УДК

ОСНОВНЫЕ БИБЛИОТЕКИ В PYTHON ДЛЯ ПРОФЕССИОНАЛЬНОГО АНАЛИЗА ДАННЫХ

Воронкин Р.А., Уланбекова А.У.

Постановка задачи: изучение основных библиотек Python для анализа данных, ознакомиться с основными функциями и возможностями каждой из этих библиотек и применить полученные знания на практике, используя реальные данные для проведения анализа и визуализации данных. **Цель работы:** заключается в изучении основных библиотек Python для профессионального анализа данных и их применении на практике. **Используемые методы**: метод анализа, метод сравнения. **Результат:** вывод о том, что каждая из библиотек имеет свои преимущества и недостатки в зависимости от конкретной задачи, поэтому важно уметь выбирать наиболее подходящий инструмент для решения задачи анализа данных. **Практическая значимость:** библиотеки в Python для профессионального анализа данных имеют огромную практическую значимость в различных областях, таких как маркетинг, финансы, медицина, наука и т.д. Применение этих библиотек может помочь компаниям улучшить свою эффективность, увеличить прибыль и улучшить качество продуктов и услуг. В медицине можно использовать анализ данных для выявления патологий и прогнозирования заболеваний. В науке анализ данных может помочь в проведении исследований и выявлении новых закономерностей.

Ключевые слова: Python, анализ данных, библиотеки, Pandas, NumPy, Matplotlib, SciPy, машинное обучение, классификация, регрессия, кластеризация, визуализация данных, статистический анализ данных, корреляция, регрессия, оценка моделей, выбор параметров модели, маркетинг, финансы, медицина, наука.

Актуальность темы

Python - это один из самых популярных языков программирования в мире. Он широко используется в различных областях, таких как наука о данных, машинное обучение, веб-разработка и многое другое. Одной из причин его популярности является наличие большого количества библиотек, которые облегчают разработку и расширяют функциональность языка. Если вы выполняете исследовательский анализ данных в Python, вам будут известны такие распространенные библиотеки, как pandas, matplotlib и scipy и

др. Все библиотеки отличные, но у каждой есть свои нюансы, на изучение или запоминание которых может потребоваться время.

Библиотеки Python - это необходимый инструмент для любого программиста, который хочет ускорить процесс разработки и создать более эффективный код. Они помогают упростить сложные задачи и расширить возможности языка программирования Python. Благодаря библиотекам Python, программисты могут создавать более качественные программы за меньшее 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 6 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

Библиотека Pandas - это мощный инструмент для работы с данными в Python. Она предоставляет высокоуровневые структуры данных, такие как DataFrame и Series, которые упрощают работу с таблицами и временными рядами.

Одним из ключевых преимуществ Pandas является возможность чтения и записи файлов различных форматов, таких как CSV, Excel, SQL и другие. Это позволяет легко импортировать данные из разных источников и сохранять результаты работы в нужном формате.

Pandas также предоставляет мощные инструменты для обработки и анализа данных. Это включает в себя функции для фильтрации, сортировки, группировки и агрегации данных. Библиотека также предоставляет возможность работать с пропущенными значениями и обрабатывать ошибки при чтении данных.

Другим важным преимуществом Pandas является возможность создания графиков и визуализации данных. Библиотека Matplotlib интегрирована в Pandas, что позволяет создавать красивые и информативные графики без необходимости использования дополнительных инструментов.

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 целом, библиотека Pandas является незаменимым инструментом для работы с данными в Python. Она предоставляет множество функций для обработки и анализа данных, а также упрощает чтение и запись файлов различных форматов. Благодаря Pandas, программисты могут быстро и эффективно обрабатывать и анализировать данные, что делает эту библиотеку популярным выбором в научных и коммерческих проектах.

2. NumPy: для углублённых расчётов

Библиотека NumPy является одной из самых популярных библиотек для работы с числовыми массивами и матрицами в Python. Она предоставляет множество функций для выполнения операций над массивами, таких как математические операции, операции логического сравнения, сортировку, фильтрацию и т.д.

NumPy также предоставляет функции для работы с многомерными массивами, что делает ее особенно полезной для научных вычислений и обработки изображений.

Библиотека NumPy широко используется в научных проектах, таких как анализ данных, машинное обучение, обработка изображений, статистика и многих других областях. Она также является одной из основных библиотек для работы с другими популярными библиотеками, такими как Pandas и Matplotlib. Кроме того, NumPy имеет высокую производительность благодаря своей реализации на языке C. Это позволяет выполнять операции над массивами быстрее, чем при использовании стандартных структур данных Python.

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

Таким образом, можно сделать вывод о том, что библиотека NumPy является необходимой для работы с числовыми данными в Python и широко используется в научных и коммерческих проектах.

3. SciPy: для математических операций

Библиотека SciPy является расширением библиотеки NumPy и предоставляет множество функций для выполнения научных вычислений, таких как численное интегрирование, оптимизация, обработка сигналов и изображений, статистика и многие другие. Одной из главных особенностей библиотеки является ее способность работать с различными типами данных, включая разреженные матрицы, что делает ее особенно полезной для обработки больших объемов данных.

Библиотека SciPy также предоставляет множество алгоритмов для решения научных задач, таких как решение дифференциальных уравнений, нахождение корней уравнений, аппроксимация функций. Она широко используется в научных и инженерных проектах, таких как моделирование физических процессов, обработка сигналов и изображений, анализ данных и машинное обучение.

Кроме того, библиотека SciPy имеет высокую производительность благодаря своей реализации на языке C и Fortran. Это позволяет выполнять научные вычисления быстрее, чем при использовании стандартных структур данных Python.

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().

Таким образом, можно сделать вывод о том, что библиотека SciPy является необходимой для выполнения научных вычислений в Python и широко используется в научных и инженерных проектах.

4. Matplotlib: для визуализации

Библиотека Matplotlib является одной из самых популярных библиотек для визуализации данных в Python. Она предоставляет множество функций

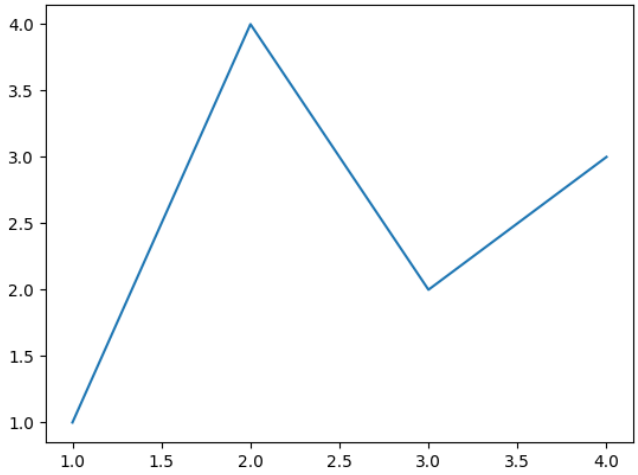
для создания графиков, диаграмм и других типов визуализации. Одной из главных особенностей библиотеки является ее гибкость и настраиваемость. Она позволяет создавать красивые и информативные графики, управляя каждым аспектом их внешнего вида.

Библиотека Matplotlib также поддерживает множество форматов файлов для сохранения графиков, включая PNG, PDF, SVG и другие. Это делает ее удобной для использования в научных публикациях и документации. Она широко используется в научных и инженерных проектах, таких как анализ данных, моделирование и визуализация результатов вычислений.

Кроме того, библиотека Matplotlib имеет большое сообщество пользователей и разработчиков, которые постоянно работают над улучшением ее функциональности и производительности. 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:

```
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.

Таким образом, можно сделать вывод о том, что библиотека Matplotlib является необходимой для визуализации данных в Python и широко используется в научных и инженерных проектах.

5. SweetViz

Библиотека SweetViz является относительно новым инструментом для анализа данных и создания отчетов о них. Она предоставляет множество функций для автоматической визуализации и анализа данных, включая графики распределения, матрицы корреляции, сводные таблицы и многое другое.

Одной из главных особенностей библиотеки является ее простота использования. Она позволяет быстро создавать отчеты о данных без необходимости написания сложного кода.

Библиотека SweetViz также поддерживает множество форматов файлов для сохранения отчетов, включая HTML, Markdown и Jupyter Notebook. Это делает ее удобной для использования в научных публикациях и документации. Она может быть полезна в различных областях, таких как анализ данных, машинное обучение и бизнес-анализ.

Кроме того, библиотека SweetViz имеет активное сообщество пользователей и разработчиков, которые постоянно работают над улучшением ее функциональности и производительности.

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.

Таким образом, можно сделать вывод о том, что библиотека SweetViz является полезным инструментом для анализа данных и создания отчетов о них, и может быть использована в различных областях. Однако, ее

функциональность все еще ограничена по сравнению с более универсальными библиотеками, такими как Matplotlib.

6. Statsmodels: для статистического анализа

Библиотека Statsmodels является мощным инструментом для анализа данных и статистического моделирования. Она предоставляет широкий спектр функций для работы с данными, включая описательную статистику, регрессионный анализ, временные ряды, анализ выживаемости и многое другое.

Одной из главных особенностей библиотеки является ее способность работать с различными типами данных, включая числовые, категориальные и временные ряды. Она также предоставляет широкий спектр статистических методов для анализа данных, включая метод максимального правдоподобия, метод наименьших квадратов и множество других.

Библиотека Statsmodels также поддерживает множество форматов файлов для сохранения результатов анализа, включая HTML, Markdown и Jupyter Notebook. Это делает ее удобной для использования в научных публикациях и документации. Она может быть полезна в различных областях, таких как экономика, финансы, социология и медицина.

Кроме того, библиотека Statsmodels имеет активное сообщество пользователей и разработчиков, которые постоянно работают над улучшением ее функциональности и производительности.

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.

Таким образом, можно сделать вывод о том, что библиотека Statsmodels является мощным инструментом для анализа данных и статистического моделирования, и может быть использована в различных областях. Однако, ее использование требует определенных знаний и навыков в области статистики и анализа данных.

Сравнение библиотек в Python

Pandas2.0 / Polars

Библиотека Polars - это быстрая и эффективная библиотека для обработки и анализа данных на основе Rust. Она предоставляет высокопроизводительные структуры данных и инструменты для работы с ними, такие как фильтрация, сортировка, группировка, агрегирование и многое другое. Polars также поддерживает множество форматов данных, включая CSV, Parquet и Arrow.

Pandas 2.0 - это новая версия библиотеки Python для анализа данных, которая включает в себя улучшенную производительность и новые функции.

Она также предоставляет высокопроизводительные структуры данных и инструменты для работы с ними, такие как фильтрация, сортировка, группировка, агрегирование и многое другое. Pandas 2.0 также поддерживает множество форматов данных, включая CSV, Excel и SQL.

Обе библиотеки предназначены для работы с данными и имеют множество функций для обработки и анализа данных. Однако Polars использует Rust для повышения производительности и эффективности, что делает его более быстрым и масштабируемым, чем Pandas. Кроме того, Polars имеет удобный интерфейс для работы с большими объемами данных и поддерживает множество форматов данных, включая Arrow.

Критерий сравнения	Pandas2.0	Polars
С чем работает	Pandas 2.0 работает с данными в форматах CSV, Excel, SQL и многих других.	Polars работает с данными в форматах CSV, Parquet, JSON и многих других.
Сильные инструменты	Улучшенная производительность: использование оптимизированных алгоритмов и параллельных вычислений позволяет обрабатывать большие объемы данных быстрее.	Высокопроизводительные структуры данных: DataFrame и Series, которые позволяют быстро обрабатывать большие объемы данных.
Как используется	Pandas2.0 предоставляет новые функции для работы с временными рядами и категориальными данными, а также поддержку параллельных вычислений для обработки больших объемов данных.	Polars поддерживает работу с временными рядами и категориальными данными, а также предоставляет возможность создания сводных таблиц и графиков.

Вывод: в целом, выбор между Polars и Pandas 2.0 зависит от того, что вы хотите сделать с вашими данными. Если вам нужно обрабатывать большие объемы данных и получать результаты быстрее, чем в Pandas, то Polars - это лучший выбор. Если же вы предпочитаете работать с Python и уже знакомы с Pandas, то Pandas 2.0 может быть более удобным выбором.

Вывод

Тема "Библиотеки в Python для профессионального анализа данных" включает в себя различные инструменты и технологии, которые используются для обработки, анализа, визуализации и моделирования данных. Основные библиотеки, которые используются в этой области, включают Pandas, NumPy, Matplotlib и SciPy. Они предоставляют мощные инструменты для работы с данными, такие как фильтрация, сортировка, группировка, агрегация, преобразование и визуализация.

Кроме того, эти библиотеки также предоставляют возможности для машинного обучения, такие как классификация, регрессия и кластеризация. Они также позволяют проводить статистический анализ данных, оценку моделей и выбор параметров модели.

Таким образом, использование библиотек в Python для профессионального анализа данных является необходимым для различных отраслей, таких как маркетинг, финансы, медицина и наука. Они позволяют быстро и эффективно обрабатывать большие объемы данных и получать ценные инсайты для принятия решений.

Основная литература:

1. Программирование на языке высокого уровня Python : учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2019. — 161 с. — (Серия : Бакалавр. Прикладной курс)

2. Уэс Маккини. Python для анализа данных: обработка данных с помощью Pandas, NumPy и IPython. O'Reilly Media, 2017.

3. Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил. — (Серия «Бестселлеры O’Reilly»)

4. Джейк ВандерПлас. Руководство по науке о данных на Python: необходимые инструменты для работы с данными. O'Reilly Media, 2016.

Интернет-ресурсы:

1. Самоучитель Python [Электронный ресурс] / https://pythonworld.ru/. Режим доступа: https://pythonworld.ru/samouchitel-python, свободный.

2. Самоучитель Python [Электронный ресурс] / http://pythoshka.ru/. Режим доступа: http://pythoshka.ru/p1138.html/samouchitel-python/p1138.html, свободный.

3. DataCamp. Шпаргалка Pandas: обработка данных на Python. https://www.datacamp.com/community/blog/python-pandas-cheat-sheet#gs.2e1z0s

4. Документация NumPy. https://numpy.org/doc/

5. Документация Pandas. https://pandas.pydata.org/docs/