Materials Today: Proceedings 65 (2022) 3333–3341

Contents lists available at ScienceDirect

Materials Today: Proceedings

journal homepage: www.elsevier.com/locate/matpr

Sentiment analysis using web scraping for live news data with machine learning algorithms

Parneet Kaur

Information Technology Department, Ajay Kumar Garg Engineering College Ghaziabad, India

ARTICLE INFO

Article history:
Available online 3 June 2022

Keywords:
Naïve Bayes
Multinomial Naïve Bayes
Logistic Regression
Precision
Reddit
Sklearn
PRAW

ABSTRACT

The web scraping approach extracts large portions of information in a short amount of time using its definition. Further, Web Scraping offers records retrieval, newsgathering, internet monitoring, aggressive advertising and marketing, and many more. The use of the internet for scraping makes getting access to a tremendous quantity of records online, smooth and simple. Like in manufacturing electronic and automobile industries product reviews and campaigning require customer feedback which can be easily gathered through web scraping of websites. It is quicker and less complicated than manually extracting records from websites. Web Scraping is turning into a great data access tool these days. Apart from web scraping, web crawling and data mining or web mining are also some of the areas or methods that permit the easy compilation and storage of information on the web. In this paper, some of the famous supervised machine learning algorithms like Naïve Bayes and Logistic Regression are implemented on the live news data to effectively check the impact of these algorithms on data. Further, the application of sentiment analysis is explored and analyzed by using machine learning algorithms. This naive approach helps in understanding the comments, blogs from famous news websites by dividing the opinions into various categories at great scale and depth. This paper aims to combine the supervised learning methods with web scraping techniques for deriving optimized results of news articles about accuracy, precision, and recall.

Selection and peer-review under responsibility of the scientific committee of the 2022 International Conference on Materials and Sustainable Manufacturing Technology.

1. Introduction

Opinion mining and sentiment analysis serve to analyze the emotional tone of a text using natural language processing (NLP). It has become a widely used way for businesses to decide and categorize reviews about a product, service, company, or idea. Also sentiment analysis act as a powerful tool for understanding the opinion whether related to news, product, or marketing campaigns. It plays a vital role when understanding the product and its brand, in analyzing customer satisfaction, reviews, and product acceptance. The term emotion-based marketing is a wide umbrella phrase that encompasses emotional customer responses, that is positive, negative, neutral, uptight, disgust, frustrated, and others. Thus, sentiment analysis assists in studying the psychology of customer responses to enhance product and brand recall. In this paper, the concept of sentiment analysis is applied to live news data. The source for the data is the Redditt political news of India from the Reddit link[1]. The reason for analyzing the news information is simple since online news websites relegate our valuable decision-making and action into predefined notions generated by others. Therefore, it seems interesting and inspiring to track what and how trends in such news websites change over time. Furthermore, newspaper articles do not use positive or negative language to convey objectivity. They embed statements in a more complex discourse to express what they believe, instead of showcasing a few important people. One of the most common problems encountered in detecting negative linguistic statements in different domains of documents, such as query analysis, sentiment analysis, tasks comprehension, medical data mining, relation extraction, and many others. This paper aims to evaluate, compare and analyze sentiment analysis on a live news dataset using supervised classification learning algorithms that are Multinomial Naive Bayes and Logistic Regression. Furthermore, Sklearn libraries were utilized to identify negation in news articles to understand the sentiment of online news archives.

E-mail address: kaurparneet@akgec.ac.in

https://doi.org/10.1016/j.matpr.2022.05.409
2214-7853/
Selection and peer-review under responsibility of the scientific committee of the 2022 International Conference on Materials and Sustainable Manufacturing Technology.

1.1. Pattern of paper

The overall paper is organized as follows: The Introduction paragraph is comprised of the main idea that what is Sentiment analysis and what are its major application domains. In Related Work, the background and history of work done in the sentiment analysis domain are highlighted. Some of the main contributions in this area are presented. Then in sections 3 and 4, the problem statement is formulated and stages of implementation in form of phases are showcased. In the Result analysis section, the outcome of the overall analysis is presented with the help of coding snippets along Manuscript Click here to view linked References with Tables of comparison between the algorithms. Finally, the conclusion of the work is provided some of the relevant areas left for future work in the sentiment analysis domain.

2. Related work

Researchers have used a variety of approaches to analyze news sentiment. This section provides a brief discussion of previous sentiment analysis work. Currently, with the advent of the digital era, we can observe a huge increase in usergenerated content on the web, which provides opinions of people on various topics. This is known as Sentiment Analysis. Sentiment analysis has been studied for decades [8]. Reis, Olmo Benevenuto, Prates and An [3] discover the relationship between news popularity and sentiment polarity. An analysis was conducted using the content of 69,907 headlines generated by four of the most prestigious media outlets: The New York Times, BBC, Reuters, and DailyMail. In this work, a research study used text features for analyzing the sentiment polarity. News headlines that are negative or positive gain greater interest than neutral news headlines. Further work done by Wiebe in 1994 [2], showcase subjective text in form of a linguistic expression of an individual's beliefs, opinions, feelings, evaluations, and speculations. Also, the work of linguist Ann Banfield (1982) [4] highlights that an experiencer's perspective and documents of private states were based on sentences that were not open to objective observation or verification. According to Esuli and Sebastiani (2006) [5], Sentiment Analysis is an emerging field that is formed with the intersection of Information Retrieval and Computational Linguistics that focus on the opinions expressed in that file. Based on sentiment lexicons, Godbole, Srinivasaiah, and Sekine [6] developed an algorithm that explores sentiment words findings and entities related to the text. An approach was designed to classify online news in the paper by Islam, Abir, Ashraf, and Mottalib [7]. Positive and negative words in a dictionary were used to find sentiment polarity. Zhou et. al.[9] implemented the ML algorithms for sentiment classification, the Naive Bayes, Maximum Entropy, and Support Vector Machine on the IMDB movie data. Their research shows that progressive classification can be used to achieve accuracy. In another paper by Pang and Lee et.al.[10] the supervised classification algorithms like Naïve Bayes, Maximum Entropy Classification, and Support Vector Machine were applied to movie reviews. Based on the above research the introduction of new techniques was started such as Deep Learning systems and Neural Network techniques. In the research by Dong and Wei [11] adaptive recursive Neural Network (AdaRNN) was applied to the Twitter data for sentiment analysis. Alm et al. (2005) in their work [12] explored automatically the task of emotion classifications in the textual data. Agarwal, Sharma et.al. [13] proposed an opinion mining model using python packages for the classification of words and SentiWordNet 3.0. Their work highlighted the positive and negative words to find the impact i.e. positive or negative sentiment in news headlines. Similarly, Lei, Rao, et.al. [14] proposed the social emotion detection model for news archives and tweets. In their model, the document selection, tagging of speech, and lexicon for social emotions were derived. Further, Vyas and Uma [15] in their work applied sentiment analysis to customer data which requires extra care so that the customer satisfaction level improves the sales. Dang et al. [16] (2020) analyzed the Deep Neural Network (DNN) and Convolutional Neural Network (CNN) on eight datasets and derived sentiments related to them. In the research by Samuel, Hamza et.al. [17] Text pattern matching technique was applied for extracting the data through HTML tags. These HTML tags were further scraped for identifying the news-related tags for the analysis.

3. Problem formulation

Today, news organizations have increasingly relied on media analytics to attract and retain readers. Due to COVID19 and other pre-pandemic trends, ad revenue has fallen dramatically this year. News companies need to know which articles resonate with readers and which don't. In light of this, an effort is made in this paper to know what makes a news article popular or unpopular through analysis. According to various features, including word count and headline & abstract length, we analyzed Reddit online news [1] to predict news popularity. Today, media companies have to know which news articles to publish or not. In this paper, we performed exploratory data analysis, to pick out trends and patterns, and figure out what factors to consider for a machine learning model. As in the paper by Khemani et. al. [20] the sentiment analysis is done for the news articles but the further classification of analyzed data using machine learning technique was missing. Hence, to cover drawbacks of previous work, an effort is made by applying supervised-based classification algorithms such as Naïve Bayesian, Regression-based algorithms in this paper, as it provides better accuracy, efficiency, and interpretation of results.

4. Research Methodology/ implementation

In this paper, the methodology for analysing the sentiment analysis of news headings is the Lexicon approach with Machine Learning based Supervised Learning algorithms are developed. In general, the sentiment analysis is carried out by applying both supervised and unsupervised methods. Supervised learning methods are applied to the labeled type sentiment documents. This data is designed to train the algorithms for the accurate classification of results. Supervised learning is further divided into two broad categories: classification and Regression. The classification algorithms accurately classify the datasets into desired sections or outputs. These algorithms are Linear classifiers, support vector machines, decision trees, and random forests. The Regression methods are used for prediction purposes. This algorithm explores the relationship between the dependent and independent elements. Linear, Logistic, and Polynomial Regression are some of the most popular algorithms for regression. Whereas the Unsupervised also known as the Lexicon-based approach for sentiment analysis is applied to the unlabeled data as it due to non-dependency on training data. Further, the algorithms in the latter are categorized as Clustering, Association, and Dimensionality reduction. Clustering is a technique that groups together the unlabeled datasets based on similarity or differences. For example, the K-means clustering. Whereas, the association is primarily based on a rule-based approach for finding a relationship in data. In the Dimensionality algorithm, the technique is only applied in case when the dataset is having high dimensions or features as compared to other datasets. The sentiment analysis on textual data can be applied in various manners like on document level, sentiment level, phrase, or even at word level [1819] with the implementation of

Dictionary-based methods. In the dictionary-based methods, the lexicon dictionary is implemented for finding the opinion words like positive, negative, or neutral. Similar to this approach Corpus-based-Approach is also done on a large corpus of datasets. This technique identifies the syntactic patterns in the corpus of words. To get started with Sentiment Analysis for Live news with Machine Learning following Algorithm is proposed which is as followed:

Step 1. Identification of the kind of information required for analysis is done.

Step 2. Identification of reliable news websites is done which included filtration of illegal websites while scraping the data.

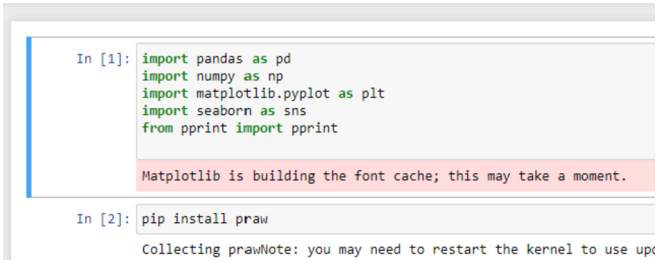

Fig. 1. Library import.

Fig. 2. PRAW API client_ID formation.

Step 3. For this analysis, the reddit.com news website is processed.

Step 4. Identification of packages such as the Natural Language Tool Kit Python package which includes the Sentiment Intensity Analyzer is installed, Pandas, Numpy, Matplotlib, Seaborn libraries are done as per the data requirement.

Step 5. An appropriate API Wrapper is installed. (For this analysis the PRAW API wrapper is used).

Step 6. Labeling of data is done using the NLTK tool kit.

Step 7. Dataset creation and Statistic Information analysis are performed.

Step 8. Implementation of tokenization and word count is performed.

Step 9. The sentiment polarity score is calculated for data perimeters.

Step 10. Normalization of final data is done and frequency count for words that represents sentiments like positive, negative, and neutral is derived.

Step 11. Data derived is converted into the.csv format for further analysis.

Step 12. The Logistic Regression and Multinomial Naïve Bayes classifier is built using the SKLearn library for the derived dataset.

Step 13. Afterward splitting the dataset into training and the testing dataset is constructed.

Step 14. Feature extraction using various methods like binary, TF-IDF, and count-based methods is being carried done.

Step 15. Performance evaluation in terms of Accuracy and MRR is performed for all feature extraction methods.

Step 16. The performance for Logistic Regression and Multinomial Naïve Bayes algorithms is compared and an optimized algorithm is generated. Identification of the kind of information required for analysis is done.

5. Result analysis

The result formation in this paper is performed through various phases. These phases include the following steps:

Phase 1: Importing of NLTK libraries and Python Reddit API Wrapper also known as PRAW as shown in Fig. 1. In Fig. 2, the

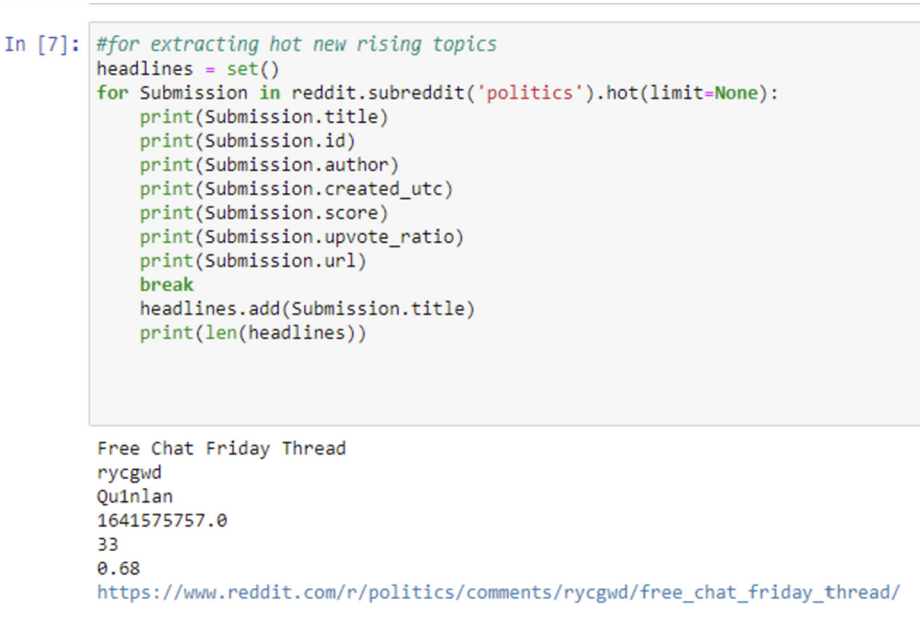

Fig. 3. Extraction of topics.

app is created with API which includes a valid client_id and client_secret key.

Phase 2: Afterward, the PRAW libraries are imported into python. To work with PRAW, the client is created. With API wrapper PRAW thousands of news headlines from various news subreddit are downloaded and JSON responses are also handled. The set for these headlines is derived to remove duplicate entries.

Phase 3: In Fig. 3, each sentiment is created into a result set list which comprised of title, id, author, score, upvote ratio and score of each headline. The length of each headline is calculated using set function as shown in Fig. 4. Further, this list is transformed into a dataframe as given in Fig. 5.

Phase 4: During phase 4, the NLTK, Vader Sentiment Analyzer is imported which gives ranking in terms of positive, negative, or neutral using a lexicon of positive and negative words on text. The Sentiment Intensity Analyzer (SIA) categorizes the headlines. Further, sentiments are derived using the polarity_scores method as displayed in Fig. 6.

Phase 5: In Fig. 7, adding the sentiment dictionary to a results list transforms the list into a data frame. From the sentiment scoring, there are four columns: Neu (for neutral), Neg (for negative), Pos(for positive), and compound. The first three represent the sentiment score percentage headline w.r.t. category. The compound column ranges represent values −1as Extremely Negative and + 1 as Extremely Positive. Further, the range is computed that represents the compound value as greater than 0.2 for + 1 label and less than −0.2 for −1 label else the values are counted as 0.

Phase 6: In Fig. 8, line number 28 represents the raw value counts for labels, and line number 29 displays the normalized keyword percentage. Thus, the positive, negative, and neutral sentiments are converted into percentage.

```
In [8]: headlines = set()
        for Submission in reddit.subreddit('politics').hot(limit=None):
            headlines.add(Submission.title)
            print(len(headlines))
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
```

Fig. 4. Headline Length calculation.

```
In [9]: df = pd.DataFrame(headlines)
        df.head()
```

Out[9]:

	0
0	Texas Senator Ted Cruz To Introduce Bill To Ov...
1	Conservatives clash: Reps. Marjorie Taylor Gre...
2	U.S. attorney general to discuss investigation...
3	'We're Ashamed of Nothing': Matt Gaetz and Mar...
4	Trump's border wall and the slow decay of Amer...

```
In [10]: df.to_csv('redditheadlines.csv', header = False, encoding='utf-8',index = False)

In [11]: pip install nltk
```

Fig. 5. Dataframe formation.

```
In [21]: from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
         sia = SentimentIntensityAnalyzer()


         results = []

         for line in headlines:
             pol_score= sia.polarity_scores(line)
             pol_score['headline']= line
             results.append(pol_score)

             pprint(results[:3], width=100)
         [{'compound': 0.0,
           'headline': 'Texas Senator Ted Cruz To Introduce Bill To Overturn Vaccine Mandate For School '
                       'Kids In D.C.',
           'neg': 0.0,
           'neu': 1.0,
           'pos': 0.0},
          {'compound': -0.4215,
           'headline': 'Conservatives clash: Reps. Marjorie Taylor Greene, Dan Crenshaw trade social media '
                       'insults',
           'neg': 0.203,
           'neu': 0.797,
```

Fig. 6. Polarity score calculation.

```
In [23]: df = pd.DataFrame.from_records(results)
         df.head()
```

Out[23]:

	neg	neu	pos	compound	headline
0	0.000	1.000	0.00	0.0000	Texas Senator Ted Cruz To Introduce Bill To Ov...
1	0.203	0.797	0.00	-0.4215	Conservatives clash: Reps. Marjorie Taylor Gre...
2	0.279	0.721	0.00	-0.4767	U.S. attorney general to discuss investigation...
3	0.146	0.704	0.15	0.0258	'We're Ashamed of Nothing': Matt Gaetz and Mar...
4	0.231	0.769	0.00	-0.4019	Trump's border wall and the slow decay of Amer...

```
In [25]: # label =1 means positive, -1 means negative
         df['label']=0
         df.loc[df['compound']>0.2, 'label']=1
         df.loc[df['compound']<-0.2, 'label']=-1
         df.head()
```

Out[25]:

	neg	neu	pos	compound	headline	label
0	0.000	1.000	0.00	0.0000	Texas Senator Ted Cruz To Introduce Bill To Ov...	0
1	0.203	0.797	0.00	-0.4215	Conservatives clash: Reps. Marjorie Taylor Gre...	-1
2	0.279	0.721	0.00	-0.4767	U.S. attorney general to discuss investigation...	-1

Fig. 7. Sentiment dictionary formation.

```
In [26]: df2=df[['headline','label']]

In [27]: df2.to_csv('reddit_headlines_label.csv', encoding='utf-8',index = False)

In [28]: # -1 means negative, 0 means neutral and 1 means positive
         df.label.value_counts()

Out[28]: -1    334
          0    307
          1    136
         Name: label, dtype: int64

In [29]: df.label.value_counts(normalize=True)*100

Out[29]: -1    42.985843
          0    39.510940
          1    17.503218
         Name: label, dtype: float64
```

Fig. 8. Count value of sentiments.

```
In [16]: print("Positive headlines:\n")
         pprint(list(df[df['label']==1].headline)[:5], width=200)

         print("Negative headlines:\n")
         pprint(list(df[df['label']== -1].headline)[:5], width=200)

         Positive headlines:

         ['Senate Democrats grow less confident in Manchin',
          'Three in four voters support banning lawmakers from trading stocks: poll',
          'Supreme Court weighs vaccine rules affecting more than 80M',
          'Eric Adams' Top Criminal Justice Advisor Left NYPD Under a Cloud',
          "Kevin McCarthy's ex-staffer says GOP Rep. Mo Brooks was 'cheering' as rioters stormed the Capitol"]
         Negative headlines:

         ["Gregg Brelsford, Alaska's U.S. House candidate, leaves GOP citing attacks on democracy",
          'A Year Later, We're Still Reluctant to Say What Really Caused the Insurrection',
          'Richard H. Clarida announces his intention to resign from the Board of Governors of the Federal Reserve System on January 14,
         2022'.
```

Fig. 9. Positive Headlines.

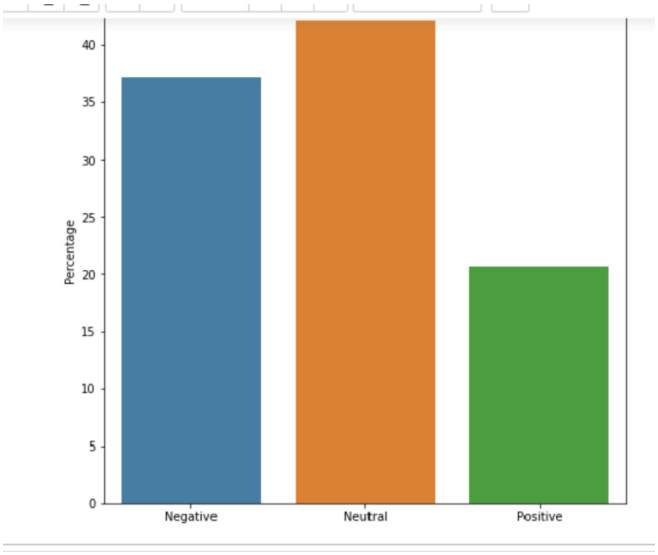

Fig. 10. Bar Plot for sentiments.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        from sklearn.feature_extraction.text import CountVectorizer
        from sklearn.feature_extraction.text import TfidfTransformer
        from sklearn.naive_bayes import MultinomialNB
        from sklearn.linear_model import LogisticRegression
        from sklearn.pipeline import Pipeline
        from sklearn import metrics
        import itertools

In [2]: data=pd.read_csv('reddit_headlines_label_1.csv')
```

Fig. 11. Import of Libraries.

Phase 7: During this phase, the positive and negative headlines are generated separately as shown in Fig. 9.

Phase 8: Fig. 10 display the bar plot of Negative, Neutral, and Positive sentiment percentages of headlines.

Phase 9: During this phase, the implementation of machine learning algorithms on the final dataset stored with the name reddit_headlines_label1.csv is processed. The classifier is built by importing the sklearn library which includes train_test split, CountVectorizer, and TfidfTransformer feature extraction functions. The Multinomial and Logistic Regression Modules are imported from sklearn for implementation of Machine learning algorithms on the derived dataset as shown in Fig. 11.

```
In [3]: #Analysing the structure of the data given to us
        data.head(10)

Out[3]:
```

	headline	label
0	North Carolina sets goal to sell 50% zero-emis...	0
1	Gregg Brelsford, Alaska's U.S. House candidate...	-1
2	A Year Later, We're Still Reluctant to Say Wha...	-1
3	Georgia prosecutor says decision on Trump elec...	0
4	Senate Democrats grow less confident in Manchin	1
5	Richard H. Clarida announces his intention to ...	-1
6	Furious Fauci Tears Into Rand Paul for Incitin...	-1
7	Home COVID tests to be covered by insurers sta...	0
8	Justice Dept. forms new domestic terrorism uni...	-1
9	Three in four voters support banning lawmakers...	1

Fig. 12. Top 10 List generation.

```
In [4]: #Getting the counts of each category
        data["label"].value_counts()

Out[4]:  0    316
        -1    278
         1    155
        Name: label, dtype: int64
```

Fig. 13. Total category wise count.

Phase 10: The code in line 3 gives total count of headlines with their sentiment label value and the total count of each category of sentiment in line 4 as shown in Figs. 12 and 13.

Phase 11: Fig. 14 shows the split function of the dataset into train and test data.

Phase 12: In this phase, the coding in Fig. 15, shows the news headlines are classified as negative, positive, and neutral using classification algorithms Multinomial Naive Bayes. First, the term frequencies and count vectorizer are applied which consider the input attributes for the classification model and output attribute as target values. The pipeline concept integrates the count vectorizer, TF-IDF, and classification model. Further, automation of workflow is performed with a machine learning pipeline. Also, data is enabled in series for transformation and correlated together in a model that is tested and evaluated to determine whether an outcome is positive or negative. The classification report of the Multinomial Naïve Bayes classifier is derived which reflects the Accuracy as 59%.

```
In [10]: X=data["headline"]
         y=data["label"]
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

In [11]: #Calculating the number of rows in our train set
         len(y_train)

Out[11]: 524

In [12]: y_train.value_counts().plot(kind='pie', labels=names, autopct='%1.0f%%', subplots=True, figsize=(8, 8))

Out[12]: array([<AxesSubplot:ylabel='label'>], dtype=object)
```

Fig. 14. Train and Test split function.

```
In [13]: text_clf = Pipeline([('vect', CountVectorizer()),
                              ('tfidf', TfidfTransformer()),
                              ('clf', MultinomialNB()),
                              ])

In [14]: text_clf = text_clf.fit(X_train, y_train)
         predicted1 = text_clf.predict(X_test)

In [15]: metrics.accuracy_score(y_test, predicted1)

Out[15]: 0.5866666666666667

In [16]: print(metrics.classification_report(y_test, predicted1, target_names=sort
```

	precision	recall	f1-score	support
Negative	0.63	0.63	0.63	81
Neutral	0.55	0.80	0.65	96
Positive	1.00	0.08	0.15	48
accuracy			0.59	225
macro avg	0.73	0.51	0.48	225
weighted avg	0.67	0.59	0.54	225

Fig. 15. Metrics Implementation for Multinomial Naïve Bayes.

```
In [6]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(news_data['headline'], label

In [9]: import sklearn
        from sklearn.preprocessing import scale
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split
        from sklearn import metrics
        from sklearn import preprocessing
        from sklearn.feature_extraction.text import TfidfTransformer, TfidfVectorizer
        tfidf_vect=TfidfVectorizer(stop_words='english',max_df=0.25)
        tfidf_train=tfidf_vect.fit_transform(X_train)
        tfidf_test=tfidf_vect.transform(X_test)

In [10]: from sklearn.linear_model import LogisticRegression
         logmodel = LogisticRegression()
         logmodel.fit(tfidf_train,y_train)

Out[10]: LogisticRegression()
```

Fig. 16. Library import for Logistic Regression.

Phase 13: During this phase, the dataset named reddit_headlines_label_1.csv is fed to the machine learning Logistic Regression model. The Dataset is segregated into training and test set. Further, the dataset is comprised of both text features (TFIDF vectorized text data). The test size dataset is taken as 25% and the remaining 75% is train data for implementation of the Logistic Regression classifier model as shown in Fig. 16. In Fig. 17, the news headlines with labels are taken as the target value and the count vectorizer, the TF-IDF model is bonded with the Logistic Regression to form a machine learning model. Lastly, the accuracy is computed with this model over the validation set. In Table 2, the accuracy of Logistic Regression is displayed which came out as 61% which is slightly higher than the Naïve Bayesian Algorithm which is 59% as shown in Table 1. The accuracy is the number of times the model predic-

```
In [10]: from sklearn.linear_model import LogisticRegression
         logmodel = LogisticRegression()
         logmodel.fit(tfidf_train,y_train)

Out[10]: LogisticRegression()

In [12]: predictions = logmodel.predict(tfidf_test)
         predictions

Out[12]: array([ 0, -1,  0, -1, -1, -1,  0,  1,  0,  0, -1,  0, -1,  0,  0,  0, -1,
                -1, -1,  0,  0,  0,  0,  0,  0, -1,  0, -1,  0, -1,  0,  0, -1, -1,
                -1,  0,  1,  1, -1, -1, -1,  0, -1,  0,  0,  0,  0, -1,  0,  0,  0,
                 1,  0,  0, -1,  0,  0,  0, -1,  0, -1,  0, -1, -1,  0,  0,  0, -1,
                 0,  1,  0,  0, -1,  0,  0, -1,  0, -1,  0,  0,  0,  0, -1, -1, -1,
                -1, -1, -1,  0,  0, -1,  1,  0,  1, -1,  0,  0,  0,  0,  0,  1, -1,
                 0,  0,  1,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0, -1, -1, -1,  1,
                -1, -1,  0, -1, -1,  0, -1, -1,  0,  0,  0,  0,  0,  0,  0, -1, -1,
                 0,  0, -1,  0, -1,  0,  1, -1,  0, -1,  0, -1,  0,  0],
               dtype=int64)

In [13]: from sklearn.metrics import classification_report
         print(classification_report(y_test, predictions))

                      precision    recall  f1-score   support
```

Fig. 17. Train and Test split function.

Table 1
Metrics implementation for naïve Bayesian.

	Precision	Recall	F1-Score	Support
−1	0.63	0.63	0.63	81
0	0.55	0.80	0.65	96
1	1.00	0.08	0.15	48
Accuracy			0.59	225
Macro avg	0.73	0.51	0.48	225
Weighted avg	0.67	0.59	0.54	225

Table 2
Metrics implementation for Logistic Regression.

	Precision	Recall	F1-Score	Support
−1	0.58	0.64	0.61	50
0	0.61	0.73	0.66	70
1	0.82	0.30	0.44	30
Accuracy			0.61	150
Macro avg	0.67	0.56	0.57	150
Weighted avg	0.64	0.61	0.60	150

tion matches with the true labels over the number of labels in the validation set.

6. Conclusion

In this paper, a sentiment classifier is built that can detect sentiment expressed through the news on a three-class scale. The model is trained and tested on two common classification algorithms – Logistic Regression and Naive Bayes. Finally, a comparison is made to check the performance of these two algorithms and some test predictions were made. The analysis of sentiments in various news headlines cum statements are being carried out concerning analysing the effects of positive, negative, or neutral statements. Additionally, the work reflects several intermediate tasks that were completed to create a dataset for analysis. As Reddit is a community-based social network platform composed of multiple threads with sub-threads thus it provides the best data source for this paper. Similarly, Supervised machine learning provides a platform for accurate detection of news text. Further, optimization for the performance metrics of news data set with sentiments is implemented using supervised algorithms. Finally, it is concluded from the above analysis that the Logistic Regression algorithm performs relatively better as compared to the Naive Bayes algorithm as shown in Table 1 and Table 2. Since the Logistic Regression came out with very few undertakings as compared to the Multinomial Naive Bayes algorithm. The future scope is to reveal the secrets of NLP with some heuristic algorithms. By extending datasets to analyze international viewpoints, NLP techniques can also be used to create new functions. To obtain more realistic results, techniques like Latent Dirichlet Allocation (LDA), POS tagging can also be explored.

CRediT authorship contribution statement

Parneet Kaur: Conceptualization, Methodology, Investigation, Formal analysis, Visualization, Writing – original draft.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: PARNEET KAUR reports article publishing charges was provided by Ajay Kumar Garg Engineering College.

References

[1] https://www.reddit.com/r/politics/.
[2] Wiebe, J., Tracking point of view in narrative. In Computational Linguistics, volume 20, 1994.
[3] J. Reis, P. Olmo, F. Benevenuto, H. Kwak, R. Prates, and J. An. Breaking the news: first impressions matter on online news. In ICWSM '15, 2015.
[4] Banfield, A. Unspeakable sentences: Narration and Representation in the Language of Fiction, 1982.
[5] Esuli, A. and F. Sebastiani. SentiWordNet: A Publicly Available Resource for Opinion Mining.‖ In Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2006, Italy.
[6] N. Godbole, M. Srinivasaiah, and S. Sekine. Large-scale sentiment analysis for news and blogs. In International Conference on Weblogs and Social Media, Denver, CO, 2007.
[7] M. U. Islam, F. B. Ashraf, A. I. Abir and M. A. Mottalib, "Polarity detection of online news articles based on sentence structure and dynamic dictionary," 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, pp. 1-5. doi: 10.1109/ICCITECHN.2017.8281777; 2017.
[8] A. Ligthart, C. Catal, B. Tekinerdogan, Systematic reviews in sentiment analysis: a tertiary study, Artif. Intell. Rev. 54 (7) (2021) 4997–5053, https://doi.org/10.1007/s10462-021-09973-3.
[9] P. Chaovalit, and L. Zhou. Movie review mining: A comparison between supervised and unsupervised classification approaches. In Proceedings of the

38th annual Hawaii international conference on system sciences, pp. 112c-112c, January 2005, doi: 10.1109/HICSS.2005.445.
[10] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. arXiv preprint cs/0205070 (2002).
[11] L.i. Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification, in: In Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers), 2014, pp. 49–54.
[12] C.O. Alm, D. Roth, R. Sproat, Emotions from text: machine learning for text-based emotion prediction, in: In Proceedings of human language technology conference and conference on empirical methods in natural language processing, 2005, pp. 579–586.
[13] A. Agarwal, V. Sharma, G. Sikka, R. Dhir, Opinion mining of news headlines using SentiWordNet., Symposium on colossal data analysis and networking (CDAN), Indore 2016 (2016) 1–5, https://doi.org/10.1109/CDAN.2016.7570949.
[14] J. Lei, Y. Rao, Q. Li, X. Quan, L. Wenyin, Towards building a social emotion detection system for online news, Future Generation Comput. Syst. 37 (2014) 438–448.
[15] V. Vyas, V. Uma, Approaches to sentiment analysis on product reviews, Sentiment analysis and knowledge discovery in contemporary business. IGI Global, Pennsylvania, 2019, pp. 15–30.
[16] N.C. Dang, M.N. Moreno-García, F. De la Prieta, Sentiment analysis based on deep learning: a comparative study, Electronics 9 (3) (2020) 483.
[17] Salem, Hamza, and Manuel Mazzara. Pattern Matching-based scraping of news websites. In Journal of Physics: Conference Series, vol. 1694, no. 1, p. 012011. IOP Publishing, 2020.
[18] A. Dandrea, F. Ferri, P. Grifoni, T. Guzzo, Approaches, tools and applications for sentiment analysis implementation, Int. J. Comput. Appl. 125 (3) (2015) 26–33.
[19] M. Devika, C. Sunitha, A. Ganesh, Sentiment analysis: a comparative study on different approaches", Procedia Comput. Sci. 87 (2016) 44–49.
[20] B. Khemani, A. Adgaonkarb, A Review on Reddit News Headlines with NLTK tool, in: Proceedings of the International Conference on Innovative Computing & Communication (ICICC), 2021, 2021,, https://doi.org/10.2139/ssrn.3834240.