**CA1 DATA SCIENCE AND**

**MACHINE LEARNING PORTFOLIO LOG**

**Date: February 28th 2025**

**Prepared by:**

Kirubel Temesgen

Student Number: C00260396

**Course:** Data Science and Machine Learning

**Supervisor:** Greg Doyle

## CRISP-DM Phase 1: Business Understanding

### Project Overview

This project focuses on implementing and evaluating three machine learning classification models: **Naïve Bayes, Support Vector Machines (SVM), and K-Nearest Neighbors (KNN)**. The goal is to apply these models to a large-scale dataset (**Spam Detection Dataset**) and assess their performance in a real-world classification problem. The project will follow the **CRISP-DM framework** to structure the workflow effectively, ensuring a clear and detailed documentation process.

### Business Objective

The objective of this project is to determine which machine learning model—Naïve Bayes, SVM, or KNN—performs best in classifying text-based messages as **spam or not spam**. This is a common problem in the domain of natural language processing (NLP) and cybersecurity, as spam detection is critical for filtering unwanted and potentially harmful messages in emails, SMS, and social media platforms.

By implementing and modifying each model, we aim to:

* **Understand** the theoretical and practical aspects of each classification algorithm.
* **Evaluate** the strengths and weaknesses of Naïve Bayes, SVM, and KNN in a real-world dataset.
* **Optimise** model performance through parameter tuning and preprocessing adjustments.
* **Document** findings, modifications, and performance metrics systematically.

### Key Questions to Address

1. **How well does each model perform in classifying spam messages?**
2. **Which model is the most efficient in terms of speed and accuracy?**
3. **What preprocessing techniques improve classification performance?**
4. **How do model hyperparameters impact results?**
5. **What practical recommendations can be made based on findings?**

### Success Criteria

The success of this project will be measured by:

* **Classification Accuracy:** Evaluating the percentage of correctly classified messages.
* **Precision & Recall:** Measuring the trade-off between false positives and false negatives.
* **F1-Score:** Assessing the overall balance of precision and recall.
* **Computational Efficiency:** Comparing the time taken to train and predict with each model.
* **Generalisability:** Ensuring the model performs well on unseen data.

### Constraints & Assumptions

* The dataset is assumed to be clean and well-labeled (spam vs. non-spam).
* Feature extraction (e.g., TF-IDF or Count Vectorization) will be applied for text-based classification.
* Performance metrics will be evaluated using **cross-validation** to ensure reliability.
* Adjustments to model parameters will be logged systematically.

## CRISP-DM Phase 2: Data Understanding

### 1. Data Collection

The dataset used for this project is the Enron Email Dataset, which consists of over 500,000 email messages. This dataset is widely used for spam detection research and contains a mix of spam and legitimate emails. The dataset is stored as a CSV file named emails.csv and includes the following key columns:

* file: The filename associated with each email.
* message: The complete email content, including metadata such as From, To, Subject, and the email body.

**File Location:**

* Dataset: D:\college\Sem2\DataSci ML\archive\emails.csv

### 2. Data Description

**Initial Data Exploration** The dataset consists of **517,401** emails with two primary columns: file and message. Additional extracted features include:

* From: The sender of the email.
* To: The recipient(s) of the email.
* Subject: The subject line of the email.
* Body: The main content of the email.

**Data Types:**

* file: Object (String)
* message: Object (String)
* From: Object (String)
* To: Object (String, with missing values)
* Subject: Object (String)
* Body: Object (String)

**Missing Values:**

* The To column has **8,929** missing values, but all other fields are fully populated.

**Duplicates:**

* No duplicate entries were found in the dataset.

### 3. Exploratory Data Analysis (EDA)

**Sender Analysis** A bar chart was created to visualize the top email senders. The most frequent senders included key Enron employees, such as kay.mann@enron.com, vince.kaminski@enron.com, and jeff.dasovich@enron.com, each contributing thousands of emails to the dataset.

**Subject Line Analysis** The most common words in subject lines were extracted and visualised. The top words included:

* **Re:**, **RE:** (indicating replies)
* **FW:** (indicating forwarded messages)
* Business-related terms such as **Enron**, **Meeting**, **Agreement**, and **Energy**.

## CRISP-DM Phase 3: Data Preparation

### Objective

In this phase, we will clean, preprocess, and prepare the dataset for training machine learning models. The key steps include handling missing values, extracting relevant features, and converting text data into a numerical format.

### Step 1: Handling Missing Values

* The To column has **8,929 missing values**.
* Since this column is **not useful for spam classification**, it will be dropped.

### Step 2: Extracting Relevant Features

For spam classification, we extract the following key features:

1. **Email Subject** (Subject column)
2. **Email Body** (Body column)
3. **Email Sender** (From column) - Some spam emails may come from suspicious addresses.
4. **Word Count in Subject & Body** - Spam emails tend to be shorter or longer than non-spam emails.

### Step 3: Labeling the Data

Since the dataset does not explicitly label emails as spam or non-spam, we will infer spam labels using:

* **Keyword-based filtering:** Emails containing words like "lottery", "win money", "free offer", "click here" in the subject or body may be considered spam.
* **Domain-based filtering:** Emails from **non-corporate domains** (not ending in @enron.com) could be potential spam.

### Labeling Rule (Binary Classification)

* **Spam (1):** If the subject/body contains common spam keywords OR if the sender domain is not @enron.com.
* **Not Spam (0):** Otherwise.

### Step 4: Feature Engineering (Text Vectorisation)

Since machine learning models cannot process raw text, we convert it into numerical format using **TF-IDF (Term Frequency-Inverse Document Frequency)**.

* **TF-IDF Vectoriser:** Converts words into numerical features based on their frequency in the dataset.
* **Stopwords Removal:** Removes common words like "the", "and", "is" that do not add value.
* **N-grams:** Uses **bigrams** (two-word combinations) to improve context understanding.

### Step 5: Finalising the Processed Dataset

We will prepare the final dataset with the following columns:

* Processed\_Subject – Cleaned subject text
* Processed\_Body – Cleaned body text
* Sender\_Domain – Extracted from the From column
* Word\_Count\_Subject – Total words in the subject
* Word\_Count\_Body – Total words in the body
* Spam\_Label – 1 (Spam) or 0 (Not Spam)