**Email Spam Detection**

|  |  |
| --- | --- |
| **Name** | **ID** |
| يارا محمد عبد الرحمن محمد | **2021170618** |
| انجي عبدالملاك خله بولص | **2022170078** |
| بسمه خالد محمود محمد | **2021170117** |
| اهداب أسامة محمد توفيق | **2022179981** |
| سارة محمد محمد اسماعيل | **2022170177** |
| شهد شريف وجدي خليفة | **2022170207** |

**Introduction:**

Spam email detection is a critical task in the field of cybersecurity and email filtering. The goal of spam email detection is to automatically identify and classify emails as either spam (unsolicited or unwanted emails) or legitimate (ham) based on their content and characteristics. In modern email systems, spam emails pose significant challenges by flooding users' inboxes with unwanted content, including advertisements, phishing attempts, malware, and fraudulent schemes. Effective spam email detection mechanisms are essential to protect users from potential security threats, prevent inbox clutter, and enhance overall email user experience.

**Importing Libraries:** Importing necessary libraries such as NumPy, Pandas, Plotly Express, Matplotlib, Seaborn, and NLTK.

**NumPy:**Essential for numerical computing, NumPy provides support for large, multi-dimensional arrays and a wide range of mathematical functions. It's fundamental for scientific computing tasks.

**Pandas:** Pandas is a powerful data manipulation and analysis library. It's primarily used for handling structured data and time series data, offering data structures like DataFrame and extensive functionality for data manipulation.

**Plotly Express:** Plotly Express is a high-level interface for creating interactive visualizations. It simplifies the process of generating various types of plots with an expressive syntax, making it suitable for data exploration and presentation.

**Matplotlib:** Matplotlib is a comprehensive plotting library offering a flexible toolkit for creating static, interactive, and animated visualizations. It's highly customizable and suitable for generating publication-quality graphics.

**Seaborn:** Seaborn is a statistical data visualization library built on top of Matplotlib. It simplifies the creation of complex statistical graphics and offers built-in themes and color palettes for enhancing the appearance of plots.

**NLTK (Natural Language Toolkit):** NLTK is a platform for working with human language data. It provides tools and resources for tasks like tokenization, stopwords removal, part-of-speech tagging, and semantic reasoning, making it essential for natural language processing tasks.

1. **Data Preprocessing techniques:**

* Handling Null Values:

Null values are missing values in the dataset. The dataset has no Nulls.

* Outlier Detection:

Outliers are data points that significantly deviate from the rest of the dataset. Outliers can be detected using statistical methods such IQR (Interquartile Range), or visualizations like box plots. The original columns do not have outliers but in column length\_times\_emails that I created has 527 outliers and I replace them with min , max.

* Encoding Categorical Variables:

Categorical variables need to be converted into numerical format before feeding them into machine learning models. I used Map to do encoding to label column.

* Tokenization:

Tokenization is the process of splitting text into individual tokens, which are typically words or phrases. It is a crucial step in natural language processing (NLP) tasks.

* Stemming:

Stemming is the process of reducing words to their root or base form by removing suffixes or prefixes. It helps in reducing the dimensionality of the feature space and improving the performance of text-based machine learning models. I used Porter stemmer.

* Feature Engineering:

I extracted the length of subjects of emails.

1. **Visualization techniques:**

* Count Plot:

The count plot displays the counts of each label (spam or ham) in a bar chart. It helps visualize the distribution of labels in the dataset.

* Pie Chart:

The pie chart shows the percentage of spam and ham emails in the dataset. It provides a visual representation of the class distribution.

* Bar Plot:

The bar plot displays the number of sent emails for each label (spam or ham). It helps compare the number of sent emails between different labels.

* Pie Chart (Distribution):

This pie chart displays the distribution of sent emails by label. Each segment represents the proportion of sent emails for a specific label.

* Scatter Plot:

The scatter plot visualizes the relationship between the label number and the number of sent emails. It helps identify any patterns or correlations between these variables.

* Box Plot:

The box plot displays the distribution of the number of sent emails. It helps identify outliers and understand the spread of the data.

* Wordcloud:

The word cloud visualizes the most common words in spam emails. It provides a visual summary of the frequently occurring words.

* Kernel Density Estimate (KDE) Plot:

The KDE plot displays the probability density function of the length of spam and ham emails. It helps understand the distribution of email lengths for each label.

* Histogram:

The histogram displays the distribution of email lengths for each label in a separate subplot. It helps compare the distribution of email lengths between different labels.

* Correlation Heatmap:

The correlation heatmap visualizes the correlation matrix between numerical features in the dataset. It helps identify relationships between variables and can be useful for feature selection or identifying multicollinearity.

**3. Feature Selection:**

**.**Chi-square Test:

After preprocessing, a contingency table is created using the'pd.crosstab()' function to examine the relationship between 'label' and 'text' columns. Subsequently, a Chi-square test is performed using the 'chi2\_contingency()' function from 'scipy.stats'. This test helps determine whether there is a statistically significant association between the two categorical variables, 'label' (spam or not spam) and 'text' (content of the emails).

* Results of Chi-square Test:

The results of the Chi-square test include the Chi-square statistic, p-value, degrees of freedom, and expected frequencies. These values provide insights into the significance of the association between 'label' and 'text' columns in the dataset.

.Feature Extraction:

Next, feature extraction is performed using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique. The 'TfidfVectorizer' class from 'sklearn.feature\_extraction.text' is used for this purpose. TF-IDF is a numerical representation of text data that accounts for the importance of terms in a document relative to a corpus.

. Mutual Information:

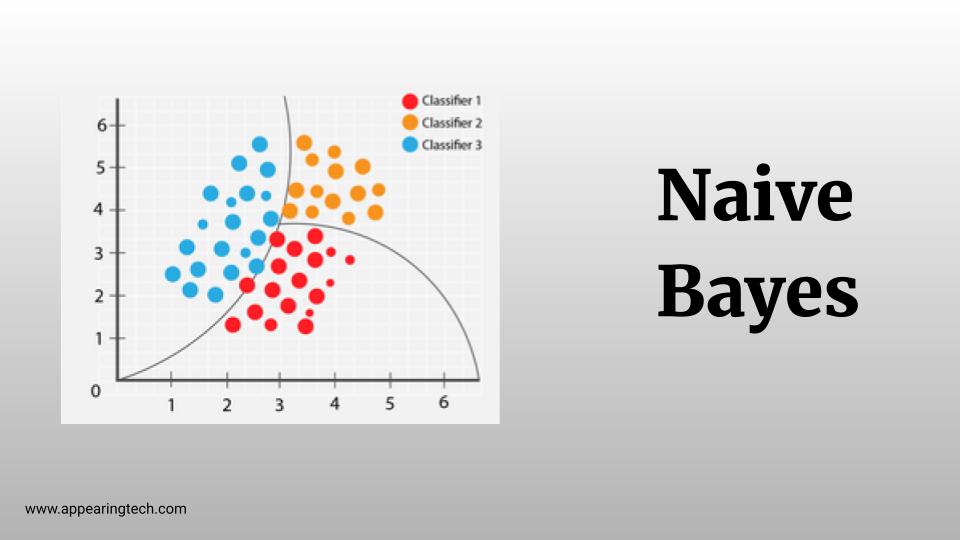
Mutual Information is computed between the transformed TF-IDF matrix and the 'label' column to assess the relevance of features (words) in predicting the spam label. Additionally, Mutual Information is calculated between the '# sent emails' column and the 'label' column to examine the relationship between the number of sent emails and the spam label**.**

**4**. **Models:**

**Multinomial Naive Bayes:**

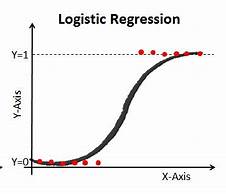
Multinomial Naive Bayes (MultinomialNB) is a probabilistic classifier based on Bayes' theorem with the assumption of independence between features. It is primarily used for text classification tasks where the features represent word frequencies or counts.

Multinomial Naive Bayes is a popular choice for text classification tasks due to its simplicity, efficiency, and effectiveness in handling high-dimensional data. However, its performance may vary depending on the specific characteristics of the dataset and the degree to which the independence assumption holds true. It's often used as a baseline model for text classification tasks and can provide competitive performance in many scenarios.



**Logistic regression:**

Logistic regression is a statistical model used for binary classification tasks. It is especially well-suited for problems where the dependent variable or target variable is categorical and has two possible outcomes, typically represented as 0 and 1. The goal of logistic regression is to estimate the probability of an event occurring based on a set of predictor variables or features. Unlike linear regression, which predicts continuous values, logistic regression models the relationship between the predictors and the binary outcome using the logistic function.



## **Decision tree:**

A Decision Tree is a popular and intuitive machine learning algorithm used for both classification and regression tasks. It operates by splitting the data into subsets based on the value of input features, forming a tree-like structure of decisions and one of its advantages is it can be implemented on non-linear data.

#### Key Features:

1-Structure:

* + **Nodes**: Represent features.
  + **Edges**: Represent decisions or rules.
  + **Leaves**: Represent outcomes or target values.

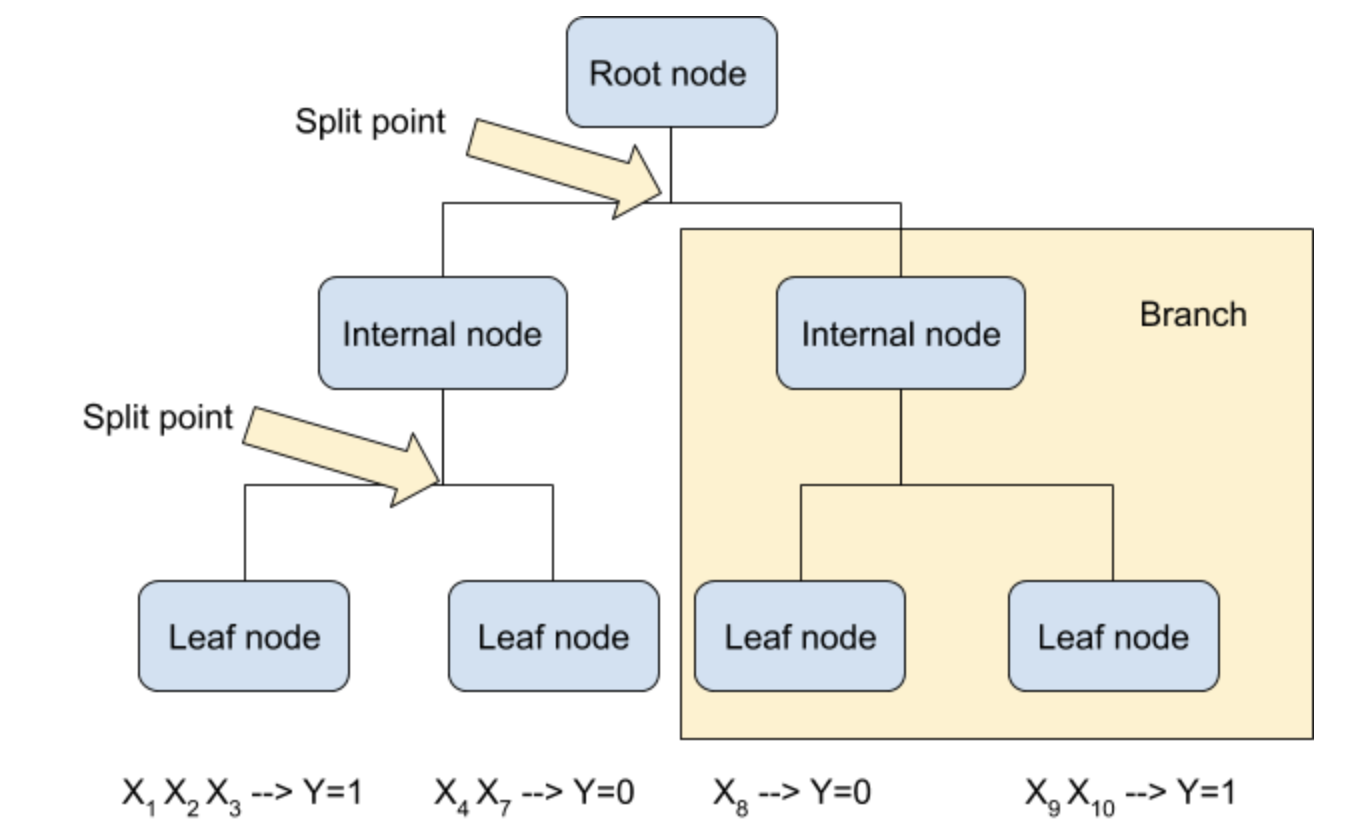
2-Splitting:

* + The process of dividing a node into two or more sub-nodes.
  + Decision Trees use criteria such as Gini impurity, entropy (for classification), or mean squared error (for regression) to decide where to split.

Math in decision tree:

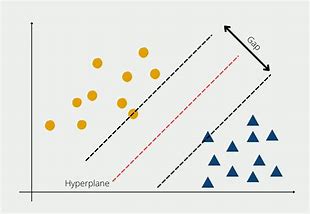
1-The entropy equation:-p \* log2(p) - (1-p) \* log2(1-p)

2-information Gain is the difference between the entropy of the parent node and the weighted average entropy of the child nodesand the result of it is in range from 0 to 1 (0-1) and 0 is the best result and the more the result gets closer to 0 the better for this feature to be selected but 1 is the worst result in this case we can ignore the feature because it doesn’t affect the result.



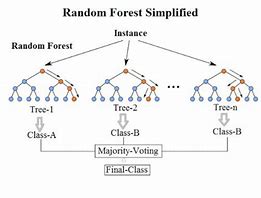
**SVM**

SVM Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It finds the optimal hyperplane that best separates different classes in the feature space by maximizing the margin between them. SVM can handle both linear and non-linear data through the use of different kernels like linear, polynomial, and radial basis function (RBF).

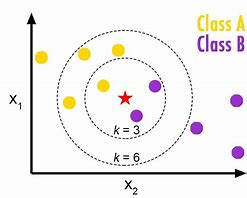


**RANDOM FOREST**

Random Forest is a machine learning algorithm used for classification tasks. It builds multiple decision trees during training. Each tree in the forest is built independently, using a random subset of the data and a random subset of features, which promotes diversity among the trees. During prediction, each tree in the forest independently makes its own prediction, and the final prediction is determined by aggregating the predictions of all the trees, by taking a majority vote. It can handle large datasets with high dimensionality and is effective in capturing complex relationships between features.



**KNN**

K-Nearest Neighbors (KNN) is a simple yet effective supervised learning algorithm for classification and regression tasks. It classifies new data points based on the majority class of their K nearest neighbors in the feature space. KNN's performance heavily relies on the choice of K and the distance metric used for calculating proximity between data points.

**Conclusion:**

**In conclusion, this email spam detection project has demonstrated the effectiveness of machine learning algorithms in identifying and filtering spam emails. Through our experimentation with logistic regression, svm , KNN, Decision tree we achieved a commendable accuracy rate of 98% on the testing dataset. Our classifier exhibited robust performance, with high precision, recall, and F1-score, indicating its ability to accurately classify both spam and non-spam emails.**

**The implications of our findings extend beyond academic research, with practical applications in enhancing email security and user experience. By implementing an effective spam detection system, organizations can mitigate the risks associated with malicious emails, improve productivity, and safeguard user privacy.**

**While our project has yielded promising results, there remain opportunities for further exploration and refinement. Future research could focus on [mention potential areas for improvement or future directions, such as exploring alternative feature selection methods, addressing imbalanced data, or integrating real-time detection capabilities.**

**In conclusion, our email spam detection project underscores the importance of leveraging machine learning techniques to combat the pervasive threat of spam emails. By continuing to innovate and refine our approaches, we can contribute to a safer and more secure digital ecosystem for users worldwide.**