**Reddit post sentiment analysis**

**A report on**

**Big Data Analytics Lab Project**

**[CSE-XXXX]**

Submitted By

**Harika Boppana-210962030**

**Harry Raj-210962022**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**April,2024**

**Reddit post sentiment analysis**

Harika Boppana1, Harry Raj2

¹ ²CSE department, MIT Manipal, India

1harika.boppana@learner.manipal.edu; 2harry.raj@learner.manipal.edu

Abstract*— The paper details our project about performing sentiment analysis on comments extracted from a reddit post. We aim to label a given post with a majority sentiment based on user’s comments made on the post. We utilize machine learning techniques to achieve this, including naive bayes, logistic regression, and random forest classifier. We also perform text processing and natural language processing tasks through concepts such as text frequency and inverse document frequency.*

Keywords***— sentiment analysis; multi-class classification; logistic regression; decision trees; web scraping; natural language processing; MLlib; PySpark.***

1. **Introduction**

As the internet grows to be easily accessible across the world, social media is becoming an inevitable part of most people's lives. There are usually two types of users of social media: the consumers and the creators. For the creators, their social media presence if often their primary or only source of income, so it becomes important for them to analyse how consumers feels about their content and how they are interacting with it. In our project, we aim to create an application that gives creators an insight into what people think of their content. The most obvious and nearly fool-proof way of knowing audience’s response is to directly take their opinions, which are available in the comments section of any social media post. Since it becomes tedious and nearly impossible to get an objectively correct idea about what the comments are like, we perform sentiment analysis and determine what the majority sentiment towards a post is. We utilize reddit for this, and aim to given redditors an accurate analysis about what the users that are commenting on their posts really feel about it. This is very valuable to posters, as it helps them understand their audience and what they like and dislike. A similar strategy can also be extended to other social media platforms such as twitter, Instagram ,YouTube, etc.

1. **Literature review**
2. **methodology**

We implement our project in five main stages:

1. **Data extraction from Reddit**

To extract data from reddit, we utilize the PRAW library. We extract comments for a given post in our project. For this, we input the link to the reddit post to the application and it automatically scrapes the latest 20 comments on the post.

1. **data preprocessing using NLP techniques:**

**2.1 removing stopwords**

The data extracted from any social media websites consists of redundancies that do not contribute to the text’s sentiment. It is important to remove all such redundancies in order to obtain an objective sentiment that is unaffected by irrelevant components. In the case of data from Reddit, one of the most common redundancies are "http","https","amp","rt","t" and "c". These undesirable components arise from the fact that users often refer to each other by their usernames or reply to a pre-existing comment and post. This results in the extraction of aforementioned data that represents user-to-user interaction. We remove all such data from our training data in order to train the model.

In addition to this, we also remove “stopwords”. Stopwords in NLP are those words that are grammatically a part of the language, but do not contribute to the sentiment of the whole sentence. Such words include: “I” “us” “we” “they” ,etc.

To remove all stopwords including the ones unique to reddit data, we utilize Mllib’s stopwords remover.

**2.2 vectorizing non-numerical data**

The models that we use to obtain outcomes are strictly only compatible with numerical inputs and labels. For this reason, we change the labels column in our original data to consist of -1,0 and 1 instead of negative, neutral and positive, respectively. Adding to this, we must also convert the text extracted to an equivalent vector form. To achieve this, we utilize a combination of Tokenizer,HashingTF, countVectorizer, idf and string indexer.

We first utilize tokenizer to split the input sentence into constituent words.

HashingTF  is a technique used to convert text data into numerical vectors. It converts text to vectors based on term frequency and using hashing of each token.The length of resultant vector is based on hashing function.

IDF (inverse document frequency) calculates the rarity of a given term. A higher IDF score indicates that a term is less common across documents in the corpus(language data), making it more valuable or informative for distinguishing between documents.

CountVectorizer is a tool used in natural language processing to convert a collection of text documents into a matrix where rows represent documents and columns represent individual words or n-grams. The values in the matrix indicate the frequency of each word or n-gram in each document.

We combine hashingTF and IDF to simulate TF-IDF vectorization.

Alternatively, to hashingTF, we use countvectorizer to convert text data to vectors. it has a few advantages as it doesnt depend on hashing function. it can handle collisions that may occur from hashingTF and makes it easier to interpret features with inverse-mapping.

We found that the combination of hashingTF-idf and countVectorizer-IDF have the exact same accuracies, which can be due to the nature of our chosen dataset.

General tranformaions pipeline used:

1. tokenizer, stop words remover, hashingTF, idf, indexer
2. tokenizer, stop words remover, count vectorizer, idf, indexer
3. **training model on a chosen dataset**

We use a sentiment analysis dataset [1] to train our models. The data is a CSV with emoticons removed and we consider the columns:

* text and sentiment from train data:

text column consists of preprocessed text where some unnecessary words are removed. Regardless, we still apply our pipeline of transformations.

* Selected text and sentiment from train data:

Selected text is the raw data without emoticons extracted from the post. We covert this be in a format we need to train the model.

* Text and sentiment from test data:

Text columns contains raw un-processed data labelled with sentiments.

We implement logistic regression and naive bayes. We implement logistic regresison in slightly different variations as:

1. logistic regression without regularization and elastic net parameters

2. logistic regression with regularization and net parameters.

Both models have different outcomes that we will discuss in later sections.

1. **testing model accuracy**

We test our models performance using multi class classification evaluator of Mllib and the metric accuracy.

1. **Experimental setup**

We have developed the entire project using Google Colab. to set up Google Colab for utilizing spark, we followed the resources in [2]. We have also browsed many datasets of kaggle [3] to find our optimal dataset.

1. **results and discussion**

|  |  |
| --- | --- |
| MODEL AND PIPELINE | OUTPUTS (test accuracy) |
| Hashingtf-idf-indexer-logistic regression | 54.41426146010186 |
| countvectorizer-idf-indexer-logistic regression | 54.41426146010186 |
| Processing pipeline and logistic regression with regParam and elasticNetParam | 66.1043988734368 |
| Naive bayes | 48.446191827494104 |

1. **Conclusions**

Based on the model performances, we conclude that the 3rd model works the best. We also implemented random forest classifier, but its accuracy was sub-optimal (around 20%) so we decided that it was not a good model for the application.

1. **Future work**

We envision to extend the project to analyze sentiments of other social platforms. We would extend our UI so that user can select a platform to analyze their posts. The feature of choosing multiple posts or their entire profile can also be integrated so that users can get an idea about what consumers think of their entire profile.

**Acknowledgement**

We express our gratitude to Dr Prakash Kalingrao Aithal for his dedicated guidance in navigating essential deep learning concepts, enabling us to construct the project. Existing research and study of the field was necessary for our own understanding and ability to apply concepts learned. We sincerely appreciate the authors of referred scientific papers for their valuable contributions directly to the field of computer science and indirectly to this project.

References  
[1] https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset

[2] medum article