**DAMG7245: Big-Data Systems and Intelligence Analytics**

**Assignment 1**

**Group 6 Members  
Pranjal Mahajan (002375449)**

**Srushti Patil (002345025)**

**Ram Putcha (002304724)**

**Comparison of Tools Used for Extraction**

**Task 1: Data extraction**

* Pdf Files
* We have used PyMuPDF to extract data for PDF files in open-source tools.
* Along with testing and researching different open-source tools to extract data from PDF we tested tools like pypdf2 and pdfminer.
* For Enterprise-level tools to extract data from the pdf we have used Azure
* Along with testing and researching different Enterprise tools to extract data from PDF we tested tools like Adobe and Diffbot
* Website URL
* We have used selenium to extract data from websites using open-source tools.
* Along with testing and researching different open-source tools to extract data from PDF we tested tools like Beautiful Soup.
* For enterprise-level to extract data from the website we have used scraping bee.
* Along with testing and researching different Enterprise tools to extract data from PDF we tested tools like Microsoft intelligence.

**Task 2: Comparison of Tools**

* Open-Source vs. Enterprise for data extraction from PDF

|  |  |  |
| --- | --- | --- |
| **Feature** | **Open-Source (PyMuPDF)** | **Enterprise (Azure)** |
| Performance | Fast and efficient for text and image extraction | Uses cloud-based AI for text recognition and processing. |
| Accuracy | High Accuracy | High Accuracy |
| Cost | Free open source | Paid service based on volume of document and time for processing. |
| Scalability | Handles moderate workload | Highly scalable |
| Integration | Requires Custom scripting | Seamless integration |

* Open-Source vs. Enterprise for data extraction from Website

|  |  |  |
| --- | --- | --- |
| **Feature** | **Open-Source (Azure)** | **Enterprise (ScrapingBee)** |
| Performance | Slower due to browser interaction | Faster as it uses headless browser |
| Accuracy | Can interact with javascript heavy sites | Designed for accuracy with built-in javascript |
| Cost | Free but requires own server setup | Paid service with pricing based on request volume |
| Scalability | Hard to scale | Easily scale |
| Integration | Requires manual setup | Provides API endpoints |

* Pros and cons

|  |  |  |
| --- | --- | --- |
| **Category** | **Open source** | **Enterprise** |
| Pros | ✔ Free to use  ✔Can be self-hosted for privacy and security. | ✔ Scalable and efficient for large workloads.  ✔ Less maintenance effort (API-based). |
| cons | ❌ Requires setup, configuration, and ongoing maintenance. | ❌ Paid services, cost scales with usage.  ❌ Less flexibility for custom implementations. |

|  |  |  |
| --- | --- | --- |
| **Category** | **Docling** | **Markitdown** |
| Pros | ✔ Provides version control for better document management.  ✔ Interactive documentation with dynamic elements. | ✔ Simple and lightweight, using standard Markdown.  ✔ Easy to write and edit with minimal setup. |
| cons | ❌ More complex setup compared to plain Markdown. | ❌ Lacks interactive features (e.g., search, live collaboration). |