**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА САПР**

**ОТЧЕТ**

**по Практическим занятиям**

**по дисциплине «Автоматизация схематического проектирования»**

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

Санкт-Петербург

2023

**1. БАЗОВЫЕ ОПЕРАЦИИ И СИНТАКСИС ЯЗЫКА PYTHON**

Установим Python 3 на компьютер, после напишем простой скрипт, открывающий csv файл, полученный по адресу:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv>.

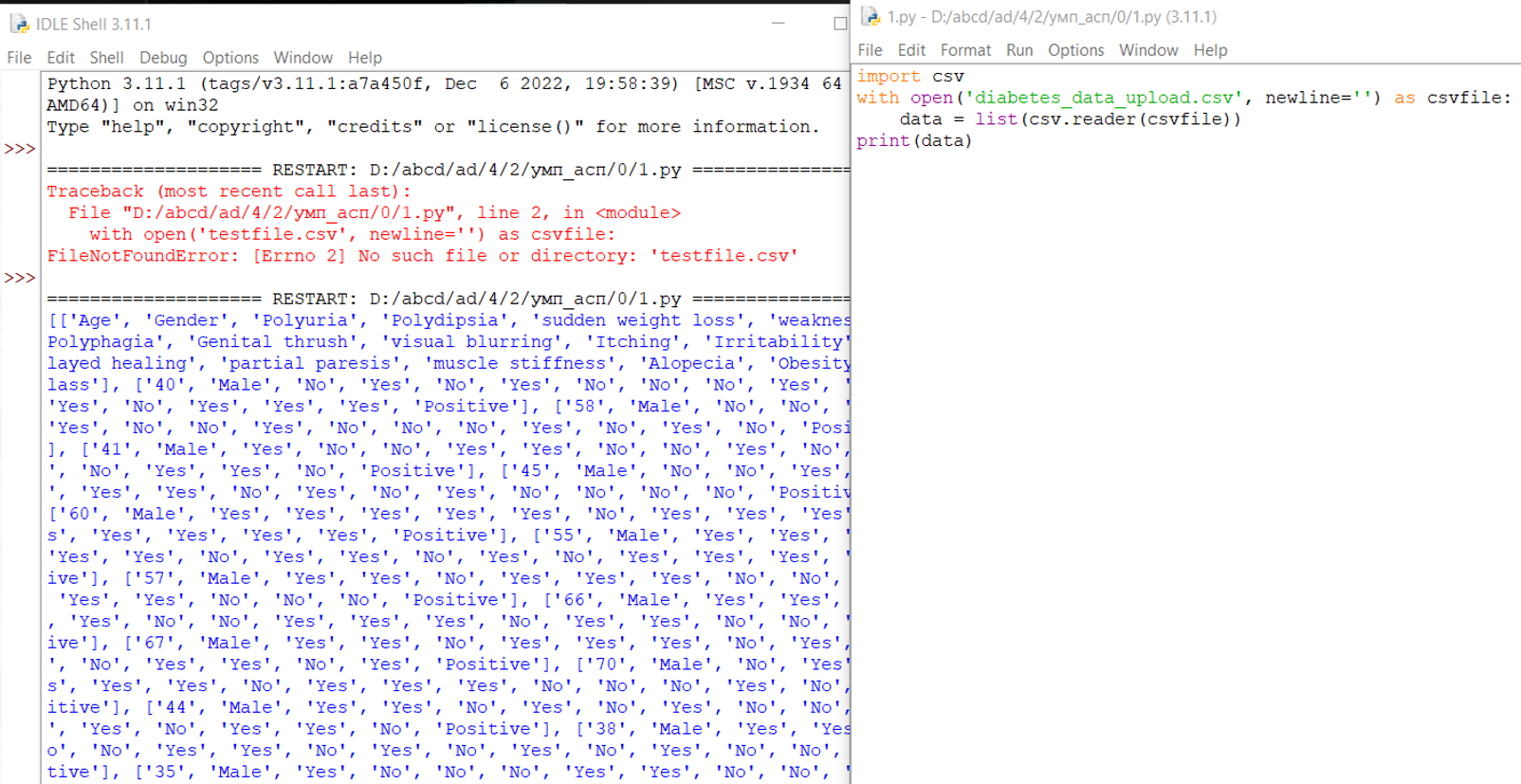


Рис. 1 — Открытие файла csv

После этого рассчитаем среднее значение по колонке Age (рис.2)



Рис. 2 — Расчет среднего значения возраста всех пациентов

Рассчитаем среднее значение возраста отдельно для мужчин и женщин (рис.3)

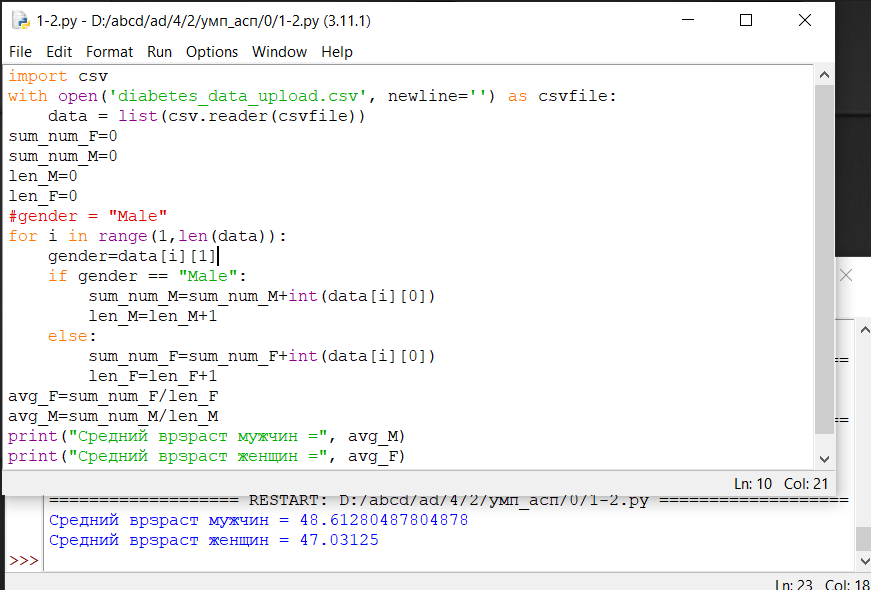


Рис. 3 — Расчет среднего значения возраста для мужчин и женщин

Построим таблицу, показывающую связь между возникновением диабета и ожирением. Для этого создадим четыре массива со значением порядковых номеров пациентов: тех, у кого есть и диабет и ожирение, те, у кого есть диабет и нет ожирения, те, у кого нет ни диабета, ни ожирение, и те, у кого нет диабета, но есть ожирение (рис.4)

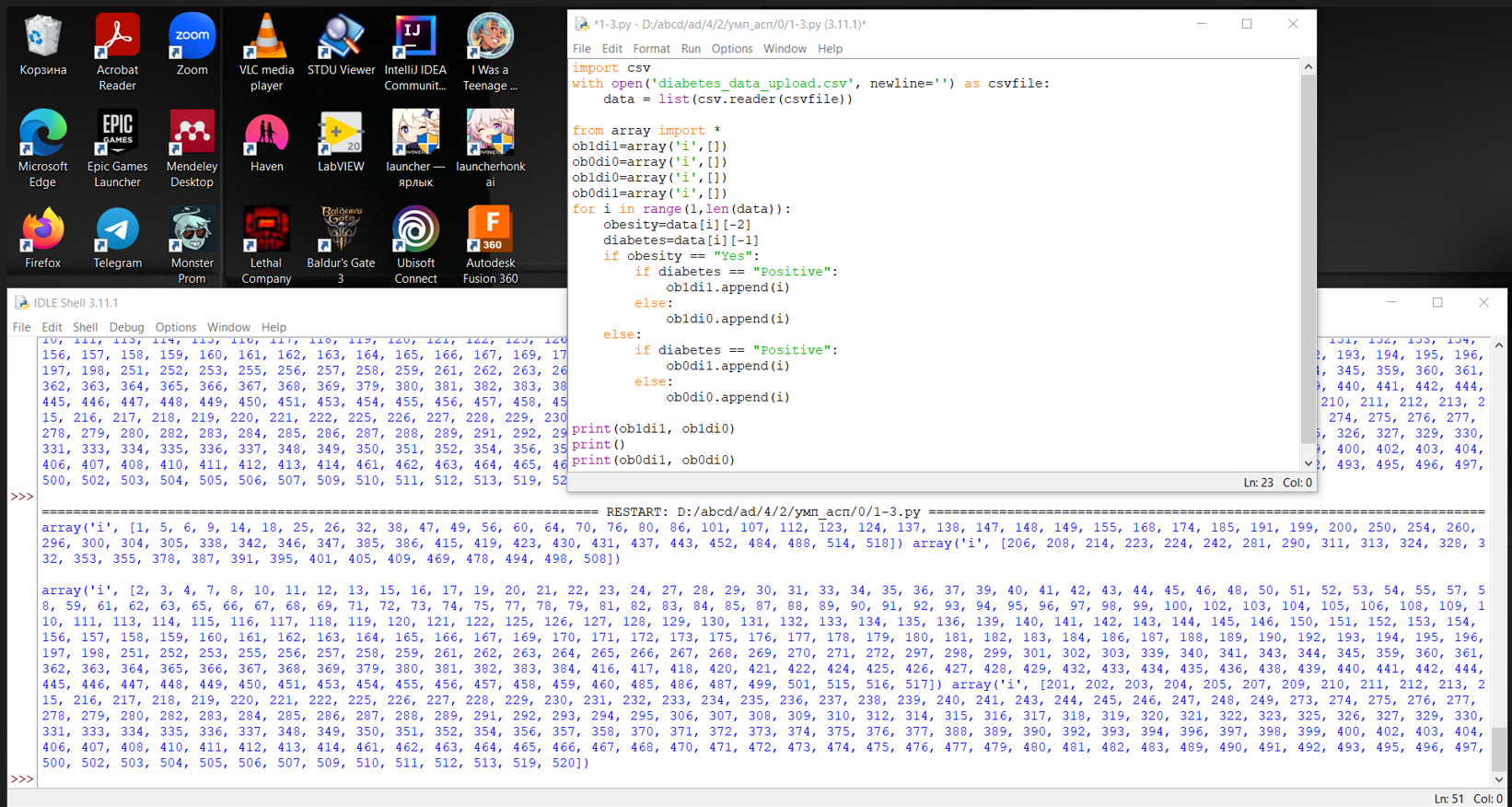


Рис. 4 — Изучение связи между диабетом и ожирением

**2. ЗАГРУЗКА И НАСТРОЙКА МОДУЛЕЙ NUMPY, PANDAS,**

**SCIKIT-LEARN, TENSORFLOW, KERAS**

Для начала установим указанные библиотеки через командную строку (рис.5)

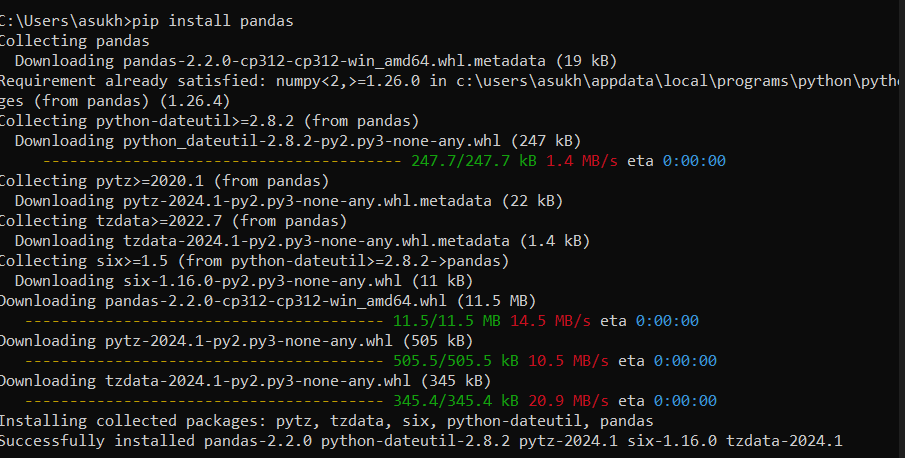
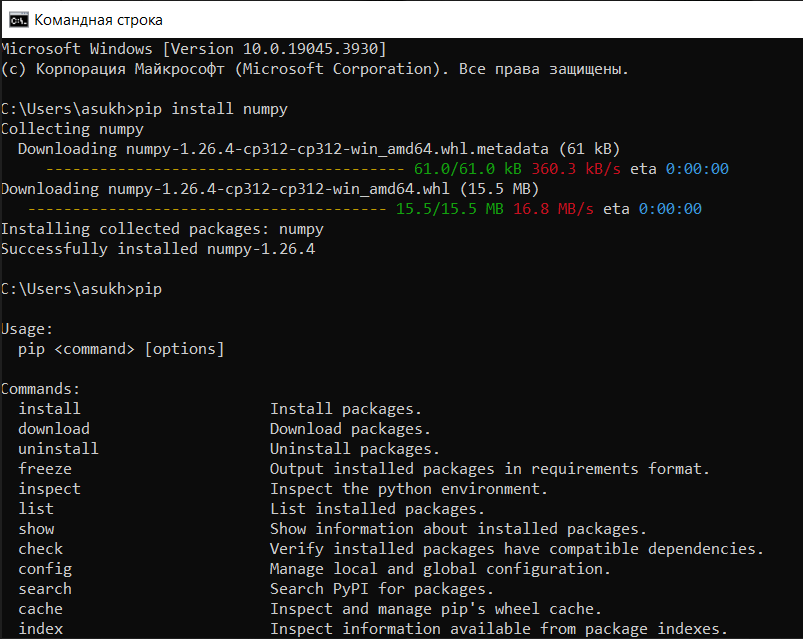
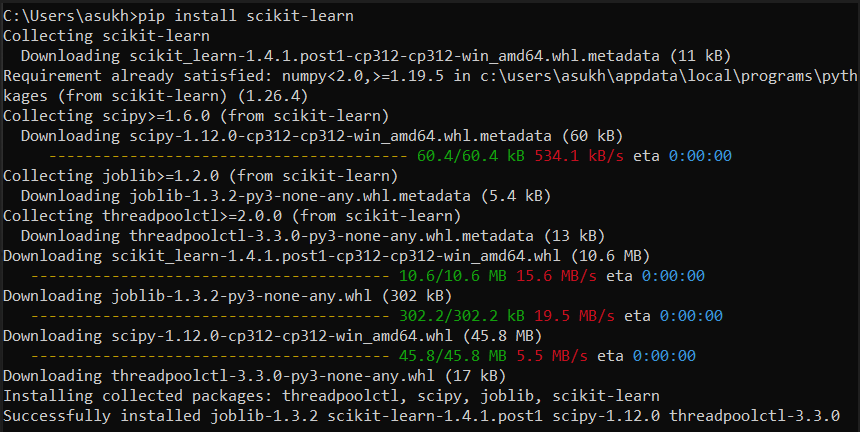
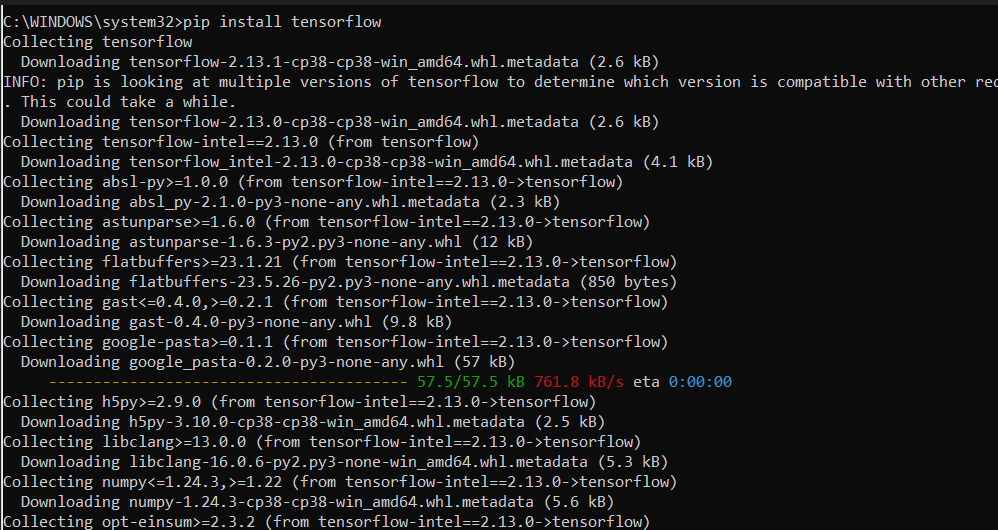
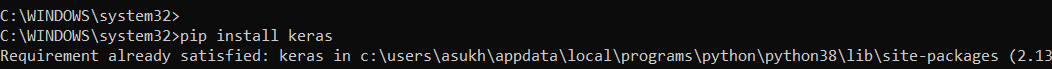
   

Рис. 5 — Установка библиотек через командную строку

Узнаем версию каждой из установленных библиотек через командную строку и через скрипт Python (рис.6)

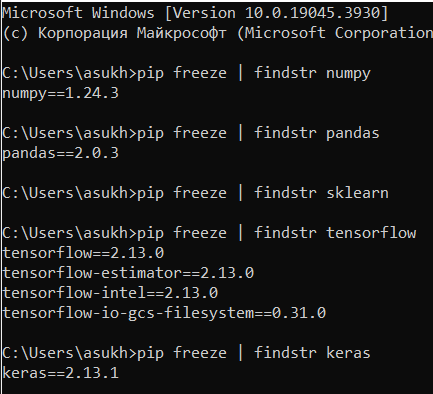
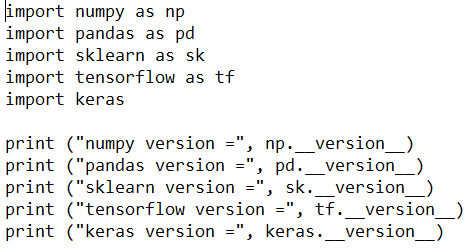
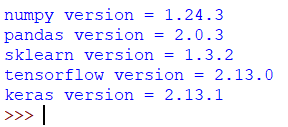
  

Рис. 6 — Версии установленных библиотек

После этого откроем ранее изученный csv файл с помощью методов библиотеки pandas (рис.7)

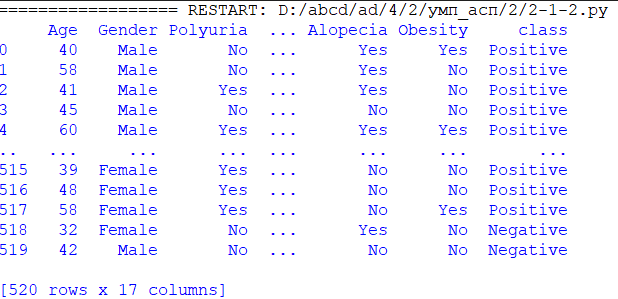


Рис. 7 — Открытие csv файла с помощью pandas

Рассчитаем среднее по колонке Age, используя средства numpy/pandas для всех людей и отдельно для мужчин и женщин. Выведем полученные значения в консоль (рис.8)

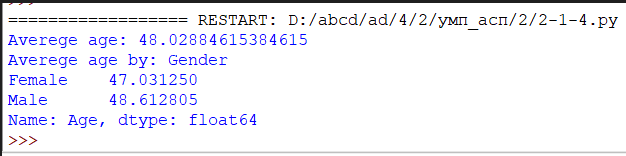


Рис. 8 — Средний возраст, используя средства numpy/pandas

Используя методы модуля time рассчитаем и выведем в консоль время выполнения операции из предыдущего пункта, а также аналогичного расчета из упражнения 1 (рис.9). Нетрудно заметить, что средства numpy/pandas позволяют выполнить эту операцию намного быстрее.

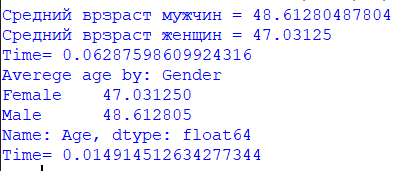
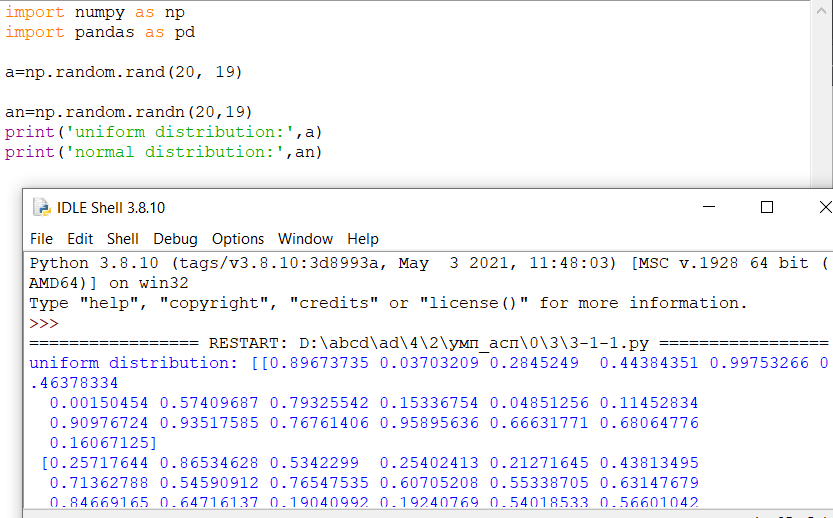


Рис. 9 — Сравнение времени выполнения нахождения среднего возраста.

**3. NUMPY ARRAYS AND FUNCTIONS**

Создадим двумерный массив numpy, длина которого равна порядковому номеру в списке группы, а ширина – номеру первой буквы фамилии для равномерно и нормально распределенных случайных чисел с помощью случайных генераторов в NumPy (рис.10)



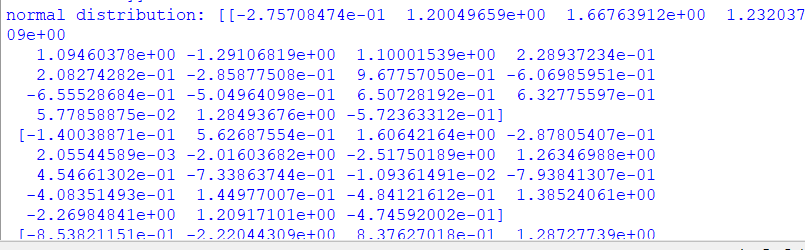


Рис. 10 — Создание массива и заполнение их случайными значениями с заданным типом распределения

Создадим в Python функцию, которая принимает NumPy массив и выводит в консоль результаты запуска указанных команд. Запустим эту функцию из другого места (был выбран другой скрипт) (рис.11)

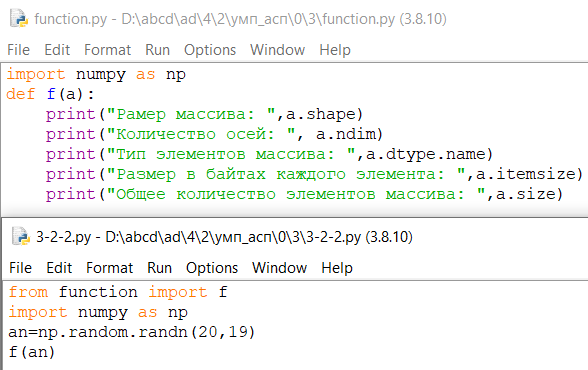
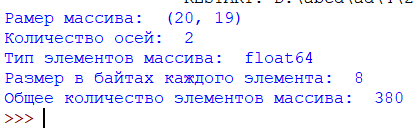
 

Рис. 11 — Создание и запуск функции

Откройте файл csv, с которым работали раньше, с помощью библиотеки pandas, определим тип каждого из столбцов данной таблицы средствами numpy и запишем полученные типы колонок в отдельный csv файл

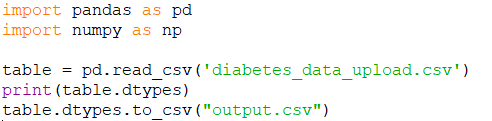
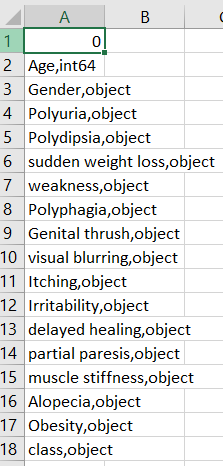
 

Рис. 12 — Изучение типов столбцов таблицы

Переконвертируем средствами numpy входную таблицу таким образом, чтобы таблица не содержала буквенных обозначений классов. Сохраните полученную таблицу в формате csv и повторим для нее нахождение среднего возраста разными способами еще раз. (Рис.13)

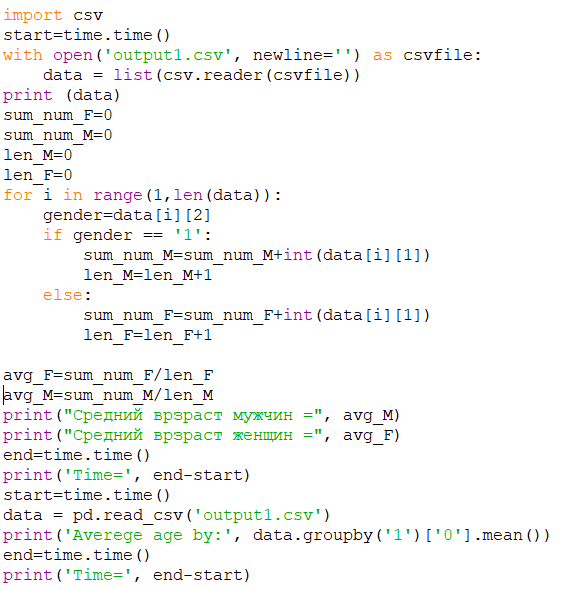
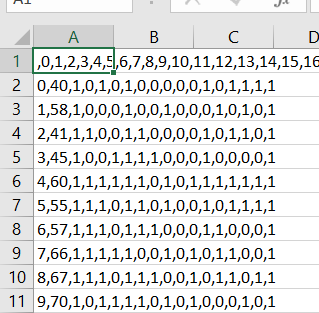
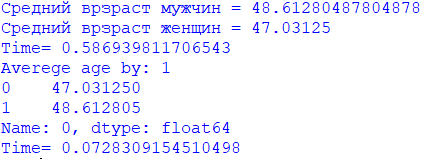
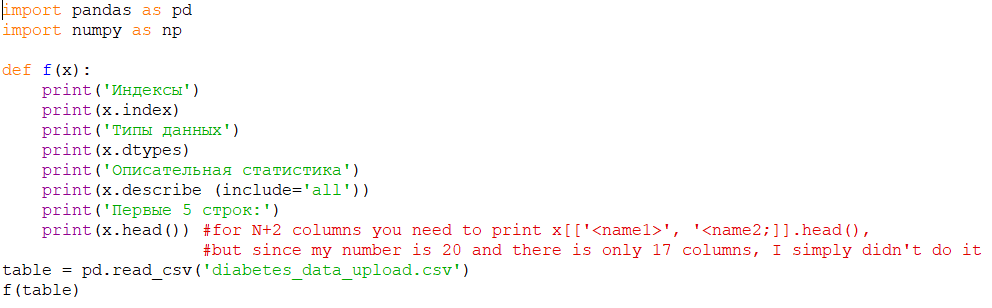
   

Рис. 13 — Избавление от буквенных значений, нахождение среднего значение возраста по новой таблице

**4. РАБОТА С ТАБЛИЧНЫМИ ДАННЫМИ В БИБЛИОТЕКЕ PANDAS**

Напишем функцию, которая принимает на вход DataFrame pandas, и выводит в консоль информацию об индексах, типах данных, описательную статистику и первые 5 строк для первых N+2 столбцов матрицы (здесь и далее N – собственный номер в списке группы). Используем данную функцию для описания дата фрейма, загруженного ранее (из csv файла) (рис.14)



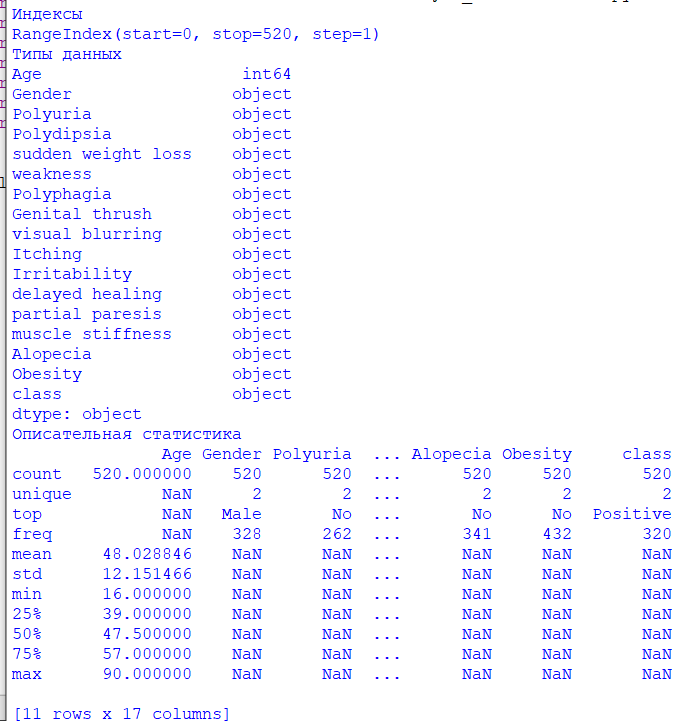
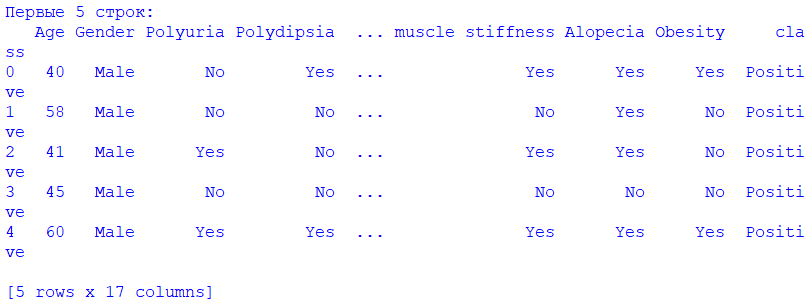
 

Рис. 14 — Исследование DataFrame pandas

Создадим два дата фрейма, полученных путем определения в один из них только строк со значением Yes в колонке N+1, а в другой – только со значением No (рис.15)

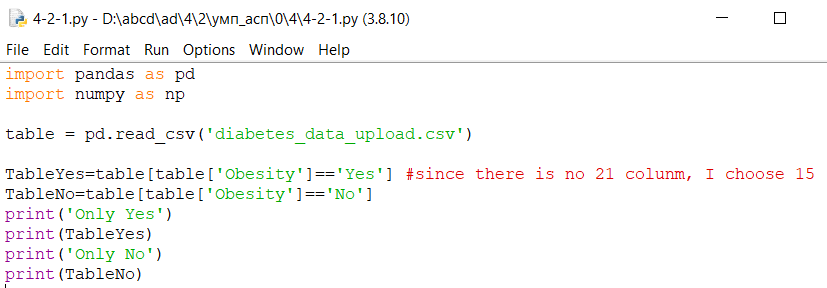
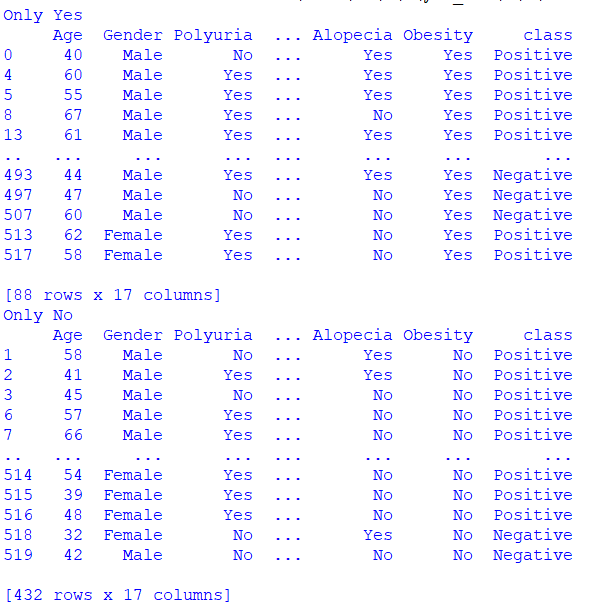
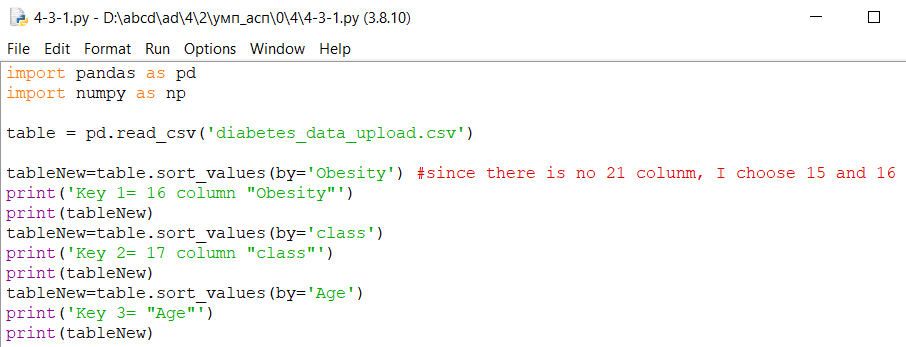
 

Рис. 15 — Создание двух фреймов с помощью группировки значений в изначальном фрейме

Выполним сортировку исходной таблицы по нескольким ключам: первый ключ – колонка N+1, второй класс – колонка – N+2, третий ключ – возраст (Age). Сохраним результат в виде отдельного DataFrame (рис.16)



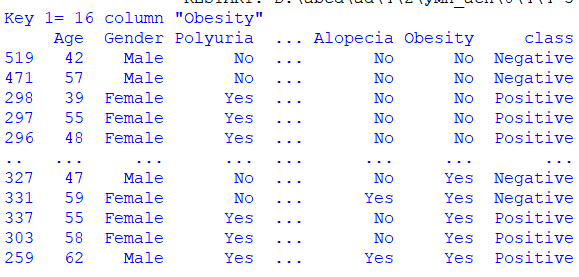
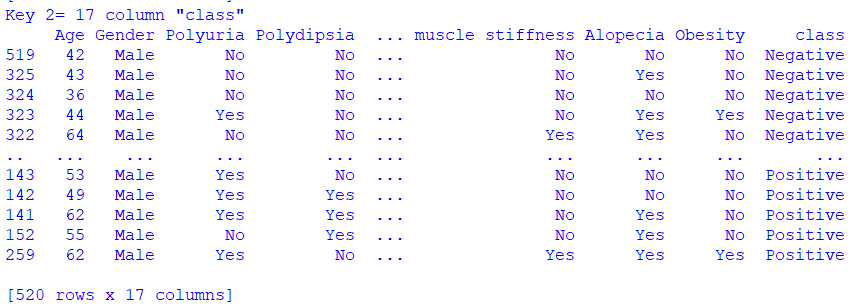
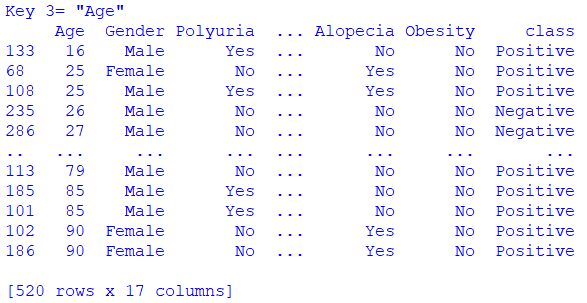
  

Рис. 16 — Сортировка исходной таблицы

Проверим, есть ли в данных пропущенные значения и удалим строки, в которых есть хотя бы один пропуск. (рис.17) Поскольку количество строк в таблице не изменилось, это говорит о том, что пропущенных значений не было.

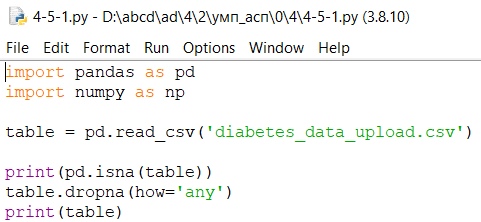
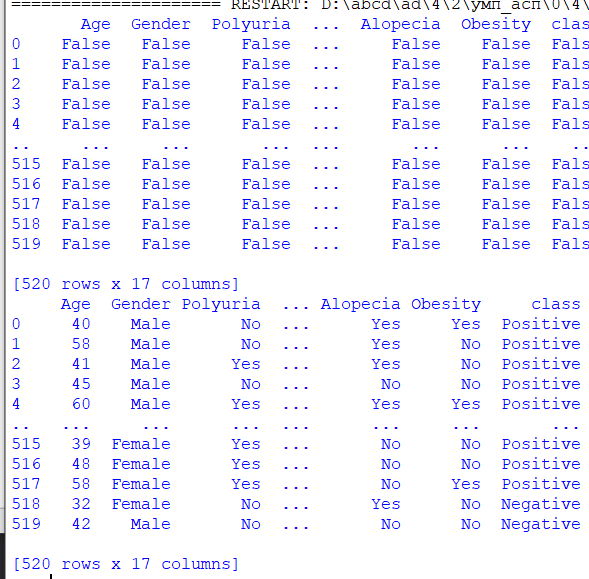
 

Рис. 17 — Проверка пропущенных строк и удаление их при нахождении

Построим гистограммы распределения Age для двух таблиц, получившихся при разделении данных в предыдущих пунктах (по возможности на соседних полях одного окна, используя функцию subplot) (рис.18)

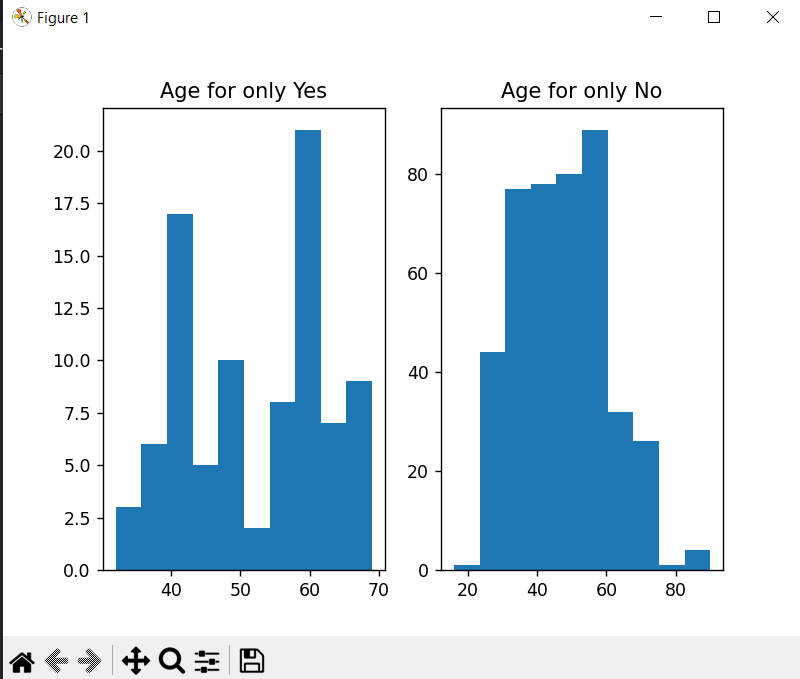
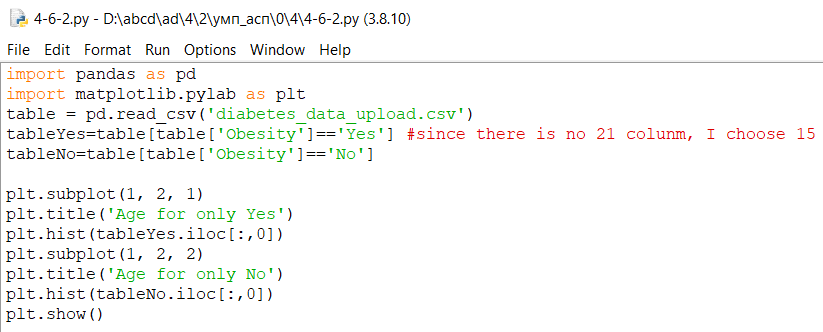


Рис. 18 — Построение гистограмм

Построим boxplot распределения Age для двух таблиц, получившихся при разделении данных в предыдущем пункте (рис.19)

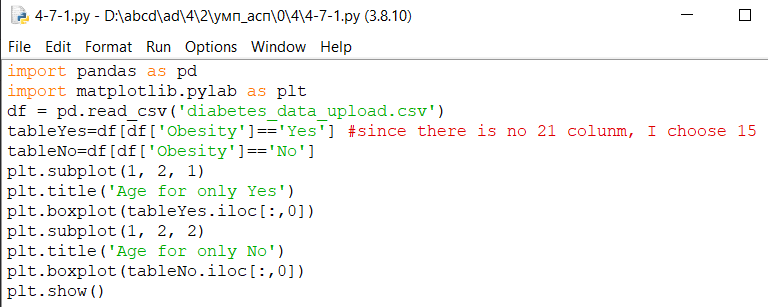
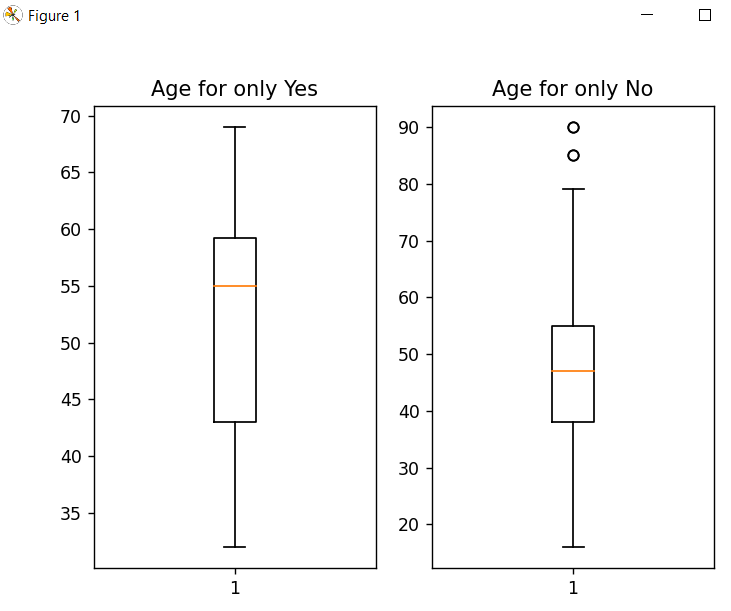
 

Рис. 19 — Построение boxplot

Построим scatter matrix для колонок Age, колонки N+1, и колонки N+2, закодировов цветом переменную class (Positive – красным, Negative – синим цветом) (рис.20)

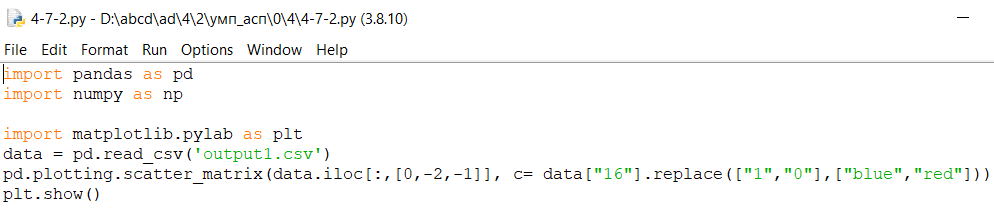
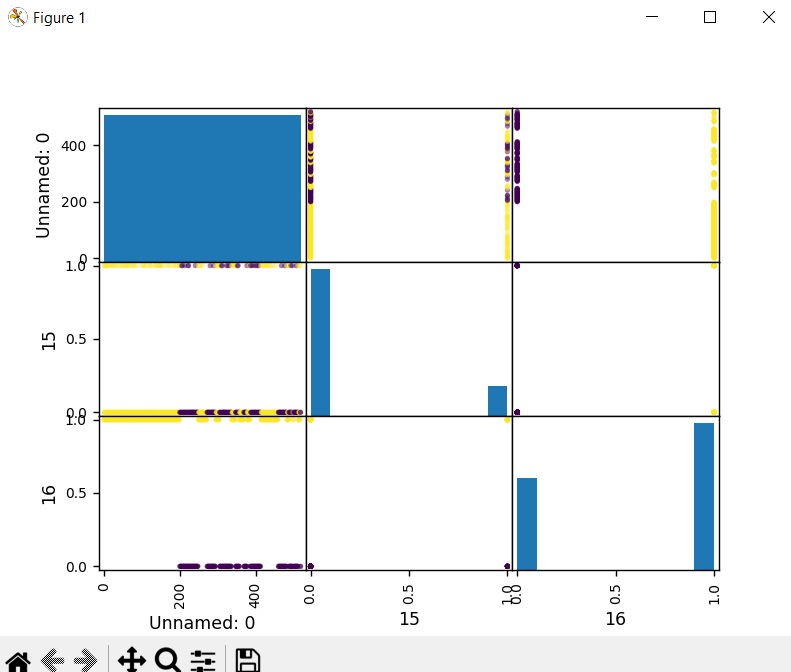
 

Рис. 20 — Построение scatter matrix

**5. МАШИННОЕ ОБУЧЕНИЕ С БИБЛИОТЕКОЙ SCIKIT-LEAR**

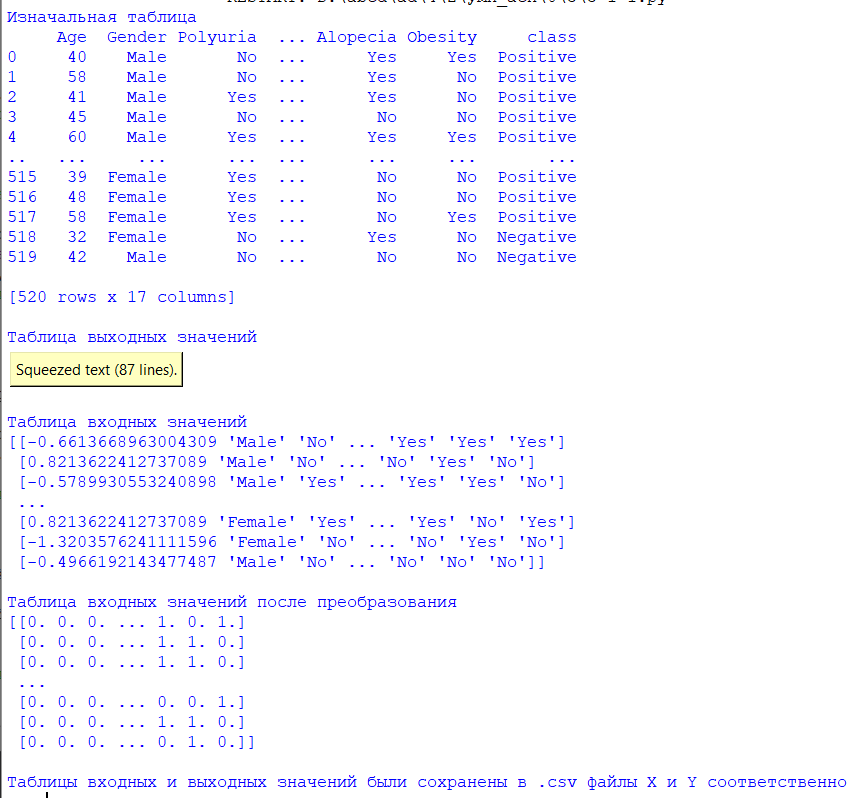
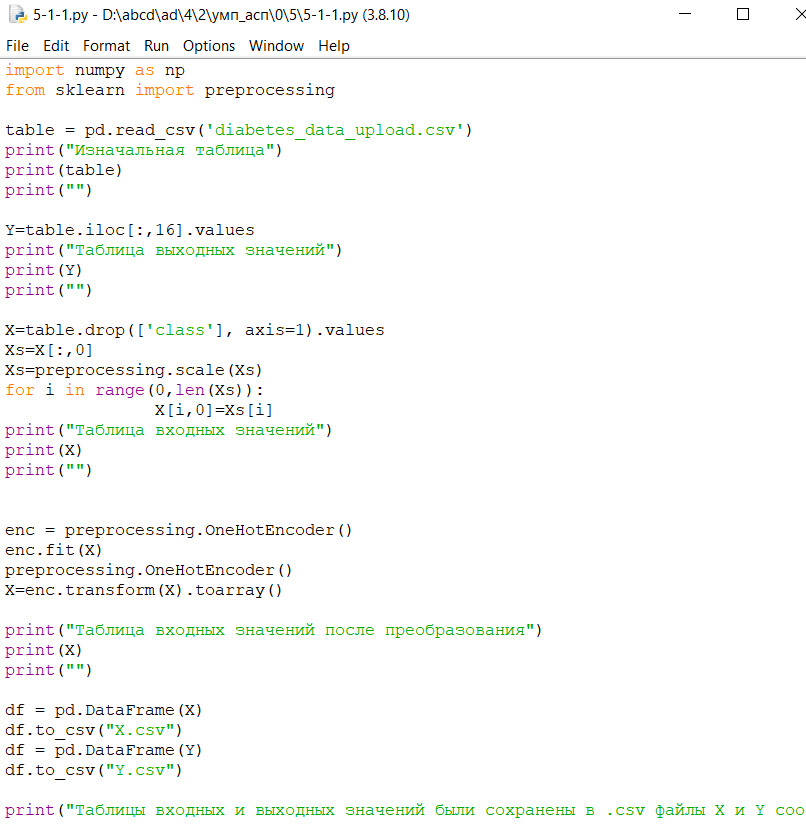
Для многих алгоритмов машинного обучения необходимо сделать предобработку данных. Для дальнейшей работы нужно разделить переменные в массивах на входные (X) и выходные (Y). В данном случае выходная переменная – наличие у человека сахарного диабета, а входные – все остальные. Используем изученные вами методы NumPy или Pandas для разделения таблицы с переменными на X и Y. (рис.21) 

Рис. 21 —Подготовка данных

Разделим ваши данные на обучающие (X\_train, Y\_train) и тестовые (X\_test, Y\_test) в пропорции 80 %/20 %, указав random\_state равным номеру в списке группы. Проверим размерности всех полученных массивов при помощи метода NumPy shape. Обучим модель с помощью логистической регрессии и натренируем модель при помощи команды fit() на обучающей выборке. Используя метод predict, предскажем наличие диабета у пациентов из тестовой выборки. Оценим, насколько хорошо ваша модель справляется с задачей при помощи accuracy\_score и еще одной метрики, подходящей для бинарной классификации. Попробуем улучшить качество классификации путем задания в модели (аргументом функции LogisticRegression()) параметра регуляризации модели C. (рис.22)

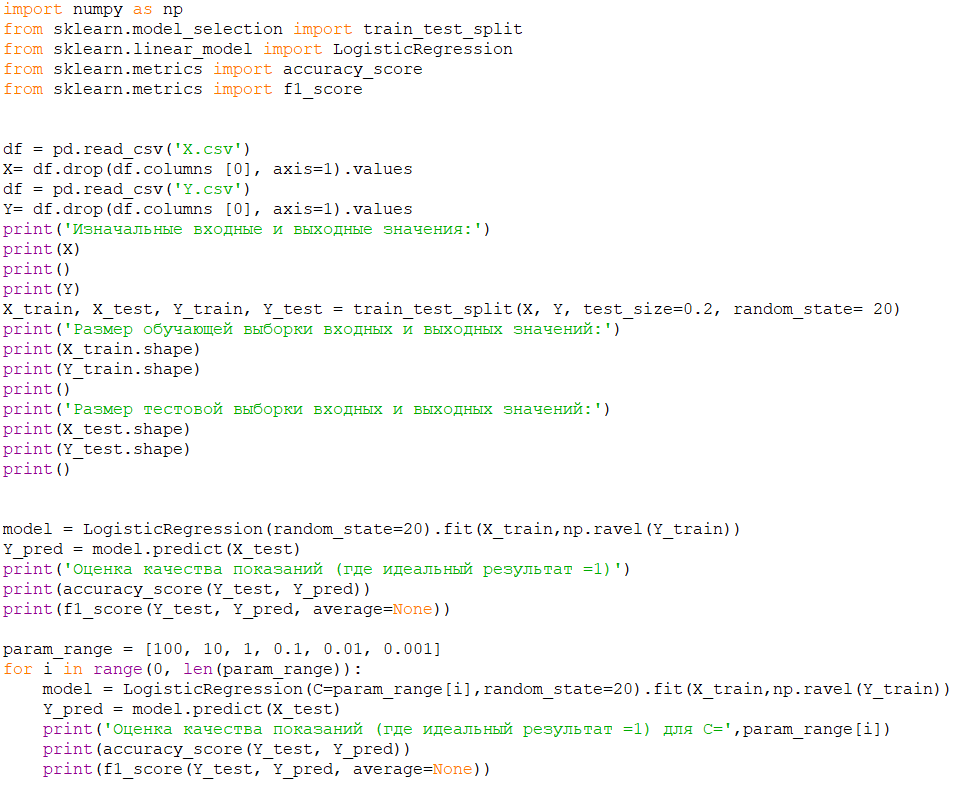
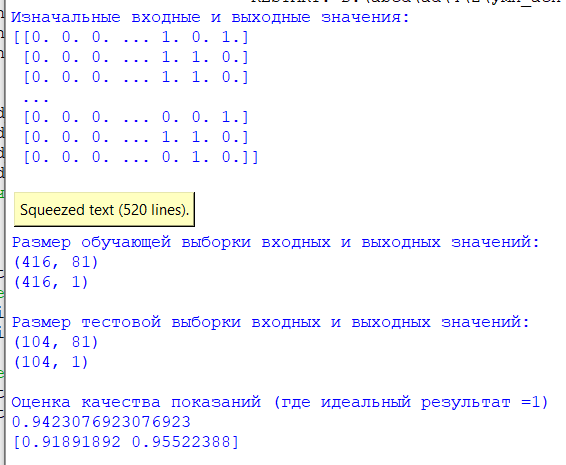
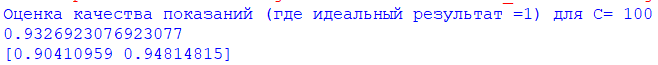
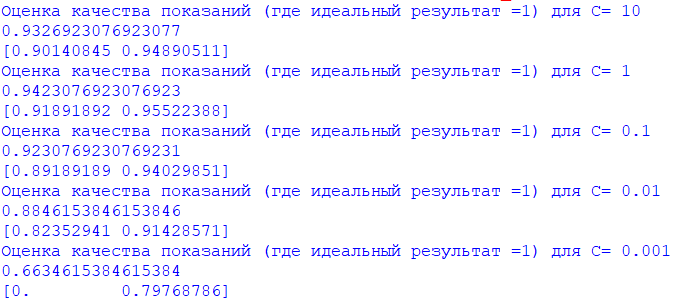
   

Рис. 22 — Обучение модели и результаты этого обучения.

**6. СОЗДАНИЕ КЛАССИФИКАТОРА С ПОМОЩЬЮ БИБЛИОТЕКИ PYTORCH**

В данном упражнении мы научимся строить простейшие нейронные сети для решения задачи регрессии. Напишем и обучим модель, пользуясь Google Colab — это бесплатный облачный сервис на основе Jupyter Notebook (рис. 23-24). Постараемся добиться лучшей точности, управляя параметрами модели.

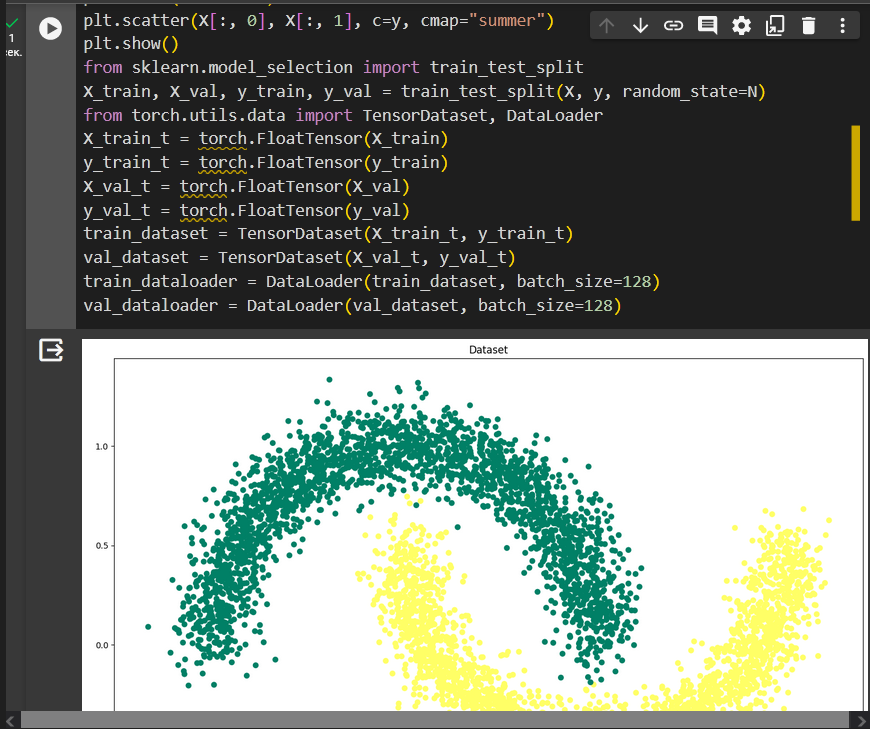


Рис. 23 — Отображение тренировочных данных

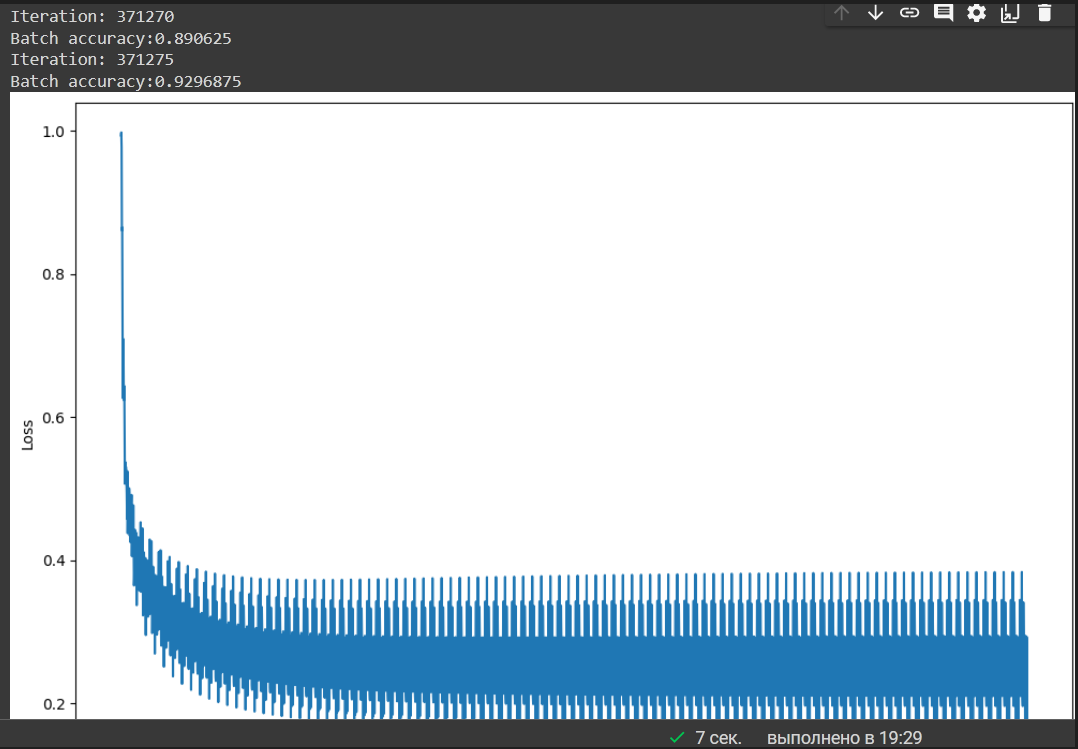


Рис. 24 — График изменения значений функции потерь

**7. СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ PYTORCH ДЛЯ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ**

В данном упражнении мы научимся строить простейшие сверточные нейронные сети для решения задачи классификации изображений.. Напишем и обучим модель, пользуясь Google Colab — это бесплатный облачный сервис на основе Jupyter Notebook (рис. 23-24). Постараемся добиться лучшей точности, управляя параметрами модели.

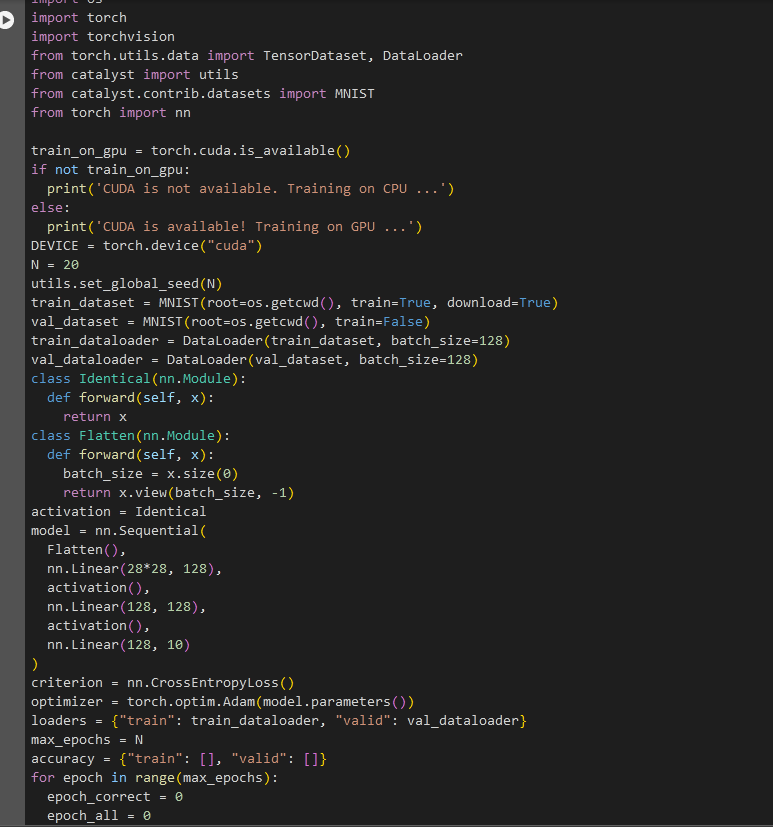
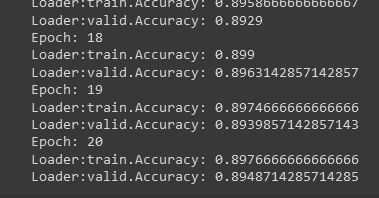
 

Рис. 25 – Результаты оценки точности работы модели

**Выводы**

В ходе данной работе было проведено знакомство с основными возможностями Python, работы с табличными данными и анализом данных, а также было проведено введение в нейронные сети на основе Python.