Homework 4: GenAI Disclosure

Question 2

Instance 1

Asked ChatGPT the following prompt:

How do I save details into a dataframe as a separate .csv file in your S3 bucket with amazon MWAA?

Output from ChatGPT:

✅ Step-by-Step: Save DataFrame to S3 in MWAA

1. Ensure IAM Role Permissions

Your MWAA execution role (typically named like `AmazonMWAA-<env-name>-ExecutionRole`) **must have permission** to write to the S3 bucket.

2. Convert DataFrame to CSV in Memory Using `StringIO`

```
import pandas as pd
from io import StringIO
import boto3
from datetime import datetime

def upload_dataframe_to_s3(df, bucket_name, s3_path):
    # Convert DataFrame to CSV in memory
    csv_buffer = StringIO()
    df.to_csv(csv_buffer, index=False)

    # Upload to S3
    s3_client = boto3.client('s3')
    s3_client.put_object(
        Bucket=bucket_name,
        Key=s3_path,
        Body=csv_buffer.getvalue()
    )
```

I used this code to inform me on how to write the code for question 2. I was confused on why we needed the "from io import StringIO" so I asked ChatGPT to explain this function. I got the following output:

✅ Why do we need `from io import StringIO`?

Because `StringIO` lets us create a **file-like object in memory** — instead of writing a file to disk — which is perfect for use in **cloud environments like Amazon MWAA** where:

- **You should not write to local disk** (the worker nodes are ephemeral and stateless)
- You want to **convert a Pandas DataFrame to CSV** format and upload it directly to **Amazon S3**

Question 3

Question 3 was modeled after the code written in Question 2 in terms of creating a DAG. ChatGPT helped fix our code so we could save the matplotlib png in our s3 bucket. It helped diagnose the errors and led to needing to update the IAM privileges. In addition to this, it was used for general debugging purposes and helped parse lengthy error messages.

Question 4

Question 4 was modeled after the code written in Question 2 and 3. There were issues with reading the data and turning datetime variables into numerical values. We used ChatGPT to assist with this transformation. We initially utilized a sliding window (similar to a convolutional neural network) to have as training data points, but this proved to be an issue, so we asked ChatGPT to change it to a normal linear regression with time variables in numerical form to predict the next 8 hours in 30 minute increments.