1) Progress made thus far

- Initial setup
- Loading the data
- Starting to learn libraries
- determined kaggle data source

2) Remaining tasks

- Fix startup issues
- Text preprocessing/wrangling data using NLTK
- Feature extraction
- Train-Test Split
- Brainstorm/design sentiment analyzer model (initially Naive Bayes Classifier)
- Implement Naive Bayes Classifier
- Test Naive Bayes Classifier for performance on a set of testing data
- If this doesn't prove effective based on our evaluation, we will convert text data into TF-IDF vectors and then apply Support Vector Machines
- Compare SVM to Naive Bayes Classifier on the same set of testing data

3)Any challenges/issues being faced

- Learning the libraries (nltk, pytorch, scikit-learn)
- Importation issues (should be fixed relatively quickly)
- More will likely arise as we finish remaining tasks but we will tackle/overcome each

```
# https://www.kaggle.com/code/kritanjalijain/amazon-reviews-starter-nlp/notebook

"""
polarity - 1 for negative and 2 for positive
title - review heading
text - review body
"""

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import tarfile # this is to extract the data from that .tgz file

# get all of the data out of that .tgz
amazon_reviews = tarfile.open('/kaggle/input/amazon-reviews/amazon_review_polarity_csv.tgz')
amazon_reviews.extractall('data')
amazon_reviews.close()

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

import os
for dirname, _, filenames in os.walk('./'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# check out what the data looks like before you get started
# look at the training data set
train_df = pd.read_csv('./data/amazon_review_polarity_csv/train.csv', header=None)
print(train_df.head())

# look at the test data set
test_df = pd.read_csv('./data/amazon_review_polarity_csv/test.csv', header=None)
print(test_df.head())

import nltk
tokens = [nltk.word_tokenize(polarity) for polarity in train_df['polarity']]
```

```
  File "<ipython-input-5-ef1f3d46d7bd>", line 32
    tokens = nltk.word_tokenize() for polarity in train_df['polarity']
                                  ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW