## **EXPERIMENT NO. 10**

**Aim:** Develop social media text analytics models for improving existing product/service by analyzing customer’s reviews/comments.

**Theory:**

Developing social media text analytics models for improving existing products or services involves extracting insights from customer reviews and comments on social media platforms. Here's a step-by- step approach to building such models:

1. **Data Collection:**

* Gather customer reviews and comments from various social media platforms where your brand is mentioned. This can include platforms like Facebook, Twitter, Instagram, Reddit, and specialized review sites like Yelp or TripAdvisor.
* Use web scraping techniques or APIs provided by social media platforms to collect textual data from these sources.

1. **Data Preprocessing:**

* Clean the text data by removing noise such as special characters, punctuation, and

irrelevant information like URLs or usernames.

* Tokenize the text into individual words or phrases.
* Normalize the text by converting everything to lowercase and removing stop words (common words like "and", "the", "is" that don't carry significant meaning).

1. **Sentiment Analysis:**

* Conduct sentiment analysis to categorize each review or comment as positive, negative, or neutral.
* Use machine learning algorithms such as Naive Bayes, Support Vector Machines (SVM), or Recurrent Neural Networks (RNNs) to train a sentiment classifier.
* Evaluate the performance of the sentiment classifier using metrics like accuracy, precision, recall, and F1-score.

1. **Aspect-Based Sentiment Analysis:**

* Perform aspect-based sentiment analysis to identify specific aspects or features of your product or service mentioned in the reviews.
* Train a model to classify each aspect mentioned in the reviews and the sentiment associated with each aspect (e.g., performance, usability, customer service).
* Use techniques like Named Entity Recognition (NER) or dependency parsing to extract relevant aspects from the text.
* Analyze the sentiment of each aspect to understand areas of strengths and weaknesses.

1. **Topic Modeling:**

* Apply topic modeling techniques such as Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) to identify key topics or themes in the reviews.
* Discover clusters of words that frequently co-occur in the reviews, representing common topics or subjects of discussion.
* Assign each review to one or more topics based on the words and phrases it contains.

1. **Feature Extraction:**

* Extract relevant features from the text data that can provide insights into customer

preferences, concerns, or suggestions.

* Features can include word frequency, sentiment scores, topic distributions, named

entities, and syntactic patterns.

1. **Insights Generation:**

* Analyze the results of sentiment analysis, aspect-based sentiment analysis, and topic

modeling to generate actionable insights.

* Identify recurring issues or pain points mentioned by customers.
* Highlight areas of strength that customers appreciate and value.
* Prioritize improvements or changes based on the frequency and severity of issues

identified.

1. **Iterative Improvement:**

* Continuously refine and improve the text analytics models based on feedback and new data.
* Monitor changes in customer sentiment over time to assess the impact of product/service enhancements.
* Adapt strategies and approaches as customer preferences and market conditions evolve.

**Code:**

import pandas as pd

import nltk

from nltk.sentiment.vader import SentimentIntensityAnalyzer

import re

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report

nltk.download('vader\_lexicon')

sid = SentimentIntensityAnalyzer()

file\_path = '/content/twitter\_validation.csv'

df = pd.read\_csv(file\_path)

print("Dataset Overview:")

print(df.head())

def preprocess\_text(text):

    text = re.sub(r"http\S+|www\S+|@\w+", "", text)

    text = re.sub(r"[^a-zA-Z\s]", "", text)

    text = text.lower()

    return text

df['cleaned\_text'] = df['Text'].apply(preprocess\_text)

def analyze\_sentiment(text):

    sentiment\_scores = sid.polarity\_scores(text)

    if sentiment\_scores['compound'] >= 0.05:

        return "Positive"

    elif sentiment\_scores['compound'] <= -0.05:

        return "Negative"

    else:

        return "Neutral"

df['predicted\_sentiment'] = df['cleaned\_text'].apply(analyze\_sentiment)

if 'label' in df.columns:

    df['label'] = df['label'].map({0: "Negative", 1: "Positive"})

    print("Classification Report:")

    print(classification\_report(df['label'], df['predicted\_sentiment']))

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    df['cleaned\_text'], df['predicted\_sentiment'], test\_size=0.2, random\_state=42

)

print("Sample Results:")

print(df[['Text', 'cleaned\_text', 'predicted\_sentiment']].head()) # Changed 'review' to 'text' here as well

else:

print("Sentiment: Neutral")

**Output:**



**Conclusion:**

Here we can scrape the product data and identify its negative reviews, also we have analyzed negative tweets for improvement.