```
pip install textblob
```

```
Requirement already satisfied: textblob in
/usr/local/lib/python3.12/dist-packages (0.19.0)
Requirement already satisfied: nltk>=3.9 in
/usr/local/lib/python3.12/dist-packages (from textblob) (3.9.1)
Requirement already satisfied: click in
/usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob)
(8.2.1)
Requirement already satisfied: joblib in
/usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob)
(1.5.1)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.12/dist-packages (from nltk>=3.9->textblob)
(2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-
packages (from nltk>=3.9->textblob) (4.67.1)
```

```
from textblob import TextBlob
# text -> it shall rain today
text = input("Enter a sentence for sentiment analysis : - ")
print("analysis done")
tb_analyse = TextBlob(text) # creates a special TextBlob object that
can analyze and process the text, Whatever TextBlob(text) returns (a
TextBlob object) is stored inside the variable tb_analyse.
print(tb_analyse.sentiment) # print the emotional tone
print('Polarity of the statement : ', tb_analyse.sentiment.polarity) #
weather person is happy or not
print('Subjective of the statement',
tb_analyse.sentiment.subjectivity) # practical things
```

```
Enter a sentence for sentiment analysis : - hi
analysis done
Sentiment(polarity=0.0, subjectivity=0.0)
Polarity of the statement :  0.0
Subjective of the statement 0.0
```

```
# lexix
tb = TextBlob("I love Python, but bugs are terrible.")
print(tb.sentiment)
# subjective -- opinion based, person's opinion
# polarity -- feelings
# Lexicon values = predefined sentiment scores for words
(positive/negative strength + subjectivity), which are used by tools
like TextBlob
```

```
Sentiment(polarity=-0.25, subjectivity=0.8)
```

```
import pickle
```

```
with open("/content/stored_data (5).pkl", "rb") as f:
    reviews = pickle.load(f)

import os
print(os.path.getsize('/content/stored_data (5).pkl'))
```

```
38120686
```

```
reviews.keys() # dict {keys : values}
```

```
dict_keys(['Endgame', 'Forest_gump', 'Johnwick', 'Joker', 'Morbius',
'Pulpfiction', 'Spiderman', 'Avengers', 'Darkknight', 'Ragnarok'])
```

```
# to clear output
from IPython.display import clear_output
import random # in above it will not show analysis done

print(reviews.keys())
movie_name = input
```

```
dict_keys(['Endgame', 'Forest_gump', 'Johnwick', 'Joker', 'Morbius',
'Pulpfiction', 'Spiderman', 'Avengers', 'Darkknight', 'Ragnarok'])
```

```
print(reviews.keys())
movie_name = input('Enter the movie you want to analyse : - ') # Asks
the user to enter the name of a movie,The input is stored in the
variable movie_name.
if movie_name in reviews.keys(): # to check movie is there in
reviews,keys
  clear_output() # This clears the output screen to avoid clutter
  num = len(reviews[movie_name]) # Assign num the number of reviews,
count reviews
  print(f'The movie has {num} reviews') # This prints the total number
of reviews available for the movie.
  pos,neg,neu = 0,0,0 # initial count
  for i in reviews[movie_name]: # loop for the movie
    tb_analyse= TextBlob(i[0]) # means we are taking the review text,
creates a TextBlob object which can analyze the text for sentiment.
    if tb_analyse.sentiment.polarity > 0.05:
      pos +=1
    elif tb_analyse.sentiment.polarity < -0.05:
        neg +=1
    else:
        neu +=1
  print(f'Positive reviews percentage: {pos*100/num:.2f}%, Negative
Reviews Percentage: {neg*100/num:.2f}%, Neutral Reviews Percentage:
{neu*100/num:.2f}%') # Corrected f-string and added formatting
else:
  print('Movie not in list')
```

```
The movie has 11357 reviews
Positive reviews percentage: 73.92%, Negative Reviews Percentage:
11.21%, Neutral Reviews Percentage: 14.87%

tb_analyse.sentiment.polarity

0.4976190476190477

tb_analyse.sentiment.subjectivity

0.6214285714285713
```