УДК 004.428.4

PYTHON - КАК ИНСТРУМЕНТ ДЛЯ АНАЛИЗА ДАННЫХ

Ермаков Олег Александрович
студент
Брозгунова Надежда Петровна
кандидат экономических наук, доцент
nadyazhm@mail.ru
Мичуринский государственный аграрный университет
г. Мичуринск, Россия

Аннотация. В статье рассматриваются возможности языка Python для анализа данных, дается сравнительная характеристика особенностей его использования, приводится пример его использования при анализе временного ряда. Показаны механизмы использования Python в крупных IT корпорациях при анализе данных.

Ключевые слова: Python, Data science, анализ данных, языки программирования, временные ряды.

Data science — одна из самых востребованных областей на сегодняшний день, а Python — один из самых популярных инструментов для анализа данных. Популярность языка Python обусловлена простотой его использования, способностью обрабатывать практически любой объем данных и возможностью создавать и развертывать индивидуальные аналитические приложения.

Python можно применять для команды, в которой есть как разработчики, так и специалисты по Data Science. Перечислим свойства этого языка, за которые его можно считать универсальным языком для анализа данных [1, 2].

Высокая продуктивность разработки. Язык интерпретируемый, поэтому на нём можно писать быстрее, чем, например, на C. Неявная, но строгая типизация обеспечивает меньший объём кода для решения задач, чем в Java. А лаконичный и ясный синтаксис позволяет быстро писать читабельный код. Сравнительная характеристика написанная одной и той же функции (расчёт факториала) на Java и на Python приведена в таблице 1.

Таблица 1

Программный код написания функции на Java и на Python

Java	Python

`class Factorial`
`{`
`  static int factorial(int n)`
`  {`
`  if (n == 0)`
`  return 1;`
`  return n*factorial(n-1);`
`  }`
 	
`  public static void main(String[] args)`
`  {`
`  System.out.println(factorial(5));`
`  }`
`}`
	`def factorial(n):`
	`  return 1 if (n==1 or n==0) else n * factorial(n - 1)`
	
	`print(factorial(5))`

Низкий порог входа для изучения. То, что Python широко используется в области Big Data, частично связано со скоростью его освоения. Потребность в анализе данных чаще всего возникает у тех, кто управляет бизнесом — аналитиков, экономистов. Осваивать тяжеловесные языки типа Java или C им нецелесообразно — в отличие от Python, который можно изучить довольно

быстро. В таблице 2 приведена сравнительная характеристика интерактивности языков Python, Java и C [2].

Таблица 2

«Интерактивность» языка (расчёты без компиляции)

Python	Introduction To Python 3: (Python Documentation Manual Part 1 & 2) by Guido Van Rossum	506 стр.
C/C++	The C++ Programming Language, 4th Edition by Bjarne Stroustrup	1376 стр.
Java	Java: The Complete Reference, 11th Edition by Herbert Schildt	1248 стр.

Аналитики также ценят Python за то, что благодаря встроенному интерпретатору он позволяет кодировать на ходу. В Data Science это актуально для проверки гипотез в интерактивном режиме.

Динамичное развитие языка. Ещё одним аргументом в пользу Python является то, что этот язык быстро и интенсивно развивается. С каждой версией производительность языка повышается, а синтаксис совершенствуется. Например, в версии 3.8 появился новый walrus оператор — `:=`, что для любого языка достаточно серьёзное событие. В низкоуровневых языках типа C++ или Java темп изменений заметно ниже — их утверждает специальная комиссия, которая собирается раз в несколько лет. В Python процесс стандартизации более открыт для комьюнити, каждый может предложить свои идеи, и их количество быстро растёт [3].

Результат. Разработанная система позволяет автоматизировать и планировать регулярные расчёты 45 атрибутов покупательского поведения (часть рассчитывается раз в неделю, часть — раз в месяц). Объём данных, накопленных в результате этих расчётов за три года, составляет 4,5 терабайт, и другие департаменты компании имеют возможность легко обращаться к ним и использовать их для своей работы. При этом система ориентирована на расширение своих функций и реализацию новых.

Таким образом, Python позволяет решать самые разноплановые задачи. Он объединяет в проекте разработчиков и специалистов, для которых программирование не является профильным навыком — бизнес-аналитиков, дата-аналитиков, дата-сайентистов [1, 4]. Отлично подходит для Agile-разработки, для гибкой оптимизации. Для компании, у которой много разноуровневых задач и ведётся работа с большими данными, Python является отличным дополнением к компетенциям в низкоуровневых языках.

Пример обработки данных временных рядов в Python представлен в таблице 3.

Таблица 3

Обработка данных временных рядов в Python

Алгоритм действий	Программный код Python
1. Импорт Пакетов данных	import numpy as np import matplotlib.pyplot as plt import pandas as pd
2. Определение функции, которая будет считывать данные из входного файла	def read_data(input_file): input_data = np.loadtxt(input_file, delimiter = None)
3. Преобразование данных во временные ряды. Создаем диапазон дат, с частотой данных один месяц. Наш файл содержит данные, которые начинаются с января 1950 года	dates = pd.date_range('1950-01', periods = input_data.shape[0], freq = 'M')
4. Создаем данные временных рядов с помощью Pandas Series	output = pd.Series(input_data[:, index], index = dates) return output if __name__=='__main__':

5. Ввод путя к входному файлу	input_file = "/Users/admin/AO.txt"
6. Преобразование столбца в формат временных рядов	timeseries = read_data(input_file)
7. Вывод графика и визуализация данных	plt.figure() timeseries.plot() plt.show()

Графическое представление данные показаны на рисунке 1.

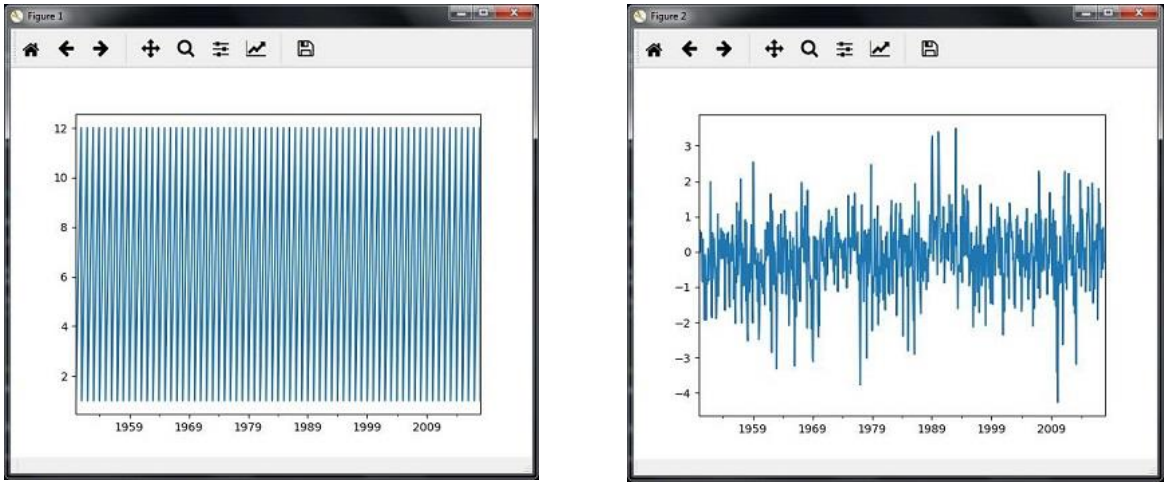

Рисунок 1 – Графическое представление данных временных рядов в Python

Не смотря на свою простоту, язык Python досатоно широко используется мировыми корпорациями.

Google поддерживал Python с самого начала. Вначале основатели Google приняли решение «Использовать Python, там, где мы можем, C++, где должны». Это означало, что C++ использовался там, где контроль над памятью был обязателен и была желательна низкая задержка. В других аспектах Python обеспечивает простоту поддержки и относительно быстрый отклик [1, 2].

Даже когда другие скрипты писались для Google на Perl или Bash, они часто переписывались на Python. Причина была в простоте развертывания и поддержки. На самом деле, по словам Стивена Леви, автора «In the Plex», самый первый парсер Google для сканирования веб-страниц был изначально написан на Java 1.0 и оказался настолько сложным, что они переписали его на Python.

Python теперь является одним из официальных серверных языков Google — C++, Java и Go — три других, которые разрешено развертывать в рабочей среде. И, в случае, если вы не уверены, насколько важен Python для Google, с 2005 по 2012 год в Google работал сам создатель Python, Гвидо ван Россум.

В довершение всего Питер Норвиг сказал:

«Python был важной частью Google с самого начала и остается таковым по мере роста и развития системы. Сегодня десятки инженеров Google используют Python, и мы ищем больше людей, владеющих этим языком» [5].

Facebook. Инженеры Facebook исключительно заинтересованы в Python, что делает его третьим по популярности языком среди гигантов социальной сети (сразу за C++ и их проприетарным PHP-диалектом Hack). В целом, было сделано более 5000 коммитов к утилитам и сервисам в Facebook, которые управляют инфраструктурой, бинарным распределением, отображением оборудования и операционной автоматизацией.

Простота использования библиотек Python означает, что производственным инженерам не нужно писать или поддерживать слишком много кода, что позволяет им сосредоточиться на улучшениях. Это также гарантирует то, что инфраструктура Facebook способна эффективно масштабироваться. Python в настоящее время отвечает за несколько сервисов по управлению инфраструктурой. К ним относятся использование TORconfig для настройки и формирования сетевого коммутатора, FBOSS для CLI коммутатора whitebox и использование Dapper для планирования и выполнения работ по техническому обслуживанию. Facebook опубликовал множество проектов на Python с открытым исходным кодом, написанных для Py3, включая Facebook Ads API и фреймворк Python Async IRCbot. Facebook в настоящее время находится в процессе обновления своей инфраструктуры и обработчиков до Python 3.4, и AsyncIO помогает их инженерам в этом процессе.

Instagram. В 2016 году командой инженеров Instagram был запущен крупнейший в мире Django-проект, полностью написанный на Python. Мин Ни,

инженер-программист из Instagram, рассказал о своем опыте инженеринга на Python для этого проекта:

«Изначально мы решили использовать Python из-за его репутации, а также за простоту и практичность, которая хорошо согласуется с нашей философией «сделай простое в первую очередь».

С тех пор команда разработчиков Instagram инвестировала время и ресурсы в поддержание жизнеспособности использования Python в огромном масштабе (~800 миллионов активных пользователей в месяц) [1, 6].

Spotify. Этот гигант стриминговой музыки — огромный сторонник Python, использующий язык в основном для анализа данных и сервисов. На бэкенде работает большое количество утилит, которые взаимодействуют через 0MQ или ZeroMQ, сетевую библиотеку с открытым исходным кодом и инфраструктуру, написанную на Python и C++.

Spotify нравится, как быстро происходит разработка при программировании на Python. В последних обновлениях архитектуры Spotify используется Gevent, который обеспечивает быстрый цикл обработки событий с высокоуровневым синхронным API. Для предоставления предложений и рекомендаций пользователям, Spotify использует большой объем аналитики. Для их интерпретации Spotify использует Luigi, модуль Python, который синхронизируется с Hadoop. Этот модуль с открытым исходным кодом обрабатывает то, как библиотеки работают вместе, и быстро объединяет журналы ошибок, чтобы дать возможность как можно быстрее устранить проблемы.

В общей сложности Spotify использует более 6000 отдельных сервисов на Python, которые работают вместе на узлах кластера Hadoop [4].

Таковы лишь некоторые особенности использования языка Python для анализа данных, которые по праву позволяют считать его передовым средством в данной области.

Список литературы:

1. PYTHON как современный язык программирования / Л.И. Никонорова, М.Г. Тимофеев, А.П. Кузнецова // Наука и Образование. – 2019. – Т. 2. – № 2. – С. 263

2. Сравнение нормального распределения и эмпирической функции распределения при статистической обработке результатов измерений / Н.В. Картечина, Л.В. Бобрович, Н.В. Пчелинцева [и др.] // Наука и Образование. – 2019. – Т. 2. – № 3. – С. 20

3. Разработка и внедрение электронной картотеки внутренних документов для автоматизации процессов реализации управленческих задач на предприятии / В.С. Васильев, Д.О. Иванов, О.А. Кулешов [и др.] // Наука и Образование. – 2020. – Т. 3. – № 2. – С. 17.

4. Аникьева, Э.Н. Интернет и киберпреступность / Э.Н. Аникьева, А.А. Дегтерева // Наука и Образование. – 2020. – Т. 3. – № 2. – С. 14

5. Проектирование и реализация интерактивной специализированной информационно-справочной системы / С.В. Федоров, И.В. Уколов, А.А. Лукин [и др.] // Наука и Образование. – 2020. – Т. 3. – № 2. – С. 3.

6. Копцев, П.Ю. Влияние информационных технологий на рост синергетического эффекта в АПК // П.Ю. Копцев, Н.В. Картечина, Ю.А. Скрипко // В сб.: Инженерное обеспечение инновационных технологий в АПК: материалы Международной научно-практической конференции – Мичуринск: Мичуринский государственный аграрный университет, 2018. – С. 187-190.

UDC 004.428.4

PYTHON - AS A TOOL FOR DATA ANALYSIS

Ermakov Oleg Alexandrovich
student
Brozgunova Nadezhda Petrovna
Candidate of Economic Sciences, Associate Professor
nadyazhm@mail.ru
Michurinsky State Agrarian University
Michurinsk, Russia

Annotation. The article discusses the capabilities of the Python language for data analysis, provides a comparative characteristic of the features of its use, and gives an example of its use in analyzing a time series. The mechanisms of using Python in large IT corporations in data analysis are shown.

Key words: Python, Data science, data analysis, programming languages, time series.