Факультет комп’ютерних наук та кібернетики

Звіт
“Social Media Sentiment Analysis”

Виконав:
Студент 4 курсу
Факультету комп’ютерних наук та кібернетики
групи ТТП-41
Корнієнко Олександр Віталійович

Київ – 2025

Зміст

Мета та опис проєкту

У межах цього проєкту ми провели аналіз настроїв (Sentiment Analysis) публікацій у соціальних мережах, зокрема твітів. Головною метою було визначити, чи має кожне повідомлення позитивний, негативний або нейтральний відтінок, а також порівняти отримані (передбачені) результати з наявною розміткою (реальною категорією).

Опис даних

Набір даних: Twitter_Data.csv, що містить текст твітів та відмітку їхнього реального настрою.

Обсяг вибірки: для прикладу взято перші 500 записів (рядків) з повного датасету.

Підготовка середовища, Імпортовано необхідні бібліотеки

```
[85]  1 import pandas as pd
      2
      3 import nltk
      4
      5 from nltk.sentiment.vader import SentimentIntensityAnalyzer
      6 from nltk.corpus import stopwords
      7 from nltk.tokenize import word_tokenize
      8 from nltk.stem import WordNetLemmatizer
      9
     10 nltk.download('all')
```

```
[86]  1 df = pd.read_csv('./sample_data/Twitter_Data.csv')
      2 df.head()
```

	clean_text	category
0	when modi promised “minimum government maximum...	-1.0
1	talk all the nonsense and continue all the dra...	0.0
2	what did just say vote for modi welcome bjp t...	1.0
3	asking his supporters prefix chowkidar their n...	1.0
4	answer who among these the most powerful world...	1.0

```
[88]  1 df = df.head(500)
```

Попередня обробка тексту

Функція preprocess_text(text) включає кілька етапів.

Перевірка типу вхідних даних: якщо text не є рядком, повертається пустий рядок. Переведення в нижній регістр. Поділ тексту на окремі слова. Видалення стопслів усунення поширених, але неінформативних слів, наприклад "the", "and", "or" тощо. Приведення слів до їхньої базової форми. (Об’єднання слів у фінальний текст (cleanedText). Після цього кроку колонка clean_text перетворюється в нову оброблену колонку cleanedText.

```python
def preprocess_text(text):
    if not isinstance(text, str):
      return ""

    tokens = word_tokenize(text.lower())
    filtered_tokens = [token for token in tokens if token not in stopwords.words('english')]

    lemmatizer = WordNetLemmatizer()
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in filtered_tokens]

    processed_text = ' '.join(lemmatized_tokens)

    return processed_text

df['cleanedText'] = df['clean_text'].apply(preprocess_text)
df.head()
```

	clean_text	category	cleanedText
0	when modi promised “minimum government maximum...	-1.0	modi promised “ minimum government maximum gov...
1	talk all the nonsense and continue all the dra...	0.0	talk nonsense continue drama vote modi
2	what did just say vote for modi welcome bjp t...	1.0	say vote modi welcome bjp told rahul main camp...
3	asking his supporters prefix chowkidar their n...	1.0	asking supporter prefix chowkidar name modi gr...
4	answer who among these the most powerful world...	1.0	answer among powerful world leader today trump...

Аналіз настроїв із VADER

Ініціалізація аналізатора настроїв:

```python
!pip install vaderSentiment
```

Визначення функції get_sentiment(text), яка перевіряє, чи вхідний текст дійсний і не порожній. Обчислює compound-оцінку настрою за допомогою polarity_scores. Приводить compound до цілочисельного значення настрою.

Якщо compound >= 0.05, текст вважається позитивним.

Якщо compound <= -0.05, текст вважається негативним (-1).

Інакше – нейтральним (0).

Створюється нова колонка sentiment, де для кожного запису обчислюється передбачений настрій.

```
[91]  1 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
      2
      3 analyzer = SentimentIntensityAnalyzer()
      4
      5 def get_sentiment(text):
      6     if not isinstance(text, str) or not text.strip():
      7         return 0
      8
      9     scores = analyzer.polarity_scores(text)
     10
     11     if scores['compound'] >= 0.05:
     12         return 1
     13     elif scores['compound'] <= -0.05:
     14         return -1
     15     else:
     16         return 0
     17
     18 df['sentiment'] = df['cleanedText'].apply(get_sentiment)
     19 df
```

	clean_text	category	cleanedText	sentiment
0	when modi promised “minimum government maximum...	-1.0	modi promised “ minimum government maximum gov...	1
1	talk all the nonsense and continue all the dra...	0.0	talk nonsense continue drama vote modi	-1
2	what did just say vote for modi welcome bjp t...	1.0	say vote modi welcome bjp told rahul main camp...	1
3	asking his supporters prefix chowkidar their n...	1.0	asking supporter prefix chowkidar name modi gr...	1
4	answer who among these the most powerful world...	1.0	answer among powerful world leader today trump...	1
...	...	...	...	...
495	subbu with you swamy right thinks has right or...	1.0	subbu swamy right think right order get work d...	1
496	some ppl gone crazy after modi came	-1.0	ppl gone crazy modi came	-1
497	why modi have not held single press conference...	-1.0	modi held single press conference ' speak answ...	0
498	again agenda only defeat bjp could have told t...	1.0	agenda defeat bjp could told development agend...	-1
499	drswamys timesnow last year debate nearly mill...	1.0	drswamys timesnow last year debate nearly mill...	0

500 rows × 4 columns

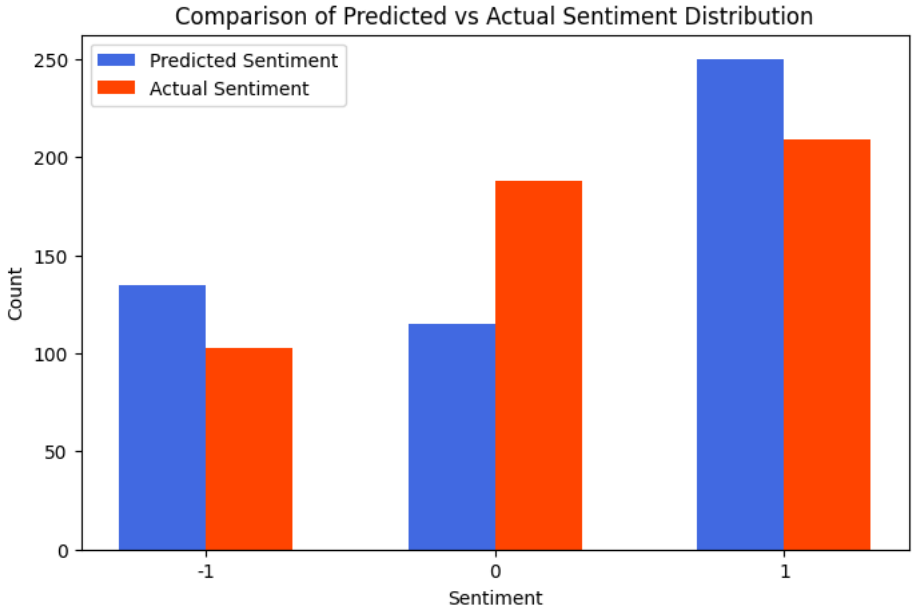
Порівняння передбаченого та реального настрою

Після визначення передбаченого настрою було виконано порівняння з реальною категорією настрою. Визначаються усі унікальні значення емоцій у двох стовпцях ('sentiment' та 'category'). Підраховується кількість позитивних, негативних та нейтральних записів (як для передбачених, так і для реальних настроїв). Для наочності створюється стовпчикова діаграма, де порівнюються категорії «Predicted Sentiment» та «Actual Sentiment».

Вісь X: можлива категорія настрою (-1, 0, 1).

Вісь Y: кількість записів для кожної категорії.

```python
import numpy as np
import matplotlib.pyplot as plt

all_sentiments = sorted(set(df['sentiment'].unique()).union(set(df['category'].unique())))

sentiment_counts = df['sentiment'].value_counts().reindex(all_sentiments, fill_value=0)
real_sentiment_counts = df['category'].value_counts().reindex(all_sentiments, fill_value=0)

x = np.arange(len(all_sentiments))
width = 0.3

plt.figure(figsize=(8, 5))
plt.bar(x - width/2, sentiment_counts.values, width=width, label="Predicted Sentiment", color='royalblue')
plt.bar(x + width/2, real_sentiment_counts.values, width=width, label="Actual Sentiment", color='orangered')

plt.xticks(x, all_sentiments)
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.title("Comparison of Predicted vs Actual Sentiment Distribution")
plt.legend()

plt.show()
```



Оцінювання точності

Для обчислення точності використовується формула

```python
correct_predictions = (df['sentiment'] == df['category']).sum()
total_predictions = len(df)
accuracy = (correct_predictions / total_predictions) * 100

print("Accuracy:", f"{accuracy:.2f}%")
```

```
Accuracy: 56.00%
```