**AI Documentation – Project 3**

With the help of ChatGPT for a variety of data preprocessing, classification, and visualization tasks, I explored thematic categorization and visualization of articles using Python in this project. This is an organized log of my conversations with ChatGPT that shows how AI helped me understand and use solutions for typical data analysis problems. In addition to being a learning journal, this conversation-based documentation shows the problems I ran across and how I fixed them with AI help.

**Conversation With Chat GPT:**

**1. Handling Date Conversion Errors:**

I'm trying to convert separate year, month, and day columns into a single datetime column in my pandas DataFrame, but sometimes I get errors due to missing or bad values. How can I handle this gracefully?

ChatGPT: You can use a try-except block to catch any exceptions when creating the datetime column. If an error occurs, assign a fallback date range. Here's how:

python

try:

df['Date'] = pd.to\_datetime(df[['year', 'month', 'day']])

except Exception as e:

print(f"Error creating Date column: {e}")

df['Date'] = pd.date\_range(start='2020-01-01', periods=len(df), freq='D')[:len(df)]

**2. Filtering Out Junk Topics**

Now I want to clean the dataset. Some rows are labeled with topic -1, which I think are junk or unclassified. Should I remove them?

ChatGPT: Yes, filtering those out is a good idea. You can use this line:

python

CopyEdit

df = df[df["Topic"] != -1]

**3. Identifying and Removing Stopwords:**

I have four topic keywords per article in my CSV. I want to remove rows where all of them are meaningless, like stopwords. I uploaded the CSV earlier. Can you help identify a comprehensive list of stopwords?

ChatGPT: Sure! Based on your data, here’s an extended list of stopwords:

python

CopyEdit

stop\_words = {

'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your',

'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her',

'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs',

'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those',

'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had',

'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if',

'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',

'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after',

'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over',

'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where',

'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other',

'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too',

'very', 's', 't', 'can', 'will', 'just', 'al', 'said', 'don', 'should', 'now',

'one', 'get', 'got', 'say', 'says', 'made', 'make', 'thing', 'things', 'like',

'see', 'still', 'also', 'new', 'news', 'use', 'used', 'using', 'every', 'many',

'much', 'back', 'even', 'really', 'another', 'year', 'years'

'i', 'me', 'my', 'myself', 'we', 'our', ..., 'year', 'years'

}

**4. Categorizing Articles into Themes**

Now I want to categorize each article into one of three broad themes based on the topic keywords. Can you help me write a function for that?

ChatGPT: Absolutely! Here’s a function that checks if any of the topic keywords match pre-defined sets:

python

def classify\_topic(row):

keywords = [str(row['topic\_1']).lower(), str(row['topic\_2']).lower(),

str(row['topic\_3']).lower(), str(row['topic\_4']).lower()]

# (your classification logic here)

**5. Customizing Visualization with Plotly**

User: I’m using Plotly Express to make a stacked bar chart. Can I assign specific colors to each theme?

ChatGPT: Yes, use the color\_discrete\_map argument like this:

python

CopyEdit

color\_discrete\_map={

'Security & Conflict': '#EF553B',

'Diplomacy & Politics': '#636EFA',

'Civilian Toll, Crisis & Aid': '#00CC96'

}

User: I also want to tweak the margins so the layout isn’t too tight around the plot. What do you suggest?

ChatGPT: Use the margin argument in fig.update\_layout():

python

margin={'l': 50, 'r': 50, 't': 80, 'b': 50}

User: One last thing, the x-axis labels are too crowded. How can I rotate them and show only every 3rd month?

ChatGPT: Use this:

python

CopyEdit

fig.update\_xaxes(

tickangle=45,

tickvals=theme\_counts['Month'].unique()[::3]

)

**6. Identifying Topic Column Dynamically**

User: I’m trying to count how many articles are associated with each topic in my DataFrame. I’m not sure which column contains the topic labels, though, as the name might vary.

ChatGPT: Here’s a way to check for multiple common names:

python

CopyEdit

topic\_column = None

for possible\_name in ['Topic\_Label', 'Topic\_Name', 'Topic', 'Label', 'topic\_label']:

if possible\_name in df.columns:

topic\_column = possible\_name

break

**7. Counting Articles Per Topic**

Now, how do I count the articles per topic using that column?

ChatGPT: Use value\_counts() and reset the index:

python

topic\_counts = df[topic\_column].value\_counts().reset\_index()

topic\_counts.columns = ["Topic", "Count"]

**8. Filtering for Top 5 Topics**

I want to focus only on the top 5 most frequent topics. How can I filter the DataFrame accordingly?

ChatGPT: First, get the top 5 topics, then filter:

python

top\_5\_topics = topic\_counts["Topic"].head(5)

df\_top5 = df[df['Topic'].isin(top\_5\_topics)].copy()

**9. Reshaping Keywords for Analysis**

I want to analyze all topic keywords (topic\_1 to topic\_4) in one column. How can I do that?

ChatGPT: Use melt() to unpivot the keyword columns:

python

CopyEdit

keyword\_cols = ['topic\_1', 'topic\_2', 'topic\_3', 'topic\_4']

melted\_keywords = df\_top5.melt(

id\_vars=['Topic'],

value\_vars=keyword\_cols,

var\_name='keyword\_rank',

value\_name='keyword'

)

**10. Counting Articles by Topic Label**

User: Lastly, how do I count how many articles belong to each of the labeled topic names?

ChatGPT: Use this:

CopyEdit

topic\_counts = df\_top5['Topic\_Label'].value\_counts().reset\_index()

topic\_counts.columns = ["Topic\_Label", "Count"]

Conclusion

Through my interaction with ChatGPT, I was able to:

* Handle errors in datetime conversion
* Clean and preprocess a topic-labeled dataset
* Identify and apply custom stopwords
* Categorize articles into thematic groups using keyword logic
* Visualize data using Plotly with color mapping, rotated labels, and custom layout
* Dynamically identify column names and reshape data for keyword analysis

My understanding of pandas, Plotly, and general data techniques has improved because of this continuous problem-solving process. As a coding assistant, ChatGPT helped me understand and effectively implement each idea by giving me concise explanations in addition to code snippets.