Aim

To analyze and classify the sentiment of customer reviews from a Yelp page using a pre-trained BERT model.

Objectives

1. To scrape customer reviews from a Yelp page.
2. To preprocess the reviews for sentiment analysis.
3. To apply a pre-trained sentiment analysis model to classify the reviews.
4. To present the sentiment scores of the reviews in a structured format.

Summary

This project aims to perform sentiment analysis on customer reviews from a Yelp page. Using web scraping techniques, reviews are collected from a specific Yelp page. These reviews are then analyzed using a pre-trained BERT model to classify the sentiment. The sentiment scores are stored in a DataFrame for easy analysis and interpretation.

Tools and Libraries Used

- **Google Colab**: For running the project
- **Hugging Face Transformers**: For using the pre-trained BERT model
- **Torch**: For handling tensor operations and model inference
- **Requests**: For making HTTP requests to web pages
- **BeautifulSoup**: For web scraping
- **re**: For regular expression matching
- **Pandas**: For data manipulation and analysis
- **NumPy**: For numerical operations

Procedure

1. **Mount Google Drive**:

CODE:

```
from google.colab import drive

drive.mount('/content/drive')
```

2. **Import Libraries**:

CODE:

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification
```

```
import torch

import requests

from bs4 import BeautifulSoup

import re

import numpy as np

import pandas as pd
```

3. **Load Pre-trained BERT Model and Tokenizer**:

CODE:

```
tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')

model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

4. **Test the Model with a Sample Review**:

CODE:

```
tokens = tokenizer.encode('It was good but couldve been better. Great', return_tensors='pt')

result = model(tokens)

sentiment = int(torch.argmax(result.logits)) + 1
```

5. **Web Scraping Yelp Reviews**:

CODE:

```
r = requests.get('https://www.yelp.com/biz/social-brew-cafe-pyrmont')

soup = BeautifulSoup(r.text, 'html.parser')

regex = re.compile('.*comment.*')

results = soup.find_all('p', {'class': regex})

reviews = [result.text for result in results]
```

6. **Store Reviews in a DataFrame**:

CODE:

```
df = pd.DataFrame(np.array(reviews), columns=['review'])
```

7. **Define Sentiment Analysis Function**:

CODE:

```
def sentiment_score(review):
```

```
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits)) + 1
```

8. **Apply Sentiment Analysis to Reviews**:

CODE:

```
df['sentiment'] = df['review'].apply(lambda x: sentiment_score(x[:512]))
```

9. **View Results**:

CODE:

```
df.head()
```

Highlights

- **Web Scraping**: Efficiently collected reviews from a Yelp page using BeautifulSoup.
- **Sentiment Analysis**: Utilized a pre-trained BERT model from Hugging Face Transformers to classify the sentiment of reviews.
- **Data Handling**: Processed and stored the reviews and their sentiment scores in a pandas DataFrame for easy manipulation and analysis.

Conclusion

This project successfully demonstrates the use of web scraping to collect data and the application of a pre-trained BERT model for sentiment analysis. The sentiment scores provide valuable insights into customer opinions, which can be beneficial for businesses to understand and improve their services.