Open Access
Full Length Article

ORIGINAL ARTICLE

Leveraging Natural Language Processing for Predictive Modeling

Areej Fatemah Meghji[1], Veena Kumari[2], Farhan Bashir[3]

[1, 2] Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan
[3] Department of Computer Science, The University of Larkano, Pakistan

ABSTRACT

Weather analysis is the process of examining parameters like atmospheric pressure, humidity, and wind speed, to make calculations and predictions regarding the current weather. Using the publicly available dataset made available by CrowdFlower on Kaggle, this research performs operations of natural language processing and predictive modeling on the provided data to determine the sentiment of a tweet, and predict the type of weather being mentioned in the tweet. We first perform the data-pre-processing operations of label engineering, feature extraction, and data visualization on the available data. Regression approaches including Gradient Boosting, Ridge, and Linear regressor, and classification approaches including Random Forest, Multinomial Naïve Bayes, Linear Support Vector, and Stochastic Gradient Descent classifier were then applied for data modeling. The results of the current research indicate that feature reduction has a positive effect on overall predictive accuracy. The results also indicate that predictive performance is higher when performing classification tasks as compared to regression tasks.

Keywords: **Weather Forecasting, Sentiment Analysis, Multi-label Classification, Temporal Reference Detection, Regression**

Author`s Contribution
[1,2] *Methodology, Writing: Original Draft Preparation, Review, and Editing, Validation, Formal Analysis, Result Reporting,* [3]*Visualization, Data Analysis, Result Reporting, Writing: Original Draft Preparation*

Address of Correspondence
Areej Fatemah Meghji,
Email: areej.fatemah@faculty.muet.edu.pk

Article info.
Received: January 14, 2026
Accepted: March 10, 2026
Published: March 24, 2026

Cite this article: *Meghji A.F., Kumari V., Bashir F. Leveraging Natural Language Processing for Predictive Modeling. Journal of Information Communication Technologies and Robotic Applications, 17(1), 2026, 13-23.*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

The rapid growth of digital technologies and the widespread use of social media platforms have fundamentally changed how information is created, distributed and consumed. Today, we live in a data-driven society where data has an impact on our day-to-day decisions and life in general. From better targeting to enhanced consumer experiences, the understanding of data influences several aspects of our life. Data is being produced at a very fast rate by millions of different sources on a global scale. Social media networks and platforms rank among the most massive and important sources of data. Twitter produces more than 12 gigabytes of data per day. In comparison, Facebook generates more than 500 terabytes of data every day, or 84 terabytes each week and 4.3 petabytes annually. This data consists of image, video, text, and other content. Data generated by these platforms exists in unstructured form and is popularly called unstructured data. Unstructured data from these sources can comprise of opinions, comments, reviews, and so on. People are more curious to automate the process these days. Since automation depend on the context and here we need to use the natural language processing for the automation because the textual data is mostly present in the human language or natural language.

Platforms like X produce massive amounts of user-generated textual data in real time, thus, they represent the opinions, experiences and observations of the population about the daily events [1]. Out of the numerous fields that are represented in the social media discourse, weather has been a strong niche, as the users often describe their observations, responses, and responses related to the changing weather conditions [2]. These textual manifestations provide not only descriptive data on weather phenomena but also give useful data on the mood of society and time.

The available studies in this field have brought forth important results, with the use of machine learning principles in activities like sentiment classification and weather forecasting using Twitter corpora. Some of the studies have only focused on the relationship between the content of tweets and meteorological variables as opposed to others that have treated sentiment analysis as a unique goal [3]. However, some of the existing strategies have significant weaknesses. First, most previous studies assume that sentiment analysis, weather classification, and temporal reference detection are independent issues, thus, not paying attention to their possible interrelations. Secondly, some of the studies use a limited set of features or single-feature settings, which may hinder the effectiveness of classification [2]. Moreover, issues of multi-output prediction are often considered without a sufficient exploration of task decomposition that results in under-optimal performance results.

To address such shortcomings, this research offers an integrative concept that simultaneously tackles sentiment analysis, temporal reference identification and weather-type categorizing used on X data, as well as using natural language processing alongside machine learning methods. Unlike the previous literature, the current work dwells upon both the multi-output and task-specific modeling paradigms, thus, allowing one to exhaustively assess the merits and limitations of the two paradigms. The proposed methodology aims at improving the overall classification, accuracy and interpretability by dividing complex predictive tasks into smaller, more focused sub-tasks in which they are more suited to.

The main aim of this research is to develop a natural language processing-based pipeline to analyze the weather-related tweets, to determine the relative effectiveness of various machine learning models when the task formulation is varied and to show how the combined analysis of sentiment, temporal context and meteorological categories can provide significant insights based on the unstructured text in the social media. Through challenging the weaknesses that characterize previous studies and suggesting a systematic, comparative approach to the research methodology, this study contributes to the overall fields of social media analytics and text-based environmental analysis.

The paper is organized into the following parts: a Literature Review is given first, providing an in-depth synthesis of the relevant literature to create the background concerning the current investigation. This is followed by a description of the investigative procedures in the Research Methodology section. Following sections include Results and Discussion where the empirical findings are questioned and a Conclusion made which reflects the findings of the research as well as discussing the limitations of the study and the future research directions.

RELATED WORK

Table 1 provides an overall overview of the available literature related to sentiment analysis, textual meteorological classification, and temporal analytics, as based on natural

language processing and machine learning approaches. The review will be used to critically evaluate the methodologies used in the past, highlight the important findings, and outline the methodological limitations in the current literature, which will provide a rational to the research path to take.

Table 1: Summary of Related Research

Ref	Research Objective	Methodology	Results
[2]	Focused on analysis of the classification problem of estimating the present weather using publicly available Twitter text data	Collected Data with Twitter API and Validated with OpenWeatherMap API. Possible Choices for predictions: Clear', 'Clouds', 'Rain', and 'Thunder-Storm'	Gaussian Naïve Bayes Classifier gives Accuracy up to 37.1%, Convolutional Neural Network with Accuracy of 58.1%
[4]	Research focused on Sentiment Analysis of English Text	Collected Data from Kaggle, Pre-Process the Data, Model Application	Balanced Data Set: Naïve Bayes Accuracy 75.5%, ID3 Accuracy 73%, Unbalanced Data Set: K-Nearest Neighbor Accuracy 38%, Decision Tree Accuracy 57.5%, Random Forest Accuracy 58%, Random Tree Accuracy 50%
[5]	Classify weather conditions in Indonesia using Twitter data by applying text mining and machine learning techniques	Support Vector Machine, Multi-nomial Naive Bayes, and Logistic Regression were trained to classify tweets into five weather Classes	SVM achieved the best performance with an accuracy of 93%, outperforming LR (88%) and MNB (53%). SVM is highly effective for text-based multi-class weather classification, especially when dealing with high-dimensional TF-IDF features
[6]	Comparing the performance of eight machine learning models to identify the most effective and reliable classifier of opinion on the topic of climate change	15,819 tweets from Kaggle categorized into four classes. Models: implemented are LR, SVM (Linear/Non-linear), XGB, DT, RF, NB, KNN, and GBM	Linear SVM with top performance accuracy of 73.92% and a weighted F1-score of 0.72. Other classifiers, BERT, were also found to provide high accuracy metrics. Conversely, the Decision Tree classifier achieved the worst result as it had an accuracy of 61%
[7]	Detection of temporality at discourse level on financial news by combining Natural Language Processing and Machine Learning	Data Collected from BloomBerg News, CNN business, Forbes. Natural Language Processing and Machine Learning Models	ZeroR Aaccuracy=60.83%, Decision Tree Accuracy 68.33%, Random Forest Accuracy 80.17%, Support Vector Accuracy 83.5%
[8]	Review and analyze existing research on machine learning techniques used for text classification on Twitter, with a focus on	Classification models (SVM, Naive Bayes, Logistic Regression, Decision Trees, Neural Networks, LSTM, ensemble models)	Traditional machine learning models such as SVM, Logistic Regression, and Naïve Bayes are widely used and perform well on Twitter text classification

	identifying commonly used methods, performance trends		tasks. SVM and Logistic Regression frequently achieved high accuracy, often above 85-90% Deep learning approaches showed superior performance in large datasets
[9]	Quantify the relationship between real-time weather conditions and public sentiment expressed on Twitter	Six regression models: Linear Regression, Ridge Regression, Support Vector Regression, Random Forest, AdaBoost, and Gradient Boosting were trained to predict the ratio of positive tweets to total tweets for given weather conditions	The sentiment classification achieved an accuracy of approximately 89.74% when compared with manual labeling. SVR performed best on real-time test data, yielding the lowest residual errors, even though Random Forest showed lower training RMSE
[10]	Analyze public sentiment toward weather conditions using Twitter data and to propose a deep neural network-based sentiment analysis mode	A Deep Neural Network (DNN) architecture was implemented to learn sentiment patterns from textual features	The model outperformed the traditional machine-learning methods in terms of the effectiveness of sentiment classification when using weather-related tweets. It displayed a strong encoding skill with regard to complex affective expressions, especially in case of very extreme or abnormal meteorological occurrences
[11]	To analyze how weather affects public sentiment in hot-climate cities using Twitter data	Text mining, tokenization, key-word extraction, bigram and correlation analysis	Found strong links between temperature and heat-related tweets; Singapore showed mostly negative sentiment, while Phoenix showed seasonal variation
[12]	Shows classification of emotions based on lexicon and crowd-sourced sensing with Twitter data and the effects of meteorological conditions on collective affect	Crowd-Sourced Sensing and Text (CSST) architecture was deployed. A word-model-based geo-locational approach, word-based emotion scoring, and real-time meteorological data were integrated	Positive weather conditions have a positive relationship with mostly positive emotions in Tweets as compared to bad weather which has a positive relationship with negative affect
[13]	To determine climate change and how the Naive Bayes algorithm can be used to monitor sentiment in real-time	5,120 Indonesian-language tweets. TF-IDF for vectorization of top 400 words. Multinomial Naïve Bayes classifier was implemented to categorize sentiments	Negative sentiment (62%), mostly about extreme heat and storm events, whereas moderate weather conditions were linked with neutral (24.36%) and positive (14.3%) sentiments. Model

			evaluation performed with F1-scores of 0.95 in negative, 0.86 in neutral and 0.83 in positive sentiment with a macro-average F1-score of 0.88

Earlier studies in this area have mostly focused on researching weather-relevant posts, and used simplistic natural language processing systems and classification methods and worked on a specific section such as sentiment analysis or meteorological event detection. Most of these works have been based on the standard pre-process pipelines and have implemented single machine learning models without a large-scale feature engineering. By contrast, the current study defines a unified and methodological structure, which combines the data collection method, advanced data pre-processing, and multitask learning in a single investigative model. The paper unfolds and comparatively assesses a set of supervised machine learning algorithms to perform activities like sentiment classification, temporal reference identification, and identification of weather-type.

RESEARCH METHODOLOGY

This study follows a supervised machine learning pipeline to classify weather-related tweets by sentiment, temporal analysis and weather type, using natural language processing on an annotated public dataset as visualized in Figure 1.

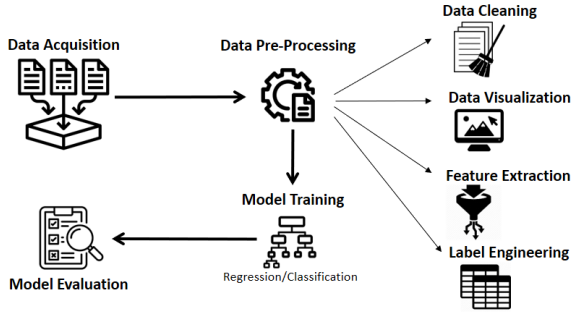

Figure 1: Research Methodology

3.1 Data Acquisition

The dataset for this study was obtained from Kaggle. The dataset Partly Sunny with a Chance of Hashtags contains approximately 77,000 English tweets annotated for weather relevance, sentiment, temporal reference, and weather type made available by CrowdFlower [14]. As presented in Figure 2, this labeled dataset has 28 columns in total. Three columns are textual, containing actual tweet, location and state of the user. The rest of columns are representing the Sentiment, Weather and Time given in the tweet.

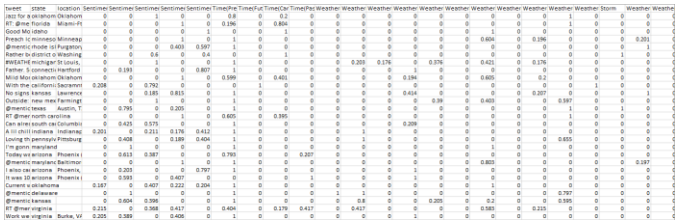

Figure 2: Screenshot of the dataset

3.2 Data Pre-processing

3.2.1 Data Cleaning

During the data cleaning and preparation stage, the raw data was purified according to its structure so that it becomes appropriately applicable to be further modeled. As this research focuses on sentiments and type of weather prediction, the time column was dropped at this stage. In the dataset, the sentiment columns contain different type of sentiments such as Positive, Negative, Neutral, Can't Tell, Not Related to Weather. There are 15 different types of weather columns: cloud, cold, dry, hot, humid, hurricane, ice, others, rain, snow, storm, sun, tornado, wind, and can't tell. For each column the extent of the weather is depicted as a decimal number between the range of 0 and 1.

The records that have missing or inconsistent values in the key fields are identified and are corrected or eliminated and non-informative attributes like location and state are eliminated to minimize noise [3]. It is then normalized by removing the URLs, user mentions, punctuation, special characters, and other non-alphanumeric material in the text of the tweet and lowercased to obtain a normalized result [1]. Lastly, the filtered tweets were coded into unit words called tokens and, along with the columns of the respective labels [4], structured into a form that could be analyzed, which offered a high-quality foundation of feature extraction and supervised learning. The code snippet of the cleaning processing of the dataset has been presented in Figure 3.

```
[ ] # print and check the output of refineTweets
    refineTweets(approachOneTrain["tweet"][0], True)

    After Execution of re.sub('@mention', '', tweet):  Jazz for a Rainy Afternoon:  {link}
    After Execution of re.sub('[^A-Za-z0-9]', ' ', tweet):  Jazz for a Rainy Afternoon    link
    After Execution of re.sub('  ', ' ', tweet):  Jazz for a Rainy Afternoon  link
    'Jazz for a Rainy Afternoon  link '

[ ] # Iterating and applying refined tweet method and storing result back
    approachOneTrain["tweet"] = [refineTweets(tweet) for tweet in approachOneTrain["tweet"]]
    approachOneTest["tweet"] = [refineTweets(tweet) for tweet in approachOneTest["tweet"]]
```

Figure 3: Code snippet of the dataset cleaning process

3.2.2 Data Visualization

The next step was the visualization of the columns available in the dataset to get a better understanding of the distribution of the sentiments and various weather conditions most commonly recorded in the research dataset. Figure 4 and Figure 5 present the 5 sentiment fields and the 15 weather fields available in the dataset respectively.

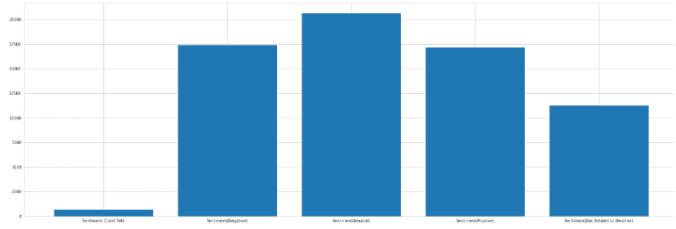

Figure 4: Visualized Sentiment Fields

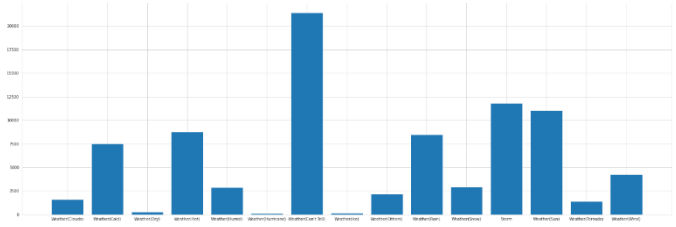

Figure 5: Visualized Weather Fields

As observed from the figure 4, the sentiment neutral is predominant followed by positive and negative sentiments. As for figure 5, the weather feature 'can't tell' has the most occurrence followed by sun, storm, and cold.

3.2.3 Feature Extraction

In the step of feature extraction, tweets after pre-processing were converted to numerical data that could be processed using machine learning. Specifically, a Term Frequency-Inverse Document Frequency (TF-IDF) strategy is used that reduces every tweet to a vectorial representation that captures the comparative importance of lexical elements within the entire data collection [3-5]. To address the complexity of the model and improve the lack of sparsity, a set of fixed vocabularies was selected consisting of approximately 100 features, selected on the shortest and longest tweet length statistics calculated. This max feature was selected using the formula given in equation i).

$$Max\ Features = \frac{Max\ (tweets)+Min\ (tweets)}{2}$$

Stop-word removal was applied so that highly frequent but semantically weak tokens (e.g., the, is, and) contributed little or no weight, thereby emphasizing more informative content words in the resulting feature space [15]. The code snippet for the tf-idf processing has been depicted in Figure 6.

```
# tfidf
tfidf = TfidfVectorizer(max_features=100, strip_accents='unicode', analyzer='word', stop_words='english')

# Transforming Train and Test Data Sets
trainX = tfidf.fit_transform(approachOneTrain["tweet"])
testX = tfidf.fit_transform(approachOneTest["tweet"])

# Printing trainX(Tweet) After Transformation
print(trainX)

  (0, 48)       0.5640926123819038
  (0, 68)       0.8257115262948431
  (1, 22)       0.5749137371524032
  (1, 52)       0.5439878651446616
  (1, 71)       0.3826527683661116
  (1, 68)       0.476579748070101 95
  (2, 98)       0.522081962553465
  (2, 84)       0.5110700037997254
  (2, 56)       0.5137073703680033
  (2, 33)       0.4498250918102083
  (3, 47)       0.3594031008804168
  (3, 11)       0.4295765448697636 5
  (3, 94)       0.171592321132757 77
  (3, 50)       0.4235614837360393
  (3, 84)       0.4328742777095003
  (3, 22)       0.4483471457187061
  (3, 71)       0.2984122058868605
```

Figure 6: Code Snippet TF-IDF Transformation

3.2.4 Label Engineering

Regression Formulation

In the first step, the feature columns were directly used as numeric response variables on their natural scales and then multivariate regression models were trained to predict all label dimensions at the same time.

Binarization to create classification targets

Experimental research found out that this formulation was both predictive performance and interpretable constrained. The labels were therefore re-conceptualized as classification targets by binarizing each column, that is, values above 0.5 were coded as 1, and values below 0.5 were coded as 0, thus creating clean binary indicators of class present or absent. This change made the task regression to multi-label classification.

Weather Label Reduction

To reduce sparsity, semantically similar weather labels were condensed into fewer categories e.g. cold, snow and ice were condensed to one cold category and other weather labels were condensed into a small number of weather categories (cold, hot, cloudy and other).

Sentiment Analysis

Lastly, to make the learning problem simpler and to take advantage of existing domain-specific tooling [16], sentiment prediction was also outsourced in some of the experimental settings by not providing sentiment columns to the supervised targets and becoming sentiment analyzers with separate external libraries.

3.3 Model Training

The method of figuring out whether a piece of writing is positive, negative, or neutral is called sentiment analysis. In order to assign weighted sentiment ratings to the entities, topics, themes, and categories contained inside a sentence or phrase, sentiment analysis systems for text analysis integrate NLP and machine learning approaches. Sentiment analysis aids data analysts at major corporations in understanding consumer experiences, doing complex market research, monitoring brand and product reputation, and gauging public opinion. In order to provide their own clients with insightful information, data analytics organizations frequently incorporate third-party sentiment analysis APIs into their own platforms for customer experience management, social media monitoring, and workforce analytics.

After building definitive sets of labels in sentiment, temporal reference and weather category, the dataset was split into training and testing parts to allow an unbiased evaluation of the model performance, which was the application of scikit-learn train test split after tf-idf transformation as depicted in Figure 7.

```
def trainTestSplitWithPreprocessing(tweets, labels):
  cleanTweet = sanitizeTweets(tweets)
  countVectorizer = CountVectorizer(max_features = 1000, stop_words = 'english', lowercase = True, analyzer = 'word')
  cleanTweets = countVectorizer.fit_transform(cleanTweet)
  cleanTweets = np.array(cleanTweets.toarray())
  lengthOfTweets = len(tweets)
  # Split Data into training and cross-validation:
  # 75% training, 25% cross-validation
  # 4 part of 100% i.e. 25%
  numTrainCV = lengthOfTweets/4
  # Total (100-25)%
  cvThreshold = lengthOfTweets - numTrainCV
  # Get first 75% records (Training Data Set)
  trainSet = cleanTweets[0:int(cvThreshold)]
  # remaining 25% records (Test Data Set)
  trainSetCV = cleanTweets[int(cvThreshold):]
  trainSetLabels = trainLabels[0:int(cvThreshold)]
  trainSetLabelsCV = trainLabels[int(cvThreshold):]
  return trainSet, trainSetLabels, trainSetCV, trainSetLabelsCV

from sklearn import multioutput
def applyMultiModel(xTrain, yTrain, xTest, yTest, model):
  model = multioutput.MultiOutputClassifier(model)
  model.fit(xTrain, yTrain)
  print(model.score(xTest, yTest))
  return model

xTrain, yTrain, xTest, yTest = trainTestSplitWithPreprocessing(approachTwoTrain["tweet"], trainLabels)
```

Figure 7: Code snippet of the training process

3.3.1 Multi-output Models

The first approach to multi-output learning involved wrapping individual base estimators into a multi-output architecture, called MultiOutputRegressor or MultiOutputClassifier, in such a way that one model can be used to make a prediction of multiple targets (e.g. sentiment, temporality and weather) per tweet [17]. The algorithms that are used in this research include:

Linear models: Linear models, including Linear Regression, Logistic Regression and Ridge Classifier, can be efficient at delivering baseline performance and interpretability, making it possible to systematically estimate the importance of the features and boundaries of the classes [18].

Support Vector Machine: Linear Support Vector Classifier (LinearSVC) is used as it has proven to be effective in the processing of large-scale textual data [19].

Naive Bayes: Gaussian and Multinomial Naive Bayes, are especially effective in text modeling, especially probabilistic models [20].

Tree-based models: Tree-based models, such as Decision Tree, Random Forest, and Gradient Boosting, consist of non-linear relationships and complex interactions of features and, therefore, significantly helps to improve predictive performance [21].

3.4 Model Evaluation

The regression models were evaluated on the metric of R^2. Table 2 presents the interpretation of the R^2 scores where R^2 is a measure of how well the variance in the dependent variable is explained by the model and can be represented as equation ii).

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

Table 2: R^2 interpretation

SNo	R^2 Value	R^2 Interpretation
1	1	Perfect Fit
2	2	Predicting only mean
3	<0	Prediction worse than predicting only mean
4	0.7-0.9	Good Performance
5	0.5-0.7	Moderate Performance

The classifiers used in the research were evaluated on the popular and widely used metric of accuracy measured as provided in equation iii).

$$accuracy = \frac{correct\ predictions\ by\ the\ model}{number\ of\ predictions\ by\ the\ model}$$

RESULTS & DISCUSSION

4.1 Applying Models on Cleaned and Preprocessed Data

In the approach followed in this research, we first cleaned the tweets and removed the pronunciations, special characters, mentions and white space. Once the text is cleaned then we are converted this text to TF-IDF and removed the stop words from

them. Once text is transformed to TF-IDF the algorithm can be applied to it since the data is converted to numeric format. Before applying algorithms, we need to check what type of output values we have. For this approach we have continuous output values, and we need to predict multiple things that are sentiment (All Column values) as well as weather type (All Column Values) so we are using multi output models. The algorithms that are used are regression algorithms because we are predicting continuous values.

The sample of sentiment columns (Positive, Negative and Neutral) is shown in Figure 8. We can observe the user sentiment over an extended period of time; the three distinct line charts are used to depict negative sentiments represented in blue at the top, positive in green at the bottom of the chart, and the neutral sentiments are represented in orange. The higher blue spikes represent a strong negative sentiment.

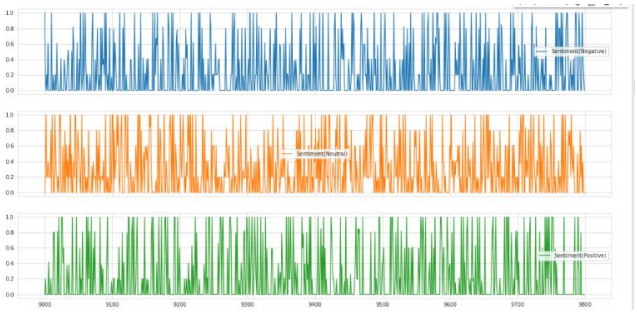

Figure 8: Sentiment analysis over time: Fluctuations in positive, negative, and neutral sentiments

We have also visualized some of the weather columns in this step as shown in Figure 9 and Figure 10. We can see that the dataset contains several weather entries which are essentially representing the type of weather.

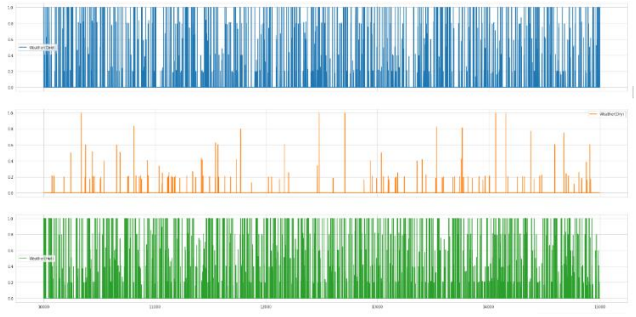

Figure 9: Time-series visualization of weather data trends (a)

To perform regression analysis, we have made use of five algorithms, all with multiple-outputs. The performance of the algorithms has been evaluated based on the R^2 score as presented in Table 3. We can analyze from the scores that the algorithms have not performed well on the data and have not been able to achieve a moderate performance. The highest score of 0.45 has been exhibited by the Gradient Boosting Regressor.

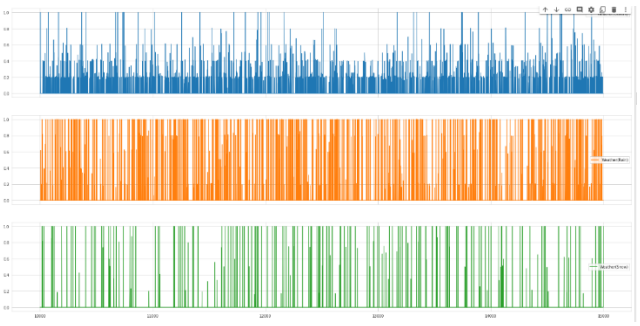

Figure 10: Time-series visualization of weather data trends (b)

Table 3: Performance Evaluation of Regression Models

Algorithm	Score
Gradient Boosting Regressor	0.45
Ridge	0.44
Linear Regression	0.44
SGD Regressor	0.42
Decision Tree Regressor	0.31

4.2 Column Reduction

In order to analyze and in an attempt to improve the performance of the algorithms, in the second approach, we are reducing the columns. For column reduction we are using a merging technique. From the data set columns, we can see that different weather classes represent the same construct. For instance, when describing the weather in terms of Sun, Humid and Dry, we are essentially representing the same type broader category of Weather which can be described as hot.

We have thus transformed the dataset and after the transformation our dataset contains all the columns for sentiment that is Sentiment (Can't tell), Sentiment (Negative), Sentiment (Neutral), Sentiment (Positive), Sentiment (Not Related to Weather), and Weather Columns are reduced to Weather (Rain), Weather (Hot), Weather (Cold), Weather (Others). Weather (Rain) is the merging result of Weather (Cloud), Weather (Rain), and Weather (Hurricane) columns. Weather (Hot) is the merging result of Weather (Sun), Weather (Humid), Weather (Dry) and Weather (Hot) columns. Weather (Cold) is the merge result of Weather (Cold), Weather (Ice), and Weather (Snow) columns. Weather (Others) is the merge result of Weather (Other), Weather (Tornado), Weather (Wind), Weather (Storm), and Weather (Can't tell). The weather data after reduction is visualized in figure 11.

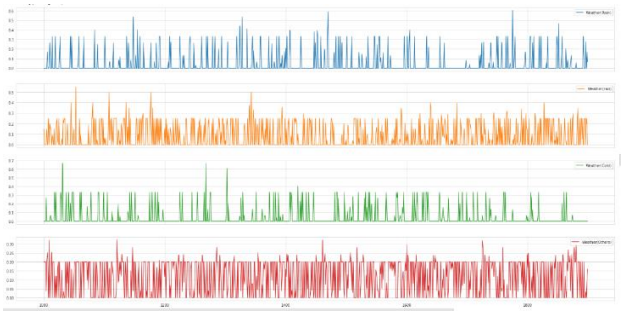

Figure 11: Time-series visualization of weather data trends with merged columns

After this transformation we are cleaning the tweets like we did in first approach removal of pronunciations, white spaces, special characters and stop words removal. After Cleaning we transform the textual data into numbers with TF-IDF and applying the algorithms. The results of different algorithms are shown in table 4. We can see a slight improvement in the algorithm scores. The highest score has increased from 0.45 to 0.53 as exhibited by Linear Regression and the Ridge Regressor.

Table 4: Performance Evaluation using merged columns

Algorithm	Score
Gradient Boosting Regressor	0.47
Ridge	0.53
Linear Regression	0.53
SGD Regressor	0.41
Decision Tree Regressor	0.39

4.3 External Sentiment Analysis

Since Sentiment Analysis is a very common task there are a lot of libraries and APIs which provide amazing results for sentiment analysis. This task can be outsourced to these libraries [16]. Now we need to find a way to prepare a machine learning model which can analyze the tweet and predict the type of weather that the tweet is referring to. same reduced columns were taken in this approach i.e. Weather (Rain), Weather (Hot), Weather (Cold), Weather (Others). The results from applying different algorithms are shown in Table 5. Multi-output regressor are used to predict the multiple columns in order to gain results from each column of weather [17]. The performance of all the models has further improved with the highest performance being exhibited by the Gradient Boosting Regressor with a score of 0.61. The comparative performance of all the three approaches has been graphically presented in Figure 12.

Table 5: Performance Evaluation using External Sentiment Analysis

Algorithm	Score
Gradient Boosting Regressor	0.61
Ridge	0.53
Linear Regression	0.53
SGD Regressor	0.43
Decision Tree Regressor	0.46

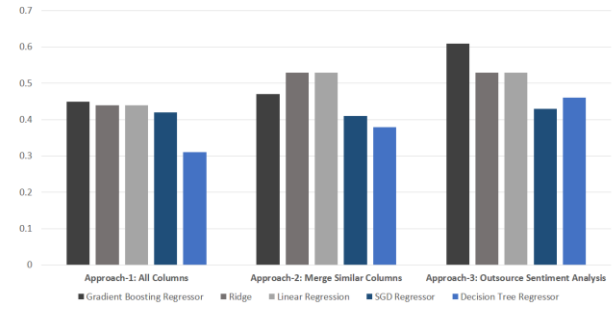

Figure 12: Regression Model Performance Evaluation Under Various Approaches

4.4 Converting Regression Problem to Classification Problem

In this approach the analysis was changed from regression to classification. We know that the values are given in decimals and range between 0 and 1. In this approach we have changed all the values that are less than or equals to 0.5 to 0 and the values greater than 0.5 are replaced by 1 as visualized in figure 13. Now we have the problem converted to classification and it is easy to apply classification algorithms on the dataset.

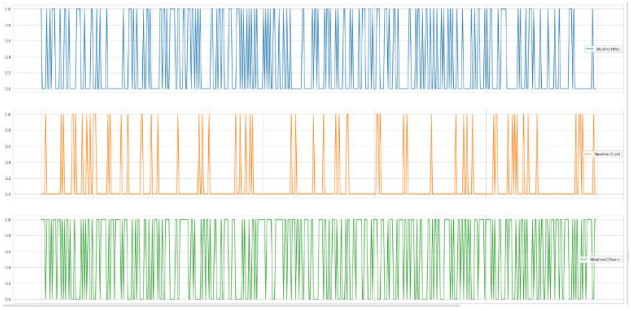

Figure 13: Transformed dataset representation

After doing this transformation on our original dataset we also reduced the weather columns as done previously. The data set had only weather columns since we decided to outsource sentiment analysis in this approach too. The multi-output model was used in this case so that we could get output of all columns. The results from this approach are provided in Table 6. We can observe a significant improvement in the results. The highest accuracy of 81%

was achieved by the Linear SVC and Logistic Regression classifier.

Table 6: Classifier Performance Evaluation

Algorithm	Accuracy
Multinomial Naïve Bayes	74%
Random Forest Classifier	80%
Linear SVC	81%
Logistic Regression	81%
Ridge Classifier	80%
Extremely Randomized Trees	75%
SGD Classifier	80%

4.5 Classification with Label Encoding

One thing that is common in the previous approaches is that we are using multioutput for the prediction of all other weather types. This makes sense if you're working with regression model because you want to predict the percentages of the weather however since the problem is converted to classification problem, we no longer need to use the multioutput because we know if one class is true other classes will be false. In case of regression, we might still want to predict the class with the highest value. In this case we can only keep the useful class during training and leave the other classes. For this approach we are using the reduced column dataset with weather columns only i.e. Weather (Rain), Weather (Hot), Weather (Cold), Weather (Others). We are assigning number 1 to Weather (Rain), 2 to Weather (Hot), 3 to Weather (Cold) and 4 to Weather (Others). By assigning these numbers now whenever there is rainy weather, we can place the number 1 in the transformed dataset to train the model. The results after this transformation are given in Table 7.

Table 7: Classifier Performance Evaluation

Algorithm	Accuracy
Multinomial Naïve Bayes	84%
Random Forest Classifier	88%
Linear SVC	89%
Logistic Regression	85%
Ridge Classifier	81%
Extremely Randomized Trees	79%
SGD Classifier	81%

CONCLUSION

Social media is producing an enormous amount of un-formatted data, and using manual methods of analysis are no longer feasible. This work presents the application of natural language processing in the area of the automated textual classification with a specific focus on identifying and classifying weather based data. Experimental results prove that the suggested methodology is effective in extracting and categorizing meteorological content using various approaches. In the application perspective, the study provides a basis of domain-specific data mining implementation. The research demonstrates the use of natural language based data pre-processing approaches and data modeling processes of regression and classification. The results demonstrate the effect of outsourcing sentiment analysis and the change in predictive performance based on column reduction. Further studies are encouraged to incorporate the use of advanced models, multilingual analysis methods, and real-time processing capabilities to enhance the strength and practical application of natural language based analysis system.

REFERENCES

[1] V. Kumari, A. F. Meghji, S. Bhatti, and D. Agha, Analyzing Temporal Variations and Public Sentiment on X: A Clustering Analysis of the 2024 Pakistan Elections, JICTRA, vol. 15, no. 1, pp. 1-16, Dec. 2025.

[2] E. Dogariu, S. Garg, B. Khadan, A. Potts, and M. Scornavacca, Using Machine Learning to Correlate Twitter Data and Weather Patterns, In Proc. IEEE MIT Undergraduate Research Technology Conf. (URTC), 2019, pp. 1-4.

[3] M. AminiMotlagh, H. Shahhoseini, and N. Fatehi, A Reliable Sentiment Analysis for Classification of Tweets in Social Networks, Social Network Analysis and Mining, vol. 13, no. 1, 2022, doi: 10.1007/s13278-022-00998-2.

[4] A. Alshamsi, R. Bayari, and S. Salloum, Sentiment Analysis in English Texts, Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 6, pp. 1683--1689, 2020.

[5] K. Purwandari, J. W. Sigalingging, T. W. Cenggoro, and B. Pardamean, Multi-Class Weather Forecasting from Twitter Using Machine Learning Approaches, Procedia Computer

Science, vol. 179, pp. 47--54, 2021.

[6] K. Anhsori and G. F. Shidik, A Comparative Analysis of Eight Machine Learning Models for Climate Change Sentiment Analysis, JST (Jurnal Sains dan Teknologi), vol. 14, no. 2, pp. 229-243, Jul. 2025, doi: 10.23887/jst-undiksha.v14i2.92672.

[7] S. García-Méndez, F. de Arriba-Pérez, A. Barros-Vila, and F. J. González-Castaño, Detection of Temporality at Discourse Level on Financial News by Combining Natural Language Processing and Machine Learning, Expert Systems with Applications, p. 116648, 2022.

[8] M. Alsurori, A. Enan, R. Alwan, W. Algumaei, S. Alturki, and E. Alkahtany, Machine Learning for Text Classification on Twitter: A Literature Review, 2023, doi: 10.11648/j.ajdmkd.20230801.12.

[9] F. Araujo, Y. Zhou, and U. Mukhija, Quantifying Correlation Between Weather and Twitter Public Sentiment, Columbia University, Tech. Rep., 2026.

[10] J. Qian, Z. Niu, and C. Shi, Sentiment Analysis Model on Weather Related Tweets with Deep Neural Network, in Proc. 10th Int. Conf. Machine Learning and Computing, Feb. 2018, doi: 10.1145/3195106.3195111.

[11] Y. Dzyuban et al., Sentiment Analysis of Weather-Related Tweets from Cities within Hot Climates," Weather, Climate, and Society}, vol. 14, no. 4, pp. 1133-1145, Oct. 2022, doi: 10.1175/WCAS-D-21-0159.1.

[12] Kamal, R. and Shah, M. A. and Maple, C. and Masood, M. and Wahid, A. and Mehmood, A., Emotion Classification and Crowd Source Sensing: A Lexicon Based Approach, IEEE Access, vol. 7, pp. 27124-27134, 2019, doi: 10.1109/access.2019.2892624.

[13] H. Henderi, S. Sofiana, A Machine Learning Approach to Indonesian Climate Change Sentiment Analysis Using Naive Bayes, IJIIS: International Journal of Informatics and Information Systems, vol. 8, pp. 33-43, 2025, doi: 10.47738/ijiis.v8i1.246.

[14] L. Biewald, T. Matthews, and W. Cukierski, Partly Sunny with a Chance of Hashtags. https://kaggle.com/competitions/crowdflower-weather-twitter, 2013. Kaggle.

[15] Y. Madani, M. Erritali, and B. Bouikhalene, Using Artificial Intelligence Techniques for Detecting Covid-19 Epidemic Fake News in Moroccan Tweets, Results in Physics, vol. 25, p. 104266, 2021.

[16] B. Saju, S. Jose and A. Antony, Comprehensive Study on Sentiment Analysis: Types, Approaches, Recent Applications, Tools and APIs, 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), Cochin, India, 2020, pp. 186-193, doi: 10.1109/ACCTHPA49271.2020.9213209.

[17] R. Lad, S. Mapari, and F. N. . Sibai, An experimental study to measure the environmental effect on human behavior and emotions using multi-output regression, J Integr Sci Technol, vol. 13, no. 6, p. 1143, May 2025, doi: 10.62110/sciencein.jist.2025.v13.1143.

[18] B. K. R. Mekala and N. K. B. Christopher, Popularity analysis of films in social media using logistic regression compared with linear regression algorithm, International Conference on Environmental Engineering and Material Sciences (ICEEMS-2021), vol. 2655, no. 1, p. 020033, 2023.

[19] M. Hasan, T. Ahmed, M. R. Islam, and M. P. Uddin, Leveraging textual information for social media news categorization and sentiment analysis, PLOS ONE, vol. 19, no. 7, p. e0307027, 2024.

[20] M. Danyal, S. Khan, M. Khan, M. Ghaffar, B. Khan, and M. Arshad, Sentiment analysis based on performance of linear support vector machine and multinomial naïve Bayes using movie reviews with baseline techniques, Journal on Big Data, vol. 5, p. 1, 2023.

[21] M. Umer, S. Sadiq, A. A. Eshmawi, M. Nappi, M. U. Sana, and I. Ashraf, ETCNN: Extra tree and convolutional neural network-based ensemble model for COVID-19 tweets sentiment classification, Pattern Recognition Letters, vol. 164, pp. 224-231, 2022.