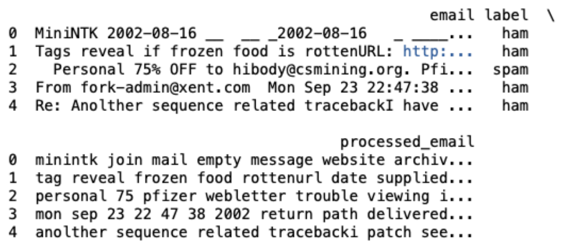
Spam Classification
Antonis Karaolis

Introduction:

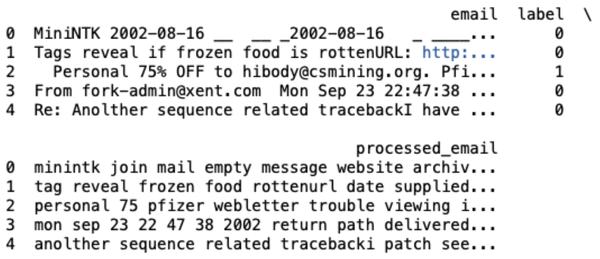
The problem of Spam detection is the issue at hand. It is a binary classification task with the objective of differentiating between spam and non-spam (ham) emails. This is an important job in the world of email services because it guards against users being inundated with unwanted or perhaps hazardous messages.
The project's initial phase included a number of preparation activities to get the data ready for modelling. The processes were concentrated on cleaning and standardizing the text data from the emails due to the fact that the data was text in nature.
The text was first edited to remove URLs, email addresses, and timestamps because they rarely offer context to the topic. To further purge the data, the text was tokenized, which means it was divided up into separate words or "tokens." To maintain uniformity and to treat words like "Email" and "email" as one and the same, these tokens were then changed to lowercase.
Stopwords, or frequent words like "the," "is," and "and," which don't reveal much about the substance, were subsequently eliminated. The tokens were then lemmatized, which is the process of reducing words to their simplest or root form (for example, "running" becomes "run").
The CountVectorizer was then used to convert the text data that had been processed into numerical form using a Bag of Words method. This procedure produced a sparse matrix, where each row corresponds to an email and each column to a distinct word from the entire corpus of text. The LabelEncoder also transformed the labels into binary number form, with 0 denoting "ham" and 1 denoting "spam." The tables below show the pre-processing procedure.



```
                                               email label  \
0  MiniNTK 2002-08-16 __  __ _2002-08-16   _ ____...   ham
1  Tags reveal if frozen food is rottenURL: http:...   ham
2    Personal 75% OFF to hibody@csmining.org. Pfi...  spam
3  From fork-admin@xent.com  Mon Sep 23 22:47:38 ...   ham
4  Re: Anolther sequence related tracebackI have ...   ham

                                     processed_email
0  minintk join mail empty message website archiv...
1  tag reveal frozen food rottenurl date supplied...
2  personal 75 pfizer webletter trouble viewing i...
3  mon sep 23 22 47 38 2002 return path delivered...
4  anolther sequence related tracebacki patch see...
```

Table 1: Data 'email' before and after pre-processing



```
                                               email  label  \
0  MiniNTK 2002-08-16 __  __ _2002-08-16   _ ____...      0
1  Tags reveal if frozen food is rottenURL: http:...      0
2    Personal 75% OFF to hibody@csmining.org. Pfi...      1
3  From fork-admin@xent.com  Mon Sep 23 22:47:38 ...      0
4  Re: Anolther sequence related tracebackI have ...      0

                                     processed_email
0  minintk join mail empty message website archiv...
1  tag reveal frozen food rottenurl date supplied...
2  personal 75 pfizer webletter trouble viewing i...
3  mon sep 23 22 47 38 2002 return path delivered...
4  anolther sequence related tracebacki patch see...
```

Table 2: Data after Label Encoder

Two models, Multinomial Naive Bayes (MNB) and Support Vector Machines (SVM), were chosen for the initial model experimentation. These models were selected due to their well-established efficacy in text classification issues. SVMs in particular have a reputation for being able to locate difficult decision boundaries and perform well on high-dimensional input, such as text. On the other hand, because of their simplicity and effectiveness, Naive Bayes classifiers, particularly the multinomial variation, serve as a standard starting point for text classification issues.

Model Training and Evaluation:

The following stage involved training and evaluating the models after data preparation and initial model setup. This phase's objective was to evaluate the models' capacity for generalization and precise classification of brand-new, unread emails.
The processed emails' (features) and labels' inputs were used to train the models using the fit function. Using the predict function, the models were utilized to generate predictions on the test data after training.
Precision, recall, F1-score, and confusion matrix were used as essential metrics to assess the models' performance. The model's capacity to prevent false positive mistakes is indicated by precision, which is the percentage of true positive predictions among all positive predictions. Recall gauges how well a model can identify positive occurrences by calculating the percentage of genuine positive predictions among all real positives. The F1-score, which offers a balanced measurement that takes into account both measures, is the harmonic mean of precision and recall.
The confusion matrix was also employed to provide a more thorough analysis of the models' performance. It provides information on the types of errors the models are producing by displaying the counts of true positive, true negative, false positive, and false negative forecasts.
The table below presents a comparison of the SVM and MNB models' performance on these metrics:

	SVM	**MNB**
Precision	0.98	0.91
Recall	0.95	0.92
F1-Score	0.97	0.92
Accuracy	0.98	0.96

Table 3: SVM vs MNB performance

Additionally, a more detailed understanding of the model performances can be seen in the plots of the confusion matrices for both models. Both models have a high proportion of true positives and true negatives, demonstrating their efficacy in correctly categorizing emails as either spam or non-spam.

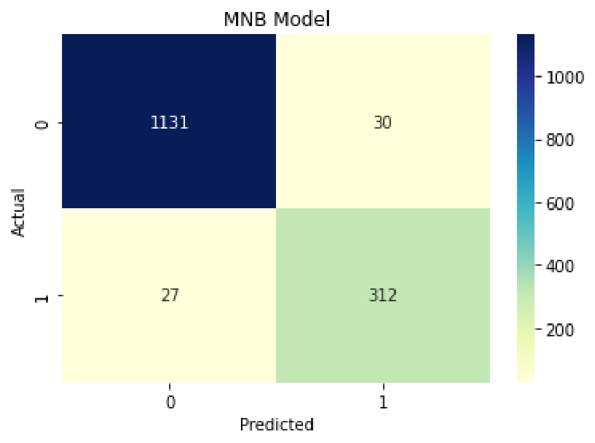


Figure 1: MNB Model Confusion Matrix

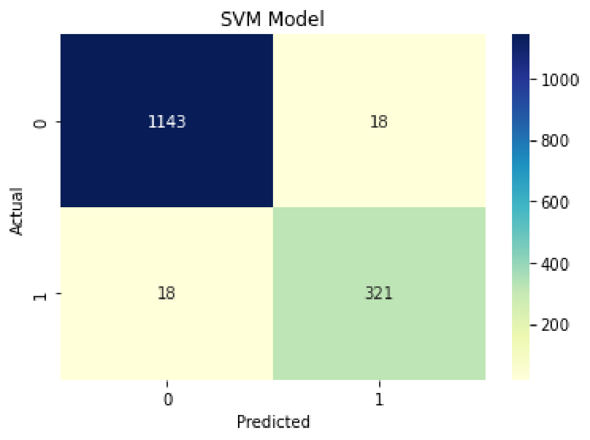


Figure 2: SVM Model Confusion Matrix

The results from the evaluation phase led to a crucial decision point in the project, the choice of the final model, SVM model.

<u>Observations and Model Selection:</u>

The model evaluation shed important light on the effectiveness of both the SVM and MNB models. When it came to classifying emails as spam or not, it was clear that both models were able to grasp the underlying patterns of the data, yielding outstanding results. However, a closer examination of the performance parameters revealed some minute variations. Across the board, the SVM model outperformed the MNB model in terms of accuracy, precision, recall, and F1-score. This shows that the SVM model was better able to predict both groups accurately and made fewer mistakes overall.
The SVM model's capacity to handle high-dimensional data and identify complicated decision boundaries may be one reason for its greater performance. Given that text data frequently has high dimensionality, particularly after pre-processing, the SVM's natural ability to handle such data may have produced better outcomes.
The MNB model, on the other hand, did well, but its core premise of feature independence may not hold true for text data, where words can be quite context-dependent. This could account for why the SVM model, which does not require this supposition, was able to better reflect the intricacy of the data.
Taking into account these findings, the SVM model was chosen as the project's ultimate classifier. The model's excellent performance on a number of criteria and its capacity to manage the complexities of high-dimensional text data were key factors in the decision. Due to the SVM model's demonstrated success on this dataset, there is a high degree of trust in its forecasts for unread emails in the future.
Figure 3 below shows the SVM model distribution. As mentioned before, ‘1’ is spam and ‘0’ ham.

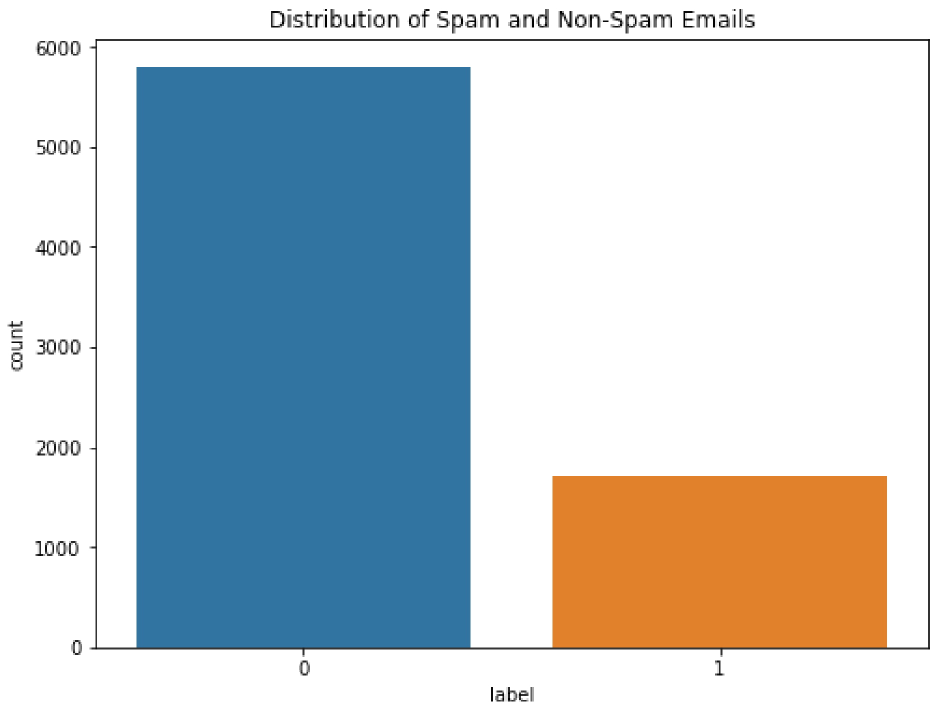


Figure 3: SVM Model Distribution

Model Optimization and Conclusion:

The SVM model was tuned for best performance because it was chosen as the final classifier, hence this was done. A few steps were included in the model optimization process, such as hyperparameter tuning and modifying class weights to take class imbalance into consideration.
Hyperparameter tuning was done using methods like grid search, which iteratively tests a variety of parameter combinations while cross-validating the results to see which tune provides the best performance. The main goal was to maximize the 'C' parameter, which regulates the cost of misclassification on the training data, and the 'gamma' parameter, which establishes the extent of each individual training example's influence.
Given the class imbalance in the dataset, adjusting the class weights was another crucial step. To rectify the imbalance, the SVM model was set up to impose a greater penalty for misclassifying the minority class.
There were difficulties despite the thorough procedure. One of the main ones was computational expense because both grid search and SVM require a lot of processing, especially when working with huge datasets. But this was overcome with painstaking iterations and good resource management.
There are numerous potential uses for the final spam classifier. By eliminating the time spent sorting through undesired emails, it can increase productivity by filtering out spam emails. Additionally, it might be implemented into email systems to automatically classify emails as "spam" and "not-spam," enhancing user experience.
There is always room for development and inquiry in terms of future directions. It could be beneficial to experiment with various models, such as deep learning models, which have demonstrated encouraging results in text classification. Further research might be done on feature engineering. The features available now are only based on text content, but additional elements, such email metadata, may also provide important signals. It might also be investigated to see if alternate text preparation methods, such as the use of various word embeddings, can enhance model performance.
In conclusion, this spam classifier was created using a set of organized procedures, including pre-processing, model selection, training, evaluation, and optimization. It illustrated how to use SVM and MNB models to solve a text categorization problem while also revealing the models' advantages and disadvantages. The result is an optimized spam classifier ready for deployment and further enhancement.