Prior to starting the scraping, the desired Wikipedia page’s html was examined thoroughly. It was therefore determined that all relevant information is contained on the “Systems” section of the page. Since the relevant data isn’t contained in an element (e.g. a div), but rather simply placed one after the other, in order to scrap the entire “Systems” section a start element (the “Systems” header) and an end element (the “See also” header) were selected using BeautifulSoup. All html between the two elements was scraped. However, the scraped content was still too lengthy for most models to handle. The scrape\_sections() function solves this problem, scraping each hurricane entry separately and then saving it as a distinct entry of the “sections” table. This was achieved by searching for div-table-link-paragraph(s)-div patterns, which is the pattern that each hurricane entry in the “Systems” section follows.

Regarding data extraction, ideally a recent gpt model would be used via API, to combine accuracy, speed and the ability to use function calling. Unfortunately, their APIs aren’t accessible without a paid plan, and running these models locally proved extremely difficult due to the large computational overhead. After testing a few other models, finally the Hugging Face question answering model “distilbert-base-uncased-distilled-squad” was chosen. The model tries to find the desired information as answers to given questions, such as “what is the name of the hurricane?”. It’s fast, can easily run locally and provides fairly good results, perhaps with the exception of the number\_of\_deaths column where it really seems to struggle. Since both the original content and the extracted data are relatively small, the extracted data can easily be reviewed by simply “manually” comparing it to the expected values based on the original content. For a more thorough and automated approach, the data could be assessed by calculating metrics like precision and recall, or by implementing some kind of validation regarding its form and content.