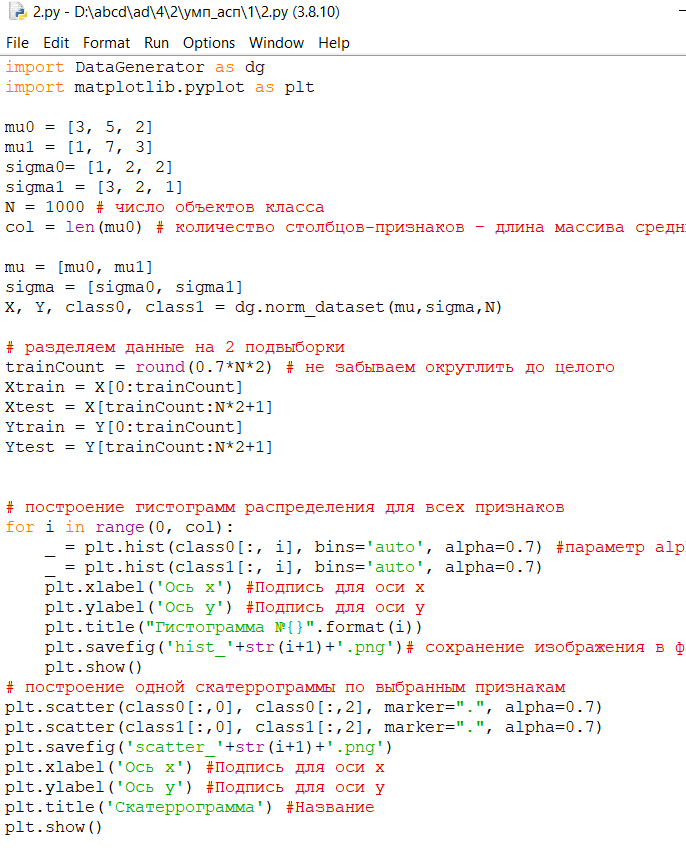


Рис. 6 — Разбиение кода на два файла. Выноска генерации данных в отдельный файл

Создадим новую функцию внутри файла генератора данных, генерирующую массив данных, распределенный в пространстве заданным образом. Форма приведена на рисунке 7.



Рис. 7 — Форма распределения данных

Основной скрипт вызывает созданную функцию генерации данных, после чего строит скатеррограмму. Для генерации данных использовалась встроенная функция sklearn.datasets, поскольку эта библиотека включала указанную модель распределения. Однако для более сложных или произвольных распределений нужно будет найти функции, приводящие к построению схожих графиков, и построить распределение, используя эти функции, где Х будет содержать координаты точек, а Y – принадлежность к классам. (Рис. 8)

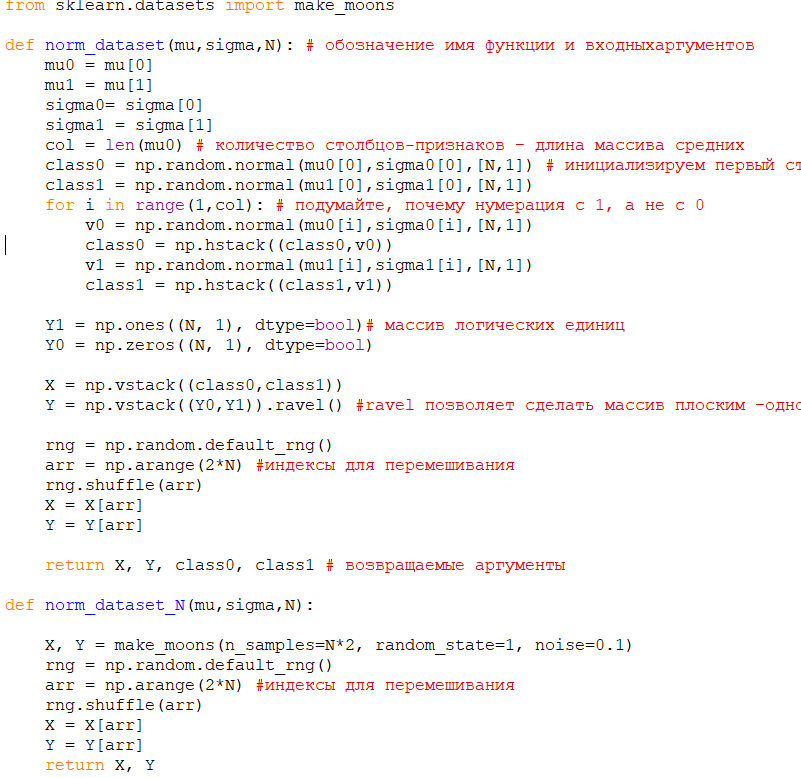
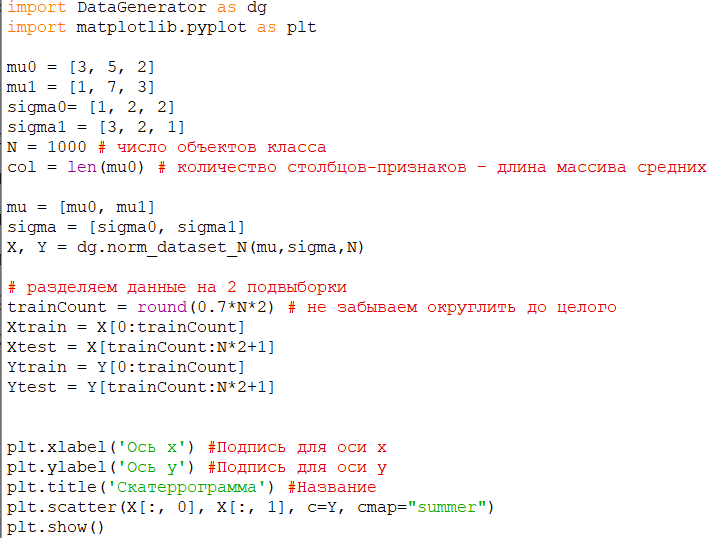


Рис. 8 — Код для указанного распределения данных

1. **ПОЛУЧЕННЫЕ ГРАФИКИ**

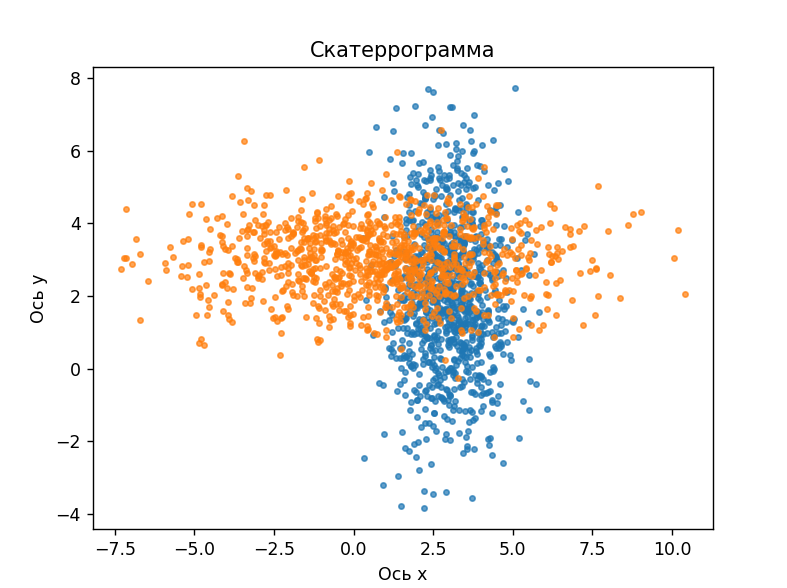
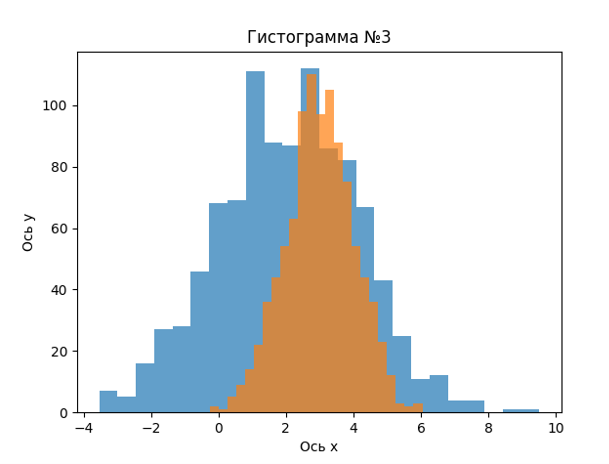
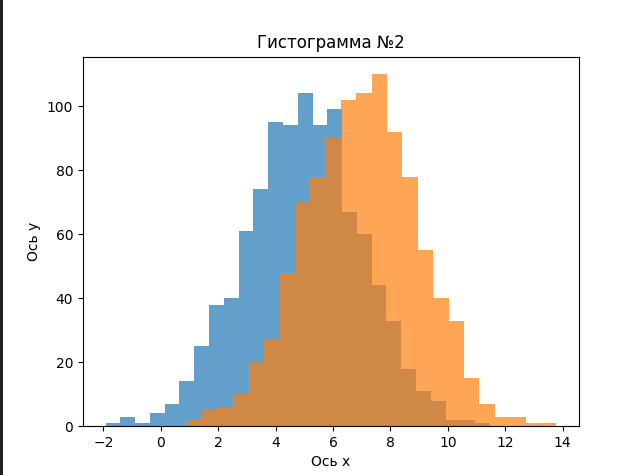
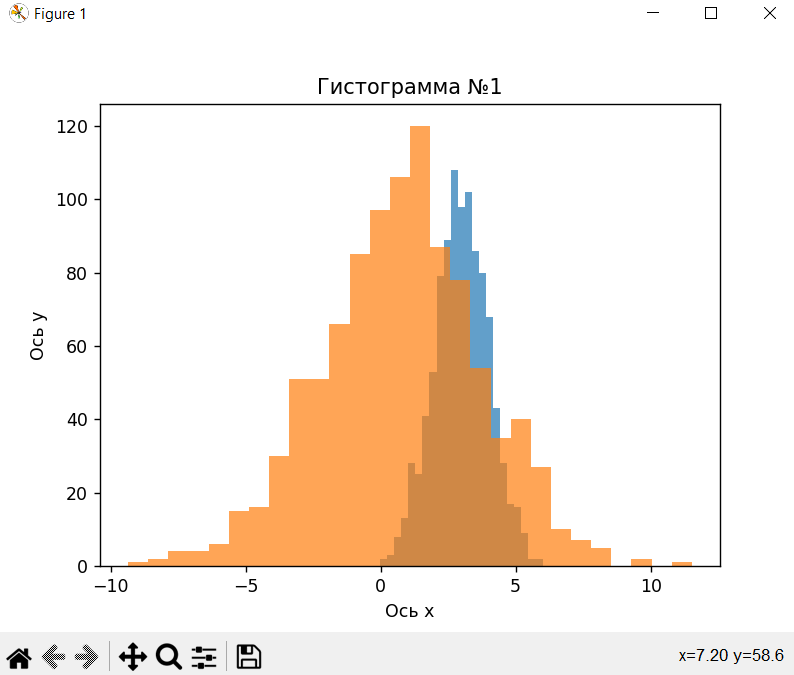


Рис. 9 — Графики, полученные в первой части задания

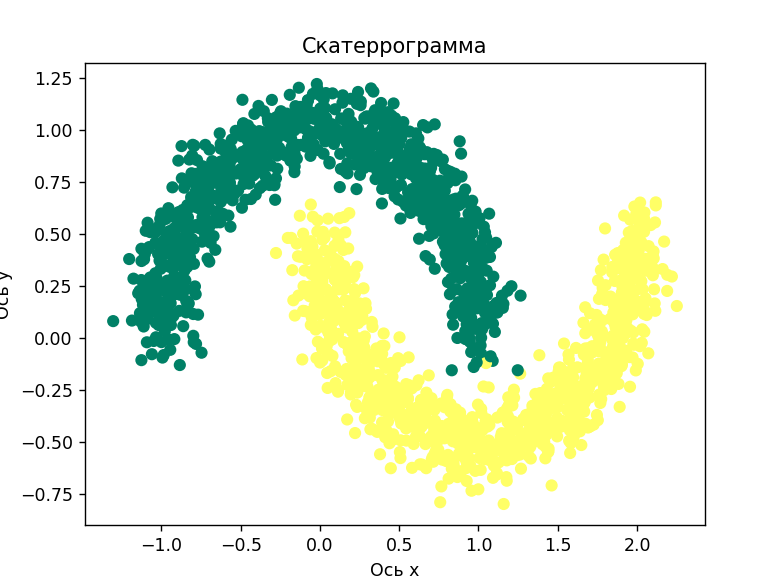


Рис. 10 — Скатеррограмма по указанному по варианту способу распределения данных

**Выводы**

В ходе данной работе были получены навыки работы с numpy-массивами и написание функций на языке Python на примере генерации массивов произвольно распределенных данных. Были созданы функции, генерирующие массивы с нормальным и казанным по варианту распределением данных, которые могут позже использоваться в других работах. Были изучены способы визуализации и было выяснено, как по скатеррограмме определить возможности последующего анализа данных и необходимости коррекции свойств распределения: чем ближе полученная скатеррограмма к графику функции распределения, тем ближе значения, полученные при создании распределенных данных, к желаемым. Гистограммы же позволяют наглядно продемонстрировать распределение данных по значениям, с указанием (цветом) к какому классу принадлежат данныею.