Volume 10, Issue 4, October 2023 ISSN: 2348-8565 (Online)
International Journal of Modern Engineering and Research Technology
Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Spam Mail Detection with Cross-Validated Machine Learning Models

Manan Goyal
Research Scholar B.Tech.
Computer Science and Engineering
National Institute of Information Technology
NIIT University, Neemrana, (Rajasthan) India
Email: manangoyal1130@gmail.com

Latika Sharma
Research Scholar B.Tech.
Computer Science and Engineering
National Institute of Information Technology
NIIT University, Neemrana, (Rajasthan) India
Email: latikasharma018@gmail.com

Swati Soni
Assistant Professor
Department of Computer Science and Engineering
Takshshila Institute of Engineering and Technology
Jabalpur, (M.P.) India
Email: swatisoni@takshshila.org

ABSTRACT

In the age of digital communication, email remains a fundamental means of information exchange. However, the proliferation of spam emails has necessitated the development of robust spam detection systems. This project centers around the binary classification of emails into two categories: "Spam" and "Ham" (non-spam). Using a dataset of 5572 email samples, our objective is to employ various machine learning models to accurately classify incoming emails and enhance the user's email experience.

The dataset, sourced from Kaggle, comprises two columns: "Category" (with values "Ham" and "Spam") and "Message." Our methodology comprises two key steps, each exploring different feature extraction techniques and a range of machine learning models. In the first step, we employ a Train-Test split, allocating 80% of the data for training and 20% for testing. For feature extraction, we employ the "CountVectorizer" technique, which counts the occurrences of terms within each document. This creates a matrix of raw term frequencies, serving as input for the classification models.

An ensemble of classification models, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and XGBoost, is then evaluated in terms of their training and testing accuracies. This analysis highlights the performance of these models for the binary classification task. Results demonstrate that SVM achieves high accuracy on the test data, making it a strong contender for spam email detection.

The second step of our methodology focuses on feature extraction using the "TfidfVectorizer," which considers both term frequency within a document and the inverse document frequency across the dataset. The dataset is preprocessed, and label encoding is applied to the "Category" column, enabling the application of this technique. A cross-validation approach is adopted for model evaluation, providing a more comprehensive understanding of model performance. The mean accuracy scores across various models are computed, revealing that Random Forest exhibits exceptional performance during cross-validation, indicating its potential for spam email detection.

Keywords:—*Email Classification, Spam Detection, Machine Learning Models, Cross-Validation, Text Feature Extraction, Data Analysis.*

I. INTRODUCTION

In the digital age, email communication remains a cornerstone of modern business and personal interactions. However, this convenience comes with its own set of challenges, one of the most persistent being the influx of unsolicited, irrelevant, or potentially harmful emails, commonly referred to as "spam." Spam emails often contain advertisements, fraudulent schemes, or other unwanted content, and they can clutter inboxes, waste time, and pose security risks.

To mitigate the impact of spam emails and enhance the user experience, the field of email classification and spam detection has gained significant attention. Email service providers, organizations, and individuals alike seek effective methods to automatically identify and segregate spam from legitimate emails, commonly referred to as "ham" emails.

Machine learning has emerged as a powerful tool in this endeavor. By leveraging the capabilities of machine learning models, it is possible to develop systems that can distinguish between spam and ham emails, providing users with a cleaner and more secure email environment. These models are trained on labeled datasets containing examples of both spam and ham emails, allowing them to learn patterns and characteristics that differentiate the two categories.

This project focuses on the task of binary classification, wherein emails are categorized into one of two classes: "Spam" or "Ham." The primary objective is to assess the performance of various machine learning models in this classification task, with a particular emphasis on cross-validation to ensure robust results. The project utilizes a real-world dataset containing 5572 emails, each labeled as either spam or ham.

The machine learning models employed in this study encompass a range of algorithms, including Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and XGBoost. By evaluating these models in terms of their training and testing accuracies, and subsequently performing cross-validation to calculate their mean accuracy, this research aims to provide insights into which models are most effective at spam detection.

The findings of this study hold relevance for email service providers seeking to enhance their spam filtering systems and for individuals and organizations looking to reduce the impact of spam on their inboxes. By leveraging the power of machine learning and cross-validation, this project contributes to the ongoing efforts to improve email security and the overall email user experience.

II. MACHINE LEARNING

Machine learning is a subset of artificial intelligence (AI) that equips computers to learn and make decisions or predictions based on data, all without explicit programming. It encompasses three primary types of learning:

1. ***Supervised Learning:*** In this approach, algorithms are trained using labeled datasets, associating input data with expected outcomes. The goal is to establish a mapping between inputs and outputs, enabling

the model to make precise predictions for new, unseen data.

2. ***Unsupervised Learning:*** Unsupervised learning operates on unlabeled datasets, seeking to uncover patterns, structures, or relationships within the data. Common tasks include clustering similar data points or reducing data dimensionality.

3. ***Reinforcement Learning:*** Reinforcement learning concentrates on training algorithms to make decisions in an environment with the aim of maximizing cumulative rewards over time. The environment is used to teach the model, providing feedback in the form of rewards or penalties based on its actions.

Machine Learning Tasks:

Machine learning encompasses two primary types of tasks:

1. ***Classification:*** A form of supervised learning where input data points are assigned to specific categories or classes. The model learns to map input data to predefined output classes, making it suitable for tasks with categorical output variables.

2. ***Regression:*** Also, a type of supervised learning, regression aims to predict continuous numerical values based on input data. In regression tasks, the model learns to establish a relationship between input features and the output variable, making it useful for tasks with continuous output variables.

Given the **nature of the Category Column in Dataset "mail_data.csv", the appropriate approach to solve this problem** is Classification**.** In proposed work following Machine Learning Classification Algorithms are used–

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Support Vector Classifier
- K Neighbors Classifier
- GNB (Gaussian Naive Bayes Classifier)
- XGB (Extreme Gradient Boosting)

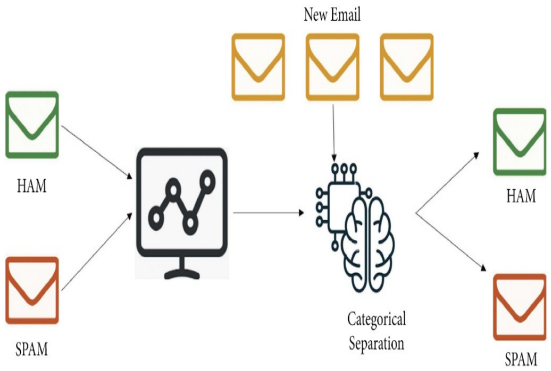

Figure 1. Spam Mail Detection using Supervised Learning

A. Feature Extraction

"Feature extraction" refers to the process of transforming the raw text data from the email messages into numerical features that can be used as input for machine learning models. This step is crucial for turning unstructured text data into a format that machine learning algorithms can work with effectively.

CountVectorizer is another technique used in natural language processing (NLP) to convert a collection of text documents into a numerical format that can be used for machine learning tasks. Unlike TfidfVectorizer, which takes into account the importance of terms in a document and across a corpus (using TF-IDF scores), CountVectorizer simply counts the frequency of each term in each document.

TfidfVectorizer stands for "Term Frequency-Inverse Document Frequency Vectorizer," and it is a common technique in natural language processing and text mining. It is used to convert a collection of raw documents into a numerical format that machine learning models can work with effectively.

1. ***Term Frequency (TF):*** This part of the technique measures how frequently a term (word) occurs in a document. It is calculated by counting the number of times a term appears in a document and then normalizing it by the total number of terms in the document. This helps identify the importance of a word within a specific document.

2. ***Inverse Document Frequency (IDF):*** This part measures how important a term is across a collection of documents (a corpus). It is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents that contain the term. Terms that occur frequently in many documents will have a lower IDF value, while terms that are unique to specific documents will have a higher IDF value.

3. ***TF-IDF:*** The TF-IDF score of a term in a document is calculated by multiplying its TF by its IDF. This results in a score that reflects both the frequency of the term within the document and its importance within the corpus. Terms with higher TF-IDF scores are considered more important for that document.

The TF-IDF vectorizer converts text documents into a matrix of TF -IDF features, where each row represents a document, and each column represents a unique term in the entire corpus. These numerical features can then be used as input for machine learning models, making it possible to perform text-based tasks like document classification, sentiment analysis, and information retrieval.

B. K- fold Cross Validation

Train-test splitting, a conventional method for evaluating machine learning models, comes with inherent disadvantages. One key drawback is the unpredictability in performance evaluation due to the randomness in selecting data points for the training and testing sets. This can result in significant variability in a model's performance, making it difficult to derive consistent conclusions about how well it generalizes to unseen data. Moreover, train -test splitting often leads to data wastage, particularly when working with small datasets. By reserving a portion of the data for testing, the amount available for model training diminishes, increasing the risk of creating less robust models that may overfit to the training data. These limitations necessitate an alternative approach.

In response to these challenges, k-Fold Cross-Validation emerges as a powerful solution. This technique divides the dataset into k subsets or folds and iteratively employs each fold for both training and testing. By repeating this process k times and averaging the model's performance across these iterations, cross-validation provides a more stable and representative estimate of a model's ability to generalize. Importantly, it ensures efficient data utilization, allowing each data point to

contribute to both the training and testing phases. This feature is particularly valuable when dealing with limited data resources. Beyond its role in enhancing stability, cross-validation plays a pivotal role in model selection and hyperparameter tuning, facilitating reliable comparisons between different models or parameter settings and thus strengthening the overall robustness of machine learning model assessment and improvement.

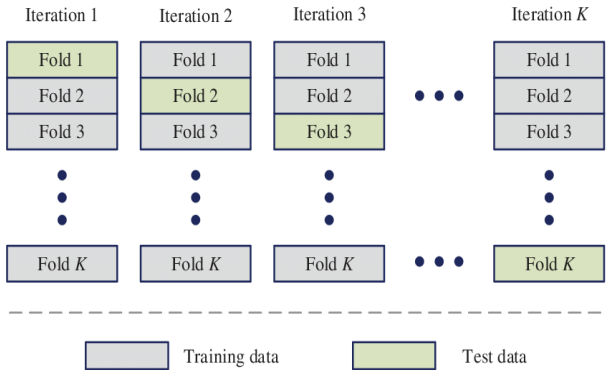

Figure 2. k-fold Cross Validation

III. Literature Review

The abstract outlines a research project that focuses on improving email spam identification. The researchers have proposed a model that classifies emails into spam and non-spam categories. They employed several techniques, including DBSCAN and Isolation Forest for outlier detection, feature selection methods like Heatmap, Recursive Feature Elimination, and Chi-Square, and implemented the model using both machine learning and deep learning techniques.

In terms of results, the proposed model achieved the following numerical values:

Machine Learning Implementation:

Accuracy: 100%, AUC (Area Under the Curve): 100, MSE (Mean Squared Error) error: 0

RMSE (Root Mean Squared Error) error: 0

Deep Learning Implementation:

Accuracy: 99%, Loss value: 0.0165

Moreover, in the data cleaning phase, the abstract mentions that outliers were removed from the dataset using two methods are DBSCAN and Isolation Forest

The abstract also provides a table that shows the performance of deep learning algorithms when combined with outlier detection techniques (DBSCAN and Isolation Forest). The results include loss values and accuracy percentages for different combinations:

Outlier Detection Technique	Deep Learning Algorithms	Loss Value	Accuracy
DBSCAN	Recurrent Neural Network (RNN)	0.6803	92.42 %
Isolation Forest	Recurrent Neural Network (RNN)	0.0450	99.28 %
DBSCAN	Gradient Descent	0.2568	89.42 %
Isolation Forest	Gradient Descent	0.0165	99.28 %
DBSCAN	Artificial Neural Network (ANN)	0.2469	91.42 %
Isolation Forest	Artificial Neural Network (ANN)	0.1333	97.95 %

Figure 3. Result analysis based on Deep Learning algorithms

several machine learning algorithms are used for the implementation of the email spam classification model. The machine learning algorithms mentioned include:

1. ***Multinomial Naïve Bayes (MNB):*** It's an extension of the Naïve Bayes algorithm, where each feature in a feature vector is assigned a weight. It is commonly used for text classification tasks.
2. ***Random Forest (RF):*** Random Forest is an ensemble learning method that creates multiple decision trees to make predictions. It's known for its robustness and accuracy.

3. ***K-Nearest Neighbor (KNN):*** K-Nearest Neighbor is a supervised learning algorithm used for classification tasks, where it classifies new data points based on their similarity to existing data points.

4. ***Gradient Boosting (GB):*** Gradient Boosting is another ensemble method that combines the output from weak learners (typically decision trees) to produce a model with improved accuracy.

These machine learning algorithms are used in combination with feature selection techniques and an ensemble method called "Stacking" to classify emails into spam and non-spam categories. The combination of these algorithms aims to enhance the accuracy and effectiveness of the email spam identification model[1].

This work[2] addresses the challenge of classifying emails as spam or non-spam. It employs a multi-step methodology:

1. ***Data Preprocessing:*** Cleaning the dataset by tokenization, stop word removal, and stemming.

2. ***Relationship Analysis:*** Assessing word relationships in email subjects and content, and scoring words based on entropy.

3. ***Feature Selection:*** Selecting the most informative words for email classification.

4. ***N-Grams:*** Generating N-grams (word sequences) from selected informative words.

5. ***TF-IDF Normalization:*** Reducing the high count of N-grams using TF-IDF.

6. ***CHI Square Feature Selection:*** Choosing top N-grams for classification.

7. ***Vocabulary Corpus:*** Constructing a vocabulary corpus for email classification.

8. ***Classification:*** Employing four classifiers, including Linear Support Vector Machine (LSVM), for email classification.

Results show that LSVM outperforms other classifiers with nearly 91% accuracy, high precision, and specificity. The research emphasizes word relationships in email content and subject as key to accurate classification, offering potential for further improvements in email filtering.

This work[3] focuses on the detection of spam emails using machine learning algorithms optimized with bio-inspired methods. The key elements of this research are as follows:

Name of the Dataset	Ref.	Spam + Ham = Total emails	Rate of Spam	Published Date
Ling-Spam	[29]	481+2412 = 2893	17%	2000
PU1	[31]	481+618 = 1099	44%	2000
SpamAssassin	[33]	1897+4150 = 6047	31%	2002
PUA	[31]	571+571 = 1142	50%	2003
PU2	[31]	142+579 = 721	20%	2003
PU3	[31]	1826+2313 = 4139	44%	2003
Enron 1 – 6 Spam	[30]	20170+16545 = 36,715	55%	2006

Figure 4. Dataset used

1. ***Background:*** Email communication has become a vital part of modern life, but it is often exploited by spammers who send unsolicited emails for fraudulent purposes.

2. ***Objective:*** The article aims to propose a method for effectively detecting spam emails by leveraging machine learning algorithms, particularly those optimized with bio-inspired techniques.

3. ***Methodology:*** The study includes a literature review to investigate effective methods used on various datasets. Machine learning models, such as Naïve Bayes, Support Vector Machine, Random Forest, Decision Tree, and Multi-Layer Perceptron, are implemented on seven different email datasets. Feature extraction and pre-processing are also carried out to prepare the data for analysis.

4. ***Bio-Inspired Algorithms:*** The research integrates bio-inspired algorithms, specifically Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), to enhance the performance of the machine learning classifiers.

5. ***Results:*** Among the models tested, Multinomial Naïve Bayes with Genetic Algorithm optimization yielded the best overall performance. The study also provides a comparison with other machine learning and bio-inspired models to identify the most suitable approach.

6. ***Conclusion:*** The project successfully implemented machine learning models with the support of bio-inspired algorithms. The study tested approximately 50,000 emails, including both numerical and alphabetical datasets. The numerical datasets had limitations in feature extraction due to word replacement with numbers. However, the alphabetical datasets performed better in feature extraction and prediction. The Naïve Bayes algorithm emerged as the top performer for spam detection, achieving 100% accuracy in some cases with GA optimization. Genetic Algorithm demonstrated more impact than PSO on various machine learning algorithms, including MNB, SGD, RF, and DT, regarding F1-Score, precision, and recall.

In summary, this research explores the use of machine learning and bio-inspired algorithms to effectively detect spam emails and concludes that Multinomial Naïve Bayes optimized with Genetic Algorithm is a robust choice for this purpose.

IV. METHODOLOGY

A. Spam Mail Prediction Data Source

- IDE - Google Collaborator / Python–Python 3
- Dataset : https://www.kaggle.com/code/mohinurabdurahimova/spam-mail-prediction-machine-learning-project/input
- File Size : 474 KB and Dataset Size:5572 rows × 2 columns
- Applied Binary Classification Problem on Data

Table 1: Dataset Attributes

Sr.	Feature	Description	Detailed Description
1	Category	Mail Type : Spam , Ham After Label Encoding Spam 0 Ham 1	Spam Mail - Fake or False Mail (Promotion Purse) Ham Mail - Not Spam Mail / True or legiti-mate Mail
2	Message	contains the actual text	It includes the text body of the communi-cation.

B. Implementation

1. ***Importing Libraries***
2. ***Drive Mount and Data Collection***
   - Reading CSV file (mail_data.csv) using pandas
3. ***Data Pre-processing Starts:***
   - Printing dataframe df / data Exploration to obtain information about a DataFrame
4. ***Check for Null Values:***
   - Checking for missing values in the dataframe using df.isnull().sum().
5. ***Data Pre-processing (Continued):***
   - Label Encoding
   - Separating the Dataset Featues into features (x) and target (y).
   - Plot Pie Chart for target (y)
   - Using train_test_split to further divide the data into training and testing sets with 80% training and 20% testing data.
   - CountVectorizer is used for Conversion of Text Data into Numerical Values
6. ***Model Creation:***

   Five Classification models are created on Training – Testing Data and evaluated using Confusion Matrix. The models are:

   - Logistic Regression Model
   - Decision Tree Classifier Model
   - Random ForestClassification Model
   - Support Vector Classifier Model
   - K Neighbors Classifier Model
   - Gaussian Naïve Bayes Classifier Model
   - XGB(Extreme Gradient Boosting) Classifier
7. ***Model Evaluation***
   - Building a Predictive Model based on Highest Model Accuracy
8. ***Text Pre-processing using TfidfVectorizer:***
   - TfidfVectorizer is used for Conversion of Text Data into Numerical Values
9. ***Cross Validation based Model Creation***
   - k-Fold Cross Validation over (x, y) data is Applied on the Models
10. ***Model Evaluation:***
    - Building a ***Predictive*** Model based on Highest Model Mean Accuracy

C. Results

- Models Train Test Accuracies and Cross Validation Mean Accuracies

Train – Test and Cross Validation Accuracy Graph & Predictive Models

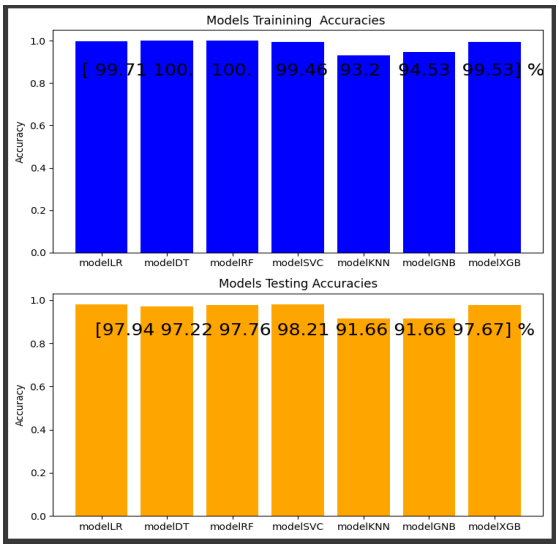

Figure 5. Model's Train Test Accuracies

Table 2. Accuracies of Different Models

Classifier Models	Train Test Accuracy	Cross Validation, CV=10	
Logistic Regression	Accuracy - Train 0.997083239847431 Test 0.979372197309417	cv_score_LR [0.96594982 0.95698925 0.9497307 0.96768402 0.96050269 0.95332136 0.9443447 0.9551167 0.95152603 0.95332136]	mean_accuracy_LR 0.9558486644401973
Decision Tree	Accuracy – Train 1.0 Test 0.9721973094170404	cv_score_DT [0.98028674 0.97132616 0.96768402 0.97486535 0.97307002 0.97307002 0.96588869 0.96947935 0.96947935 0.97307002]	mean_accuracy_DT 0.9718219725487923
Random Forest	Accuracy - Train 1.0 Test 0.9775784753363229	cv_score_RF [0.99103943 0.97670251 0.98204668 0.97845601 0.97845601 0.97845601 0.98025135 0.97666068 0.97127469 0.97486535]	mean_accuracy_RF 0.9788208721839347
Support Vector Classifier	Accuracy - Train 0.9946152120260264 Test 0.9820627802690582	cv_score_SVC [0.9874552 0.96594982 0.97666068 0.98025135 0.97307002 0.96588869 0.97307002 0.97307002 0.97307002 0.97307002]	mean_accuracy_SVC 0.9741555825820608
KNeighbour	Accuracy - Train 0.9320170518285843 Test 0.9165919282511211	cv_score_KNN [0.90143369 0.91218638 0.91202873 0.90664273 0.90305206 0.90125673 0.9048474 0.91202873 0.90305206 0.91382406]	mean_accuracy_KNN 0.9070352567196258
GNB	Accuracy - Train 0.9452546555979359 Test 0.9165919282511211	cv_score_GNB [0.89247312 0.89247312 0.87253142 0.88150808 0.86175943 0.89048474 0.88150808 0.87253142 0.88868941 0.88150808]	Mean Accuracy GNB 0.881546688287871
XGB	Accuracy - Train 0.9952883105227731 Test 0.9766816143497757	cv_score_XGB [0.98566308 0.97132616 0.97486535 0.98025135 0.98025135 0.97307002 0.98025135 0.96947935 0.96768402 0.98384201]	Mean Accuracy: 0.9766684040848632

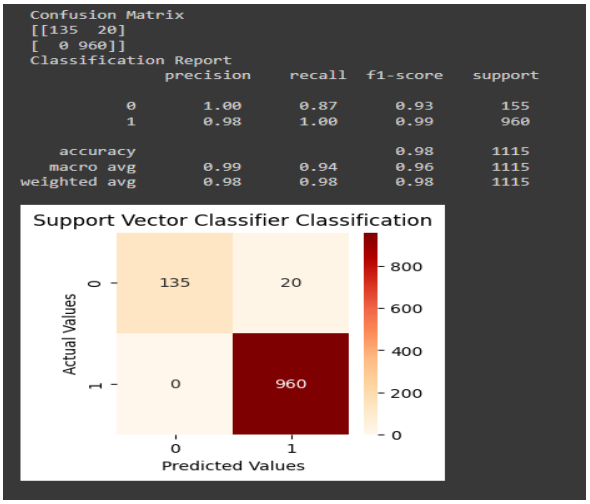

Figure 6. Based on Highest Accuracy Building a Predictive Model

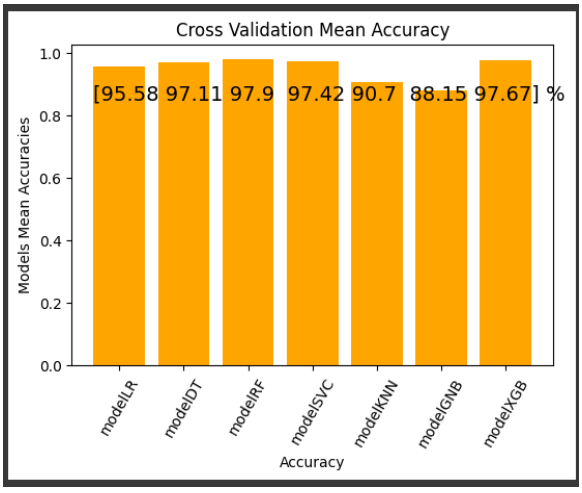

Figure 7. Models Cross Validation Mean Accuracy

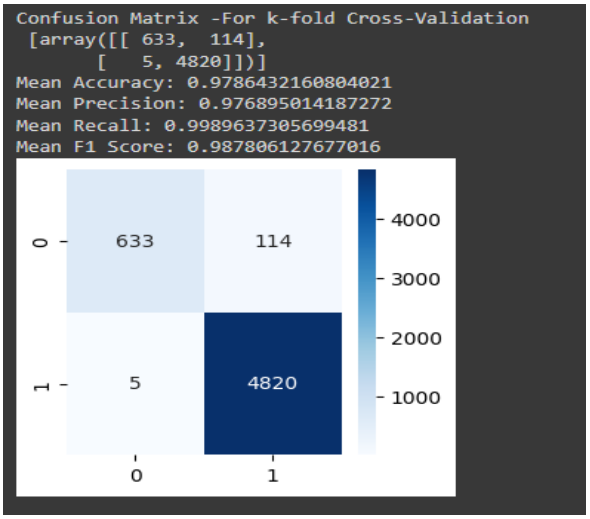

Figure 8. Based on Highest Cross Validation Mean Accuracy Building a Predictive Model

V. Conclusion And Future Work

Conclusion

The performance metrics of various classification models on the task are as follows:

In this task, the objective was to predict whether an email is spam (fake or false mail) or ham (legitimate mail) using machine learning. The dataset, consisting of 5572 rows and 2 columns, was obtained from a Kaggle source. After preprocessing, the dataset featured two primary attributes: "Category," which indicated the type of mail (spam or ham), and "Message," containing the text body of the communication. we conducted a comprehensive analysis, from data preprocessing to model evaluation. After encoding the "Category" feature to represent spam as 0 and ham as 1, we divided the dataset into training and testing sets. ML Models Created on Train -Test sets and evaluated. To enhance the performance of the models k-fold cross validation applied over the complete data set and then evaluated.

- Logistic Regression achieved a remarkable train accuracy of 99.71% and a solid test accuracy of 97.94%. The mean cross-validation accuracy was 95.58%.
- Decision Tree exhibited perfect train accuracy of 100% and an excellent test accuracy of 97.22%. The mean cross-validation accuracy was 97.18%.
- Random Forest also showed a perfect train accuracy of 100% and an impressive test accuracy of 97.76%. The mean cross-validation accuracy was 97.88%.
- Support Vector Classifier (SVC) attained a high train accuracy of

99.46% and an impressive test accuracy of 98.21%. The mean cross-validation accuracy was 97.42%.

- K Neighbors (KNN) had a train accuracy of 93.20% and a test accuracy of 91.66%. The mean cross-validation accuracy was 90.70%.
- Gaussian Naïve Bayes (GNB) achieved a train accuracy of 94.53% and a test accuracy of 91.66%. The mean cross-validation accuracy was 88.15%.
- Extreme Gradient Boosting (XGBoost) demonstrated a train accuracy of 99.53% and a test accuracy of 97.67%. The mean cross-validation accuracy was 97.67%.

Based on the highest cross-validation mean accuracy, the Random Forest model, with a mean accuracy of 97.88%, was selected to build the predictive model. The k-fold cross-validation for Random Forest exhibited a mean accuracy of 97.94% and strong precision, recall, and F1 score, highlighting its robust performance in consistently identifying spam and legitimate emails.

In summary, the Random Forest model, with its high cross-validation mean accuracy, proved to be the most reliable choice for classifying email messages, offering an impressive balance between learning from the training data and generalizing effectively to new, unseen email samples.

Future Work

In the context of Spam Mail Prediction, future work should focus on leveraging advanced technologies like deep learning models, particularly transformer-based architectures, to enhance the detection of intricate spam patterns and context in email content. Additionally, exploring hyperparameter tuning and ensemble methods can further boost model performance. It's vital to consider feature engineering, advanced text preprocessing, and real-time detection systems to swiftly identify emerging threats. Integrating email metadata and multimodal data, like images or attachments, can provide a holistic view for classification.

To ensure practical deployment, scalability and efficiency must be addressed to seamlessly integrate these advanced models into real-world email systems. Robust security awareness programs within email platforms can educate users about evolving spam tactics. Moreover, adopting a broader range of evaluation metrics will provide a more comprehensive view of model performance. These innovations, along with deep learning, real-time detection, and multimodal data integration, are essential to counter evolving spam tactics and safeguard users effectively.

REFERENCES:

[1] Fahima Hossain, Mohammed Nasir Uddin, Rajib Kumar Halder, “Analysis of Optimized Machine Learning and Deep Learning Techniques for Spam Detection”, 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) | 978-1-6654-4067-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEMTRONICS52119.2021.9422508.

[2] Aakanksha Sharaff, Ulligaddala Srinivasarao, “Towards classification of email through selection of informative features”, 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T), 978-1-7281-4997-4/20/$31.00 ©2020 IEEE.

[3] Simran Gibson , Biju Issac, Li Zhang, Seibu Mary Jacob, "Detecting Spam Email With Machine Learning Optimized With Bio-Inspired Metaheuristic Algorithms", January 2020, IEEE Access 8:187914-187932, DOI:10.1109/ACCESS.2020.3030751.

[4] Lakshmana Phaneendra Maguluri, R. Ragupathy, Sita Rama Krishna Buddi, Vamshi Ponugoti ,Tharun Sai Kalimil, "Adaptive Prediction of Spam Emails Using Bayesian Inference", Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019), IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4.

[5] Naeem Ahmed, Rashid Amin, 1Hamza Aldabbas, Deepika Koundal, Bader Alouffi, and Tariq Shah1, "Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges", Hindawi Security and Communication Networks Volume 2022, Article ID 1862888, 19 pages https://doi.org/10.1155/2022/1862888.

[6] B. Uday Reddy, S. Nagasai Tej , Md. Shoheb, Dr. Krishna Samalla, Y. Sreenivasulu, "Spam Mail Prediction Using Machine Learning", © 2023 IJCRT | Volume 11, Issue 5 May 2023 | ISSN: 2320-2882.

[7] M.A. Nivedha, S.Raja, "Detection of Email Spam using Natural Language Processing Based Random Forest Approach", IJCSMC, Vol. 11, Issue. 2, February 2022, pg.7 – 22.

[8] Sneha Bobde, Sharvari Role, Lokesh Khadke, Tejas Shirude, Ms. Shital Kakad, "Email Spam Detection Using Hybridization of SVM and Random Forest", International Journal of Research in Engineering and Science (IJRES), ISSN (Online): 2320-9364, ISSN (Print): 2320-9356, www.ijres.org Volume 11 Issue 7 ǁ July 2023 ǁ PP. 188-193.

[9] Taher Jodiawala, Manav Bhagia, Prateek Duhoon, Shubhay Islaniya, Nivedeeta Mukherjee, Nutan Dolzake, "Intelligent Spam Mail Detection System", International Research Journal of Engineering and Technology (IRJET), Volume: 10 Issue: 06 | Jun 2023 www.irjet.net p-ISSN: 2395-0072.

[10] Hardik N Patel, Shilpa Serasiya, "Machine Learning for Email Spam Messages Detection", International Journal of Progressive Research in Engineering Management and Science (IJPREMS), Vol. 02, Issue 05, May-2022, pp : 44-48.

[11] Hari K.C., "Comparative Analysis and Prediction of Spam Email Classification Using Supervised Machine Learning Techniques", International Research Journal of Modernization in Engineering Technology and Science, Volume:03/Issue:05/May-2021 Impact Factor-5.354 www.irjmets.com.

* * * * *