

/*Practical No : 07,
Name: Dhiraj Vijay Barwal ,
Roll No : 08,
Class: TECSD,
Batch: T1 */

PREDICT SENTIMENT FROM SOCIAL MEDIA OR CUSTOMER REVIEW OF PRODUCT

```
import pandas as pd
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, confusion_matrix

# Load the dataset
file_path = '/mnt/data/reviews.csv' # Update this with the correct file path
df = pd.read_csv(file_path)

# Inspect the dataset (Ensure it has 'Review' and 'Sentiment' columns)
print("First few rows of the dataset:\n", df.head())

# Preprocessing: Remove null values and unnecessary columns
df = df.dropna(subset=['Review', 'Sentiment'])
df['Sentiment'] = df['Sentiment'].map({'Positive': 1, 'Negative': 0}) # Encode sentiment

# Define features (X) and target (y)
X = df['Review'] # Text data
y = df['Sentiment'] # Target labels

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Convert text data to numerical using TF-IDF Vectorizer
tfidf = TfidfVectorizer(max_features=5000, stop_words='english')
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

# Initialize and train a Logistic Regression model
model = LogisticRegression(random_state=42, class_weight='balanced')
model.fit(X_train_tfidf, y_train)

# Predict sentiment for the test data
y_pred = model.predict(X_test_tfidf)

# Evaluation Metrics
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
print("Accuracy:", accuracy)
print("\nClassification Report:\n", classification_rep)
```

```
print("\nConfusion Matrix:\n", conf_matrix)

# Predict sentiment for a new review
new_review = ["The product is amazing and exceeded my expectations!"] # Replace with a sample review
new_review_tfidf = tfidf.transform(new_review)
sentiment_prediction = model.predict(new_review_tfidf)
print(f"\nThe sentiment is predicted as: {'Positive' if sentiment_prediction[0] == 1 else 'Negative'}")
```

OUTPUT:

First few rows of the dataset:

	Review	Sentiment
0	I love this product! It's amazing.	Positive
1	The product didn't work as advertised.	Negative
2	Great quality and fast shipping!	Positive
3	Not worth the money.	Negative
4	Excellent purchase experience!	Positive