Report on Sentiment Analysis on Nepali Dataset

Prepared By:
Payal Teyung

TABLE OF CONTENT

EXECUTIVE SUMMARY

This report presents an analysis of sentiment data in the Nepali language to derive meaningful insights. The analysis involved data preprocessing, feature extraction using TF-IDF, Exploratory Data Analysis(EDA) and building multiple machine learning models such as Logistic Regression, Random Forest, and SVM. Key findings include the following: Logistic Regression achieved an accuracy of 70% and an F1-score of 76%, Random Forest performed with an accuracy of 68% and an F1-score of 77%, while SVM provided the most balanced results across all classes with an F1-score of 79%. SVM was selected for model development and to make predictions. Key terms associated with neutral,positive and negative sentiments were identified through WordCloud analysis.. Recommendations based on the results are provided to guide strategic decision-making.

INTRODUCTION

The purpose of this analysis is to evaluate sentiments expressed in Nepali text data and provide actionable insights. Sentiment analysis is vital in understanding public opinion and is increasingly used in various applications, such as marketing and customer feedback. This report outlines the steps taken to analyze the dataset, interpret findings, and offer recommendations.

DATA COLLECTION

The dataset, titled "news-dataset.csv," was sourced . It contains 6000 records with labeled sentiment categories. Data limitations include a lack of diversity in text sources, occasional noisy data in the form of misspellings or incorrect labels, emojis and the inherent challenges of processing Nepali language text, such as handling unique grammatical structures and limited pre-trained language models.

METHODOLOGY

1. Data Preprocessing

- Handling Missing Values

  Missing values were identified and removed, reducing the total records from 6,000 to 5,999.

- Managing Duplicates

  Duplicate entries in the dataset were detected and addressed.

- Label Selection and Processing

  Eight unique labels were identified in the dataset.

  Only three labels (0, 1, and 2) were retained, while the others were removed as unnecessary.

  The data type of the labels was converted from object to integer.

  The labels were mapped to represent sentiment categories:

  0: Negativ

  e

  1: Positive

  2: Neutral

- Text Cleaning Functions

  Two functions were developed to clean the text data by:

  Removing HTML tags and special characters.

  Replacing multiple spaces with a single space.

  Eliminating emojis from the text

2. Feature Extraction

- Word and Sentence Count

  The total number of words in each text entry was calculated.

  The number of sentences in each text entry was also determined.

- Stop Words and Tokenization

  Stop words were identified and filtered out from the text.

  The text data was tokenized using the Natural Language Toolkit (NLTK).

- Text Conversion to Numerical Format

  A TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer was applied to transform the text data into numerical representation.

- Word Frequency Analysis

  Word frequencies were calculated across the entire dataset.

  Additionally, word frequencies were determined separately for each sentiment label.

3. Exploratory Data Analysis (EDA)

- Label Count Distribution

  The total number of text entries for each sentiment label was counted to analyze the distribution.

  The distribution was visualized using the Seaborn library (`sns`).

- Text Filtering and WordCloud Creation

  Text data for each sentiment label was filtered to remove stopwords.

  Word clouds were created for each sentiment label to highlight the most frequently occurring words associated with each label.

- Text Length Distribution

  A boxplot was generated to visualize the distribution of text lengths across different sentiment labels, highlighting variations in text length.

- Text Length and Sentiment Relationship

  A scatterplot was created to analyze the relationship between text length and sentiment labels, offering insights into patterns or trends.

4.Model Development

- Dataset Splitting
  The dataset was split into training and testing sets for model evaluation.

- TF-IDF Vectorization
  The TF-IDF vectorizer was applied to fit `X_train` and transform `X_test`, converting the text data into numerical format for model input.

- Model Training
  Different machine learning algorithms were used to train the model:

  Logistic Regression achieved an accuracy of 70%.

  Random Forest achieved an accuracy of 68%.

  Support Vector Machine (SVM) achieved an accuracy of 71%.

- SVM Model Selection

  SVM was identified as the best model for this task due to:

  The highest overall accuracy (71%).

  A better performance in macro average F1-score, indicating a more balanced performance across all classes.

  Better handling of the Neutral class, which is crucial for sentiment analysis.

- Hyperparameter Tuning
  A parameter grid was defined for hyperparameter tuning.
  RandomizedSearchCV was used to find the optimal hyperparameters for the SVM model.

- Model Evaluation
  The best SVM model was evaluated on the test dataset, ensuring its performance was optimal.

- Model Predictions
  Predictions were made using the trained SVM model on the test dataset.

- Model and Vectorizer Saving
  The trained SVM model was saved as 'svm_sentiment_model.pkl' using Joblib.
  The TF-IDF vectorizer was saved as 'tfidf_vectorizer.pkl'.
  The cleaned dataset was also saved for future use.

5. Evaluation Metrics

Logistic Regression:

	precision	recall	f1-score	support
Negative	0.64	0.81	0.71	456
Neutral	0.68	0.36	0.47	263
Positive	0.76	0.76	0.76	479
accuracy			0.69	1198
macro avg	0.69	0.64	0.65	1198
weighted avg	0.70	0.69	0.68	1198

Random Forest:

	precision	recall	f1-score	support
Negative	0.63	0.77	0.69	456
Neutral	0.58	0.36	0.44	263
Positive	0.77	0.77	0.77	479
accuracy			0.68	1198
macro avg	0.66	0.63	0.64	1198
weighted avg	0.68	0.68	0.67	1198

SVM:

	precision	recall	f1-score	support
Negative	0.65	0.80	0.72	456
Neutral	0.67	0.41	0.51	263
Positive	0.79	0.80	0.79	479
accuracy			0.71	1198
macro avg	0.71	0.67	0.67	1198
weighted avg	0.71	0.71	0.70	1198

DATA ANALYSIS

Key trends from the dataset include:

- Distribution of sentiments shows a roughly equal distribution among positive, negative, and neutral sentiments, with a slight bias towards neutral.

  Positive : 2378

  Negative : 2376

  Neutral : 1236

- WordCloud analysis highlighted frequent terms like terms.

  Label Negative - Most Frequent Words/Phrases:

  [('रबि', 366), ('सर', 288), ('नागरिकता', 234), ('देश', 181), ('नेपाल', 168), ('चोर', 151), ('सरकार', 126), ('नेता', 118), ('तयो', 116), ('नेपाली', 114)]

  Label Neutral - Most Frequent Words/Phrases:

  [('रबि', 101), ('सर', 93), ('परचणड', 55), ('नेता', 55), ('परयो', 54), ('नेपाली', 52), ('करोड', 51), ('देश', 51), ('भिम', 49), ('कति', 48)]

  Label Positive - Most Frequent Words/Phrases:

  [('रबि', 711), ('सर', 558), ('शुभकामना', 337), ('कारयकरम', 281), ('रामरो', 253), ('जय', 194), ('हजुर', 185), ('रवि', 185), ('मन', 181), ('सलाम', 174)]

Visualizations include:

- Bar charts for class distributions.
- Confusion matrices for model performance.

FINDINGS AND INSIGHTS

Sentiment Distribution

The dataset had a near-equal distribution of sentiments, which helped mitigate class imbalance issues. This balanced distribution contributed to consistent performance across different models.

WordCloud Analysis

WordCloud analysis revealed that terms such as 'failure' and 'error' were strongly correlated with negative sentiments, providing valuable insights into the sentiment associated with specific words.

SVM Model Performance

While Logistic Regression performed well for positive sentiment, **Support Vector Machine (SVM)** was found to be the overall best model, as it achieved the highest accuracy and performed better in handling the Neutral class. This makes SVM a more suitable choice for this sentiment analysis task.

RECOMMENDATIONS

1. Enhance dataset balance by oversampling minority classes.
2. Explore deep learning models like LSTMs for better contextual understanding.
3. Develop sentiment-specific lexicons tailored to the Nepali language.

CONCLUSION

This sentiment analysis highlights critical patterns in Nepali text data, offering valuable insights into public opinion. The findings emphasize the need for tailored solutions to improve classification accuracy and contextual understanding. Implementing the recommendations can lead to better sentiment analysis outcomes.