**An Examination of Python and its Intersection with the Role of Programming in Understanding Human Behavior**

William "Alec" Akin

Purdue University Global

IT244 Python Programming

Unit 9 Assignment

Dr. Ed Lavieri

April 9th, 2022

It's fascinating to me that we have so much time and effort as a society to build computers, programming languages, and this advanced share of a wealth of knowledge that covers so many different areas of humanity, yet, we often lack understanding of ourselves. The collective behavior that we've found in computing and programming can give us this understanding. In this essay, I will describe several different topics including, but not limited to, what is "Big Data," how Python fits into "big data," how "big data" and programming allow us to analyze human behavior, how businesses use "big data," and what tools and libraries we can use within Python within these contexts.

Generally speaking, and in my opinion, programming helps us better understand human behavior by providing a way to model highly complex situations and environments – like human behavior. When you have a model of something, you can then start to predict what might happen next -and this concept is beneficial and exceptionally powerful. Python specifically is an excellent choice for this because it's far more accessible than other languages used in data analysis, is easy to read and write, and has a vast number of libraries and packages available to work with large data sets, perform machine learning and artificial intelligence operations, and many of the functions that surround these focus points. A couple of the libraries that I've used to work with data sets - some quite large like the datasets the nonprofit I helped found and server on the board for, some much smaller like a large CSV. When working with Excel and CSVs are generally of any size, I love the flexibility and freedom which the "Pandas" package. The ones that I've used (more so experimented with) that are more AI and ML-focused are Google's "TensorFlow," among some others that focus on Natural Language Processing (TensorFlow, 2022).

So what is "Big Data?" And why is it a term we keep hearing about in school, in the workplace, and on so many vendors' ad copies and their respective commercial websites? The following quote, in my opinion, does an exceptional job of describing big data in a way that is short, concise, but highly descriptive (Taylor, 2022):

"Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size."

As Taylor (2022) describes, we have not only data that is extremely large in size (typically, we think of this as not the only number or rows/columns if it's tabular data, or "documents" if it's a non-relational dataset) in terms of bytes and disk usage, but it's also typically growing at an exponential or linear rate. As Taylor (2022) goes on to say, this has made it nearly impossible, if not entirely impossible, to utilize the tools we'd historically use to analyze it and has brought about this next generation and "explosion" of development I've seen in the number of tools available in this space throughout my career.

So, where does all of this data come from? And why does it grow so rapidly? I can certainly speak to this from the cybersecurity perspective – not only from a focus on the data we use in cybersecurity but also on my experience with protecting "big data" as a product of being on a security team charged with that task for other groups and the overall organization. In cybersecurity, we often talk of "big data" related to a SIEM or "Security Information Event Management." SIEM provides us with a single aggregation point for all data derived from security logs and logs in general from across an enterprise network. This can include every web request to every web server, every time a packet traverses a firewall and the metadata and contextfual data surrounding that sequence, proxy and web content filter logs, ARP tables, and so on, so much more. This is just a brief example, but if you try to conceptualize even just tracking singular packets, their source, destination, ports and connection, and session properties, its data becomes extraordinarily vast, complex, and extensive. Let's take this example to the business side (often referred to within the context of business intelligence – e.g., taking big data and turning it into predictive analytics, which I've had a lot of exposure and experience with).

We can use an example like all of the purchases made by customers at a large retail chain like Walmart. If you think about how many stores Walmart has, how many customers it sees daily, and how many items are scanned at checkout, the number of data points is vast. According to Djordjevic (2021), an average of 265 million people globally visit Walmart stores per week. Even if just a fraction of those people made just one purchase, the amount of data Walmart is collecting is just purchases. By utilizing the flexibility, ease of use, accessibility, and several libraries that Python has to offer, it can help Walmart and similar organizations that need to make sense of their data and derive actionable insights which push successful strategy based on the immense about of data they create on a day by day (or even minute by minute).

So how does a company even collect, parse, and make sense of so much data? This is where big data solutions and technologies come into play. As the business world and the world, in general, started to create more data points through technological use, there came the point where, as I mentioned in the introduction, traditional software and specific methodologies couldn't handle the amount of data that was being thrown at it. According to Wikipedia contributors (2022), there was an emergence of software and vendors that could handle data at a massive scale and produce analytics and reports from that data in the 1980s and 1990s. For example, Teradata started marketing a product called "DBC 1012," which leveraged parallel processing and managed to store and analyze the data set, which was 1TB (Wikipedia contributors, 2022).

This kind of computing and ability wasn't accessible to the general population at that time – not just in terms of cost (because it essentially required super-computing) but also because Programming languages like Python didn't exist yet, or were still in their infancy. As I mentioned in the introduction, Python (and the kinds of computing now available to the average person) have allowed many people to create exciting and meaningful products and solutions that were not achievable until these things became more widespread and public. For example, my experience around this subject has to do with a nonprofit I helped found called the "Police Data Accessibility Project" (PDAP), which aims to collect public police and law enforcement data from across the United States, synthesize it into a standard database scheme, and make it easily searchable and exportable for the benefit of researches, academics, journalists, and the public (Pardes, 2020). We have had many challenges in getting going and not just from an organizational and legal foundation perspective but also a technical perspective. It is only possible for us to have made the progress we've made and continue to make because of Python, its accessibility, its number of libraries and guides, and its accessibility to our volunteers.

So what are some tools that we use either at PDAP or could be used by a company like Walmart to synthesize, make sense of, store, etc., the kinds of big data we encounter in these use-cases? Since I have experience with this, especially regarding PDAP, I'm going to give some examples of some Python packages and frameworks that we use that can provide some examples. The first one is a framework that consists of several different libraries and packages that form an entire "pipeline" from taking data to finalizing it for storage or publication – it's called "Data Factory" (Datahub, 2018). I will dig into this as a focus in a moment, but the second one that we leverage is a library called "Scrapy" (Scrapy developers, 2022). Scrapy is one of the first steps in many big data pipelines in that it takes information from a website (something like a table, file, text, etc.) and pulls it down to storage. At PDAP, we use the Scrapy Python library to gather publicly available police data from nearly 18,000 police agencies across the country to have an up-to-date data set that can be published. There are many steps between Scrapy and the final datasets, but it is crucial for our big data pipeline because it's where our data comes from. The third Python library that I find the most useful (even in my day-to-day work at my day job and personal security research) is called "Pandas." Pandas is an incredibly excellent library that allows me to work with relational/tabular data (like Excel spreadsheets and CSVs) to do advanced analytics, reformatting, searching, synthesizing data from multiple sources, and so much more than I don't take advantage of (The Pandas Development Team, n.d.).

I find the most exciting and helpful out of this list and most applicable to an organization like Walmart, academia, PDAP, and more, is Data Factory from Datahub. I say this because it provides a complete "pipeline" (a common term we use at PDAP to describe the process from data acquisition to presentation to the user) for a development team. According to Datahub (2018) themselves, Data Factory has the following flow from collection to storage:

1. Data is loaded in from a variety of sources and file types
2. Data is normalized, cleaned, and made portable
3. Data is transformed to create a standard structure from various other types of data and content
4. Data is validated and follows your own rules and verification processes
5. Data is stored

This is incredibly helpful for more minor, open-source projects like PDAP and will also reduce developer time and effort significantly by providing an existing framework to add custom code to fit the needs of the company or organization.

We are still working on fully integrating the Data Factor model into our pipeline. At the moment, our data (PDAP's data) is being stored here <https://www.dolthub.com/organizations/pdap> as we leverage the open-source Dolt project (think of it as a mashup between the Git protocol and a SQL database) because it allows us to perform "data bounties" where contributors get a small some for submitting data via a grant from one of our donors (DoltHub, Inc., 2022). Based on our strategic projects plans and development pipeline, Data Factory is going to provide us with a pipeline from source to storage where we can then use our analytics tools like our Splunk instance to give insights and where the users and consumers of our data can use other open-source Python tools actually to perform the analysis and utilize the data for research, policy decisions, and other projects.

In summary, big data is here to stay, and I'd bet that it will only continue to get "bigger." It will be vital for programming languages like Python and others which are similar to continue to allow developers and open source projects to create the libraries, frameworks, and other tools we need to keep big data accessible to the ordinary person but also make it easier for a large organization to integrate big data processes and analytics into their business. From my experience professionally and personally, the best decisions and strategies are founded on good data. We need to make sure that we use the right tools at the right time and that they are sustainable, scalable, and provide actionable insights to their consumers.

**References**

Datahub. (2018). *Data Factory*. Retrieved April 10, 2022, from https://datahub.io/data-factory

Djordjevic, M. (2021, November 19). *30 Incredible Walmart Statistics That Prove It Is Still A Retail Giant*. SaveMyCent. Retrieved April 9, 2022, from https://savemycent.com/walmart-statistics/

DoltHub, Inc. (2022, April 10). *GitHub - dolthub/dolt: Dolt – It's Git for Data*. GitHub. Retrieved April 10, 2022, from https://github.com/dolthub/dolt

Pardes, A. (2020, July 7). *A Plan to Make Police Data Open Source Started on Reddit*. Wired. Retrieved April 10, 2022, from https://www.wired.com/story/police-accountability-data-project-open-source-reddit/

Police Data Accessibility Project. (n.d.). *About*. PDAP.Io. Retrieved April 10, 2022, from https://pdap.io/about.html

Scrapy developers. (2022, April 8). *Scrapy at a glance — Scrapy 2.6.1 documentation*. Scrapy - Read the Docs. Retrieved April 10, 2022, from https://docs.scrapy.org/en/latest/intro/overview.html

Taylor, D. (2022, March 26). *What is Big Data? Introduction, Types, Characteristics, Examples*. Guru99. Retrieved April 9, 2022, from https://www.guru99.com/what-is-big-data.html

TensorFlow. (2022, April 10). *GitHub - tensorflow/tensorflow: An Open Source Machine Learning Framework for Everyone*. GitHub. Retrieved April 10, 2022, from https://github.com/tensorflow/tensorflow

The Pandas Development Team. (n.d.). *pandas - Python Data Analysis Library*. Pandas. Retrieved April 10, 2022, from https://pandas.pydata.org/about/

Wikipedia contributors. (2022, April 1). *Big data*. Wikipedia. Retrieved April 10, 2022, from https://en.wikipedia.org/wiki/Big\_data