ARBAMINCH UNIVERSITY FACULTY OF COMPUTING AND SOFTWARE ENGINEERING

Course: Big Data Analytics & Business Intelligence

Group F Assignment (Social media sentiment)

WEEK 4 & 5

Name	Id
Abigiya Solomon ---------	NSR/458/14
Betel Tarekegn ----------	NSR/3010/14
Elsa Husen ----------	NSR/852/13
Meklit Mathewos ----------	NSR/2460/14
Solomon Aragaw ----------	NSR/1998/14

Contents

Introduction

Social media platforms generate massive amounts of user-generated content daily, containing valuable insights into public opinion, trends, and customer sentiment. However, analyzing social media text poses significant challenges due to its short, informal, and noisy nature. Users often employ slang, abbreviations, emojis, and inconsistent grammar, making it difficult for traditional methods to capture sentiment accurately.

Machine learning provides a powerful approach to automatically analyze and classify social media text based on sentiment. By converting text into numerical representations through feature engineering techniques such as TF-IDF and applying robust classification algorithms, models can learn patterns that differentiate positive, negative, and neutral posts.

This project focuses on building a comprehensive pipeline for social media sentiment analysis. Three machine learning algorithms - Logistic Regression, Multinomial Naive Bayes, and Linear Support Vector Machine (SVM) were selected for their efficiency, reliability, and suitability for high-dimensional text data. The pipeline integrates data preprocessing, feature extraction, model training, evaluation, comparison, and storage of artifacts, ensuring reproducibility, scalability, and deployment readiness for real-world applications.

Select ML algorithms

Three machine learning algorithms were selected to perform sentiment analysis on social media text data. Social media data is usually short, informal, noisy, and high-dimensional after text vectorization. Therefore, the selected models needed to be efficient, reliable, and well-suited for text classification tasks. The three chosen models are Logistic Regression, Multinomial Naive Bayes, and Linear Support Vector Machine (SVM).

Logistic Regression

Logistic Regression is one of the most widely used algorithms for text classification and sentiment analysis. Although its name includes the word "regression," it is actually a classification algorithm. It works by learning the relationship between input features (TF-IDF features extracted from social media text) and the sentiment labels.

Logistic Regression is effective because it handles high-dimensional sparse data very well. Text data converted into TF-IDF (Term Frequency – Inverse Document Frequency) vectors often contains thousands of features, and Logistic Regression can process these efficiently. The model estimates the probability that a given text belongs to a particular sentiment class (positive, negative, or neutral) and assigns the class with the highest probability. The influence of individual words on sentiment prediction can be understood by examining model weights. This makes it useful for analyzing which terms contribute most to positive or negative sentiment in social media posts.

Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic machine learning model based on Bayes' Theorem. It is especially popular for text classification tasks such as spam detection and sentiment analysis. The model assumes that the presence of one word in a document is independent of the presence of other words, which is known as the "naive" assumption. Multinomial Naive Bayes performs well because it works naturally with word frequency and TF-IDF features. It is fast to train, requires less computational power, and performs efficiently even on large datasets. This makes it suitable for big data environments and real-time sentiment analysis.

One of the key strengths of Naive Bayes is its simplicity and speed. It can quickly learn patterns in text data and produce reliable results with minimal tuning. For social media data, where posts are short and vocabulary can be large, this efficiency is a major advantage.

Linear Support Vector Machine (SVM)

Linear Support Vector Machine is a powerful classification algorithm that aims to find the best decision boundary between different sentiment classes. It works by maximizing the margin between data points of different classes, which often leads to strong generalization performance. Linear SVM is highly effective because it handles high-dimensional text features extremely well. When text is transformed using TF-IDF, the feature space becomes very large, and Linear SVM is designed to work efficiently in such environments. one of the major advantages of Linear SVM is its ability to reduce overfitting. By focusing on the most important data points (support vectors), the model becomes robust to noise, which is very common in social media text such as slang, abbreviations, and misspellings.

Feature engineering

Feature engineering is a crucial step in any machine learning pipeline, especially when dealing with text data from social media platforms. Social media text is typically noisy, informal, and inconsistent, containing slang, abbreviations, hashtags, emojis, and misspellings. Without proper preprocessing, these issues can negatively impact model performance. Feature engineering converts raw text into a structured format suitable for machine learning algorithms and improves the quality of predictions.

Data Cleaning

The first step in feature engineering is data cleaning. This involves several tasks to prepare raw social media text for analysis:

1. Lowercasing: All text is converted to lowercase to ensure that words like “Happy” and “happy” are treated as the same token. This reduces redundancy and improves model consistency.

2. Removing Non-Alphabetic Characters: Punctuation, numbers, special symbols, and emojis are removed, leaving only meaningful words. This step helps to reduce noise in the dataset while preserving the essential content.

3. Stopword Removal: Commonly used words such as “the,” “is,” “and,” and “a” do not carry significant sentiment information. These stopwords are removed from the text to focus on words that contribute to sentiment. Eliminating stopwords reduces feature dimensionality and improves the relevance of extracted features.

Feature Transformation: TF-IDF

Once the text is cleaned, the next step is to convert it into numerical features that machine learning algorithms can process. One of the most effective techniques for text data is Term Frequency-Inverse Document Frequency (TF-IDF).

TF-IDF captures how important a word is in a document relative to the entire corpus. Words that appear frequently in a single document but rarely in other documents are given higher importance. This helps the model focus on terms that are more likely to indicate sentiment rather than common words that appear across all texts.

- Positive words like “love,” “great,” or “amazing” gain higher scores in posts with positive sentiment.
- Negative words like “hate,” “bad,” or “terrible” are emphasized in negative posts.
- Neutral or common words are automatically downweighted, reducing their influence on predictions.

Sentiment Label Generation

The dataset requires sentiment labels. In this project, sentiment is determined based on the polarity of the cleaned text using natural language processing tools. Each post is classified into one of three categories:

- Positive sentiment: The text expresses favorable opinions or emotions.

- Negative sentiment: The text expresses discontent or unfavorable opinions.
- Neutral sentiment: The text neither strongly conveys positive nor negative emotion.

Storing Preprocessing Tools

To ensure reproducibility and consistency across training, testing, and deployment stages, the TF-IDF vectorizer used to transform text into numerical features is stored as a reusable object. This allows new social media posts to be converted using the same feature mapping, ensuring that models make predictions based on the same learned feature space.

Training Initial Models

After completing feature engineering and generating TF-IDF vectors from social media text, the next crucial step in the machine learning pipeline is training the models. Model training is the process by which algorithms learn patterns from the labeled dataset, allowing them to predict sentiment for unseen posts.

Preparing Data for Training

Before training the dataset is divided into two subsets:

1. Training Set: This portion of the data is used by the models to learn the relationship between input features (TF-IDF) and sentiment labels. The models examine patterns in word usage and how specific words are associated with positive, negative, or neutral sentiment.
2. Testing Set: A separate portion of the data is reserved for later evaluation. This ensures that the models are tested on unseen data, allowing assessment of their generalization ability. Typically, an 80/20 split is used, with 80% of the data for training and 20% for testing.

Training Logistic Regression

Logistic Regression is trained on the TF-IDF features of the training set. During training, the model estimates weights for each word feature to maximize the probability that the text belongs to the correct sentiment class. Logistic Regression can efficiently handle thousands of features created by the TF-IDF transformation. The training process adjusts model parameters iteratively,

ensuring that posts with positive, negative, or neutral sentiment are classified as accurately as possible. The result is a model capable of predicting sentiment probabilities for new posts.

Training Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic model that uses word frequencies to estimate the likelihood of each sentiment class. During training, the model calculates the probability of each word occurring in positive, negative, or neutral posts.

This approach is particularly well-suited for social media text, where some words strongly indicate sentiment while others are common across all posts. Multinomial Naive Bayes quickly learns these word-sentiment associations, resulting in a model that can efficiently predict sentiment for both small and large datasets. Its simplicity and speed make it an excellent choice for initial model development.

Training Linear Support Vector Machine (SVM)

Linear SVM takes a different approach by finding the optimal boundary (hyperplane) that separates different sentiment classes in the high-dimensional feature space. Using TF-IDF vectors, each post becomes a point in this space, and the model seeks the hyperplane that maximizes the margin between classes.

This training method allows Linear SVM to handle high-dimensional data effectively and reduces the risk of overfitting. The model learns which features (words) are most significant for distinguishing between positive, negative, and neutral sentiment. Although training Linear SVM can require more computational resources than Logistic Regression or Naive Bayes, it often achieves strong performance on text classification tasks.

Model Storage for Future Use

After training, the models are saved as reusable objects. Storing trained models allows them to be loaded later without retraining, making the pipeline efficient and reproducible. This is especially important for social media applications, where new posts are continuously generated and predictions need to be made in real time.

Each model - Logistic Regression, Multinomial Naive Bayes, and Linear SVM—is stored in a structured directory for easy access. This ensures that the machine learning pipeline can be executed seamlessly during deployment, with consistent preprocessing, feature extraction, and prediction steps.

Importance of Training Multiple Models

Training multiple models provides several benefits for social media sentiment analysis:

- Comparison: By training different algorithms, we can later compare performance and choose the most suitable model for the final pipeline.
- Robustness: Different models capture different patterns in the data, increasing the chance of finding an accurate solution.
- Efficiency: Some models, like Naive Bayes, are extremely fast, while others, like SVM, may provide higher accuracy. Having multiple trained models allows flexibility in balancing speed and performance.

Evaluating Performance

Proper evaluation is crucial to ensure that the models are reliable, accurate, and suitable for real-world social media applications.

Preparing for Evaluation

Before evaluation the dataset is divided into training and testing sets. The testing set consists of posts that the models have not seen during training. Using unseen data allows for a fair assessment of the model's generalization ability its capacity to make correct predictions on new social media content.

The TF-IDF features created during feature engineering are applied to the testing data to maintain consistency. The trained models - Logistic Regression, Multinomial Naive Bayes, and Linear SVM—are then used to predict sentiment labels for each post in the testing set.

Compare algorithms

By evaluating multiple models on the same testing set, we can determine which model performs best overall and for each sentiment category:

- Logistic Regression offers a balance between interpretability and performance.
- Multinomial Naive Bayes is often very fast and performs well with sparse text data.
- Linear SVM typically provides strong classification boundaries in high-dimensional feature space, though it may require more computational resources.

Comparing Machine Learning Models

Comparing models allows us to understand the strengths and weaknesses of each approach and to select the most appropriate model for deployment in real-world applications.

Purpose of Model Comparison

Social media text data is diverse and challenging. Users often express opinions in informal language, including slang, abbreviations, emojis, and sarcasm. Therefore, different machine learning algorithms may perform differently depending on how they handle high-dimensional TF-IDF features and subtle sentiment cues. Comparing models helps to:

- Identify which model achieves the highest overall accuracy.
- Determine which model performs best for specific sentiment classes (positive, negative, neutral).
- Understand the trade-offs between simplicity, speed, and predictive performance.
- Decide which model is best suited for deployment and scalability.

Insights from Model Comparison

1. Logistic Regression

    ✓ Provides consistent performance across all sentiment classes.

- ✓ Balances simplicity with interpretability, allowing understanding of which words influence sentiment predictions.
- ✓ May slightly underperform on more nuanced or highly imbalanced data.

2. Multinomial Naive Bayes
   - ✓ Extremely fast to train and predict, making it suitable for large-scale social media datasets.
   - ✓ Performs well when word frequencies strongly correlate with sentiment.
   - ✓ Its simplicity sometimes limits performance on posts with complex or subtle sentiment.
3. Linear Support Vector Machine (SVM)
   - ✓ Often achieves the highest accuracy and F1-scores due to its ability to separate classes with a clear margin.
   - ✓ Robust to high-dimensional TF-IDF features common in social media text.
   - ✓ Requires more computational resources and longer training time compared to simpler models.

Visualization of Comparison

Comparison is often visualized with:

- Bar charts showing accuracy and weighted F1-score for each model.
- Side-by-side confusion matrices for each sentiment class.

These visualizations make it easier to quickly understand which model performs best overall and for each type of sentiment.

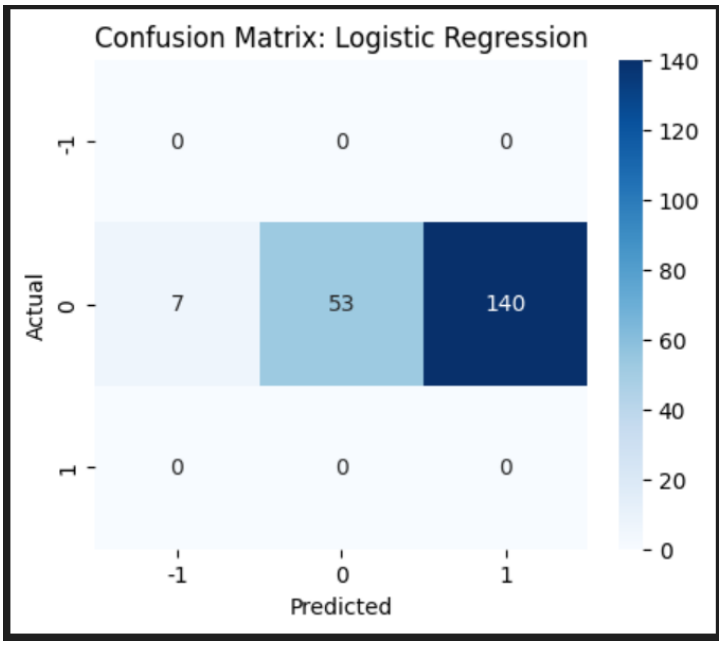

Figure 1 Logistic Regression Evaluation

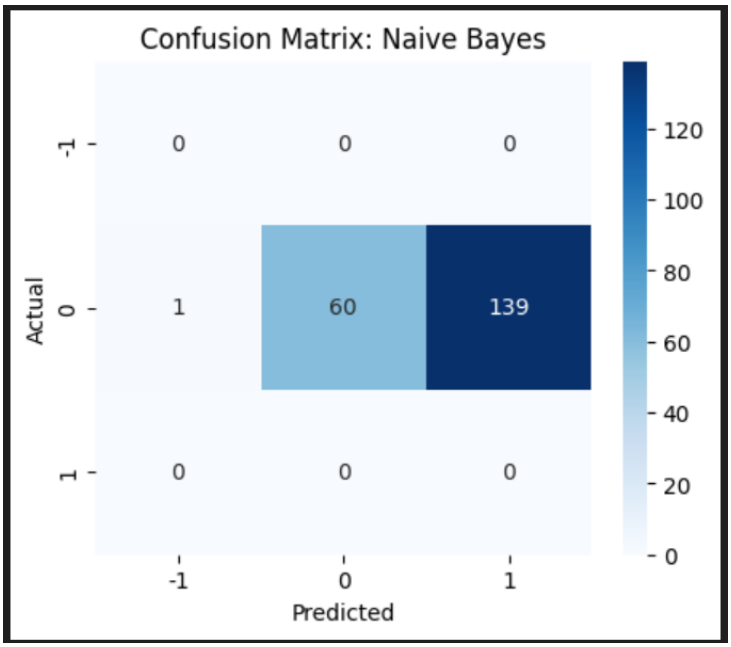

Figure 2 Naive Bayes Evaluation

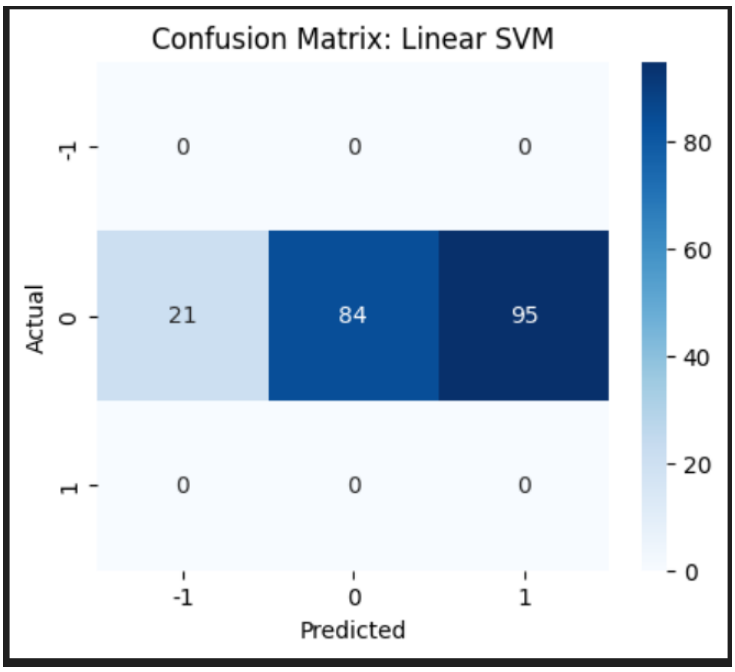

Figure 3 Linear SVM Evaluation

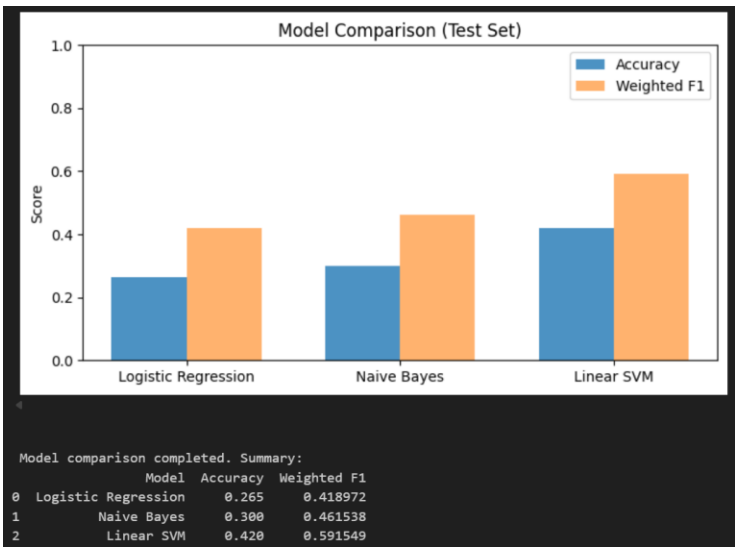

Figure 4 Comparison

Finalizing the Machine Learning Pipeline

The pipeline is a structured workflow that ensures the entire process - from raw social media text to sentiment prediction - is reproducible, efficient, and deployable.

Organizing Data and Artifacts

The finalized pipeline begins by organizing all input data and output artifacts. The cleaned social media dataset, along with computed sentiment labels, is stored in a centralized location, such as a distributed file system like HDFS. This ensures that all subsequent steps can access the same data consistently.

Saving intermediate datasets and features is important because:

- ✓ It allows the pipeline to be reused for new data without repeating initial processing steps.
- ✓ It improves reproducibility by maintaining a snapshot of the cleaned and preprocessed dataset.
- ✓ It facilitates collaboration, allowing multiple team members or systems to access the same structured data.

Feature Engineering Integration

Feature engineering is integrated into the pipeline to transform raw social media posts into numerical features suitable for machine learning. This involves:

- Text cleaning: Lowercasing, removing non-alphabetic characters, and eliminating stopwords.
- Vectorization: Converting cleaned text into TF-IDF features, which capture the importance of each word relative to the corpus.

The TF-IDF vectorizer is saved as a reusable object, ensuring that new posts can be transformed consistently using the same feature space. This guarantees that predictions on new social media data are accurate and aligned with the training process.

Model Training within the Pipeline

The finalized pipeline includes the training of multiple machine learning models. In this case:

- Logistic Regression: Efficient and interpretable, suitable for general sentiment classification.

- Multinomial Naive Bayes: Fast and effective for word frequency-based classification.
- Linear SVM: Robust for high-dimensional TF-IDF features and capable of handling complex decision boundaries.

Each model is trained on the processed dataset and then saved as an artifact. Storing models ensures that they can be reloaded for evaluation, prediction, or deployment without retraining.

Centralized Storage of Artifacts

All critical artifacts - processed datasets, TF-IDF vectorizers, and trained models - are saved in a centralized folder structure, typically in HDFS or a local storage system. Centralization ensures:

- Ease of deployment: Models and vectorizers are ready for integration into applications.
- Reproducibility: The same pipeline can be rerun with consistent results.
- Maintainability: Artifacts can be versioned and updated without affecting other parts of the pipeline.

Benefits of a Finalized Pipeline

1. Reproducibility: Every step, from preprocessing to model training, can be executed consistently.
2. Scalability: The pipeline can handle increasing volumes of social media data without manual intervention.
3. Efficiency: Automated storage and reuse of intermediate artifacts reduce computation time.
4. Deployment-Readiness: Trained models and vectorizers are ready to be deployed for real-time sentiment analysis.

Conclusion

The development of a finalized machine learning pipeline for social media sentiment analysis enables systematic, reproducible, and efficient processing of large-scale textual data. By carefully integrating data cleaning, TF-IDF feature extraction, and the training of multiple models, the pipeline ensures that sentiment predictions are accurate and reliable. Comparing the performance of Logistic Regression, Multinomial Naive Bayes, and Linear SVM highlighted the strengths and limitations of each algorithm, providing insights into their suitability for different scenarios. Logistic Regression offers interpretability and balanced performance, Naive Bayes delivers speed and efficiency, and Linear SVM achieves strong accuracy in high-dimensional feature spaces. Centralized storage of processed datasets, feature extraction tools, and trained models further enhances the pipeline's robustness and facilitates future deployment. Overall, this pipeline serves as a scalable and effective framework for monitoring social media sentiment, supporting data-driven decision-making, and enabling real-time analysis of public opinion in dynamic digital environments.