**Email Spam Detection Using Machine Learning**

**By:**

**Charles Ohene Apaw**

**Date: 02/11/2024**

**Table of Contents**

 **Executive Summary**

* Overview of Project Goals and Achievements

 **Introduction**

* Background
* Business Question
* Industry Context

 **Problem Statement**

* Challenges Addressed by the Project

 **Stakeholders**

* Primary and Secondary Stakeholders

 **Data Collection and Preparation**

* Data Source and Structure
* Preprocessing Steps
* Train-Test Split Details

 **Exploratory Data Analysis (EDA)**

* Key Findings
* Visualizations
* Challenges

.**Unsupervised Modeling and Feature scaling.**

**. Train -Test Split**

 **Modeling Approach**

* Selected Models
* Model Selection Rationale
* Hyperparameters and Optimization

 **Evaluation Metrics**

* Metrics Used and Their Relevance

 **Results and Impact**

* Model Performance Summary
* Business Impact Analysis

 **Error Analysis**

* Summary of Common Errors
* Recommendations and Insights

 **Deployment and Practical Applications**

* Deployment Plan
* Challenges and Solutions

 **Future Work and Improvements**

* Suggestions for Enhancing the Model
* Scalability and Real-world Adaptation

 **Conclusion**

* Summary of Findings
* Reinforcement of Business Benefits

 **References**

* Cited Data Sources and Literature

 **Appendix**

* Supplementary Data and Visualizations

**. Executive Summary**

This report outlines a comprehensive analysis and implementation of a machine learning model for email spam detection. The project aimed to develop a predictive model to improve productivity and security by accurately classifying emails as spam or not spam. The project resulted in a reliable spam detection system using Logistic Regression, Balanced Random Forest, and XGBoost, achieving an accuracy of over 97% with XGBoost. The deployment considerations and future work are also discussed to ensure long-term effectiveness and adaptability.

**Introduction**

* **Business Question**: How can we accurately detect and classify spam emails to improve organizational productivity and security?
* **Background**: With the exponential growth of digital communication, spam emails have become a significant burden, leading to productivity loss, security risks, and increased operational costs. Research shows that businesses can lose $20.5 billion annually due to spam emails, with an average loss of $1,934 per employee.
* **Industry Context**: This project is highly relevant in industries like finance and healthcare, where secure and efficient communication is vital for operations and compliance.

**Problem Statement**

* Spam emails significantly reduce productivity, pose security threats, and increase costs due to server load and IT maintenance. Addressing this issue with effective spam detection solutions can save companies both time and resources, ultimately protecting valuable data and reducing overhead expenses.

**. Stakeholders**

* **Primary Stakeholders**: IT and cybersecurity teams, company executives.

**Secondary Stakeholders**: Employees and customers who benefit from reduced phishing attempts and secure communication.

**Data Collection and Preparation**

* **Source**: The dataset was sourced from Kaggle and contained 5,172 emails, segmented into word frequency counts.
* **Data Structure**: The dataset had 3,002 columns, including:
  + One identifier column (Email No.) for privacy.
  + A Prediction column as the target variable (0 for non-spam, 1 for spam).
  + The remaining 3,000 columns represented the most common words used in the emails.
* **Preprocessing**:
  + **Duplicates**: Identified and removed 541 duplicate rows.
  + **Unnecessary Columns**: Dropped non-informative columns like Email No..
  + **Missing Values**: Confirmed that no missing values were present.
* **Train-Test Split**: The data was split into 80% training and 20% testing sets to ensure reliable evaluation while preventing overfitting.

#### **EXPLORATORY DATA ANALYSIS**

#### **Total Word Count Distribution Analysis**

**Overview:** The following histogram visualizes the total word count distribution across both spam and non-spam emails in the dataset.

A graph of a number of words

Description automatically generated

**Analysis:** The graph illustrates that:

* The majority of emails, regardless of classification, have a word count concentrated within the 0-1000 range.
* Non-spam emails generally show a slightly broader range in word counts, whereas spam emails display more clustering within lower counts.
* Both categories display a significant drop in frequency as the word count surpasses 3000.

**Key Insights:**

* **Feature Relevance**: The word count distribution supports the use of this feature in predictive modelling, as distinct patterns are observed between spam and non-spam emails.
* **Model Refinement**: Integrating word count as a feature can enhance the model’s ability to flag emails as spam or non-spam based on known patterns of content length.

### **Business Implications:** By understanding these distribution characteristics, organizations can bolster spam detection systems to mitigate productivity loss and protect employees from potentially harmful email content.

### **Challenges in Visualizing the Top 20 Words in Emails**

During the exploratory data analysis (EDA) phase, one of the tasks was to visualize the most frequently occurring words in spam and non-spam emails. This analysis was crucial for understanding the linguistic patterns and characteristics that differentiate spam from legitimate emails. However, I encountered some challenges that required iterations and adjustments to overcome.

**Initial Challenge:**

Initially, when I created the histogram to visualize the most common words, I mistakenly generated a plot that displayed the frequency of individual letters rather than complete words (see Figure 1). This error occurred due to how the dataset was processed and the code implemented for token extraction.

Figure 1: Initial Visualization (Letters Instead of Words)

A red and blue graph

Description automatically generated

This visualization did not provide any meaningful insights relevant to distinguishing between spam and non-spam emails, as it focused on single letters like 'e', 't', 'a', rather than words that contribute context to the classification.

**Resolution:**

To address this issue, I refined the code to ensure that the histogram captured entire words instead of individual characters. This adjustment was necessary to extract practical and valuable insights for the business problem. The revised visualization successfully displayed the top 20 most common words for both spam and non-spam emails (see Figure 2).

Figure 2: Corrected Visualization (Top 20 Words in Spam and Non-Spam Emails)

A red and blue graph

Description automatically generated

With this correction, the updated visualization revealed important patterns, such as the frequent appearance of certain trigger words like "your" and "here" in spam emails, compared to more neutral or formal words like "thank" and "please" in non-spam emails.

**Boxplot of "money" Word Frequency in Spam vs. Non-Spam Emails**

A graph with numbers and lines

Description automatically generated with medium confidence

To further investigate the significance of individual words in determining the nature of an email (spam or non-spam), a boxplot was created to represent the frequency of the word *"money"* across the two categories. The plot highlights the following observations:

* **Insights from the Boxplot**:
  + The boxplot clearly shows that the word *"money"* appears more frequently in spam emails compared to non-spam emails.
  + The distribution in the non-spam category shows minimal to no occurrences of the word *"money"*, with most data points clustering around zero.
  + In the spam category, however, outliers indicate a few emails where the word appears multiple times, reinforcing the notion that words related to financial incentives or clickbait content are significant indicators of spam.
* **Relevance to the Business Problem**: This insight reinforces the model's ability to detect spam based on commonly used words that might signify unwanted or fraudulent content. Such findings can guide enhancements in spam filters, ensuring that emails containing these indicative words are flagged more accurately. This is critical for businesses aiming to protect their communications from spam and phishing attempts that could compromise data security.

**Challenges Faced**: During the analysis phase, a key challenge was identifying which words provided the most insight into the classification. Focusing on words like *"money"* was strategic as it aligns with typical spam content themes. This step validated that analysing specific word frequencies can enhance model interpretability and offer practical guidance for feature engineering.

**Pairplot Analysis of Selected Features**

A graph of a graph of a graph

Description automatically generated with medium confidence

The above pairplot visualizes the relationship between three selected features: **"free," "you," and "money,"** across the dataset, categorized by the prediction labels (0 for non-spam and 1 for spam). This plot offers a multidimensional view of how these key words interact with each other in the context of identifying spam emails.

* **Interpretation of Pair Relationships**:
  + The plot illustrates that the presence of the word "money" tends to be more common in spam emails (red dots), as shown in its distribution against other words.
  + The word "free" also shows a higher concentration in spam emails compared to non-spam emails, although it appears less frequent overall.
  + The term "you" is seen in both spam and non-spam emails but has distinct clustering patterns, indicating that its presence alone is not a definitive indicator of spam.
* **Insights**:
  + These visual patterns emphasize the predictive power of certain words in distinguishing spam from non-spam. Words such as "money" and "free" are strongly associated with spam, while "you" is common in both types but shows a differentiable spread.
  + The diagonal KDE plots highlight the density distribution for each feature, helping to identify where most data points are clustered.

This pairplot aids in understanding the correlations and scatter between different terms that contribute to classification decisions, enhancing model interpretability.

**K-Means Clustering Analysis with Feature Scaling**

**Objective and Purpose**: The primary aim of applying K-Means clustering was to identify patterns within the dataset and segment the emails into meaningful groups. This step was conducted to enhance the dataset's analysis and provide deeper insights into the structure of spam and non-spam emails.

**Preprocessing Step – Feature Scaling**: Before performing K-Means clustering, feature scaling was applied to the data. This step was crucial to ensure that all features contributed equally to the clustering process. Since K-Means is sensitive to the scale of data, normalization helped maintain a balanced contribution of each word frequency, preventing any disproportionately high influence from features with larger scales.

A screenshot of a computer screen

Description automatically generated

**Visual Interpretation**: The plot above represents the K-Means clustering outcome, visualized in a two-dimensional space using Principal Component Analysis (PCA). Each data point signifies an email, while the centroids (highlighted in red) indicate the center points around which the clusters are formed.

**Key Observations**:

1. Clusters illustrate groupings of emails with similar content characteristics.
2. The spread and density of data points provide insight into the variability and similarity of email types within the dataset.
3. Well-defined clusters suggest distinct patterns, potentially aiding in identifying characteristics typical of spam or non-spam.

**Significance for the Project**: Understanding the distribution and characteristics of these clusters helps inform the feature engineering process and contributes to refining the supervised learning model's classification approach. Identifying cluster patterns supports more effective differentiation between nuanced content variations.

**Challenges and Solutions**: Initial challenges included determining the optimal number of clusters and interpreting the dimensional data. Feature scaling and PCA were vital in addressing these challenges by standardizing the data and reducing complexity for better visualization and analysis.

A diagram of a person with scales and data

Description automatically generated

In preparation for model training and evaluation, the dataset was divided into training and testing subsets:

* **Training Set (80%)**: Used for fitting the models and learning the relationships between features and the target.
* **Testing Set (20%)**: Reserved for assessing the model’s generalization to new, unseen data.

This step ensures that the model's performance is not overly optimized on the training data and can be reliably tested for real-world applicability.

**Final Scaling**: To standardize the data before modelling, final scaling was applied. This involved transforming feature values to a uniform scale, enhancing model stability and performance, especially for algorithms sensitive to feature magnitudes.

### This combination of splitting and scaling helps maintain the integrity of the training process and prevents data leakage, ensuring that model evaluations are as accurate as possible.

### **Supervised Modeling Approach and Evaluation Metrics**

**Introduction to the Models**: For the task of spam detection, three supervised models were chosen:

1. **Logistic Regression with Class Weights**: Selected for its simplicity and interpretability. The class weights help handle the imbalanced dataset by assigning higher importance to the minority class (spam).
2. **Balanced Random Forest**: This ensemble method was chosen due to its ability to handle class imbalance effectively by undersampling the majority class. It is robust in reducing overfitting and improving performance across imbalanced datasets.
3. **XGBoost (Extreme Gradient Boosting)**: A powerful boosting algorithm known for its high accuracy and efficiency. XGBoost was selected for its proven effectiveness in handling complex patterns and feature interactions in data.

**Why These Models Were Used**:

* **Logistic Regression**: Provides a straightforward baseline for comparison and interprets coefficients as feature importances, which is valuable for understanding model decisions.
* **Balanced Random Forest**: Effective in handling class imbalance and provides robust performance through ensemble learning, which averages decisions from multiple trees.
* **XGBoost**: Delivers high predictive performance, especially useful in datasets with potential non-linear relationships and interactions between features.

**Evaluation Metrics and Their Significance**: To assess the model performance, the following evaluation metrics were used:

* **Accuracy**: The overall correctness of the model. Although useful, it may not be sufficient alone for imbalanced datasets.
* **Precision**: The proportion of true positive predictions among all positive predictions. Indicates how many selected items are relevant.
* **Recall (Sensitivity)**: The proportion of actual positives correctly identified. Critical for understanding the model's ability to detect spam.
* **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure when there is an uneven class distribution.
* **AUC (Area Under the ROC Curve)**: Represents the capability of the model to distinguish between classes. A higher AUC value indicates better performance in distinguishing spam from non-spam.

**Significance of Metrics**:

* **Precision** is vital when minimizing false positives is important, ensuring legitimate emails aren't flagged incorrectly.
* **Recall** is essential when maximizing true spam detection is the goal, ensuring most spam is caught.
* **F1-Score** provides a balance between precision and recall, suitable for evaluating models where class distribution is uneven.
* **AUC** offers insight into the model's performance over various threshold settings.

**Model Comparison and Performance Analysis**

A graph of different colored bars

Description automatically generated

The chart above compares the performance of three different models: **Logistic Regression with Class Weights**, **Balanced Random Forest**, and **XGBoost**, using four key evaluation metrics: **Accuracy**, **Precision**, **Recall**, and **F1-Score**.

**Analysis:**

* **Accuracy**: XGBoost achieved the highest accuracy, indicating that it correctly classified the most emails overall. This suggests its strong predictive ability across both spam and non-spam categories.
* **Precision**: The models show that XGBoost and Logistic Regression with Class Weights have similar precision scores, meaning that when they predicted an email as spam, it was correct most of the time.
* **Recall**: Both XGBoost and Logistic Regression had significantly higher recall than the Balanced Random Forest. High recall indicates that these models were effective at identifying most of the spam emails.
* **F1-Score**: This harmonic mean of precision and recall further reinforced that XGBoost and Logistic Regression provided the best balance between false positives and false negatives, with XGBoost slightly leading.

**Significance to the Project:**

The results show that XGBoost consistently outperformed the other models, making it the most reliable choice for accurate spam detection in this project. However, Logistic Regression with Class Weights also performed well, especially in handling class imbalances. This comparison is crucial in deciding which model should be deployed for real-world application to maintain high levels of spam filtering efficiency.

**Contrasting Model Predictions on Dummy Emails**

A screenshot of a computer screen

Description automatically generated

**Objective**: To evaluate the robustness and variability of the models' predictions using realistic, simulated email content.

**Initial Test**: In our initial evaluation, a sample email containing the phrase "Congratulations! You’ve won a $1000 gift card..." was tested. All models (Logistic Regression with Class Weights, Balanced Random Forest, and XGBoost) consistently predicted the email as spam with high confidence. This demonstrated the models' ability to accurately identify obvious spam content. The results underscored that these models were reliable for straightforward cases.

**Further Testing with Diverse Content**: To explore more nuanced cases, two new dummy emails were tested:

1. **Email Content**: "Important update: please review your recent activity..."
2. **Model Performance**:
   * Logistic Regression with Class Weights predicted "Not Spam" with 40.1% probability.
   * Balanced Random Forest predicted "Not Spam" with a higher probability of 58.0%.
   * XGBoost flagged it as "Spam" with a probability of 83.4%.

**Analysis**: The contrasting predictions for these emails highlight the following:

* **Strength in Ensemble**: While XGBoost identified the second email as spam, other models flagged it as non-spam, indicating that having a set of diverse models can provide different perspectives and catch more subtle cases. This is particularly useful when filtering emails that are not overtly spammy but contain potential spam characteristics.
* **Model Limitations**: The varied results show that even strong models can have difficulty with borderline cases or ambiguous language. Thus, incorporating an ensemble approach can be more effective in maintaining a robust spam detection system.
* **Business Impact**: Employing multiple models for email classification can enhance the accuracy and reduce the likelihood of false negatives or positives, contributing to a more reliable spam detection system and better decision-making processes in email management.

**Conclusion**: The experiment demonstrated that while all models performed well on clear spam, there was variability when faced with ambiguous content. This showcases the importance of model diversity in complex real-world applications.

**Error Analysis**

**Objective**: To identify and analyse the types of errors made by the models and provide insights into potential areas for improvement.

**Approach**: Error analysis was conducted by reviewing instances where models misclassified emails. This involved analysing false positives (legitimate emails marked as spam) and false negatives (spam emails marked as legitimate).

**Findings**:

1. **Common Misclassifications**:
   * **False Positives**: Instances where legitimate business communications containing promotional language (e.g., "offer" or "exclusive") were flagged as spam.
   * **False Negatives**: Certain spam emails with more sophisticated language or contextually neutral words that did not trigger spam indicators in the models.
2. **Sources of Model Errors**:
   * **Overlap in Language**: A key source of misclassification was the similarity in language between some spam and non-spam emails. This led to difficulties in distinguishing emails that had subtle indications of spam.
   * **Dataset Strength**: The dataset used primarily focused on word frequency, which may have limited the models' ability to fully comprehend the context or intent behind emails.
3. **Possible Data Leakage Concerns**:
   * During the analysis, it was observed that all three models predicted certain emails with unusually high confidence. While no explicit data leakage was identified, this warranted a review to ensure that feature engineering or preprocessing did not inadvertently leak information between training and testing phases.

**Recommendations for Future Work**:

* **Enhance Feature Set**: Introduce additional features, such as metadata analysis (e.g., sender information, subject line).
* **Contextual Analysis**: Incorporate NLP models capable of understanding the context of the content, which would better handle nuanced language.
* **Model Diversity**: Continue using a diverse set of models to leverage different strengths in detecting complex spam patterns.

### **Conclusion**: The error analysis revealed that while the models performed well overall, they struggled with borderline cases where language overlap was prominent. Addressing these limitations by enhancing feature engineering and incorporating more context-aware models will improve future iterations.

### **Results and Impact**

**Performance Metrics of Models**

The table below summarizes the performance metrics for the three supervised models evaluated: Logistic Regression with Class Weights, Balanced Random Forest, and XGBoost. The metrics considered include accuracy, precision, recall, F1-score, and AUC (Area Under the Curve).

A screenshot of a graph

Description automatically generated

**Key Insights:**

* **XGBoost** outperformed other models with the highest accuracy (97.2%) and F1-Score (96%). It also maintained a high recall of 99%, indicating strong sensitivity in identifying spam.
* **Balanced Random Forest** showed perfect recall (100%), which is beneficial for capturing all spam emails, but had lower precision, suggesting a higher rate of false positives.
* **Logistic Regression (CW)** balanced precision and recall effectively with an AUC of 0.97, making it reliable for consistent spam detection.

**Impact Explanation:** The implementation of the best-performing model, **XGBoost**, in an organizational setting offers significant practical benefits:

* **Cost Savings**: By integrating this robust spam detection system, businesses can potentially save up to $20.5 billion annually and approximately $1,934 per employee by minimizing productivity losses and reducing technical support needs.
* **Finance Industry**: The model enhances data protection against phishing and breaches, maintaining client trust and ensuring the security of financial data.
* **Healthcare Sector**: The solution supports uninterrupted secure communication, vital for protecting patient data and maintaining confidentiality.

### **Conclusion**: The deployment of these models, particularly XGBoost, not only improves the accuracy of spam filtering but also supports enhanced productivity and operational efficiency, reinforcing trust across critical industries.

### **Future Work and Improvements**

In order to enhance the performance and applicability of the spam detection models, several future directions can be considered:

**1. Expanding the Dataset:**

* Collecting more diverse email samples across different languages and regions would help improve the generalizability of the model.
* Integrating emails with multimedia content or attachments to simulate real-world scenarios more closely.

**2. Feature Engineering:**

* **Contextual NLP Features**: Extracting additional features such as sentiment scores, subject line characteristics, and named entity recognition (NER) can provide better context for classification.
* **Email Structure Analysis**: Including header analysis (e.g., 'from' fields and routing paths) may contribute to identifying spam that relies on sender obfuscation.

**3. Model Enhancements:**

* **Hyperparameter Tuning**: Further tuning of the hyperparameters for models like XGBoost and Balanced Random Forest to explore their optimal configurations.
* **Ensemble Methods**: Combining more sophisticated ensemble techniques such as stacking or bagging with deep learning models to assess improvements in precision and recall.

**4. Addressing Limitations:**

* While recall was maximized in models like the Balanced Random Forest, this sometimes came at the expense of precision. Future work should focus on finding a balanced approach to minimize false positives without compromising recall.
* Exploring methods to reduce model bias and improve performance on hard-to-classify, ambiguous emails.

**5. Deployment and Real-World Integration:**

* **Real-Time Processing**: Develop a system capable of detecting spam in real-time, considering computational constraints and latency issues.
* **User Feedback Loop**: Incorporate mechanisms for end-users to provide feedback on false positives and false negatives, thereby allowing for continuous learning and model refinement.
* **Security Integration**: Strengthen integration with existing cybersecurity tools for automated incident response and protection against potential phishing or malicious content.

**Conclusion**: These improvements and expansions would contribute to a more robust spam detection system, capable of adapting to changing spam techniques and providing organizations with enhanced protection and productivity.

**Conclusion**

**Summary of Findings**: The analysis and modelling of email data for spam classification highlighted the effectiveness of applying machine learning techniques, such as Logistic Regression with class weights, Balanced Random Forest, and XGBoost. The evaluation metrics revealed high levels of accuracy, recall, and F1-score, indicating robust model performance. The best-performing models, particularly XGBoost, demonstrated significant predictive power in identifying spam emails, while maintaining a balance between false positives and false negatives.

**Reinforcement of Business Benefits**: Implementing the spam detection model would bring tangible benefits, including:

* **Cost Savings**: Reducing the impact of spam emails, which can help save approximately $20.5 billion annually for businesses and approximately $1,934 per employee through improved productivity and reduced technical support costs.
* **Enhanced Data Protection**: Strengthening security in industries like finance and healthcare by minimizing exposure to phishing and data breach risks.
* **Operational Efficiency**: Ensuring continuous, reliable communication and safeguarding sensitive information in email exchanges.

This project has demonstrated that a multi-model approach increases the reliability and effectiveness of spam detection systems, supporting organizations in maintaining secure, efficient communication channels.

**10. References**

1. **Kaggle Email Dataset** – The primary dataset used for spam classification analysis.
2. **Scikit-learn Documentation** – Utilized for model implementation and hyperparameter tuning.
3. **Research Articles** – References to key literature on machine learning techniques and real-world spam detection systems.
4. **Industry Reports** – Supporting data for the financial impact of spam, e.g., productivity loss and security risks (source: Radicati and Nucleus Research).