Enhancing Sentiment Analysis Accuracy by Optimizing Hyperparameters of SVM and Logistic Regression Models

Yellu Siri¹, Suhail Afroz², and Rella Usha Rani³*

¹Student, CSE, CVR College of Engineering, Hyderabad
²Associate Professor, CSE Department, CVR College of Engineering, Hyderabad
³Professor, CSE (AI&ML) Department, CVR College of Engineering, Hyderabad

Abstract. The Analysis of Sentiments expressed on Twitter is a widely practiced application of Natural Language Processing (NLP) and Artificial Intelligence (AI). This process involves examining tweets to determine the emotional tone conveyed within the message. AI-based approaches are employed in Twitter sentiment analysis, typically following these steps: Data Collection, Data Preprocessing, and Sentiment Analysis, where AI techniques like Support Vector Machines (SVM) and Logistic Regression are utilized to categorize tweets into positive, negative, or neutral sentiments. Twitter data is a valuable source of information, serving diverse purposes such as real-time updates, user feedback, brand monitoring, market research, digital marketing, and political analysis. The Twitter API (Application Programming Interface) provides developers with tools and functionalities to access and interact with Twitter data, including tweets, user profiles, and timelines, enabling a wide range of applications and services. However, Twitter sentiment analysis presents challenges such as handling sarcasm, irony, colloquial language, and coping with the sheer volume and rapid flow of Twitter data. Nevertheless, with effective preprocessing techniques and AI methods, Twitter sentiment analysis can yield valuable insights into public opinion on various topics.

Key Words: Twitter, AI, SVM, NLP

1 Introduction

Twitter feeling investigation examinations the opinion or feeling of tweets. It utilizes regular language handling and AI calculations to characterize tweets naturally as sure, negative, or impartial in view of their substance. It tends to be finished for individual tweets or a bigger dataset connected with a specific point or occasion. [4,5] A Twitter feeling investigation decides pessimistic, good, or nonpartisan feelings. Feeling examination or assessment mining alludes to recognizing as well as characterizing the opinions that are communicated in the text source. Tweets are in many cases helpful in creating an immense measure of opinion

* Corresponding author: teaching.usha@gmail.com
information upon examination. This information is valuable in grasping the assessment of individuals via web-based entertainment for various subjects.

2 Motivation

Twitter information opinion investigation is a famous strategy used to dissect the feeling of tweets or other text-put together happy imparted to respect to the stage. The inspiration for directing feeling investigation on a solitary informational index could be to acquire bits of knowledge into the sentiments and perspectives of Twitter clients on a specific point or occasion. For example, an organization might need to comprehend how individuals feel about their image, item, or administration in view of tweets referencing their image name. On the other hand, a political mission might need to check public feeling towards a specific strategy or competitor by investigating tweets connected with the point. Opinion investigation can likewise be valuable for following public responses to letting the cat out of the bag occasions, like catastrophic events, political emergencies, or major games. By dissecting the opinion of tweets connected with the occasion, analysts can acquire bits of knowledge into how individuals are feeling and answering progressively. By and large, the inspiration for leading feeling investigation on a solitary informational index is to acquire a more profound comprehension of individuals' mentalities, suppositions, and feelings towards a specific subject, brand, or occasion.

3 Literature Survey

A writing overview on feeling examination of a solitary dataset of Twitter information would include an exhaustive survey of existing exploration and concentrates on this subject. [1,2,3] The study would plan to recognize the different methodologies and strategies that have been utilized to perform feeling examination on a solitary dataset of Twitter information and their viability in accomplishing the goal. The writing review would cover subjects, for example, the preprocessing strategies used to clean and set up the information for opinion examination, the different AI calculations utilized to group the tweets into feeling classes, and the different pre-prepared models utilized for this errand.

It would likewise investigate the different assessment measurements used to gauge the precision of the feeling examination models and the difficulties related with dissecting Twitter information like mockery, incongruity, and logical language. [6, 7] A writing review on feeling examination of a solitary dataset of Twitter information would give a thorough comprehension of the different procedures and approaches utilized in this field and assist with recognizing the best strategies for performing opinion investigation on Twitter information connected with a particular goal.

4 Existing Work

There are a few existing works that have performed feeling examination on single datasets of Twitter information. Here are a few existing works: "Opinion Examination of Twitter Information for Anticipating Financial exchange Developments" by V. Pakkiraiah and K. Vinod Kumar: This study expected to anticipate the financial exchange developments utilizing feeling examination of Twitter information. The creators utilized a Help Vector Machines (SVM) calculation to characterize the tweets into positive, negative, or nonpartisan feeling classes. "A Relative Investigation of Feeling Examination Methods on Twitter Information" by S. Singh and R. Gupta: This study looked at the presentation of various feeling investigation strategies, including AI calculations like Gullible Bayes and SVM,
dictionary-based techniques like Senti Word Net and VADER. The creators assessed the exactness of these procedures on a dataset of tweets connected with the Indian general decisions. "Feeling Examination of Twitter Information Utilizing AI Methods" by P. R. P. V. R. K. Prasad and K. V. N. Sunitha: This study utilized a dataset of tweets connected with the Indian Head Association (IPL) cricket competition to perform feeling examination. The creators utilized the Irregular Woodland calculation to characterize the tweets into positive, negative, or nonpartisan opinion classes. "Dissecting Public Opinion on Corona virus through Twitter Information: A Feeling Investigation Study" by F. Rahman, A. Al Hasan, and M. M. Hossain: This study utilized opinion examination to investigate general society. By and large, feeling investigation of Twitter information is a famous exploration point, and there is an abundance of existing work that can educate the improvement regarding an opinion examination framework for a solitary dataset.

4.1 Limitations of existing work

A portion of the restrictions of existing work on opinion examination of Twitter information include:

1. Ambiguity in language: Twitter is known for its casual language and utilization of shoptalk, which can make it challenging for opinion examination models to group the feeling of a tweet precisely.

2. Contextual comprehension: Opinion examination models frequently battle with understanding the setting of a tweet, which can prompt misclassification of feeling. For instance, a tweet containing the expression "not terrible" may be named positive, even though the opinion is really impartial or negative.

3. Limited preparation information: Opinion investigation models require a lot of prepared information to get familiar with the subtleties of language and precisely group feeling. Be that as it may, preparing information for Twitter opinion investigation can be restricted and one-sided towards specific socioeconomics or themes.

4. Handling of mockery and incongruity: Twitter clients frequently use mockery and incongruity to communicate their opinion, which can be challenging for feeling investigation models to distinguish and arrange precisely.

5. Dynamic language use: Twitter language is continually advancing and changing, making it challenging for opinion examination models to stay aware of the most recent language patterns and shoptalk.

5 Proposed System Design

Proposed work for leading opinion examination on a solitary dataset of Twitter information could incorporate the accompanying advances: Issue plan: Characterize the issue explanation and examination question. For instance, "What is the feeling of Twitter clients towards a specific brand or item?" Information assortment: Gather the dataset of tweets connected with the exploration question. Utilize important catchphrases, hash tags, and search questions to assemble significant information. Information preprocessing: Clean the dataset by eliminating any immaterial data like URLs, usernames, and hash tags. Tokenize the leftover text into individual words and eliminate any stop words.

Information naming: Appoint marks to the dataset as indicated by the feeling classification (good, pessimistic, or impartial). Utilize manual naming or pre-prepared models for robotized marking.

Model choice and preparation: Select a fitting feeling examination model, for example, a Guileless Bays or Backing Vector Machine (SVM) model and train it on the marked dataset.
Model assessment: Assess the presentation of the model by contrasting its anticipated marks and the real names. Compute measurements like exactness, accuracy, review, and F1 score.

Model streamlining: Improve the model by changing hyper parameters and exploring different avenues regarding various calculations.

Representation and investigation: Picture the outcomes utilizing diagrams, outlines, or other perception instruments. Break down the opinion of the Twitter information and draw bits of knowledge. End and future work: Sum up the discoveries and make determinations. Recognize regions for future work and examination. By and large, the proposed work includes a deliberate way to deal with opinion examination on a solitary dataset of Twitter information, which incorporates information assortment, preprocessing, naming, model determination and preparing, assessment, improvement, representation, investigation, and end.

5.1 Architecture

![Fig. 1. Architecture of the Proposed System Design](image)

6 Experimental Results

In this research, we offer a thorough implementation of the unique sentiment analysis machine learning method we developed in order to improve the precision of sentiment categorization in natural language processing tasks. By using sophisticated deep learning algorithms, our implementation compares the accuracy of current sentiment analysis models to find the highest accuracy yielding model. We will outline the main steps of our implementation in this first section, giving readers an overview of the procedure that will be covered in more detail in later sections. Data pretreatment, feature engineering, model architecture, training procedures, and evaluation measures are all included in this. We intend to provide readers a clear grasp of our process and the format of this paper by outlining these crucial phases up front. This will allow readers to follow our work from conception to conclusion.
6.1 Step 1: Import the Necessary Dependencies

To embark on our journey towards building an effective sentiment analysis model, we first gather the essential Python libraries that form the cornerstone of our project. These libraries encompass a wide spectrum of capabilities, from data manipulation and visualization to natural language processing and machine learning.

1. NumPy and pandas: NumPy's numerical computing prowess and pandas' versatile data manipulation capabilities lay the groundwork for efficient data handling and preprocessing. Together, they enable us to prepare and structure our dataset for model training.

2. re (Regular Expressions): Regular expressions are indispensable for text preprocessing tasks such as cleaning, tokenization, and pattern matching. They help us extract meaningful information from unstructured text data.

3. matplotlib.pyplot and seaborn: These visualization libraries allow us to gain insights from our data through informative graphs and plots. They help us explore the characteristics of our dataset and visualize model performance.

4. wordcloud: The wordcloud library is a powerful tool for generating word clouds, which provide an intuitive representation of the most frequent words in our text data. This visualization aids in understanding the prominent sentiment-bearing words.

5. scikit-learn (sklearn): Scikit-learn is a comprehensive machine learning library that provides a wide array of tools for classification and evaluation. It's indispensable for training and evaluating our sentiment analysis model.

6. textblob: TextBlob simplifies text processing tasks, including sentiment analysis. Its ease of use allows us to quickly implement sentiment classification and assess its performance.

7. NLTK (Natural Language Toolkit): NLTK provides a vast collection of natural language processing tools and resources. It's particularly valuable for tasks such as tokenization, stemming, and stopword removal.

By importing and harnessing the capabilities of these libraries, we ensure a robust foundation for our sentiment analysis implementation. In the following sections, we will delve into the specific functions and methods offered by each of these libraries, demonstrating how they contribute to the success of our project.

```python
# utilities
import pandas as pd
import numpy as np
import re
# plotting
import seaborn as sns
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from matplotlib import style
style.use('ggplot')
from textblob import TextBlob
# nltk
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
# sklearn
```
6.2 Step 2: Read and Load the Dataset

In the second phase of our implementation, we concentrate on obtaining the dataset, which is one of the key components of our sentiment analysis project. In order to start, we begin by sourcing a comprehensive dataset that aligns with our sentiment classification task. The dataset consists of 11020 elements, 0 to 11019, making up our chosen dataset. There are 16 columns in the data.

The data's 11020 row count, 11020 column length, and shape of dataset is (11020, 16) are critical for developing and testing our sentiment analysis model's performance. The user_name, user_location, user_description, user_created, user_followers, user_friends, user_favorites, user_verified, date, text, hashtags, source, favorites, and is_retweet columns are all meticulously curated into the dataset. The robustness of our model depends critically on the quality and representativeness of our dataset.

6.3 Step 3: Exploratory Data Analysis

In this pivotal step, we embark on an insightful journey into our dataset, illuminating its characteristics and laying the groundwork for informed decisions in subsequent stages. We begin by examining the top five records of our data to gain an initial understanding of its structure and content. Next, we retrieve vital information about the dataset, including data types, unique values, and statistical summaries. Determining the length and shape of the dataset provides essential context regarding its size and dimensions.

As part of data quality assessment, we meticulously check for null values, ensuring that our dataset is free from missing information that could affect the analysis. We also identify the columns or features present in the data, each of which plays a distinct role in our sentiment analysis task. To focus our analysis on the textual aspect of the data, we create a separate data frame containing only the text data column. This specialized data frame becomes the focal point for text preprocessing and sentiment classification.

Throughout this exploratory data analysis, we extract valuable insights that inform our understanding of the dataset's intricacies, setting the stage for subsequent steps where we harness these insights to develop an effective sentiment analysis model.

6.4 Step 4: Data Preprocessing

In this crucial phase, we refine and prepare our textual data for sentiment analysis by implementing a series of preprocessing steps: 1. Apply Preprocessing on Text Data: We begin by applying text preprocessing techniques, including but not limited to lowercasing, removal of special characters and punctuation, and tokenization. These steps enhance the consistency and cleanliness of our text data.

```python
def data_processing(text):
    text = text.lower()
    text = re.sub(r"https\S+|www\S+https\S+", '', text, flags=re.MULTILINE)
    text = re.sub(r'\@w+|\#', '', text)
    text = re.sub(r'[^\w\s]', '', text)
```
2. Remove Duplicate Data: Duplicate data can skew our analysis and model training. We identify and eliminate duplicate records, ensuring that each piece of text is unique within our dataset.

```python
text_df = text_df.drop_duplicates('text')
```

3. Data Stemming: Stemming is a process that reduces words to their root form (e.g., "running" to "run"). This helps in consolidating variations of words and simplifies feature extraction.

```python
stemmer = PorterStemmer()
def stemming(data):
    text = [stemmer.stem(word) for word in data]
    return text

text_df['text'] = text_df['text'].apply(lambda x: stemming(x))
```

4. After these preprocessing steps, we gain insights into the modified dataset:

- Data Information after Removing Duplicate Data: We provide updated statistics on the dataset, including data types, unique values, and summary statistics. This information reflects the dataset's characteristics post-duplicate removal.

```python
text_df.info()
```

- Polarity Calculation: We calculate the polarity of each text, quantifying the sentiment orientation as positive, negative, or neutral. This polarity score serves as a foundation for sentiment analysis.

```python
def polarity(text):
    return TextBlob(text).sentiment.polarity

text_df['polarity'] = text_df['text'].apply(polarity)
text_df.head(10)
```

- Sentiment Calculation: Building on polarity, we further classify the sentiment of each text, categorizing them as positive, negative, or neutral based on predefined thresholds. This categorization enables us to analyze the overall sentiment distribution within the dataset.

```python
def sentiment(label):
    if label <0:
        return "Negative"
    elif label ==0:
        return "Neutral"
    elif label>0:
        return "Positive"

text_df['sentiment'] = text_df['polarity'].apply(sentiment)
text_df.head()
```

Through meticulous data preprocessing, we transform raw text into a refined, structured, and sentiment-aware dataset. This step is instrumental in ensuring the quality and consistency of our data, setting the stage for the development of a robust sentiment analysis model.
6.5 Step 5: Distributed Data Visualization

In this step, we harness the power of distributed data visualization tools to gain deeper insights into our sentiment analysis dataset. Leveraging libraries such as seaborn (sns), we create a variety of informative plots and visualizations to uncover patterns, trends, and sentiment distribution within our data. We utilize traditional plots, such as histograms and bar charts, to depict sentiment distribution, and employ word clouds to visually emphasize the most frequently occurring words, providing a holistic view of the textual aspects of our dataset. These visualizations serve as a vital bridge between data exploration and model development, enabling us to make informed decisions and refine our sentiment analysis approach.

![Fig. 2. Countplot of Sentiment Distribution](image1)

![Fig. 3. Distribution of Sentiments](image2)
6.6 Step 6: Transforming the Dataset Using Count Vectorizer

In this step, we employ the Count Vectorizer technique to transform our text data into numerical features, making it suitable for machine learning. Count Vectorization converts text into a matrix of word counts, representing each document as a vector in a high-dimensional space. This enables us to quantify the frequency of words in each text, a crucial step for training a sentiment analysis model. Simultaneously, we split our dataset into distinct train and test subsets, allocating data for model training and evaluation. This partitioning ensures that our model's performance is rigorously assessed on unseen data, allowing us to gauge its generalization capabilities accurately.

```python
vectorizer = CountVectorizer()
vectorizer.fit(text_df)
X = vectorizer.fit_transform(text_df)
```

6.7 Step 7: Model Building

In this critical phase, we embark on the process of constructing our sentiment analysis models. To comprehensively address the problem statement, we employ two distinct models: Logistic Regression and Support Vector Classification (SVM). Our strategy is to explore a spectrum of classifiers, from relatively simple to more complex models, to discern their effectiveness in handling our dataset.

Logistic Regression serves as a foundational model, leveraging a linear approach to capture relationships within the data. Its simplicity makes it an ideal starting point for understanding the baseline performance of our sentiment analysis task. This initial model aids in understanding the dataset's characteristics and establishing a foundation for subsequent improvements.

```python
logreg = LogisticRegression()
logreg.fit(x_train, y_train)
logreg_pred = logreg.predict(x_test)
logreg_acc = accuracy_score(logreg_pred, y_test)
```
On the other hand, Support Vector Classification (SVM), a powerful algorithm based on the principles of support vector machines, introduces a more intricate and non-linear perspective. By incorporating SVM, we aim to uncover nuanced patterns and relationships within the data that may be overlooked by simpler models.

SVCmodel = LinearSVC()
SVCmodel.fit(x_train, y_train)
svc_pred = SVCmodel.predict(x_test)
svc_acc = accuracy_score(svc_pred, y_test)
print("test accuracy: {:.2f}%".format(svc_acc*100))

6.7.1 Hyperparameter Tuning in Model Building

In the pursuit of enhancing our models' performance, we engage in the crucial practice of hyperparameter tuning during Step 7. Hyperparameters are configurations that govern the learning process of our models, influencing their ability to capture intricate patterns in the data. Through systematic experimentation and optimization, we fine-tune these hyperparameters, seeking the optimal combination that maximizes our models' predictive accuracy. This process not only refines the performance of Logistic Regression and Support Vector Classification but also ensures that our sentiment analysis models are finely calibrated to the unique nuances of the textual data. Hyperparameter tuning stands as a pivotal step in our quest for an optimal and well-generalized sentiment analysis solution.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>84.64%</td>
</tr>
<tr>
<td>Tuned Logistic Regression</td>
<td>85.92%</td>
</tr>
<tr>
<td>SVC</td>
<td>87.34%</td>
</tr>
<tr>
<td>Tuned SVC</td>
<td>87.58%</td>
</tr>
</tbody>
</table>

Fig. 5. Model Accuracy Comparison on Dataset: SVC Outperforms Logistic Regression with Tuning

Our overarching goal is to identify the model that excels in sentiment analysis on our dataset. By systematically comparing and evaluating the performance of these two models, we pave the way for data-driven decisions regarding the most effective approach. This critical step guides us towards achieving optimal sentiment classification results.

7 Conclusion

The performance evaluation of various models on the dataset reveals distinct accuracy rates. The baseline Logistic Regression model achieves a commendable accuracy of 84.64%, demonstrating its efficacy in capturing patterns within the sentiment data. After hyperparameter tuning, the accuracy of Logistic Regression further improves to 85.92%, showcasing the significance of parameter optimization.
The Support Vector Classification (SVC) model emerges as a robust performer, achieving an accuracy of 87.34%. This underscores the model's capability to discern intricate patterns in the sentiment-laden text. Through hyperparameter tuning, the SVC model achieves a slight but notable enhancement, reaching an accuracy of 87.58%.

These results suggest that both Logistic Regression and SVC, especially after hyperparameter tuning, are viable candidates for sentiment analysis on this dataset. The marginal difference in accuracy between the tuned models indicates the sensitivity of these classifiers to parameter adjustments. The choice between Logistic Regression and SVC may depend on considerations of computational efficiency and interpretability, with each model offering a nuanced trade-off.

In navigating the landscape of sentiment analysis, these findings guide the selection of an appropriate model tailored to the specific nuances of the dataset. Further fine-tuning and exploration of alternative models could provide avenues for even greater accuracy, offering opportunities for continued optimization in sentiment analysis tasks.

References

1. Sentiment140: This dataset contains 1.6 million tweets that have been manually labeled as positive or negative. The dataset can be downloaded from here: http://help.sentiment140.com/for-students/
2. Twitter Sentiment Analysis Dataset: This dataset contains 1,578,627 tweets that have been labeled as positive, negative, or neutral. The dataset can be downloaded from here: https://www.kaggle.com/kazanova/sentiment140
3. SemEval-2017 Task 4: This dataset contains 10,000 tweets that have been labeled as positive, negative, or neutral. The dataset can be downloaded from here: https://competitions.codalab.org/competitions/16380#learn_the_details
4. Stanford Sentiment Treebank: This dataset contains 215,154 phrases that have been labeled as positive or negative. The dataset can be downloaded from here: https://nlp.stanford.edu/sentiment/treebank.html
5. Kaggle Twitter US Airline Sentiment: This dataset contains tweets about US airlines, labeled as positive, negative or neutral. The dataset can be downloaded from: https://www.kaggle.com/crowdflower/twitter-airline-sentiment.