**NAAN MUDHALVAN - PHASE 1 PROJECT SUBMISSION**

**Project 1: Building a Smarter AI-Powered Spam Classifier**

**TEAM MEMBERS:**

1. NALINKUMAR S - (2021504304)
2. DINESHKUMAR S - (2021504710)
3. PRAVEEN G R - (2021504307)
4. VENGADESHWARAN S - (2021504708)

**Problem Statement:**

 Develop an AI-powered spam classifier using natural language processing (NLP) and machine learning techniques to accurately distinguish between spam and non-spam messages in emails or text messages.

**AIM:**

The aim of this project is to develop an AI-powered spam classifier using Natural Language Processing (NLP) and machine learning techniques. This classifier will accurately distinguish between spam and non-spam messages in emails or text messages**.**

**Dataset Link:**

[**https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset**](https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset)

**PROJECT OBJECTIVE:**

The objective of this project is to develop a highly accurate AI-powered spam classifier using Natural Language Processing (NLP) and machine learning techniques. The aim is to reliably distinguish between spam and non-spam messages in emails or text messages, with a focus on minimizing false positives and false negatives while maintaining a high level of overall accuracy.

**Project Steps**

**Phase 1: Problem Definition and Design Thinking**

In this phase, we'll define the problem statement, understand the requirements, and plan the design of our AI-powered spam classifier.

**Problem Definition:**

 The problem is to build an AI-powered spam classifier that can accurately distinguish between spam and non-spam messages in emails or text messages. The goal is to reduce the number of false positives (classifying legitimate messages as spam) and false negatives (missing actual spam messages) while achieving a high level of accuracy.

**Design Thinking:**

1. Data Collection: We will need a dataset containing labeled examples of spam and nonspam messages. We can use a Kaggle dataset for this purpose.
2. Data Preprocessing: The text data needs to be cleaned and preprocessed. This involves removing special characters, converting text to lowercase, and tokenizing the text into individual words.
3. Feature Extraction: We will convert the tokenized words into numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency).
4. Model Selection: We can experiment with various machine learning algorithms such as Naive Bayes, Support Vector Machines, and more advanced techniques like deep learning using neural networks.
5. Evaluation: We will measure the model's performance using metrics like accuracy, precision, recall, and F1-score.
6. Iterative Improvement: We will fine-tune the model and experiment with hyperparameters to improve its accuracy.

**Phase 2: Innovation**

In this phase, we'll explore innovative techniques and approaches to building our spam classifier.

One innovative technique we can explore is using pre-trained language models like BERT for feature extraction. These models have demonstrated superior performance in NLP tasks.

**Phase 3: Development Part 1**

In this phase, we'll start building the core components of our spam classifier.

**Phase 4: Development Part 2**

In this phase, we'll continue building our spam classifier by selecting a machine learning algorithm, training the model, and evaluating its performance.

**Phase 5: Project Documentation & Submission**

In this phase, we'll document the entire project and prepare it for submission

**Documentation**

* Clearly outline the problem statement, design thinking process, and the phases of development.
* Describe the dataset used, data preprocessing steps, and feature extraction techniques.
* Explain the choice of machine learning algorithm, model training, and evaluation metrics.
* Document any innovative techniques or approaches used during the development.

**Submission**

* Compile all the code files, including the data preprocessing, model training, and evaluation steps.
* Provide a well-structured README file that explains how to run the code and any dependencies.
* Include the dataset source and a brief description.
* Share the submission on platforms like GitHub or personal portfolio for others to access and review.

**CONCLUSION:**

The project outlines a structured approach to building an AI-powered spam classifier using NLP and machine learning. Leveraging the SMS Spam Collection Dataset, it emphasizes meticulous data preprocessing and feature extraction. Innovative techniques like pre-trained models are considered, and development is divided into focused phases. Comprehensive documentation ensures accessibility, while platform submission encourages community engagement. This project exhibits a thoughtful and pragmatic strategy for effective spam message classification.