🎬 Movie Review Sentiment Analysis Workshop

Using Wikipedia API, IMDB Scraping & ChatGPT

🎯 Workshop Objectives

1. Fetch movie data using Wikipedia API
2. Extract IMDB reviews through web scraping
3. Process and filter reviews
4. Perform sentiment analysis using ChatGPT
5. Visualize results with D3.js

🛠️ Setup Requirements

```
# Required libraries
!pip install wikipedia-api
!pip install beautifulsoup4
!pip install requests
!pip install pandas
```

🔍 Check Installation:

```
import wikipediaapi
import requests
from bs4 import BeautifulSoup
import pandas as pd
import json
```

🌐 Using Wikipedia API

Step 1: Initialize Wikipedia API

```
# Your turn! Initialize Wikipedia API
# Hint: Use wikipediaapi.Wikipedia()
```

Expected Output:

```
wiki = wikipediaapi.Wikipedia(
    language='en',
    user_agent='MovieReviewBot/1.0'
)
```

🎥 Finding Movie Information

Step 2: Search for Movie Page

```
# Your turn! Search for a movie
# Hint: Use wiki.page()
```

Interactive Exercise:

Let's search for "The Godfather" together!

🔗 Extracting IMDB URL

Step 3: Get External Links

```
# Your turn! Extract external links
# Hint: Use page.extlinks
```

🤔 Discussion:

- How can we filter for IMDB links?
- What patterns do we notice in IMDB URLs?

🕷 Accessing IMDB Reviews

Step 4: Navigate to Reviews Page

```
# Transform movie URL to reviews URL
# Example: /title/tt0068646/ → /title/tt0068646/reviews
```

🎯 **Practice:**

Modify the URL pattern together!

📝 Scraping Reviews

Step 5: Extract Non-Spoiler Reviews

```
def scrape_reviews(url):
    # Your turn! Write the scraping logic
    # Hint: Look for class='text show-more__control'
```

🚫 **Important:** Why can't we scrape spoiler reviews?

💾 Saving Reviews

Step 6: Create JSON Structure

```
reviews_data = {
    'movie_title': movie_title,
    'reviews': []
}

# Your turn! Add reviews to the structure
```

✂️ Filtering Reviews

Step 7: Length-Based Filtering

```
def filter_reviews(reviews, min_words=100):
    # Your turn! Filter reviews by word count
    # Hint: Use len(review.split())
```

🤔 Why filter?

- ChatGPT context limitations
- Quality of analysis
- Meaningful insights

🤖 Introduction to Prompt Engineering

Types of Prompts:

1. Basic Prompt:

```
Analyze the sentiment of this review:
[Review Text]
```

2. Structured Prompt:

```
Please analyze this movie review and provide:
1. Sentiment (Positive/Negative/Neutral)
2. Key themes
3. Emotional intensity (1-5)
[Review Text]
```

📚 Few-Shot Prompting Example

```
Example 1:
Review: "A masterpiece of cinematography with outstanding performances"
Analysis: Positive sentiment, themes: visual excellence, acting quality

Example 2:
Review: "Boring plot with terrible pacing"
Analysis: Negative sentiment, themes: story structure, engagement

Now analyze:
[New Review]
```

🎯 Creating Our Sentiment Analysis Prompt

```
You are a film critic specializing in sentiment analysis.
For each review below, provide:
1. Sentiment score (-1 to 1)
2. Main emotions detected
3. Key aspects mentioned (acting, plot, etc.)
4. One-line summary

Format: JSON

Reviews:
[Reviews List]
```

💾 Saving Analysis Results

Step 8: Structure the Output

```
analysis_results = {
    'movie_title': movie_title,
    'analysis': []
}

# Your turn! Add ChatGPT analysis to structure
```

Visualizing Results with D3.js

```
// Load analysis results
d3.json('analysis_results.json').then(data => {
    // Your turn! Create visualization
    // Hint: Try a sentiment distribution chart
})
```

🎯 Hands-on Exercise

In pairs (15 minutes):

1. Pick an Oscar-winning movie
2. Get its IMDB reviews
3. Create a custom prompt
4. Analyze 5 reviews
5. Share interesting findings!

🔍 Common Challenges & Solutions

1. Rate limiting
2. HTML structure changes
3. ChatGPT token limits
4. Data cleaning needs

👥 **Discussion:** What challenges did you face?

💡 Best Practices

1. Error handling for API calls
2. Review filtering strategy
3. Prompt design principles
4. Data validation steps
5. Output formatting

🎉 Workshop Wrap-up

Next Steps:

1. Try different movies
2. Experiment with prompts
3. Enhance visualizations
4. Compare different eras/genres

Resources:

- Wikipedia API Documentation
- ChatGPT Prompt Engineering Guide

🤝 Q&A Session

- What surprised you most?
- Which part was most challenging?
- How would you improve the analysis?

Remember to share your results with the group!