Introduction

AI tools were intended to be used primarily to contribute to technical aspects of the project, such code generation and resolving errors. The planned approach as represented in the guidelines was to rely on AI for coding assistance while ensuring that all analytical decisions, interpretations, and insights derived from the dataset were produced independently. In other words, AI was used as a programming aid rather than a tool for shaping the conceptual or analytical content of the project.

AI performed well for tasks involving data manipulation, debugging, code structuring, and providing clear explanations of unfamiliar functions or methods. It significantly reduced development time by returning useful code snippets along with a detailed explanation. ChatGPT-5 was mainly used for producing code snippets while colabs built in Gemini agent was used for most error handling and small changes to code (the small error handling prompts were not stored).

Some limitations were also observed. AI occasionally proposed code that did not fully align with the dataset's structure, produced HTML selectors that did not match the webpage during web scraping, or introduced unnecessary complexity to simple problems. It was important to constantly remind the model to "keep it simple". The model generally becomes pretty useless after a conversation of more than 25-30 interactions and a new chat had to be made. These issues required validation, adjustment, or complete rewriting. This shows the importance of verifying AI-generated code and maintaining full understanding of every step implemented.

Prompts

- *i want to remove all rows that dont have the value 'european paintings' in the 'department' column to reduce the scope of this project using pandas*
- *how do i check if there is an entire column with just nan values?*
- *Return a list of all empty columns in the dataframe and then drop these columns*
- *Generate a heatmap of the columns so I can better visualise the missing data*
- *ok lets start with adding the met museum url. they all follow the same structure https://www.metmuseum.org/art/collection/search/###### where the hashtag contains there object ID. we need to add this as a column first*
- *lets just return the html text of this site for now so i can see the structure of the site (https://www.metmuseum.org/art/collection/search/435570) utilising beautifulsoup*
- *ok lets start with trying to get the image url: THis is where it is in the website [screenshot of html elements].*
- *ok lets now try to get the on view true or false bit done. remember keep it as simple as possible no need for error handling as of now [screenshot of relevant html elements]*

- *should you not be looking for this in a specific tag or just generally throughout all text on the webpage?*
- *Ok now lets return the overview description [screenshot of relevant html elements]*
- *This is my code for processing a dataset from github and attempting to scrape some information from a website to add to the dataset. Can you make a function that goes through the list of metobjecturl's and gathers the three data pieces from the webpage (image url, on_view, overview text) I want to keep this in a dataframe and then merge it with the other dataset at the end once all of them have been scraped.*
- *i was told by my lecturer it is better principle to use inplace=True when working with dataframes rather than following a df =df.changes format. can you look through my code and point out where i can change my code to utilise the inplace=true rather than how i have it.*
- *I have a list of visualisation I want to portray with the newly merged dataframe. Generate a relevant chart based on the title of what ill ask utilising seaborn and motplotlib:*
    - *Distribution of artwork completion dates*
    - *Top artists by number of works*
    - *Top artist nationalities*
- *Ok now lets move on to looking at our on_view field and how it adds to what we already have. what impact can this have on the data?*
- *Produce a chart with the following titles:*
    - *On-View Status Distribution*
    - *Artwork Dates by On-View Status*
    - *Top 10 Artists with Works Currently On View*
    - *On-View Status by Artist Nationality*
- *Generate a word cloud of the overview_text field so we can better understand what are the most popular words*