Q&A for the Group Project

Problem 1

1.1 How can I change my prompt such that my output can be formatted in bullet points?

The original output of the prompt (completion) is in a text format. You can improve the format by using examples (one-shot learning or few-shot learning). However, the prompt can only do so much. If you would like to improve your output by adding empty spaces between paragraphs or making the keywords bold, you will need to edit the output. This is NOT required in this class, and I suggest you focus on other feedback that could be addressed using prompt engineering techniques.

[This is not required at all for this class] For curious minds, this is one example to improve the output formatting by changing the index.js:

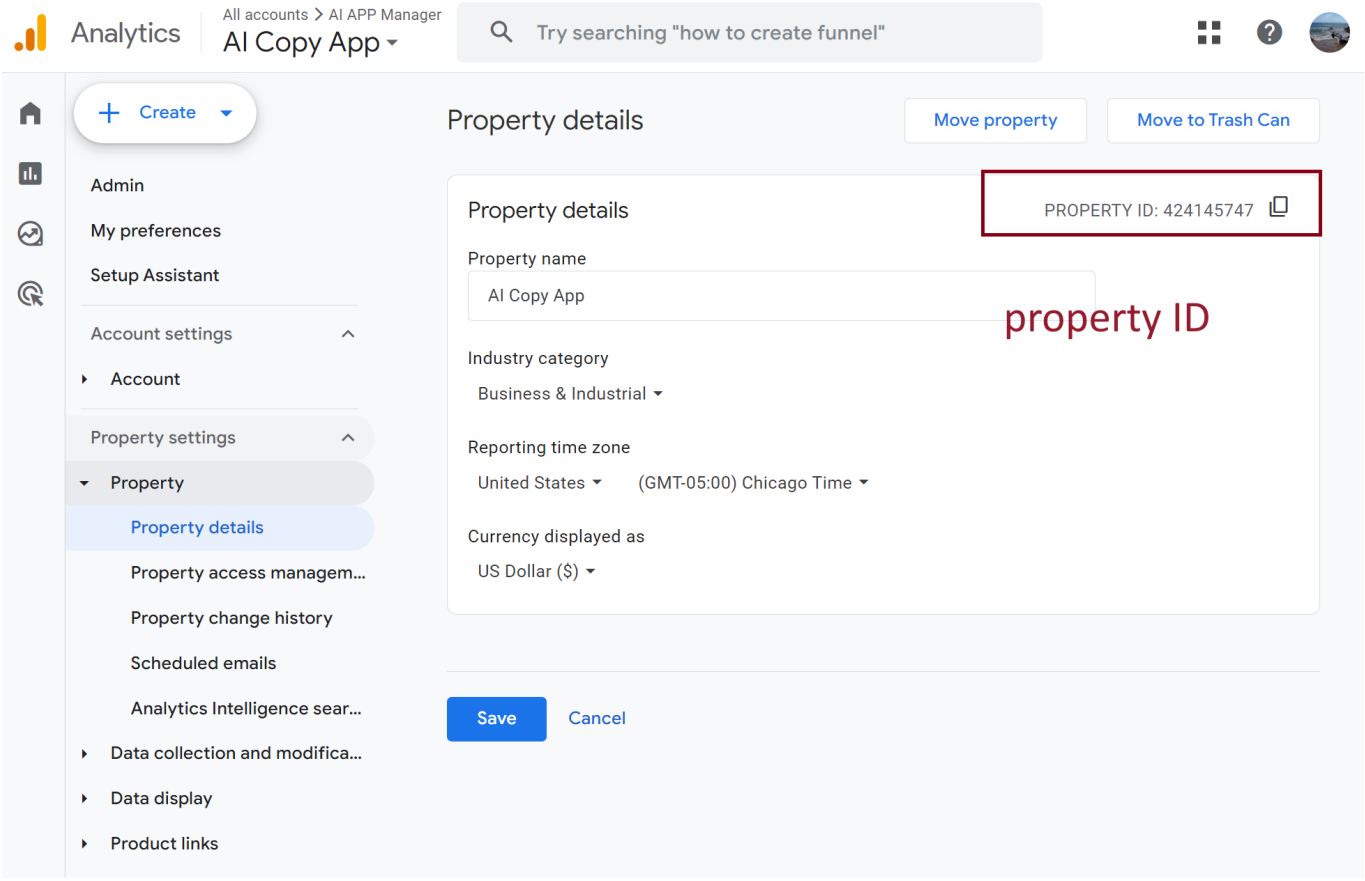
```
 //saving the completion into my_response
  var my_response=` ${data.reply.choices[0].text}`
// Split the text into paragraphs
  var paragraphs = my_response.split("\n");
// Join the paragraphs with <br> tags between them. This tag will create
empty lines between paragraphs
  var htmlContent = paragraphs.join("<br>");
// Replace occurrences of "keyword" with the same text wrapped in <strong>
tags. The <strong> tag will format the keywords bold.
  var htmlContent = htmlContent.replace("keyword", function(match) {
    return "<strong>" + match + "</strong>";
    });
// instead of using insertAdjacentTEXT, we use insertAdjacentHTML to
insert the output.
 document.getElementById('ad-output').insertAdjacentHTML('beforeend',
htmlContent)
 document.getElementById('ad-input').style.display = 'none'
 document.getElementById('ad-output').style.display = 'block'
```

Problem 2

2.1 How can we find the property ID?

Visit your property in Google Analytics, choose your property using the property picker on the top left of the page. Access the admin page by clicking the gear button at the bottom left. Go to Property settings ->

property details. You can copy your Property ID on this page.



2.2 How do I adapt the Jupyter notebook to get data for my own app?

Please refer to the Jupyter notebook for Week 5 (Week_5_AB_Test.ipynb). The Jupyter notebook is available on GitHub Repo (https://github.com/tantantan12/itom6219).

When you call the function `sample_run_report_conversion(property_id="424145747")` to retrieve data, you can change the property_id to your own property_id.

> The code for Week_5_AB_Test.ipynb requires the API file apt-port-251804-905e08b9e9e3.json. This file is downloadable on Canvas Module for Week 5.

2.3 What if I only have "submit" data but not "again" data?

For the group project, it is okay to only have submit data but not again data. No points will be deducted as long as you have submit data.

2.4 What dimensions and filters should I use to retrieve data from Google Analytics API?

You will need to use the dimension of date, source, medium, and eventName. Your filter can be based on eventName. Refer to Section 2.1 of Week_5_AB_Test.ipynb for more details. Make sure to adjust the filter to make the report work for your purposes.

Problem 3

3.1 Where do I find the data?

All the data collected for Problem 3 can be found in the ITOM6219 repo: https://github.com/tantantan12/itom6219/tree/main/Group%20Project/tweets_by_account

These csv files included 100 tweets created by each of the users of your interest.

Each student can choose to use one or more than one data files even if the data was not requested by you.

3.2 What if I want to combine multiple tweet files?

You can surely combine these csv files to work on a richer dataset. The code snippet below combined the data from expedia and tripadvisor.

```
import pandas as pd
# TF-IDF Vectorization
df1=pd.read_csv("expedia.csv")
df2=pd.read_csv("tripadvisor.csv")
frames = [df1, df2]
df = pd.concat(frames)
df
```

3.3 Which Jupyter notebook should I refer to?

Please refer to Week 3 - Getting Data from Twitter.ipynb. This notebook can be found on this here.

In this notebook, I used Section 1 to retrieve this csv file. You can start from Section 2. Specifically, find the following code snippet and change the csv file name to the file of your interest. Make sure you download the file from GitHub and upload it to Google Colab before importing it.

```
import pandas as pd
df=pd.read_csv("grammarly_tweets.csv") # change this file name to your
file.
docs=df['text']
tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,
stop_words='english')
tfidf = tfidf_vectorizer.fit_transform(docs)
tfidf_df = pd.DataFrame(tfidf.toarray(),
columns=tfidf_vectorizer.get_feature_names_out())
tfidf_df
```

3.4 Which metric should I use to measure engagement?

You can use retweet_count, reply_count, like_count, quote_count, bookmark_count, or impression_count. To gaurantee sufficient variation, please take a look at these metrics and pick one that has enough variation. That is, please do not pick a metric whose value is mostly zero.

Feel free to pick two metrics or even more. The chosen metrics will be your outcome (y). You may want to log-transform the outcome before running the predictive analysis. The predictors (x) of your predictive model should be the topics you identified.

Problem 4

4.1 Where can I find the data for Problem 4?

The data for Problem 4 can be found here.

All these queries are saved in json format. You can use any dataset of your choice, even if you did not request the dataset. However, pick the dataset mindfully as you are looking for influencers for your application.

If you need additional data, email janetan@smu.edu. Datasets will be ready within 24 hours.

4.2 How to import json files? Which Jupyter notebook should I refer to?

This question tests your ability to construct a mention network based on tweets in order to identify influencers. The relevant notebook is "Week 4 - Social Network Analysis.ipynb". In this notebook, I collected tweet data based on queries in Section 1.1. You can get started with Section 1.2. Make sure that you change 'data/search_tweets.json' to the file you uploaded.

```
# Opening JSON file
f = open('data/search_tweets.json')
# returns JSON object as a dictionary
data = json.load(f)
```

4.3 How to combine multiple json files so that you can work on more tweets?

You do not have to use multiple json files because the tweets resulting from different queries are unlikely to include the same set of users. However, you can give it a try if you did not have a network that is sufficiently big. To combine more than one json files, you can follow the example below. In this example, I combined data from 'group_2_family_trip.json and 'next_holiday.json'.

```
import json
f = open('group_2_family_trip.json')

# returns JSON object as a dictionary
data1 = json.load(f)

f = open('next_holiday.json')

# returns JSON object as a dictionary
data2 = json.load(f)

data={}
data['includes']={}
data['data']=data1['data']+data2['data']
data['includes']['users']=data1['includes']['users']+data2['includes']
['users']
```

4.4 Where is my edge list?

Following the Jupyter notebook of Week 4, you can use the following code to generate the edgelist.

```
edge=[] # This is an empty list.
i=0
for tweet in filtered_tweets:
    source=tweet['username']
    i=i+1
    print("This is the {}th user!".format(i))
    j=1
    for mention in tweet['entities']['mentions']:
            pair={"source":source,"target":mention["username"]}
            edge.append(pair)
            print("This is the {}th mentioned user!".format(j))
            j=j+1
edge_df=pd.DataFrame(edge)
```

4.5 My network graph does not look pretty. What are different ways of displaying the network?

Instead of using `pos = nx.circular_layout(G)`, youo can use `pos = nx.kamada_kawai_layout(G)` to better display the network.

If your network is too large, you can reduce the size of your network by generating a subgraph.

You can also choose to turn off the label display by allowing `with_labels=False`.

4.6 I encountered an error while calculating eigenvector centrality. What can I do?

If you encountered this error `PowerIterationFailedConvergence: (PowerIterationFailedConvergence(...), 'power iteration failed to converge within 100 iterations')`, you can reduce the tolerence level of the algorithm by adding `tol=1.0e-3`to the code as below. The default value is 1.0e-6. You can increase it further if `tol=1.0e-3` does not work.

```
nx.eigenvector_centrality(G, tol=1.0e-3)
```

Alternatively, you can ignore eigenvector and focus on the other centrality measures.

4.7 Do I need to use external attributes for Problem 4?

The short answer is **NO**. In the class practice, we included external attributes (users_info.csv). **However**, for this assignment, we only need to calculate centrality measures to filter Twitter users before we manually examine our top choices by visiting their Twitter profile pages.

4.8 How to find the profile page of a Twitter user given the username?

The Twitter profile page for "Ellahorantommo" is https://twitter.com/ellahorantommo.

5 Problem 5

5.1 Do we need to run the AB test using Netlify?

NO. For problem 5, we only need to design the AB test and show the alternative version in a branch of our GitHub repo. You do not need to run it and you do not need to collect data for it.