Code

```
    import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt

# Function to classify review sentiment
def classify_sentiment(review):
    analysis = TextBlob(str(review))
    polarity = analysis.sentiment.polarity

    if polarity > 0.1:
        return '😊 Happy'
    elif polarity < -0.1:
        return '😞 Sad'
    else:
        return '😐 Neutral'

csv_path = 'IMDB Dataset.csv'

# Load the dataset
df = pd.read_csv(csv_path, on_bad_lines='skip', encoding='utf-8')

# Ensure the review column is named correctly
if 'review' not in df.columns:
    text_columns = [col for col in df.columns if 'text' in col.lower() or 'review' in col.lower()
or 'comment' in col.lower()]
    df.rename(columns={text_columns[0]: 'review'}, inplace=True)

# Apply sentiment analysis
df['sentiment'] = df['review'].apply(classify_sentiment)
total_reviews = len(df)

# Count sentiment results
sentiment_counts = df['sentiment'].value_counts()
percentages = (sentiment_counts / total_reviews) * 100

# Prepare result summary
results_df = pd.DataFrame({
    'Sentiment': ['😊 Happy', '😐 Neutral', '😞 Sad'],
    'Count': [
        sentiment_counts.get('😊 Happy', 0),
        sentiment_counts.get('😐 Neutral', 0),
        sentiment_counts.get('😞 Sad', 0)
    ],
```

```
    'Percentage': [
        round(percentages.get('😊 Happy', 0), 1),
        round(percentages.get('😐 Neutral', 0), 1),
        round(percentages.get('😞 Sad', 0), 1)
    ]
})

# Print sentiment breakdown
print("=========================================")
print(f"TOTAL REVIEWS ANALYZED: {total_reviews}")
print("=========================================")
print("\nSentiment Breakdown:")
print("-----------------------------")
for _, row in results_df.iterrows():
    print(f"{row['Sentiment']}: {row['Count']} reviews ({row['Percentage']}%)")
print("-----------------------------")

# Pie chart
plt.figure(figsize=(10, 8))
colors = ['#a8e6cf', '#ffd3b6', '#ff8b94']
explode = (0.05, 0.05, 0.05)

def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        val = int(round(pct * total / 100.0))
        return f'{pct:.1f}%\n({val})'
    return my_autopct

plt.pie(
    results_df['Count'],
    labels=results_df['Sentiment'],
    colors=colors,
    explode=explode,
    autopct=make_autopct(results_df['Count']),
    startangle=90,
    shadow=True,
    textprops={'fontsize': 12}
)

plt.axis('equal')
plt.title(f'Sentiment Distribution of {total_reviews} Movie Reviews', pad=20, fontsize=14)
plt.tight_layout()
plt.show()
```

Outputs:-

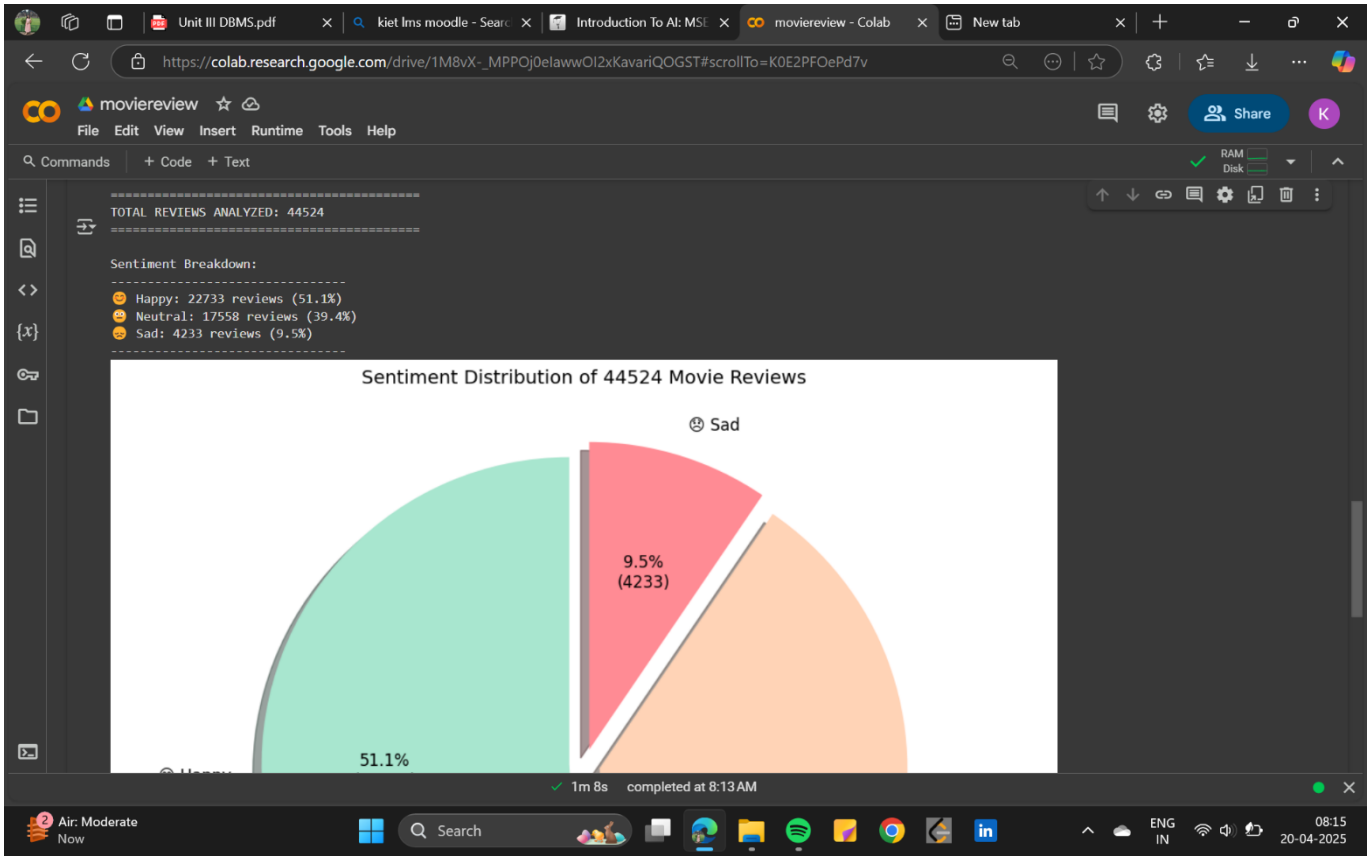

Figure-1

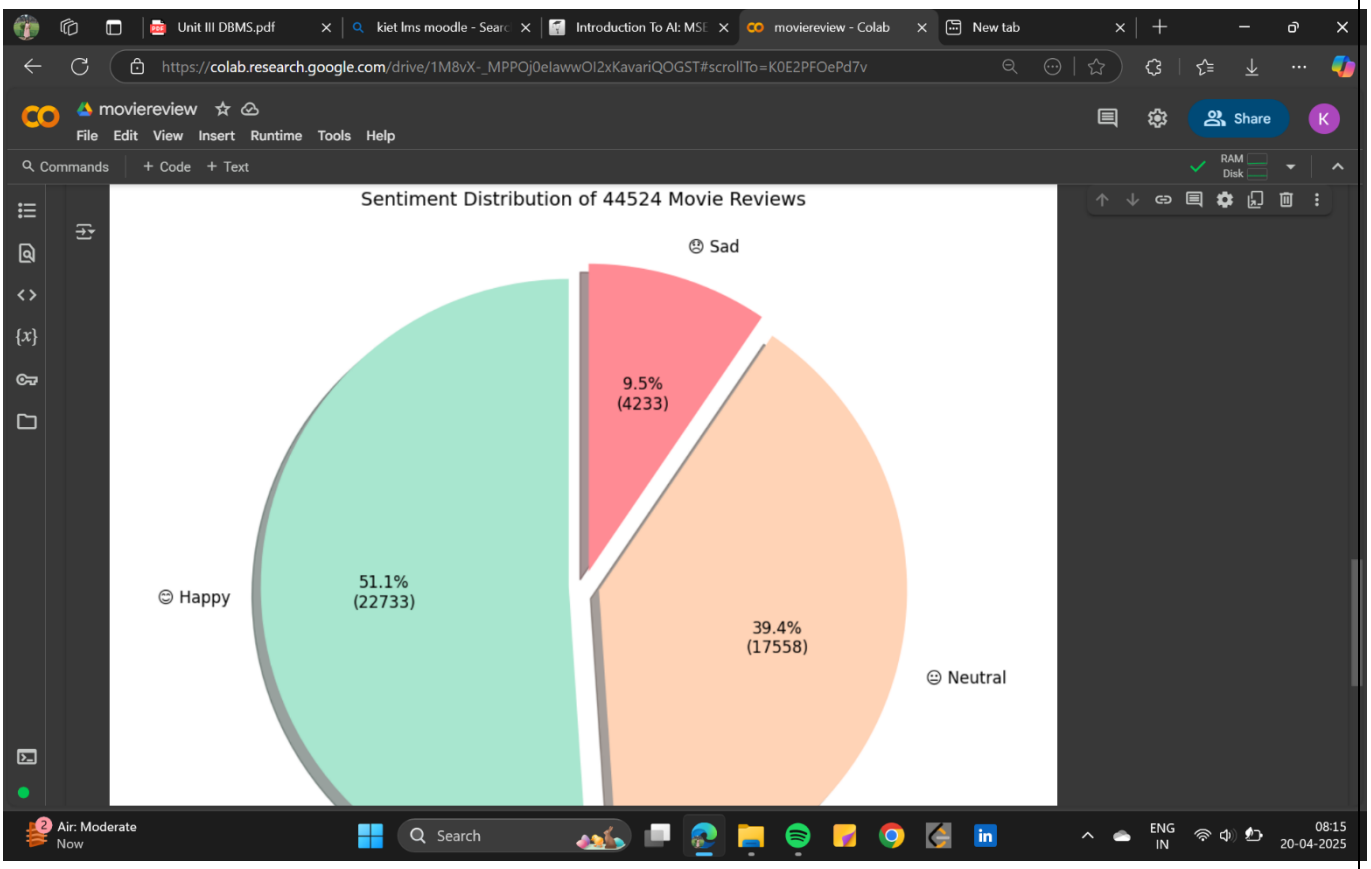
Unit III DBMS.pdf
kiet lms moodle - Searc
Introduction To AI: MSE
moviereview - Colab
New tab
https://colab.research.google.com/drive/1M8vX-_MPPOj0elawwOI2xKavariQOGST#scrollTo=K0E2PFOePd7v
moviereview
File Edit View Insert Runtime Tools Help
Share
Commands
+ Code
+ Text
RAM
Disk
Sentiment Distribution of 44524 Movie Reviews
Sad
9.5%
(4233)
Happy
51.1%
(22733)
39.4%
(17558)
Neutral
Air: Moderate
Now
Search
ENG
IN
08:15
20-04-2025

Figure-2

OUTPUT EXPLANATION

To analyze movie reviews from a CSV file (IMDB Dataset.csv) and classify each review as either:

😊 Happy (Positive sentiment)

😐 Neutral (Balanced sentiment)

😞 Sad (Negative sentiment)

What the Output Shows:

1. TOTAL REVIEWS ANALYZED
   This tells you the number of reviews processed from the dataset.

2. SENTIMENT BREAKDOWN
   This section shows:
   - Each sentiment category (with emoji)
   - The number of reviews in each category
   - The percentage of total reviews

3. PIE CHART A VISUAL REPRESENTATION OF THE SENTIMENT DISTRIBUTION:
   Each slice = sentiment category
   Size = proportional to the number of reviews
   Labels show both % and review count
   Exploded slices make each segment pop visually