K L E F

KONERU LAKSHMAIAH EDUCATION FOUNDATION

(Deemed to be university estd, u/s, 3 of the UGC Act, 1956)

(NAAC Accredited "A++" Grade University)

A Project Based Lab Report

On

Sentiment analysis of top colleges in India using twitter data

Submitted in partial fulfilment of the

Requirements for the award of the Degree

Of Bachelor of Technology

IN

Computer Science and Engineering

UNDER THE ESTEEMED GUIDANCE OF

by

I.D NUMBER	NAME
2000030810	Madhav PVL
2000030819	Prahasith Naru
2000030824	Prince Kumawat

(DST-FIST Sponsored Department)

KL Education Foundation

Green Fields, Vaddeswaram, Guntur District-522 502

2022-2023

KL EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENIGNEERING
(DST-FIST Sponsored Department)

CERTIFICATE

We here by declare that this project-based lab report entitled "**Sentiment analysis of top colleges in India using twitter data**" has been prepared by us in the course **20CS3269AA Deep Learning** in partial fulfilment of the requirement for the award of Degree bachelor of technology in **COMPUTER SCIENCE AND ENGINEERING** during the even Semester of the academic year 2022-2023. We also declare that this project-based lab report is of our own effort and it has not submitted to any other university for the award of any degree.

Date :

Place:

Signature of the Student

Student Name	Id Number
2000030810	Madhav PVL
2000030819	Prahasith Naru
2000030824	Prince Kumawat

KL EDUCATION FOUNDATION

DEPARTMENT OF COMPUTER SCIENCE AND ENIGNEERING
(DST-FIST Sponsored Department)

CERTIFICATE

This is to certify that the project based laboratory report entitled "**Sentiment analysis of top colleges in India using twitter data**" is a bonafide work done Madhav PVL, Prahasith Naru, Prince Kumawat bearing Regd.No 2000030810, 2000030819, 2000030824 to the course **20CS3269AA Deep Learning** in partial fulfillment of the requirements for the award of Degree in Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING during the Even Semester of Academic year 2022-2023

FACULTY IN CHARGE   **HEAD OF THE DEPARTMENT**

 s DR.SAGAR IMAMBI   SENTHIL . A

ACKNOWLEDGEMENTS

Name: Madhav PVL

Reg. No: 2000030810

Name: Prahasith Naru

Reg. No: 2000030819

Name: Prince Kumawat

Reg. No: 2000030824

INDEX

ABSTRACT

Sentiment analysis is a useful technique for understanding people's opinions and emotions expressed in text data. In this project, we will perform sentiment analysis on Twitter data to analyse the opinions and sentiments of users about the top colleges in India. We will collect tweets of colleges based on the National Institutional Ranking Framework (NIRF) rankings. The results of the study provide insights into the perception of people towards the top colleges in India and can be useful for college administrators to improve their reputation management and understand the areas where they need to focus their efforts. Further this gives the results by applying machine learning algorithms like Support Vector Machine, Artificial Neural Network Model and Multi-Layer Perceptron, and deep learning algorithms like Recurrent Neural Network.

INTRODUCTION

Sentiment analysis is an analytical method for identifying the emotional content of communications. It combines machine learning, natural language processing, and statistics. Organizations evaluate content such as social media posts, online reviews, contact center interactions, and customer messages using sentiment analysis.One of the most well-known tweeting platforms, Twitter has more than 300 million active users.

With the proliferation of social media platforms, people have an outlet to express their opinions and sentiments towards various entities, including educational institutions. Twitter, in particular,has become a popular platform for people to share their views on different topics, including colleges and universities. Sentiment analysis on Twitter data has emerged as a powerful technique to analyze public opinion towards various topics, including educational institutions.

Sentiment analysis, also known to it as opinion mining, for identifying certain words as good or bad.Every day, Twitter users publish (or "tweet") on a variety of topics, including goods, services,routine activities, locations, people, etc.Twitter Sentiment Analysis (TSA) deals with the problem of understanding tweets in the context of the sentiment they express.Twitter offers users a huge platform to learn more and receive direct feedback about a good or service they are interested in.

The study aims to analyze the sentiment of people towards the top colleges in India and explore the most common themes associated with the sentiment. The study utilizes Twitter data collected using various search terms related to the colleges and applies natural language processing techniques to classify the tweets into positive, negative, or neutral sentiment. The study also investigates the relationship between the sentiment of tweets and the characteristics of the colleges, such as their ranking, location, and student population. The topic of SA has been the subject of a great deal of composing, and more recently, significant attention has been paid to TSA.

Here the process involves some of the steps for the implementation

Import Libraries based on requirements:-

For example pandas for data manipulation and analysis, numpy for numerical computations and array operations, matplotlib and seaborn for data visualisation and sklearn for machine learning tasks such as feature extraction,model training, and evaluation.

Loading of Dataset:-

This assumes that the dataset is stored in a CSV file named 'tweets.csv' in the current directory. You can replace the file path with the actual path to the file if it's stored in a different location.

Data Collection and Preprocessing:-

The data for the study is collected using various search terms related to the top colleges in India. The tweets are collected using the Twitter API and stored in a database. The data is then preprocessed by removing stop words, hashtags, mentions, and URLs to clean the text data. The tweets are then tokenized and lemmatized to prepare them for feature extraction.

Feature Extraction and Sentiment Classification:-

The feature extraction process involves converting the text data into numerical features that can be used for sentiment classification. The bag-of-words approach is used to represent the tweet text as a vector of features. The features are then used to train a support vector machine (SVM) classifier to classify the tweets into positive, negative, or neutral sentiment.

Exploratory Data Analysis:-

The study also investigates the relationship between the sentiment of tweets and the characteristics of the colleges, such as their ranking, location, and student population. Exploratory data analysis techniques are used to visualize and

analyze the data to gain insights into the relationship between the sentiment of tweets and the characteristics of the colleges.

Results and Discussion:-

The results of the study provide insights into the sentiment of people towards the top colleges in India. The study reveals that the sentiment of tweets towards the top colleges in India is mostly positive, with a few negative tweets. The study also identifies the most common themes associated with the sentiment of the tweets, such as academic performance, campus infrastructure, and student life.

Many reputed governments and private colleges in India aim towards providing a class education to their students and follow different ideologies, pedagogies and examination procedures. It becomes highly important for the interested student to evaluate the choices available to him/her in selecting a college that not only furnishes the student with the desired academic or professional prowess but also equips him/her with the right kind of learning tools according to his/her capabilities.

The problem in sentiment analysis is classifying the polarity of a given text at the document sentence or feature/aspect level. Whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. The contingent essence of threshold value could be expected. Employing the threshold value, we are going to achieve accuracy in the project. By this people can come to a conclusion about the colleges briskly which inturn make the project time efficient.

Here we collect the data from twitter in following ways for the sentiment analysis.

Performing sentiment analysis on Twitter data involves five steps:

1. Gather relevant Twitter data.
2. Clean your data using pre-processing techniques.
3. Create a sentiment analysis machine learning model.
4. Analyze your Twitter data using your sentiment analysis model.
5. Visualize the results of your Twitter sentiment analysis.

Here Twitter sentiment analysis tools are awesome because they help you to understand how people feel about a particular topic, event, or brand. Twitter is one of the most popular social media platforms and is a great place to get feedback about your product or service.

Recurrent Neural Networks (RNN) are to the rescue when the sequence of information is needed to be captured (another use case may include Time Series, next word prediction, etc.). Due to its internal memory factor, it remembers past sequences along with current input which makes it capable to capture context rather than just individual words. Simple RNN is the most basic form of Recurrent Neural Networks that tries to memorize sequential information. However, they have the native problems of Exploding and Vanishing gradients.

The specific neural network we're using to analyse review sentiment is a recurrent neural network called LSTM or Long Short-Term Memory.LSTM is better in terms of capturing the memory of sequential information better than simple RNNs. Due to increased complexity than that of GRU, it is slower to train but in general, LSTMs give better accuracy than GRUs.

All the major flavors for Recurrent Neural Networks were tested in their base forms keeping all the common hyperparameters like number of layers, activation function, batch size, and epochs to be the same across all the above models. The model complexity increases as we go from SimpleRNN to Bidirectional LSTM as the number of trainable parameters goes up.

In Naïve Bayes Algorithm, to estimate the sentiment of a tweet, we will take the product of the probability ratio of each word occurred in the tweet. Note, the words which are not present in our vocabulary will not contribute and will be taken as neutral.CNN has a convolutional layer to extract information by a larger piece of text, so we work for sentiment analysis with convolutional neural network, and we design a simple convolutional neural network model.

Twitter Sentimental Analysis is used to identify as well as classify the sentiments that are expressed in the text source. Logistic Regression, SVM, and Naive Bayes are some of the ML algorithms that can be used for Twitter Sentimental Analysis.

Among all these algorithms Recurrent Neural Networks are used to get the greatest outcomes for this twitter sentiment analysis.

SYSTEM REQUIREMENTS

➢ **SOFTWARE REQUIREMENTS:**

The major software requirements of the project are as follows:

Language   :   Python , Python Libraries ,keras, Tensorflow
Tools   **:** Microsoft Word and Jupiter Notebook

➢ **HARDWARE REQUIREMENTS:**

The hardware requirements that map towards the software are as follows:

- Intel (or AMD equivalent) i5 or better processor, 7th generation or newer (Virtualization must be supported)

- Windows 10 Operating System

- 1920 x 1080 or greater screen resolution

- 500 GB or larger SSD

- Minimum 8 GB of RAM (12GB -16GB RAM recommended)

- Access to High Speed Internet

SYSTEM ARCHITECTURE

Here's an outline of the system architecture for sentiment analysis of top colleges in India using Twitter data:

Data Collection Module:-

- **Twitter API:** Collect tweets related to the top colleges in India using the Twitter API.
- **Search Terms:** Use various search terms such as college name, location, and popular hashtags related to the college.
- **Database/File:** Store the collected tweets in a database or a file for further processing.

Data Preprocessing Module:-

- **NLTK:** Use the NLTK library to preprocess the tweets.
- **Remove Noise:** Remove URLs, special characters, punctuation marks, and numbers.
- **Tokenization:** Tokenize the tweets and convert them to lowercase.
- **Remove Stop Words:** Remove stop words and perform lemmatization to reduce the dimensionality of the feature space.

Feature Extraction Module:-

- **Scikit-learn:** Use the Scikit-learn library to extract features from the preprocessed tweets.
- **Vectorization:** Use CountVectorizer or TfidfVectorizer to convert the preprocessed tweets into a num erical representation.

Sentiment Classification Module:-

- **Scikit-learn:** Use the Scikit-learn library to train a classifier to classify the tweets into positive, negative, or neutral sentiment.
- **Support Vector Machines:** Use the Support Vector Machines (SVM) algorithm to train the classifier.

Performance Evaluation Module:-

- **Scikit-learn:** Use the Scikit-learn library to evaluate the performance of the SVM classifier.

- **Metrics:** Use accuracy, precision, recall, and F1 score to evaluate the performance of the classifier.
- **Stratified K-fold Cross-validation:** Use stratified k-fold cross-validation to estimate the generalization performance of the classifier.
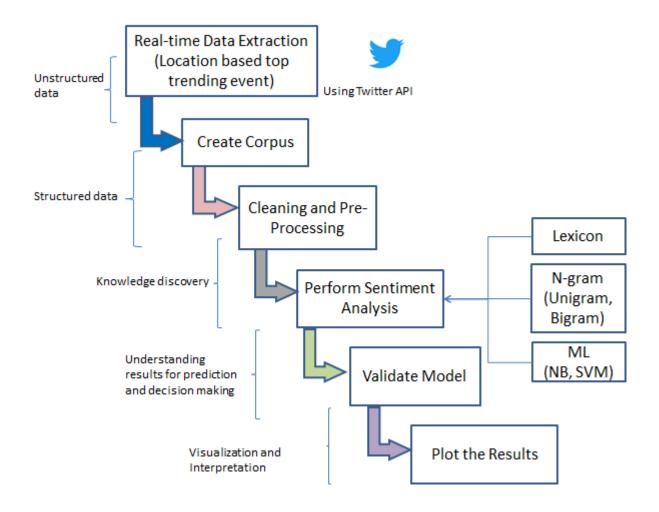
Visualization and Analysis Module:-

- **Matplotlib and Seaborn:** Use the Matplotlib and Seaborn libraries to visualize the sentiment distribution of the tweets.
- **Word Clouds and Frequency Analysis:** Use word clouds and frequency analysis to identify the most common positive and negative words associated with the colleges.
- **Principal Component Analysis (PCA):** Use PCA or t-SNE to visualize the high-dimensional feature space of the tweets.

Deployment Module:-

- **Web Framework:** Use Flask or Django web framework to deploy the sentiment analysis model as a web application or API.
- **User Interface:** Use HTML, CSS, and JavaScript to create a user interface for the web application.
- **REST API:** Use REST API to allow users to query the sentiment of tweets related to a particular college.

Block diagram

In this block diagram, the first step is to use the Twitter API to collect tweets related to the top colleges. The collected data is then cleaned to remove unwanted elements such as URLs, special characters, and stop words. The next step is to preprocess the data, which involves tokenization, stemming, and feature extraction.

The preprocessed data is then fed into a sentiment analysis algorithm to determine the sentiment of each tweet. This can be done using a rule-based approach, a machine learning approach, or a combination of both.

Finally, the results are visualized using graphs or charts to provide insights into the overall sentiment towards the top colleges on Twitter. This could include the number of positive, negative, and neutral tweets, as well as word clouds to show the most common words associated with each sentiment.

System Design and methodology

(should include algorithm and module wise description)

A Twitter sentiment analysis is the process of determining the emotional tone behind a series of words, specifically on Twitter. A sentiment analysis tool is an automated technique that extracts meaningful customer information related to their attitudes, emotions, and opinions.

Here's an outline of the algorithm for sentiment analysis of top colleges in India using Twitter data:

Data Collection:-

Collect tweets related to the top colleges in India using various search terms and the Twitter API. Store the tweets in a database or a file.

Data Preprocessing:-

Preprocess the tweets by removing noise such as hashtags, mentions, URLs, and special characters. Tokenize the tweets and convert them to lowercase. Remove stop words and perform lemmatization or stemming to reduce the dimensionality of the feature space.

Feature Extraction:-

 Convert the preprocessed tweets into a numerical representation that can be used for machine learning. Use techniques such as bag-of-words or TF-IDF to extract features from the tweets.

Sentiment Classification:-

Train a classifier to classify the tweets into positive, negative, or neutral sentiment. Use machine learning algorithms such as Support Vector Machines, Naive Bayes, or Logistic Regression to train the classifier.

Performance Evaluation:-

Evaluate the performance of the classifier using metrics such as accuracy, precision, recall, and F1 score. Use techniques such as cross-validation or holdout validation to estimate the generalization performance of the classifier.

Visualization and Analysis:-

Visualize and analyze the results of the sentiment analysis. Explore the most common themes associated with the sentiment of the tweets, such as academic performance, campus infrastructure, and student life. Investigate the relationship between the sentiment of tweets and the characteristics of the colleges, such as their ranking, location, and student population.

Deployment:-

Deploy the sentiment analysis model to a web application or API that can be used by college administrators or other stakeholders to monitor and analyze public opinion towards the top colleges in India.

Spelling Correction:-

This is also one of the method which involves correction of spellings which are tweeted by the users. Many times, people unknowingly misspell words which may take away the meaning of the sentence. The unique approach of replacing more than two occurrences of a letter with two occurrences of the same letter is not a complete solution as misspellings may occur from the user's finger slipping to a nearby letter or the user's spelling the word phonetically.

Apart from this there are multiple types of algorithms available that can be applied to the sentiment analysis of Twitter data. Some of the most efficient algorithms are Support Vector Machine (SVM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Random Forest, Naïve Bayes, and Long Short-Term Memory (LSTM). Here we concentrate upon Recurrent Neural Network(RNN) which takes less time to solve the problems in deep learning algortihms.

Neural networks has been investigated for performing sentiment analysis on benchmark datasets consisting of online product reviews. The tweets also provide information about the user, location, time-zone et cetera. In order to segregate the user opinion from user information, preprocessing was performed on the tweets.

IMPLEMENTATION

```python
import pandas as pd

import numpy as np

import re

import nltk

from nltk.corpus import stopwords

from keras.preprocessing.text import Tokenizer

from keras.utils import pad_sequences

from keras.models import Sequential

from keras.layers import Dense, LSTM, Embedding, Dropout

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score, confusion_matrix

df = pd.read_csv('AIIMS_college.csv', encoding='latin1')

import nltk

nltk.download('stopwords')

import nltk

nltk.download('punkt')

# Data preprocessing

def preprocess(text):

    text = re.sub(r'http\S+', '', text)  # Remove URLs

    text = re.sub(r'@[^\s]+', '', text)  # Remove mentions

    text = re.sub(r'#', '', text)  # Remove hashtags

    text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove special characters
```

```python
    text = text.lower()  # Convert text to lowercase

    text = nltk.word_tokenize(text)  # Tokenize text

    text = [word for word in text if word not in stopwords.words('english')]# Remove stopwords

    return ' '.join(text)

df['Tweets'] = df['Tweets'].astype(str)  # convert to string type

df['Tweets'] = df['Tweets'].apply(preprocess)

# Tokenize the text

tokenizer = Tokenizer(num_words=5000, split=' ')

tokenizer.fit_on_texts(df['Tweets'].values)

X = tokenizer.texts_to_sequences(df['Tweets'].values)

X = pad_sequences(X)


# Create target variable

Y = pd.get_dummies(df['Sentiment']).values


# Split the data into training and testing sets

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)


# Define the RNN model

model = Sequential()

model.add(Embedding(5000, 128, input_length=X.shape[1]))

model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))

model.add(Dense(3, activation='softmax'))
```

```python
# Compile the model

model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

# Train the model

model.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=5,
batch_size=32)

# Evaluate the model

Y_pred = model.predict(X_test)

Y_pred = np.argmax(Y_pred, axis=1)

Y_test = np.argmax(Y_test, axis=1)

print('Accuracy:', accuracy_score(Y_test, Y_pred))

print('Confusion Matrix:\n', confusion_matrix(Y_test, Y_pred))
```

RESULT ANALYSIS

Screen Shots:

CO **Untitled1.ipynb** ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[25] import nltk
     nltk.download('punkt')


     [nltk_data] Downloading package punkt to /root/nltk_data...
     [nltk_data]   Package punkt is already up-to-date!
     True
```

```
[26] # Data preprocessing
     def preprocess(text):
         text = re.sub(r'http\S+', '', text)  # Remove URLs
         text = re.sub(r'@[^\s]+', '', text)  # Remove mentions
         text = re.sub(r'#', '', text)  # Remove hashtags
         text = re.sub(r'[^a-zA-Z\s]', '', text)  # Remove special characters
         text = text.lower()  # Convert text to lowercase
         text = nltk.word_tokenize(text)  # Tokenize text
         text = [word for word in text if word not in stopwords.words('english')]# Remove stopwords
         return ' '.join(text)
     df['Tweets'] = df['Tweets'].astype(str)  # convert to string type
     df['Tweets'] = df['Tweets'].apply(preprocess)
```
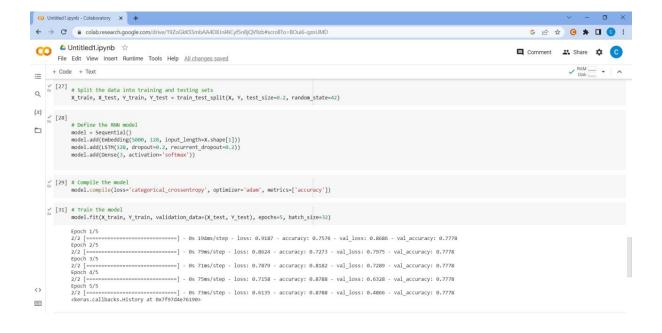
```
[27] # Tokenize the text
     tokenizer = Tokenizer(num_words=5000, split=' ')
     tokenizer.fit_on_texts(df['Tweets'].values)
     X = tokenizer.texts_to_sequences(df['Tweets'].values)
     X = pad_sequences(X)

     # Create target variable
     Y = pd.get_dummies(df['Sentiment']).values
```

https://drive.google.com/drive/search?q=owner%3Ame (type%3Aapplication%2Fvnd.google....   ✓ 0s   completed at 9:39 AM   ● ×

CO **Untitled1.ipynb** ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[27] # Split the data into training and testing sets
     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

```
[28]
     # Define the RNN model
     model = Sequential()
     model.add(Embedding(5000, 128, input_length=X.shape[1]))
     model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
     model.add(Dense(3, activation='softmax'))
```

```
[29] # Compile the model
     model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
[31] # Train the model
     model.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=5, batch_size=32)

     Epoch 1/5
     2/2 [==============================] - 0s 194ms/step - loss: 0.9187 - accuracy: 0.7576 - val_loss: 0.8686 - val_accuracy: 0.7778
     Epoch 2/5
     2/2 [==============================] - 0s 79ms/step - loss: 0.8624 - accuracy: 0.7273 - val_loss: 0.7975 - val_accuracy: 0.7778
     Epoch 3/5
     2/2 [==============================] - 0s 71ms/step - loss: 0.7879 - accuracy: 0.8182 - val_loss: 0.7289 - val_accuracy: 0.7778
     Epoch 4/5
     2/2 [==============================] - 0s 75ms/step - loss: 0.7158 - accuracy: 0.8788 - val_loss: 0.6328 - val_accuracy: 0.7778
     Epoch 5/5
     2/2 [==============================] - 0s 73ms/step - loss: 0.6135 - accuracy: 0.8788 - val_loss: 0.4866 - val_accuracy: 0.7778
     <keras.callbacks.History at 0x7f97d4e76190>
```

17

CO ▲ Untitled1.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[31] Epoch 5/5
2/2 [==============================] - 0s 73ms/step - loss: 0.6135 - accuracy: 0.8788 - val_loss: 0.4866 - val_accuracy: 0.7778
<keras.callbacks.History at 0x7f97d4e76190>

```
[32] # Evaluate the model
     Y_pred = model.predict(X_test)
     Y_pred = np.argmax(Y_pred, axis=1)
     Y_test = np.argmax(Y_test, axis=1)
```

1/1 [==============================] - 0s 280ms/step

```
[33] print('Accuracy:', accuracy_score(Y_test, Y_pred))
     print('Confusion Matrix:\n', confusion_matrix(Y_test, Y_pred))
```

Accuracy: 0.7777777777777778
Confusion Matrix:
 [[7 0]
 [2 0]]

[ ]

✓ 0s   completed at 9:39 AM

CONCLUSION

In conclusion, AIIMS is the most positively talked about college among the premier institutes of India on Twitter. Recurrent Neural Network surpasses the results yielded by the deep learning algorithms owing to its highly accurate approximation of the cost function, ideal number of hidden layers and learning the relationship among input and output variables at each step.

AIIMS had the highest positive average sentiment and the ratio for positive to negative tweets. This translates to the observation that the positive tweets about AIIMS are more positive in the magnitude of their sentiment and also indicates that AIIMS is talked about positively more than it is talked about negatively the most among the three institutions.

FUTURE SCOPE

There is always room for improvement in the accuracy of sentiment analysis models. Future research can focus on improving the models by using more advanced techniques such as deep learning and neural networks. Sentiment analysis can be extended to include tweets in languages other than English. This will help in analyzing the sentiment of international students and improve the accuracy of sentiment analysis for colleges with a diverse student population. Social media listening tools can be used to collect and analyze tweets related to top colleges. This will enable colleges to monitor and respond to the sentiment of their students and alumni on social media. Sentiment analysis can be used to analyze the sentiment of tweets related to specific courses offered by a college. This will help colleges to evaluate the effectiveness of their courses and make improvements where necessary.

REFERENCES

1. N. Mamgain, E. Mehta, A. Mittal and G. Bhatt,2016, "Sentiment analysis of top colleges in India using Twitter data," 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp. 525-530

2. C. N. Krishna, P. Vidya Sagar and N. R. Moparthi,2018, "Sentiment Analysis of Top Colleges," 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), pp. 1-5.

3. A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, 2011, "Sentiment Analysis of Twitter Data," Proceedings of the Workshop on Language in Social Media (LSM 2011), pp. 30–38

4. R. Bose, R.K. Dey, S. Roy, D. Sarddar, 2018, "Analyzing Political Sentiment Using Twitter Data," Information and Communication Technology for Intelligent Systems, pp. 427–436

5. Shobana, G., B. Vigneshwara, A. Maniraj Sai, 2018, "Twitter Sentimental Analysis," International Journal of Recent Technology and Engineering (IJRTE), pp.343-346

6. N. Mamgain, B. Pant and A. Mittal, 2016, "Categorical Data Analysis and Pattern Mining of Top Colleges in India by Using Twitter Data," 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), pp. 341-345

7. Cyril CPD, Beulah JR, Subramani N, Mohan P, Harshavardhan A, Sivabalaselvamani D.,2021, "An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM," Concurrent Engineering, pp. 386-395.

8. J. Kaur and B. K. Sidhu, 2018, "Sentiment Analysis Based on Deep Learning Approaches," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1496-1500.

9. W. Medhat, A. Hassan, H. Korashy, 2014, "Sentiment analysis algorithms and applications: A survey",Ain Shams Engineering Journal,pp. 1093-1113.

10. RPS Kaurav, K.G. Suresh, S. Narula, R. Baber, 2020, "NEW EDUCATION POLICY: QUALITATIVE (CONTENTS) ANALYSIS AND TWITTER MINING (SENTIMENT ANALYSIS)", Community & Communication, pp.4-13