

Язык Python и библиотеки для анализа данных: задание 1

Кантор Виктор

Инструкция для тех, кто пропустил часть занятий:

1. Выполнение этого задания поможет вам частично наверстать материал пропущенных занятий. Постарайтесь не пренебрегать им и не откладывать на последний момент.
2. В любом задании, где говорится о том, что вам не приходилось пока делать, сначала найдите в поисковике несколько примеров. Больше всего стоит ориентироваться на документацию Python и tutorial по библиотеке, используемой в каждом из заданий.
3. Дальше, когда вы уже немного представляете, о чем речь, можно обратиться к Ipython Notebook с занятием. Если вдруг у вас все еще есть проблемы с работой в Ipython Notebook — сначала почитайте, как делать это, либо попросите объяснить семинариста (это было на первом занятии).

1 Python

1. Чем отличаются типы list и tuple? Какой из них можно использовать в качестве ключа в dict и добавлять в set? Почему? На какой структуре данных реализован dict в python? Чем отличается range от xrange в Python2.x? Что такое list comprehension? Как работают функции map и reduce в Python?
2. Напишите рекурсивную функцию, возвращающую N-ное число Фибоначчи. Проверьте работу для небольших N. Выясните, для какого N начинаются проблемы с глубиной рекурсии. Дополнительное задание для любознательных: попробуйте выяснить, можно ли в Python изменить допустимую глубину рекурсии.
3. Попробуйте измерить время работы предыдущей функции при разных N.
4. Напишите функцию, возвращающую (через запятую) модуль и аргумент комплексного числа по заданной мнимой и действительной части. Возможно, вам потребуется импортировать какие-то тригонометрические функции из модуля math. Приведите примеры работы. Каков тип возвращаемого вашей функцией значения?

5. Напишите docstrings к вашей последней функции. Изучите библиотеку doctest и попробуйте проверить, что написанные в docstrings примеры выполняются, без явного копирования кода оттуда (т.е. с помощью doctest).
6. Реализуйте класс комплексных чисел с самым базовым функционалом. Цель не продемонстрировать умение спроектировать класс из каждого второго домашнего задания по любому языку программирования, а посмотреть на особенности реализации классов на питоне.
7. Изучите примеры работы с декораторами из официальной документации. Фактически, декоратор - это функция, которая принимает на вход другую функцию и возвращает её в модифицированном виде. Синтаксически предусмотрена возможность использовать декоратор, написав перед объявлением функции имя_декоратора. Попробуйте найти реализацию (или сделать сами) кэширующего декоратора и декоратора, вычисляющего время работы функции. Попробуйте применить эти декораторы в разной последовательности к функции, вычисляющей числа Фибоначчи, сравнить и объяснить результаты. Объясните также, как работают использованные вами декораторы.
8. Выясните, чем отличается задание поля x через "self.x =" от объявления метода с декоратором property.
9. Напишите setter и getter для полей класса комплексного числа.

2 Requests/urllib + BeautifulSoup

Скачайте html-код страницы с новостью с любого новостного сайта и, распарсив его с помощью библиотеки BeautifulSoup или любой другой аналогичной, получите текст статьи (без html-тегов или вкраплений java script).

3 NumPy

1. Создайте numpy.ndarray размерностью 3x4x2 и продемонстрируйте разные способы обращения по индексам из документации NumPy.
2. Попробуйте сравнить по производительности numpy.array и list, выполняя какую-то стандартную операцию с каждым из типов много раз.
3. Изучите numpy.linspace(), напишите свою реализацию аналогичной функции с помощью list comprehension, дающий тот же результат, только в виде list, а не numpy.ndarray. Сравните по производительности два варианта на достаточно больших массивах.

4 Matplotlib

1. Что делает в IPython Notebook команда %matplotlib inline?

- Используя `numpy.linspace` и генераторы, постройте в `matplotlib` график какой-нибудь элементарной функции. В минимальном варианте вам потребуются только `pyplot` из `matplotlib` и методы `pyplot.plot()` и `pyplot.show()`. Помните, что если импортируете модуль под каким-то именем (например, `plt`), то обращаться надо уже к нему, а не к `pyplot`.
- Постройте графики еще нескольких элементарных функций (на том же рисунке) и сделайте к ним подписи (легенду) с формулами (в подписях можно использовать `LATEX`, как обычно, в знаках `$`). Добавьте какие-нибудь подписи к осям координат и название графика. Сохраните изображение в файл одной строчкой на Python.
- Изучите документацию `matplotlib` и попробуйте построить на одном изображении 4 системы координат, на верхних двух - квадратичную и кубическую параболы, на нижних - экспоненту и логарифм.
- Изучите вопрос построения `heatmap` и изобразите его для функции $f(x, y) = 3xy + x - 2y$ в области $[0; 5]^2$.
- А теперь постройте трехмерный график той же функции в той же области.

5 Всё вместе + SciPy

Сгенерируйте точки (y, x) (500 будет достаточно), удовлетворяющие зависимости $y = x^3 + 2x^2 - 3x + 2$, затем, с помощью `numpy.random` добавьте к координатам y нормальный шум. С помощью `scipy` восстановите исходную зависимость, считая известной степень полинома. Вам поможет `curvefit` или `scipy.optimize.minimize`, запущенный для суммы квадратичных отклонений. У восстановленной зависимости найдите экстремумы численно. Покажите на графике все вместе: исходные точки, восстановленную зависимость, найденные экстремумы.

Дополнительное задание: подберите коэффициенты многочлена, точно проходящего через точки (без добавления нормального шума), составив и решив с помощью `scipy` систему линейных уравнений.

6 Pandas + Scikit-learn

Считайте выборку из примера про детектирование кожи на фотографии с семинара с помощью библиотеки `pandas`. Продемонстрируйте индексацию по строкам и столбцам, `labeled-based` и `index-based`. Добавьте в `DataFrame` столбцы, соответствующие попарным произведениям признаков и модулям попарных разностей. Запустите на исходных признаках и на модифицированных `KNeighboursClassifier`, `LogisticRegression` и `Random Forest` из `sklearn` и сравните качество работы каждого метода до и после преобразования признаков.

7 Дополнительное задание: Bokeh

Выполните упражнения из tutorial: <http://bokeh.pydata.org/en/latest/tutorial/index.html>

Есть три секции с упражнениями (от базовых до продвинутых), каждая будет оцениваться на столько же баллов, что и задание на scіru.