POLYTECHNIC UNIVERSITY OF THE PHILIPPINES

EMERGING TECHNOLOGIES IN COMPUTER ENGINEERING

"Sentiment Analysis with Python"

In Partial Fulfillment of the Requirements for the

Bachelor of Science in Computer Engineering

By:

HICANA, JHANNA MAE G.

To:

Prof. Simon Salvador Tidon

2022

Table of Contents

Figure List

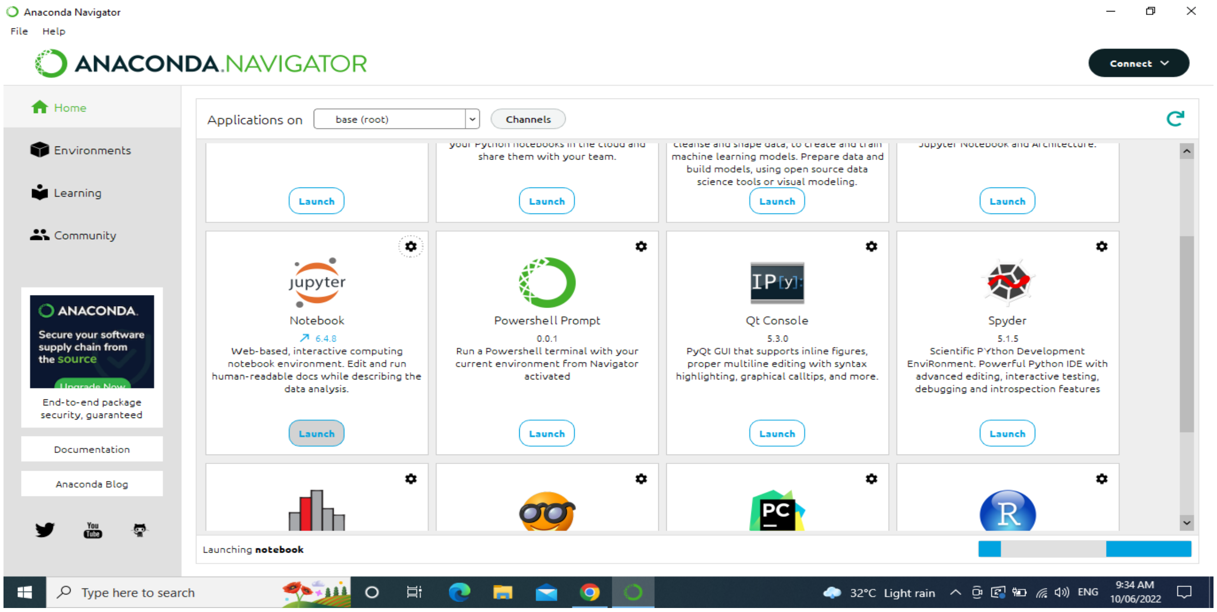
Introduction

One of the most popular projects in the industry. Every customer facing industry (retail, telecom, finance, etc.) is interested in identifying their customers' sentiment, whether they think positive or negative about them. Python sentiment analysis is a methodology for analyzing a piece of text to discover the sentiment hidden within it. It accomplishes this by combining machine learning and natural language processing (NLP). Sentiment analysis allows you to examine the feelings expressed in a piece of text.

Initial Setup

Figure 1. Launch jupyter from Anaconda Navigator

From Anaconda Navigator, launch Jupyter Notebook, which opens the root folder in your machines default browser.



your Python notebooks in the cloud and share them with your team.

cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

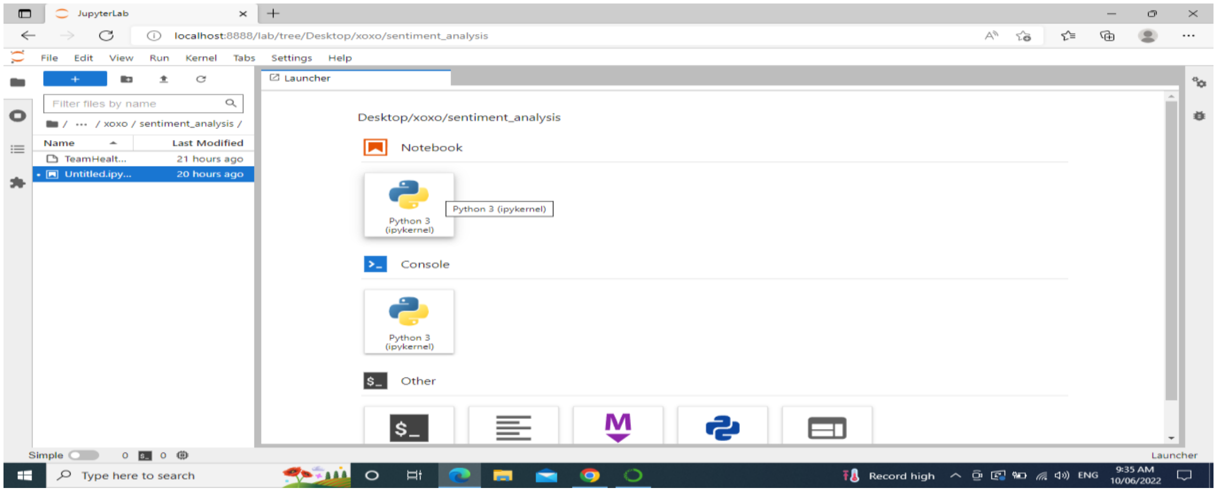
Run a Powershell terminal with your current environment from Navigator activated

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

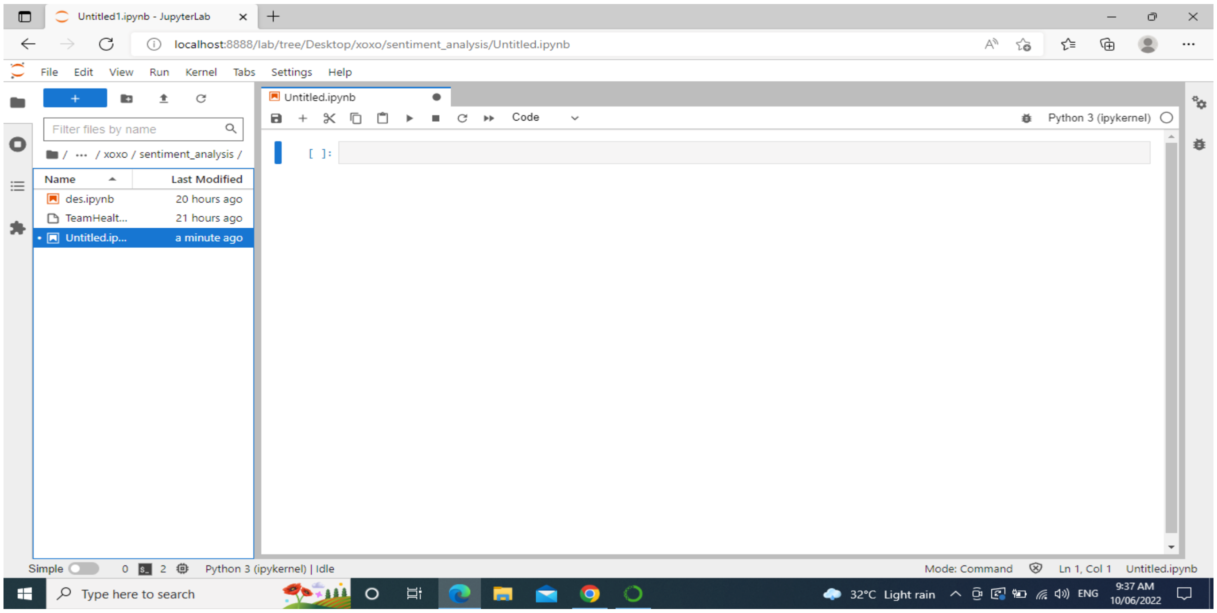
Figure 2. Create a new Python notebook

The new notebook (file name untitled.ipynb) will open in the same web browser.


JupyterLab
localhost:8888/lab/tree/Desktop/xoxo/sentiment_analysis
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ··· / xoxo / sentiment_analysis /
Name
Last Modified
TeamHealt... 21 hours ago
Untitled.ipy... 20 hours ago
Launcher
Desktop/xoxo/sentiment_analysis
Notebook
Python 3 (ipykernel)
Python 3 (ipykernel)
Console
Python 3 (ipykernel)
Other
Simple
Launcher
Type here to search
Record high
ENG
9:35 AM
10/06/2022

=

Figure 3. New Notebook


Untitled1.ipynb - JupyterLab
localhost:8888/lab/tree/Desktop/xoxo/sentiment_analysis/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ··· / xoxo / sentiment_analysis /
Name
Last Modified
des.ipynb 20 hours ago
TeamHealt... 21 hours ago
Untitled.ip... a minute ago
Untitled.ipynb
Code
Python 3 (ipykernel)
[ ]:
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Untitled.ipynb
Type here to search
32°C Light rain
ENG
9:37 AM
10/06/2022

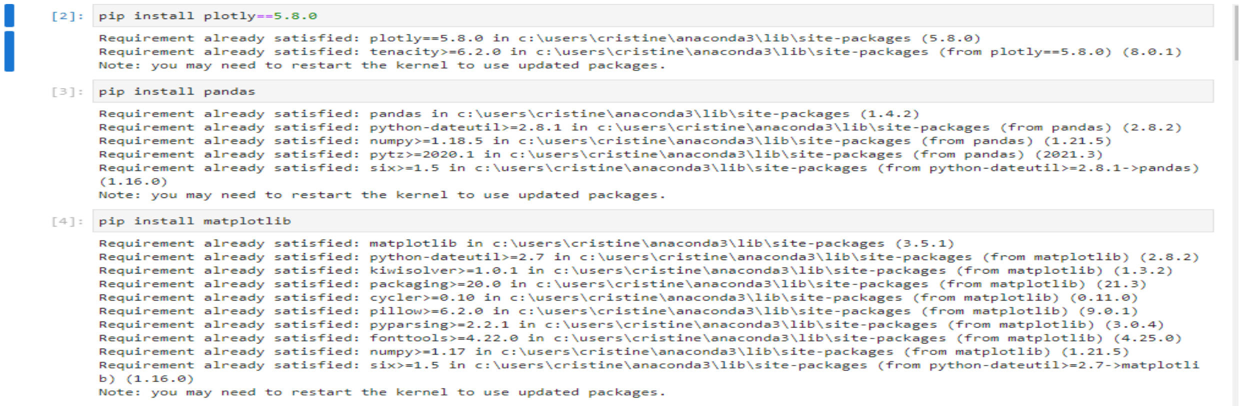
Import modules for sentiment analysis

This section introduces readers to Python modules used for sentiment analysis

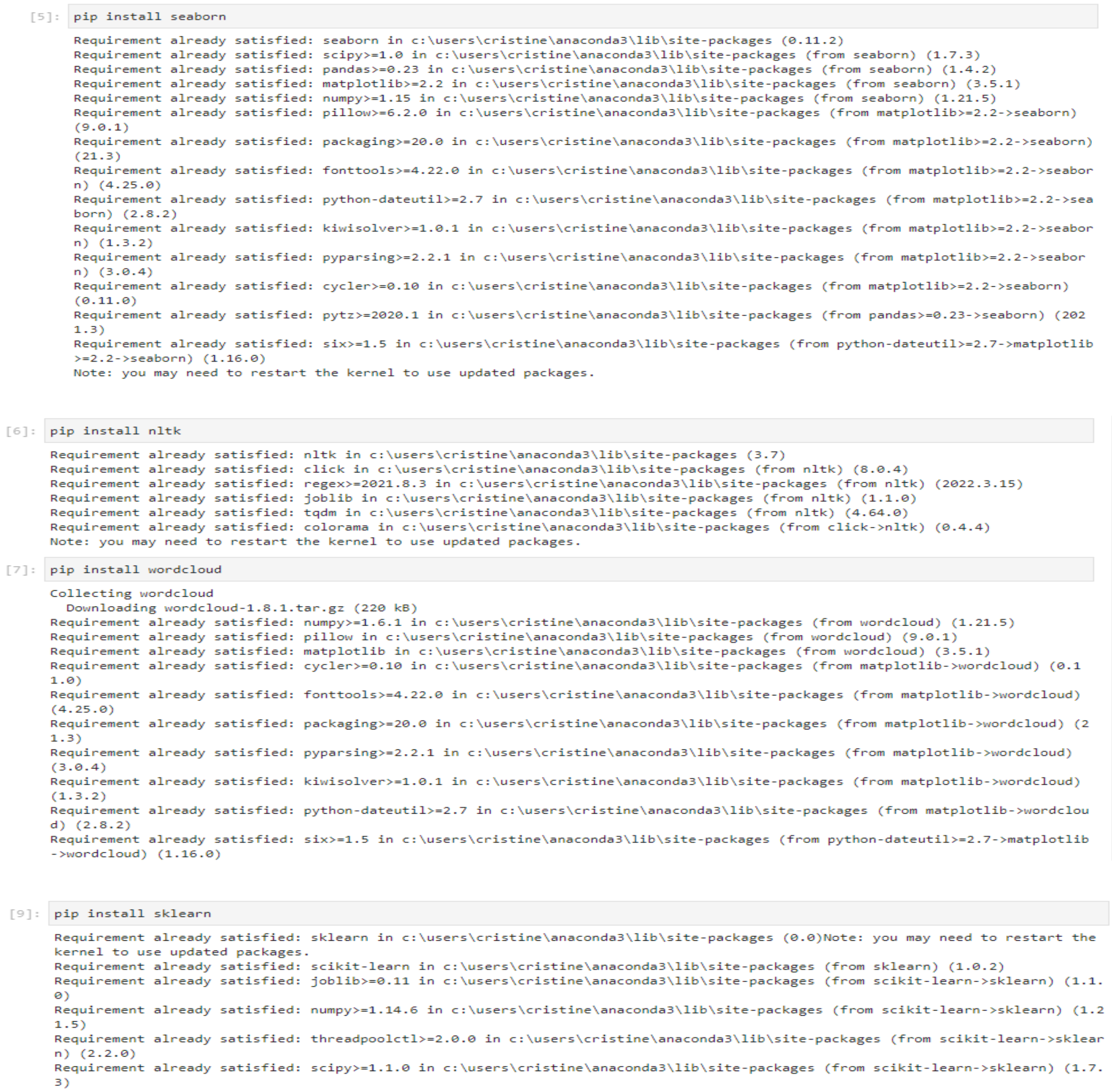
- ✓ The plotly Python library is an interactive, open-source plotting library that supports over 40 unique chart types covering a wide range of statistical, financial, geographic, scientific, and 3-dimensional use-cases.
- ✓ pandas is one of the most widely used open-source tools for data manipulation and analysis. Developed in 2008, pandas provides an incredibly fast and efficient object with integrated indexing, called DataFrame. It comes with tools for reading and writing data from and to files and SQL databases. It can manipulate, reshape, filter, aggregate, merge, join and pivot large datasets and is highly optimized for performance.
- ✓ matplotlib is an easy-to-use, popular and comprehensive library in Python for creating visualizations. It supports basic plots (like line, bar, scatter, etc.), plots of arrays & fields, statistical plots (like histogram, boxplot, violin, etc.), and plots with unstructured coordinates.
- ✓ The Natural Language Toolkit, commonly known as NLTK, is a comprehensive open-source platform for building applications to process human language data. It comes with powerful text processing libraries for typical Natural Language Processing (NLP) tasks like cleaning, parsing, stemming, tagging, tokenization, classification, semantic reasoning, etc. NLTK has user-friendly interfaces to several popular corpora and lexical resources Word2Vec, WordNet, VADER Sentiment Lexicon, etc.

- ✓ Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

```
pip install plotly==5.8.0
pip install pandas
pip install matplotlib
pip install seaborn
pip install nltk
pip install wordcloud
pip install sklearn
```



```
[2]: pip install plotly==5.8.0

     Requirement already satisfied: plotly==5.8.0 in c:\users\cristine\anaconda3\lib\site-packages (5.8.0)
     Requirement already satisfied: tenacity>=6.2.0 in c:\users\cristine\anaconda3\lib\site-packages (from plotly==5.8.0) (8.0.1)
     Note: you may need to restart the kernel to use updated packages.

[3]: pip install pandas

     Requirement already satisfied: pandas in c:\users\cristine\anaconda3\lib\site-packages (1.4.2)
     Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\cristine\anaconda3\lib\site-packages (from pandas) (2.8.2)
     Requirement already satisfied: numpy>=1.18.5 in c:\users\cristine\anaconda3\lib\site-packages (from pandas) (1.21.5)
     Requirement already satisfied: pytz>=2020.1 in c:\users\cristine\anaconda3\lib\site-packages (from pandas) (2021.3)
     Requirement already satisfied: six>=1.5 in c:\users\cristine\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas)
     (1.16.0)
     Note: you may need to restart the kernel to use updated packages.

[4]: pip install matplotlib

     Requirement already satisfied: matplotlib in c:\users\cristine\anaconda3\lib\site-packages (3.5.1)
     Requirement already satisfied: python-dateutil>=2.7 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
     Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (1.3.2)
     Requirement already satisfied: packaging>=20.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (21.3)
     Requirement already satisfied: cycler>=0.10 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
     Requirement already satisfied: pillow>=6.2.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (9.0.1)
     Requirement already satisfied: pyparsing>=2.2.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
     Requirement already satisfied: fonttools>=4.22.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (4.25.0)
     Requirement already satisfied: numpy>=1.17 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib) (1.21.5)
     Requirement already satisfied: six>=1.5 in c:\users\cristine\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotli
     b) (1.16.0)
     Note: you may need to restart the kernel to use updated packages.
```



```
[5]: pip install seaborn

     Requirement already satisfied: seaborn in c:\users\cristine\anaconda3\lib\site-packages (0.11.2)
     Requirement already satisfied: scipy>=1.0 in c:\users\cristine\anaconda3\lib\site-packages (from seaborn) (1.7.3)
     Requirement already satisfied: pandas>=0.23 in c:\users\cristine\anaconda3\lib\site-packages (from seaborn) (1.4.2)
     Requirement already satisfied: matplotlib>=2.2 in c:\users\cristine\anaconda3\lib\site-packages (from seaborn) (3.5.1)
     Requirement already satisfied: numpy>=1.15 in c:\users\cristine\anaconda3\lib\site-packages (from seaborn) (1.21.5)
     Requirement already satisfied: pillow>=6.2.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn)
     (9.0.1)
     Requirement already satisfied: packaging>=20.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn)
     (21.3)
     Requirement already satisfied: fonttools>=4.22.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seabor
     n) (4.25.0)
     Requirement already satisfied: python-dateutil>=2.7 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->sea
     born) (2.8.2)
     Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seabor
     n) (1.3.2)
     Requirement already satisfied: pyparsing>=2.2.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seabor
     n) (3.0.4)
     Requirement already satisfied: cycler>=0.10 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib>=2.2->seaborn)
     (0.11.0)
     Requirement already satisfied: pytz>=2020.1 in c:\users\cristine\anaconda3\lib\site-packages (from pandas>=0.23->seaborn) (202
     1.3)
     Requirement already satisfied: six>=1.5 in c:\users\cristine\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib
     >=2.2->seaborn) (1.16.0)
     Note: you may need to restart the kernel to use updated packages.
```

```
[6]: pip install nltk

     Requirement already satisfied: nltk in c:\users\cristine\anaconda3\lib\site-packages (3.7)
     Requirement already satisfied: click in c:\users\cristine\anaconda3\lib\site-packages (from nltk) (8.0.4)
     Requirement already satisfied: regex>=2021.8.3 in c:\users\cristine\anaconda3\lib\site-packages (from nltk) (2022.3.15)
     Requirement already satisfied: joblib in c:\users\cristine\anaconda3\lib\site-packages (from nltk) (1.1.0)
     Requirement already satisfied: tqdm in c:\users\cristine\anaconda3\lib\site-packages (from nltk) (4.64.0)
     Requirement already satisfied: colorama in c:\users\cristine\anaconda3\lib\site-packages (from click->nltk) (0.4.4)
     Note: you may need to restart the kernel to use updated packages.
```

```
[7]: pip install wordcloud

     Collecting wordcloud
       Downloading wordcloud-1.8.1.tar.gz (220 kB)
     Requirement already satisfied: numpy>=1.6.1 in c:\users\cristine\anaconda3\lib\site-packages (from wordcloud) (1.21.5)
     Requirement already satisfied: pillow in c:\users\cristine\anaconda3\lib\site-packages (from wordcloud) (9.0.1)
     Requirement already satisfied: matplotlib in c:\users\cristine\anaconda3\lib\site-packages (from wordcloud) (3.5.1)
     Requirement already satisfied: cycler>=0.10 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.1
     1.0)
     Requirement already satisfied: fonttools>=4.22.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordcloud)
     (4.25.0)
     Requirement already satisfied: packaging>=20.0 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2
     1.3)
     Requirement already satisfied: pyparsing>=2.2.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordcloud)
     (3.0.4)
     Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordcloud)
     (1.3.2)
     Requirement already satisfied: python-dateutil>=2.7 in c:\users\cristine\anaconda3\lib\site-packages (from matplotlib->wordclou
     d) (2.8.2)
     Requirement already satisfied: six>=1.5 in c:\users\cristine\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib
     ->wordcloud) (1.16.0)
```

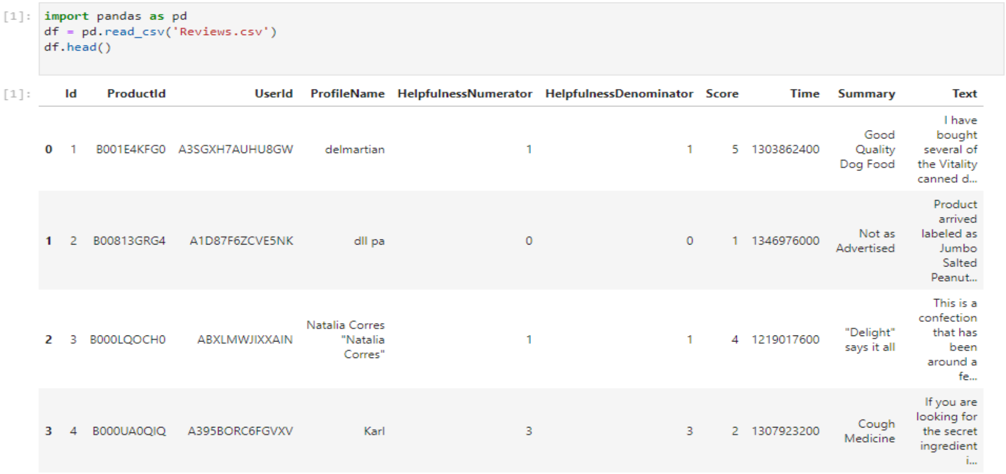
```
[9]: pip install sklearn

     Requirement already satisfied: sklearn in c:\users\cristine\anaconda3\lib\site-packages (0.0)Note: you may need to restart the
     kernel to use updated packages.
     Requirement already satisfied: scikit-learn in c:\users\cristine\anaconda3\lib\site-packages (from sklearn) (1.0.2)
     Requirement already satisfied: joblib>=0.11 in c:\users\cristine\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.1.
     0)
     Requirement already satisfied: numpy>=1.14.6 in c:\users\cristine\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.2
     1.5)
     Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\cristine\anaconda3\lib\site-packages (from scikit-learn->sklear
     n) (2.2.0)
     Requirement already satisfied: scipy>=1.1.0 in c:\users\cristine\anaconda3\lib\site-packages (from scikit-learn->sklearn) (1.7.
     3)
```

Figure 4 - Import relevant modules

Analysis

In this step I used Reviews.csv file from Kaggle's Amazon Fine Food Reviews dataset to perform the analysis. These will guide you through the end to end process of performing sentiment analysis on a large amount of data.

```
import pandas as pd
df = pd.read_csv('Reviews.csv')
df.head()
```



```
[1]: import pandas as pd
     df = pd.read_csv('Reviews.csv')
     df.head()
```

[1]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...

Figure 5 – Read Dataframe

The Data Frame contains some product, user and review information. The data that we will be using most for this analysis is "Summary", "Text", and "Score."

- ✓ Text — This variable contains the complete product review information.
- ✓ Summary — This is a summary of the entire review.
- ✓ Score — The product rating provided by the customer.

Data Analysis

Let’s look at the variable “Score” to see if majority of the customer ratings are positive or negative. I install plotly library first.

```
import matplotlib.pyplot as plt
import seaborn as sns
color = sns.color_palette()
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls
import plotly.express as px
fig = px.histogram(df, x="Score")
fig.update_traces(marker_color="turquoise",marker_line_color='rgb(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='Product Score')
fig.show()
```

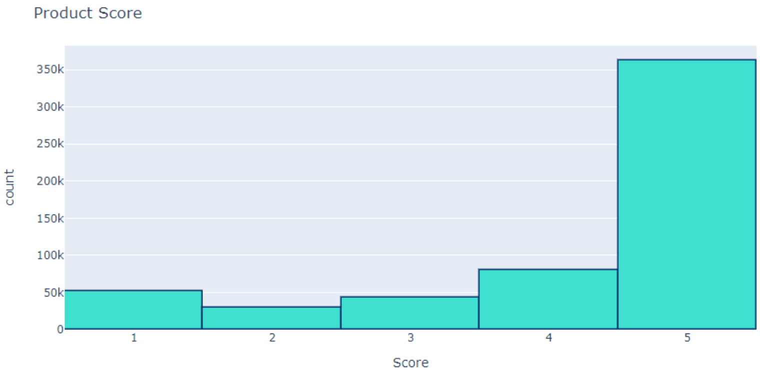


Figure 6 – Product Score

Generating Wordcloud

I will create a wordcloud to see the most frequently used words in the reviews.

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from wordcloud import WordCloud, STOPWORDS
stopwords = STOPWORDS
stopwords.update(["br", "href"])
textt = " ".join(review for review in df.Text)
wordcloud = WordCloud(stopwords=stopwords).generate(textt)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.savefig('picturetxt.png')
plt.show()
```

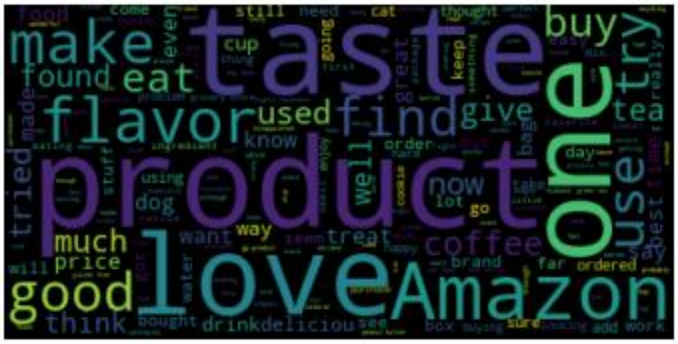


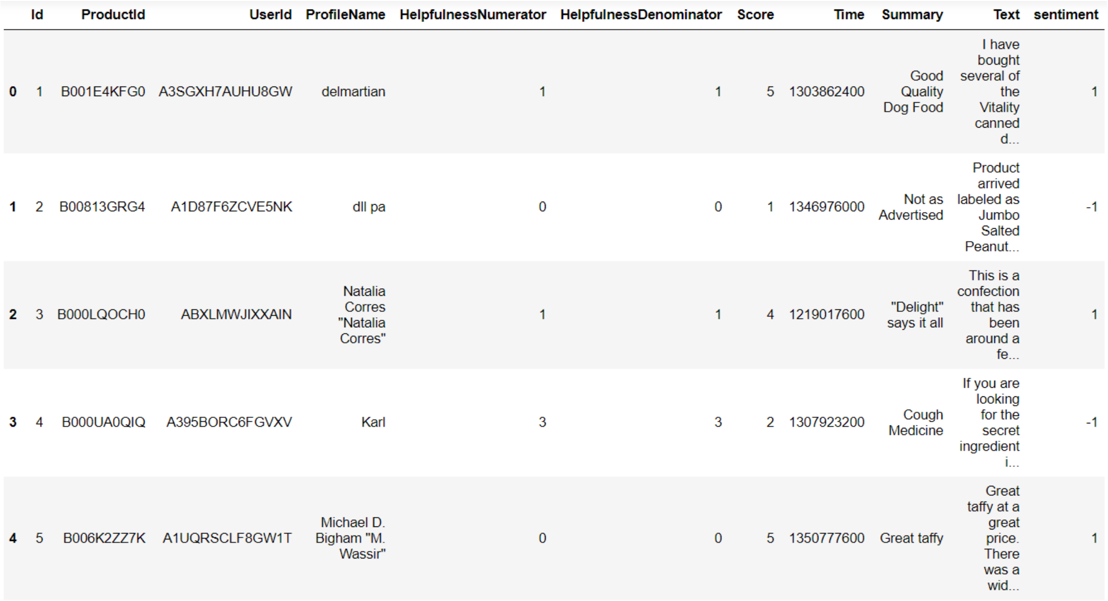
Figure 7 - Wordcloud

Classification

In this step, we will classify the reviews into positive and negative, so we can use this as training data for our sentiment classification model.

- Positive reviews will be classified as +1, and negative reviews will be classified as -1.
- We will classify all reviews with 'Score' > 3 as +1, indicating that they are positive.
- All reviews with 'Score' < 3 will be classified as -1. Reviews with 'Score' = 3 will be dropped, because they are neutral. Our model will only classify positive and negative reviews.

```
df = df[df['Score'] != 3]
df['sentiment'] = df['Score'].apply(lambda rating : +1 if rating > 3
else -1)
```



	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	sentiment
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...	1
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...	-1
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...	1
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...	-1
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...	1

Figure 8 - Classifying Tweet

More Data Analysis

Now that we are done classifying positive and negative tweet, we are going to generate wordcloud for each.

```
positive = df[df['sentiment'] == 1]
negative = df[df['sentiment'] == -1]
```

Positive Sentiment

```
stopwords = set(STOPWORDS)
stopwords.update(["br", "href","good","great"])
pos = ",".join(review for review in positive.Summary)
wordcloud2 = WordCloud(stopwords=stopwords).generate(pos)
plt.imshow(wordcloud2, interpolation='bilinear')
plt.axis("off")
plt.show()
```

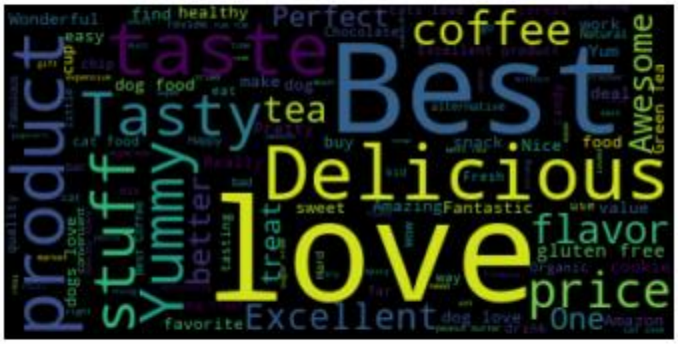


Figure 9 – Positive Sentiment

Negative Sentiment

```
stopwords = set(STOPWORDS)
neg = ','.join([str(review) for review in negative.Summary])
wordcloud3 = WordCloud(stopwords=stopwords).generate(neg)
plt.imshow(wordcloud3, interpolation='bilinear')
plt.axis("off")
plt.savefig('wordcloudneg.png')
plt.show()
```

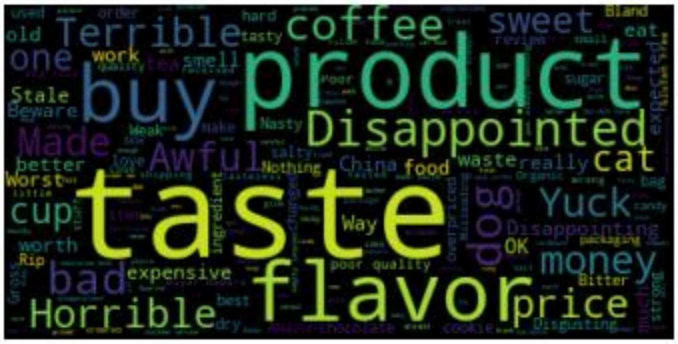


Figure 10 – Negative Sentiment

Distribution of Reviews with Sentiment

```
df['sentimentt'] = df['sentiment'].replace({-1 : 'negative'})
df['sentimentt'] = df['sentimentt'].replace({1 : 'positive'})
fig = px.histogram(df, x="sentimentt")
fig.update_traces(marker_color="indianred",marker_line_color='rgb
(8,48,107)',
                  marker_line_width=1.5)
fig.update_layout(title_text='Product Sentiment')
fig.show()
```

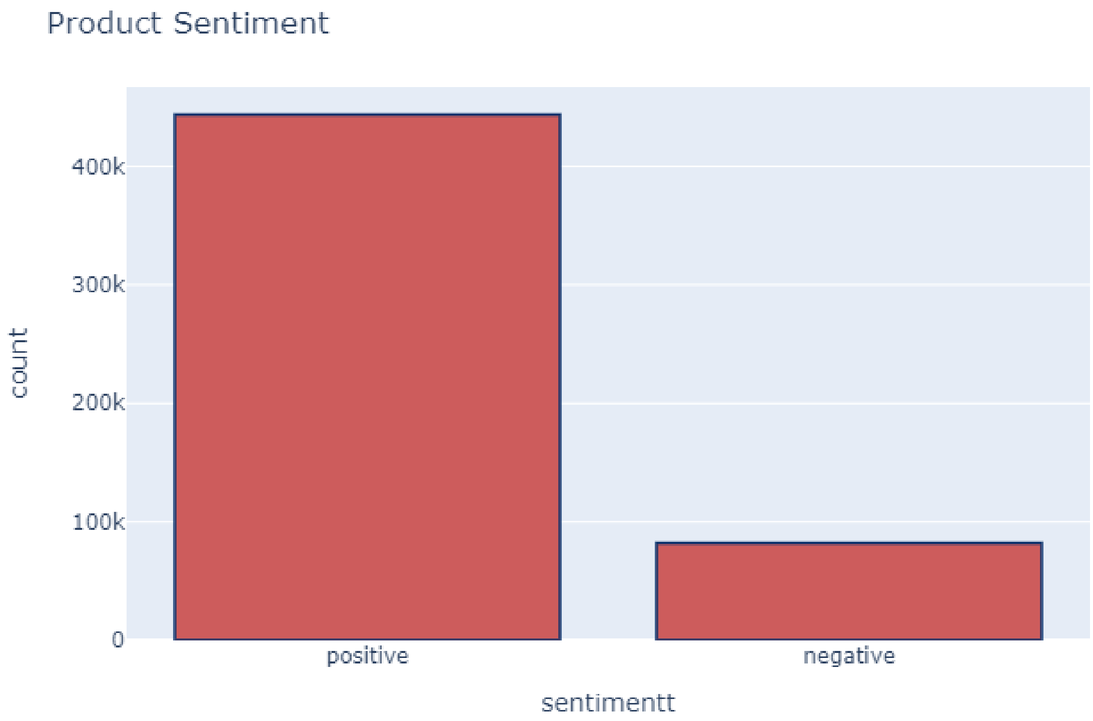


Figure 11 - Distribution of Reviews

Building the Model

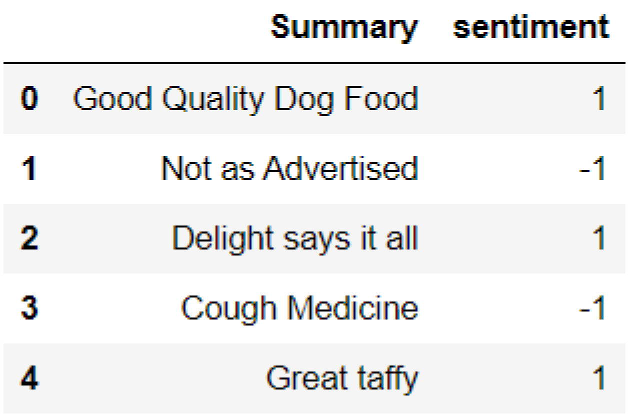
We can build the sentiment analysis model! This model will take reviews in as input. It will then come up with a prediction on whether the review is positive or negative. This is a classification task, so we will train a simple logistic regression model to do it.

- Data Cleaning - We will be using the summary data to come up with predictions. First, we need to remove all punctuation from the data.

```
def remove_punctuation(text):
    final = "".join(u for u in text if u not in ("?", ".", ";", ":",  "!",""))
    return final
df['Text'] = df['Text'].apply(remove_punctuation)
df = df.dropna(subset=['Summary'])
df['Summary'] = df['Summary'].apply(remove_punctuation)
```

- Split the Dataframe - The new data frame should only have two columns — “Summary” (the review text data), and “sentiment” (the target variable).

```
dfNew = df[['Summary','sentiment']]
dfNew.head()
```



	Summary	sentiment
0	Good Quality Dog Food	1
1	Not as Advertised	-1
2	Delight says it all	1
3	Cough Medicine	-1
4	Great taffy	1

Figure 13 – New Dataframe

We will now split the data frame into train and test sets. 80% of the data will be used for training, and 20% will be used for testing.

```
# random split train and test data
index = df.index
df['random_number'] = np.random.randn(len(index))
train = df[df['random_number'] <= 0.8]
test = df[df['random_number'] > 0.8]
```

- Create a Bag of Words - Next, we will use a count vectorizer from the Scikit-learn library.
  This will transform the text in our data frame into a bag of words model, which will contain a sparse matrix of integers. The number of occurrences of each word will be counted and printed. We will need to convert the text into a bag-of-words model since the logistic regression algorithm cannot understand text.

```
# count vectorizer:
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(token_pattern=r'\b\w+\b')
train_matrix = vectorizer.fit_transform(train['Summary'])
test_matrix = vectorizer.transform(test['Summary'])
```

- Import Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

- Split target and independent variables

```
X_train = train_matrix
X_test = test_matrix
y_train = train['sentiment']
y_test = test['sentiment']
```

- Fit model on data

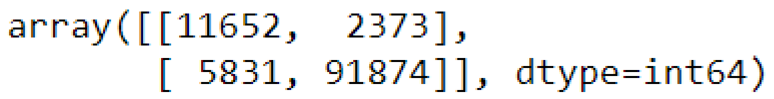
```
lr.fit(X_train,y_train)
```

- Make predictions

```
predictions = lr.predict(X_test)
```

Testing

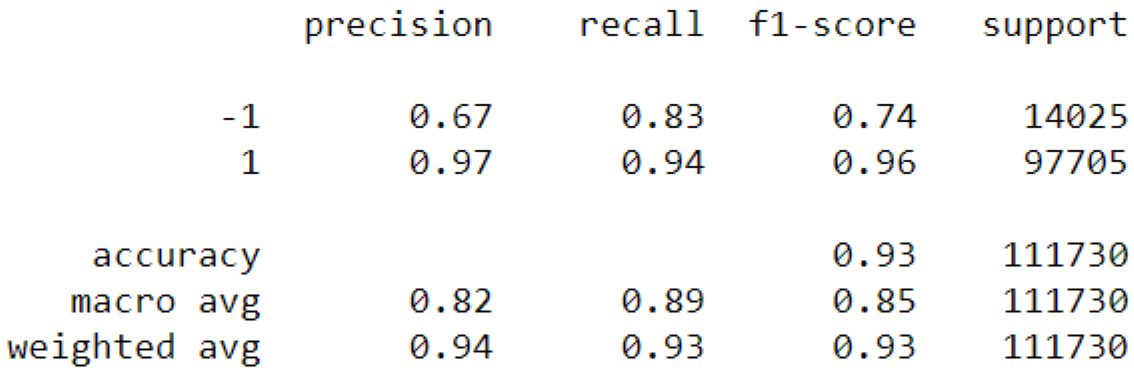
```
from sklearn.metrics import confusion_matrix,classification_report
new = np.asarray(y_test)
confusion_matrix(predictions,y_test)
```



```
array([[11652,  2373],
       [ 5831, 91874]], dtype=int64)
```

Figure 14 – Confision Matrix

```
print(classification_report(predictions,y_test))
```



```
              precision    recall  f1-score   support

          -1       0.67      0.83      0.74     14025
           1       0.97      0.94      0.96     97705

    accuracy                           0.93    111730
   macro avg       0.82      0.89      0.85    111730
weighted avg       0.94      0.93      0.93    111730
```

Figure 15 – Classification report