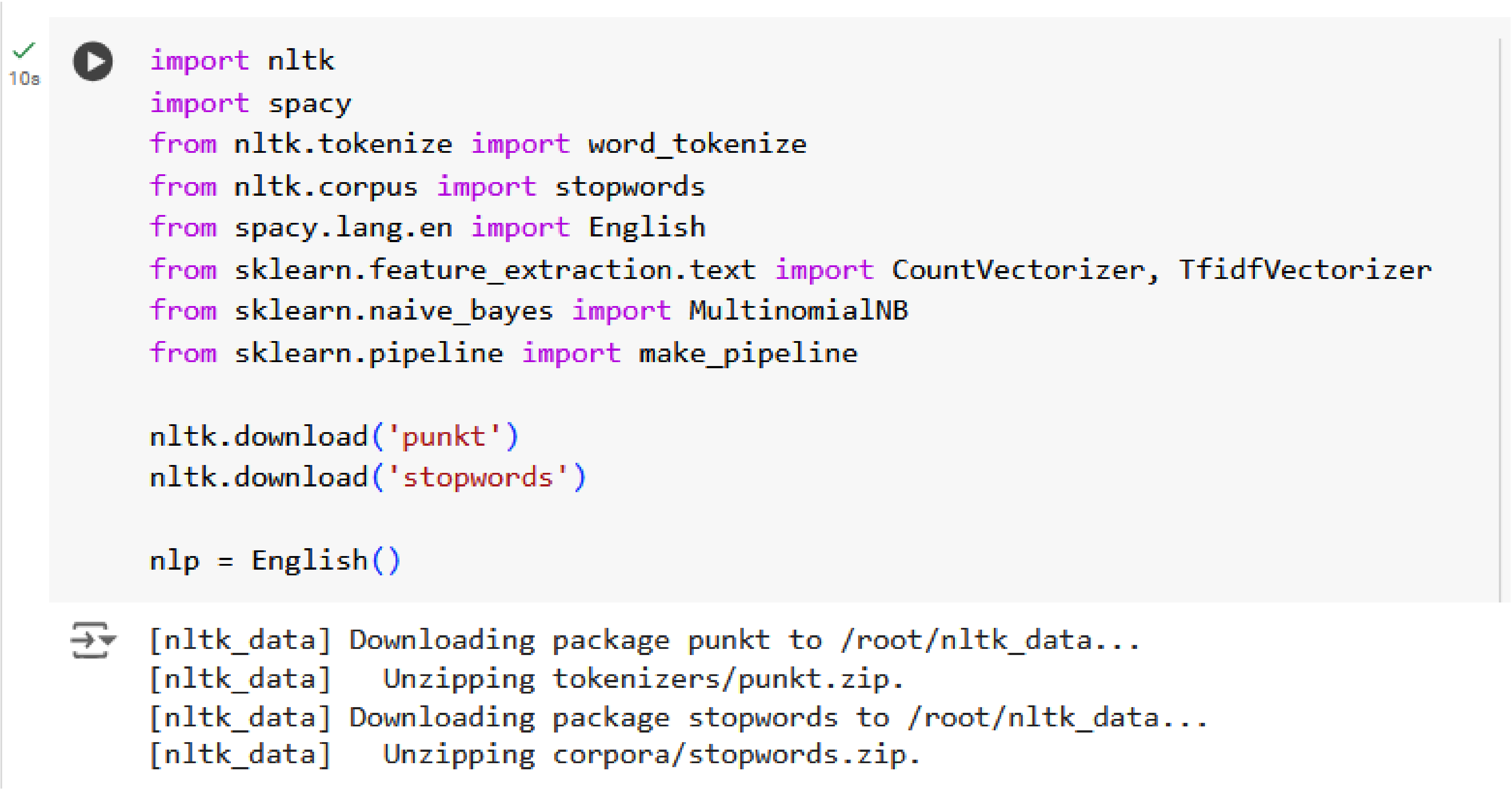
Luis Anton P. Imperial
BCS32

S-CSIS312
Natural Language Processing

Thursday, September 5, 2024
Session 1 Mini-Activity

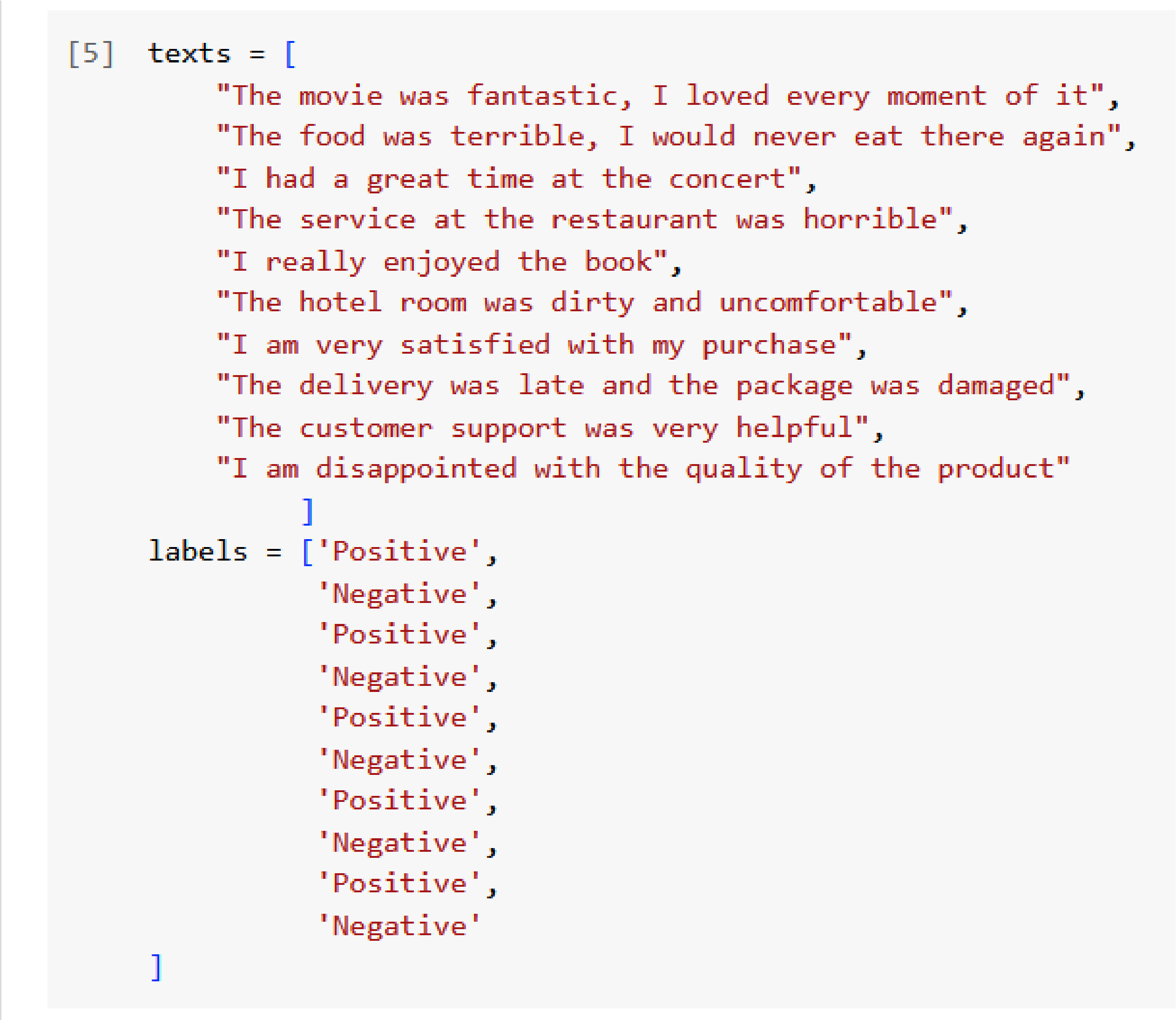
We understood the differences between different feature extraction functions available in the Python machine learning package *scikit-learn*. There are multiple vectorizers available, two of which are CountVectorizer and TfidfVectorizer.

Preparing the Packages and Dataset

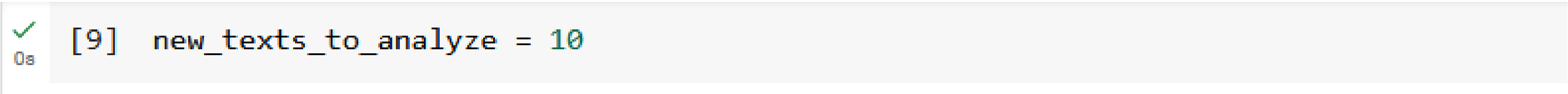


```
import nltk
import spacy
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from spacy.lang.en import English
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline

nltk.download('punkt')
nltk.download('stopwords')

nlp = English()
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```



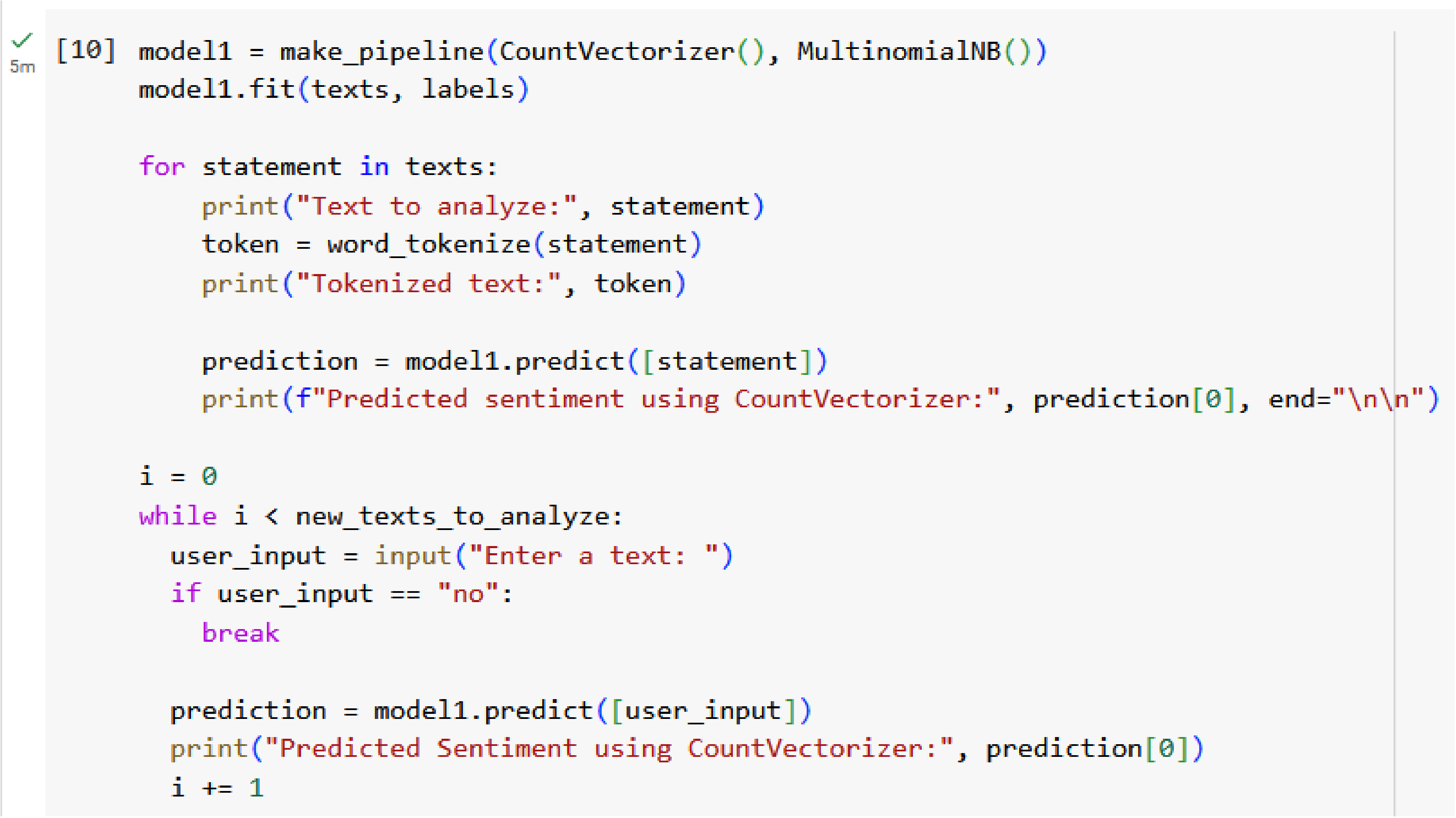
```
[5]  texts = [
         "The movie was fantastic, I loved every moment of it",
         "The food was terrible, I would never eat there again",
         "I had a great time at the concert",
         "The service at the restaurant was horrible",
         "I really enjoyed the book",
         "The hotel room was dirty and uncomfortable",
         "I am very satisfied with my purchase",
         "The delivery was late and the package was damaged",
         "The customer support was very helpful",
         "I am disappointed with the quality of the product"
              ]
     labels = ['Positive',
               'Negative',
               'Positive',
               'Negative',
               'Positive',
               'Negative',
               'Positive',
               'Negative',
               'Positive',
               'Negative'
     ]
```



```
[9]  new_texts_to_analyze = 10
```

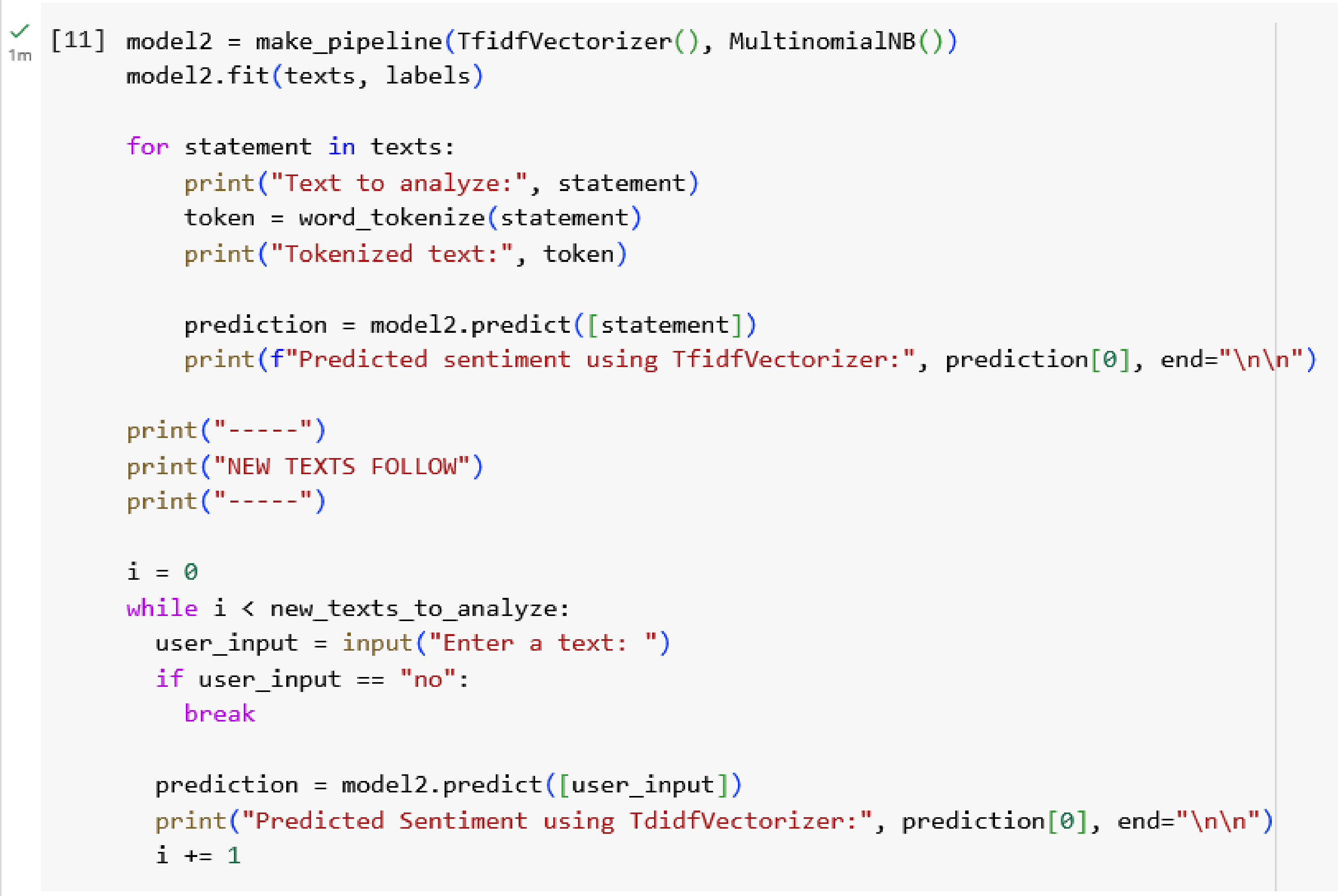
Initiating the Vectorizer

With CountVectorizer



```
model1 = make_pipeline(CountVectorizer(), MultinomialNB())
model1.fit(texts, labels)

for statement in texts:
    print("Text to analyze:", statement)
    token = word_tokenize(statement)
    print("Tokenized text:", token)

    prediction = model1.predict([statement])
    print(f"Predicted sentiment using CountVectorizer:", prediction[0], end="\n\n")

i = 0
while i < new_texts_to_analyze:
  user_input = input("Enter a text: ")
  if user_input == "no":
    break

  prediction = model1.predict([user_input])
  print("Predicted Sentiment using CountVectorizer:", prediction[0])
  i += 1
```

With TfidfVectorizer



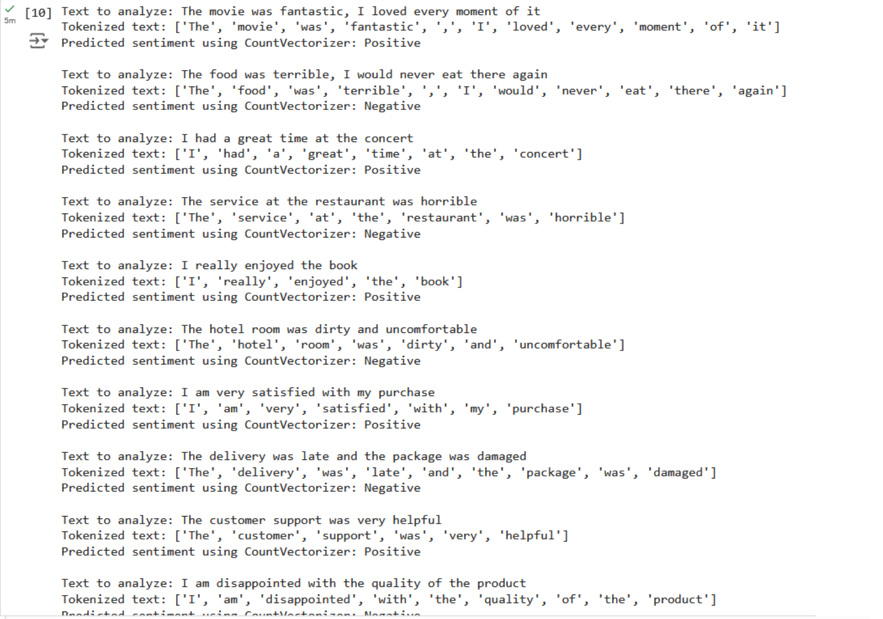
```
model2 = make_pipeline(TfidfVectorizer(), MultinomialNB())
model2.fit(texts, labels)

for statement in texts:
    print("Text to analyze:", statement)
    token = word_tokenize(statement)
    print("Tokenized text:", token)

    prediction = model2.predict([statement])
    print(f"Predicted sentiment using TfidfVectorizer:", prediction[0], end="\n\n")

print("-----")
print("NEW TEXTS FOLLOW")
print("-----")

i = 0
while i < new_texts_to_analyze:
  user_input = input("Enter a text: ")
  if user_input == "no":
    break

  prediction = model2.predict([user_input])
  print("Predicted Sentiment using TdidfVectorizer:", prediction[0], end="\n\n")
  i += 1
```

Luis Anton P. Imperial
BCS32

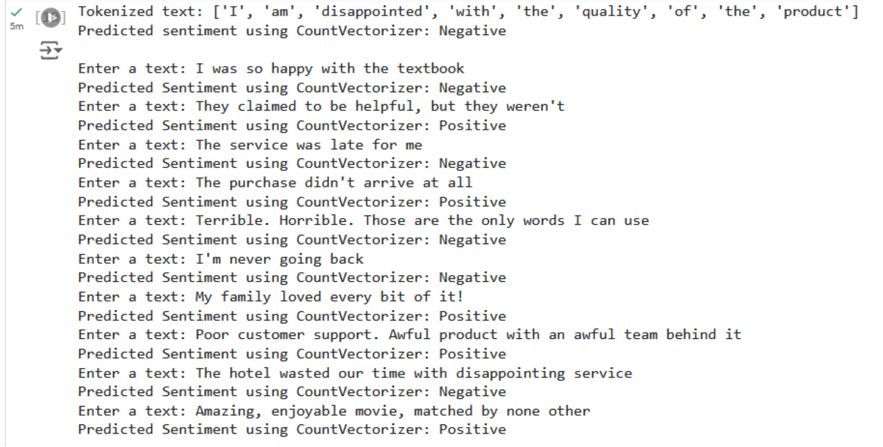
S-CSIS312
Natural Language Processing

Thursday, September 5, 2024
Session 1 Mini-Activity

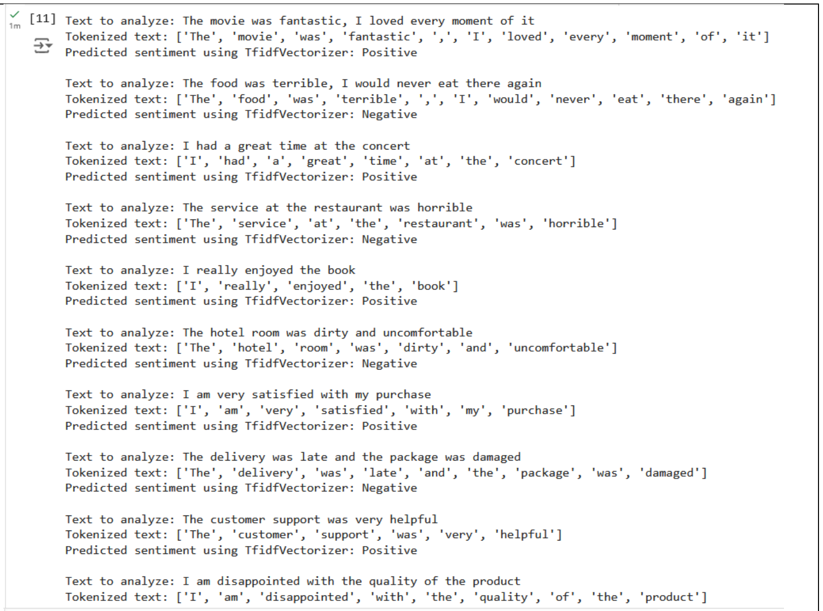
Outputs



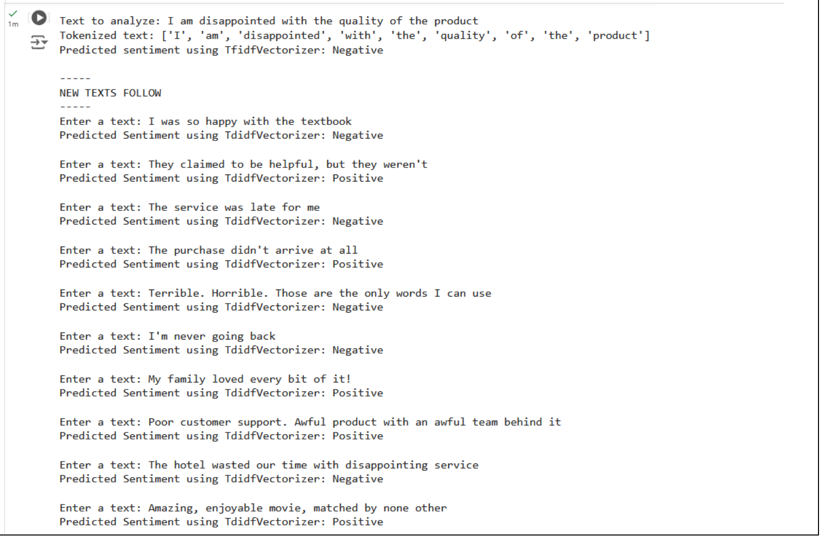
```
[10] Text to analyze: The movie was fantastic, I loved every moment of it
     Tokenized text: ['The', 'movie', 'was', 'fantastic', ',', 'I', 'loved', 'every', 'moment', 'of', 'it']
     Predicted sentiment using CountVectorizer: Positive

     Text to analyze: The food was terrible, I would never eat there again
     Tokenized text: ['The', 'food', 'was', 'terrible', ',', 'I', 'would', 'never', 'eat', 'there', 'again']
     Predicted sentiment using CountVectorizer: Negative

     Text to analyze: I had a great time at the concert
     Tokenized text: ['I', 'had', 'a', 'great', 'time', 'at', 'the', 'concert']
     Predicted sentiment using CountVectorizer: Positive

     Text to analyze: The service at the restaurant was horrible
     Tokenized text: ['The', 'service', 'at', 'the', 'restaurant', 'was', 'horrible']
     Predicted sentiment using CountVectorizer: Negative

     Text to analyze: I really enjoyed the book
     Tokenized text: ['I', 'really', 'enjoyed', 'the', 'book']
     Predicted sentiment using CountVectorizer: Positive

     Text to analyze: The hotel room was dirty and uncomfortable
     Tokenized text: ['The', 'hotel', 'room', 'was', 'dirty', 'and', 'uncomfortable']
     Predicted sentiment using CountVectorizer: Negative

     Text to analyze: I am very satisfied with my purchase
     Tokenized text: ['I', 'am', 'very', 'satisfied', 'with', 'my', 'purchase']
     Predicted sentiment using CountVectorizer: Positive

     Text to analyze: The delivery was late and the package was damaged
     Tokenized text: ['The', 'delivery', 'was', 'late', 'and', 'the', 'package', 'was', 'damaged']
     Predicted sentiment using CountVectorizer: Negative

     Text to analyze: The customer support was very helpful
     Tokenized text: ['The', 'customer', 'support', 'was', 'very', 'helpful']
     Predicted sentiment using CountVectorizer: Positive

     Text to analyze: I am disappointed with the quality of the product
     Tokenized text: ['I', 'am', 'disappointed', 'with', 'the', 'quality', 'of', 'the', 'product']
     Predicted sentiment using CountVectorizer: Negative
```



```
Tokenized text: ['I', 'am', 'disappointed', 'with', 'the', 'quality', 'of', 'the', 'product']
Predicted sentiment using CountVectorizer: Negative

Enter a text: I was so happy with the textbook
Predicted Sentiment using CountVectorizer: Negative
Enter a text: They claimed to be helpful, but they weren't
Predicted Sentiment using CountVectorizer: Positive
Enter a text: The service was late for me
Predicted Sentiment using CountVectorizer: Negative
Enter a text: The purchase didn't arrive at all
Predicted Sentiment using CountVectorizer: Positive
Enter a text: Terrible. Horrible. Those are the only words I can use
Predicted Sentiment using CountVectorizer: Negative
Enter a text: I'm never going back
Predicted Sentiment using CountVectorizer: Negative
Enter a text: My family loved every bit of it!
Predicted Sentiment using CountVectorizer: Positive
Enter a text: Poor customer support. Awful product with an awful team behind it
Predicted Sentiment using CountVectorizer: Positive
Enter a text: The hotel wasted our time with disappointing service
Predicted Sentiment using CountVectorizer: Negative
Enter a text: Amazing, enjoyable movie, matched by none other
Predicted Sentiment using CountVectorizer: Positive
```



```
[11] Text to analyze: The movie was fantastic, I loved every moment of it
     Tokenized text: ['The', 'movie', 'was', 'fantastic', ',', 'I', 'loved', 'every', 'moment', 'of', 'it']
     Predicted sentiment using TfidfVectorizer: Positive

     Text to analyze: The food was terrible, I would never eat there again
     Tokenized text: ['The', 'food', 'was', 'terrible', ',', 'I', 'would', 'never', 'eat', 'there', 'again']
     Predicted sentiment using TfidfVectorizer: Negative

     Text to analyze: I had a great time at the concert
     Tokenized text: ['I', 'had', 'a', 'great', 'time', 'at', 'the', 'concert']
     Predicted sentiment using TfidfVectorizer: Positive

     Text to analyze: The service at the restaurant was horrible
     Tokenized text: ['The', 'service', 'at', 'the', 'restaurant', 'was', 'horrible']
     Predicted sentiment using TfidfVectorizer: Negative

     Text to analyze: I really enjoyed the book
     Tokenized text: ['I', 'really', 'enjoyed', 'the', 'book']
     Predicted sentiment using TfidfVectorizer: Positive

     Text to analyze: The hotel room was dirty and uncomfortable
     Tokenized text: ['The', 'hotel', 'room', 'was', 'dirty', 'and', 'uncomfortable']
     Predicted sentiment using TfidfVectorizer: Negative

     Text to analyze: I am very satisfied with my purchase
     Tokenized text: ['I', 'am', 'very', 'satisfied', 'with', 'my', 'purchase']
     Predicted sentiment using TfidfVectorizer: Positive

     Text to analyze: The delivery was late and the package was damaged
     Tokenized text: ['The', 'delivery', 'was', 'late', 'and', 'the', 'package', 'was', 'damaged']
     Predicted sentiment using TfidfVectorizer: Negative

     Text to analyze: The customer support was very helpful
     Tokenized text: ['The', 'customer', 'support', 'was', 'very', 'helpful']
     Predicted sentiment using TfidfVectorizer: Positive

     Text to analyze: I am disappointed with the quality of the product
     Tokenized text: ['I', 'am', 'disappointed', 'with', 'the', 'quality', 'of', 'the', 'product']
```



```
Text to analyze: I am disappointed with the quality of the product
Tokenized text: ['I', 'am', 'disappointed', 'with', 'the', 'quality', 'of', 'the', 'product']
Predicted sentiment using TfidfVectorizer: Negative

-----
NEW TEXTS FOLLOW
-----
Enter a text: I was so happy with the textbook
Predicted Sentiment using TdidfVectorizer: Negative

Enter a text: They claimed to be helpful, but they weren't
Predicted Sentiment using TdidfVectorizer: Positive

Enter a text: The service was late for me
Predicted Sentiment using TdidfVectorizer: Negative

Enter a text: The purchase didn't arrive at all
Predicted Sentiment using TdidfVectorizer: Positive

Enter a text: Terrible. Horrible. Those are the only words I can use
Predicted Sentiment using TdidfVectorizer: Negative

Enter a text: I'm never going back
Predicted Sentiment using TdidfVectorizer: Negative

Enter a text: My family loved every bit of it!
Predicted Sentiment using TdidfVectorizer: Positive

Enter a text: Poor customer support. Awful product with an awful team behind it
Predicted Sentiment using TdidfVectorizer: Positive

Enter a text: The hotel wasted our time with disappointing service
Predicted Sentiment using TdidfVectorizer: Negative

Enter a text: Amazing, enjoyable movie, matched by none other
Predicted Sentiment using TdidfVectorizer: Positive
```

Luis Anton P. Imperial	S-CSIS312	Thursday, September 5, 2024
BCS32	Natural Language Processing	Session 1 Mini-Activity

Insights on the Two Functions

CountVectorizer simply converts the given texts into a matrix of token counts. This means that if a token is repeated multiple times, it will be prioritized as important parts of a label. Similarly, whatever input is inserted into the function is the only source of prioritization for CountVectorizer.

Since we have not provided the module with a dictionary categorizing every word in the English language, text containing new words are of an unknown entity, and are simply set aside. This also means that very common words, such as articles (i.e. “the”, “a” and “an”) and conjunctions (“and” and “or”), are rated highly, which is contrary to the inner workings of any language.

TfidfTransformer transforms this basic count matrix into a normalized representation based on TF-IDF (meaning, term-frequency times inverse document-frequency).

Instead of using the raw frequencies of occurrence in a given set of texts, the goal is to scale down the tokens that occur too frequently in favor of ones that occur in a smaller scale in the training corpus.