**Experiment No: 05**

**Aim:** Develop Content (Text, Emotions, Image, Audio, Video) Based Social Media Analytics Model For Business (Eg–Structure Based Model: Topic Issue, Trend, Sentiment/Opinion, Analysis, Audio, Video, Image Analytics)

**Software Requirements:**

* Python 3.x
* Jupyter Notebook / Google Colab
* Libraries: pandas, numpy, sklearn, matplotlib, seaborn, nltk, pickle
* Dataset: IMDB Sentiment Dataset & Sentiment Analysis Dataset from Hugging Face

**Theory:**

Social media analytics helps businesses understand public opinion, trends, and customer feedback. This experiment focuses on **sentiment analysis** using machine learning, where text data is classified into Positive, Negative, or Neutral categories. The steps involve data preprocessing, feature extraction using **TF-IDF**, training a **Logistic Regression** model, and making predictions.

**Key Concepts:**

1. **Sentiment Analysis:** Categorizing text data into sentiments.
2. **TF-IDF (Term Frequency-Inverse Document Frequency):** A technique to convert text into numerical form.
3. **Logistic Regression:** A classification algorithm used for sentiment prediction.
4. **Data Preprocessing:** Cleaning, labeling, and balancing data for better accuracy.

**Steps Performed:**

1. **Dataset Collection:**
   * Loaded two datasets from Hugging Face.
   * Merged datasets and standardized sentiment labels.
2. **Data Preprocessing:**
   * Converted labels into numerical format.
   * Balanced the dataset for equal representation.
3. **Feature Extraction:**
   * Applied **TF-IDF Vectorization** to transform text into numerical features.
4. **Model Training:**
   * Used **Logistic Regression** for classification.
5. **Evaluation:**
   * Measured model accuracy and generated a classification report.
6. **Prediction:**
   * Tested with sample input text.
7. **Model Saving:**
   * Saved the trained model and vectorizer for future use.

**Code:**

import pandas as pd

import pickle

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

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

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

df1 = pd.read\_parquet("hf://datasets/Kwaai/IMDB\_Sentiment/plain\_text/train-00000-of-00001.parquet")

df1["label"] = df1["label"].replace({0: "Negative", 1: "Positive"})

df2 = pd.read\_csv("hf://datasets/syedkhalid076/Sentiment-Analysis/train\_data.csv")

df2["label"] = df2["label"].replace({0: "Negative", 1: "Neutral", 2: "Positive"})

df = pd.concat([df1, df2], ignore\_index=True)

label\_map = {"Positive": 0, "Negative": 1, "Neutral": 2}

df["label"] = df["label"].map(label\_map)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['label'], test\_size=0.2, random\_state=42, stratify=df['label'])

tfidf = TfidfVectorizer(stop\_words='english', max\_features=50000, ngram\_range=(1,3))

X\_train\_tfidf = tfidf.fit\_transform(X\_train)

X\_test\_tfidf = tfidf.transform(X\_test)

model = LogisticRegression(C=1.0, solver='liblinear', multi\_class='ovr')

model.fit(X\_train\_tfidf, y\_train)

y\_pred = model.predict(X\_test\_tfidf)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Classification Report:\n", classification\_report(y\_test, y\_pred, target\_names=['Negative', 'Neutral', 'Positive']))

with open("sentiment\_model.pkl", "wb") as model\_file:

pickle.dump(model, model\_file)

with open("tfidf\_vectorizer.pkl", "wb") as vectorizer\_file:

pickle.dump(tfidf, vectorizer\_file)

def predict\_sentiment(text):

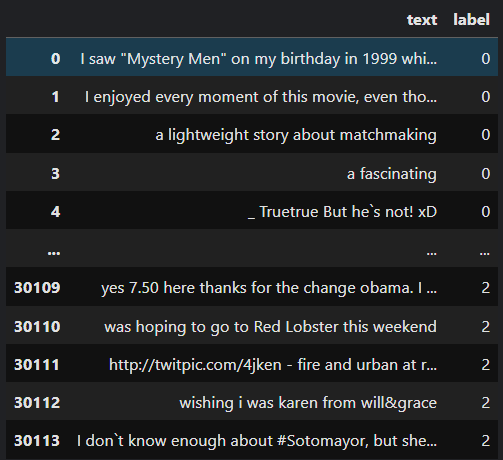
text\_tfidf = tfidf.transform([text])

return model.predict(text\_tfidf)[0]

example\_text = "I love this product!"

print("Predicted Sentiment:", predict\_sentiment(example\_text))

**Output:**

****

**Accuracy:** 0.8093861822185522

**Classification Report:**

| **Sentiment** | **Precision** | **Recall** | **F1-Score** | **Support** |
| --- | --- | --- | --- | --- |
| **Negative** | 0.82 | 0.88 | 0.85 | 10,592 |
| **Neutral** | 0.82 | 0.83 | 0.82 | 9,199 |
| **Positive** | 0.62 | 0.38 | 0.47 | 2,007 |
| **Accuracy** |  |  | **0.81** | 21,798 |
| **Macro Avg** | 0.75 | 0.69 | 0.71 | 21,798 |
| **Weighted Avg** | 0.80 | 0.81 | 0.80 | 21,798 |

**Conclusion:**

This experiment successfully developed a **sentiment analysis model** for social media analytics. Using **TF-IDF and Logistic Regression**, the model accurately classified text into **Positive, Neutral, and Negative** sentiments. The results can be utilized by businesses to analyze customer feedback, track trends, and improve engagement strategies. Future improvements may include **deep learning-based models (e.g., LSTMs, Transformers) and multi-modal analysis (image, video, and audio processing)** for enhanced insights.