NAANMUDHALVAN

Phase5

Name : SHYAM.T

Register number : 210821243056

College : Kings Engineering College

Domain : Artificial Intelligence

Project : Building a Smarter AI-Powered Spam Classifier

**Program:**

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score

from sklearn.linear\_model import LogisticRegression

import pandas as pd

data = pd.read\_csv("/content/spam collection.csv")

X = data['EmailText']

y = data['Label']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

vectorizer = CountVectorizer()

X\_train\_counts = vectorizer.fit\_transform(X\_train)

X\_test\_counts = vectorizer.transform(X\_test)

model = MultinomialNB()

model.fit(X\_train\_counts, y\_train)

y\_pred = model.predict(X\_test\_counts)

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

classifier = LogisticRegression(solver='liblinear')

classifier.fit(X\_train\_counts, y\_train)

y\_pred = classifier.predict(X\_test\_counts)

print("Accuracy: ", accuracy\_score(y\_test, y\_pred))

print("Precision: ", precision\_score(y\_test, y\_pred))

print("Recall: ", recall\_score(y\_test, y\_pred))

print("F1-score: ", f1\_score(y\_test, y\_pred))

**Design Thinking Process:**

**Understand the Problem:**

* Identify the characteristics of spam messages.
* Analyze existing spam filters and their limitations.

**Define Objectives:**

* Improve spam classification accuracy.
* Minimize false positives (legitimate messages classified as spam) and false negatives (spam messages not detected).

**Data Collection:**

* Gather a labeled dataset of messages, including both spam and non-spam messages.

**Data Preprocessing:**

* Data Cleaning:
* Remove duplicate messages.
* Handle missing or noisy data.
* Text Preprocessing:
* Tokenization: Split messages into words or phrases.
* Stopword removal: Eliminate common and uninformative words.
* Lemmatization or stemming: Reduce words to their base forms. ● Handle special characters, URLs, and numbers.

**Feature Extraction:**

* Bag of Words (BoW): Create a vector representation of messages.
* TF-IDF (Term Frequency-Inverse Document Frequency): Assign importance scores to words based on their frequency.
* Word Embeddings (e.g., Word2Vec, GloVe): Represent words as dense vectors.
* N-grams: Capture word sequences and context.

**Model Selection:**

* Choose machine learning algorithms suitable for text classification:
* Logistic Regression, Naive Bayes, Support Vector Machines, Decision Trees, Random Forest, Gradient Boosting, Neural Networks, etc.

**Model Training:**

* Split the dataset into training, validation, and test sets.
* Train the chosen model(s) using the training data.
* Tune hyperparameters using the validation set to optimize model performance.

**Evaluation Metrics:**

* Use appropriate metrics to evaluate the model:
* Accuracy, Precision, Recall, F1-score, ROC-AUC, Confusion Matrix.
* Optimize for precision and recall, depending on the specific requirements (e.g., minimize false positives or false negatives).

**Innovative Techniques:**

1. Ensemble Methods:
   * Combine multiple models (e.g., Random Forest, Gradient Boosting) to improve performance and reduce overfitting.
2. Deep Learning:
   * Experiment with neural networks, such as LSTM or CNN, to capture complex patterns in text data.
3. Active Learning:
   * Implement an active learning approach to iteratively improve the model by selecting the most informative samples for labeling.
4. Explainability:
   * Utilize techniques like SHAP values or LIME to explain model predictions and build user trust.

**Phases of Development:**

Data Collection and Preprocessing:

* + Collect and clean the dataset.
  + Preprocess the text data.

Feature Extraction:

* + Extract relevant features from text.

Model Selection:

* + Choose appropriate machine learning algorithms or deep learning architectures.

Model Training and Tuning:

* + Train the model on the preprocessed data.
  + Tune hyperparameters for optimal performance.

Evaluation:

* + Evaluate the model's performance using appropriate metrics.

Iteration:

* + Implement innovative techniques and iterate on the model for improvements.

Deployment:

* + Deploy the spam classifier as a service or integrate it into an existing system.

Monitoring and Maintenance:

* + Continuously monitor the model's performance and retrain as needed.