FI_Studrev_sent_analysis

July 30, 2025

[36]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from textblob import TextBlob
from wordcloud import WordCloud
import nltk
```

[2]:
```
#Download NLTK corpora
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\BOOKWORM\AppData\Roaming\nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[2]: True

- Load Data

[37]:
```
#Load dataset
stu_data = pd.read_csv(r"C:\Users\BOOKWORM\Desktop\Folders\BOOKWORM\DATA␣
 ↪ANALYSIS\FI\Task 3\Student data\course_data_clean.csv")

#link to dataset (https://www.opendatabay.com/data/ai-ml/
 ↪ea4dcd19-f608-4acb-9d84-5d9d58d3f7f4)
```

[38]:
```
# 3. Inspect the Data
print("Initial shape:", stu_data.shape)
print("Columns available:", stu_data.columns)
stu_data = stu_data[['reviews', 'course_rating', 'course_rating_int']].copy()
stu_data.dropna(inplace=True)
```

```
Initial shape: (14838, 10)
Columns available: Index(['course_code', 'course_title', 'num_ratings',
'useful', 'easy', 'liked',
       'num_reviews', 'reviews', 'course_rating', 'course_rating_int'],
      dtype='object')
```

1 Clean & Prepare

- We will focus on these columns: reviews (text)
- course_rating_int (0 == disliked, 1 == liked)

[26]:
```
# Check for missing values
stu_data.isnull().sum()

# Drop rows with missing reviews
stu_data = stu_data.dropna(subset=["reviews"])
```

2 Sentiment Analysis with TextBlob

[39]:
```
# Function to calculate sentiment polarity
def get_sentiment(text):
    return TextBlob(text).sentiment.polarity

# Apply to review column
stu_data['sentiment_score'] = stu_data['reviews'].apply(get_sentiment)

# Label as Positive, Neutral, Negative
def label_sentiment(score):
    if score > 0.1:
        return 'Positive'
    elif score < -0.1:
        return 'Negative'
    else:
        return 'Neutral'

stu_data['sentiment_label'] = stu_data['sentiment_score'].apply(label_sentiment)
```

3 Visualize Sentiment

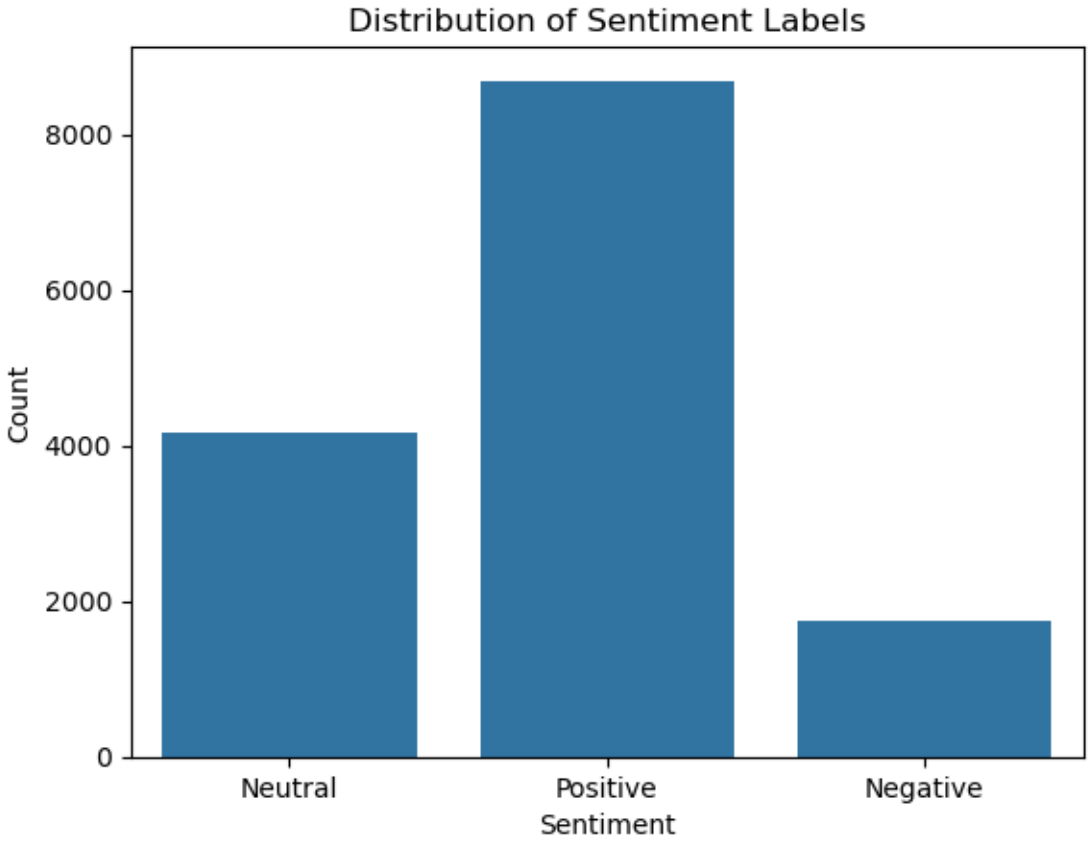
- Bar chart of sentiment labels:

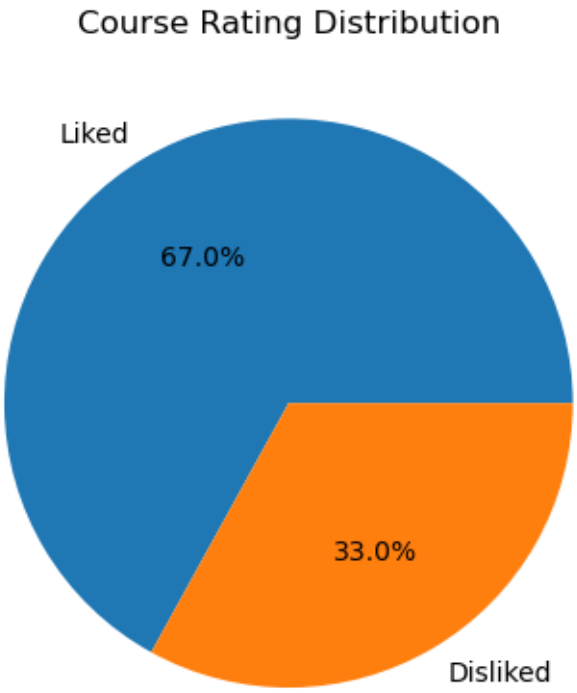
[40]:
```
sns.countplot(x='sentiment_label', data=stu_data)
plt.title("Distribution of Sentiment Labels")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()

#Pie chart of course ratings:

stu_data['course_rating_int'].value_counts().plot.pie(autopct='%1.1f%%',␣
 ↪labels=["Liked", "Disliked"])
plt.title("Course Rating Distribution")
```

```
plt.ylabel("")
plt.show()
```





4 Word Cloud of Reviews

[41]:
```
text = " ".join(review for review in stu_data['reviews'])
wordcloud = WordCloud(width=800, height=400, background_color='white').
 ↪generate(text)

plt.figure(figsize=(10,5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title("Common Words in Course Reviews")
plt.show()
```

5 Correlation between sentiment and course rating

[42]:
```
pd.crosstab(stu_data['sentiment_label'], stu_data['course_rating_int'],␣
 ↪normalize='index') * 100
```

[42]:

course_rating_int	0.0	1.0
sentiment_label		
Negative	73.596793	26.403207
Neutral	44.998801	55.001199
Positive	19.130335	80.869665

6 Summary report

[43]:
```
print("Sentiment Summary:\n", stu_data['sentiment_label'].
 ↪value_counts(normalize=True) * 100)
print("\nAverage sentiment score for liked courses:",
      stu_data[stu_data['course_rating_int'] == 1]['sentiment_score'].mean())
print("Average sentiment score for disliked courses:",
      stu_data[stu_data['course_rating_int'] == 0]['sentiment_score'].mean())
```

```
Sentiment Summary:
 sentiment_label
Positive    59.508488
Neutral     28.539157
Negative    11.952355
Name: proportion, dtype: float64
```

```
Average sentiment score for liked courses: 0.2257902886566242
Average sentiment score for disliked courses: 0.002293934014016523
```

6.1 Recommendations for Course Improvement

Based on the sentiment analysis and student feedback, the following recommendations are suggested for course organizers:

1. *Enhance Teaching Methods and Resources*
   - Negative feedback pointed to outdated or ineffective teaching styles.
   - Recommendation: Adopt interactive tools like live coding, discussion forums, and recorded lectures.
2. *Provide More Academic Support*
   - Students reported difficulty in understanding some course concepts.
   - Recommendation: Organize office hours, tutoring sessions, or peer learning groups.
3. *Improve Course Pacing and Structure*
   - Several comments mentioned abrupt transitions or inconsistent workloads.
   - Recommendation: Review the curriculum flow to ensure a balanced and progressive learning experience.
4. *Clarify Assessment and Expectations*
   - Complaints were seen about unclear grading and assignment deadlines.
   - Recommendation: Share rubrics and weekly schedules in advance to help students stay aligned.
5. *Foster a More Engaging Learning Environment*
   - Some reviews described the course as boring or dry.
   - Recommendation: Introduce real-life applications, guest speakers, or project-based learning.

These suggestions aim to enhance both the academic and emotional experience of students in future cohorts.

[ ]: