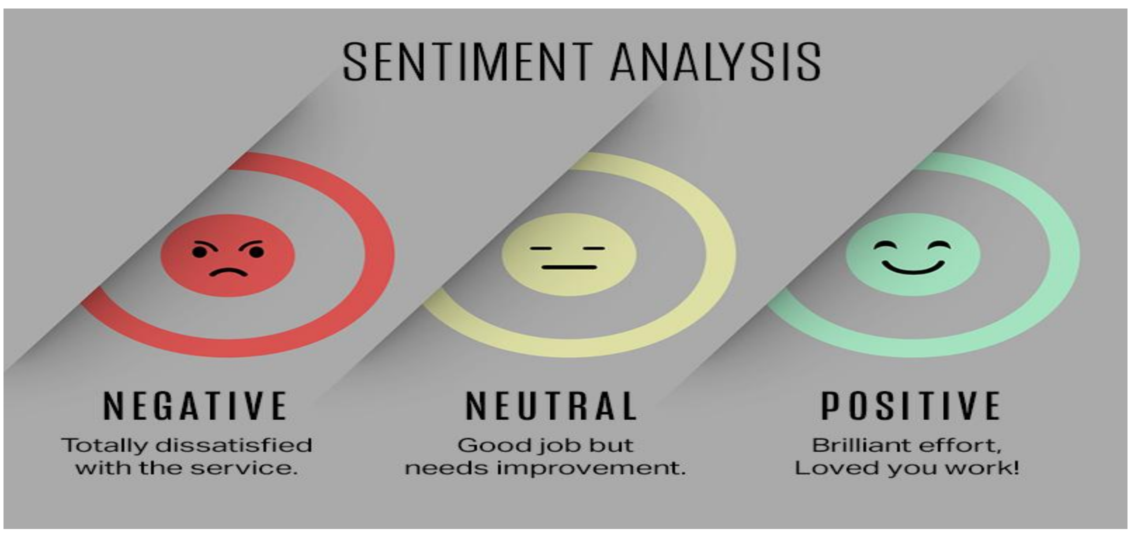
Sentimental Analysis For Marketing

Phase-5

Abstract



In the age of digital marketing and social media, understanding customer sentiment has become pivotal for businesses seeking a competitive edge. This project endeavors to harness the capabilities of sentiment analysis, a branch of natural language processing (NLP), to extract valuable insights from customer feedback and opinions. By scrutinizing text data from diverse sources such as social media, product reviews, and surveys, this project aims to categorize sentiments as positive, negative, or neutral and delve deeper into the nuances of consumer emotions.

Sentiment analysis in marketing involves analyzing customer opinions and feedback to gauge their feelings towards a product, brand, or campaign. It helps marketers understand customer sentiment, identify trends, and make data-driven decisions. This analysis can be done using natural language processing (NLP) tools and techniques to categorize opinions as positive, negative, or neutral. It helps in shaping marketing strategies, improving products, and engaging with customers effectively.

Software Requirements:

Front End Design: HTML

Database:MySQL

Programming Language:Python

In today's hyperconnected world, businesses face an ever-increasing challenge to understand and engage with their customer base effectively. Sentiment analysis, a subset of natural language processing (NLP), has emerged as a powerful tool to decipher the emotions and opinions expressed by customers in the digital realm. This project explores the application of sentiment analysis in marketing, revealing how it can unearth valuable insights, improve customer satisfaction, drive targeted campaigns, and ultimately, bolster a brand's competitive advantage. By harnessing the wealth of data generated by customer feedback and social media, businesses can craft strategies that resonate with their audience on a deeper level, fostering lasting customer relationships.

Objective:

The project's primary objectives encompass:

1. Robust Sentiment Analysis Model: Developing a sophisticated machine learning model capable of accurately classifying sentiments, while accounting for linguistic complexities, context, and idiomatic expressions.

2. Data Acquisition and Preprocessing: Collecting and meticulously preparing data from a multitude of channels, ensuring data cleanliness and optimizing it for analysis.

3. Real-time Monitoring and Reporting: Implementing a system for continuous sentiment tracking across various platforms, providing marketers with timely and actionable insights.

4. Trend Identification and Strategic Insights: Discerning emerging sentiment trends and patterns to empower marketers with data-driven decision-making tools, aiding in the refinement of marketing strategies and the prompt resolution of customer concerns.

5. Integration with Marketing Initiatives: Seamlessly integrating sentiment analysis findings into marketing campaigns, including personalized content generation, targeted advertising, and proactive customer engagement.

6. Continuous Evaluation and Enhancement: Periodically assessing the performance of the sentiment analysis model and refining it in response to evolving market dynamics and user feedback.

This project endeavors to equip marketing teams with the necessary tools and knowledge to forge deeper connections with customers, bolster brand reputation, and propel business growth. Through the utilization of sentiment analysis, organizations can gain a strategic advantage in the modern, customer-centric business landscape.

Advantages:

1. Customer Insights:

Sentiment analysis provides valuable insights into how customers perceive your products, services, and brand. It helps you understand their preferences, pain points, and expectations.

2. Data-Driven Decisions:

By analyzing sentiment data, marketers can make data-driven decisions, refine marketing strategies, and allocate resources effectively. This leads to more efficient and cost-effective campaigns.

3. Competitive Analysis:

You can use sentiment analysis to monitor and compare sentiment about your brand with that of your competitors. This information can help you identify areas for improvement and opportunities for differentiation.

4. Real-Time Feedback:

Sentiment analysis tools can provide real-time feedback, allowing you to respond promptly to customer concerns and address issues as they arise. This enhances customer satisfaction and loyalty.

5. Product Development:

Sentiment analysis can inform product development by identifying features that customers love or dislike. This helps in creating products that align better with customer expectations.

6. Personalization:

Sentiment analysis can contribute to personalized marketing efforts. You can customize content and offers based on individual customer sentiments and preferences.

Elaboration:

Sentiment analysis for marketing, also known as opinion mining, is a valuable tool for businesses to gauge public sentiment about their products, services, or brand. Here's an elaboration on how it works and its importance in marketing:

1. Definition of Sentiment Analysis:

Sentiment analysis is the process of using natural language processing and machine learning techniques to determine the sentiment or emotion expressed in text data, such as social media posts, reviews, comments, and more.

2.Importance in Marketing:

Customer Insights:

Sentiment analysis allows businesses to understand how customers feel about their products or services. This insight can help in product development and refinement.

Reputation Management:

Monitoring online sentiment helps in managing a brand's reputation by identifying and addressing negative sentiment in a timely manner.

Competitor Analysis:

Businesses can analyze sentiment surrounding competitors to identify gaps in the market or areas where they can outperform rivals.

Marketing Campaigns:

It can inform marketing strategies by understanding what resonates with the target audience and what doesn't.

Customer Service:

Sentiment analysis can identify customer issues and complaints, allowing companies to provide better customer support.

3. Techniques Used:

Lexicon-Based Analysis:
This method uses predefined sentiment scores for words or phrases to determine sentiment in the text.

Machine Learning:

ML models are trained on labeled data to classify text into positive, negative, or neutral sentiments.

Aspect-Based Sentiment Analysis:

This approach dissects text to determine sentiment at a more granular level, focusing on specific aspects of a product or service.

4. Challenges:

Context Understanding:
It can be challenging to discern sarcasm, irony, or cultural nuances in text.

Data Volume:
Analyzing large volumes of data can be resource-intensive.

Accuracy: Achieving high accuracy in sentiment analysis, especially for complex, nuanced opinions, is an ongoing challenge.

5. Tools and Platforms:

Several tools and platforms offer sentiment analysis services, such as Google Cloud Natural Language API, IBM Watson, and more. These tools help automate the process.

6. Real-World Examples:

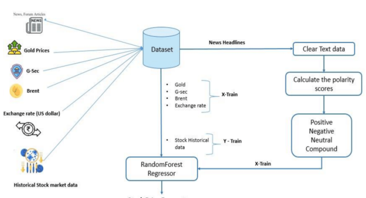
- Companies often track social media sentiment to adjust their marketing strategies.
- Online retailers use sentiment analysis to understand product reviews and improve product listings.

7.Ethical Considerations:

It's essential to respect privacy and handle customer data responsibly when conducting sentiment analysis.

In conclusion, sentiment analysis in marketing is a valuable technique that helps businesses gain insights into customer opinions, refine their strategies, and manage their reputation effectively. It plays a crucial role in today's data-driven marketing landscape.

Block Diagram:



Block Diagram for Sentiment Analysis System:

A block diagram for a sentiment analysis system might consist of the following key components:

1.Data Input Block:

Represents the data sources, such as social media, review websites, or surveys.

2.Preprocessing Block:

This block includes processes like data cleaning, tokenization, and text normalization.

3.Sentiment Analysis Block:

Contains the sentiment analysis algorithm or model, which classifies the text data into positive, negative, or neutral sentiments.

4.Positive Sentiment Action Block:

Depicts actions taken in response to positive sentiment, such as marketing campaigns, promotions, or content creation.

5.Negative Sentiment Action Block:

Shows actions taken in response to negative sentiment, such as customer support, product improvement, or crisis management.

6.Tracking and Reporting Block:

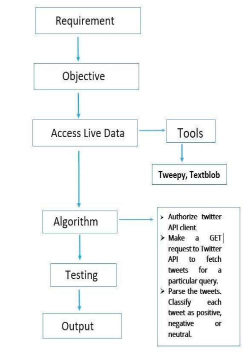
Represents the components responsible for monitoring sentiment trends and generating reports.

7.Data Output Block:

This is where the analyzed data, reports, or insights are provided to the marketing team for decision-making.

Please note that you would typically use specialized software or drawing tools to create the actual flowchart and block diagram. These textual descriptions can serve as a guideline for creating visual representations of the sentiment analysis process in marketing.

Flow Chart:



Flowchart for Sentiment Analysis in Marketing:

1.Start:

The process begins here.

2.Data Collection:

This block represents the collection of textual data from various sources like social media, customer reviews, or surveys.

3. Data Preprocessing:

Data preprocessing involves cleaning and preparing the text data. It might include tasks like removing stop words, special characters, and stemming.

4.Sentiment Analysis:

This is a decision block where the sentiment analysis process takes place. It branches into three paths:

Positive Sentiment:

If the sentiment is positive, the flow continues to a block for positive sentiment actions.

Negative Sentiment:

If the sentiment is negative, the flow continues to a block for negative sentiment actions.

Neutral Sentiment:

If the sentiment is neutral, the flow might proceed to a block for tracking or further analysis.

5.Positive Sentiment Actions:

This block includes actions that can be taken in response to positive sentiment, like promoting positive reviews, creating marketing content, or thanking customers.

6.Negative Sentiment Actions:

This block includes actions taken in response to negative sentiment, such as addressing customer complaints, improving products or services, or managing a PR crisis.

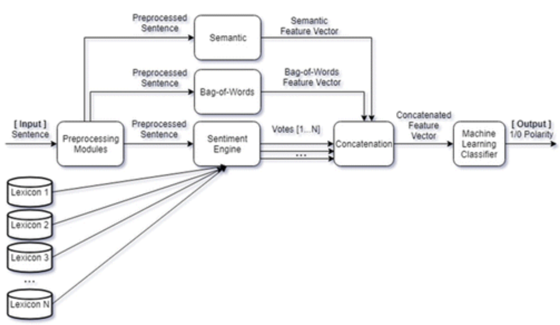
7.Tracking and Reporting:

This block represents the process of tracking sentiment trends over time and generating reports for marketing teams.

8.End:

The process concludes here.

Enhanced Framework:



Building project:

We started building our project by loading the dataset, performing text preprocessing, and conducting analysis in Colab Notebook.

Sentimental Analysis with Python:

To build a machine learning model to accurately classify whether customers are saying positive or negative.

Steps to build Sentiment Analysis Text Classifier in Python

1. Data Preprocessing:

As we are dealing with the text data, we need to preprocess it using word embeddings.

```
import pandas as pd
df =
pd.read_csv("./DesktopDataFlair/Sentime
nt-Analysis/Tweets.csv")
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold	name
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	NaN	cairdin
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	NaN	jnardino
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	NaN	yvonnalynn
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN	jnardino
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN	jnardino

We only need the text and sentiment column.

```
review_df =
df[['text','airline_sentiment']]

print(review_df.shape)
review_df.head(5)
```

```
(14640, 2)
Out[4]:
```

	text	airline_sentiment
0	@VirginAmerica What @dhepburn said.	neutral
1	@VirginAmerica plus you've added commercials t...	positive
2	@VirginAmerica I didn't today... Must mean I n...	neutral
3	@VirginAmerica it's really aggressive to blast...	negative
4	@VirginAmerica and it's a really big bad thing...	negative

```
df.columns
```

```
Out[23]: Index(['tweet_id', 'airline_sentiment', 'airline_sentiment_confidence',
        'negativereason', 'negativereason_confidence', 'airline',
        'airline_sentiment_gold', 'name', 'negativereason_gold',
        'retweet_count', 'text', 'tweet_coord', 'tweet_created',
        'tweet_location', 'user_timezone'],
       dtype='object')
```

```
review_df =
review_df[review_df['airline_sentiment'
] != 'neutral']

print(review_df.shape)
review_df.head(5)
```

```
(11541, 2)
Out[22]:
```

	text	airline_sentiment
1	@VirginAmerica plus you've added commercials t...	positive
3	@VirginAmerica it's really aggressive to blast...	negative
4	@VirginAmerica and it's a really big bad thing...	negative
5	@VirginAmerica seriously would pay $30 a fligh...	negative
6	@VirginAmerica yes, nearly every time I fly VX...	positive

The labels for this dataset are categorical. Machines understand only numeric data. So, convert the categorical values to numeric using the factorize() method. This returns an array of numeric values and an Index of categories.

```
sentiment_label =
review_df.airline_sentiment.factorize()
sentiment_label
```

```
In [6]: sentiment_label = tweet_df.airline_sentiment.factorize()
        sentiment_label

Out[6]: (array([0, 1, 1, ..., 0, 1, 1], dtype=int64),
         Index(['positive', 'negative'], dtype='object'))
```

2. Build the Text Classifier:

For sentiment analysis project, we use LSTM layers in the machine learning model. The architecture of our model consists of an embedding layer, an LSTM layer, and a Dense layer at the end. To avoid overfitting, we introduced the Dropout mechanism in-between the LSTM layers.

```
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import
LSTM,Dense, Dropout, SpatialDropout1D
from tensorflow.keras.layers import
Embedding


embedding_vector_length = 32
model = Sequential()
model.add(Embedding(vocab_size,
embedding_vector_length,
input_length=200))
model.add(SpatialDropout1D(0.25))
model.add(LSTM(50, dropout=0.5,
recurrent_dropout=0.5))
model.add(Dropout(0.2))
model.add(Dense(1,
activation='sigmoid'))
model.compile(loss='binary_crossentropy
',optimizer='adam', metrics=
['accuracy'])

print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 200, 32)           423488
_________________________________________________________________
spatial_dropout1d (SpatialDr (None, 200, 32)           0
_________________________________________________________________
lstm (LSTM)                  (None, 50)                16600
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 51
=================================================================
Total params: 440,139
Trainable params: 440,139
Non-trainable params: 0
_________________________________________________________________
None
```

3. Train the sentiment analysis model:

Train the sentiment analysis model for 5 epochs on the whole dataset with a batch size of 32 and a validation split of 20%.

```
history =
model.fit(padded_sequence,sentiment_lab
el[0],validation_split=0.2, epochs=5,
batch_size=32)
```

The output while training looks like below:

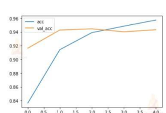
```
Epoch 1/5
289/289 [==============================] - 471s 2s/step - loss: 0.4916 - accuracy: 0.7980 - val_loss: 0.2133 - val_accuracy: 0.
9164
Epoch 2/5
289/289 [==============================] - 457s 2s/step - loss: 0.2282 - accuracy: 0.9118 - val_loss: 0.1624 - val_accuracy: 0.
9428
Epoch 3/5
289/289 [==============================] - 423s 1s/step - loss: 0.1755 - accuracy: 0.9340 - val_loss: 0.1667 - val_accuracy: 0.
9446
Epoch 4/5
289/289 [==============================] - 420s 1s/step - loss: 0.1292 - accuracy: 0.9519 - val_loss: 0.1678 - val_accuracy: 0.
9402
Epoch 5/5
289/289 [==============================] - 420s 1s/step - loss: 0.1117 - accuracy: 0.9600 - val_loss: 0.1818 - val_accuracy: 0.
9433
```

```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'],
label='acc')
plt.plot(history.history['val_accuracy'
], label='val_acc')
plt.legend()
plt.show()

plt.savefig("Accuracy plot.jpg")
```

Output:



Let's execute sentiment analysis model
Define a function that takes a text as input and outputs its prediction label.

```
def predict_sentiment(text):
    tw =
tokenizer.texts_to_sequences([text])
    tw = pad_sequences(tw,maxlen=200)
    prediction =
int(model.predict(tw).round().item())
    print("Predicted label: ",
sentiment_label[1][prediction])


test_sentence1 = "I enjoyed my journey
on this flight."
predict_sentiment(test_sentence1)

test_sentence2 = "This is the worst
flight experience of my life!"
predict_sentiment(test_sentence2)
```

Python Sentiment Analysis Output

Final Building:

We further building our project by loading the Data Set and Describing that and Cleaning the data and Visualising the distributions and Evaluation in Google colab Notebook.

Let’s Import the necessary Modules and take a look at the data:

```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import math
import warnings
warnings.filterwarnings('ignore') # Hides warning
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore",category=UserWarning)
sns.set_style("whitegrid") # Plotting style
np.random.seed(7) # seeding random number generator

df = pd.read_csv('amazon.csv')
print(df.head())
```

Describing the Dataset:

Overall description about the dataset should be contain in this
The purpose of this to done a overall relationship between the data and the future predictions based upon that.

```
                     id  ... reviews.username
0  AVqkIhwDv8e3D1O-lebb  ...          Adapter
1  AVqkIhwDv8e3D1O-lebb  ...           truman
2  AVqkIhwDv8e3D1O-lebb  ...            DaveZ
3  AVqkIhwDv8e3D1O-lebb  ...           Shacks
4  AVqkIhwDv8e3D1O-lebb  ...        explore42

[5 rows x 21 columns]
```

Describing the Dataset

```python
data = df.copy()
data.describe()
```

	reviews.id	reviews.numHelpful	reviews.rating	reviews.userCity	reviews.userProvince
count	1.0	34131.000000	34627.000000	0.0	0.0
mean	111372787.0	0.630248	4.584573	NaN	NaN
std	NaN	13.215775	0.735653	NaN	NaN
min	111372787.0	0.000000	1.000000	NaN	NaN
25%	111372787.0	0.000000	4.000000	NaN	NaN
50%	111372787.0	0.000000	5.000000	NaN	NaN

We need to clean up the name column by referencing asins (unique products) since we have 7000 missing values:

Describing the Dataset

```
data = df.copy()
data.describe()
```

	reviews.id	reviews.numHelpful	reviews.rating	reviews.userCity	reviews.userProvince
count	1.0	34131.000000	34627.000000	0.0	0.0
mean	111372787.0	0.630248	4.584573	NaN	NaN
std	NaN	13.215775	0.735653	NaN	NaN
min	111372787.0	0.000000	1.000000	NaN	NaN
25%	111372787.0	0.000000	4.000000	NaN	NaN
50%	111372787.0	0.000000	5.000000	NaN	NaN
75%	111372787.0	0.000000	5.000000	NaN	NaN
max	111372787.0	814.000000	5.000000	NaN	NaN

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34660 entries, 0 to 34659
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    34660 non-null  object
 1   name                  27900 non-null  object
 2   asins                 34658 non-null  object
 3   brand                 34660 non-null  object
 4   categories            34660 non-null  object
 5   keys                  34660 non-null  object
 6   manufacturer          34660 non-null  object
 7   reviews.date          34621 non-null  object
 8   reviews.dateAdded     24039 non-null  object
 9   reviews.dateSeen      34660 non-null  object
 10  reviews.didPurchase   1 non-null      object
 11  reviews.doRecommend   34066 non-null  object
 12  reviews.id            1 non-null      float64
 13  reviews.numHelpful    34131 non-null  float64
 14  reviews.rating        34627 non-null  float64
 15  reviews.sourceURLs    34660 non-null  object
 16  reviews.text          34659 non-null  object
 17  reviews.title         34655 non-null  object
 18  reviews.userCity      0 non-null      float64
 19  reviews.userProvince  0 non-null      float64
 20  reviews.username      34658 non-null  object
dtypes: float64(5), object(16)
memory usage: 5.6+ MB
```

```
data["asins"].unique()
```

```
array(['B01AHB9CN2', 'B00VINDBJK', 'B005PB2T0S', 'B002Y27P3M',
       'B01AHB9CYG', 'B01AHB9C1E', 'B01J2G4VBG', 'B00ZV9PXP2',
       'B0083Q04TA', 'B018Y229OU', 'B00REQKWGA', 'B00IOYAM4I',
       'B018T075DC', nan, 'B00DU15MU4', 'B018Y225IA', 'B005PB2T2Q',
       'B018Y23MNM', 'B00OQVZDJM', 'B00IOY8XWQ', 'B00LO29KXQ',
       'B00QJDU3KY', 'B018Y22C2Y', 'B01BFIBRIE', 'B01J4ORNHU',
       'B018SZT3BK', 'B00UH4D8G2', 'B018Y22BI4', 'B00TSUGXKE',
       'B00L9EPT8O,B01E6AO69U', 'B018Y23P7K', 'B00X4WHP5E', 'B00QFQRELG',
       'B00LW9XOJM', 'B00QL1ZN3G', 'B0189XYY0Q', 'B01BH83OOM',
       'B00BFJAHF8', 'B00U3FPN4U', 'B002Y27P6Y', 'B006GWO5NE',
       'B006GWO5WK'], dtype=object)
```

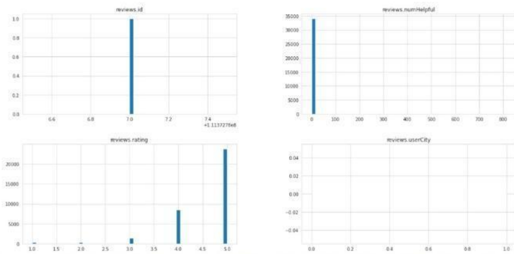
```
asins_unique = len(data["asins"].unique())
print("Number of Unique ASINs: " + str(asins_unique))
```

#Output– Number of Unique ASINs: 42

Visualizing the distributions of numerical variables:

Visualizing the distributions of numerical variables:

```
data.hist(bins=50, figsize=(20,15))
plt.show()
```




```
from sklearn.model_selection import StratifiedShuffleSplit
print("Before {}".format(len(data)))
dataAfter = data.dropna(subset=["reviews.rating"])
# Removes all NAN in reviews.rating
print("After {}".format(len(dataAfter)))
dataAfter["reviews.rating"] = dataAfter["reviews.rating"].astype(int)

split = StratifiedShuffleSplit(n_splits=5, test_size=0.2)
for train_index, test_index in split.split(dataAfter,
                                           dataAfter["reviews.rating"]):
    strat_train = dataAfter.reindex(train_index)
    strat_test = dataAfter.reindex(test_index)
```

#Output-

Before 34660
After 34627

Outliers in this case are valuable, so we may want to weight reviews that had more than 50+ people who find them helpful.

Majority of examples were rated highly (looking at rating distribution). There is twice amount of 5 star ratings than the others ratings combined.

Split the data into Train and Test:

Lets see all the different names for this product that have 2 ASINs:

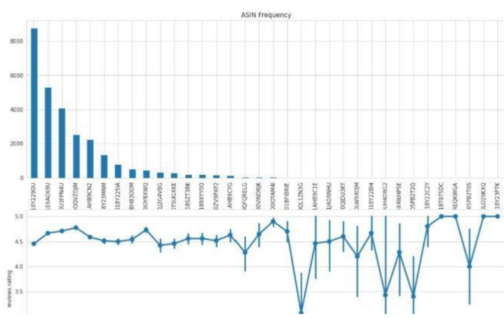
```
different_names = reviews[reviews["asins"] ==
                          "B00L9EPT8O,B01E6AO69U"]["name"].unique()
for name in different_names:
    print(name)
print(reviews[reviews["asins"] == "B00L9EPT8O,B01E6AO69U"]["name"].value_co
```

```
#Output
Echo (White),,,
Echo (White),,,
Amazon Fire Tv,,,
Amazon Fire Tv,,,
nan
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker - Black,,,
Amazon - Amazon Tap Portable Bluetooth and Wi-Fi Speaker - Black,,,
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver Aluminum,,,
Amazon Fire Hd 10 Tablet, Wi-Fi, 16 Gb, Special Offers - Silver Aluminum,,,
Amazon 9W PowerFast Official OEM USB Charger and Power Adapter for Fire Tab
```

Data Exploration (Training Set):

We will use regular expressions to clean out any unfavorable characters in the dataset, and then preview what the data looks like after cleaning.

```
plt.show()
```



Sentimental Analysis:

Using the features in place, we will build a classifier that can determine a review's sentiment.

```
def sentiments(rating):
    if (rating == 5) or (rating == 4):
        return "Positive"
    elif rating == 3:
        return "Neutral"
    elif (rating == 2) or (rating == 1):
        return "Negative"
# Add sentiments to the data
strat_train["Sentiment"] = strat_train["reviews.rating"].apply(sentiments)
strat_test["Sentiment"] = strat_test["reviews.rating"].apply(sentiments)
print(strat_train["Sentiment"][:20])
```

#Output-

Output:

Final output with the above code should be compiled and described in this,
The output is:

#Output-

4349	Positive
30776	Positive
28775	Neutral
1136	Positive
17803	Positive
7336	Positive
32638	Positive
13995	Positive
6728	Negative
22009	Positive
11047	Positive
22754	Positive
5578	Positive
11673	Positive
19168	Positive
14903	Positive
30843	Positive
5440	Positive
28940	Positive
31258	Positive

Team Members

I.JaharaAsmi
M.Muthulakshmi
K.Santhiya
M.Shajitha Barveen
M.SathiyaJothi

JP College Of Engineering