https://exitexamappbonus-kspxbogu9ugfurjzw38l8z.streamlit.app

Share

TF-IDF Sentiment Prediction & Word Importance

Enter your product review:

This product exceeded my expectations! The quality is fantastic, and I would definitely buy it again

Predict Sentiment

Sentiment: Positive

Top positive words influencing prediction:

fantastic, definitely, exceeded, quality, product, buy, expectations

Top negative words influencing prediction:

expectations, buy, product, quality, exceeded, definitely, fantastic

Manage app

https://exitexamappbonus-kspxbogu9ugfurjzw38l8z.streamlit.app

Share

TF-IDF Sentiment Prediction & Word Importance

Enter your product review:

I am very disappointed with this purchase. The item was poorly made and did not work as advertised.

Predict Sentiment

Sentiment: Negative

Top positive words influencing prediction:

work, purchase, advertised, item, did, poorly, disappointed

Top negative words influencing prediction:

disappointed, poorly, did, item, advertised, purchase, work

Manage app

AIML_Exit_Exam_Bonus.ipynb

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all

RAM Disk

Files

..
drive
sample_data
lr_model.pkl
tfidf.pkl

Disk 68.01 GB available

[1] 1m

```python
import pandas as pd
import string
import re
import pickle
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression

# Load and preprocess data
df = pd.read_csv("/content/drive/MyDrive/exit_exam/Reviews.csv")
df = df[['Text', 'Score']].dropna()
df = df[df['Score'] != 3]
df['Sentiment'] = df['Score'].apply(lambda x: 'Positive' if x >= 4 else 'Negative')

def clean_text(text):
    text = text.lower()
    text = re.sub(f"[{string.punctuation}]", " ", text)
    text = re.sub(r"\d+", "", text)
    return text

df['Clean_Text'] = df['Text'].apply(clean_text)

X = df['Clean_Text']
y = df['Sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 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)

# Logistic Regression model
lr = LogisticRegression(max_iter=1000)
lr.fit(X_train_tfidf, y_train)

# Save model and vectorizer
with open("tfidf.pkl", "wb") as f:
    pickle.dump(tfidf, f)
with open("lr_model.pkl", "wb") as f:
    pickle.dump(lr, f)
```

Providing word importance alongside sentiment predictions offers actionable insights for the marketing team. Instead of just seeing a sentiment score, marketers can identify which specific words or phrases most strongly influence customer perception—whether positive or negative.

Variables Terminal

4:09 PM Python 3

CAD/INR -0.77%

Search

ENG IN 16:18 15-10-2025

Providing word importance alongside sentiment predictions offers actionable insights for the marketing team. Instead of just seeing a sentiment score, marketers can identify which specific words or phrases most strongly influence customer perception—whether positive or negative.
