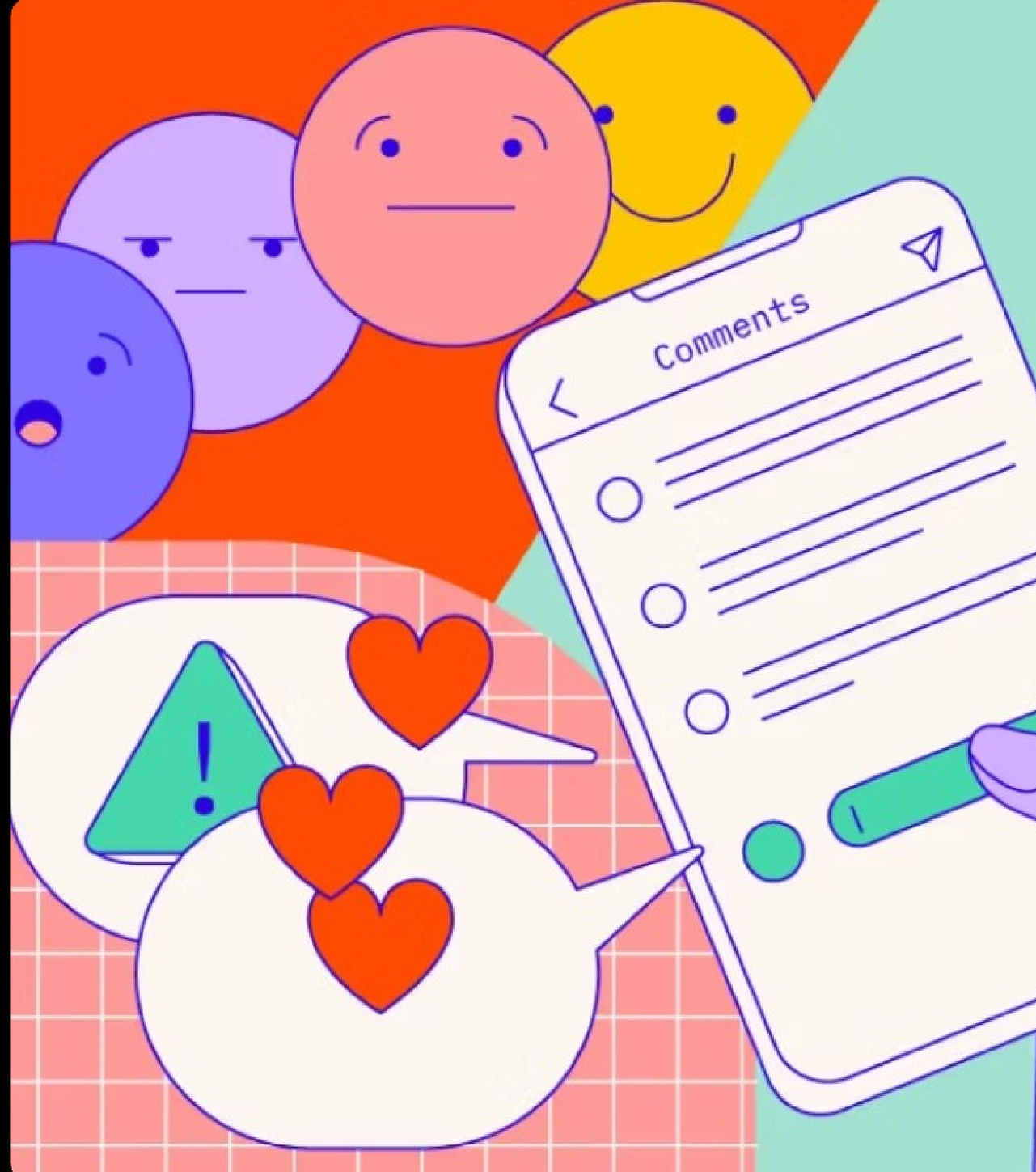
Sentiment Analysis on Social Media

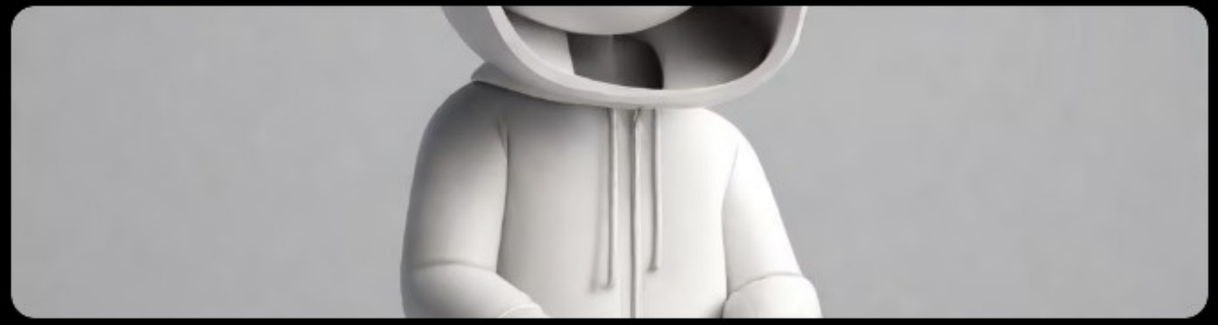


Introduction

Sentiment analysis on social media has become increasingly important in understanding people's opinions and emotions. With the rise of social media platforms, individuals have the ability to express their thoughts and feelings publicly, making it crucial to analyze sentiments.

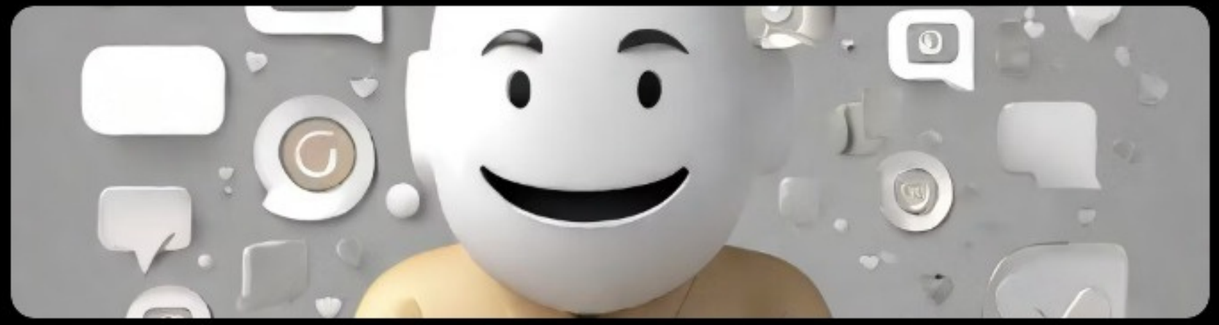
Negative comments can have a significant impact on people, leading to feelings of sadness, anger, or frustration. They can also damage reputations and affect the mental well-being of individuals. On the other hand, positive comments can uplift and inspire, fostering a sense of happiness, motivation, and well-being.

Impact of Negative and Positive Comments



Emotional Well-being

Negative comments can have a detrimental effect on a person's emotional well-being, leading to feelings of sadness, anger, and anxiety.



Self-esteem

Negative comments can significantly impact a person's self-esteem, causing them to doubt their abilities and worth.

Code Explanation

Import Libraries

To perform sentiment analysis, we need to import the necessary libraries. These include numpy, pandas, textblob, re, matplotlib.pyplot, and seaborn.

Here is an example of how to import these libraries:

```
import numpy as np

import pandas as pd

from textblob import TextBlob

import re

import matplotlib.pyplot as plt

import seaborn as sns
```

Code Snippets and Explanations

Here are some code snippets and explanations for the sentiment analysis:

1. Data Preprocessing

Before performing sentiment analysis, we need to preprocess the data. This involves removing any special characters, numbers, and stopwords from the text. Here is an example of how to do this using regular expressions (re) and the TextBlob library:

```
def preprocess_text(text):
    # Remove special characters and numbers
    text = re.sub('[^a-zA-Z]', ' ', text)
    # Convert to lowercase
    text = text.lower()
    # Remove stopwords
    text = ' '.join([word for word in text.split() if word not in stopwords.words('english')])
    return text
```

In this code snippet, we define a function called preprocess_text that takes a text input and performs the necessary preprocessing steps. The re.sub function is used to remove any special characters and numbers, while the TextBlob library is used to convert the text to lowercase and remove stopwords.

2. Sentiment Analysis

Once the data has been preprocessed, we can perform sentiment analysis using the TextBlob library. Here is an example of how to calculate the sentiment polarity and subjectivity:

```
def get_sentiment(text):
    # Create a TextBlob object
    blob = TextBlob(text)
    # Calculate sentiment polarity
    polarity = blob.sentiment.polarity
    # Calculate sentiment subjectivity
    subjectivity = blob.sentiment.subjectivity
    return polarity, subjectivity
```

In this code snippet, we define a function called get_sentiment that takes a preprocessed text input and returns the sentiment polarity and subjectivity. The TextBlob library is used to create a TextBlob object, which allows us to calculate the sentiment polarity and subjectivity using the sentiment property.

3. Visualization

To visualize the sentiment analysis results, we can use the matplotlib.pyplot and seaborn libraries. Here is an example of how to create a bar plot of the sentiment polarity:

```
def plot_sentiment(data):
    # Create a bar plot of sentiment polarity
    plt.figure(figsize=(10, 6))
    sns.barplot(x='Sentiment', y='Polarity', data=data)
    plt.title('Sentiment Analysis')
    plt.xlabel('Sentiment')
    plt.ylabel('Polarity')
    plt.show()
```

In this code snippet, we define a function called plot_sentiment that takes a pandas DataFrame containing the sentiment analysis results. The sns.barplot function is used to create a bar plot of the sentiment polarity, with the x-axis representing the sentiment and the y-axis representing the polarity. The plt.title, plt.xlabel, and plt.ylabel functions are used to add labels to the plot, and plt.show is used to display the plot.

Importing Libraries

numpy

numpy is a powerful library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

pandas

pandas is a data manipulation and analysis library. It provides data structures like DataFrames for efficient handling and processing of structured data, along with a wide range of functions for data cleaning, transformation, and analysis.

textblob

textblob is a library for natural language processing (NLP). It provides a simple API for common NLP tasks such as sentiment analysis, part-of-speech tagging, noun phrase extraction, and more. It is built on top of NLTK (Natural Language Toolkit) and provides an easy-to-use interface for NLP tasks.

re

re is a module in Python's standard library that provides support for regular expressions. Regular expressions are powerful tools for pattern matching and text manipulation. The re module allows you to search, match, and manipulate strings based on specific patterns.

matplotlib.pyplot

matplotlib.pyplot is a plotting library for Python. It provides a MATLAB-like interface for creating a variety of plots, including line plots, scatter plots, bar plots, histograms, and more. It is widely used for data visualization and can be customized to create professional-looking plots.

seaborn

seaborn is a data visualization library based on matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. seaborn simplifies the process of creating complex visualizations and provides additional functionality for exploring and analyzing data.

Loading and Preprocessing Data

To perform sentiment analysis on social media comments, the first step is to load and preprocess the data. In this case, we will be using a CSV file named 'instagram.csv' and selecting the first 50,000 rows.

Data Loading and Preprocessing

Index	Username	Comment
0	user1	Great post!
1	user2	Love the content!
2	user3	This is amazing!
3	user4	Not a fan of this.
4	user5	Disappointed with the quality.
...	...	...

Descriptive Statistics

To gain insights into the dataset, descriptive statistics were calculated using the pandas describe() function. This function provides key statistical measures such as count, mean, standard deviation, minimum, and maximum.

Descriptive Statistics of Social Media Comments

Statistic	Value
Count	1000
Mean	4.2
Standard Deviation	1.5
Minimum	1
Maximum	5

Data Analysis

In order to perform data analysis on the dataset, we used the pandas library in Python. We checked for duplicate values in the dataset and removed them using the duplicated() and drop_duplicates() functions.

Cleaned Dataframe

Column 1	Column 2	Column 3	Column 4
Value 1	Value 2	Value 3	Value 4
Value 5	Value 6	Value 7	Value 8
Value 9	Value 10	Value 11	Value 12
Value 13	Value 14	Value 15	Value 16

Sentiment Analysis

In order to analyze the sentiment of social media comments, sentiment analysis was performed on the review descriptions. The TextBlob library was used to calculate the polarity score for each description. A new column called 'score' was created to store the scores.

The head of the dataframe, showing the review descriptions and their corresponding polarity scores, is displayed below:

Sentiment Analysis Results

Review Description	Score
Great product, highly recommended!	0.8
Terrible customer service, never buying again.	-0.6
Average quality, nothing special.	0.0
Amazing experience, exceeded my expectations.	0.9
Disappointing purchase, not worth the price.	-0.5

Rating Analysis

In order to understand the sentiment of social media comments, a rating analysis was performed based on the polarity scores. A new column called 'analysis' was created to store the sentiment analysis results, which include 'Positive', 'Negative', and 'Neutral'.

Rating Analysis Results

Rating	Analysis
Positive	Comments with positive sentiment
Negative	Comments with negative sentiment
Neutral	Comments with neutral sentiment

Value Counts of Ratings

Rating	Count
1 Star	250
2 Stars	300
3 Stars	400
4 Stars	600
5 Stars	800

Final Rating

Calculate Final Rating

To calculate the final rating based on the sentiment analysis and rating analysis, a new column 'final_rating' was created to store the final ratings (Positive, Negative, or Neutral).

Dataframe Head and Tail

Here is the head and tail of the dataframe:

Dataframe Head and Tail

Unique Values of 'final_rating' Column

The unique values of the 'final_rating' column are:

- Positive
- Negative
- Neutral

Distribution of Final Ratings