November 3, 2023

C964: Computer Science Capstone

Jeremiah A. Lairson

*Task 2 parts A, B, C and D*

[Part A: Letter of Transmittal 2](#_Toc150086981)

[*Letter* 2](#_Toc150086982)

[Part B: Project Proposal Plan 3](#_Toc150086983)

[*Project Summary* 3](#_Toc150086984)

[*Data Summary* 5](#_Toc150086985)

[*Implementation* 5](#_Toc150086986)

[*Timeline* 7](#_Toc150086987)

[*Evaluation Plan* 7](#_Toc150086988)

[*Resources and Costs* 8](#_Toc150086989)

[Part C: Application 9](#_Toc150086990)

[Part D: Post-implementation Report 10](#_Toc150086991)

[*Solution Summary* 10](#_Toc150086992)

[*Data Summary* 10](#_Toc150086993)

[*Machine Learning* 11](#_Toc150086997)

[*Validation* 11](#_Toc150086998)

[*Visualizations* 13](#_Toc150086999)

[*User Guide* 17](#_Toc150087000)

[Reference Page 18](#_Toc150087001)

# Part A: Letter of Transmittal

## *Letter*

November 3, 2023

Dr. Mark Winslow, CTO

Diplomat Technical Solutions

2133 Millennium Dr Ste A

Indianapolis, IN 46204

Dear Dr. Winslow,

I am pleased to present a comprehensive proposal aimed at addressing a critical concern that our organization faces – the efficient and accurate categorization of incoming emails. The email management system in our organization is currently challenged by an overwhelming volume of incoming emails that encompass a mixture of legitimate correspondence and unwanted, potentially harmful, spam emails.

Our proposed solution centers around the implementation of a Naive Bayes Classifier. This machine learning algorithm is specifically designed to discern, categorize, and filter incoming emails into distinct categories: legitimate messages and spam. It effectively distinguishes between these categories using modern natural language processing and machine learning techniques.

This proposed solution presents numerous advantages for our organization, including a remarkable enhancement in productivity by streamlining email processing. The system, by effectively filtering out spam emails, greatly reduces the time and resources spent on manual filtering, thereby allowing employees to focus on crucial tasks. Furthermore, by accurately identifying and flagging potential security threats and malicious content, the system significantly bolsters our organization's cybersecurity. We hypothesize that leveraging machine learning, particularly the Naive Bayes algorithm, will substantially improve email categorization.

The proposed implementation plan is outlined to ensure a seamless integration of the Naive Bayes Classifier into our existing infrastructure. It includes a detailed cost-benefit analysis, encompassing the financial commitment, estimated timeline, necessary data infrastructure, and ethical considerations. Moreover, the plan involves comprehensive training sessions to equip our team members with the necessary knowledge and skills for utilizing the classifier effectively.

As an experienced developer in the realm of machine learning and data analytics, I am confident in successfully executing this implementation. My background includes several successful projects of similar nature, and I am committed to ensuring a smooth transition and efficient operation of this solution within our organization.

The implementation of this solution will revolutionize our email management processes, enhance our security protocols, and optimize our organizational productivity. I look forward to having the opportunity to discuss this proposal in more detail and provide further insights on its potential benefits. It will require a budget in the range of seven hundred thousand to one million dollars, which includes costs for a machine learning team, necessary software, and cloud hosting. It is expected to save the company many times more over multiple years of operation.

I am excited about the transformative potential that this solution holds for our organization. Please do not hesitate to contact me to schedule a meeting or further discussions. Thank you for considering this proposal. I am eager to bring this solution to our organization and to work toward its successful implementation and operation.

Sincerely,



Jeremiah Lairson, Senior Software Engineer

# Part B: Project Proposal Plan

## *Project Summary*

The proposal aims to address a pressing organizational need related to spam emails. Employees are currently burdened with an overwhelming volume of spam emails that consume their time and potentially pose security risks. This impacts their productivity and overall satisfaction. By implementing a machine learning-based solution, the organization will seek to streamline email management and enhance email security, thus improving the working environment for employees.

Our organization has a strong history of embracing technology to improve internal processes. However, the context has changed in recent years. With the exponential rise of spam emails across the industry, our organization's employees have been grappling with the manual filtering of these emails. The need to improve email security due to evolving security threats in the email landscape has never been more critical.

For the purposes outlined thus far, we will implement a spam-filtering application utilizing naive Bayes classification. The proposed application is a machine learning-based solution designed to alleviate the organizational burden caused by an overwhelming influx of spam emails. Utilizing industry-standard machine learning algorithms, the application will intelligently analyze and categorize incoming emails into 'spam' and 'non-spam' categories, significantly reducing the volume of unwanted emails that currently consumes employees' time and compromises email security. The system is engineered to seamlessly integrate within the existing email infrastructure without disrupting day-to-day operations. The user guide, which accompanies the application, will provide necessary instructions, specifically tailored for the organization's middle management personnel. It will offer a step-by-step guide on how to access, use, and interpret the system's output without requiring extensive technical expertise. The guide will encompass detailed visuals, simple language, and practical examples to ensure ease of understanding and optimal utilization of the application within the organization.

The implementation of this machine learning-powered solution will yield multifaceted advantages for the client's operational framework. By systematically categorizing and curating incoming emails, the application will effectively reduce the deluge of spam emails, liberating employees from the exhaustive task of manual email sorting. This streamlined email management process will significantly enhance the organization's productivity, enabling employees to allocate their time more efficiently toward core responsibilities. Moreover, the application's heightened email security measures will mitigate the risk of potentially malicious content, safeguarding sensitive information and ensuring a more secure communication environment. Consequently, the reduction in the volume of spam and the augmented email security measures will culminate in a marked improvement in employees' job satisfaction, as their work becomes more focused, less disrupted, and better protected against emerging security threats.

## *Data Summary*

The data for the working application will be passed from the company email server before reaching the client.  
 For the purposes of the demo application, the data will be simulated using a test set of spam and non-spam emails collected from Appache’s public SpamAssassin datasets. This information will apply practically to the training purposes of the classification model because they offer a generally unbiased sample of real communications and the sort of language that appears commonly in spam.

There are no legal concerns pertaining to the data that will be used. All sensitive information has been cleaned before publishing and no identifying information is included.

## *Implementation*

Development will follow the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, including the following phases:

**• Business Understanding:**

In this phase, we will define the objectives of the project, which include enhancing email security and improving employee productivity. We will engage with stakeholders to understand their specific requirements and concerns related to email security and spam emails. This will help us set clear business goals and success criteria for the project.

**• Data Understanding:**

The data understanding phase involves data collection and an initial assessment of the dataset that will be used to train the Naive Bayes classification model. We will gather historical email data, including both spam and non-spam emails, to create a labeled dataset. The data will be inspected for quality, completeness, and relevance. Additionally, we will perform exploratory data analysis to gain insights into the characteristics of the emails and identify any patterns in spam emails.

**• Data Preparation:**

In the data preparation phase, we will preprocess and clean the collected data to make it suitable for machine learning. This includes handling missing data, addressing outliers, and transforming the text data into a format that can be used for model training. Additionally, we will conduct data sampling or balancing if necessary to ensure the model is not biased towards the majority class (non-spam emails).

**• Modelling:**

The modeling phase is where we develop and train the Naive Bayes classification model. We will split the preprocessed data into training and testing sets. The Naive Bayes algorithm will be implemented to build the classification model. Hyperparameter tuning and cross-validation will be performed to optimize the model's performance. The trained model will be assessed for accuracy and efficiency.

**• Evaluation:**

In the evaluation phase, the model's performance will be rigorously assessed. We will use various evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) to measure how well the model can distinguish between spam and non-spam emails. The model will be validated using a hold-out dataset that was not seen during training. If the model doesn't meet predefined performance criteria, it will be fine-tuned and re-evaluated.

**• Deployment:**

Once the Naive Bayes classification model demonstrates satisfactory performance, it will be integrated into the organization's email system. During deployment, we will ensure a seamless transition to the new system. This may include configuring real-time email classification and user training to adapt to the new system. Continuous monitoring and updates will be part of the deployment phase to keep the model effective in identifying evolving spam patterns.

01/01/2024 – Proposal decision (accepted or denied)

The project's journey begins with a crucial decision on whether to proceed with the proposal. This stage involves the initial evaluation of the project's viability and alignment with organizational objectives.

02/01/2024 – Presentation of technical proof of concept

Following approval, the project enters the development phase. By this date, the team will have created a technical proof of concept to demonstrate the feasibility and viability of the Naive Bayes spam filter.

04/01/2024 – Submission of working model for review

Building on the proof of concept, this milestone marks the submission of a functional spam classification model for review. The model is expected to demonstrate its capability to identify spam emails with precision.

06/01/2024 – Deployment and testing

In this critical phase, the spam filter is integrated into the organization's email system and subjected to rigorous testing. It ensures that the model seamlessly operates in the real-world email environment and effectively filters spam.

## *Timeline*

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected start date | Anticipated end date |
| project initiation, requirements, and planning | 01/01/2024 | 01/31/2024 | Project kickoff and team assembly.  Requirement gathering and analysis.  Initial project planning and task assignment. |
| technical proof of concept | 02/01/2024 | 03/30/2024 | Implementation of the Naive Bayes classification algorithm.  Development of a basic prototype for demonstration.  Initial tests to validate algorithm functionality. |
| model refinement and data preparation | 04/01/2024 | 05/31/2024 | Fine-tuning the Naive Bayes model with real data.  Preprocessing and formatting of email data.  Comprehensive testing and validation of the model. |
| deployment and testing | 06/01/2024 | 08/31/2024 | Integration of the spam filter into the organization's email system.  Real-world testing with a subset of users.  Feedback collection and fine-tuning. |

## *Evaluation Plan*

In the development of the application, a series of verification and validation methods will be deployed to ensure its accuracy, robustness, and user acceptance. Verification activities will involve comprehensive checks to confirm the accuracy of the model. Through code reviews, inspections, and rigorous testing at multiple levels, including unit and integration testing, the model's accuracy in spam detection will be verified, aiming to achieve at least 95% accuracy. For robustness, continuous evaluations will be performed under diverse conditions and varied datasets to guarantee consistent performance, even in the presence of noise or unexpected data. On the other hand, validation efforts will focus on user satisfaction and system acceptance. User feedback, gathered through alpha and beta testing, will play a vital role in understanding user acceptance, ensuring general satisfaction with the system. These combined verification and validation procedures aim to create a software product that not only meets the technical specifications but also aligns with end-users' needs and expectations, ensuring a reliable and user-accepted solution.

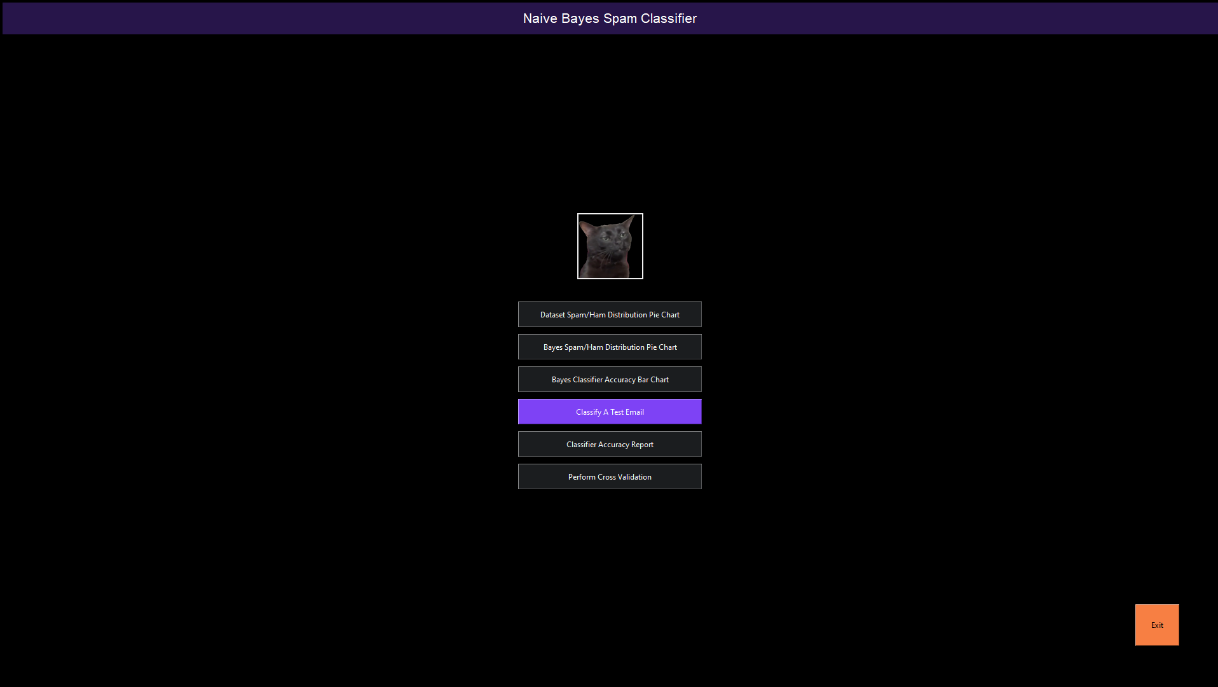
|  |  |
| --- | --- |
| **Objective** | **Success Criteria** |
| Accuracy | The model should maintain at least 95% accuracy in spam detection |
| Robustness | The model must maintain consistent performance and make accurate predictions under varying conditions and in the presence of statistical noise or unexpected data |
| User Acceptance | User feedback should suggest general satisfaction with the new system |

## *Resources and Costs*

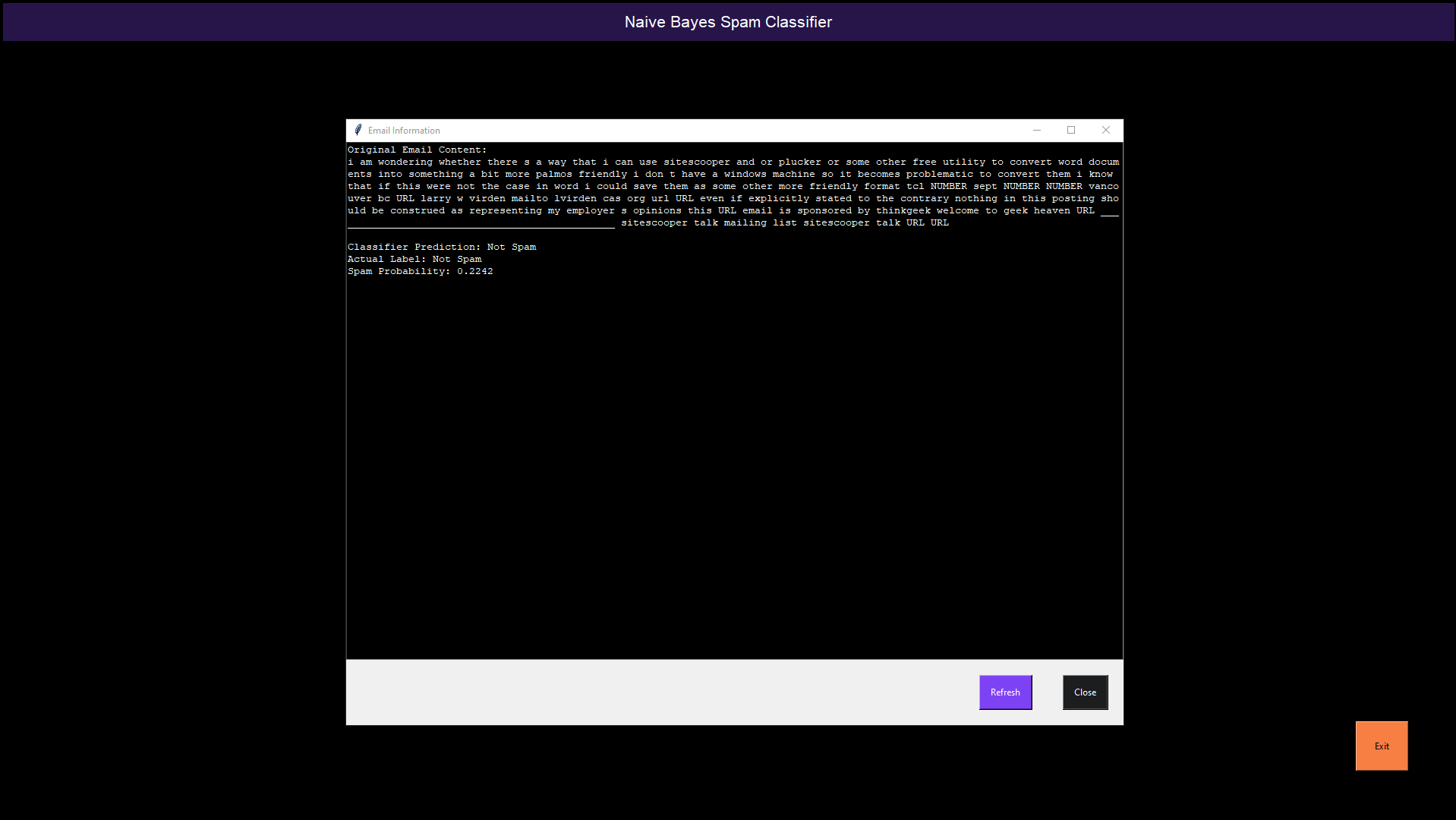
|  |  |  |
| --- | --- | --- |
| **Resource** | **Description** | **Cost** |
| Data and Machine Learning Team | A selection of machine learning engineers and data scientists is required to assemble the working model, as well as some use of system administrators. | Estimated $700,000 |
| Software Requirements | Python, Scikit-Learn, Jupyter Notebook, and some email integration API are needed. | The email integration API cost will be included in the team cost estimate. |
| AWS Flexible Cloud Hosting Platform | The hosting and hardware infrastructure will be offloaded to an AWS EC2 instance to maximize usage to cost efficiency. | Estimated $10,000 |
|  | **Total** | Estimated $710,000 |

A Panopto-hosted video of the application running in a virtual environment can be found [here](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cd23a3ae-9f46-43c5-b5f4-b0b20171d659).

# Part C: Application



*Screenshot of the main GUI window*

**

*Screenshot of the classifier pane. This window tests a random email from the 30% test dataset using the multinomial Naive Bayes classifier function.*

# Part D: Post-implementation Report

## *Solution Summary*

The project was developed to address a prevalent issue of email spam inundation impacting organizational productivity and posing potential security risks. With a continuous influx of spam emails, the manual filtering process impeded the effectiveness of employees. The implemented solution involved the creation of a machine learning-based spam classification system. Leveraging the Naive Bayes classifier and k-fold cross-validation techniques, the model effectively discerns between spam and legitimate emails, offering an automated mechanism to segregate incoming emails, thereby streamlining the email management process. This solution aimed to significantly reduce the burden on employees while concurrently enhancing email security within the organizational domain.

The initial problem revolved around the overwhelming volume of spam emails affecting employee productivity and potentially posing security vulnerabilities. The solution involved implementing a machine learning model to classify emails as spam or not spam, thereby mitigating the manual effort required for sorting emails. Using a dataset consisting of email contents and their respective labels (0 for non-spam and 1 for spam), the model was trained using the Naive Bayes classifier. This classification model has undergone intensive evaluation through k-fold cross-validation, as well as a number of other accuracy testing metrics, to ensure its effectiveness and generalizability. By effectively distinguishing between spam and non-spam emails, the application provides an automated mechanism to filter incoming emails, significantly reducing the time and effort employees spend on manual email sorting. This not only enhances productivity but also fortifies email security by identifying and isolating potentially harmful content.

## *Data Summary*

## The [dataset](https://www.kaggle.com/datasets/ozlerhakan/spam-or-not-spam-dataset) was provided by Hakan Osler on Kaggle, as a compilation of select public data published by Apache from their SpamAssassin datasets.

## It perfectly fit the usage of the application because it is a collection of spam and non-spam emails. The application categorizes between spam and non-spam emails. There is no closer association between the practical usage of the application and the training and testing datasets.

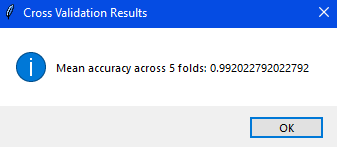
## The data processing for this project entailed a supervised learning approach to building a spam classifier using the Multinomial Naive Bayes algorithm. Initially, the dataset's 'email' column, containing email content, was inspected to handle any missing values by replacing them with empty strings. Following this, the vectorizer was applied to transform the textual email content into a better-suited format for machine learning. This transformation involves computing Term Frequency-Inverse Document Frequency (TF-IDF) scores, enabling feature extraction from the text. Subsequently, the dataset was split into training and test sets, with 70% allocated for training and 30% for testing. Because the dataset was skewed with a much higher content of non-spam emails, it was necessary to oversample the minority class of spam messages to improve classifier accuracy. The classifier was then initialized and trained using the training dataset to learn the relationships between the email features and their corresponding labels. Finally, the trained classifier was utilized to generate predictions on the test dataset, enabling an assessment of its predictive accuracy and effectiveness in spam detection.

## *Machine Learning*

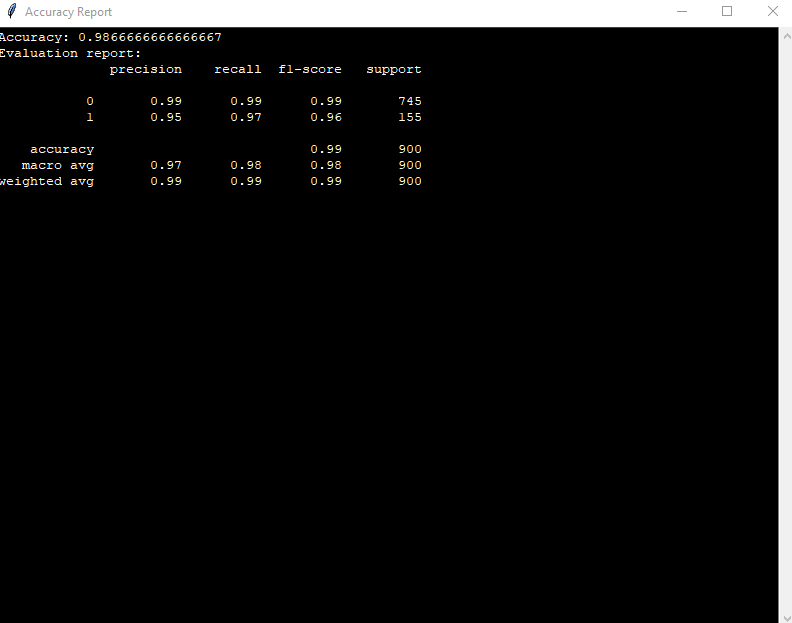
Naive Bayes spam filtering is an effective text classification technique used in email filtering to distinguish between spam and legitimate (ham) emails. What sets Naive Bayes apart is its simplicity and efficiency in computation, which makes it a popular choice for spam detection. The method applies Bayes' theorem along with the 'naive' assumption of independence among features to calculate the probability that an email belongs to a particular category based on its contents. Developed by incorporating statistical learning principles, Naive Bayes analyzes the frequency of words within emails, building a probabilistic model from labeled training data. Despite its assumption of feature independence, Naive Bayes showcases robust performance in practice. The selection of Naive Bayes was justified by its ability to handle high-dimensional data, its ease of implementation, and its history of accurate and rapid classification in various text-based applications, including spam filtering. Additionally, this method requires relatively little training data, making it an efficient choice for the development of a spam filter.

## *Validation*

For validating the Naive Bayes classifier used for spam detection, the chosen approach is k-fold cross-validation. This method partitions the dataset into k equal-sized subsets, where each subset acts as a test set exactly once, and the remaining data as training sets. This helps in evaluating the model's performance across various iterations, mitigating issues related to the dataset's division into distinct train-test sets. By averaging the performance metrics obtained from each fold, it provides a robust assessment of the model's generalization capability. The plan involves employing k-fold cross-validation to assess the Naive Bayes classifier's performance and generalize its effectiveness in handling spam detection. It will furnish reliable estimates of the model's accuracy, precision, recall, and F1 scores (term definitions on page 14) across multiple partitions, contributing to a more comprehensive evaluation of its performance (Brownlee, 2023).

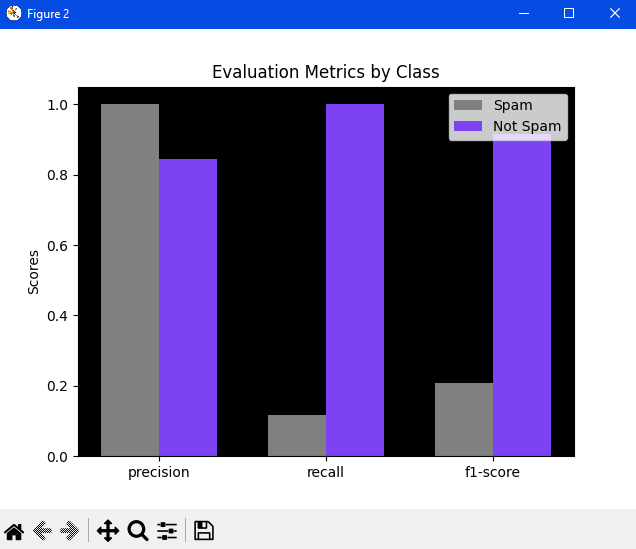


*K-fold cross-validation result in application interface*



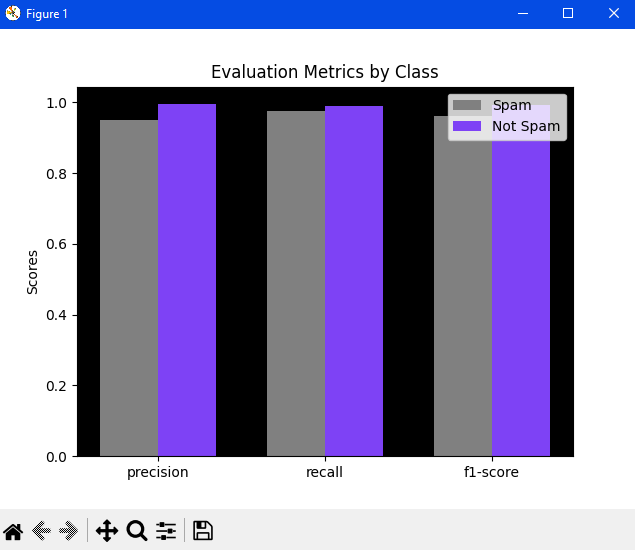
*Model accuracy report after minority class oversampling.*

## *Visualizations*



*Evaluation of classification criteria for each category of email.*

*(This screenshot was taken before applying minority oversampling)*

******

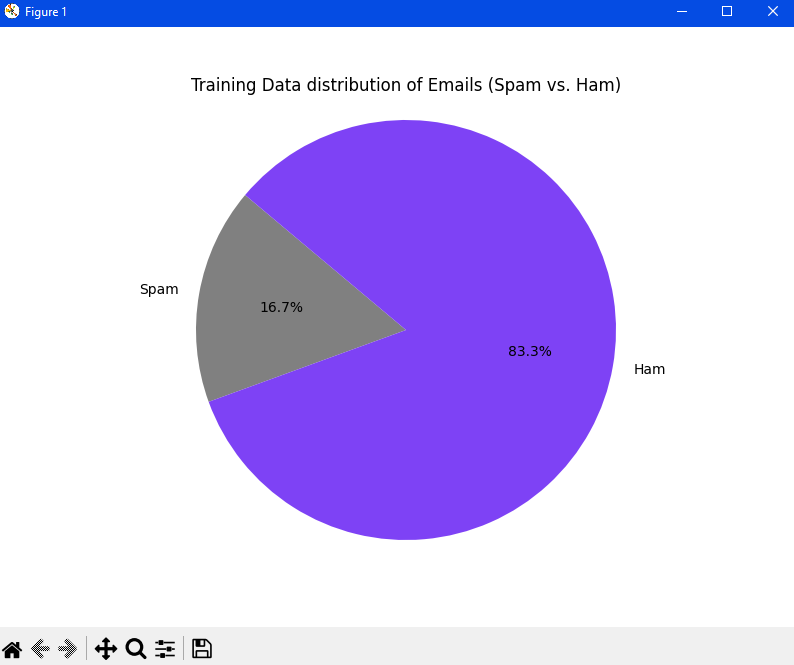
*Classification evaluation after applying minority oversampling shows significantly less disparity in accuracy metrics.*

***Precision*** *quantifies the model's ability to correctly classify emails as spam among those it predicted as spam. It helps identify false positives.*

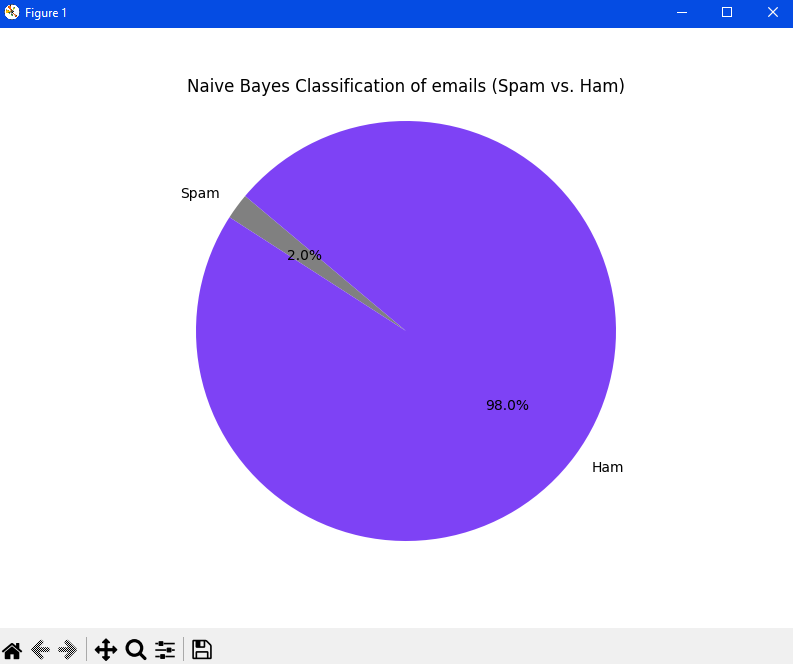
***Recall*** *measures the model's capacity to correctly classify all actual spam emails. It assists in identifying false negatives.*

*The* ***F1 score*** *combines precision and recall into a single metric, providing a balanced assessment of the model's performance.*

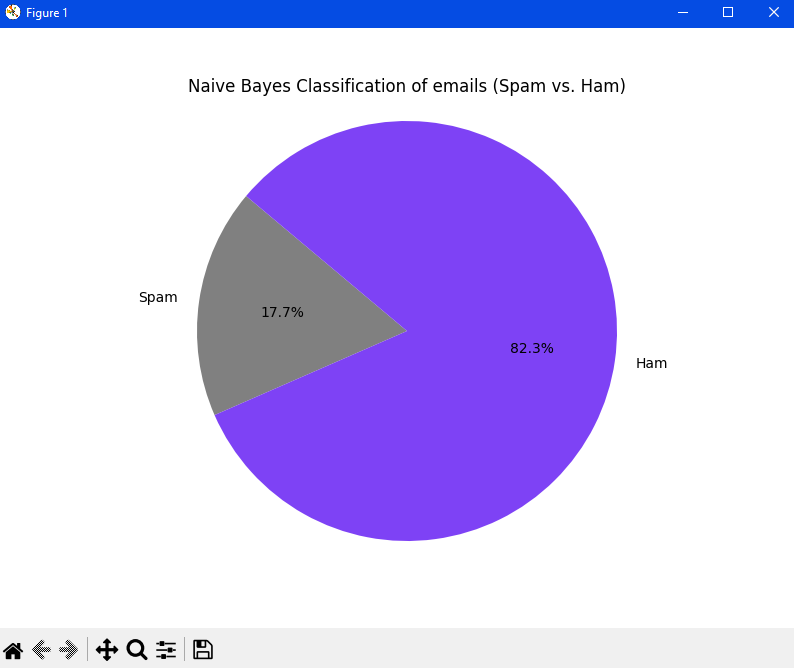
(Scikit-Learn Developers, 2023)



*Pie chart displaying the distribution of spam vs. non-spam emails in the training dataset.*



*A pie chart showing the distribution of spam vs. non-spam emails after classification.  
This graphic was generated before incorporating oversampling and clearly shows the bias toward classifying emails as non-spam due to their overwhelming presence in the dataset.*

**

*This pie chart of the Bayes classification results was generated after the inclusion of minority oversampling and overrepresents spam email content as compared with the original dataset, though ultimately it is greatly more accurate.*

## *User Guide*

1. [Download](https://www.anaconda.com/download) and install Anaconda. We will use this program to create a virtual environment.
2. Navigate to the folder containing the project files in the Anaconda terminal.
3. Create a new virtual environment with the command: “conda create --name NB”. The ‘NB’ in this case is the environment name and can be whatever you wish.
4. Activate the new virtual environment using the command: “conda activate NB” (or the name you chose)
5. Run the script with the command: “python NaiveBayes.py” The script will install the necessary dependencies before opening the program GUI.

[This video (from Part C)](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=cd23a3ae-9f46-43c5-b5f4-b0b20171d659) shows the necessary steps in Anaconda to run the application.

# Reference Page

# *Works Cited*

Brownlee, J. (2023, October 4). *A Gentle Introduction to k-fold Cross-Validation*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/k-fold-cross-validation/

Scikit-Learn Developers. (2023). *sklearn.metrics.f1\_score*. Retrieved from Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\_score.html

# *Selected Reading*

# *These articles aided in researching and composing the AI paper upon which this project was based.*

Graham, P. (20013, January). *Better Bayesian Filtering*. Retrieved from Paul Graham's Website: http://www.paulgraham.com/better.html

Graham, P. (2002, August). *A Plan For Spam*. Retrieved from Paul Graham's Website: http://www.paulgraham.com/spam.html

Kharwal, A. (2023, April 11). *Here’s How Naive Bayes Algorithm Works*. Retrieved from The Clever Programmer: https://thecleverprogrammer.com/2023/04/11/heres-how-naive-bayes-algorithm-works/

TURING. (2023). *An Introduction to Naive Bayes Algorithm for Beginners*. Retrieved from TURING: https://www.turing.com/kb/an-introduction-to-naive-bayes-algorithm-for-beginners