Project Title

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Name of Student: Divvela Hema Harshini

Email id: harshini.divvela07@gmail.com

Under the Guidance of

Mr. Abdul Aziz Md

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor, Mr. Abdul Aziz sir, for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for this project. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional. All the guidance provided by my mentor helped me in improving my understanding regarding spam mail classification and AI techniques usage.

Throughout this internship his guidance was incredible because of his guidance only I have successfully completed this internship and learned a lot of new things. The way he teaches was so simple and understandable I am very thankful to the mentor .His guidance helped me improve the intrest in learning new things .Thank you

ABSTRACT

As there is an issue that needs to be dealt with through manual sorting and organizing of emails. Spam emails do not only make the inbox cluttered but also make the inbox unsafe. Traditional filters employing rules-they have a set definition of rules they use to define spam-have been found to be counterproductive for the reason that spammers keep changing their spam format. Machine learning(ML) can be used in addressing the above as it keeps learning from the data and thus transferring experience into the new entered data with new spams emerged through time. This project aims toward developing a machine learning model, which classifies emails as `spam' or `not spam' with a high degree of accuracy so that maximum manual effort is not spent on the manual handling and management of the emails.

The data for the study includes a labeled dataset containing emails as either spam or not spam. Pre-processed with cleaning, tokenization, stopword and punctuation removal, and special symbols elimination, the text will have Natural Language Processing (NLP) techniques applied to it. The Count Vectorizer will be used for converting the cleaned text into a numerical features matrix-basing on the word frequencies. This feature matrix is then utilized for training a Naive Bayes classifier for learning the distinguishing patterns between spam and legitimate emails. The model's performance is evaluated using standard metrics, such as accuracy, to assess the effectivity of performance related to proper email classification.

These challenges comprise i) imbalanced datasets, ii) rapidly evolving spam tactics, and iii) requirement of efficient real-time processing. Such algorithms can relatively use the machine learning model, especially with NLP, as it adapts to new emerging concepts of spam tactics thereby improving accuracy of email filtering overtime.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1Problem Statement:

Imagine you receive a ton of emails every day. It would be impossible to read and sort them all manually. A machine learning model can help by automatically separating important emails from junk (or spam). This way, you can focus on the emails that matter most to you.

Spammers constantly change their tactics to bypass traditional rule-based filters. A machine learning model can learn from patterns in new data and adapt over time, making it more effective at detecting evolving spam content.

1.2Motivation:

The spam emails, is one of the common issues in the digital world today, the problem seems all-pervasive. Spam clogs inboxes and offers security risks, not to mention wasting valuable time. The opportunity to contribute toward a solution that can adequately filter and mitigate spam is indeed very motivating. I would like to develop a spam classification model robust enough to enhance email security, improve user experience, and protect people from potential cyber threats. Moreover, this project would help me apply my interest in natural language processing and machine learning to solve a real-world problem

1.3Objective:

The objective of this project is to develop a highly intelligent and adaptive machine learning model that can autonomously classify and separate important emails from spam with high accuracy and adaptiveness. Preprocessing the email text for effective NLP analysis using techniques such as tokenization, stopword removal, lemmatization, and feature extraction methods like BoW or TF-IDF will allow the Model be subjected to a supervised machine learning algorithm Naive Bayes for classification purposes selected because it is best suited for text data and highly efficient computes. Build an intelligent spam filtering solution to learn from evolving patterns in the spam world; thus, it can adapt to new spam techniques over time. The model is meant to minimize false positives and negatives in spam detection, increasing productivity by reducing distractions from spam emails and improving security by detecting potentially harmful content. The system is scalable, privacy-oriented, and optimized for real-time email filtering; thus, it can be easily integrated into existing email platforms. In TP, this high-performance model has elegant metrics like precision, recall and F1 score that enable it to become an efficient and secure means of email communication.

1.4.1 Scope of the Project:

Streaming Analytics:

Implementing real-time classification systems that can process incoming emails immediately, enabling prompt actions.

Adversarial Attacks:

Developing techniques to defend against adversarial attacks that aim to deceive classification models.

Industry Specific Models:

Building models specific to certain industries (such as finance, health care, e-commerce) with specific classification needs

1.4.2 Limitation of the Project:

Feature Independence Assumption: Naive Bayes assumes that features are independent, which may not always hold true in real-world scenarios.

Sensitivity to Data Sparsity:

If a word appears infrequently in the training data, it can significantly impact the classification accuracy**.**

Difficulty in Handling Complex Patterns:

Naive Bayes may fail to capture complex patterns in spam emails, especially those using sophisticated techniques to evade detection.

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

Relevant Work in Spam Mail Classification

- **Machine Learning Algorithms for Spam Filtering (2006):**

Authors: Androutsopoulos et al. Study: Application of machine learning techniques such as Naive Bayes, SVM, and K-Nearest Neighbors (KNN) in spam filtering.
Findings: Naive Bayes is efficient for text-based data: competitive accuracy and speed. SVM shows good accuracy for non-linear data. However, the resource optimization required is fairly high.

- **Spam Filtering Using TF-IDF and Machine Learning (2010):**

Authors: Zhang et al. Summary: Feature extraction techniques (such as TF-IDF and word counts) incorporated with classifiers like Logistic regression, Decision trees, and Naïve Bayes were used to explore various combinations.
Findings: Naïve Bayes works well for sparse, highly dimensional cases but independence assumption limits performance with highly correlated features.

- **Random Forest for Spam Classification (2015):**

Authors: Youn and McLeod. Summary: They compared Decision Trees and Random Forest in the classification of spam emails.
Findings: Compared with a single Decision Tree, Random Forest yielded improved accuracy and reduced overfitting but at a high computational overhead.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

Here are the few existing models for spam mail classification:

1. **SVM Support Vector Machine**

As one of the most classical algorithms, SVM is the choice for spam classification as it finds a hyperplane with the maximum margin distance separating the classes (spam and ham). It is compatible with both linear and nonlinear separations when it utilizes kernel functions such as the Radial Basis Function kernel.

How SVM Works in Spam Classification:

SVM creates hyperplane decision boundaries in feature space for classifying emails into spam or non-spam categories.
Feature vectors have a textual conversion, for example, by the use of word frequencies or TF-IDF scores.
SVM, by means of static kernel functions, allows complex geometric forms of non-linear decision boundaries for very flexible data sets

Advantages:

Complex data has higher classification accuracy, especially when using non-linear kernels.
It handles very well with high-dimensional spaces, for example, text classification.

Challenges:

- Very sensitive to hyperparameter tuning (C, kernel choice).
- It is computationally extensive and much slower when handling really big datasets.
- It becomes impossible to interpret the model in the case of non-linear kernels.

2. **Logistic Regression**

Overview:
Logistic regression is a linear classifier used for binary classification purposes. It estimates the probability of a given input belonging to one class or another using a logistic (sigmoid) function.

How It Works:

Uses the logistic function:
Let P(class | features) = 1/(1 + ex - w T x), where w are weights and x are feature values. Features usually are word counts or TF-IDF scores from emails.
Gains:

Simple, interpretable, efficient
Performs well where features-to-target relationship is approximately linear.

Challenges:

Highly non-linear decision boundaries.
Not as competitive with advanced text methods.

3. . **Decision Trees and Random Forests**

Overview: Decision Trees are structured just like flowcharts in which decision nodes are based on some feature and the leaves are the class labels. Random Forest is an ensemble of multiple decision trees.

How It Works:

Decision Trees: Split the dataset recursively using feature values to create tree structure.
Random Forest: Construct multiple decision trees and aggregate the predictions from them, usually using majority voting.
Advantages:

- Can handle categorical and continuous features.
- Easy to interpret, especially for decision trees.
- Reduces overfitting and variance through averaging of multiple trees in Random Forest.

Challenges:

- Decision Trees may fit well but can overfit on lesser data; they thus need proper pruning.
- Random Forests are robust but can also be expensive computations and cumbersome interpretations due to the ensemble nature.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

Limitations in Current Approaches:

1. **SVM (Support Vector Machine):**
   - **Sensitivity to Hyperparameter:**
     SVM is very crucial with parameters such as C and the kernel, without much scope for user judgment.
   - **Computational Intensity:** Such systems are rendered useless as they break in computation analysis when subjected to very large datasets.
   - **Non-interpretability**:The decision boundary cannot be easily understood because of the nonlinearity of the kernel.

2. **Logistic Regression:**
   - **Prohibited Decision Boundaries**:These are worst because they performed very poorly in cases of non-linear separations in the data.
   - **Less Competitive:** Further,it is lagging behind while competing with newly advanced algorithms for very large and complex datasets.

3. Decision Trees and Random Forests:
   - **Overfitting for Trees:** Single tree is good for model building, but these trees need to be pruned to prevent overfitting.
   - **Complex with the Random Forests**: This much processing is costly and complex in nature.

The Capability of My Project Using Naive Bayes to Bridge

1. **Simplicity and Speed:**
   - Naive Bayes is fast and computationally feasible; being extremely efficient in handling massive data it does not impose hyperparameter tuning like SVM or Random Forest.

2. **Interpretability:**
   - This easily interprets straightforward probabilities for classifications, as compared to non-linear kernels in SVM or ensembles in Random Forest.

3. **Handling Text Data**:
   - Naive Bayes is probably the best performer when it comes to text data, using conditional probabilities directly, thus making a good classifier for spam emails.

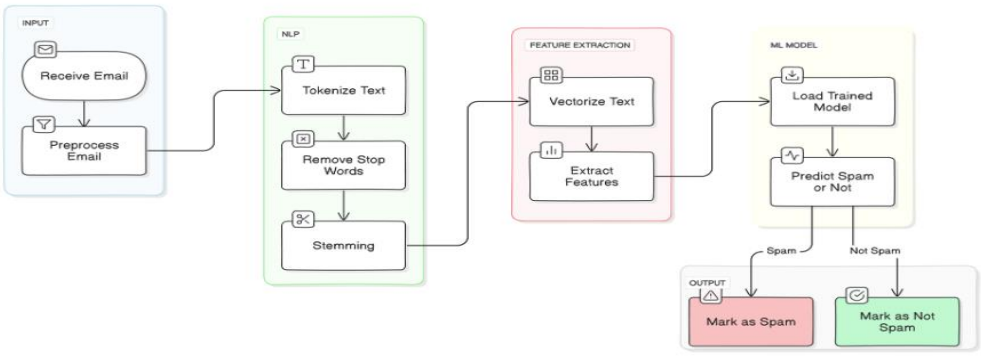
4. **Resistance to Overfitting:**
   - Where Decision Trees overfit, Naive Bayes makes the assumption of independence among features, which makes it often possible to generalize better.

5. **Efficiency for Sparse Features**:
   - Works great with sparse representations like word frequency.

CHAPTER 3

Proposed Methodology

3.1 System Design



Here it shows the architecture of the model that detects whether the model is spam or not by using the naïve bayes algorithm .we first train the model with some dataset that consist of lable data that classifies as spam or ham(not spam).When we give the input i.e email to the model it first perform the preprocessing using the NLP techniques like tokenization of text, Removes the stop Words and Stemming .After preprocessing we perform the feature extraction like vectorization of text that converts the text into numerical data after that the model checks and compare with the emails according to its features and it predicts whether the email is spam or not .In this way the model predicts whether the email is spam or not the spam mail

3.2Requirement Specification

3.2.1Hardware Requirements:

A basic PC or personal computer is as much good for a simple spam mail classification project with the Naive Bayes algorithm and machine learning. However, as the complexity of the model and, ultimately, the size of the dataset grow that simple hardware might not suffice for some classifications. Here is a minimum requirement for the hardware:

Processor: Intel Core i5 or the equivalent

RAM: 16GB

Storage: 256GB SSD

Operating System: Windows 11

3.2.2 Software Requirements:

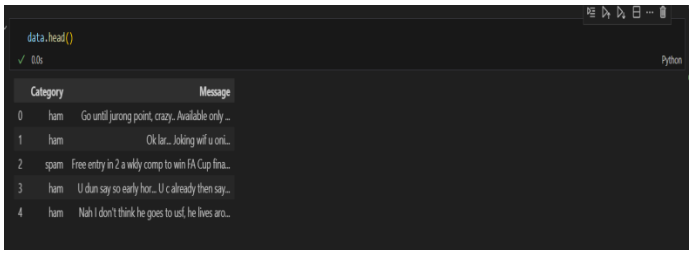
Programming Language : Python 3.X
Environment: Anaconda
Data Analysis and Machine Learning Libraries: Pandas, Numpy, Scikit-learn, ipykernel
IDE: VS code
UI: Streamlit

CHAPTER 4

Implementation and Result

4.1Snap Shots of Result:

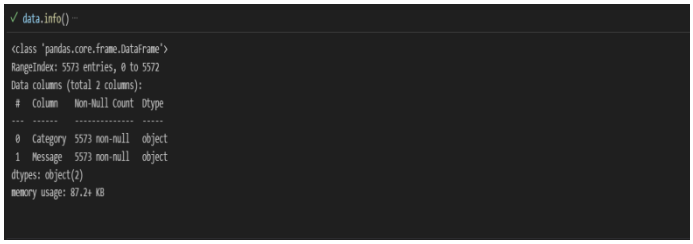
Figure4.1.1-Data head



In the above it show the head of the data i.e first 5

Rows of the dataset

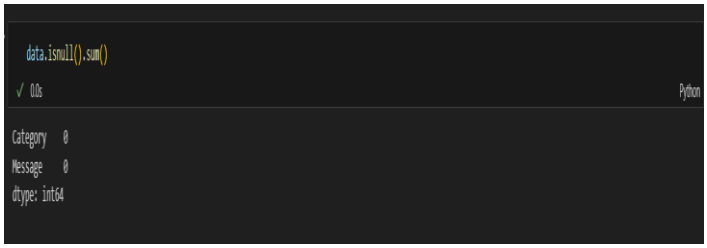
Figure 4.1.2-Data Information



In this it shows the information regarding the dataset

Like range , datatype, memory etc

Figure-4.1.3-Isnull()



Here it shows there is no null values in the available columns

Figure-4.1.4 -Accuracy

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split

# Assuming 'Category' is the target variable (spam or not)
y = data['Category']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_transformed, y, test_size=0.2, random_state=42)

# Initialize and train the Naive Bayes model
model = MultinomialNB()
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)

# Check accuracy
from sklearn. metrics import accuracy_score
print("Accuracy:", accuracy_score(y_test, y_pred))
```

✓ 0.0s

```
Accuracy: 0.9811659192825112
```

Here we used Multinomial Naivebayes algorithm for spam mail classification and we got an accuracy of 0.98

Figure-4.1.5-Result of the model as Spam Mail

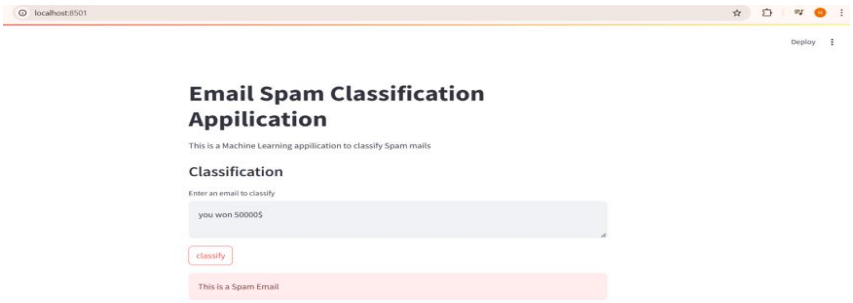
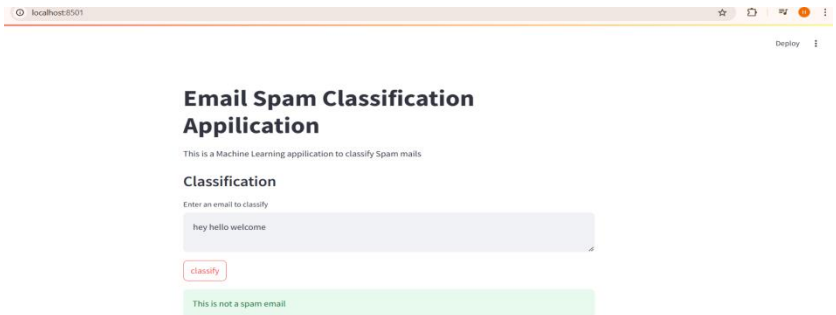


Figure-4.1.6-Result of the model as Not Spam Mail



4.2GitHub Link for Code:

https://github.com/DivvelaHemaHarshini/Spammail_Classification

4.2.1Streamlit Link:

https://spammailclassification-z6ckqhljt9792ovzgfkk2w.streamlit.app/

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

1. Improvement of Feature Engineering

The use of n-grams:

Do not consider just simply words but bi-grams and tri-grams as it helps capture more context from the email text.

Semantic Representations:

Use word embeddings (Word2Vec, GloVe, etc.) for the representation of words semantically rather than through frequency-based methods.

Metadata Incorporation:

Email Metadata such as sender reputation, domain analysis, and email headers can feed textual features.

2. Hybrid Models and Ensemble Techniques

Naive Bayes with Boosting:

Improving Naive Bayes with boosting algorithms such as AdaBoost or Gradient Boosting in an attempt to reduce bias and variance.

Stacking with Deep Learning Models:

Combine Naive Bayes and deep learning models such as CNNs or RNNs for handling more complicated patterns.

Artificial Bee Colony Optimization: Apply optimization techniques such as the Artificial Bee Colony algorithm to optimize and fine-tune the feature selection and classification thresholds.

3. Continuous Learning

Incremental training:

Update periodically through new data incorporation so that the model would reinvent itself with new spam patterns.

Active learning:

Active learning pertaining to labeling for uncertain cases to improve the model's learning.

4. **Robustness and Security**

Adversarial Training:

Train with adversarial samples to prepare against attacks that try to work around spam filters.

Federated Learning:

Update the model through federated learning so that it does not other share data with the users.

5. Interpretableness and User Feedback Integration

Explainable AI:

Explainable Naive Bayes to explain why an email is detected as spam or not.

User Feedback Mechanism:

Users can feedback on emails misclassified in terms of spam and non-spam for future updating of the model.

6. **Handling High-Dimensional Data**

Dimensionality Reduction: Using techniques such as PCA or t-SNE

5.2 Conclusion:

Naive Bayes has solved many problems of these existing methods, like imbalanced datasets, new trendy model spam, and quick online detection, for its project E-mail classification. They're really tough features considering the whole problem. Project in discussion really has made spam detection probabilistic yet has increased the accuracy and efficiency.

Besides that, the project tells which phenomena need continuous improvement such as incorporating or learning new techniques for spam detection with time, like continue applying best forms of deep learning and NLP.

It would surely direct how this program will likely establish a platform over which special spam filtering devices can effectively work against innovative ones. It will do so as according to the total contribution in suggesting innovations such as end-user engagement in real-time adaptive feedback systems for a more customized and accurate detection system. It would thus provide an infrastructure for constructing intelligent, cost-effective spam classification systems which will learn and adapt as time proceeds to newer patterns while addressing most contemporary and expected challenges concerning email security.

Thus, guiding some more approaches like Natural Language Processing and deep learning would give the project a further insight into what the differences of variations between spam and legitimate email are and finally could afford more satisfaction to the customer by reducing high levels of false positives.

The future spam filter would grant validity in an ever-evolving tactic across spammers'. Therefore, it fulfills the entire domain by giving ideas for innovations regarding user involvement in feedback systems and adaptability.

REFERENCES

[1] https://www.kaggle.com/datasets/ashfakyeafi/spam-email-classification

[2] Gangavarapu, T., Jaidhar, C. D., & Chanduka, B. (2020, February 22).Applicability of machine learning in spam and phishing email filtering: review and approaches. Artificial Intelligence Review; Springer Science+Business Media. https://doi.org/10.1007/s10462-020-09814-9