PERFORMANCE COMPARISON OF NLP LIBRARIES FOR THE PROBLEM OF FRAUD MESSAGE DETECTION

Project Report Submitted

To

Gujarat University

M.Sc. (Artificial Intelligence & Machine Learning)

SEMESTER-I

GUIDED BY: SUBMITTED BY:

Dr. Suchit Purohit Anju Jha(04)

Priyanshi Nagar(DS-04)

Dipali Pandya(DS-05)



**DEPARTMENT OF COMPUTER SCIENCE**

**GUJARAT UNIVERSITY, AHMEDABAD**

**YEAR: 2023-24**

**ACKNOWLEDGEMENT**

We are grateful for the help and direction we received in finishing our study on the Performance Comparison of NLP Libraries for the Problem of Fraud Message Detection from our principal, Dr. Jyoti Parikh, as well as our head professor, Dr. Suchit Purohit. It was an excellent opportunity to learn.

We are extremely appreciative of the efforts made by Our Mentor, Prof. Sidhanth, Vaishnavi Ma'am, Sanskruti Ma'am, and Our Seniors (AIML) in the computer science department at Gujarat University to the completion of our project titled Performance Comparison of NLP Libraries for the problem of fraud message detection.

We would like to take this opportunity to express Our gratitude to all group members. The project would not have been successful without their cooperation and inputs.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr No. | Title | Pg No. |
| 1. | Abstract | 4 |
| 2. | Introduction | 5 |
| 3. | Problem Statement | 7 |
| 4. | Tools, Technologies and Principles | 8 |
| 5. | Methodology | 10 |
| 6. | Data set | 12 |
| 7. | Implementation Details | 13 |
| 8. | Results and Analysis | 17 |
| 9. | Conclusion | 18 |
| 10. | Future | 19 |
| 11. | References | 20 |

**ABSTRACT**

This research project aims to increase the effectiveness of fraudulent message detection by comprehensively comparing the performance of different Natural Language Processing (NLP) libraries. The proposed solution integrates NLP techniques with machine learning (ML) algorithms to combat SMS fraud. In this study, various NLP libraries will be evaluated for their effectiveness in analyzing textual content, specifically using sentiment analysis to distinguish between authentic and malicious messages.

Research emphasizes incorporating ML algorithms to train a robust fraud detection model that facilitates continuous learning and adaptation. System adaptability is critical to staying ahead of evolving fraud techniques. The experimentation and evaluation process is rigorous and aims to verify the effectiveness of the system and its advanced defense mechanisms against fraudulent messages.

The primary goal of this research is to compare the performance of different NLP libraries when applied to the specific problem of fraudulent message detection. Evaluation criteria include, but are not limited to, accuracy, precision, recall and F1 scores. By thoroughly assessing and comparing the capabilities of various NLP libraries, the study aims to identify the most effective tools for enhancing user security, data protection and privacy in the digital age.

This research contributes valuable insights to fraud detection by offering a nuanced understanding of the strengths and limitations of various NLP libraries. The performance comparison results will inform practitioners and researchers in selecting the most appropriate NLP tools for developing robust fraud detection systems, ultimately contributing to the advancement of security measures in the context of SMS communication.

**INTRODUCTION**

Today, dominated by rapid technological progress, telecommunications, the Internet and social media are the main channels for the rapid exchange of information. The proliferation of user-generated content, including reviews, opinions, feedback, news and recommendations, has given these channels a key role as important sources of valuable information. Leveraging the prowess of advanced technologies, particularly in the field of Natural Language Processing (NLP), has become instrumental in extracting meaningful insights from this vast and dynamic reservoir of textual data.

The rapid evolution of technology has transformed the way we communicate, with telecommunications, the Internet, and social media serving as primary channels for information exchange. The vast amount of user-generated content, encompassing reviews, opinions, feedback, news, and recommendations, has turned these platforms into valuable sources of information. This information-rich environment has prompted the adoption of advanced technologies, particularly in Natural Language Processing (NLP), to extract meaningful insights from the dynamic reservoir of textual data.

NLP, a prominent branch of artificial intelligence (AI), operates at the intersection of computers and human natural language, using linguistic nuances to derive valuable information and facilitate human-computer interaction. Its applications are multifaceted and include text classification tasks such as spam detection and sentiment analysis, text generation, language translations, and document classification. This technological landscape not only improves our understanding of textual data, but also enables us to derive useful information from the language-rich content that predominates in digital communication.

Amid these technological developments, the rise of Short Text Message Service (SMS) fraud has emerged as an urgent problem that poses a significant threat to individuals navigating the digital landscape. At the same time, the increasing reliance on digital platforms for various activities has increased the risk of individuals falling victim to fraudulent schemes. Exploiting both social and technological vulnerabilities, SMS scams deftly trick users into gaining unauthorized access to their sensitive information. The resulting damages, both financial and intangible, underscore the urgency of implementing effective detection and prevention measures to limit the pervasive impact of SMS fraud.

This research seeks to address this imperative by delving into the field of NLP, specifically focusing on the performance of various NLP libraries in the context of fraudulent message detection. As digital communication becomes more and more an integral part of everyday life, the need to strengthen our defenses against fraudulent activity becomes more important. By performing a nuanced comparison of the performance of NLP libraries, this study seeks to identify the most effective tools for strengthening security measures, thereby contributing to the ongoing debate on combating SMS fraud and protecting user trust, data integrity and privacy in today's digital age.

**PROBLEM STATEMENT**

**Evaluating NLP libraries for their accuracy in detecting spam messages using pre-defined data.**

The purpose of this project is to understand how we can use NLP and ML to develop a system to build a SMS spam detection model. Using different NLP libraries to check the performance of the model, how accurately it can predict or classify a message category. Here we use Three NLP library to process RE, NLTK, PATTERN data. we will create a classification model to determine whether a text message is spam or ham. Additionally, we will learn how to implement Logistic Regression, Naïve Bayes, and a simple rule-based approach to SMS fraud detection.

System performance and efficiency will be evaluated using a variety of metrics, including accuracy, precision, and score. The aim of the project is to demonstrate the superiority of the proposed machine learning approach in terms of performance and accuracy.

**TOOLS, TECHNOLOGIES AND PRINCIPLES USED**

1. **Input data and NLP libraries:**

Data Source: The input data consists of text messages, commonly sent in human speech (English).

NLP Libraries: Using Python's NLP libraries, specifically RE, NLTK, and Pattern, to process and transform text data into a format suitable for machine learning.

2. **Text Preprocessing and Data Mining:**

Text Preprocessing: Cleaning and transforming the textual data using NLP techniques to remove irrelevant or unimportant information. This involves tasks such as lowercasing, punctuation removal, and stemming/lemmatization.

Data Mining: Extracting relevant information from the processed dataset using NLP libraries, focusing on preserving meaningful content.

3. **Supervised Learning:**

Labeling Dataset: The dataset is labeled, indicating the desired output or classification for each input message.

Supervised Learning Approach: Employing supervised learning algorithms due to the availability of labeled data. The methodology specifically mentions the use of Logistic Regression and Naïve Bayes, which are common choices for binary classification tasks.

4. **Feature Extraction:**

Feature Representation: Converting the preprocessed textual data into a numeric form suitable for machine learning models.

Feature Extraction: Extracting relevant features from the processed dataset, ensuring that the model can understand and learn patterns from the transformed data.

5. **Machine Learning Model Training:**

Algorithm Selection: Choosing Logistic Regression and Naïve Bayes algorithms for training the machine learning model.

Training Process: Utilizing the labeled dataset to train the selected algorithms. The goal is to establish a mapping between the input features (transformed textual data) and the corresponding output labels.

6. **Performance Evaluation:**

Metrics: Assessing the performance of the trained models using metrics such as accuracy, precision, recall, and F1 score.

Comparison: Comparing the performance of models trained with different NLP libraries to evaluate their effectiveness in the context of fraud message detection.

7. **Graph Plotting:**

Tools: Utilizing Matplotlib and Seaborn for graph plotting.

Visualization: Creating visual representations, such as confusion matrices or other relevant plots, to provide a clearer understanding of the model's performance.

8. **Iterative Optimization:**

Feedback Loop: Analyzing the results and iteratively optimizing the models and preprocessing techniques based on performance feedback.

Fine-Tuning: Adjusting parameters and refining the methodology to enhance the efficiency and accuracy of the models.

9. **Numeric Conversion for Analysis:**

Feature Utilization: Leveraging the numeric features extracted from the processed dataset for analysis and model development.

Model Development: Using the numeric representation to train and develop machine learning models capable of analyzing and making predictions on textual data.

**METHODOLOGY**

The methodology for detecting Fraud Message using Machine Learning (ML) algorithms, including Naïve Baye’s and Logistic Regression in combination with Natural Language Processing (NLP), involves several steps:

**1. Data Collection:** Collect a labeled dataset comprising both legitimate and Fraud text samples, covering various Textual Techniques and variations.

2. **Data Pre-processing:** Preprocess the collected data by removing irrelevant information such as repeated words, Numeric, Punctuations etc. Apply text preprocessing techniques like tokenization, stemming, and stop word removal to convert the text data into a suitable format for ML and NLP algorithms.

3. **Feature Extraction:** Extract relevant features from the preprocessed text data.

4. **Data Split:** Divide the dataset into training and testing sets. The training set is used to train the ML models, while the testing set is used to evaluate their performance.

5. **Model Training:** Train ML models such as Naïve Baye’s and logistic Regression using the training data. These models learn from the counter vectorizer and TF-IDF extracted features and the corresponding labels (Spam or Ham).

**6. Model Evaluation:** Evaluate the trained models using the testing data and calculate performance metrics such as accuracy, precision, recall, and F1-score. This assessment helps determine the effectiveness of each ML algorithm in detecting Fraud Message.

**7. Model Deployment:** Deploy the trained ML models in a test data for detecting Fraud Message. By creating a prediction model using any random message to check the validation of the model.

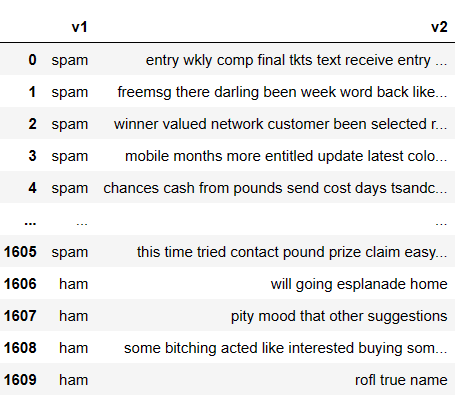
This can involve integrating the models into cellphone network security systems to identify and prevent Fraudulent.

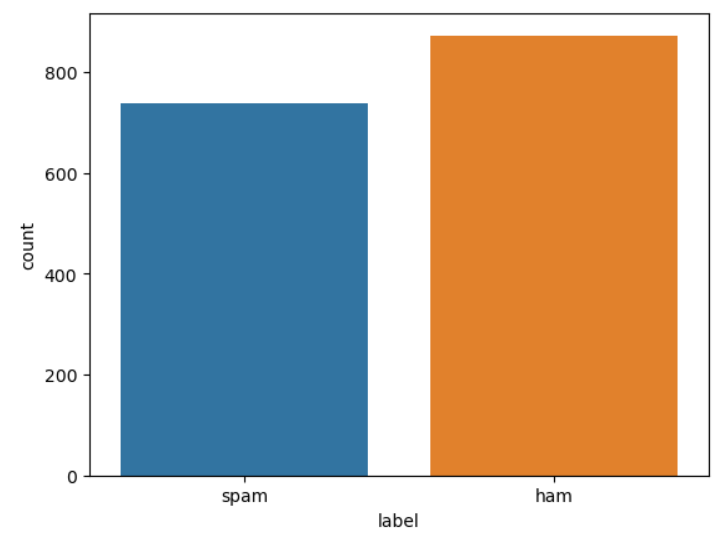
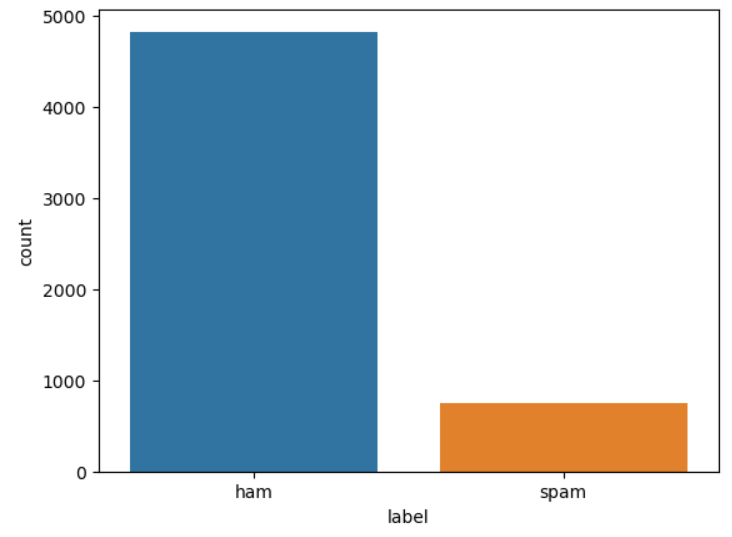
8. **Ongoing Monitoring and Updating:** Continuously monitor the performance of the deployed models and update them with new Textual samples. As Text Fraudulence techniques evolve over time, periodic retraining of the models ensures their effectiveness in detecting emerging Fraud attacks.

By combining ML algorithms like Naïve Bayes and Logistic Regression with NLP techniques, this methodology facilitates the development of robust Text Fraud detection systems that can analyze text content and identify suspicious patterns indicative of Fraud Messages.

**DATASET:**

The SMS (text) data was downloaded from [UCI datasets](https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection). It contains 5,574 SMS phone messages. The data were collected for the purpose of mobile phone spam research and have already been labeled as either spam or ham.



****As the data was unstructured having more than 4,000 as ham message and around 750 spam messages, so we have tried to balance the dataset by keeping total 1,610 SMS labelled as ham and spam.

**IMPLEMENTATION DETAILS**

We will be using python and related libraries for developing the system which can detect Fraud message from the given dataset. Using Jupiter notebook for python as environment and then imported the panda library to load the dataset for further datamining, preprocessing and applying machine learning techniques.

For preprocessing purpose will be using Regular expression, NLTK and pattern library selective methods. The RE library is known for its powerful regular expression capabilities, which are useful for pattern matching and text manipulation. We will be using Sub method from RE library performs global search and global replace on the given string. It is used for substituting a specific pattern in the string.

**NLTK Library:**

1. Data Preprocessing:

**Lowercasing:** Convert all text to lowercase.

**Punctuation Removal:** Use regular expressions to remove non-alphabetic characters.

**Most Frequent Word Removal:** Identify the most frequent words and remove them.

**Short Word Removal:** Exclude words with a length less than or equal to 3.

2. Stemming and Lemmatization:

**Porter Stemming:** Reduce words to their base or root form using the Porter Stemmer.

**Snowball Stemming:** Another stemming approach using the Snowball Stemmer.

**WordNet Lemmatization:** Reduce words to their base form using WordNet

3. Text Preprocessing:

**Stopword Removal:** Remove common English stopwords.

**Alphanumeric and Lowercasing:** Retain only alphabetic characters and convert to lowercase.

4. Feature Extraction and Oversampling:

**TF-IDF Vectorization:** Transform the preprocessed text into numerical vectors using TF-IDF.

**Count Vectorization:** An alternative vectorization method using CountVectorizer.

**SMOTE (Synthetic Minority Over-sampling Technique):** Over-sample the minority class using SMOTE.

5. Train-Test Split:

**Splitting the Data:** Divide the dataset into training and testing sets.  
6. Model Training and Evaluation (Logistic Regression):

**Logistic Regression Model:** Train a logistic regression classifier.

**Prediction and Evaluation:** Predict on the test set and evaluate the performance using accuracy, classification report, and confusion matrix.

7. Model Training and Evaluation (Naive Bayes):

**Multinomial Naive Bayes Model:** Train a multinomial naive bayes classifier.

**Prediction and Evaluation:** Predict on the test set and evaluate the performance using accuracy, classification report, and confusion matrix.

**RE Library:**

1. Data Loading and Initial Processing:

**Load Data:** Read the CSV file into a pandas DataFrame.

**Column Selection:** Choose relevant columns ('v1' and 'v2').

2. Text Preprocessing:

**Lowercasing:** Convert all text to lowercase.

**Regular Expression Cleaning:** Remove non-alphanumeric characters.

3. TF-IDF Vectorization:

**Vectorization:** Use TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical vectors.

**Feature Matrix (‘**X’**):** Transform the processed text into feature vectors.

4. Train-Test Split:

**Splitting the Data:** Divide the dataset into training and testing sets.

5. Logistic Regression Model:

**Model Initialization:** Create a Logistic Regression model.

**Training the Model:** Fit the model to the training data.

6. Model Evaluation:

**Predictions:** Predict on the test set.

**Accuracy and Classification Report:** Evaluate the performance using accuracy and a classification report.

**Confusion Matrix Visualization:** Visualize the confusion matrix.

7. Custom Message Prediction:

**User Input:** Collect user input regarding previous site visits.

**New Message Processing:** Preprocess the new message.

**Model Prediction:** Use the trained model to predict the label of the new message.

**Pattern Library:**

1. Data Loading and Initial Processing:

**Load Data:** Read the CSV file into a pandas DataFrame.

**Column Selection and Renaming:** Choose relevant columns ('v1' and 'v2') and rename them to ('label' and 'message').

2. Text Preprocessing using Pattern Library:

**Parse and Lemmatize:** Use the Pattern library to parse and lemmatize the text, considering only nouns.

**Create Processed Messages:** Apply the preprocessing function to create a new column 'processed\_message'.

3. Train-Test Split:

**Splitting the Data:** Divide the dataset into training and testing sets.

4. Classification Using Custom Rule:

**Custom Rule for Classification:** Define a custom rule-based function to classify messages.

**Accuracy and Confusion Matrix:** Evaluate the performance using accuracy and a confusion matrix.

5. Visualization:

**Confusion Matrix Visualization:** Visualize the confusion matrix.

6. Custom Message Prediction:

**User Input:** Collect user input regarding previous site visits.

**New Message Processing:** Preprocess the new message.

**Model Prediction:** Use the trained model (custom rule) to predict the label of the new message.

**RESULT, ANALYSIS**

**Generating predicted list using rule-based function on test set:**

Applying the classify messages function to the X test data to generate a list of predicted labels for each message. The resulting list of predicted labels is stored in the y pred variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Library** | **Accuracy %** | **Classification report** | **Graph** |
| 1. | NLTK | 93% |  |  |
| 2. | RE | 91% |  |  |
| 3. | Pattern | 57% |  |  |

**CONCLUSION**

**NLTK Library:**

The use of the NLTK library in the performance comparison for fraud message detection showcases its effectiveness in text preprocessing and feature extraction. Leveraging techniques like stemming and stop-word removal, NLTK contributes to building a robust model for fraud detection. Its integration with machine learning algorithms enhances the system's capability to differentiate between genuine and fraudulent messages, thereby bolstering security measures in the digital realm.

**RE Library:**

The implementation of the regular expressions (re) library in the performance comparison for fraud message detection demonstrates its utility in text processing and pattern matching. While re is powerful for basic string manipulation and filtering, its limitations in handling more complex linguistic features and semantics make it less suitable for comprehensive fraud message detection. In comparison to dedicated NLP libraries, re may fall short in capturing nuanced language structures and context, highlighting the need for more sophisticated tools in addressing the challenges of fraud detection in textual content.

**Pattern Library:**

The utilization of the Pattern library in the performance comparison for fraud message detection showcases its effectiveness in handling various natural language processing (NLP) tasks. Pattern's capabilities in sentiment analysis, part-of-speech tagging, and text processing contribute to its versatility. However, the library's performance may be influenced by the specific characteristics of fraud messages, and it might not provide the same depth of linguistic analysis as more specialized NLP libraries. While Pattern offers valuable features, its suitability for fraud detection depends on the specific requirements and complexities of the task, and a comprehensive evaluation against dedicated NLP libraries is essential for making informed choices in fraud detection applications.

**FUTURE**

In the future, the performance comparison of NLP libraries for fraud message detection will likely involve integrating advanced deep learning models, such as transformer-based architectures, for improved contextual understanding. Additionally, the incorporation of explainable AI, multimodal analysis, and real-time processing will enhance accuracy and responsiveness in identifying fraudulent patterns.

**Using Deep Learning Models:**

One of the prominent directions for future research involves the integration of advanced deep learning models, particularly transformer-based architectures. Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have demonstrated exceptional capabilities in understanding contextual nuances within text.

Transformer-based models excel in capturing long-range dependencies and contextual information, making them well-suited for the complexities of natural language. Integrating these models into fraud message detection systems can enhance the understanding of linguistic nuances, improving the accuracy of identifying fraudulent patterns.

**Real-time Processing and Adaptive Systems:**

The demand for real-time fraud detection is expected to drive the development of systems capable of processing and analyzing messages as they occur. Real-time processing is essential for identifying and responding to emerging threats promptly.

Adaptive systems that continuously learn from new data and adjust their models over time will be crucial. This adaptability ensures that fraud detection models remain effective in the face of evolving tactics employed by malicious actors.

**REFERENCES**

[1] Prusty, S.R., Sainath, B., Jayasingh, S.K., Mantri, J.K. (2022). SMS Fraud Detection Using Machine Learning. In: Udgata, S.K., Sethi, S., Gao, XZ. (eds) Intelligent Systems. Lecture Notes in Networks and Systems, vol 431. Springer, Singapore. https://doi.org/10.1007/978-981-19-0901-6\_52

[2] Chan PPK, Yang C, Yeung DS, Ng WWY (2015) Spam filtering for short messages in adversarial environment. Neurocomputing 155:167 176. <https://doi.org/10.1016/j.neucom.2014.12.034>

[3] Saurabh Kr. Srivasatava, Roshan Kumari, Sandeep Kr. Singh, "An ensemble based NLP feature assessment in binary classification", 2017 International Conference on Computing, Communication and Automation (ICCCA), pp.345-349, 2017.

[4] NLP: Spam Detection in SMS (text) data using Deep Learning

[5] JETIR --Detection of Fraud Applications using Sentiment Analysis