SchrodingersCats

Intel Products

SENTIMENT ANALYSIS

from Online Reviews

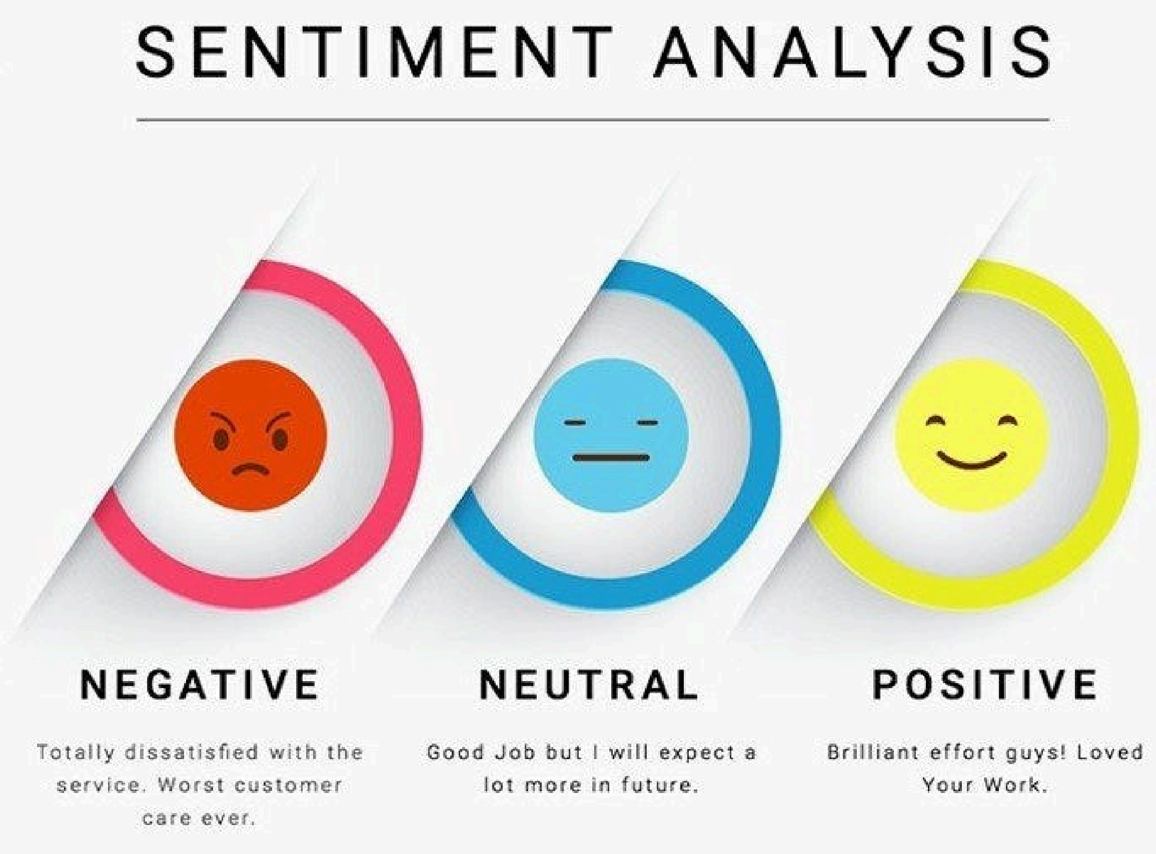
SchrodingersCats

INTRODUCTION

Understanding Consumer Feedback Through Data

In today's digital age, consumer reviews have become a goldmine of insights. Analyzing these reviews allows us to understand public sentiment and improve product offerings.

SENTIMENT ANALYSIS



SchrodingersCats

SENTIMENT ANALYSIS

What is Sentiment Analysis?

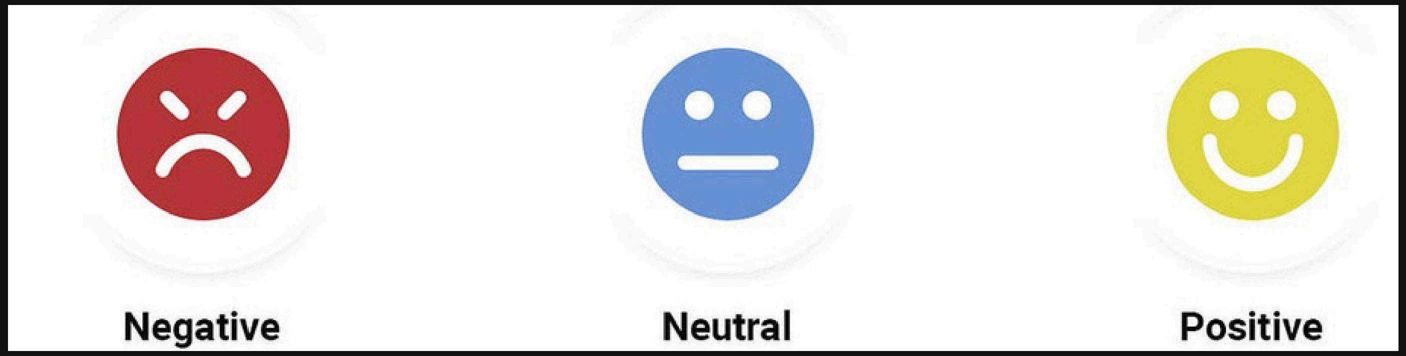
- Definition: Sentiment analysis, also known as opinion mining, is the process of determining the emotional tone behind a series of words.
- Importance: Helps in understanding the opinions expressed in reviews, tweets, and other user-generated content.

Why Focus on Intel Products?

- Market Position: Intel is a leader in the semiconductor industry, known for its innovation in processors, memory, and other technology solutions.
- Consumer Impact: As a major player, understanding customer feedback on Intel products can guide future product development and marketing strategies.

SchrodingersCats

OBJECTIVES



Analyze Consumer Sentiment:

Identify positive, negative, and neutral sentiments from online reviews of Intel products.

Discover Key Trends

Highlight common themes and issues raised by consumers.

Provide Insights

Offer actionable recommendations for Intel to enhance customer satisfaction and product quality.

SchrodingersCats

SOLUTION

1. DATA COLLECTION:

Identify Review Sources

Gather reviews from major online retail platforms such as Amazon, Best Buy, Newegg, and dedicated tech review sites.

AI-Powered Web Scraping

Use AI-based web scraping tools to automate the extraction of review data including review text and ratings.

Data Aggregation:

Collect a diverse dataset covering various Intel products like processors, graphics cards, and other hardware components.

2. PREPROCESSING:

Data Cleaning:

Use AI-powered data cleaning tools to remove duplicates, irrelevant information, and any non-English reviews to ensure data quality.

Text Normalization:

Employ NLP libraries (e.to convert text to lowercase, remove punctuation, stop words, and perform stemming/lemmatization to standardize the text data.

Tokenization:

Utilize AI-based tokenization methods for splitting the review text into individual words or tokens.

SchrodingersCats

SOLUTION

3. SENTIMENT ANALYSIS TECHNIQUES:

Pretrained Sentiment Models:

Leverage pretrained AI sentiment analysis models for initial sentiment scoring of each review.

Custom Machine Learning Models:

Train custom machine learning classifiers (e.g., using Scikit-learn, TensorFlow) on labeled sentiment data to predict the sentiment of new reviews.

Deep Learning Models:

Implement advanced models like LSTM or BERT using AI frameworks for more accurate sentiment classification, especially for longer and more complex review texts.

4. INTERPRETATION AND INSIGHTS:

Identify Key Themes:

Utilize AI text summarization tools to extract and summarize common themes and recurring issues from the reviews.

Highlight Strengths and Weaknesses:

Use AI-driven analytics to determine the most praised features and the most criticized issues with Intel products.

Recommendations

Provide actionable recommendations for product improvement and customer engagement based on AI-driven insights.

SchrodingersCats

ANALYSIS RESULTS

1.

OVERALL SENTIMENT DISTRIBUTION:

The sentiment analysis categorized reviews into positive, negative, and neutral. This helps us gauge the general customer sentiment toward Intel products.

Pie Chart: Displaying the percentage distribution of positive, negative, and neutral reviews.

2.

SENTIMENT TRENDS OVER TIME:

Tracking sentiment over time shows how customer opinions have changed. It highlights key periods, such as product launches or issues.

Line Graph: Sentiment scores plotted over time (monthly or quarterly).

SchrodingersCats

ANALYSIS RESULTS

3.

SENTIMENT BY PRODUCT CATEGORY:

Breaking down sentiment by product category (CPUs, GPUs, laptops) helps identify which products are well-received and which need improvement.

Bar Chart: Comparing sentiment scores across different product categories.

4.

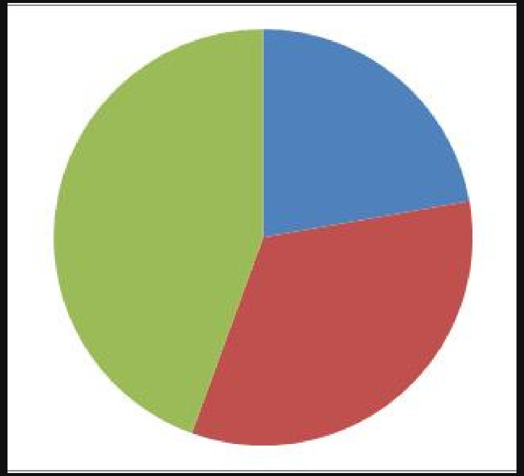
FEATURE-SPECIFIC SENTIMENT ANALYSIS:

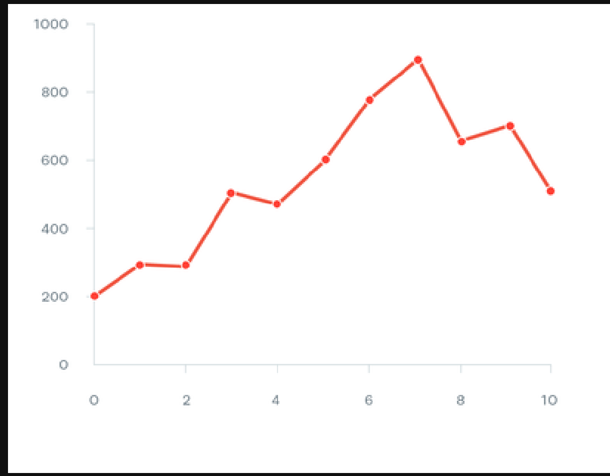
Analyzing sentiment for specific product features (performance, design, price) provides targeted insights into what customers like or dislike.

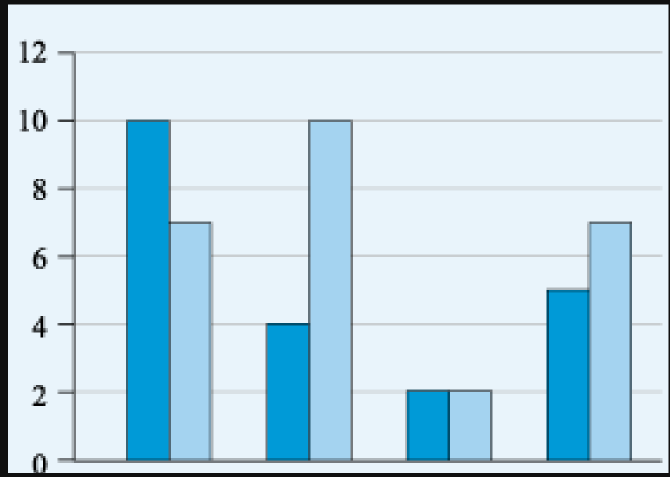
Stacked Bar Chart: Showing sentiment distribution for specific features.

ANALYSIS RESULTS

VISUAL EXAMPLES:







Pie Chart Example:

Positive: 60%
Neutral: 25%
Negative: 15%

Line Graph Example:

X-axis: Time (months)
Y-axis: Sentiment Score

Pie Chart Example:

X-axis: Features (Performance, Design, Price, etc.)
Y-axis: Sentiment Distribution

SchrodingersCats

CODE

```
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
import nltk

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
Step 3: Load and Preprocess Data
# Load data
data = pd.read_csv('intel_reviews.csv')
# Preprocessing function
def preprocess_text(text):
# Lowercase
 text = text.lower()
 # Remove punctuation
 text = re.sub(r'[^\w\s]', '', text)
 # Tokenize
tokens = word_tokenize(text)
 # Remove stop words
 stop_words = set(stopwords.words('english'))
tokens = [word for word in tokens if word not in stop_words]
# Lemmatize
lemmatizer = WordNetLemmatizer()
tokens = [lemmatizer.lemmatize(word) for word in tokens]
return ' '.join(tokens)
# Apply preprocessing
data['cleaned_review'] = data['review'].apply(preprocess_text)
Step 4: Feature Extraction
# Split data into training and testing sets
```

SchrodingersCats

CODE

```
X_train, X_test, y_train, y_test = train_test_split(data['cleaned_review'], data['sentiment'], test_size=0.2, random_state=42)

# Convert text data to TF-IDF features
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
Step 5: Train Naive Bayes Classifier
# Train Naive Bayes classifier
nb_classifier = MultinomialNB()
nb_classifier.fit(X_train_tfidf, y_train)
# Predict on test data
y_pred = nb_classifier.predict(X_test_tfidf)
# Evaluate model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(conf_matrix)
print('Classification Report:')
print(class_report)
Data Collection Method Using Web Scraping (OPTIONAL)
from bs4 import BeautifulSoup
import requests
# Example scraping function for a single product page
def scrape_reviews(url):
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')
reviews = []
for review in soup.find_all('div', class_='review'):
text = review.find('p').get_text()
sentiment = 'positive' if 'positive' in review['class'] else 'negative' reviews.append({'review': text, 'sentiment': sentiment})
return reviews
url = 'http://example.com/product-reviews'
reviews = scrape_reviews(url)
reviews_df = pd.DataFrame(reviews)
reviews_df.to_csv('intel_reviews.csv', index=False)
```

TEAM DETAILS

INSTITUTION NAME

B.S.Abdur Rahman Crescent Institute of Science & Technology

FACULTY MENTOR NAME:

Dr. V. Jean Shilpa

TEAM NAME:

SchrodingersCats

TEAM LEADER NAME:

Syed Wazafeer SN

TEAM MEMBER NAME:

Subash

SchrodingersCats

THANK YOU!