Booz Allen®

Automation and Process Improvement in Cost Estimating

Anil Divvela

Agenda

- Introduction
- Python Overview
- Popular Ways to use Python
- Walk-Through of Simple Use Cases
- Conclusion
- Questions
- Resources

Introduction

Anil Divvela, M.B.A., CCEA®

Background:

- 13 years in Cost Estimating and Analysis
- B.S., Industrial Engineering, Purdue
- M.B.A, Boston College
- Certified Cost Estimator Analyst (CCEA)

Objective:

- Examine alternative approaches to completing simple tasks through the use of Python
- Automating small tasks can lead to efficiencies in various stages of an analysis

Python Overview

What is Python?

- A general-purpose programming language that can run on Unix, Windows, or Macs

Benefits

- Easy to Learn!
- Saves time for recurring tasks
- Open Source
- Versatility

How to get started?

- Download the latest stable version of Python and an Integrated Development Environment (IDE)

**verify your IT policies*

Basic terminology to be aware of

- IDE
- Library
- Function
- Variable

Popular Ways to Use Python

Low Complexity*

- Search for text across files
- Read and calculate data in Excel spreadsheets of any size
- Reformatting raw data
- File organization
- Send reminder emails and text notifications
- Email organization
- Input validation
- Split, merge, watermark, and encrypt PDFs
- Fill out online forms
- Create and write new files

Medium Complexity

- Web scraping
- Extract Transact Load (ETL) of data from outside sources and saved to relational database
- Data Analysis
- Data Visualization
- Reports and Dashboard creation (can integrate with Power BI and other software)
- Image processing

High Complexity

- Financial modeling and forecasting
- Web development
- Artificial Intelligence and Machine Learning
- Automated test cases
- Blockchain development
- Robotics applications
- Virtual reality applications

** Source: Automate the Boring Stuff by Al Sweigart*

Use Case 1: Search and Replace Text within Multiple Files

Business Case: As a time savings measure, most analysts utilize templates or previously developed products in support of new projects. Populating names of projects, organizations, even people can be time consuming. If the project(s) you’re supporting have various documents (acquisition, budget, reporting), then there may be a significant investment in your time to update said information.

Purpose: To automate searching for and replacing text across multiple files

Step 1: Reset directory path to where files are located

Step 2: Define text replacement function

Step 3: Replace text

Step 4: Confirm / QA results

Python Libraries:

- OS: provides functions for interacting with the operating system

Use Case 2: Download and Extract (Unzip) Files from the Internet

Business Case: Data collection is a very common responsibility for an analyst. Gathering recurring performance reports, metrics, or other files that are needed can be time consuming. Python can help with automating the downloading, extracting, and organizing of files.

Purpose: To automate downloading and extracting files from the internet

Step 1: Determine your parameters (website, file format)

Step 2: Run request for downloading the ZIP file

Step 3: Download ZIP file

Step 4: Read ZIP file as an Object

Step 5: Extract/Unzip file into a specific location

Python Libraries:

- Requests: a library for making HTTP requests.
- Zipfile: provides functions to create, read, write, append, and list ZIP files

Use Case 3: Gathering Data from Online Sources

Business Case: Analysts need to be on the lookout for alternative data sources. With the natural migration of information and reporting onto the cloud/websites, leveraging a web scraper to pull in data can save time with better extraction of information and consistency, compared to other methods.

Purpose: To gather data from online sources via a web scraper

Step 1: Send a GET request to the website you want to scrape from

Step 2: Parse the HTML code

Step 3: Extract the relevant information from the HTML code

Step 4: Store the information in a dataframe

Step 5: Add a delay between requests to avoid overwhelming the website

Step 6: Export the data to a CSV file

Python Libraries:

- Requests: a library for making HTTP requests.
- Beautiful Soup: a library for getting data out of HTML, XML, and other markup languages
- Pandas: A library for working with datasets
- Time: A library that provides various functions to work with time-related operations

* *Photo Source:* Premium Vector | Website platform related with internet (freepik.com)

Conclusion

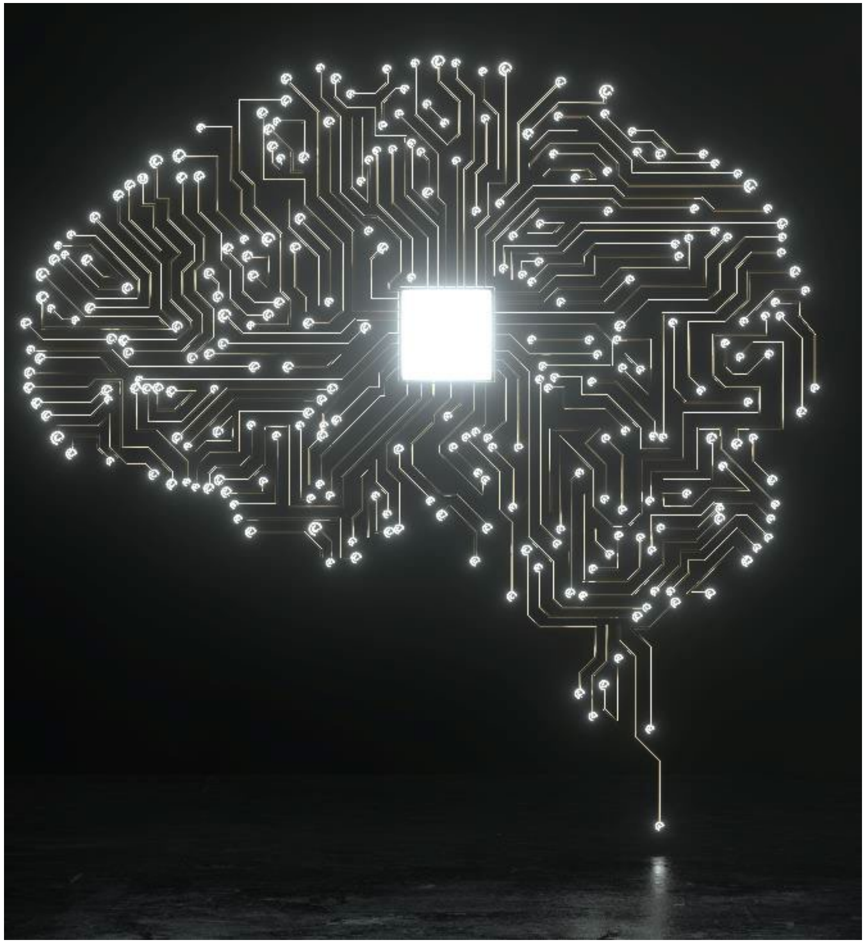
Automating what might be considered small taskings can add up to time savings

- The data collection process can take a significant amount of time for analysts and any improvement can be instrumental in success

Python is a great way to add efficiency in your work

- Python is a versatile programming language
- Use of Python can help evolve your approach to analysis

Not everything created needs to be revolutionary, even small improvements can make a major difference!



Questions

Resources

Resources:

- Automate the Boring Stuff by Al Sweigart
- Python Documentation contents — Python 3.12.3 documentation
- GitHub - openlists/PythonResources: A list of openly available resources for learning and using the Python programming language.
- Integrated Development Environments (IDE):
  - Jupyter Notebooks
  - Replit
  - Visual Studio Code
  - PyDev
  - PyScripter
  - Pycharm