Your code looks mostly correct, but there are a few potential issues that might be causing the sheet creation to fail:

1. **API Rate Limits**: Smartsheet has API rate limits. If you exceed these limits, your requests might be throttled or rejected. Check if you are hitting any rate limits.
2. **Permissions**: Ensure that the API key you are using has the necessary permissions to create sheets and add rows in Smartsheet.
3. **Error Handling**: Add error handling to capture any exceptions or errors returned by the Smartsheet API. This can provide more insight into what might be going wrong. For example:

Try:

new\_sheet = ss\_client.Home.create\_sheet(new\_sheet\_spec)

ss\_client.Sheets.add\_rows(new\_sheet.id, combined\_rows)

print(f"Combined sheet created with ID: {new\_sheet.id}")

except smartsheet.exceptions.SmartsheetException as e:

print(f"Error: {e.message}")

1. **Data Validation**: Ensure that the data you’re trying to add to the new sheet is valid and matches the expected format. Sometimes, invalid data can cause the API request to fail.
2. **API Response**: Check the response from the create\_sheet and add\_rows methods to ensure they are successful. You can print the response to see if there are any error messages.
3. **Network Issues**: Ensure that there are no network issues preventing your code from reaching the Smartsheet API.

Here’s an updated version of your code with added error handling:

**CODE**

from awsglue.dynamicframe import DynamicFrame

import sys

import boto3

import pandas as pd

import numpy as np

import json

import smartsheet

from datetime import datetime, timedelta

import math

from io import StringIO, BytesIO

import itertools

from awsglue.utils import getResolvedOptions

from awsglue.context import GlueContext

from pyspark.context import SparkContext

# Initialize Glue context

glueContext = GlueContext(SparkContext.getOrCreate())

args = getResolvedOptions(sys.argv, ['TempDir'])

# Retrieve Smartsheet API key from AWS Secrets Manager

clientsecret = boto3.client('secretsmanager')

secret\_name = 'prod/API/Smartsheets'

response = clientsecret.get\_secret\_value(SecretId=secret\_name)

secret = json.loads(response['SecretString'])

API\_KEY = secret['API\_KEY']

# Initialize Smartsheet client

ss\_client = smartsheet.Smartsheet(API\_KEY)

# Retrieve data from Smartsheets

sheet1 = ss\_client.Sheets.get\_sheet("6317117771042692")

sheet2 = ss\_client.Sheets.get\_sheet("2612031110729604")

# Combine rows from both sheets

combined\_rows = []

for row in sheet1.rows:

combined\_rows.append(row)

for row in sheet2.rows:

combined\_rows.append(row)

# Create a new sheet with the combined data

new\_sheet\_spec = smartsheet.models.Sheet({

'name': 'RCCA\_sheet\_data',

'columns': sheet1.columns # Assuming both sheets have the same columns

})

try:

# Create the new sheet

new\_sheet = ss\_client.Home.create\_sheet(new\_sheet\_spec)

# Add the combined rows to the new sheet

ss\_client.Sheets.add\_rows(new\_sheet.id, combined\_rows)

print(f"Combined sheet created with ID: {new\_sheet.id}")

except smartsheet.exceptions.SmartsheetException as e:

print(f"Error: {e.message}")