**AI-POWERED SPAM CLASSIFIER**

Phase 2   
 Innovation on the Project

|  |  |
| --- | --- |
| **PROJECT TITLE** | AI – Powered Spam Classifier |
| **SKILLS TAKEN AWAY** | * Python (+Python ML) * Csv files Handling * (CNN,Navis Bayes) ML algorithm * UI deployment |
| **DOMAIN** | ML - Cybersecurity |

***Project Definition :***

The "AI-Powered Spam Classifier" project's primary goal is to develop an intelligent system that automatically categorizes incoming messages as either spam or legitimate using advanced AI and ML techniques. This system seeks to enhance communication channels, improve user experience, and reduce unwanted content.

A diagram of a company

Description automatically generated

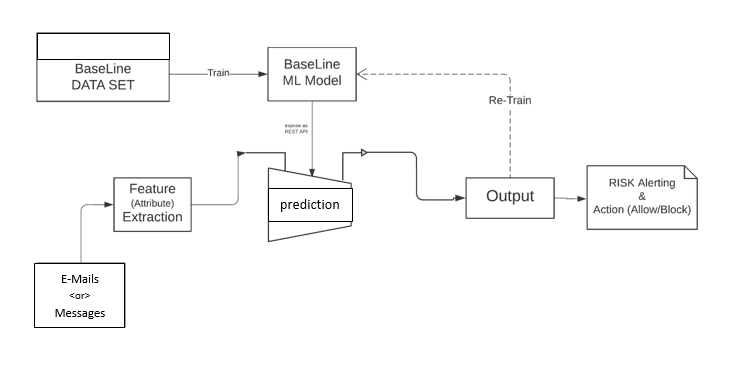
***Innovation :***

|  |  |
| --- | --- |
| Feedbacking Loop 🔁 | Encourage users to provide feedback on the effectiveness of their custom rules, allowing them to fine-tune and improve their rules over time with respect to our project. |
| User-Defined Rules 💡 | Allow users to define their own spam detection rules or criteria based on their preferences and requirements. |

***Project Execution Procedures :***

* **Collecting Data**:
  + Find or create a dataset with examples of both spam and non-spam (ham) messages.
  + Here, we’re collecting the dataset for our project from *Kaggle.com* website.
  + Dataset Link : <https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset>
* **Preparing Data**:
  + Clean and organize the data. Remove unnecessary information and ensure it's ready for analysis.
  + Make the data be labelled in order to execute the supervised learning model.
* **Splitting Data**:
  + Divide the data into 3 parts: training, validation, and test sets. This helps assess our model's performance.
* **Choosing a Simple Model**:
  + Pick an easy-to-understand machine learning (ML) algorithm like *Naive Bayes* or *Logistic Regression*.
* **Training & Evaluating Model Performance:**
  + Use training data to teach model to recognize spam.
  + Assess how well our model does on the validation data. Look at metrics like accuracy and precision.
* **Fine-Tuning & Testing the Model**:
  + Adjust our model's settings (if needed) based on evaluation results to make it better at detecting spam.
* **Testing the Model**:
  + Check how well our model performs on the test data. This tells us how it might do perform in the real world.
* **Deploying the Model**:
  + If satisfied with the results, put our model into action, whether it's for email filtering or comment filtering.
* **Documenting the Project**:
  + Write a report or summarizing our project, including the steps we took and the results into GitHub.

***Flow Diagram :***

***Conclusion :***

The "AI-Powered Spam Classifier" project has delivered an effective spam filtering solution, enhancing communication while safeguarding user data. Its robust model, real-time integration, and privacy measures ensure a secure and efficient user experience, setting a standard for responsible AI deployment.

**Team Members:**

Harshavarthan. N, Anbuselvan. A, Rahul. S, Arul Prakash. A