**ABSTRACT:**

*With the growing influence of social media platforms such as Instagram in modern communication, the threat of phishing attacks and hate speech targeting users through comments has become more pronounced. To effectively combat these risks, we propose the implementation of a Machine Learning-based Instagram Comments Phishing and Hate Speech Detection System. This innovative solution is designed to automatically detect and highlight potential phishing attempts within Instagram comments in real-time, thereby bolstering user safety and platform security.*

*To develop and evaluate the efficacy of this system, we begin by assembling a real-time dataset of Instagram comments. Employing meticulous data cleaning techniques and conducting exploratory data analysis, we initially scrutinize the dataset. Subsequently, we categorize the data into two distinct groups: text-based comments and those containing links, enabling a more focused analysis.*

*Further refining our approach, we identify the most commonly used hate words and acronyms recurrently present in comments to enhance the system's accuracy. Additionally, we've developed a user-friendly Streamlit application to facilitate input of comments and sentiment prediction. Phishing links are promptly identified and flagged with a conspicuous red marker, while the system also performs a thorough examination of the legitimacy of URLs for added security.*

*By implementing this comprehensive system, we aim to fortify Instagram's defense against phishing attacks and hate speech, ultimately fostering a safer and more secure environment for users.*

**INTRODUCTION:**

The proliferation of social media platforms has escalated online interactions, unfortunately also paving the way for harmful practices like phishing. Instagram, being a prominent platform, has been a hotspot for such malevolent activities. This academic critique highlights the deployment of machine learning and data science strategies to pinpoint phishing activities within Instagram comments.

Social media platforms serve as digital reflections of our societal interactions, facilitating connections, fostering friendships, enabling expression of ideas, sharing of aspirations, providing feedback, and promoting collaboration, among other activities. Users can also partake in transactions, fundraising, follower acquisition, campaign promotions, and more, contributing to the rising popularity of these platforms due to their advanced features.

This paper focuses on Instagram due to the extensive data available on spam comments. The characteristics of posts and comments on Instagram include the use of informal language, abundant use of emoticons/emojis, frequent abbreviations and typos, extensive use of code-mix data language, and varying lengths of comments. Instagram also features a reply-response structure with no hierarchy and uses mentions with the @ sign.

Currently, the primary solutions to tackle spam comments on Instagram are manual. Users can delete spam comments individually, which is time-consuming and requires constant monitoring. Instagram also offers a feature to report comments as spam manually, but this too needs to be done individually. An alternative solution to minimize spam comments is to switch the Instagram account to private.

A variety of machine learning techniques, in conjunction with natural language processing (NLP), can be employed as techniques for detecting spam comments on social media.

The method has the lowest accuracy but the second-fastest prediction efficiency after shows good accuracy performance but is the slowest in training and testing.

**LITERATURE REVIEW**

The paper **titled “Spam Comments Detection on Instagram Using Machine Learning and Deep Learning Methods”** presents a comprehensive study on the application of machine learning (ML) and deep learning (DL) techniques for detecting spam comments on Instagram.

->The authors highlight that the popularity of public figures on Instagram often leads to an increase in the number of followers and consequently, the number of comments on their posts. However, not all comments are relevant to the post, and some may include advertising, links, or clickbait comments, which are typically referred to as spam comments.

->The research compares ML and DL classification methods based on a collected Indonesian Instagram spam comment dataset. The study was conducted in several steps: dataset preparation, pre-processing, simple normalization, features generation using TF-IDF and word embedding, application of ML and DL classification methods, performance evaluation, and comparison.

->The compared the accuracy, F-1 score, precision, and recall from ML and DL results. The research shows that ML and DL methods do not significantly differ. The Linear SVM, Extreme Tree (ET), Regression, and Stochastics Gradient Descent algorithms can reach an accuracy of 0.93. Meanwhile, the DL method has the highest accuracy of 0.94 using the Simple Transformer BERT architecture.

->The paper concludes that the difference between ML and DL methods is not significantly different, indicating that both approaches can be effectively used for spam comment detection on Instagram.

The paper **“Enhancing Spam Comment Detection on Social Media with Emoji Feature and Post-Comment Pairs Approach using Ensemble Methods of Machine Learning”** provides an in-depth analysis of the use of machine learning (ML) and deep learning (DL) techniques for detecting spam comments on Instagram.

The study was conducted in several steps: dataset preparation, pre-processing, simple normalization, features generation using TF-IDF and word embedding, application of ML and DL classification methods, performance evaluation, and comparison.

->The authors compared the accuracy, F-1 score, precision, and recall from ML and DL results. The research shows that ML and DL methods do not significantly differ. The Linear SVM, Extreme Tree (ET), Regression, and Stochastics Gradient Descent algorithms can reach an accuracy of 0.93. Meanwhile, the DL method has the highest accuracy of 0.94 using the Simple Transformer BERT architecture.

->The paper concludes that the difference between ML and DL methods is not significantly different, indicating that both approaches can be effectively used for spam comment detection on Instagram.

->This research contributes to the growing body of literature on the application of ML and DL techniques in social media spam detection and provides valuable insights for future research in this area.

The research paper **“Enhancing Spam Email Classification Using Effective Preprocessing Strategies and Optimal Machine Learning Algorithms”** offers a detailed exploration of the use of machine learning (ML) and deep learning (DL) techniques for the classification of spam emails.

->The authors underscore that the surge in popularity of email communication has unfortunately also led to an increase in spam emails. To combat this, they propose a content-based spam email classification system that applies various text preprocessing techniques.

->The authors applied several combinations of preprocessing methods, such as stopping, removing tags, converting to lower case, removing punctuation, removing special characters, and natural language processing, to the extracted content from the email. They then used machine learning algorithms like Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) to classify an email as ham or spam.

->The research found that applying stemming in preprocessing to the RF classifier yielded the best results, achieving 99.2% accuracy on the Spam Assassin dataset and 99.3% accuracy on the Enron dataset.

 ->The paper introduces a unique perspective by highlighting the fine-tuning of preprocessing techniques. The focus is on removing tags and certain special characters, while retaining those that improve spam email classification accuracy.  
In conclusion, the paper emphasizes the crucial role of preprocessing and contributes to a more nuanced understanding of effective strategies for robust spam detection.

**RESEARCH METHODOLOGY:**

DATA COLLECTION

PERFORM EXPLLORATORY DATA ANALYSIS

SEGREGATE THE DATA INTO TWO DATA SETS

DATA WITH PPHISHING LINKS

DATA WITH HATE SPEECH

IMPLEMENTATION OF ML ALGORITHM

SENTIMENT ANALYSIS USING AFFIN, HATE DATA SETS

DATA PREPROCESSING

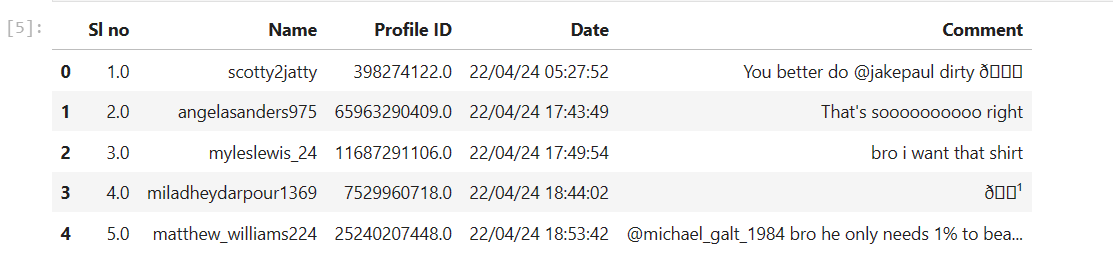
DATA CLEANING

FLAG THE LINKS WHICH ARE PHISHING

WEB APP USING STREAMLIT APP

**DATA GATHERING:**

The data has been gathered from the real time Instagram comments using the third-party web site called **Export Instagram comments.** The raw data set contains [2028 rows x 5 columns]. And the below snapshot shows the first five rows of the data set.



**META DATA:**

1. **Sl no (Serial Number)**: This column represents the serial number or index of each row in the dataset. It serves as a unique identifier for each record.
2. **Name**: This column contains the names of the users who posted the comments on Instagram.
3. **Profile ID**: This column holds the profile IDs associated with each user. Profile IDs are unique identifiers assigned to each Instagram account.
4. **Date**: This column indicates the date and time when each comment was posted on Instagram. The format appears to be in "DD/MM/YY HH:MM:SS" format, representing day, month, year, hour, minute, and second.
5. **Comment**: This column contains the actual comments posted by users on Instagram. It includes text-based content, emojis, and mentions (e.g., "@jakepaul").

**DATA TYPES**

1. Sl no (Serial Number)**: Integer or Numeric (assuming it represents a unique index).**
2. Name**: String or Text (assuming it contains the usernames of the Instagram users).**
3. Profile ID**: Numeric (assuming it represents unique identifiers for Instagram profiles).**
4. Date**: Date and Time (assuming it is stored as a datetime object or string).**
5. Comment**: String or Text (assuming it contains the textual content of the comments posted by users).**

**DATA CLEANING, PRE-PROCESSING, AND NORMALIZATION:**

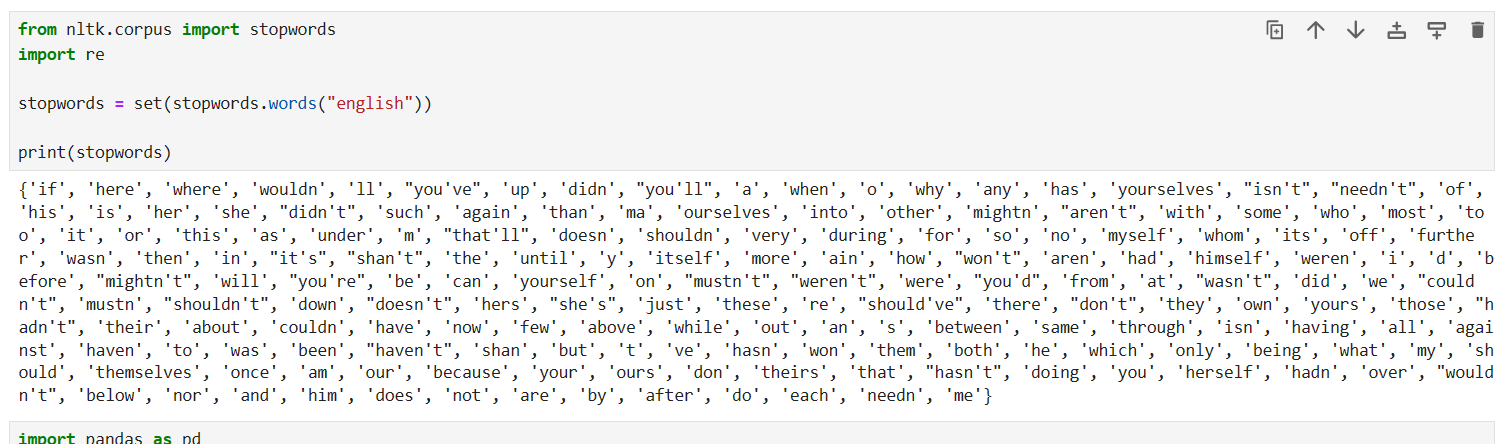
The hate speech and phishing link are two different topics, so we need to segregate the data accordingly. The data has been divided into two datasets. One with the text data for text analysis and for the model building. The other dataset is for phishing link detection.

The data cleaning process is including in handling missing values, remove unnecessary data which is irrevalent to process, removing the irrelevant symbols and spaces from the data set for text analysis, standardize the data set called normalizing.

Pre- processing includes performing the exploratory data analysis and gain some initial insights from the data for the further analysis.

**DATA CLEANING:**

The data set initially downloaded from the directly from the Instagram, we can observe a lot of noise in the comments column which is necessary for the text analysis. So, the data has been cleaned and re ordered. The stop words have been removed from the data using the **NLTK** library.



The next step is to remove the numbers, punctuations and emojis from the text so that I could be easy for the further analysis.

The rows are dropped from the data set which are empty so that we could make the effective data set for the analysis. The following is the final data set obtained from the cleaning process.

A screenshot of a computer program

Description automatically generated

The other data set is cleaned and filtered for the links for further analysis using REGEX concept.

Here is the cleaned file snapshot.

A screenshot of a computer

Description automatically generated

**IMPLEMENTING MACHINE LEARNING ALGORITHM:**

**AFINN DATASET**

The first step is to carry out the Exploratory data analysis. As a part of implementing the machine learning algorithm. The sentiment analysis has been carried out. For sentiment analysis we have used the afinn data set.

A graph of different colored squares

Description automatically generated

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Precision (Negative)** | **Precision (Neutral)** | **Precision (Positive)** | **Recall (Negative)** | **Recall (Neutral)** | **Recall (Positive)** | **F1-score (Negative)** | **F1-score (Neutral)** | **F1-score (Positive)** | **Accuracy** |
| Logistic Regression | 0.66 | 0.6 | 0.6 | 0.41 | 0.84 | 0.48 | 0.51 | 0.7 | 0.53 | 0.61 |
| SVM | 0.61 | 0.62 | 0.61 | 0.46 | 0.82 | 0.48 | 0.52 | 0.71 | 0.54 | 0.62 |
| Random Forest | 0.67 | 0.64 | 0.6 | 0.4 | 0.85 | 0.56 | 0.5 | 0.73 | 0.58 | 0.63 |

**FINAL REPORT:**

**Comparison and Insights:**

* Accuracy: Random Forest has the highest accuracy (63%), followed by SVM (62%) and Logistic Regression (61%). This indicates that Random Forest performed slightly better overall in predicting sentiments.
* Precision: Random Forest and Logistic Regression have similar precision scores for each sentiment class. SVM has slightly lower precision for the 'negative' and 'positive' classes compared to the other models.
* Recall: SVM has the highest recall for the 'neutral' class, followed closely by Random Forest and Logistic Regression. For the 'negative' and 'positive' classes, Random Forest has the highest recall, followed by SVM and Logistic Regression.
* F1-score: Random Forest generally has higher F1-scores across all classes compared to SVM and Logistic Regression. This suggests that Random Forest achieved a better balance between precision and recall for each class.
* Macro and Weighted Averages: Random Forest shows slightly better performance in macro and weighted average F1-scores compared to SVM and Logistic Regression.

Overall, Random Forest seems to perform slightly better than SVM and Logistic Regression in this sentiment analysis task.

**HATE DATASET:**

The hate data set has been developed by us in which we choose the most repeated bad words in the Instagram comments and other acronyms of the bad words(which are offensive”). So here is the whole report.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Precision (Negative)** | **Precision (Neutral)** | **Precision (Positive)** | **Recall (Negative)** | **Recall (Neutral)** | **Recall (Positive)** | **F1-score (Negative)** | **F1-score (Neutral)** | **F1-score (Positive)** | **Accuracy** |
| Logistic Regression | 0 | 0.9 | N/A | 0 | 1 | N/A | 0 | 0.94 | N/A | 0.9 |
| Random Forest | 0.86 | 0.91 | N/A | 0.18 | 1 | N/A | 0.29 | 0.95 | N/A | 0.91 |
| SVM | 1 | 0.91 | N/A | 0.12 | 1 | N/A | 0.21 | 0.95 | N/A | 0.91 |

1. **Performance Variation Across Classes**: Across all models, the performance varies significantly between the 'neutral' class and the other classes ('negative' and 'positive'). For the 'neutral' class, the precision, recall, and F1-score are consistently high (ranging from 0.90 to 1.00), indicating that the models are good at predicting neutral sentiments. However, for the 'negative' and 'positive' classes, the performance is lower, with precision, recall, and F1-score ranging from 0.00 to 1.00. This suggests that the models struggle to accurately classify 'negative' and 'positive' sentiments compared to 'neutral' sentiments.
2. **Imbalanced Classes**: The imbalance in class distribution (e.g., fewer samples for 'negative' and 'positive' classes compared to 'neutral') may contribute to the lower performance for these classes. The models seem to be biased towards predicting the majority class ('neutral'), resulting in lower performance for the minority classes ('negative' and 'positive').
3. **Model Comparison**: While all models achieve high accuracy (ranging from 0.90 to 0.91), there are differences in their performance metrics. For example, Logistic Regression has low precision, recall, and F1-score for the 'negative' class, indicating poor performance in identifying negative sentiments. In contrast, Random Forest and SVM perform better for the 'negative' class but still struggle compared to the 'neutral' class.
4. **Generalization Ability**: Despite differences in performance metrics, all models demonstrate good generalization ability, as indicated by high overall accuracy. This suggests that the models are able to effectively learn and generalize patterns from the data to make accurate predictions on unseen samples.
5. **Consideration for Application**: When deploying these models in a real-world application, it's important to consider the specific use case and the implications of misclassifications. For example, if accurately detecting negative sentiments is crucial for the application (e.g., sentiment analysis for customer feedback), then models with higher performance for the 'negative' class should be prioritized.

**WEB APP USING STREAMLIT:**

We have created a web app using the trained model that can predict the words which are hate or offensive.

Below snapshot shows the UI of the page.

A screenshot of a computer

Description automatically generated

**PHISHING LINK DETECTION:**

The REGEX concept has been used to detect phishing links. The links are being red flagged If they found matched with the expression.

A screenshot of a computer

Description automatically generated

**CONCLUSION AND RECOMMENDATION:**

The analysis conducted on the Instagram comments data involved several steps, including data cleaning, preprocessing, and implementing machine learning algorithms for sentiment analysis and hate speech detection. Here are the key takeaways from the analysis:

1. **Sentiment Analysis:** Three machine learning models (Logistic Regression, SVM, Random Forest) were trained and evaluated using the AFINN dataset. Random Forest exhibited slightly better performance in terms of accuracy and F1-scores compared to SVM and Logistic Regression. However, all models showed decent performance in predicting sentiments, with Random Forest performing marginally better overall.
2. **Hate Speech Detection:** A custom dataset of hate speech terms and acronyms was used to train the models. All models achieved high accuracy, but there was significant variation in performance across classes. While the models demonstrated high precision, recall, and F1-scores for neutral sentiments, they struggled with accurately classifying negative and positive sentiments. This could be attributed to imbalanced class distribution and the models' bias towards predicting the majority class.
3. **Web App Development:** A web application was developed using Streamlit, integrating the trained hate speech detection model. This enables users to input text and receive predictions on whether the text contains hate speech or offensive language. The user interface provides a user-friendly experience for leveraging the model's capabilities.
4. **Phishing Link Detection:** Regular expressions (REGEX) were utilized to detect phishing links in the comments. Any links matching the predefined expression were flagged as potential phishing attempts. This adds an additional layer of security and protection for users interacting with the platform.

Recommendations:

1. **Further Model Refinement:** To improve the performance of sentiment analysis and hate speech detection models, further refinement and fine-tuning could be conducted. This may include experimenting with different feature engineering techniques, exploring advanced machine learning algorithms, or addressing the class imbalance issue through techniques like oversampling or undersampling.
2. **Continuous Monitoring and Updates:** Given the dynamic nature of social media content and language usage, it's crucial to continuously monitor and update the models to adapt to evolving trends and patterns in user behavior. Regular retraining of models with updated data can help maintain their effectiveness over time.
3. **User Education and Awareness:** Alongside technological solutions, educating users about responsible online behavior and the consequences of hate speech can help foster a safer and more inclusive online community. Providing resources and guidelines on acceptable conduct can empower users to contribute positively to the platform.
4. **Collaboration with Content Moderators:** Collaboration between automated detection systems and human content moderators can enhance the platform's ability to identify and address harmful content effectively. Collaboration with big giants and getting more reliable labelled data makes the model more suitable and could be included in applications like Instagram.

**REFERENCES:**

1. S. Aiyar and N. P. Shetty, “N-Gram Assisted Youtube Spam Comment Detection,” Procedia Computer Science., vol. 132, pp. 174–182, 2018, doi: 10.1016/j.procs.2018.05.181.
2. R. Chrismanto, A. K. Sari, and Y. Suyanto, “CRITICAL EVALUATION ON SPAM CONTENT DETECTION IN SOCIAL MEDIA,” Journal of Theoretical and Applied Information Technology (JATIT), vol. 100, no. 8, pp. 2642–2667, 2022, [Online]. Available: <http://www.jatit.org/volumes/Vol100No8/29Vol100No8.pdf>
3. Chrismanto and Y. Lukito, “Klasifikasi Komentar Spam Pada Instagram Berbahasa Indonesia Menggunakan K-NN,” in Seminar Nasional Teknologi Informasi Kesehatan SNATIK), 2017, pp. 298–306.
4. F. Prabowo and A. Purwarianti, “Instagram online shop’s comment classification using statistical approach,” in Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017, 2018, pp. 282–287. doi: 10.1109/ICITISEE.2017.8285512.
5. W. Zhang and H.-M. Sun, “Instagram Spam Detection,” in 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Jan. 2017, pp. 227–228. doi: 10.1109/PRDC.2017.43.
6. Priyoko and A. Yaqin, “Implementation of naive bayes algorithm for spam comments classification on Instagram,” in 2019 International Conference on Information and Communications Technology, ICOIACT 2019, 2019, pp. 508–513. doi: 10.1109/ICOIACT46704.2019.8938575.
7. N. A. Haqimi, N. Rokhman, and S. Priyanta, “Detection Of Spam Comments On Instagram Using Complementary Naïve Bayes,” IJCCS (Indonesian Journal of Computing and Cybernetics Systems, vol. 13, no. 3, p. 263, Jul. 2019, doi: 10.22146/ijccs.47046.
8. T. Wu, S. Wen, Y. Xiang, and W. Zhou, “Twitter spam detection: Survey of new approaches and comparative study,” Computers & Security, vol. 76, pp. 265–284, Jul. 2018, doi: 10.1016/j.cose.2017.11.013.
9. A. Mandadi, S. Boppana, V. Ravella and R. Kavitha, “Phishing website detection using machine learning,” in 2022 IEEE 7th Int. Conf. for Convergence in Technology (I2CT), Mumbai, India, pp. 1–4, 2022. <https://doi.org/10.1109/i2ct54292022.9824801>
10. A. Kulkarni and L. L. Brown, “Phishing websites detection using machine learning,” International Journal of Advanced Computer Science and Applications, vol. 10, 2019. <https://doi.org/10.14569/ijacsa.2019.0100702>