

Analyse des Sentiments sur Amazon

Ahmed HABBANI

December 15, 2024

Introduction

Ce projet vise à analyser les avis des clients sur Amazon afin de comprendre leurs sentiments à l'aide de techniques de traitement du langage naturel (NLP). Il aide les entreprises à obtenir des informations précieuses à partir des retours clients pour améliorer leur satisfaction.

1 Étapes du Pipeline NLP

1.1 1. Collecte et Chargement des Données

L'objectif est de charger les données textuelles à analyser.

Listing 1: Code Python

```
1 df = pd.read_csv("/content/sample_data/amazon.csv")
```

1.2 2. Nettoyage des Données Textuelles (Text Cleaning)

Cette étape permet de normaliser les textes en supprimant les caractères non pertinents et en convertissant les textes en minuscules.

Listing 2: Code Python

```
1 df.reviewText.head()
2 review_example = re.sub("[^a-zA-Z]", '', review_example)
3 review_example = review_example.lower().split()
4
5 rt = lambda x: re.sub("[^a-zA-Z]", ' ', str(x))
6 df["reviewText"] = df["reviewText"].map(rt)
7 df["reviewText"] = df["reviewText"].str.lower()
```

1.3 3. Analyse Exploratoire des Données Textuelles

Examiner les données pour identifier les colonnes manquantes, les doublons et les classes uniques.

Listing 3: Code Python

```
1 check_dataframe(df)
2 check_class(df)
```

1.4 4. Analyse Lexicale

Extraction des caractéristiques linguistiques comme la polarité et la subjectivité des textes avec TextBlob.

Listing 4: Code Python

```
1 df[['polarity', 'subjectivity']] = df['reviewText'].apply(
2     lambda text: pd.Series(TextBlob(text).sentiment)
3 )
```

1.5 5. Analyse des Sentiments (Sentiment Analysis)

Détection des sentiments associés à chaque texte à l'aide de VADER `SentimentIntensityAnalyzer`.

Listing 5: Code Python

```
1 analyzer = SentimentIntensityAnalyzer()
2 for index, row in df.iterrows():
3     review_text = row['reviewText']
4     score = analyzer.polarity_scores(review_text)
5
6     neg = score['neg']
7     pos = score['pos']
8
9     if neg > pos:
10         df.loc[index, 'sentiment'] = "Negative"
11     elif pos > neg:
12         df.loc[index, 'sentiment'] = "Positive"
13     else:
14         df.loc[index, 'sentiment'] = "Neutral"
```

1.6 6. Visualisation des Données Catégoriques

Création de graphiques comme des barres et des camemberts pour explorer les distributions des sentiments.

Listing 6: Code Python

```
1 categorical_variable_summary(df, 'overall')
```

1.7 7. Filtrage des Données pour Identifier des Insights

Tri et exploration des avis positifs/négatifs selon des critères définis (*e.g. wilson_lower_bound*).

Listing 7: Code Python

```
1 df[df['sentiment'] == 'Positive'].sort_values("wilson_lower_bound"  
    ↪ ", ascending=False).head(10)
```

1.8 8. Mise en Forme des Sorties et des Résultats

Affichage des résultats sous forme de tableau pour une analyse exploitable.

Listing 8: Code Python

```
1 print(df[['reviewText', 'polarity', 'subjectivity', 'sentiment'  
    ↪ ]].head())
```