**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Дисциплина «Введение в информационные технологии»

Лабораторная работа №2

«Освоение многомерных массивов и математических операций с NumPy»

Выполнила:

Студентка группы БВТ2303

Морозова Ольга

**Цель работы:**

Изучить основные возможности библиотеки NumPy для работы с многомерными массивами, выполнения математических операций, индексирования и срезов данных, а также применения функций для обработки и анализа данных.

**Ход работы:**

**Задание 1.**

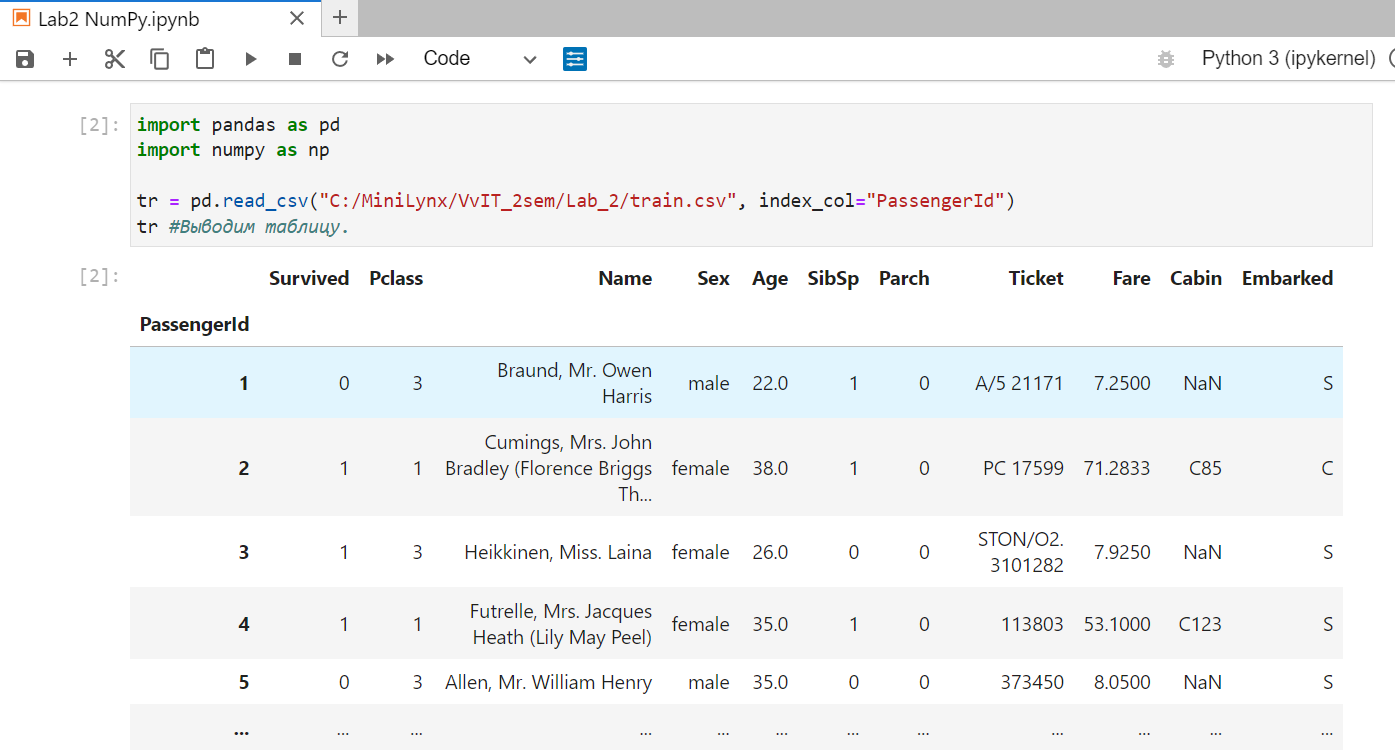
1. Загрузите датасет пассажиров титаника с помощью библиотеки pandas.

*import pandas as pd*

*import numpy as np*

*tr = pd.read\_csv("C:/MiniLynx/VvIT\_2sem/Lab\_2/train.csv", index\_col="PassengerId")*

*tr #Выводим таблицу.*



1. Создать (одномерный) массив numpy, содержащий данные о выживших пассажирах (колонка "Survived").

*sur = tr['Survived'].to\_numpy()*

или

*sur = np.array(tr['Survived'])*

1. После считаем общее количество выживших пассажиров.

Есть 3 варианта сделать это через NumPy:

*np.count\_nonzero(sur == 1)*

Или

*index, count = np.unique(sur, return\_counts=True)*

*dict(zip(index, count))*

Или

*np.bincount(sur)*



А также есть методы без использования NumPy:

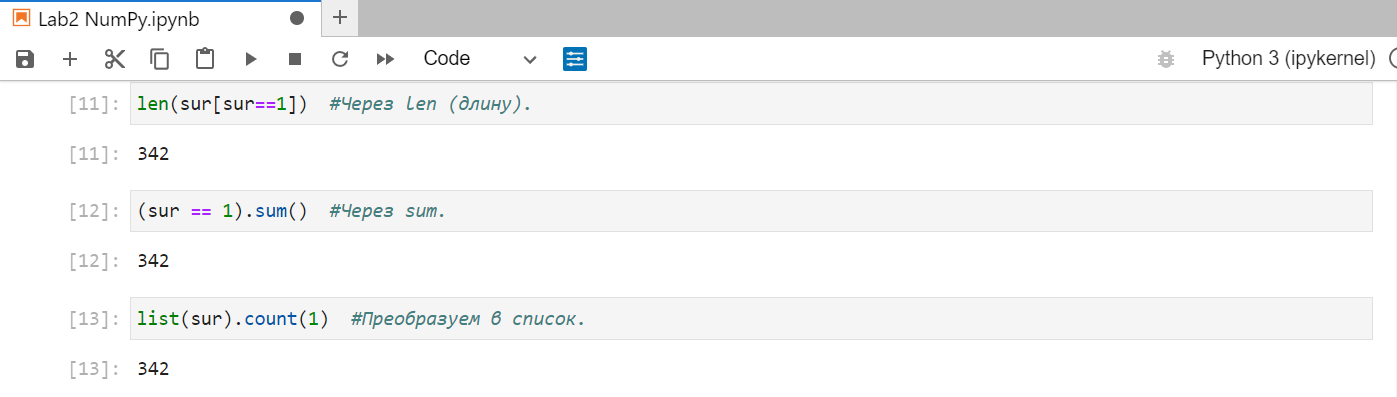
*len(sur[sur==1]) #Через len (длину).*

Или

*(sur == 1).sum() #Через sum.*

Или

*(sur == 1).sum() #Через sum.*



Как видим, все они выводят одно значение.

**Задание 2.**

1. Создайте двумерный массив numpy, содержащий данные о возрасте (колонка "Age") и стоимости билета (колонка "Fare") пассажиров.

Есть 2 способа:

*ar = np.array(tr[['Age', 'Fare']])*

*ar*

Или

*tr[['Age', 'Fare']].values*



1. Отсортируйте массив по возрасту пассажиров.

В переменные записываем первые значения отсортированного массива.

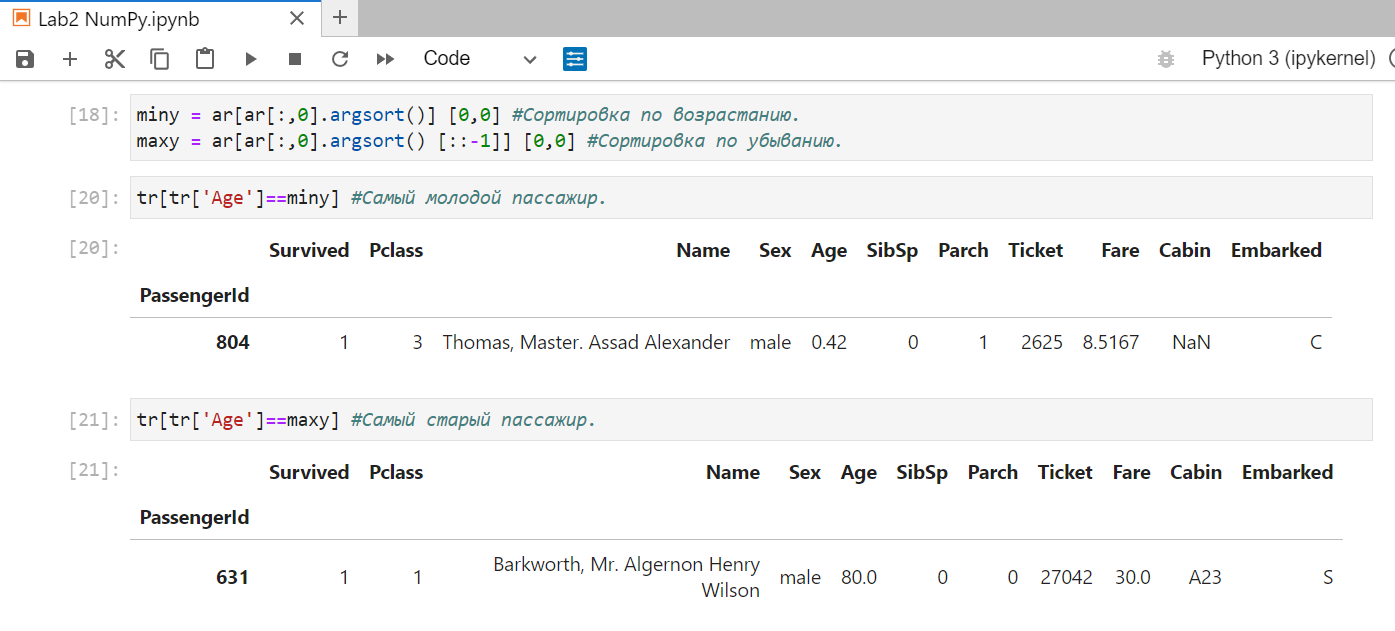
*miny = ar[ar[:,0].argsort()] [0,0] #Сортировка по возрастанию.*

*maxy = ar[ar[:,0].argsort() [::-1]] [0,0] #Сортировка по убыванию.*

1. Выведите информацию о самом молодом и самом старом пассажире.

*tr[tr['Age']==miny] #Самый молодой пассажир.*

*tr[tr['Age']==maxy] #Самый старый пассажир.*



**Задание 3.**

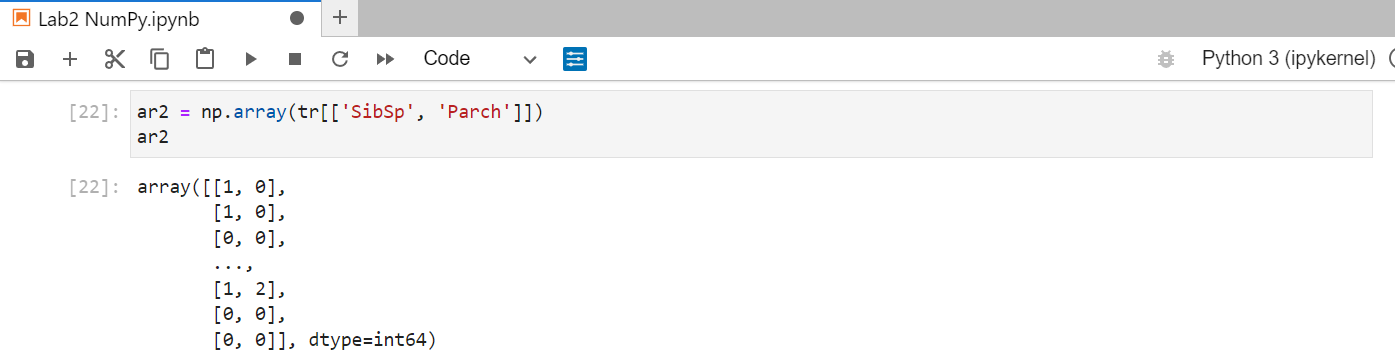
1. Создайте массив numpy, содержащий данные о количестве родственников (колонки "SibSp" и "Parch") у пассажиров.

SibSp - это число братьев, сестер или супругов на борту у человека.

Parch - количество родителей или детей, с которыми путешествовал каждый пассажир.

*ar2 = np.array(tr[['SibSp', 'Parch']])*

*ar2*



1. Посчитайте общее количество родственников на борту.

Это можно сделать через изначальную таблицу:

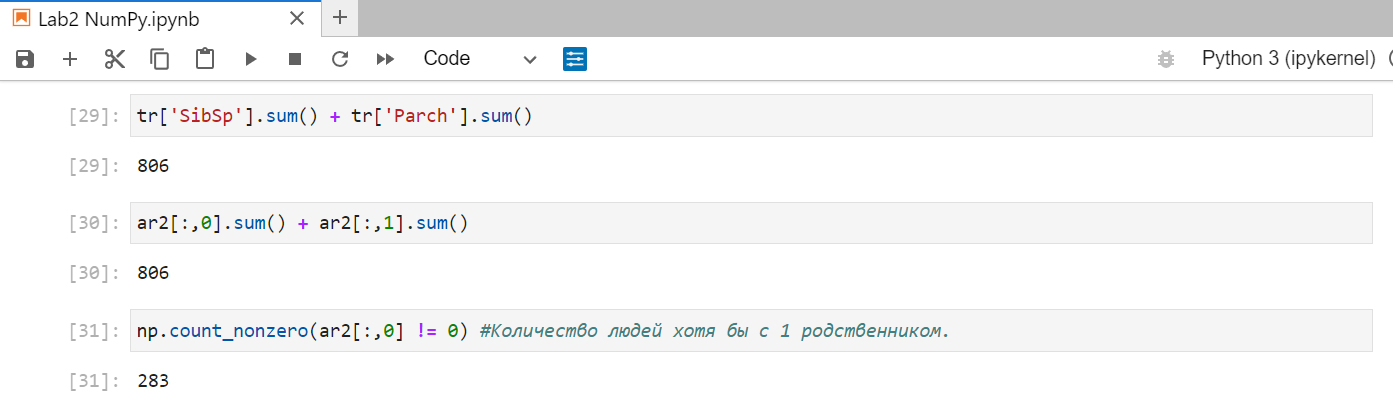
*tr['SibSp'].sum() + tr['Parch'].sum()*

Или через созданный массив NumPy:

*ar2[:,0].sum() + ar2[:,1].sum()*

Как видим, оба способы выводят одно значение.

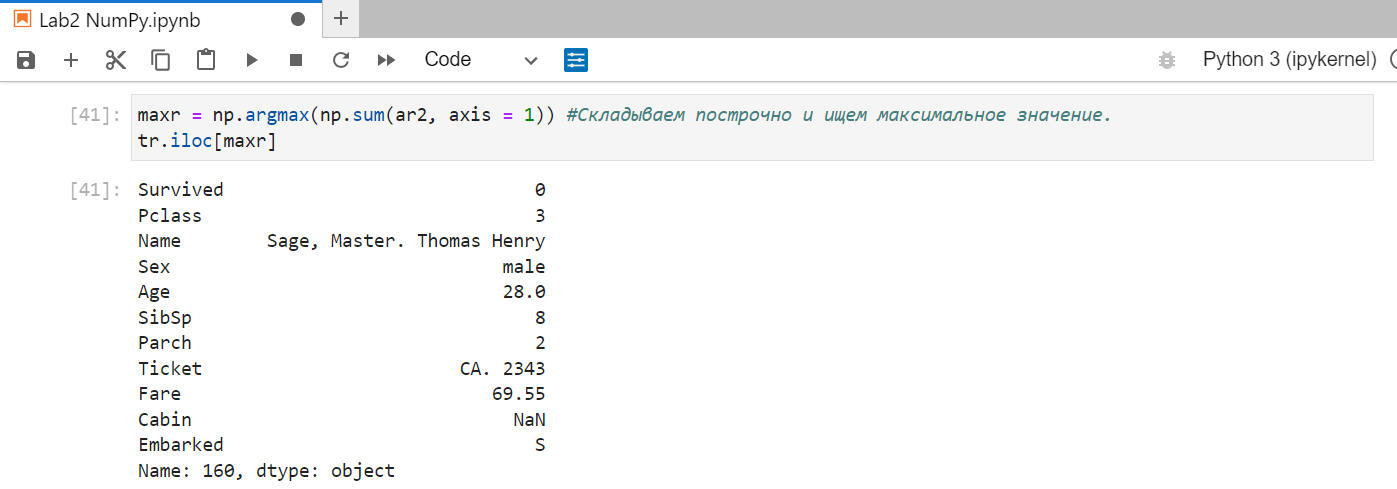
*np.count\_nonzero(ar2[:,0] != 0) #Количество людей хотя бы с 1 родственником.*



1. Определите пассажира с наибольшим количеством родственников.

*maxr = np.argmax(np.sum(ar2, axis = 1)) #Складываем построчно и ищем максимальное значение.*

*tr.iloc[maxr]*



**Вывод:**

Мы изучили основные возможности библиотеки NumPy для работы с многомерными массивами, выполнения математических операций, индексирования и срезов данных, а также применения функций для обработки и анализа данных.