Product reviews sentiment analysis

Radu Andreea-Denisa, Bianca Iorgoaea, Baciu Cristian, Dragoș Vlad
Gr. 244, SDI

1 Introduction

This project aims to utilize techniques of sentiment analysis specifically applied to user reviews for products sold on Amazon website. There is an increasing importance of online feedback in consumer decision-making, with a significant impact on a product or on a brand reputation. Besides being a relevant topic, sentiment analysis is an interesting and useful application of data mining.

Our main objective was to develop and apply machine learning models for sentiment analysis, focusing on categorizing reviews into two main classes: positive and negative. In our research process, we chose to apply data mining techniques before training two popular and effective machine learning algorithms, with the purpose of comparing their performances and evaluating their effectiveness in the specific context of sentiment analysis in product reviews.

2 Methodology

2.1 Dataset

The chosen dataset includes 7,000 text reviews for various products available on Amazon, together with the related label. A 1-star or 2-star review was considered negative, and a 3- or 4-star review was considered positive, eliminating neutral reviews. The dataset was further divided into a training set and a testing set, keeping the proportions of 80% and 20%, respectively.

2.2 Data preprocessing

Preprocessing the available data is the first step which ensures an increased quality of text analysis.

A pre-trained Natural Language Processing (NLP) model, capable of tokenizing and analyzing English text, was used. Lines containing both labels and reviews were selected for our project. The review text is stripped of leading and trailing whitespaces and converted to lowercase. Additionally, the NLP model is used to tokenize the review text, and a list comprehension is employed in order to filter out stopwords. The corresponding function is shown in Figure 1.

```
def process_file(file_path):

    nlp = spacy.load("en_core_web_sm")
    with open(file_path, 'r', encoding='utf-8') as file:
        reviews =[]
        for line in file:
            parts = line.split(maxsplit=1)
            if len(parts) == 2:
                label, review = parts
                review = review.strip().lower()  # Lowercase the text and remove leading/trailing whitespaces
                cleaned_text = ' '.join([token.text for token in nlp(review) if token.text.lower() not in STOP_WORDS])
                reviews.append((label, cleaned_text))
    return reviews
```

Figure 1: Preprocessing data function

Figure 2 displays a sample of how the data looks like after performing the preprocessing task.

2.3 Data mining techniques

Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique was used for feature extraction step, before training machine learning models. The term frequency-inverse document frequency is calculated for each term in the document, providing a numerical

	Text
0	stuning non - gamer : sound track beautiful ! paints senery mind recomend people hate vid . game...
1	best soundtrack . : reading lot reviews saying best ' game soundtrack ' figured write review dis...
2	amazing ! : soundtrack favorite music time , hands . intense sadness " prisoners fate " (means ...
3	excellent soundtrack : truly like soundtrack enjoy video game music . played game music enjoy tr...
4	remember , pull jaw floor hearing : played game , know divine music ! single song tells story ga...
5	absolute masterpiece : sure actually taking time read played game , heard tracks . aware , mitsu...
6	buyer beware : self - published book , want know -- read paragraphs ! 5 star reviews written ms ...
7	glorious story : loved whisper wicked saints . story amazing pleasantly surprised changes book ....
8	star book : finished reading whisper wicked saints . fell love caracters . expected average roma...
9	whispers wicked saints : easy read book want reading , easy down.it left wanting read follow , h...
10	worst ! : complete waste time . typographical errors , poor grammar , totally pathetic plot add ...
11	great book : great book , , read fast . boy book twist turns keeps guessing wanting know going h...
12	great read : thought book brilliant , realistic . showed error human . loved fact writer showed ...
13	oh : guess romance novel lover , discerning . beware ! absolute drivel . figured trouble typo pr...
14	awful belief ! : feel write wasting money . book written 7th grader poor grammatical skills age ...
15	try fool fake reviews . : glaringly obvious glowing reviews written person , author . misspellin...
16	romantic zen baseball comedy : hear folks 'em like anymore , talking " sea " . cool story young ...
17	fashionable compression stockings ! : dvt doctor required wear compression stockings . wore ugly...
18	jobst ultrasheer thigh high : excellent product . , difficult older people . feel like day worko...
19	sizes recomended size chart real : sizes smaller recomended chart . tried sheer ! . guess buy it...

Figure 2: Sample of first 20 text reviews after preprocessing

representation of the importance of that term. As shown in Figure 3, We chose to set the limit for the number of unique terms to the top 5000 most significant ones, helping to control the dimensionality of the resulting feature space. This vectorization technique was applied on both training and testing subset.

```
# Converting into numerical data using TF-IDF technique
tfidf_vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')

X_train_tfidf = tfidf_vectorizer.fit_transform(train_df['Text'])
```

Figure 3: TF-IDF technique

2.4 Implementation of machine learning models

In our sentiment analysis project, we opted to employ the Support Vector Machine (SVM) algorithm, specifically utilizing a linear kernel. The choice of SVM for sentiment analysis

is based in its effectiveness in handling high-dimensional feature spaces and its ability to perform well in binary classification tasks. The decision to use a radial basis function (RBF) kernel is motivated by its flexibility in capturing complex relationships in the data. A smaller C value is set, in order to obtain a more regularized model. This can be beneficial in sentiment analysis to prevent overfitting.

```
svm_model = SVC(C = 1.0, kernel='rbf', gamma='scale')
svm_model.fit(X_train_tfidf, y_train)
```

Figure 4: SVM model

The second machine learnig model employed is a simple neural network of three layers. The sparse TF-IDF matrix was previously converted into a dense NumPy array. Many machine learning algorithms, including neural networks, often expect dense input rather than sparse matrices. Given the dimensions of the dataset, 30 epochs were conveniently set for this training.

```
model = Sequential()
model.add(Dense(128, input_shape=(X_train_dense.shape[1],), activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))  # single neuron for binary classification

# Compile the model
model.compile(optimizer=Adam(learning_rate=0.001), loss='binary_crossentropy', metrics=['accuracy'])

model.fit(X_train_dense, y_train_encoded, epochs=30, batch_size=32)
```

Figure 5: Backpropagation deep learning model

3 Results and discussion

By analyzing the accuracy, precision, recall and F1-score for both machine learning models, SVM obtained better results for all metrics compared to the backpropagation deep learning model. The results are synthesized in the figure below.

	Model	Accuracy	Precision	Recall	F1-score
0	SVM	96%	95%	97%	96%
1	Backpropagation	81%	82%	78%	80%

Figure 6: Obtained metrics for both models

Support Vector Machines are known to perform well in situations where the dataset is relatively small and not highly complex. Similarly, the nature of the text data present in the product reviews dataset are more easily captured by a linear model like SVM. As an alternative explanation for the results, SVMs are effective in high-dimensional spaces, and TF-IDF representation often results in a high-dimensional feature space. SVMs can handle this well, while neural networks with three layers may struggle with high-dimensional input spaces.

4 Further improvement

The performance of the machine learning algorithms could be improved with pre-trained word embeddings, which use word vectors obtained by context, co-occurrence of the words, semantic and syntactic similarity. Word2Vec or Glove are examples of pre-trained word embeddings wihch may enhance the understanding of context once they are incorporated in

the models.

The implementation of Gated Recurrent Unit (GRU) layers within the trained models can offer significant advantages. This holds the potential to improve the understanding of the contextual nuances within reviews, ultimately enhancing the sentiment analysis performance of the models.