**SENTIMENTAL ANALYSIS ON LIVE TWEETS**

Sentiment analysis on live tweets involves analyzing real-time data from Twitter to determine the sentiment or emotional tone expressed in the tweets. This type of analysis can be useful for various applications, including market research, public opinion monitoring, customer feedback analysis, and event tracking.

**Sentiment Analysis Techniques**

**Lexicon-Based Approaches:** Using predefined dictionaries of words labeled with their sentiment values (e.g., positive, negative, neutral).

**Machine Learning Models:** Training models like Naive Bayes, Support Vector Machines (SVM), or Logistic Regression on labeled datasets.

**Deep Learning Models:** Using advanced models such as Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), or Transformer-based models like BERT.

**Example Workflow**

**Collect Tweets:** Use the Twitter API to stream tweets containing the keyword "new product launch."

**Preprocess Tweets:** Clean and preprocess the collected tweets.

**Analyze Sentiment:** Apply a pre-trained sentiment analysis model to classify each tweet as positive, negative, or neutral.

**Visualize Results:** Display the results on a real-time dashboard, showing sentiment distribution and trends over time.

**Future of Sentimental Analysis**

* Improved Accuracy with Advanced AI Models
* Multilingual and Multimodal Analysis
* Real-Time and Predictive Sentiment Analysis
* Emotion Detection and Fine-Grained Sentiment Analysis
* Integration with Other Technologies
* Ethical and Privacy Considerations
* Domain-Specific Applications

**Challenges**

**Noisy Data:** Tweets often contain slang, abbreviations, and typos, making preprocessing challenging.

**Sarcasm and Irony:** Detecting sarcastic or ironic tweets requires more advanced natural language processing techniques.

**Language Variability:** Tweets can be in different languages and dialects, necessitating multilingual sentiment analysis capabilities.

**Context Dependency:** Sentiment can depend on context, which may not always be clear from a single tweet.

**Applications**

**Brand Monitoring:** Companies can track public sentiment about their products or services.

**Political Campaigns:** Politicians and parties can gauge public opinion on various issues and adjust their strategies accordingly.

**Crisis Management:** Organizations can detect and respond to negative sentiment during crises or emergencies.

**Market Research:** Businesses can analyze consumer sentiment towards new products, services, or marketing campaigns.

**Code**

# Commented out IPython magic to ensure Python compatibility.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# %matplotlib inline

df = pd.read\_csv('/content/Reviews.csv')

df.head()

df.shape

df.info()

df.Summary.head()

df.Text.head()

!pip install textblob

from nltk.corpus import stopwords

from textblob import TextBlob

from textblob import Word

# Lower casing and removing punctuations

df['Text'] = df['Text'].apply(lambda x: " ".join(x.lower() for

x in x.split()))

df['Text'] = df['Text'].str.replace('[^\w\s]', "")

df.Text.head(5)

string = "yoo the food was delicious "

print(string.replace("delicious", "Yummy"))

stop = stopwords.words('english')

df['Text'] = df['Text'].apply(lambda x: " ".join(x for x in

x.split() if x not in stop))

df.Text.head()

import nltk

nltk.download('wordnet')

df['Text'] = df['Text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()]))

df.Text.head()

! pip install wordcloud

from wordcloud import WordCloud

from wordcloud import STOPWORDS

df.columns

df.Score.value\_counts()

import seaborn as sns

sns.countplot(data = df, x = 'Score')

reviews = df

reviews.dropna(inplace=True)

score\_1 = reviews[reviews['Score'] == 1]

score\_2 = reviews[reviews['Score'] == 2]

score\_3 = reviews[reviews['Score'] == 3]

score\_4 = reviews[reviews['Score'] == 4]

score\_5 = reviews[reviews['Score'] == 5]

reviews\_sample = pd.concat([score\_1,score\_2,score\_3,score\_4,score\_5],axis=0)

reviews\_sample.reset\_index(drop=True,inplace=True)

reviews\_str = reviews\_sample.Summary.str.cat()

wordcloud = WordCloud(background\_color='white').generate(reviews\_str)

plt.figure(figsize=(10,10))

plt.imshow(wordcloud,interpolation='bilinear')

plt.axis("off")

plt.show()

negative\_reviews = reviews\_sample[reviews\_sample['Score'].isin([1,2]) ]

positive\_reviews = reviews\_sample[reviews\_sample['Score'].isin([4,5]) ]

negative\_reviews\_str = negative\_reviews.Summary.str.cat()

positive\_reviews\_str = positive\_reviews.Summary.str.cat()

wordcloud\_negative = WordCloud(background\_color='white').generate(negative\_reviews\_str)

wordcloud\_positive = WordCloud(background\_color='black').generate(positive\_reviews\_str)

# Plot

fig = plt.figure(figsize=(10,10))

ax1 = fig.add\_subplot(211)

ax1.imshow(wordcloud\_negative,interpolation='bilinear')

ax1.axis("off")

ax1.set\_title('Reviews with Negative Scores',fontsize=20)

fig = plt.figure(figsize=(10,10))

ax2 = fig.add\_subplot(212)

ax2.imshow(wordcloud\_positive,interpolation='bilinear')

ax2.axis("off")

ax2.set\_title('Reviews with Positive Scores',fontsize=20)

plt.show()

!pip install vaderSentiment

import seaborn as sns

import re

import os

import sys

import ast

plt.style.use('fivethirtyeight')

cp = sns.color\_palette()

from vaderSentiment.vaderSentiment

import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()

emptyline=[]

for row in df['Text']:

vs=analyzer.polarity\_scores(row)

emptyline.append(vs)

df\_sentiments=pd.DataFrame(emptyline)

df\_sentiments.head()

df\_c = pd.concat([df.reset\_index(drop=True), df\_sentiments], axis=1)

df\_c.head(3)

df\_c['Sentiment'] = np.where(df\_c['compound'] >= 0 , 'Positive','Negative')

df\_c.head(5)

result=df\_c['Sentiment'].value\_counts()

result.plot(kind='bar', rot=0, color=['plum','cyan']);

**Output**



