**Critical Findings about pypdf2:**

In Python, it’s a free and open-source pure-python library capable of splitting, merging, cropping, and transforming the pages of PDF files. It can also add custom data, viewing options, and passwords to PDF files. PyPDF2 can retrieve text and metadata from PDFs as well. The low-level API (based on Pygments) permits composing programs that produce or productively control documents. The high-level API (based on ReportLab) empowers the making of perplexing documents like forms, books, or magazines with only a couple of lines of code.

**PyPDF2 supports:**

* Converting PDF files into images (png or jpeg) or text files;
* Creating new PDF documents from scratch;
* Editing existing PDFs by adding, removing, replacing, or modifying pages;
* Modifying existing PDFs by rotating pages, adding watermarks, changing fonts, etc.;
* Signing documents with digital signatures (certificates must be present);

PyPDF2 has been designed with performance in mind. It uses native C code to handle the most time-consuming tasks (such as parsing) but never sacrifices the simplicity of its interface. The library is also thread-safe, and its memory footprint is not much larger than the one required by Python (around 1MB).

**Benefits:**

* Extracting the useful data from a pdf file.
* Converting PDF to word or another format.
* Merging Multiple PDFs Together
* Modifying the contents of a PDF Document
* Splitting a Large Document into smaller ones

**Real-time usage:**

Having issues reading the hand-written documents. Otherwise working fine.

**Critical Findings about AWS textract:**

Amazon Textract is a machine learning (ML) service that automatically extracts text, handwriting, and data from scanned documents. It goes beyond simple optical character recognition (OCR) to identify, understand, and extract data from forms and tables. Strong AI and ML algorithms are behind them; however, there aren't any open-source models to dive into the specifics.

Firstly, whenever a new or a scanned document is sent into Textract, it creates a list of block objects for all the detected text. For example, say an invoice consists of hundred words today, AWS makes hundred block objects for all the words. These blocks contain information about a detected item, where it's located, and the confidence that Amazon Textract has in the accuracy of the processing.

Usually, most of the documents are made of the following blocks:

* Page
* Lines and words of text
* Form data (Key-value pairs)
* Tables and Cells
* Selection elements

However, the content inside the blocks changes based on the operation we call. For the text detection operation, the blocks return the pages, lines, and words of detected text. If we’re using the document analysis operations the blocks will return the detected pages, key-value pairs, tables, selection elements, and text.

**Files are available for both of the models.**