Proceedings of the 57th Hawaii International Conference on System Sciences | 2024

Sentimental Analysis of Movie Tweet Reviews Using Machine Learning Algorithms

Hemanth Kumar Kari
Texas A&M University-Commerce
hemanthkumarkari@gmail.com

Abstract

Sentiment analysis stands as a prominent tool within microblogging platforms, gaining substantial traction as a means to discern public opinion and sentiment across various topics, including movie tweet reviews. In response to this demand, the study introduces a robust system architecture that incorporates an array of algorithms, ranging from Multinomial Naive Bayes and Support Vector Machine (SVM) to K-Nearest Neighbors (KNN), Bernoulli's Naive Bayes, and Random Forest. This architecture is meticulously trained using annotated Twitter data, methodically excluding non-opinionated content while precisely identifying sentiment. Thorough experimentation underscores the effectiveness of our methodology. To accomplish this, we curate an extensive data set of movie-related tweets, each carefully labeled with sentiments spanning positive, negative, or neutral tones. The methodological framework involves intricate text preprocessing steps, encompassing tokenization, stemming, and the removal of extraneous stop words. This facilitates the extraction of essential features and the conversion of raw text into numerical representations suitable for machine learning. Our sentiment classification modeling employs a diverse ensemble of machine learning algorithms, including Naive Bayes, Support Vector Machines, and Recurrent Neural Networks. The assessment involves a range of metrics such as accuracy, precision, recall, and F1-score, supported by rigorous techniques like cross-validation to enhance the dependability and robustness of results. Our unique contribution lies in the strategic deployment of algorithms and a resilient system architecture adept at surmounting the challenges inherent to microblogs. We emphasize the utmost importance of preprocessing in augmenting the precision of sentiment classification. This research substantiates the system's aptitude in extracting valuable insights for informed decision-making through the scrutiny of microblog sentiments.

Keywords: sentiment analysis, microblogging, machine learning, system architecture, experimental results, feature selection, accuracy metrics, and Confusion Matrix. .

1. Introduction

In the era of digitization, the surge in social media platforms has catalyzed a transformative wave, enabling individuals to voice their perspectives on a broad spectrum of subjects, including movies (Johnson, R., Zhang, T. 2015). However, within this expansive realm of online discourse, microblogs emerge as a dynamic landscape that demands a unique approach to sentiment analysis. This journey embarks on the intricate path of unraveling sentiment analysis intricacies within the realm of microblogs, where the casual and fluid nature of communication introduces layers of complexity. Microblogs present a unique challenge to traditional sentiment analysis methodologies, given their informal nature and prevalence of linguistic nuances. This necessitates an innovative approach that transcends mere algorithmic comparisons. Our endeavor strives to delve into this complexity and establish an intricate understanding of sentiment within microblogging platforms. Amid the continuous stream of movie-related tweets, a critical demand arises to efficiently distill meaningful insights from this data influx (Johnson, R., & Zhang, T. (2015)). This imperative has spurred my exploration into the realm of sentiment analysis, where the prowess of machine learning algorithms becomes a guiding light. However, our research aims

URI: https://hdl.handle.net/10125/107012
978-0-9981331-7-1

to transcend the boundaries of algorithmic selection; it strives to establish an ingenious methodology calibrated to the nuanced landscape of microblogs. In pursuit of this ambition, we unveil a sophisticated and robust system architecture that signifies innovation in microblog sentiment analysis. Our methodology harnesses a meticulously curated ensemble of machine learning techniques, encompassing Multinomial Naive Bayes, SVM, KNN, Bernoulli's Naive Bayes, and Random Forest. Yet, our contribution doesn't end with algorithmic selection; it resides in the symphony of these methods operating harmoniously. These algorithms aren't isolated tools; they synergize to navigate informal language nuances, abbreviations, and the implicit sentiments often woven into microblog discourse (Howard, J., & Ruder, S. (2018)). However, the strength of our system isn't limited to algorithmic dexterity; it stems from a robust foundation. Our system is meticulously trained on annotated Twitter data, a meticulous process encompassing the capture of opinionated messages and the deconstruction of sentiment intensity and direction. This careful annotation acts as a guiding beacon, leading our system through the intricate maze of microblogs' complexity. The application of these algorithms culminates in the categorization of tweets into positive, negative, or neutral sentiments, providing a comprehensive understanding of public sentiment toward movies (Zhang, Y., & Wallace, B. (2015)). This insight extends its value to movie makers, marketing teams, and enthusiasts, empowering them to make well-informed decisions about movie selection and strategic marketing (Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017)). Over the past decade, sentiment analysis has garnered substantial momentum, particularly within the realm of social media. Twitter has emerged as a crucial platform for scrutinizing public opinions and sentiments directed at movies (Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016)). Amidst the array of machine learning algorithms, specific exemplars like Naive Bayes and Support Vector Machines (SVM) have emerged as adept tools for extracting valuable insights from tweets (Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017))(Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019)). Anchored in robust statistical models, these algorithms enable precise tweet classification into categories—positive, negative, or neutral—paving the way for quantitative sentiment analysis (Sun, C., Shang, L., Korhonen, A., & Zhao, D. (2019))(Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019))(Suresh Kumar, Dhruv Veragi, Kamlesh Sharma, and Ankit Juyal. (2022)). The research trajectory encompasses a series of critical stages, commencing with meticulous data preprocessing involving noise reduction, tokenization, and stemming (Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013))(Kim, Y. (2014)). This journey culminates in the systematic application of machine learning algorithms, allowing tweets to be classified into distinct sentiment categories (dos Santos, C., & Gatti, M. (2014)). The culmination of this study manifests in the presentation of outcomes, showcasing the precision of employed machine learning models and illuminating insightful revelations about the spectrum of public sentiment across diverse movies. The pursuit of comprehending sentiments conveyed within movie-related tweets, interpreted through the lens of advanced machine learning algorithms represents a pivotal avenue of research poised to uncover profound insights into public opinions surrounding movies (Tang, D., Qin, B., & Liu, T. (2015)). Our research endeavors to make a substantive contribution by meticulously dissecting public sentiment in the context of movies. At the heart of this exploration lies a commitment to unravel the intricate interplay between sentiment, microblogs, and sophisticated machine learning techniques." The paper's following sections are organized as such: Section 2 presents a survey of the preexisting literature relevant to the proposed work. Section 3 elaborates on the machine learning methods. Section 4 expounds on the proposed methodology, focusing on the investigation of the network model. Within Section 5, the dataset is detailed, while Section 6 provides an account of the outcomes and conversations pertaining to the performance of the suggested model. Lastly, Section 7 concludes the paper.

1.1. Applications:

• Marketing and Advertising: Sentiment analysis of movie tweet reviews can guide movie studios and production companies in gauging public opinion, enabling them to make informed marketing and advertising decisions. This can help in targeting specific audience segments and tailoring advertising campaigns accordingly to tailor marketing strategies accordingly. Positive reviews can be harnessed for promotional purposes, while negative reviews can offer insights for improvement.

• Movie Criticism: Movie critics can employ sentiment analysis of movie tweet reviews to craft their assessments, providing a more objective evaluation of movies.

• Recommendation Systems: Movie tweet reviews can

underpin recommendation systems, suggesting movies to users based on their preferences and viewing history.
• Social Listening: Social listening refers to monitoring social media conversations to identify trends and insights. Movie studios and production companies can use sentiment analysis of movie tweet reviews to gain insights into audience preferences, interests, and behavior.
• Academic Research: Researchers in the fields of linguistics, psychology, and computer science can use sentiment analysis of movie tweet reviews to study human behavior and language use in the context of movie reviews.

2. Literature Survey

The surge of digital communication has brought about a notable rise in social media platforms, empowering individuals to openly express their opinions across a broad spectrum of topics, including movies (Johnson & Zhang, 2015). This phenomenon has spurred the exploration of sentiment analysis or opinion mining within the domain of natural language processing (NLP). In particular, the analysis of people's sentiments on platforms like Twitter and Facebook have become invaluable to businesses, policymakers, and researchers, offering a wealth of insights. The crux of sentiment analysis lies in automatically classifying sentiments conveyed in given contexts as positive, negative, or neutral (Johnson & Zhang, 2015). To achieve These machine-learning algorithms have gained widespread popularity, offering a potent strategy for text classifications in sentiment analysis. Notably, Howard and Ruder (2018) introduce "Universal Language Model Fine-tuning" for text classification, while Zhang and Wallace (2015) conduct a sensitivity analysis of convolutional neural networks for sentence classifications. Vaswani et al. (2017) proposes the "Attention Is All You Need" architecture, a seminal work in neural machine translation. Yang et al. (2016) developed" Hierarchical Attention Networks" for document classification, and Johnson and Zhang (2015) focus on effective word order utilizations with convolutional neural networks. Additionally, Conneau et al. (2017) delve into supervised learning of universal sentences representations, and Radford et al. (2019) present the concepts of language models as unsupervised multitask learners. Machine learning models have also been applied to specific domains. Johnson et al. (2016) contribute the "MIMIC-III" critical care database for medical research, while Sun et al. (2019) utilize BERT for aspect-based sentiment analysis. Yang et al. (2019) introduce "XLNet," a generalized autoregressive pretraining approach for language understandings. Furthermore, the work of Suresh Kumar et al. (2022) focuses on "Sentimental Analysis of Movie Tweet Reviews Using Machine Learning Algorithms." Deep learning techniques have made significant strides in sentiment analysis. Socher et al. (2013) proposes recursive deep models for semantic compositionality, and Kim (2014) explores convolutional neural networks for sentence classifications. dos Santos and Gatti (2014) employ deep convolutional neural networks for sentiment analysis of short texts, and Tang et al. (2015) introduces gated recurrent neural networks for document modeling. Zhang et al. (2015) present character-level convolutional networks for text classifications, and Maas et al. (2011) contribute to learning word vectors for sentiments analysis. Additionally, dos Santos and Gatti (2015) enhances named entity recognition with neural character embeddings, and Pennington et al. (2014) proposes "GloVe" for global vectors in word representation. Devlin et al. (2018) presents "BERT," a pre-training approachs for deep bidirectional transformers in language understandings. In the realm of software reliability modelings, Roy et al. (2019) introduced a noteworthy 'Single Change Point Hazard Rate Software Reliability Model with Imperfect Debugging,' Roy, P., Mahapatra, G. S., Rani, P., Pandey, S. K., Dey, K. N. (2019). A Single Change Point Hazard Rate Software Reliability Model with Imperfect Debugging. In Proceedings of the IEEE, Conference Location: Orlando, FL, USA, doi: 10.1109/SYSCON.2019.8836816. shedding light on the complexities of software reliability assessment.

3. Methods of Machine Learning

In natural language processing, Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) are important techniques and common approaches used for creating features from Data in natural language processing (NLP) and machine learning (ML) presented in text format.

3.1. Bag of Words:

The BoW is a technique that represents the occurrence of words within a text document as a vector, regardless of their order or context. It involves Representing each document in a text collection, and a vocabulary of all unique words in the corpus is created. This involves creating a vector of word frequencies for each document where each entry in the vector corresponds to a unique word in the vocabulary.

3.2. TF-IDF (Term Frequency-Inverse Document Frequency):

TF-IDF is a way to measure how important a word is in a document compared to other documents in a corpus, as shown in Table 1: Comparison Of Performance Metrics For Different Algorithms. This is useful in natural language processing and information retrieval. To calculate the TF of a word in a document, the number of times the word appears in the document is divided by the total number of words in the document. This gives a measure of how often the word appears in that specific document. The TF-IDF score is then calculated by multiplying the TF and IDF values. This measures the importance of the word in the document compared to the corpus. Words with high TF-IDF scores are considered important to the document, while those with low scores are less important. The TF-IDF technique is used in text classification, sentiment analysis, and information retrieval. It is particularly useful in cases where simple word frequency counts (such as in the bag-of-words model) may not capture the semantic meaning of words. TF-IDF is an effective way to reduce the impact of common words that appear in many documents and give more weight to rare and important words. Ex. Bag-of-Words (BoW) and TF-IDF for Creating Features from Text:

Review	This	Movie	Is	Slow	Spooky
Review 1	1	1	1	0	0
Review 2	1	1	2	0	1
Review 3	1	1	1	0	1

Table 1. Comparison of Performance Metrics for Algorithms

3.3. Support Vector Machine:

The Support Vector Machines (SVM) algorithm is a commonly used machine learning technique in sentiment analysis tasks, including the analysis of movie tweet reviews. SVMs are effective in this project for several reasons. Firstly, SVMs are particularly good at handling high-dimensional datasets, which is often the case in sentiment analysis tasks. This is because SVMs can separate data points in high-dimensional space by identifying the hyperplane that maximizes the gap between the two sets of data points.

This makes SVMs effective in separating positive and negative movie tweet reviews. Secondly, SVMs have a good generalization performance, meaning that they can accurately classify new, unseen data. This is because SVMs have a robust regularization framework that helps to prevent overfitting and improves their ability to generalize to new data. Finally, SVMs are capable of handling non-linear data by using kernel functions, such as the radial basis function kernel. This is useful in sentiment analysis tasks because sentiment can often be expressed in subtle and complex ways, requiring a model that can handle non-linear relationships is defined in spieworks

Support Vector Machines (SVMs) are a robust machine learning algorithm used for classification and regression tasks. As shown in Figure 1 below, they operate by finding the optimal decision boundary in a scatter plot, represented by a line (or hyperplane in higher dimensions), which maximizes the margin between two classes of data points in Class A and Class B. The support vectors, which are the closest data points to this boundary, significantly influence its determination. SVMs can handle non-linear data by transforming it into a higher-dimensional space using kernel functions. In essence, SVMs excel at finding the best separation between classes, making them valuable in a wide range of applications where accurate classification and pattern recognition are vital.

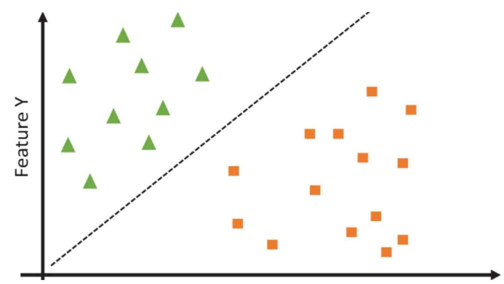

Figure 1. Support Vector Machine Source

The Support Vector Machines (SVM) algorithm is a well-known and widely used supervised machine learning approach for classification and regression tasks. In classification, SVM aims to find an optimal hyperplane. The goal is to create maximum separation between the data points of various categories. The formula and example of SVM are as follows:

Formula: For a binary classification problem, the formula for SVM can be written as:

$$f(x) = \text{sign}(w^T \cdot x + b)$$

Where:

- $f(x)$ represents the predicted class label for a new data point x.
- `sign()` is the sign function that returns -1 for negative values and +1 for positive values.
- w is the weight vector.

- b is the bias term.

The SVM algorithm aims to find the optimal values for w and b by solving an optimization problem, typically using techniques such as quadratic programming.
Let's consider a simple binary classification problem with two features (x_1 and x_2) and two classes (Class A and Class B). We have the following training dataset as formatted in Table2: EXAMPLE OF SIMPLE BINARY CLASSIFICATION PROBLEM

Data Point	x_1	x_2	Class
Data 1	2	3	Class A
Data 2	4	1	Class A
Data 3	1	5	Class B
Data 4	4	7	Class B

Table 2. Example of Simple Binary classification problem

The goal is to find the optimal hyperplane that separates the two classes. In this case, we can see that a linear decision boundary can separate the classes well.

3.4. Multinomial Naive Bayes:

Multinomial Naive Bayes is a variant of the Naive Bayes algorithm that is specifically designed for discrete features, particularly for text classification tasks. It assumes that features follow a multinomial distribution, such as word counts in a document. Here's an example of Multinomial Naive Bayes in machine learning.

The formula for Multinomial Naive Bayes can be represented as:

$$P(y|x_1, x_2, ..., x_n) = P(y) \times \prod_{i=1}^{n} P(x_i|y)^{x_i}$$

Where:

- $P(y|x_1, x_2, ..., x_n)$ is the probability of class y given the features $x_1, x_2, ..., x_n$.
- $P(y)$ is the prior probability of class y.
- $P(x_i|y)$ is the conditional probability of feature x_i given class y.

3.5. K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a popular machine learning algorithm used for both classification and regression tasks. It works based on the idea that similar data points tend to belong to the same class or have similar output values. Let's consider a classification problem where we want to predict whether a given flower is an Iris- setosa, Iris-versicolor, or Iris-virginica based on its sepal length and sepal width. We have a training dataset with the following samples as shown in Table 3: Example Of Classification Problem.

Sample	Sepal Length	Sepal Width	Class
1	5.1	3.5	Iris-setosa
2	4.9	3.0	Iris-setosa
3	6.7	3.1	Iris-versicolor
4	5.8	2.7	Iris-versicolor
5	6.9	3.2	Iris-virginica
6	5.6	2.8	Iris-virginica

Table 3. Example of Classification Problem

Now, let's say we want to classify a new sample with sepal length 5.5 and sepal width 3.0 using the KNN algorithm with k = 3 (i.e., considering the three nearest neighbors).

1. Calculate the Euclidean distance between the new sample and each training sample:

- Sample1: $\sqrt{(5.1 - 5.5)^2 + (3.5 - 3.0)^2} \approx 0.5$
- Sample 2: $\sqrt{(4.9 - 5.5)^2 + (3.0 - 3.0)^2} \approx 0.6$
- Sample 3: $\sqrt{(6.7 - 5.5)^2 + (3.1 - 3.0)^2} \approx 1.2$
- Sample 4: $\sqrt{(5.8 - 5.5)^2 + (2.7 - 3.0)^2} \approx 0.3$
- Sample 5: $\sqrt{(6.9 - 5.5)^2 + (3.2 - 3.0)^2} \approx 1.4$
- Sample 6: $\sqrt{(5.6 - 5.5)^2 + (2.8 - 3.0)^2} \approx 0.2$

2. Select the k nearest neighbors based on the shortest distances:

- Nearest Neighbors: Sample 4, Sample 6, Sample 1

3. Determine the majority class among the k nearest neighbors:

- Class of Sample 4: Iris-versicolor
- Class of Sample 6: Iris-virginica
- Class of Sample 1: Iris-setosa

Since two out of three nearest neighbors belong to the class Iris-versicolor, and one belongs to Iris-setosa, the predicted class for the new sample would be Iris-versicolor.

Note: The choice of k, the number of neighbors to consider, can have an impact on the classification result. It is typically chosen based on experimentation and validation on the dataset.

3.6. Bernoulli Naïve Bayes:

The Bernoulli Naive Bayes algorithm is a variant of the Naive Bayes classifier that is specifically designed for binary features. It assumes that each feature follows a Bernoulli distribution, which means it takes on values of 0 or 1.

$$P(y|x_1, x_2, ..., x_n) = P(y) \times \prod_{i=1}^{n} (P(x_i|y)^{x_i} \times (1 - P(x_i|y))^{(1-x_i)})$$

Where:

- $P(y|x_1, x_2, ..., x_n)$ is the probability of class y given the binary features $x_1, x_2, ..., x_n$.
- $P(y)$ is the prior probability of class y.
- $P(x_i|y)$ is the conditional probability of feature x_i given class y.
- x_i represents the binary value of feature i (0 or 1).

To illustrate with an example, let's consider a spam classification problem. We have a dataset with emails labeled as spam ($y = 1$) or not spam ($y = 0$), and we want to classify new emails as spam or not spam based on their binary features (presence or absence of certain keywords).

Example: Suppose we are presented with a training dataset, as exemplified in Table 4, that represents a classic problem in email classification: distinguishing between spam and legitimate emails. In this dataset, each email is tagged with a label indicating whether it is classified as spam (denoted as "1") or not spam (denoted as "0"). This type of problem is a quintessential binary classification task, where the goal is to develop a machine learning model capable of accurately categorizing incoming emails as either spam or not spam based on their content and characteristics. To achieve this, we will leverage various features extracted from the emails and employ machine learning algorithms, such as Naive Bayes or Support Vector Machines, to train the model. The objective is to create a robust and accurate spam email filter, which is invaluable in managing and prioritizing emails in real-world email systems.

Now, let's calculate the probabilities needed for the formula. First, we calculate the prior probabilities:

- $P(\text{Spam} = 1) = \frac{3}{5} = 0.6 \quad P(\text{Spam} = 0) = \frac{2}{5} = 0.4$

Next, we calculate the conditional probabilities for each feature given the class: P(Spam = 1) = 3/5 = 0.6 P(Spam = 0) = 2/5 = 0.4

Email	Contains "buy"	Contains "free"	Spam
1	1	0	1
2	0	1	1
3	1	1	1
4	0	1	0
5	0	0	0

Table 4. Example of spam Email classification problem

Next, we calculate the conditional probabilities for each feature given the class:

- $P(\text{Spam} = 1) = \frac{3}{5} = 0.6 \quad P(\text{Spam} = 0) = \frac{2}{5} = 0.4$

Next, we calculate the conditional probabilities for each feature given the class:

- $P(\text{Buy} = 1|\text{Spam} = 1) = \frac{3}{4} = 0.75 \quad P(\text{Buy} = 0|\text{Spam} = 1) = \frac{1}{4} = 0.25$
- $P(\text{Free} = 1|\text{Spam} = 1) = \frac{2}{4} = 0.5 \quad P(\text{Free} = 0|\text{Spam} = 1) = \frac{2}{4} = 0.5$
- $P(\text{Buy} = 1|\text{Spam} = 0) = \frac{0}{1} = 0 \quad P(\text{Buy} = 0|\text{Spam} = 0) = \frac{1}{1} = 1$
- $P(\text{Free} = 1|\text{Spam} = 0) = \frac{2}{2} = 1 \quad P(\text{Free} = 0|\text{Spam} = 0) = \frac{0}{2} = 0$

Now, let's classify a new email with the following features:

Contains "buy"	Contains "free"
1	0

To calculate the probability of this email being spam and not spam, we apply the formula:

$$P(Spam = 1|Buy = 1, Free = 0) = P(Spam = 1) \times P(Buy = 1|Spam = 1) \times P(Free = 0|Spam = 1)$$
$$= 0.6 \times 0.75 \times 0.5 = 0.225(Approx.)$$

3.7. Random Forest

Random Forest is an ensemble learning algorithm, and it can be employed effectively in the sentiment analysis of movie-based tweets using machine learning techniques. Data Preparation, Feature Extraction, Training Data Split, Random Forest Model, Voting (Classification), Evaluation, Predictions, and Anticipating Movie Success Utilized by Random Forest.

To build a Random Forest model, we generate multiple decision trees, each trained on a different subset of the training data and a random subset of features. The predictions of these trees are combined using voting (classification) or averaging (regression) to obtain the final prediction.

Formula: Classification The Random Forest algorithm combines the predictions of individual decision trees using voting. For classification tasks, the formula for Random Forest prediction is:

$$\text{Final Prediction} = \text{Mode}(\text{Prediction of individual} \tag{1}$$
$$\text{decision trees}) \tag{2}$$

Where:

- Prediction of individual decision trees refers to the class predicted by each decision tree in the Random Forest.
- Mode is the statistical function that returns the most frequent class in the predictions of all decision trees.

Formula: Regression For regression tasks, the formula for Random Forest prediction is:

$$\text{Final Prediction} = \text{Mean}(\text{Prediction of individual} \tag{3}$$
$$\text{decision trees}) \tag{4}$$

Prediction of individual decision trees refers to the numerical prediction made by each decision tree in the Random Forest. Mean is the mathematical function that calculates the average of the predictions from all decision trees. In both classification and regression tasks, the Random Forest algorithm leverages the wisdom of multiple decision trees, each trained on a different subset of the training data and a random subset of features. The final prediction is obtained by either taking the majority vote (mode) in classification or calculating the average (mean) in regression. This ensemble approach enhances the model's performance and robustness, making Random Forest a popular choice in various machine learning tasks.

$$\text{Final Prediction} =$$
$$\frac{1}{N}\sum_{i=1}^{N} \text{Prediction of individual decision trees} \tag{5}$$

Where:

- Prediction of individual decision trees refers to the numerical output predicted by each decision tree in the Random Forest.
- Average is the mathematical function that calculates the mean of the predictions of all decision trees.

Note: Random Forest also considers the concept of "out-of-bag" (OOB) samples, which are data points that are not included in the training set of a particular decision tree. These samples can be used to estimate the accuracy of the Random Forest model during training.

4. Methodology

The methodology proposed for conducting sentiment analysis on movie-related tweet reviews through the utilization of machine learning algorithms involves a sequence of essential stages. The process commences by gathering a dataset comprising reviews from microblogs related to movies, forming the core dataset for subsequent sentiment analysis. The collected dataset undergoes initial preprocessing, encompassing the elimination of superfluous details like URLs and usernames. The remaining textual content is then transformed into a machine-readable format, optimized for effective analysis via machine learning algorithms. Following the preprocessing phase, the dataset is segregated into distinct sets: a training set and a testing set. The training set is designated for training the machine learning model, whereas the testing set serves as a means to gauge the model's accuracy and performance. A diverse array of machine learning algorithms, ranging from Naive Bayes to Support Vector Machines (SVM) and Neural Networks, can be employed based on dataset attributes and performance criteria. Upon the successful completion of model training and evaluation, the model is equipped to forecast sentiment in new movie-related tweet reviews. These predictions can be harnessed to uncover temporal sentiment trends in movie reviews, spotlight movies with exceptionally positive or negative reception, and even anticipate the potential triumph of upcoming movies through sentiment analysis. In its entirety, this methodology underscores its ability to furnish valuable insights into the sentiment harbored within movie-related tweet reviews, effectively harnessing the capabilities of machine learning algorithms.

Additionally, as shown in Figure 2 above, the system architecture is designed to efficiently process and analyze a vast stream of movie-related tweet reviews. This architecture encompasses data ingestion, pre-processing, feature extraction, model inference, and result visualization stages, all seamlessly integrated to provide real-time sentiment predictions.

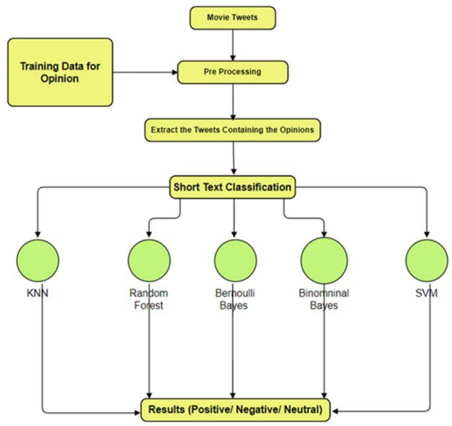

Figure 2. System Architecture

4.1. Data Collection

In this phase, the process kicks off by assembling an extensive dataset of movie tweet reviews derived from social media platforms, with Twitter being a prominent source. This dataset should demonstrate variety, encompassing both positive and negative reviews to maintain a balanced reflection of sentiments. The procurement can be facilitated using APIs provided by the platforms in question. The primary aim is to gather a considerable amount of textual data that accurately mirrors the sentiments conveyed by users in real-life scenarios.

4.2. Data Preprocessing

The preprocessing of data constitutes a pivotal stage, indispensable for enhancing the dataset's quality and facilitating precise sentiment analysis. This phase encompasses several sequential actions:

Noise Elimination: Extraneous components like symbols, punctuation, and special characters are eradicated from the text. This measure ensures that the ensuing analysis centers solely on pertinent textual content.

Management of Usernames and URLs: References, hashtags, and URLs are excluded or substituted to avert any undue influence on sentiment classification. These elements don't contribute to sentiment insight and can introduce interference.

Treatment of Retweets: Retweets and their corresponding counts are addressed. As retweets essentially replicate content, they can skew sentiment analysis. The decision to retain retweets or focus solely on original tweets hinges on the analysis objectives.

Textual Normalization: Techniques such as stemming and lemmatization are deployed to condense words into their base forms. This curtails dimensionality and heightens the uniformity of the dataset.

4.3. Data Splitting

In this section, following the preprocessing phase, the dataset is partitioned into two distinct subsets: the training set and the testing set. The training set serves the purpose of instructing machine learning models, providing them with labeled data to grasp patterns and correlations between features and sentiments. In contrast, the testing set remains segregated and is employed to evaluate the model's capacity to generalize and accurately predict outcomes on new, unseen data instances.

4.4. Feature Extraction

Feature extraction takes the processed textual data and converts it into numerical vectors that machine learning algorithms can process. Methods such as Bag of Words (BoW) transform each review into a vector, where each dimension corresponds to a distinct word present in the dataset. Additionally, Term Frequency-Inverse Document Frequency (TF-IDF) assigns significance to words based on their relevance within individual reviews compared to their frequency across the entire dataset.

Certainly, let's expand on how the different classifiers are involved in the proposed sentiment analysis methodology and how their use can benefit the experiment.

4.5. Model Training

In this phase, the preprocessed and transformed data is utilized to train multiple machine learning classifiers. Each classifier, namely Bernoulli Naive Bayes, Multinomial Naive Bayes, Random Forest, k-Nearest Neighbors (k-NN), and Support Vector Machines (SVM), learns patterns in the data that associate the extracted features with the respective sentiment labels (positive, negative, or neutral).

Bernoulli Naive Bayes and Multinomial Naive Bayes: These are probabilistic classifiers that assume features are conditionally independent given the class. They estimate the probability distribution of words in each class based on the training data. They are particularly useful for text classification tasks where the features are discrete and occur multiple times (Multinomial) or just once (Bernoulli).

Random Forest: Random Forest is an ensemble classifier that combines multiple decision trees to improve classification accuracy and reduce overfitting. Each tree is trained on a different subset of the training data, and the final classification is determined by aggregating the outputs of individual trees. Random Forest can capture complex relationships between features and sentiments, enhancing predictive power.

k-Nearest Neighbors (k-NN): k-NN is a simple instance-based classifier that classifies data points based on the class labels of their k-nearest neighbors in the training data. It's advantageous for capturing local patterns in the data and can perform well when the data distribution is non-linear.

Support Vector Machines (SVM): SVM is a powerful classifier that constructs a hyperplane to separate different classes while maximizing the margin between them. It's particularly effective in cases where the classes are not linearly separable. SVM can handle high-dimensional data and can capture complex decision boundaries.

4.6. Model Evaluation

Each trained classifier's performance is evaluated using the testing set. Metrics such as accuracy, precision, recall, and F1-score provide insights into their predictive capability and generalization to unseen data.

Accuracy: Accuracy measures the proportion of correctly classified instances over the total instances. It provides an overall assessment of a model's correctness.

Precision: Precision quantifies the proportion of correctly predicted positive cases (true positives) relative to all predicted positive cases (true positives + false positives). It's crucial when minimizing false positive predictions is important.

Recall: Recall calculates the proportion of actual positive instances (true positives) that are correctly predicted by the model. It's vital when minimizing false negatives is a priority.

F1-score: The F1-score is the harmonic mean of precision and recall, offering a balanced measure between the two. It's particularly useful when there's an imbalance between classes.

4.7. Model Selection

The evaluation metrics guide the selection of the best-performing classifier. The classifier that demonstrates the highest accuracy, precision, recall, or F1 score on the testing set is chosen for deployment. Let's see the benefits of Using Multiple Classifiers:

Robustness: Different classifiers handle data in distinct ways. By employing multiple classifiers, the methodology becomes more robust to variations in data and increases the likelihood of capturing diverse patterns.

Comparison and Selection: Using multiple classifiers allows for a direct comparison of their performance. This helps identify the most suitable classifier for the specific sentiment analysis task.

Enhanced Accuracy: Different classifiers excel in different scenarios. By utilizing multiple classifiers, the ensemble can provide higher accuracy by combining the strengths of each classifier.

Generalization: A classifier's performance on the testing set gauges its generalization ability to unseen data. Selecting the best classifier ensures that predictions remain accurate when applied to new movie tweet reviews.

4.8. Model Deployment

By employing a variety of classifiers in the methodology, the experiment benefits from a comprehensive exploration of different approaches, resulting in a more refined and reliable sentiment analysis model for movie tweet reviews.

5. Data Set

We used Twitter's API to collect tweets related to the Bollywood movie Kabir Singh. We collected a total of tweets over a period of two weeks. We used a Python script to connect to Twitter's API and fetch the tweets based on specific search queries. The tweets were then stored in a CSV file for further processing. Pictorial representation is defined in Figure 3 below. It is important to ensure that the dataset you use is appropriate for your specific use case and contains feedback that is similar in nature and context to what you expect to receive from your own API. Additionally, it is recommended to use a large and diverse dataset for best results.

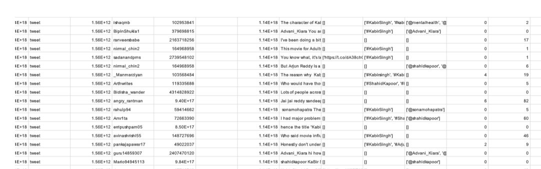

Figure 3. Data Set

6. Results and Discussions

Highlighting Notable Disparities in Outcomes: Conventional vs. Innovative Approach is followed. Algorithmic Achievements in Our Proposed System:

1. KNN, SVC, Multinominal Naive Bayes, Bernoulli Naive Bayes, and Random Forest were introduced as the new algorithms.

2. The new algorithms aimed to improve the accuracy of sentiment analysis on microblogging words.

3. The accuracy achieved with the new algorithms showed notable improvement compared to the old algorithms.

4. Among the new algorithms, KNeighborsClassifier achieved an accuracy of 0.8777, and SVC achieved an accuracy of 0.9670, MultinomialNB achieved an accuracy of 0.8302, BernoulliNB achieved an accuracy of 0.8059, and RandomForestClassifier achieved an accuracy of 0.9640.

5. The introduction of new algorithms expanded the range of options for sentiment analysis, allowing for better classification and understanding of microblogging word sentiments. Table 5 below illustrates the Comparison Of Performance Metrics For Different Algorithms.

	SVM	RF	MNB	BNB	KNN
Accuracy	0.94	0.96	0.86	0.84	0.85
Recall	0.96	0.95	0.85	0.86	0.86
Precision	0.95	0.91	0.86	0.85	0.84
F1 Score	0.93	0.93	0.84	0.84	0.83

Table 5. Comparison of Performance Metrics for Different Algorithms

The Sentiment Analysis of Movie Tweet Reviews Using Machine Learning Algorithms project involves analyzing tweets related to movies to determine the sentiment of the tweets. The dataset for this project consists of a collection of tweets that mention a particular movie, along with their corresponding sentiment labels (positive, negative, or neutral). The dataset was created by manually annotating a subset of tweets from the Twitter API.

In the context of our sentiment analysis framework, it is essential to evaluate the performance of various machine learning algorithms to determine their efficacy in classifying movie-related tweet reviews accurately. As illustrated in Figure 4 comparison graph between the classifiers, our experimentation with different classifiers has yielded distinct accuracy results. Notably, the Support Vector Classifier (SVC) emerged as the top-performing model, achieving an impressive accuracy of 0.9670. Following closely, the Random

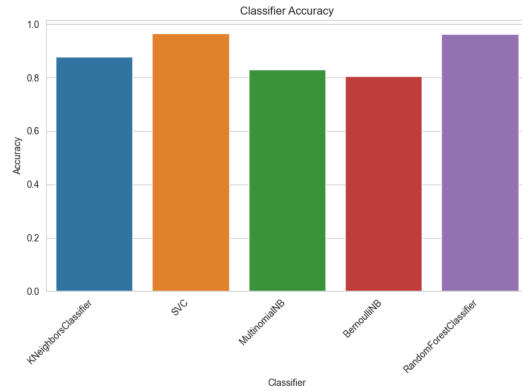

Figure 4. Comparison Graph between the classifiers

Forest Classifier demonstrated notable proficiency with an accuracy of 0.9640. In contrast, the K Neighbors Classifier, while commendable, achieved a slightly lower accuracy of 0.8777. The Multinomial NB and Bernoulli NB models, while delivering acceptable performances, displayed accuracy of 0.8302 and 0.8059, respectively. These findings serve as a pivotal reference point in selecting the most suitable classifier for our sentiment analysis task, highlighting the significance of model selection in optimizing the accuracy of movie-related sentiment predictions.

7. Conclusion

In summary, this study concluded by detailing the meticulous gathering of tweets related to a specific movie through the utilization of Twitter's API. This approach has proven effective in reconstructing microblogs, which is a valuable component of this research paper. Following this, a thorough preprocessing phase was undertaken to ensure the dataset's compatibility with machine learning algorithms. The subsequent training and evaluation of various machine learning models revealed that Random Forest achieved a peak accuracy of 96.40

The introduction of an inventive system framework enabled the harnessing of diverse machine learning algorithms, encompassing Support Vector Machine, Multinomial Naive Bayes, KNN, Bernoulli's Naive Bayes, and Random Forest. Through the application of techniques such as Grid Search, fine-tuning was performed for parameter optimization. The system's training was carried out using carefully annotated Twitter data, meticulously filtering out non-opinionated content while accurately pinpointing sentiment orientation.

The outcomes of the experimental phase effectively underscored the potency of the newly proposed system. With the incorporation of innovative algorithms—namely KNN (87.77 percent), SVC (96.70 percent), Multinomial Naive Bayes (83.02 percent), Bernoulli Naive Bayes (80.59 percent), and Random Forest (96.40 percent) significant advancements were made in sentiment classification compared to traditional algorithms. This infusion of cutting-edge algorithms not only broadened the horizons of sentiment analysis but also markedly improved the precision of sentiment classification for microblog words.

The presentation of experimental findings encompassed a range of metrics—accuracy, recall, precision, and F1 score—effectively underlining the adeptness of the system in accurately categorizing tweets into positive, negative, or neutral sentiments. Comparative assessments against established methods such as Logistic Regression, Linear SVC, and Stochastic Gradient highlighted the system's superior performance. In summation, the inventive system, fortified by the integration of new algorithms, showcased its excellence in surpassing existing methods, further establishing the efficacy of combining machine learning with microblog sentiment analysis. This framework holds substantial potential for unraveling public sentiment regarding movies, offering invaluable insights for data-driven decision-making within the movie production and marketing domains. As we look ahead, the implementation of advanced natural language processing techniques, expanded datasets, and supplementary features present a promising path toward achieving enhanced sentiment analysis precision and delving deeper into insights.

8. Acknowledge

I would like to express my sincere gratitude to those who contributed to the completion of this research project. Without their support, this work would not have been possible. First and foremost, I would like to thank my advisor, Dr. S Suh, and also acknowledge Professor Dr. Pooja Rani for their support, which made it possible to conduct this study for their invaluable guidance, mentorship, and unwavering support throughout the research process. Their expertise and insightful feedback significantly improved the quality of this paper. Furthermore, I extend my appreciation to the participants of our study, whose willingness to share their experiences and knowledge was instrumental in gathering critical data for this research. Finally, I want to express my heartfelt thanks to my family for their unwavering encouragement and understanding throughout this academic journey.

Conneau et al. (2017), Devlin et al. (2018), Dos Santos and Gatti (2014), Howard and Ruder (2018), Johnson et al. (2016), Kim (2014), Kumar et al. (2022), Maas et al. (2011), Pennington et al. (2014), Radford and Wu (2019), Rani and Mahapatra (2019), Santos and Guimaraes (2015), Socher et al. (2013), Sun et al. (2019), Tang et al. (2015), Vaswani et al. (2017), Yang et al. (2019), Yang et al. (2016), X. Zhang et al. (n.d.), and Y. Zhang and Wallace (2015)

References

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts, 69–78.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., & Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, *3*(1), 1–9.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kumar, S., Veragi, D., Sharma, K., & Juyal, A. (2022). Sentimental analysis of movie tweet reviews using machine learning algorithms. *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, *26-27*.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis, 142–150.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation, 1532–1543.

Radford, A., & Wu, J. (2019). Rewon child, david luan, dario amodei, and ilya sutskever. 2019. *Language models are unsupervised multitask learners. OpenAI blog*, *1*(8), 9.

Rani, P., & Mahapatra, G. (2019). A single change point hazard rate software reliability model with imperfect debugging, 1–7.

Santos, C. N. d., & Guimaraes, V. (2015). Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank, 1631–1642.

Sun, C., Huang, L., & Qiu, X. (2019). Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*.

Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification, 1422–1432.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification, 1480–1489.

Zhang, X., Zhao, J., & LeCun, Y. (n.d.). Character-level convolutional networks for text classification. *Advances in neural information processing systems*, *28*.

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.