ТОП-9 БИБЛИОТЕК В PYTHON ДЛЯ ПРОФЕССИОНАЛЬНОГО АНАЛИЗА ДАННЫХ

Язык программирования Python часто используют аналитики данных. Для этого в нем существуют расширения — библиотеки, наборы готовых инструментов для более эффективной работы.

Если вы выполняете исследовательский анализ данных в Python, вам будут известны такие распространенные библиотеки, как pandas, matplotlib и seaborn и др. Все библиотеки отличные, но у каждой есть свои нюансы, на изучение или запоминание которых может потребоваться время.

В последние годы появилось несколько мощных библиотек Python с низким уровнем кода, которые значительно ускоряют и упрощают этап исследования данных и анализа проектов.

В этой статье я познакомлю вас с 10 из этих библиотек Python, которые улучшат ваш рабочий процесс анализа данных. Все они могут быть запущены в среде Jupyter notebook.

1.PANDAS: для подготовки данных

Прежде чем анализировать данные, их нужно подготовить: собрать, очистить от ошибок и дублей, структурировать. Чтобы быть уверенными в результате по окончании анализа, важно убедиться в качестве данных вначале. Библиотека для анализа данных на Python pandas помогает преобразовывать структурированные данные и содержит встроенные инструменты для их очистки.

Особенности библиотеки pandas:

- Позволяет работать с огромными объёмами данных, в том числе объединять их и разделять.
- Поддерживает DataFrames — специальные объекты, которые позволяют эффективнее анализировать данные, превращая их в индексированные структурированные массивы.

- Принимает данные из множества источников: баз данных, таблиц Excel и других. Преобразует данные разных форматов в пригодные для анализа языком Python.

С помощью pandas можно:

- Индексировать, переименовывать, сортировать и объединять массивы данных.
- Обновлять, добавлять и удалять данные.
- Восстанавливать и обрабатывать недостающие данные.
- Визуализировать данные.

Для установки pandas выполним в командной строке команду:

```
pip install pandas
```

Библиотека pandas может быть импортирована следующим образом:

```
import pandas as pd
```

В библиотеке pandas определены два класса объектов для работы с данными:

- Series — одномерный массив, который может хранить значения любого типа данных;
- DataFrame — двумерный массив (таблица), в котором столбцами являются объекты класса Series.

2.YData (ранее профилирование Pandas)

Библиотека профилирования YData, ранее известный как Профилирование Pandas, позволяет создавать подробные отчеты на основе фрейма данных pandas. Он очень прост в навигации и предоставляет информацию об отдельных переменных, анализе отсутствующих данных, корреляциях данных и взаимодействиях.

Одна небольшая проблема с профилированием YData - это возможность обрабатывать большие наборы данных, что может замедлить генерацию отчета.

Профилирование YData может быть установлено через терминал с использованием pip:

```
pip install ydata-profiling
```

После установки библиотеки в вашей среде Python мы можем просто импортировать ProfileReport модуль из библиотеки вместе с pandas. Pandas используется для загрузки наших данных из файла CSV или другого формата.

```
import pandas as pd
from ydata_profiling import ProfileReport

df = pd.read_csv('Data/Xeek_Well_15-9-15.csv')
ProfileReport(df)
```

Как только данные будут прочитаны, мы можем передать наш фрейм данных в ProfileReport , и отчет начнет генерироваться.

Время, необходимое для создания отчета, будет зависеть от размера вашего набора данных. Чем больше набор данных, тем больше времени потребуется для его создания.

3. NumPy: для углублённых расчётов

После того как библиотека pandas помогла убедиться в качестве данных, можно перейти к расчётам. Например, посчитать выручку торговой точки по номенклатуре товара. В Excel пришлось бы объединять, суммировать и делить, а в Python может хватить одной строки записи, чтобы сделать расчёт по таблице из 10 000 строк. В этом помогает библиотека NumPy. Она считается одной из основных библиотек Python для анализа данных.

Особенности библиотеки NumPy:

- Множество структур данных, которые позволяют эффективнее проводить поиск, аналитику и структурирование.

- Возможность проводить сложные научные расчёты с математическими формулами, в том числе над данными в многомерных массивах.
- Инструменты для преобразования данных в разные форматы.
- Работа с числовыми и другими типами данных.

С помощью NumPy можно:

- Умножать, добавлять, выравнивать, индексировать массивы, проводить их срезы, изменять форму.
- Создавать стековые и широковещательные массивы, разбивать их на секции.
- Проводить вычисления по формулам линейной алгебры, которые нужны для сложного анализа данных на Python.

Для установки NumPy выполним в командной строке команду:

```
conda install -c anaconda numpy
```

NumPy или Numerical Python — это библиотека Python, которая предлагает следующее:

1. Мощный N-мерный массив
2. Высокоуровневые функции
3. Инструменты для интеграции кода C/C++ и Fortran
4. Использование линейной алгебры, Преобразований Фурье и возможностей случайных чисел

4. SciPy: для математических операций

С увеличением опыта специалиста будут усложняться и задачи: придётся прибегать к линейной алгебре, интерполяции, интеграции, статистике и другим сложным математическим операциям. В этом специалисту по анализу данных помогает библиотека SciPy, которая построена на базе массивов и функций NumPy.

Особенности SciPy:

- Быстрое и надёжное выполнение сложных операций благодаря оптимизации.
- Широкий набор функций и инструментов для разнообразных операций.
- Содержит множество подпакетов для конкретных задач, например преобразования Фурье.

С помощью SciPy можно:

- Проводить сложные математические вычисления: например, решать дифференциальные уравнения или находить численное решение интегралов.
- Обрабатывать изображения.
- Работать с генетическими алгоритмами.
- Проводить сложные инженерные вычисления.

Для установки SciPy выполним в командной строке команду:

```
pip install scipy
```

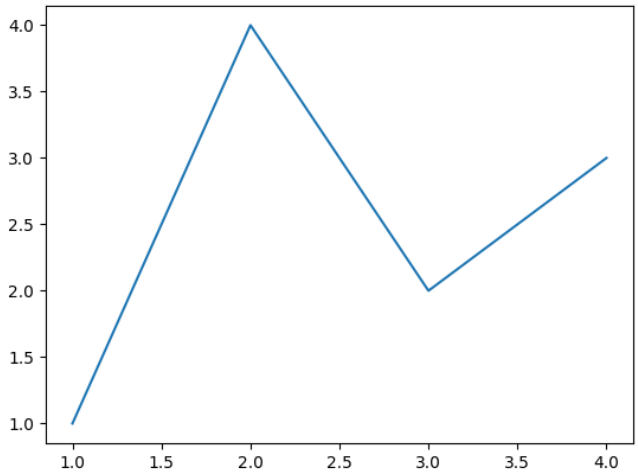
SciPy предоставляет набор специальных функций, используемых в математической физике: эллиптические настраиваемые функции, гамма, бета и так далее. Для их поиска нужно использовать функцию help().

5. Matplotlib: для визуализации

После анализа данные нужно представить в удобном для восприятия виде. Для этого используют инструменты визуализации. Они есть в некоторых других пакетах, но Matplotlib поддерживает максимум различных графиков и диаграмм.

Matplotlib отображает ваши данные на Figures (например, windows, виджеты Jupyter и т. Д.), Каждый из которых может содержать одну или несколько Axesобластей, Где точки могут быть указаны в терминах координат xy (или тета-r на полярном графике, x-y-z на 3D-графикеи т.д.). Самый простой способ создания фигуры с осями - использовать pyplot.subplots. Затем мы можем использовать Axes.plotдля рисования некоторых данных по осям:

```
рис, ax = plt.подзаголовки()  # Создайте фигуру, содержащую одну ось.
ax.plot([1, 2, 3, 4], [1, 4,  2, 3])  # Нанесите некоторые данные на оси.
```



Matplotlib позволяет строить графики такого вида на основе данных и математических функций.

Особенности Matplotlib:

- Позволяет быстро строить диаграммы и графики разных видов, настраивать их оформление.
- Поддерживает API для интеграции графиков в разработанные приложения.
- Умеет форматировать диаграммы и графики для более простого восприятия.

С помощью Matplotlib можно:

- Строить 2D-фигуры.
- Формировать на основе данных линейные, точечные, столбчатые, круговые и другие диаграммы.
- Рисовать контурные графики.
- Формировать поля векторов и спектрограммы.
- Быстро встраивать визуализацию в сервисы, программы и приложения.

6. SweetViz

SweetViz - это еще одна библиотека с низким уровнем кода для интерактивной визуализации и исследования данных. С помощью пары строк кода мы можем создать интерактивный HTML-файл для изучения наших данных.

Sweetviz можно установить через терминал, используя pip:

```
pip install sweetviz
```

После установки мы можем импортировать его в наш ноутбук и загружать наши данные с помощью pandas.

```
import sweetviz as sv
import pandas as pd

df = pd.read_csv('Data/Xeek_Well_15-9-15.csv')
```

Одна из небольших проблем, которые обнаружились в этой библиотеке, заключается в том, что вам нужен широкий экран, чтобы иметь возможность просматривать весь горизонтальный контент без прокрутки.

7. Statsmodels: для статистического анализа

В Python очень мало встроенных инструментов для статистического анализа — этим он уступает некоторым другим языкам для анализа данных, например R. Библиотека statsmodels исправляет этот недостаток. Она объединяет графические возможности Matplotlib, инструменты подготовки данных pandas и математический функционал NumPy и SciPy. В неё встроены некоторые возможности библиотеки Patsy, которые позволяют реализовать формулы из языка R.

Особенности statsmodels:

- Позволяет эффективнее работать на Python тем, у кого есть опыт в R, так как поддерживает многие методы из этого языка.
- Подходит для статистических вычислений.
- Поддерживает одномерный и двумерный анализ данных, что позволяет строить обобщённые модели и проверять гипотезы.

- Чаще всего применяется специалистами по Data Science для сложных вычислений и машинного обучения.
- Хорошо совместима с другими библиотеками и инструментами Python.
- Упрощает решение некоторых сложных математических задач.

С помощью statsmodels можно:

- Строить сложные статистические модели, например линейную регрессию.
- Проводить статистические тесты.
- Вычислять корреляцию.
- Строить обобщённые линейные и байесовские модели.
- Проверять гипотезы различными методами.

Для установки Statsmode выполним в командной строке команду:

```
import statsmodels.api as sm
```

Statsmodels поддерживает определение моделей с использованием формул в стиле R и pandas фреймов данных.

8. Plotly: для трёхмерной визуализации

Иногда для аналитики необходимы не просто графики и диаграммы, а более сложные конструкции: карты, трёхмерные диаграммы и другие сущности. Plotly поддерживает практически все виды визуализаций, которые используют в науке и анализе данных. Изюминка библиотеки Plotly — в её интерактивности: можно водить по графику мышкой и видеть значения срезов данных.

Особенности Plotly:

- Поддерживает трёхмерные визуализации и их продвинутые настройки.
- Позволяет экспортировать результаты анализа в особом формате — JSON. Его удобно открывать в других приложениях.

- Обладает одним из самых широких списков поддерживаемых диаграмм.
- Умеет отправлять данные в облачные сервисы, чтобы работать там с ними дальше.
- На основе этой библиотеки существует ещё одна, Dash — она позволяет строить интерактивные дашборды для демонстрации данных.

С помощью Plotly можно:

- Строить любые обычные диаграммы и графики: круговые, Ганта, древовидные.
- Формировать научные карты: тепловые, контурные, логарифмические, с полями векторов.
- Строить финансовые графики.

Plotly можно установить через терминал, используя pip:

```
pip install plotly
```

Перед началом работы необходимо импортировать модуль. В разных частях шпаргалки для разных задач нам понадобятся как основной модуль, так и один из его подмодулей.

9. Scikit-learn: для машинного обучения

Обычно моделями машинного обучения занимаются специалисты по Data Science, однако аналитикам тоже иногда приходится иметь с ними дело. Как правило, для их написания и настройки используют scikit-learn. Это одна из основных библиотек Python для Data Science.

Особенности библиотеки scikit-learn:

- В библиотеку встроены все базовые функции для машинного обучения.
- Можно создавать модели для обучения как с учителем, так и без учителя.
- Доступно подключение механизмов оценки созданных моделей.

- Есть интеграция с NumPy, SciPy и другими библиотеками для вычислений.

С помощью scikit-learn можно:

- Создавать машинные модели для классификации, кластеризации, сегментации, визуализации данных и других манипуляций.
- Выбирать модели из нескольких.
- Настраивать параметры и особенности модели.
- Предварительно обрабатывать входные данные для обучения.

Для установки Scikit-learn выполним в командной строке команду:

```
pip install -U scikit-learn
```

Он также предоставляет несколько наборов данных, которые вы можете использовать для тестирования ваших моделей.

Scikit-Learn поддерживает:

- предварительную обработку данных;
- уменьшение размерности;
- выбор модели;
- регрессии;
- классификации;
- кластерный анализ.

Список литератур

1.