NLP TERM REPORT Group 5

December 3, 2023

1. K SIVASAI ,S20210010102
2. P L NARASIMHA MURTHY , S20210010185

1 Abstract

In this Project We will give the rating of the user based on the sentiment which we extract by the given input text of the user.

We have used multiple NLP Techniques like Lemmatisation,POStagging,N-grams and we finally Implemented it using ML Model Naive Bayes Classifier and got accuracy of 65

2 Introduction

We have chosen this project develop our ideas in the movie reviews.Movies are becoming popular and there are acting as one of the majors for providing the entertainment.There are many sites for giving reviews these days for movies. Though there are lot of sites There is a need to a genuine site which can give ratings based on user texts

3 Importance of the Project

Emotion of a user tweeted or given to a site in the reviews needs to be get rated as many people do not have time to read all reviews so a number system is required based on the sentiment we give ratings for particular user which in

turn gives a average of that rating s which helps to find the genuine reviews of most users.

our Passion towards movies made us to recognize the need of reviews in the field of cinema. Reviews mainly gives boost to the movie viewers to help them know about the movie which in result gives good collections to the movie.S our vision is to predict the movie reviews such that we can give rating to the movie review given movie viewers to let the public know to how much extent the movie is Good. so we have chosen this Project to tell the opinion of people who already watched the movie to people who wants to know about the movie.

3.1 Research Papers

To implement this Project we have gone through some references and research papers which include:

1)Movies Reviews Sentiment Analysis and Classification 2019 IEEE.

In this paper The dataset is taken form IMDB In which The ratings are classified as Positive and negative And classifiers such as naive bayes,svms and Decision tress are used.

2)A Study of Sentiment Analysis: Concepts, Techniques, and Challenges.

In this Paper Text preparation, sentiment detection is focussed mainly and Lexicon-Based Approach, Machine Learning Approach,Hybrid Techniques Approach are used.

Referring these Papers We came to know about the models and how sentiment detection can be done using these models and we made a Pipeline To build our Project

4 Pipeline And Procedure

1. We started with taking dataset from the Kaggle.
2. Taking some new data through web scraping from a website called Screenrant.
3. Data Cleaning is done by removing the missing values and removing unnecessary columns in the data set .

4. After that Tokenisation is done and word Expansions and abbrevations are also done

```
abbreviations = {
    "e.g.": "for example",
    "i.e.": "that is",
    "etc.": "et cetera",
    "Mr.": "Mister",
    "Dr.": "Doctor",
    "St.": "Saint",
    "Ave.": "Avenue",
    "Apt.": "Apartment",
    "Mfg.": "Manufacturing"
}
```

```
def expand_abbreviations(text):
    expanded_tokens = []
    for token in text.split():
        if token in abbreviations:
            expanded_tokens.append(abbreviations[token])
        else:
            expanded_tokens.append(token)
    return " ".join(expanded_tokens)


dataset['Phrase']=dataset['Phrase'].apply(expand_abbreviations)
```

Figure 1: Caption for your image.

5. Then Lower casing of letters and abbrevation expansion is done

```
# Lowercasing
def lowercase_text(text):
    if isinstance(text, str):
        return text.lower()
    else:
        return str(text).lower()

dataset['Phrase']=dataset['Phrase'].apply(lowercase_text)

dataset.head()
```

Figure 2: Lowercasing Of text.

6. stop words are removed by creating particular dictionary on our own of stopwords which is suitable for dataset and they are removed.

Figure 3: stopwords

7. After that Lemmatisation is done where we converted different words to their root form using dictionary of words suitable for our dataset

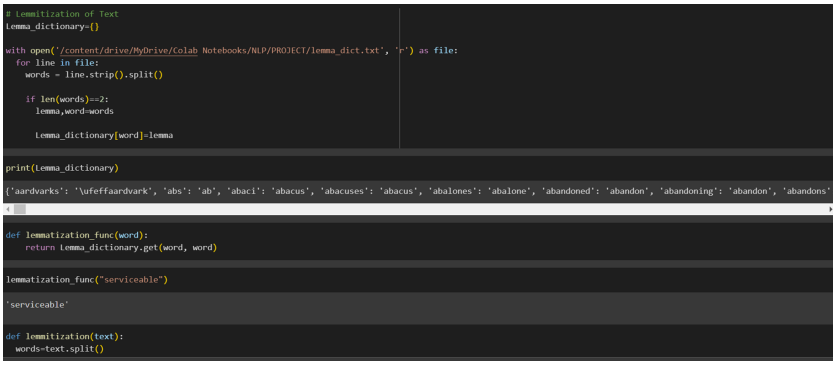


Figure 4: Lemmatisation.

8. Then we implemented NER(Name Entity Recognition)

	Phrase	Sentiment	Keywords	Positive_Score	Negative_Score	NER_Entities
0	series escapade demonstrate adage good goose a...	2	[good, series, escapade, demonstrate, adage]	1.4	0.0	[]
1	quiet introspective entertain independent wort...	5	[quiet, introspective, entertain, independent,...	0.7	0.0	[marvel]
2	even fan ismail merchant work suspect would ha...	2	[even, fan, ismail, merchant, work]	0.0	0.0	[]
3	positively thrill combination ethnography intr...	4	[positively, thrill, combination, ethnography,...	0.0	0.0	[disney]
4	aggressive self glorification manipulative whi...	2	[aggressive, self, glorification, manipulative...	0.0	0.0	[]

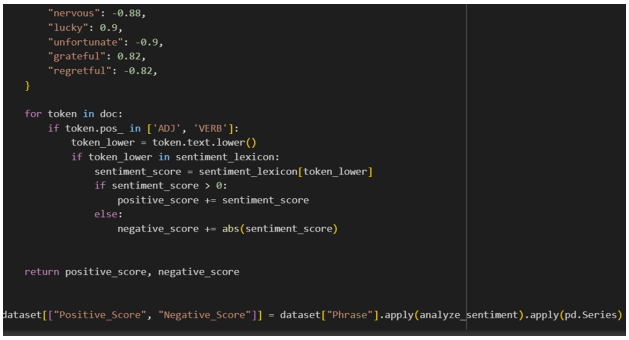
Figure 5: NER

9. keywords to a sentence are done to find keywords which will be used to build for POStagging dictionary

	Phrase	Sentiment	Keywords
0	series escapade demonstrate adage good goose a...	2	[good, series, escapade, demonstrate, adage]
1	quiet introspective entertain independent wort...	5	[quiet, introspective, entertain, independent,...
2	even fan ismail merchant work suspect would ha...	2	[even, fan, ismail, merchant, work]
3	positively thrill combination ethnography intr...	4	[positively, thrill, combination, ethnography,...
4	aggressive self glorification manipulative whi...	2	[aggressive, self, glorification, manipulative...

Figure 6: Keywords Extraction.

10. After this POStagging is done to add extra features



```
    "nervous": -0.88,
    "lucky": 0.9,
    "unfortunate": -0.9,
    "grateful": 0.82,
    "regretful": -0.82,
}

for token in doc:
    if token.pos_ in ['ADJ', 'VERB']:
        token_lower = token.text.lower()
        if token_lower in sentiment_lexicon:
            sentiment_score = sentiment_lexicon[token_lower]
            if sentiment_score > 0:
                positive_score += sentiment_score
            else:
                negative_score += abs(sentiment_score)


return positive_score, negative_score

dataset[["Positive_Score", "Negative_Score"]] = dataset["Phrase"].apply(analyze_sentiment).apply(pd.Series)
```

Figure 7: Postagging

11. After that we went for Bag of words,vectorization,and TF-IDF to convert it into a form which will be ready to use it train the data

12. As TF-IDF got more accuracy among all we considered it to implement it in our model

```
def calculate_tfidf(phrasess):
    tf_matrix = []
    for document in phrasess:
        words = document.lower().split()
        total_words = len(words)
        tf_dict = {}
        for word in set(words):
            tf_dict[word] = words.count(word) / total_words
        tf_matrix.append(tf_dict)

    idf_dict = {}
    total_documents = len(phrasess)
    for document in phrasess:
        words = set(document.lower().split())
        for word in words:
            idf_dict[word] = idf_dict.get(word, 0) + 1

    idf_matrix = {word: math.log(total_documents / (count + 1)) for word, count in idf_dict.items()}

    tfidf_matrix = []
    for tf_dict in tf_matrix:
        tfidf_dict = {word: tf * idf_matrix[word] for word, tf in tf_dict.items()}
        tfidf_matrix.append(tfidf_dict)

    return tfidf_matrix
```

Figure 8: TF-IDF

13. In model Training we used Naive Bayes Classifier and after Training we predicted the model using test data available to us

14. And we finally created a Pickle file which will store our project and it is used to take input and give the rating based on the sentiment of the given text

```
[ ] import pickle

[ ] with open('model_pickle_file','wb') as file:
        pickle.dump(model,file)

[ ] with open('model_pickle_file','rb') as file:
        mp = pickle.load(file)

[ ] X_test.to_pickle('data_columnset.pkl')

[ ] with open('data_columnset.pkl', 'rb') as file:
        loaded_dataframe = pd.read_pickle(file)
```

Figure 9: Pickle File

5 Results

After Building this model we got an accuracy of 65 percent and it predicts the given statement as shown below

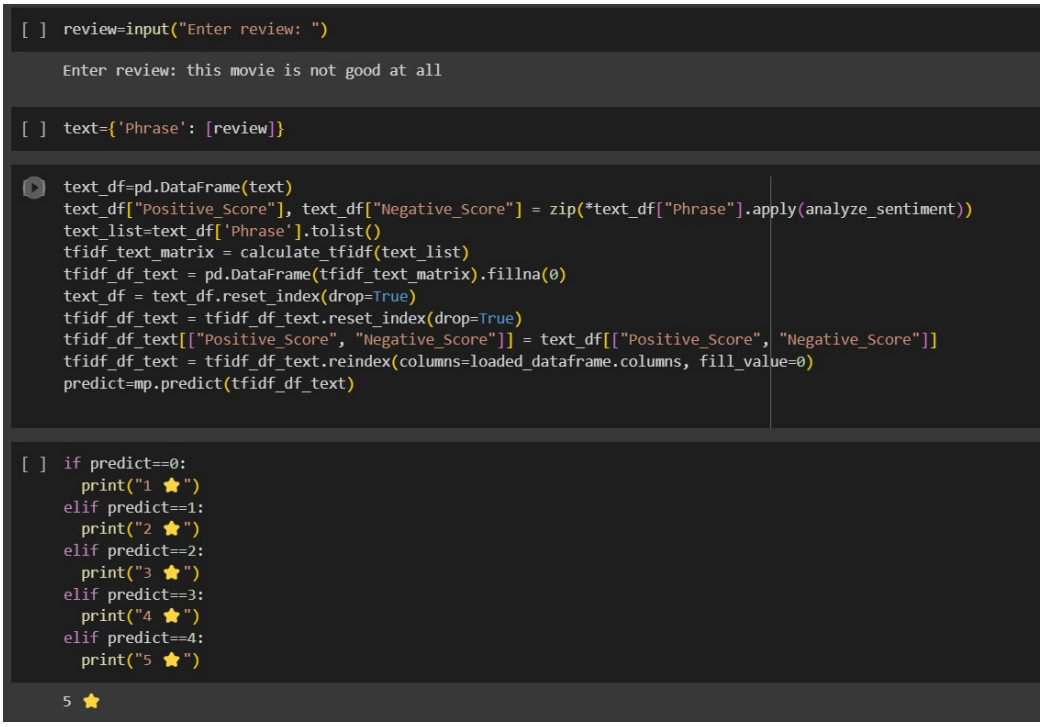


Figure 10: Result

6 Conclusion

From this Project we predicted the emotions of the review given by the users and given ratings.We have done this project in English Language only.But we can extend it other languages also mainly due to the impact of movie reviews in the field of movies.So In future we hope we can implement our Project in other languages also