Project Proposal:
Sentiment Analysis

Sample Project Proposal

Topic

Natural Language Processing (NLP)

Description

A sentiment analysis API uses natural language processing (NLP) tasks to not only identify aspects of the products from the Amazon reviews but also enable brands to look beyond star ratings.

We will assess these tools to generate insightful customer information that can be harnessed for product betterment. For this project, we will be analyzing Toilet Paper brand reviews that has a lot of comments and reviews on Amazon

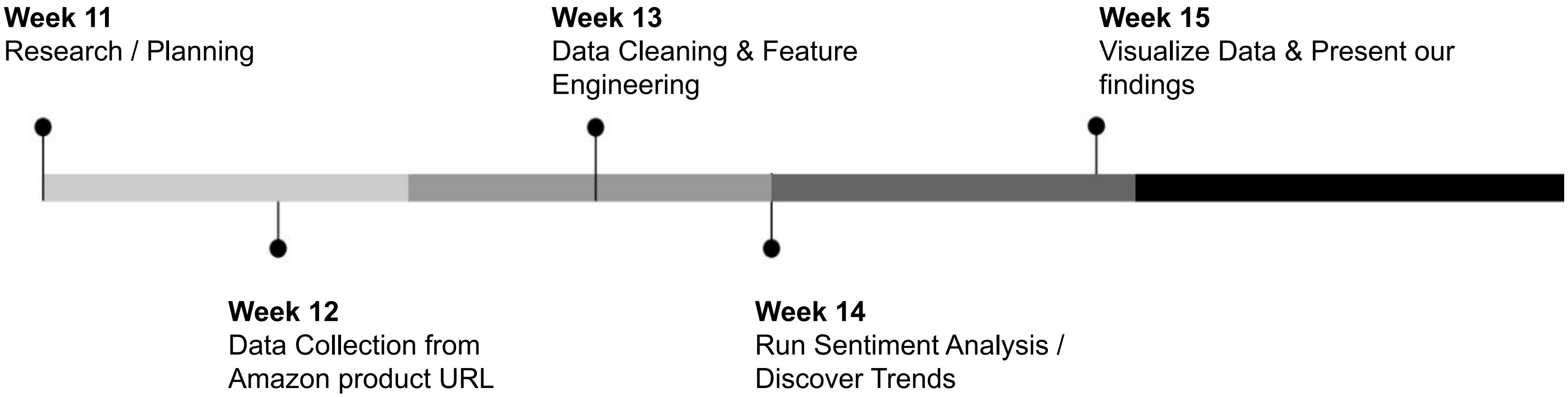
Expected Duration

6 Weeks

Team Member

Juni Heo, Jun Lee, Yeonseo Lee

- Timeline -



Week 11
Research / Planning

Week 12
Data Collection from Amazon product URL

Week 13
Data Cleaning & Feature Engineering

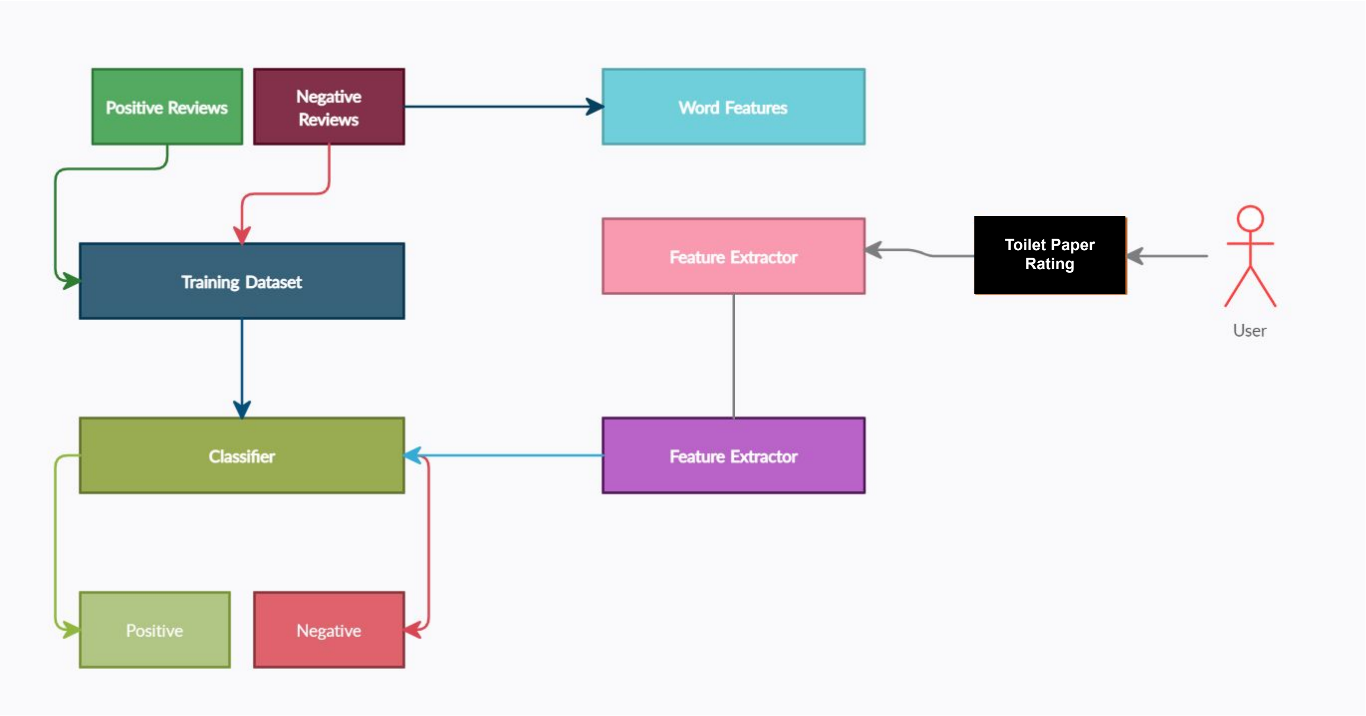
Week 14
Run Sentiment Analysis / Discover Trends

Week 15
Visualize Data & Present our findings

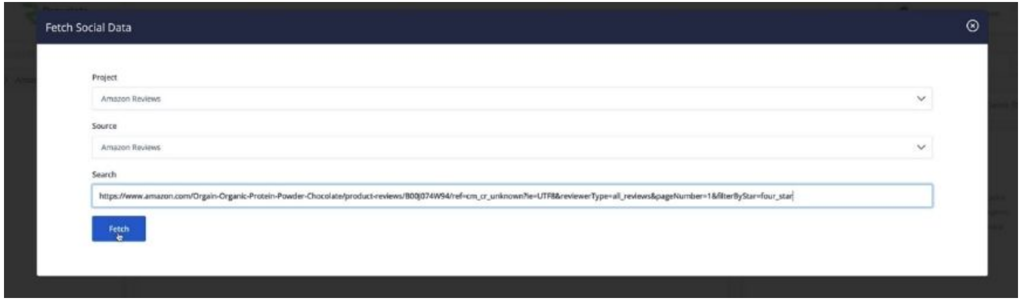
Methodology

1. Sampling from imbalanced datasets
2. Enquiring about the sentiment value of the reviews with the dictionary-based sentiment analysis tools, which are part of NLTK, a natural language processing toolkit, used in Python
3. Evaluate algorithm (Data evaluation with scikit-learn in Python)
4. Analyzing the reviews with a state-of-the-art deep learning technique, namely with the DistilBERT model
    - Pytorch & transformers packages.
5. Evaluate the model and create descriptive statistics
6. Visualize findings about preferable and non-preferable words related to Toilet Paper products using Altair

Project overview



Data Visualization

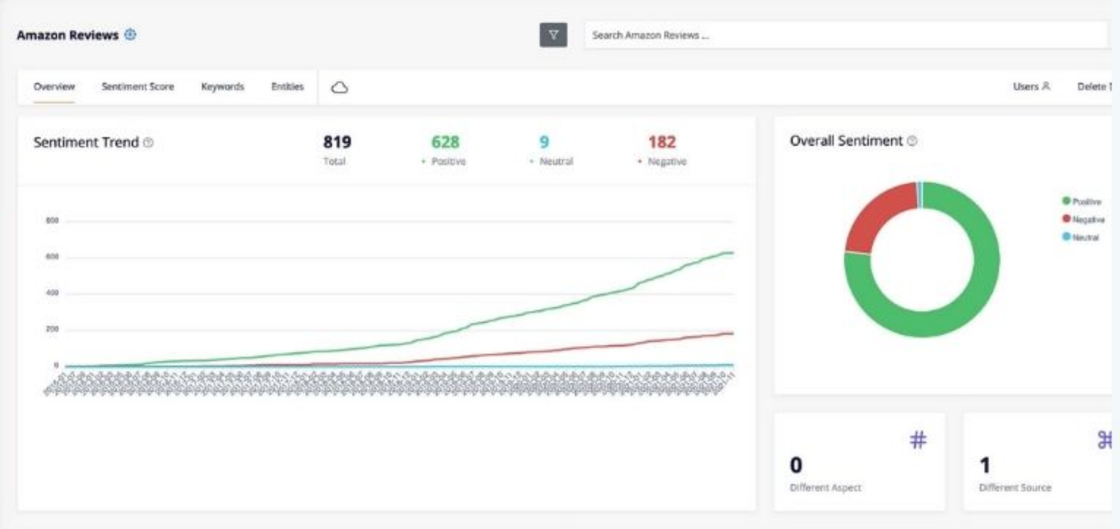


Fetch Social Data

Project: Amazon Reviews

Source: Amazon Reviews

Search: https://www.amazon.com/Orgain-Organic-Protein-Powder-Chocolate/product-reviews/B00J074W94/ref=cm_cr_unknown?ie=UTF8&reviewerType=all_reviews&pageNumber=1&filterByStar=four_star



Amazon Reviews

Sentiment Trend

819 Total | 628 Positive | 9 Neutral | 182 Negative

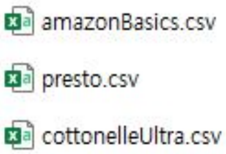
Overall Sentiment

0 Different Aspect

1 Different Source

Data Retrieve

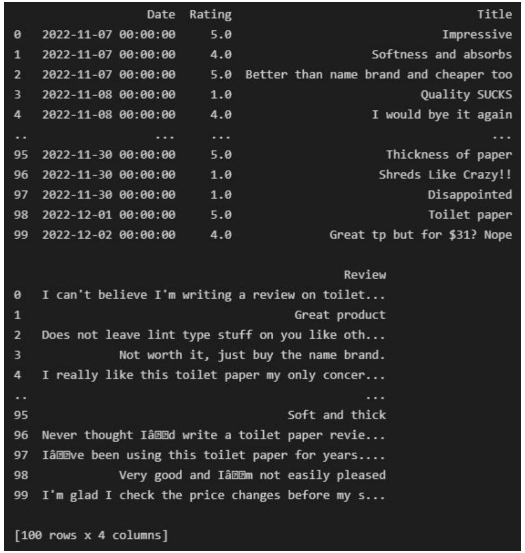
1. Review data retrieved to CSV from Amazon products (100 reviews from 4 different brands)
2. Column Extraction



amazonBasics.csv

presto.csv

cottonelleUltra.csv



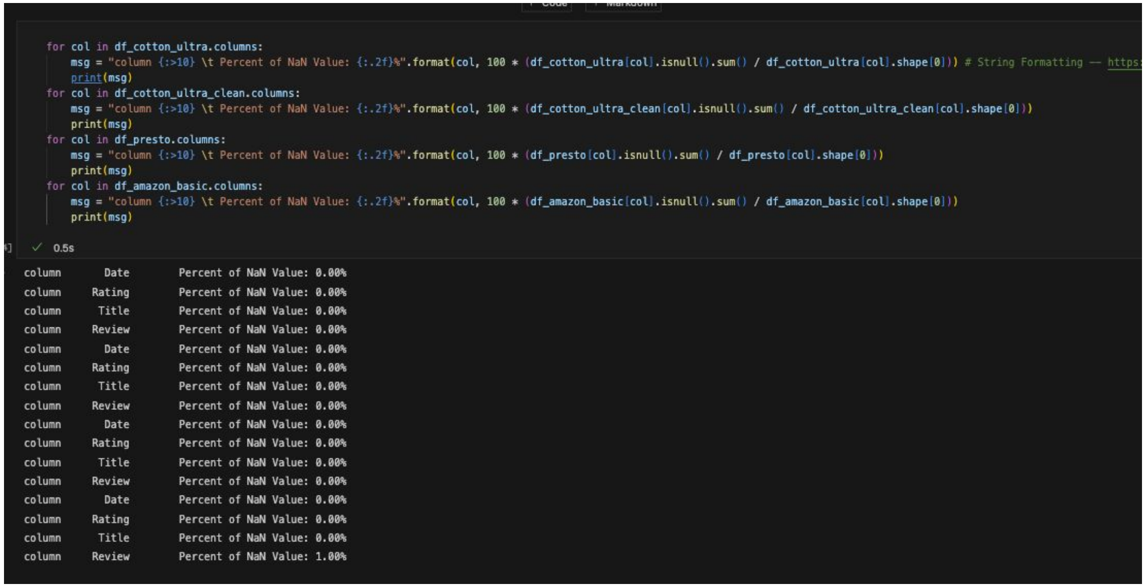
```
                   Date  Rating                             Title
0   2022-11-07 00:00:00     5.0                        Impressive
1   2022-11-07 00:00:00     4.0                Softness and absorbs
2   2022-11-07 00:00:00     5.0  Better than name brand and cheaper too
3   2022-11-08 00:00:00     1.0                     Quality SUCKS
4   2022-11-08 00:00:00     4.0                I would bye it again
..                  ...     ...                               ...
95  2022-11-30 00:00:00     5.0                Thickness of paper
96  2022-11-30 00:00:00     1.0                Shreds Like Crazy!!
97  2022-11-30 00:00:00     1.0                      Disappointed
98  2022-12-01 00:00:00     5.0                      Toilet paper
99  2022-12-02 00:00:00     4.0            Great tp but for $31? Nope

                                               Review
0   I can't believe I'm writing a review on toilet...
1                                       Great product
2   Does not leave lint type stuff on you like oth...
3              Not worth it, just buy the name brand.
4   I really like this toilet paper my only concer...
..                                                ...
95                                     Soft and thick
96  Never thought Iâ▯▯d write a toilet paper revie...
97  Iâ▯▯ve been using this toilet paper for years....
98             Very good and Iâ▯▯m not easily pleased
99  I'm glad I check the price changes before my s...

[100 rows x 4 columns]
```

Feature Engineering

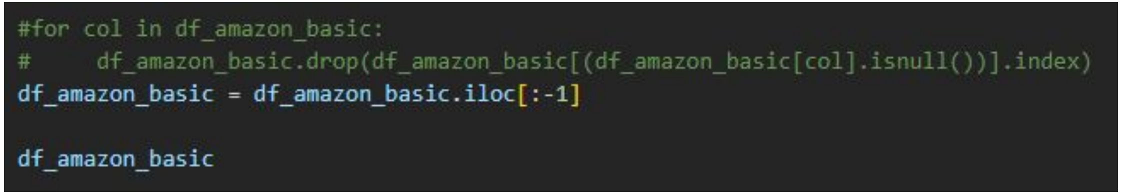
1. Null Data Check
2. Drop missing data



```
for col in df_cotton_ultra.columns:
    msg = "column {:>10} \t Percent of NaN Value: {:.2f}%".format(col, 100 * (df_cotton_ultra[col].isnull().sum() / df_cotton_ultra[col].shape[0])) # String Formatting — https:
    print(msg)
for col in df_cotton_ultra_clean.columns:
    msg = "column {:>10} \t Percent of NaN Value: {:.2f}%".format(col, 100 * (df_cotton_ultra_clean[col].isnull().sum() / df_cotton_ultra_clean[col].shape[0]))
    print(msg)
for col in df_presto.columns:
    msg = "column {:>10} \t Percent of NaN Value: {:.2f}%".format(col, 100 * (df_presto[col].isnull().sum() / df_presto[col].shape[0]))
    print(msg)
for col in df_amazon_basic.columns:
    msg = "column {:>10} \t Percent of NaN Value: {:.2f}%".format(col, 100 * (df_amazon_basic[col].isnull().sum() / df_amazon_basic[col].shape[0]))
    print(msg)
```

✓ 0.5s

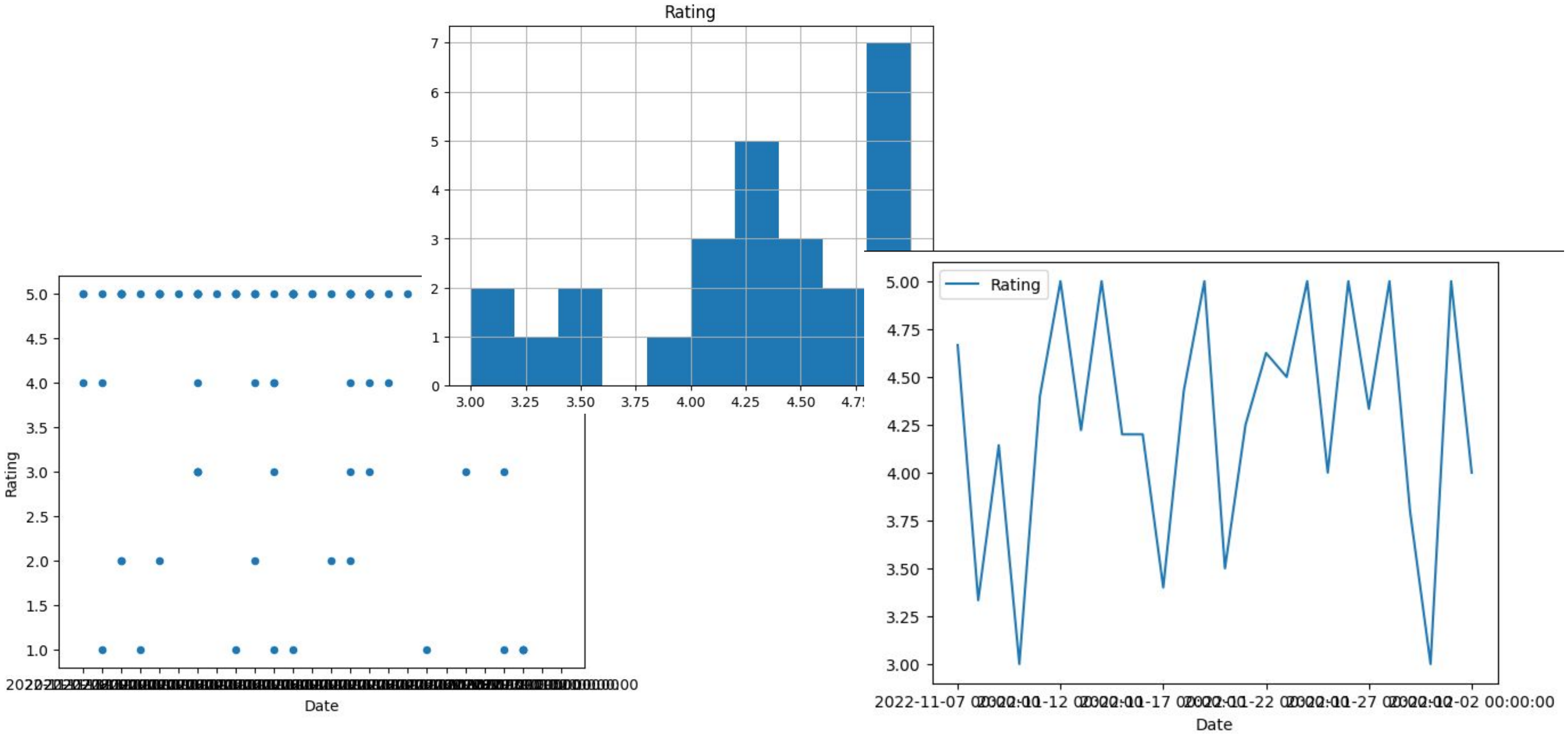
```
column       Date       Percent of NaN Value: 0.00%
column     Rating       Percent of NaN Value: 0.00%
column      Title       Percent of NaN Value: 0.00%
column     Review       Percent of NaN Value: 0.00%
column       Date       Percent of NaN Value: 0.00%
column     Rating       Percent of NaN Value: 0.00%
column      Title       Percent of NaN Value: 0.00%
column     Review       Percent of NaN Value: 0.00%
column       Date       Percent of NaN Value: 0.00%
column     Rating       Percent of NaN Value: 0.00%
column      Title       Percent of NaN Value: 0.00%
column     Review       Percent of NaN Value: 0.00%
column       Date       Percent of NaN Value: 0.00%
column     Rating       Percent of NaN Value: 0.00%
column      Title       Percent of NaN Value: 0.00%
column     Review       Percent of NaN Value: 1.00%
```



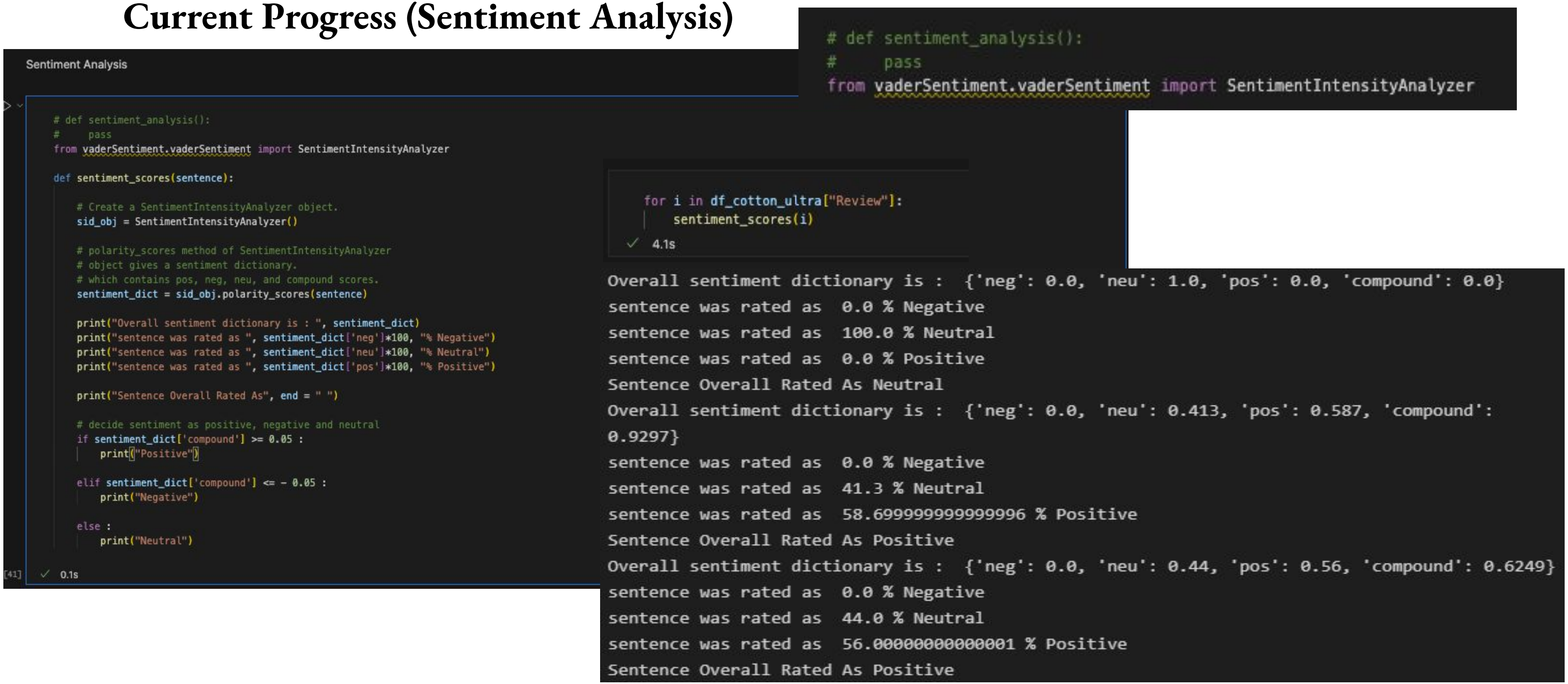
```
#for col in df_amazon_basic:
#     df_amazon_basic.drop(df_amazon_basic[(df_amazon_basic[col].isnull())].index)
df_amazon_basic = df_amazon_basic.iloc[:-1]

df_amazon_basic
```

Data Visualization (Rating by Date)



Current Progress (Sentiment Analysis)



Sentiment Analysis

```
# def sentiment_analysis():
#     pass
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

def sentiment_scores(sentence):

    # Create a SentimentIntensityAnalyzer object.
    sid_obj = SentimentIntensityAnalyzer()

    # polarity_scores method of SentimentIntensityAnalyzer
    # object gives a sentiment dictionary.
    # which contains pos, neg, neu, and compound scores.
    sentiment_dict = sid_obj.polarity_scores(sentence)

    print("Overall sentiment dictionary is : ", sentiment_dict)
    print("sentence was rated as ", sentiment_dict['neg']*100, "% Negative")
    print("sentence was rated as ", sentiment_dict['neu']*100, "% Neutral")
    print("sentence was rated as ", sentiment_dict['pos']*100, "% Positive")

    print("Sentence Overall Rated As", end = " ")

    # decide sentiment as positive, negative and neutral
    if sentiment_dict['compound'] >= 0.05 :
        print("Positive")

    elif sentiment_dict['compound'] <= - 0.05 :
        print("Negative")

    else :
        print("Neutral")
```

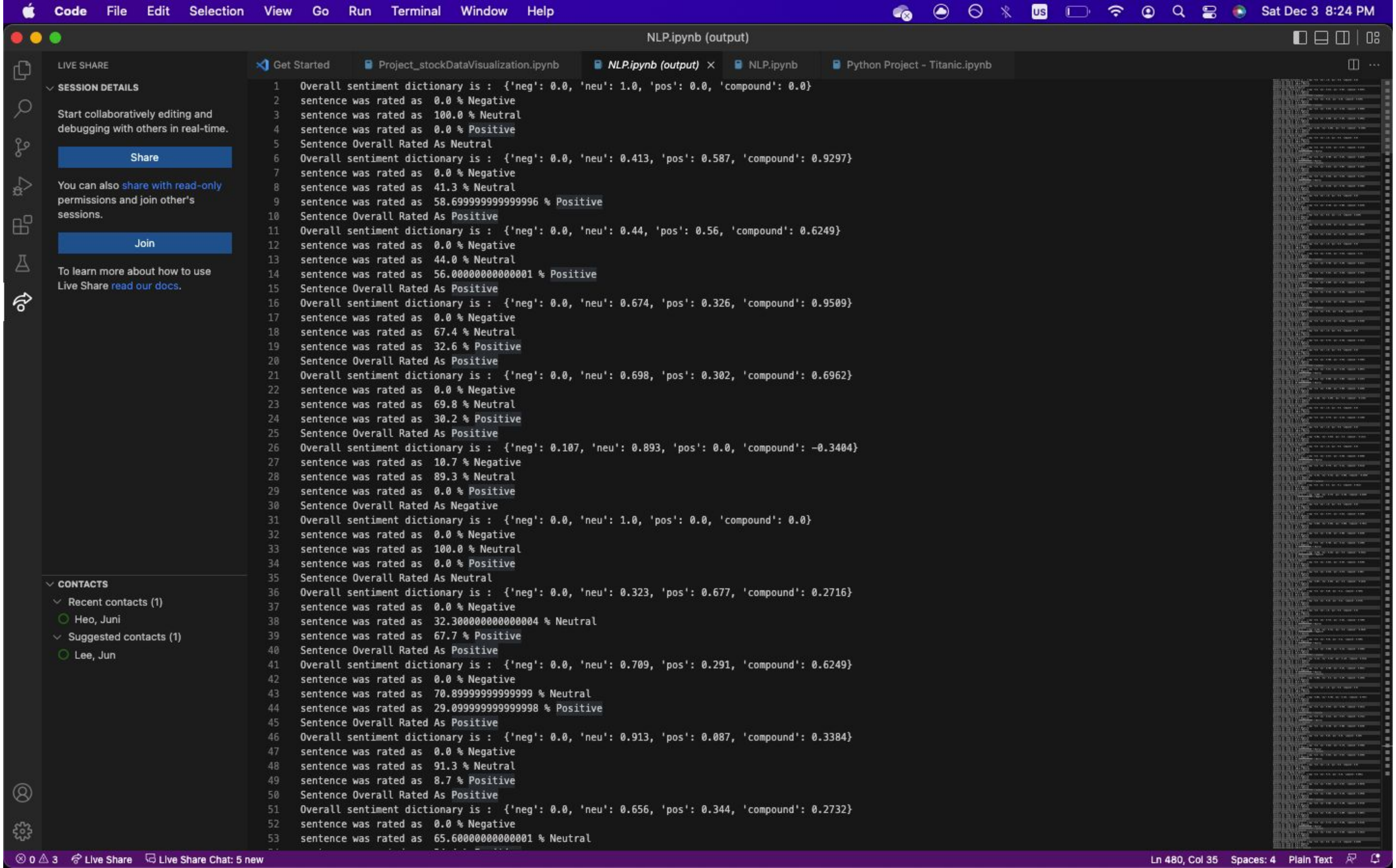
[41] ✓ 0.1s

```
# def sentiment_analysis():
#     pass
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

```
for i in df_cotton_ultra["Review"]:
    sentiment_scores(i)
```

✓ 4.1s

```
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
sentence was rated as  0.0 % Negative
sentence was rated as  100.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Neutral
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.413, 'pos': 0.587, 'compound':
0.9297}
sentence was rated as  0.0 % Negative
sentence was rated as  41.3 % Neutral
sentence was rated as  58.699999999999996 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.44, 'pos': 0.56, 'compound': 0.6249}
sentence was rated as  0.0 % Negative
sentence was rated as  44.0 % Neutral
sentence was rated as  56.00000000000001 % Positive
Sentence Overall Rated As Positive
```



Start collaboratively editing and debugging with others in real-time.

You can also share with read-only permissions and join other's sessions.

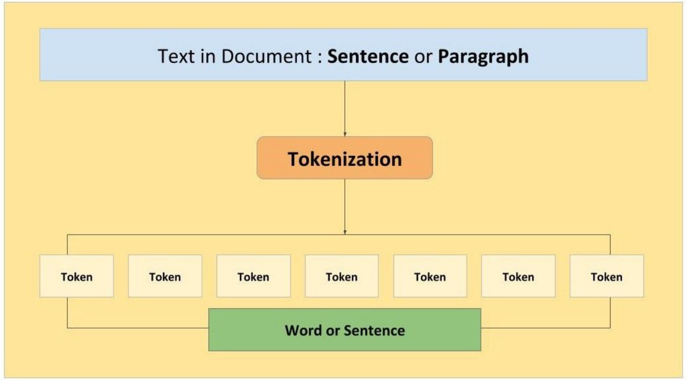
To learn more about how to use Live Share read our docs.

```
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
sentence was rated as  0.0 % Negative
sentence was rated as  100.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Neutral
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.413, 'pos': 0.587, 'compound': 0.9297}
sentence was rated as  0.0 % Negative
sentence was rated as  41.3 % Neutral
sentence was rated as  58.699999999999996 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.44, 'pos': 0.56, 'compound': 0.6249}
sentence was rated as  0.0 % Negative
sentence was rated as  44.0 % Neutral
sentence was rated as  56.00000000000001 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.674, 'pos': 0.326, 'compound': 0.9509}
sentence was rated as  0.0 % Negative
sentence was rated as  67.4 % Neutral
sentence was rated as  32.6 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.698, 'pos': 0.302, 'compound': 0.6962}
sentence was rated as  0.0 % Negative
sentence was rated as  69.8 % Neutral
sentence was rated as  30.2 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.107, 'neu': 0.893, 'pos': 0.0, 'compound': -0.3404}
sentence was rated as  10.7 % Negative
sentence was rated as  89.3 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Negative
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}
sentence was rated as  0.0 % Negative
sentence was rated as  100.0 % Neutral
sentence was rated as  0.0 % Positive
Sentence Overall Rated As Neutral
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.323, 'pos': 0.677, 'compound': 0.2716}
sentence was rated as  0.0 % Negative
sentence was rated as  32.300000000000004 % Neutral
sentence was rated as  67.7 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.709, 'pos': 0.291, 'compound': 0.6249}
sentence was rated as  0.0 % Negative
sentence was rated as  70.89999999999999 % Neutral
sentence was rated as  29.099999999999998 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.913, 'pos': 0.087, 'compound': 0.3384}
sentence was rated as  0.0 % Negative
sentence was rated as  91.3 % Neutral
sentence was rated as  8.7 % Positive
Sentence Overall Rated As Positive
Overall sentiment dictionary is :  {'neg': 0.0, 'neu': 0.656, 'pos': 0.344, 'compound': 0.2732}
sentence was rated as  0.0 % Negative
sentence was rated as  65.60000000000001 % Neutral
```

Next Steps

Tokenizing Amazon review text with NLTK in python

- **Split** and **filter** text data in preparation for analysis
- Analyze **word frequency**
- Find **concordance** and **collocations** using different methods
- Perform quick sentiment analysis with NLTK's built-in classifier
- Define features for **custom classification**
- Use and compare **classifiers** for sentiment analysis with NLTK



Taking a closer look at Natural Language Processing phase (out next steps)

1. Syntax Analysis (Parsing)
    a. Process of arranging words and checking grammar (*removing unnecessary words*)
    b. ex) New York goes to John. This sentence New York goes to John is rejected by the Syntactic Analyzer as it makes no sense.
2. Semantic Analysis
    a. Examining the meaning of context through analyzing *tokenized words/phrases*
    b. ex) “The guava ate an apple.” The line is syntactically valid, yet it is illogical because guavas cannot eat.
3. Discourse Integration
    a. Assessing the “feeling of context” = looking at preceding sentences to accurately find meanings of contect
    b. ex) “Billy Bought it” - ‘it’ is ambiguous and the meaning isn’t provided ⇒ REJECT
4. Pragmatic Analysis
    a. Applying a set of rules to interpret the result
    b. ex) “Switch on the TV’ in a sentence = request to turn on the TV