**CSCE 5215 – MACHINE LEARNING**

**Project Title – Spam Message Classification using Machine Learning**

**GROUP-25**

**ABDUL AHAD MOHAMMED | ANIL KUMAR BHOGADI | PRANAY KUMAR REDDY MALREDDY**

**GOALS & OBJECTIVES:**

**MOTIVATION:**

The motivation for a project on spam message classification using machine learning typically stems from the need to address the pervasive issue of unwanted or unsolicited messages, commonly known as spam. Here are some key motivations for undertaking such a project:

* Security Concerns
* Resource Efficiency
* User Experience Improvement

**SIGNIFICANCE:**

The project "Spam Message Classification using Machine Learning" holds significant importance in the field of natural language processing (NLP) and cybersecurity. Here are some key aspects of its significance:

* Email filtering
* Spam Detection
* Text Classification

**OBJECTIVES:**

Spam messages are becoming highly common in different means of communication, including

email, text messaging, and social media. The objective of this project is to reduce and filter

undesired communication by using machine learning techniques to accurately classify messages

as spam or non-spam. We can improve user experiences and safeguard individuals and

businesses from spam-related hazards by building an effective spam message classifier.

**FEATURES:**

A project on spam message classification using machine learning typically involves the development of a model that can automatically identify whether a given message is spam or not. Here are some common features and steps involved:

* Data collection
* Preprocessing data
* EDA
* Model training

**INCREMENT – 1:**

**Related Work (Background):**

There has been a great deal of research done on the identification of SMS spam; machine learning classifiers have been used in several studies to address the problem of separating spam from legitimate (ham) SMS conversations. In relation to this, Gupta et. al. (2018) performed aresearch of spam SMS detection with machine learning classifiers that featured at the 2018 International Conference on Contemporary Computing and evaluated the efficacy of several machine learning models in identifying spam emails. The study provides insightful data on classifier performance and acts as a standard against which similar algorithms may be evaluated for effectiveness.

Furthermore, in 2017, Shafi'I et al. provided an overview of mobile SMS spam filtering approaches which was published in IEEE Access, highlighted the wide range of approaches taken by authors to prevent SMS spam in the literature. It provides a thorough examination of the benefits and drawbacks of various strategies, including machine learning-based ones. This assessment helps to clarify the range of SMS spam filtering methods and direct the selection of practical strategies.

Saha et.al. in 2021 used machine learning techniques to solve the issue of SMS spam detection. This effort adds to the body of knowledge and offers insightful information to improve spam detection systems' functionality as they attempted to apply machine learning approaches to increase the accuracy of spam identification while providing a thoughtful perspective on the latest advancements in SMS filtering.

**Dataset:**

With the wide range of messages this collection offers, it is feasible to quickly train and evaluate the machine learning model using data from several websites. For this investigation, Kaggle provided the SMS Spam Collection, which consists of more than 5,500 English-language SMS messages classified as either "spam" or "ham" (authentic). This dataset serves as training and testing data for a machine learning model that classifies spam communications. The dataset is cleaned, tokenized, and assessed during the phases of data collection and preparation, which improves the accuracy and lifespan of the model.

**Detail design of Methods:**

In order to get insight into the qualities of the dataset, the project employs a methodology that starts with preprocessing and data collecting and progresses to exploratory data analysis. Pie and bar charts are used to display the distribution of the dataset, which was acquired via the Kaggle API. Pandas is then used to import and process the dataset. Using the Seaborn and Matplotlib packages, the EDA phase includes statistical analysis such as message length distribution and common word identification in spam and non-spam messages.

A graph of a number of spam and ham messages

Description automatically generated

The pad sequences and Keras Tokenizer functions are used to construct a tokenization and padding process for the machine learning model. Three layers make up the architecture of the model: dense layers, global average pooling layer, and embedding layer. Adam optimizer and binary crossentropy loss are used in the compilation of the model. Training takes place throughout 50 epochs, and TensorFlow's Keras utilities' plot model function is used to view the architecture of the model. The accuracy of the trained model is assessed, and the findings are shown. To be used later, the finished model is stored.

Sample texts are forecasted to show off the model's capability and show how well it can categorize messages as spam or not. Furthermore, real-time predictions can be obtained by processing user input, which has practical applications for spam identification across a range of communication channels.

**Analysis:**

During the analysis phase, the dataset, model performance, and important takeaways from the machine learning strategy are all thoroughly examined. Aspects of the analysis are covered in this section.

* **Label Distribution:** The first analysis looks at how the labels (spam and ham) are distributed in the SMS collection. Pie charts and bar charts give a clear picture of the composition of the dataset. Class imbalances can be better understood by examining the pie chart, which provides the distribution %, while the bar chart shows the quantity of spam and ham messages.

A pie chart with numbers and text

Description automatically generated

* **Message Length Distribution:** It is important to examine the message length distribution in both spam and ham transmissions. Histograms make it possible to spot patterns by showing the frequency of messages relative to various message lengths. This research helps determine whether the kind and length of messages are correlated.

A graph of a message

Description automatically generated

* **Common Words Analysis:** Further analysis delves into the most common words present in both spam and ham messages. Bar charts reveal the top words along with their respective counts, providing insights into the distinctive language used in spam and non-spam messages. This step is essential for feature selection and understanding the vocabulary that influences predictions.

A graph of a number of words

Description automatically generated

A graph of a number of words

Description automatically generated

* **Model Training and Evaluation:** The core of the analysis involves training the machine learning model. The chosen model architecture includes an embedding layer, global average pooling, and dense layers. The model is compiled with binary cross-entropy loss and the Adam optimizer. Training over a specified number of epochs is performed using the training dataset, with validation on a separate test dataset.
* **Model Evaluation Metrics:** The model's performance is assessed using evaluation metrics, particularly accuracy. The percentage accuracy is calculated based on the model's predictions compared to the actual labels in the test dataset. This metric provides a quantitative measure of how well the model generalizes to new, unseen data.

A computer screen with text

Description automatically generated

* **Sample Predictions:** The analysis concludes with sample text predictions using the trained model. Sample messages, both spam and ham, are inputted into the model, and predictions are displayed along with confidence scores. This qualitative assessment offers a practical understanding of the model's decision-making process.

A screenshot of a computer screen

Description automatically generated

**Implementation**

The project is implemented as a Python notebook on Google Colab, leveraging various libraries and tools for machine learning.

* **Setting up the Kaggle Environment:** The initial step involves setting up the Kaggle environment on Google Colab. The **get\_kaggle\_dataset** function is utilized to download the SMS Spam Collection dataset from Kaggle. The function handles authentication, downloading, and extraction of the dataset, making it accessible for further processing.
* **Loading and Preprocessing Data:** The SMS dataset, obtained from Kaggle, is loaded into a Pandas DataFrame. The dataset consists of labeled SMS messages, categorized as spam or ham. Columns are appropriately named as "label" and "message." Data preprocessing includes encoding labels, analyzing message lengths, and visualizing label distribution.
* **Exploratory Data Analysis (EDA):** Exploratory Data Analysis is conducted to gain insights into the dataset. Visualizations using Seaborn and Matplotlib showcase label distribution, message length distribution, and the most common words in spam and ham messages. These visual aids make it easier to comprehend the dataset's properties.
* **Model Development:** TensorFlow and Keras are used in the development of the machine learning model. An embedding layer, global average pooling, and dense layers make up the architecture. The Adam optimizer and binary cross-entropy loss are used to construct the model. There are defined hyperparameters for vocabulary size, embedding dimensions, and maximum sequence length.
* **Model Training:** Using the preprocessed dataset, the model is trained over a predetermined number of epochs. Training and validation accuracy are monitored to assess the model's convergence. The trained model is then evaluated using a separate test dataset, and performance metrics, such as accuracy, are calculated.
* **Sample Predictions:** Sample predictions are made using the trained model. A set of predefined sample texts, including both spam and ham messages, is inputted into the model. Predictions are generated, indicating whether each message is classified as spam or ham, along with confidence scores.
* **Model Saving and Loading:** The trained model is saved for future use. It can be loaded back into the notebook using TensorFlow's **load\_model** function. This enables easy reuse of the model for making predictions on new data.

The entire implementation is encapsulated within a Python notebook, providing a clear and executable record of the entire machine learning pipeline.

**Preliminary Results**

The preliminary results showcase promising performance of the machine learning model in classifying SMS messages as spam or ham. After training for 50 epochs, the model achieved an impressive accuracy of 98.42% on the test dataset. This high accuracy suggests that the model has effectively learned and generalized patterns from the training data to make accurate predictions on unseen messages.

One important measure of the model's total prediction performance is the accuracy metric. An accuracy of 98.42% in this case indicates that roughly 98.42% of the test messages were properly identified by the model. Based on the attributes retrieved after training, the model's ability to distinguish between spam and ham communications is demonstrated by its high accuracy rate.

It is crucial to remember that, although accuracy offers a broad picture of the model's effectiveness, additional assessment measures like precision, recall, and F1-score may offer a more in-depth perspective, particularly when dealing with imbalanced datasets. Recall is the percentage of correctly predicted spam messages out of all actual spam messages, whereas precision is the percentage of correctly predicted spam messages out of all messages predicted as spam.

Additionally, the choice of 50 epochs for training the model suggests that the model has undergone sufficient iterations to converge and capture the underlying patterns in the dataset. Further experimentation with hyperparameters, architecture, or training duration could be explored to fine-tune the model for potential performance improvements.

These preliminary results lay a solid foundation for the effectiveness of the developed model in spam classification. Future iterations and refinements may focus on optimizing the model further, considering additional evaluation metrics and potentially addressing challenges such as overfitting or underfitting in the training process.

Overall, the achieved accuracy of 98.42% is a promising indication of the model's capability to successfully identify and classify spam messages, demonstrating the potential practical utility of the implemented machine learning solution.

**Project Management:**

Implementation Status Report:

Work Completed:

* Description: As per our knowledge we have finished the project as there is no complications involved in this project.
* Responsibility:

Background Work, Dataset – Pranay

Detail Design, Analysis – Anil, Ahad

Implementation – Ahad, Anil, Pranay

* Contribution:

Ahad – 100% (Implementation – 50%, Detail design, Analysis – 50%)

Anil – 100% (Detail design, Analysis – 50%, Implementation – 50%)

Pranay – 100% ( Background work – 50%, Implementation – 50%)

Work to be Completed:

* Most of our work is completed by now.
* Issues/Concerns: No issues/concerns, All of our team members cooperated very well.

**References**

Gupta, M., Bakliwal, A., Agarwal, S., & Mehndiratta, P. (2018, August). A comparative study of spam SMS detection using machine learning classifiers. In *2018 eleventh international conference on contemporary computing (IC3)* (pp. 1-7). IEEE.

Shafi’I, M. A., Abd Latiff, M. S., Chiroma, H., Osho, O., Abdul-Salaam, G., Abubakar, A. I., & Herawan, T. (2017). A review on mobile SMS spam filtering techniques. *IEEE Access*, *5*, 15650-15666.

Gupta, S. D., Saha, S., & Das, S. K. (2021, February). SMS spam detection using machine learning. In *Journal of Physics: Conference Series* (Vol. 1797, No. 1, p. 012017). IOP Publishing.