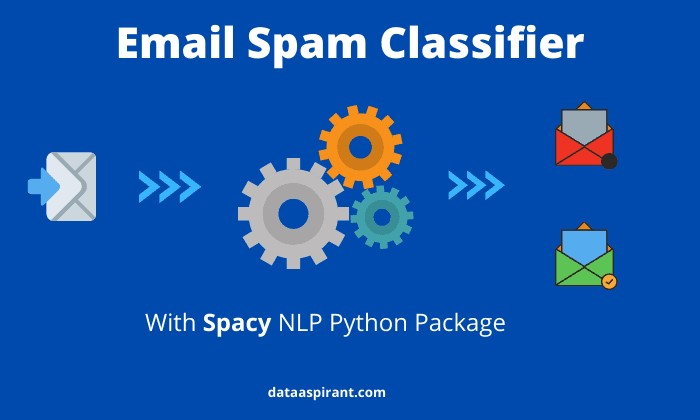
## BUILDING A SMARTER AI-POWERED SPAM CLASSIFIER

***Phase 2: Innovation***

***Project:Building a smarter AI-powered spam classifier***

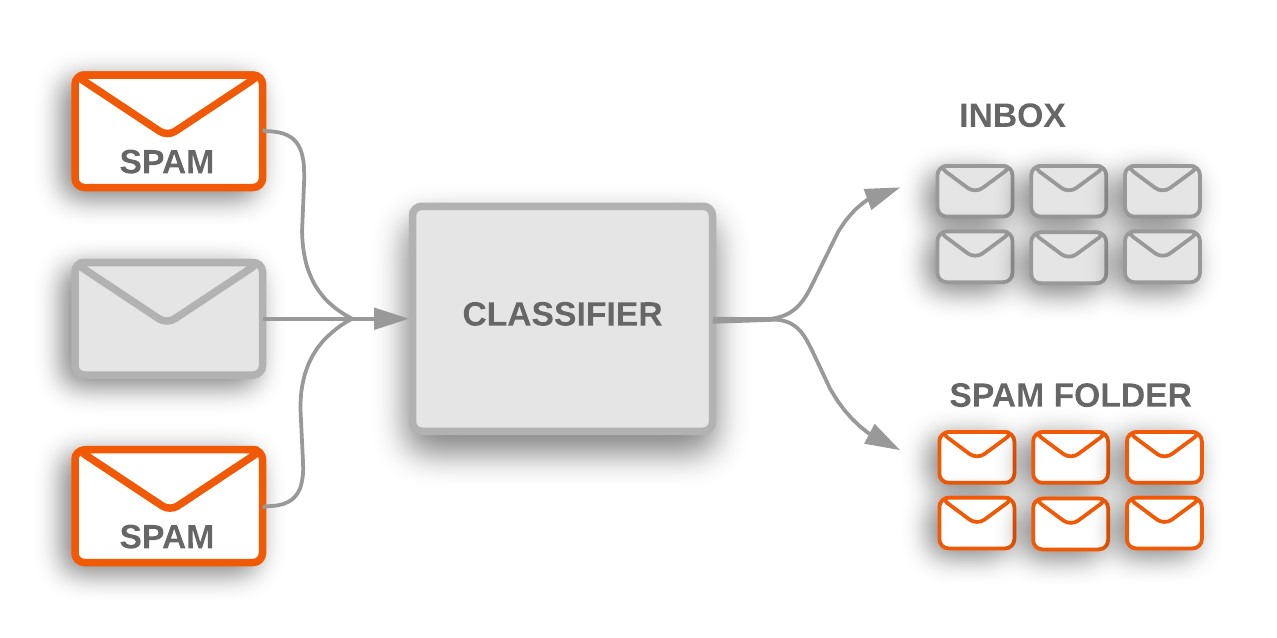


***INTRODUCTION:***

In our digital age, where messages flood our inboxes and communication channels, distinguishing between genuine messages and unwanted spam has never been more critical. To tackle this challenge, we turn to artificial intelligence, which empowers us to create smarter spam classifiers. These intelligent systems aim to keep our digital spaces clutter-free, ensuring we interact with only the messages that truly matter. In this exploration, we'll uncover the essential steps and principles behind constructing a more intelligent spam classifier, one that adapts to evolving spam tactics and safeguards the quality of our digital communication.

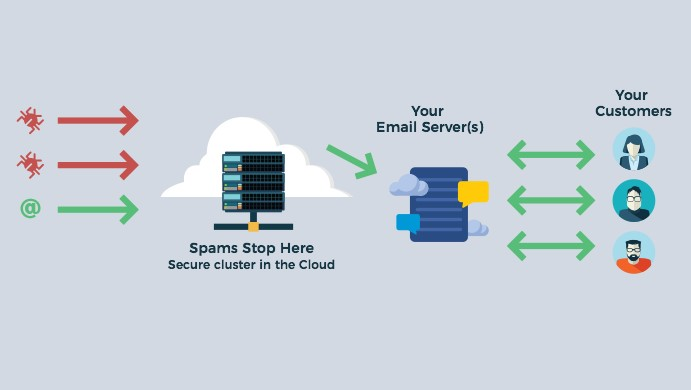
A spam classifier is an artificial intelligence system designed to differentiate between legitimate, or "ham," messages and unwanted, unsolicited, and potentially harmful "spam" messages. The goal of a spam classifier is to filter out spam, allowing users to focus on relevant and safe communication.

***Innovation of Spam classifier:***



***Key Objectives:***

In building a smarter AI-powered spam classifier, the primary objectives are clear:



1. ***High Accuracy:***

The classifier should accurately distinguish between spam and legitimate messages to minimize false positives and false negatives.

1. ***Adaptability:***

The system should continuously learn and adapt to new spam patterns and techniques, ensuring its long-term effectiveness.

1. ***Efficiency:***

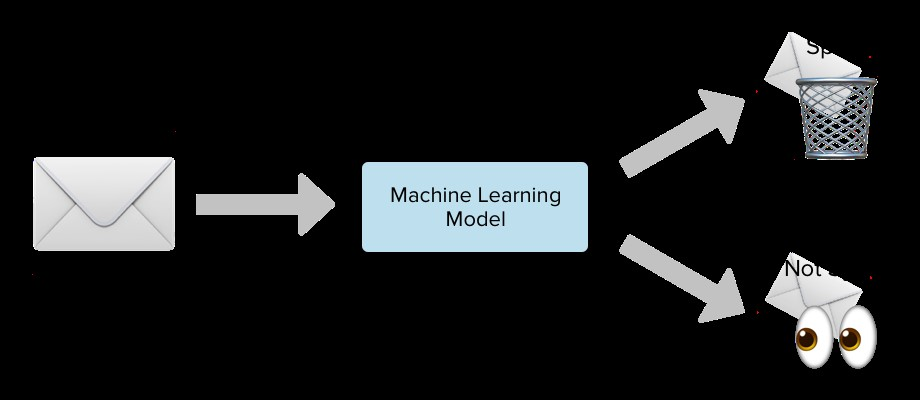
The classifier must be capable of processing a high volume of incoming messages in real-time or near-real-time without compromising performance.

1. ***Ethical and Legal Compliance:***

It's crucial that the classifier respects user privacy and complies with data protection regulations.

**Building a smarter AI-powered spam classifier involves several steps.** This guide outlines the key steps to help you design and implement such a system:

***DESIGN THINKING:***



1. ***Define the Problem:***

- Clearly define the problem you want to solve. In this case, the problem is spam classification.

***2. Data Collection:***

- Gather a substantial amount of data for training and testing your AI model. This data should include both spam and non-spam (ham) examples. You can use publicly available datasets or collect your own data.

***3. Data Preprocessing:***

- Clean and preprocess the data, including:

- Text normalization: Convert text to lowercase, remove punctuation, and handle special characters.

- Tokenization: Split text into words or tokens.

- Stop word removal: Eliminate common words that don't carry much meaning.

- Lemmatization or stemming: Reduce words to their base form.

- Vectorization: Convert text data into numerical form (e.g., TF-IDF or word embeddings).

***4. Data Splitting:***

- Split your dataset into training, validation, and testing sets. Common splits are 70% for training, 15% for validation, and 15% for testing.

***5. Feature Engineering:***

- Experiment with various features, such as n-grams, word embeddings, or even metadata like sender information.

***6. Model Selection:***

- Choose a machine learning or deep learning algorithm for your spam classifier. Common choices include:

- Naive Bayes

- Support Vector Machines (SVM)

- Decision Trees

- Random Forests

- Recurrent Neural Networks (RNNs)

- Convolutional Neural Networks (CNNs)

- Transformer-based models (e.g., BERT or GPT)

***7. Model Training:***

- Train your chosen model on the training data. Experiment with different hyperparameters to optimize performance. Consider techniques like cross-validation.

***8. Evaluation Metrics:***

- Select appropriate evaluation metrics for your problem, such as accuracy, precision, recall, F1-score, and ROC-AUC. These metrics will help you measure the performance of your classifier.

***9. Model Evaluation:***

- Evaluate your model on the validation set and fine-tune it based on the results. Repeat this process until you're satisfied with the performance.

***10. Hyperparameter Tuning:***

- Use techniques like grid search or random search to find the best hyperparameters for your model.

***11. Model Testing:***

- Test your final model on the separate testing dataset to assess its real-world performance.

***12. Model Deployment:***

- Deploy your model in a production environment. You may need to build an API to make predictions accessible to other systems or users.

***13. Monitoring and Maintenance:***

- Continuously monitor the model's performance in the real world. Implement strategies to retrain the model periodically to adapt to changing spam patterns.

***14. User Feedback Loop:***

- Incorporate user feedback to improve the classifier. Users can flag false positives and false negatives, helping you make the model smarter over time.

***15. Security and Privacy:***

- Ensure that the data you collect and the predictions made by your model comply with security and privacy regulations.

***16. Scale and Optimization:***

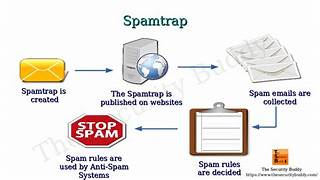
- As your system grows, optimize its performance and scalability. Consider distributed computing and parallel processing to handle large volumes of data.

***17. Documentation and Reporting:***

- Document your model's architecture, data sources, and methodologies. Provide clear reporting on its performance and any biases that may arise.

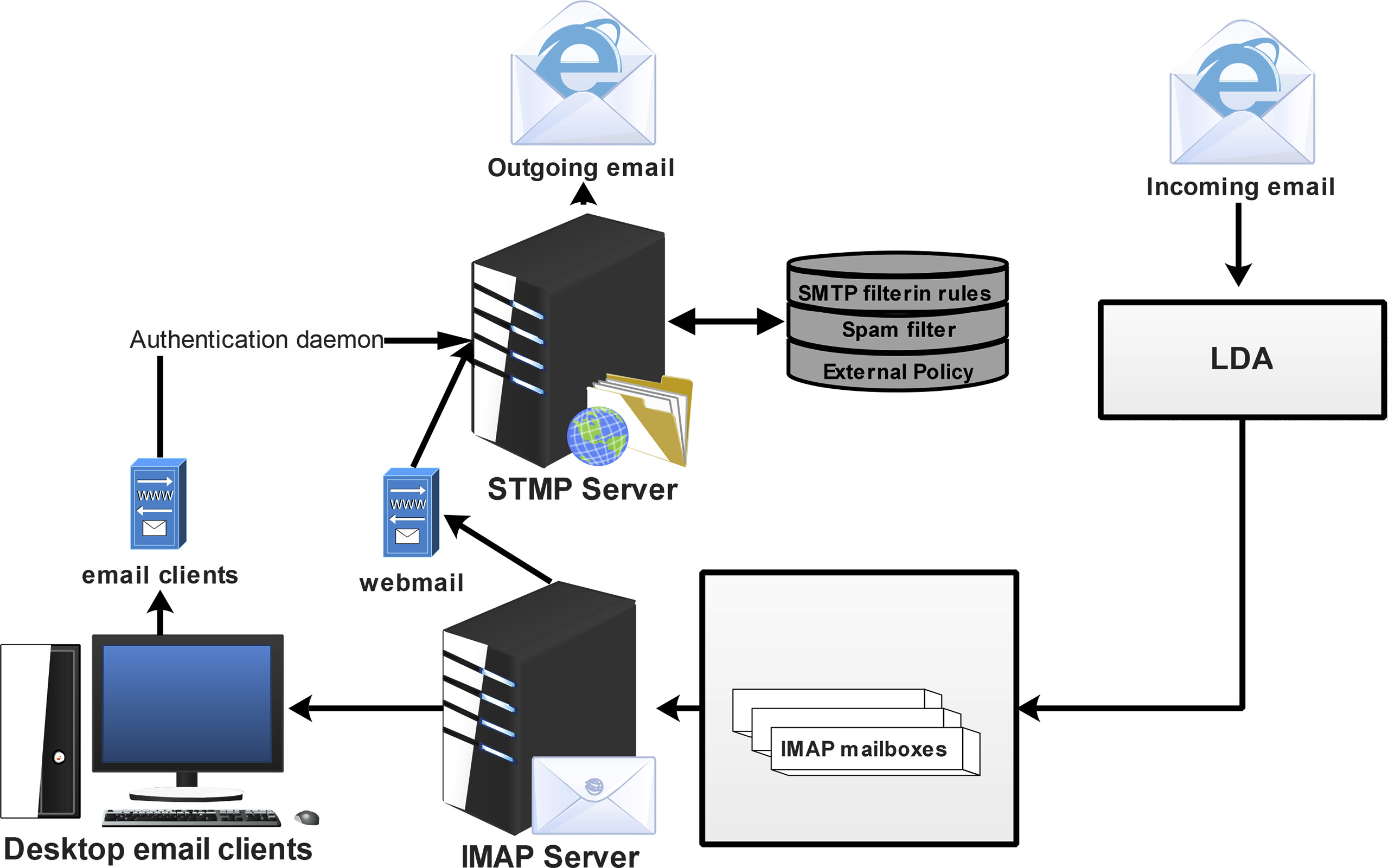
***18. Ethical Considerations:***

- Be aware of potential biases and ethical concerns in your data and model, and take steps to mitigate them.



Building a smarter AI-powered spam classifier is an ongoing process. Regular updates, retraining, and adaptation to new spam tactics are essential to maintain its effectiveness. Additionally, consider the ethical and privacy aspects of handling user data and classify messages while implementing security measures to protect sensitive information.

***Solve the problem:***



Solving the problem of building a smarter AI-powered spam classifier requires a systematic approach and careful consideration of various aspects. Here's how to address the key challenges:

***1. Data Quality and Quantity:***

- Problem: Insufficient or low-quality training data can hinder the effectiveness of your spam classifier.

- Solution: Collect a diverse and representative dataset of spam and ham messages. Ensure data cleanliness, and consider data augmentation techniques.

***2. Feature Selection and Engineering:***

- Problem: Inadequate or irrelevant features may lead to suboptimal performance.

- Solution: Experiment with different features and data representations (e.g., TF-IDF, word embeddings, metadata). Choose features that capture the essence of spam messages effectively.

***3. Model Selection:***

- Problem: Selecting the right algorithm or architecture for your AI model can be challenging.

- Solution: Evaluate various machine learning and deep learning models (e.g., Naive Bayes, SVM, neural networks). Choose the one that performs best for your specific dataset.

***4. Hyperparameter Tuning:***

- Problem: Poorly tuned hyperparameters can lead to subpar model performance.

- Solution: Use techniques like grid search or random search to find the best hyperparameters for your model. Optimize for metrics like precision, recall, or F1-score.

***5. Evaluation Metrics:***

- Problem: Choosing the wrong evaluation metrics may misrepresent your classifier's performance.

- Solution: Select appropriate metrics (e.g., precision, recall, F1-score) based on your problem's specific requirements. Consider a balance between false positives and false negatives.

***6. Overfitting and Generalization:***

- Problem: Overfit models perform well on the training data but poorly on new, unseen data.

- Solution: Regularize your model, use techniques like dropout, and consider early stopping to prevent overfitting. Cross-validation helps assess generalization.

***7. Ethical and Privacy Concerns:***

- Problem: Handling user data and filtering messages must be done with ethical and privacy considerations.

- Solution: Anonymize user data, respect privacy regulations, and be transparent about data usage. Implement measures to mitigate biases in your classifier.

***8. User Feedback and Continuous Improvement:***

- Problem: Failing to incorporate user feedback and adapt to evolving spam tactics can lead to reduced effectiveness.

- Solution: Develop a feedback loop for users to report false positives and false negatives. Continuously monitor and retrain your model to stay ahead of new spam techniques.

***9. Scalability and Deployment:***

- Problem: Deploying the model in a production environment can be complex.

- Solution: Build an efficient deployment pipeline, consider distributed computing for scalability, and implement security measures to protect user data.

***10. Monitoring and Maintenance:***

- Problem: Neglecting model monitoring can result in performance degradation over time.

- Solution: Regularly monitor your model's performance, and set up alerts for anomalies. Plan for periodic retraining and updates to adapt to changing spam patterns.

By addressing these challenges systematically and continuously improving your spam classifier, you can create a smarter AI-powered system that effectively filters out unwanted messages while preserving the quality and security of digital communication.

*Program:*

***Input:***

import numpy as np

import pandas as pd

import osfor dirname, \_,

filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

***Output:***

/kaggle/input/sms-spam-collection-dataset/spam.csv

***Input:***

import pandas as pd

df = pd.read\_csv("/kaggle/input/sms-spam-collection-dataset/spam.csv", sep='**\t**', encoding='ISO-8859-1')

df[['label', 'message']] = df['v1,v2,,,'].str.split(',', n=1, expand=True)

df.drop('v1,v2,,,', axis=df.head(10)

***Output:***

|  | label | message |
| --- | --- | --- |
| 0 | ham | "Go until jurong point, crazy.. Available only... |
| 1 | ham | Ok lar... Joking wif u oni...,,, |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | "Nah I don't think he goes to usf, he lives ar... |
| 5 | spam | "FreeMsg Hey there darling it's been 3 week's ... |
| 6 | ham | Even my brother is not like to speak with me. ... |
| 7 | ham | As per your request 'Melle Melle (Oru Minnamin... |
| 8 | spam | WINNER!! As a valued network customer you have... |
| 9 | spam | Had your mobile 11 months or more? U R entitle... |

***Input:***

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras

import layers

from sklearn.model\_selection

import train\_test\_split

from sklearn.feature\_extraction.text

import TfidfVectorizer

from sklearn.naive\_bayes

import MultinomialNB

from sklearn.metrics

import classification\_report,accuracy\_score

from sklearn.metrics

import confusion\_matrix

from tensorflow.keras.layers

import TextVectorization

from sklearn.metrics

import precision\_score, recall\_score, f1\_score

import tensorflow\_hub as hub

path = '/kaggle/input/sms-spam-collection-dataset/spam.csv'df = pd.read\_csv(path, encoding = 'latin-1')df.head()

***Output:***

|  | v1 | v2 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 |
| --- | --- | --- | --- | --- | --- |
| 0 | ham | Go until jurong point, crazy.. Available only ... | NaN | NaN | NaN |
| 1 | ham | Ok lar... Joking wif u oni... | NaN | NaN | NaN |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... | NaN | NaN | NaN |
| 3 | ham | U dun say so early hor... U c already then say... | NaN | NaN | NaN |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... | NaN | NaN | NaN |

df = df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis = 1)

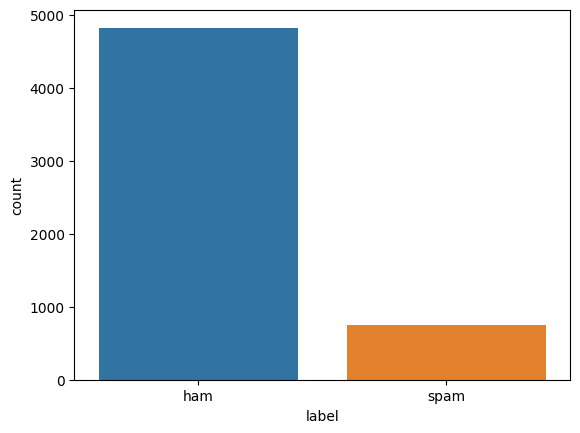
df.columns = ['label', 'text']

df['label\_enc'] = df.label.map({'ham': 0, 'spam': 1})

sns.countplot(x = df.label)

***Output:***

<Axes: xlabel='label', ylabel='count'>



avg\_words\_len=round(sum([len(i.split()) for i **in** df['text']])/len(df['text']))avg\_words\_len

s = set()for sent **in** df['text']:

for word **in** sent.split():

s.add(word)total\_words\_length=len(s)print(total\_words\_length)

X, y = np.asanyarray(df['text']), np.asanyarray(df['label\_enc'])

new\_df = pd.DataFrame({'text': X, 'label': y})X\_train, X\_test, y\_train, y\_test = train\_test\_split(

new\_df['text'], new\_df['label'], test\_size=0.2, random\_state=42)X\_train.shape, y\_train.shape, X\_test.shape, y\_test.shape

***Output:***

((4457,), (4457,), (1115,), (1115,))

***Input:***

tfidf\_vec = TfidfVectorizer().fit(X\_train)X\_train\_vec,X\_test\_vec = tfidf\_vec.transform(X\_train),tfidf\_vec.transform(X\_test)

baseline\_model = MultinomialNB()baseline\_model.fit(X\_train\_vec,y\_train)

***Output:***

MultinomialNB

***Input:***

y\_pred = baseline\_model.predict(X\_test\_vec)

nb\_accuracy = accuracy\_score(y\_test, y\_pred)

print(nb\_accuracy)

print(classification\_report(y\_test, y\_pred))

***Output:***

0.9623318385650225

precision recall f1-score support

0 0.96 1.00 0.98 965

1 1.00 0.72 0.84 150

accuracy 0.96 1115

macro avg 0.98 0.86 0.91 1115

weighted avg 0.96 0.96 0.96 1115

***Input:***

confusion\_matrix(y\_test, y\_pred)

***Output:***

array([[965, 0],

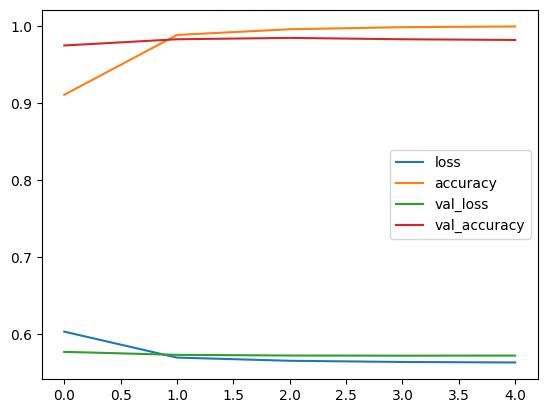
[ 42, 108]])

***Input:***

pd.DataFrame(history\_1.history).plot()

***Output:***

<Axes: >

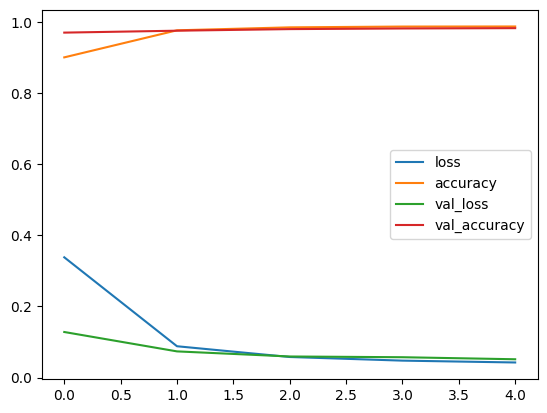


***Input:***

pd.DataFrame(history\_3.history).plot()

***Output:***

<Axes: >

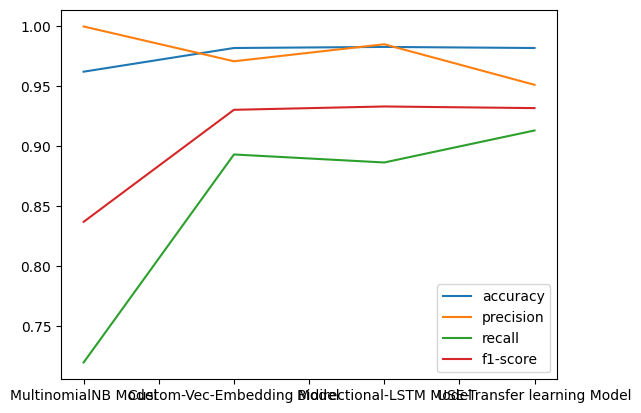


***Input:***

total\_results.plot()

***Output:***

<Axes: >



***Conclusion:***

In a world inundated with digital communication, the development of a smarter AI-powered spam classifier has become a vital defense against the incessant deluge of spam. As we conclude this exploration of building such a system, it's crucial to emphasize the significance of this technology in today's digital landscape.

Spam classification is not just about inbox cleanliness; it's about ensuring the security, efficiency, and relevance of digital communication. Here are the key takeaways from our journey:

1. ***AI's Role in Spam Classification:***

Artificial intelligence, particularly machine learning and deep learning, has revolutionized spam classification. These technologies provide the adaptability and intelligence required to combat evolving spam tactics.

1. ***Data Is Key:***

Building an effective spam classifier starts with high-quality data. Data collection, preprocessing, and feature engineering are fundamental steps in the process. The richer and more diverse your dataset, the smarter your AI model can become.

1. ***Model Selection and Training:***

Choosing the right algorithm and carefully tuning hyperparameters is essential for model success. The training process is iterative and often requires fine-tuning to achieve high accuracy.

1. ***Evaluation and Deployment***:

Robust evaluation metrics help you understand how well your spam classifier is performing. Deployment involves integrating your model into a production environment, ensuring that it operates efficiently and effectively.

1. ***Monitoring and Continuous Improvement:***

The battle against spam is ongoing. Continuous monitoring of your system is necessary to adapt to new spam tactics and maintain its effectiveness. User feedback plays a crucial role in this iterative process.

1. ***Ethical and Privacy Considerations:***

As we harness the power of AI to filter messages, we must be vigilant about privacy and ethical concerns. Respect user privacy, and address any potential biases that may arise.

***6.Scale and Security:***

As your system grows, optimize its performance and security. Protect sensitive user data and ensure it complies with relevant regulations.

In conclusion, building a smarter AI-powered spam classifier is a dynamic process that leverages cutting-edge technology to enhance the quality and security of digital communication. It is not just a tool to declutter inboxes; it's a safeguard for individuals and organizations against unwanted and potentially harmful messages.

By undertaking this journey and following the steps outlined in this guide, you are well-equipped to create an effective and adaptable spam classifier. In doing so, you contribute to a safer and more efficient digital world, where the benefits of communication can thrive, free from the clutter of spam.

***\*\*\* thanking you\*\*\****