TASK 3

By- PINKY PAMECHA

PROBLEM STATEMENT:

We have unlabelled set of paragraphs , messages and emails.We have to identify which of the given blocks of texts are spam texts.

1.IMPORTING LIBRARIES

import numpy as np

import pandas as pd

import matplotlib.pyplot as pd

from nltk

import wordcloud from WordCloud

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.cluster import KMeans

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import classification\_report

2.EXTRACTING DATA

3.TRY UNDERSTAND SOME COMMON WORDS/PATTERN OF SPAM and NON-SPAM MESSAGES

We can make word cloud for visual representation of words that have high frequency in data for basic data analysis.

4.PREPROCESSING:

1)Text Lowercasing: Convert all text to lowercase to ensure uniformity in the data.

2)Remove Non-ASCII Characters: Remove non-ASCII characters, including emojis and special symbols, as they might not be relevant to spam classification.

3)Tokenization: Split the text into individual words (tokens) to prepare them for further analysis.

4)Stopword Removal: Remove common stopwords (e.g., "the," "and," "is") as they don't carry much information for spam detection.

5)Stemming or Lemmatization: Reduce words to their base form to handle variations of the same word (e.g., "running" -> "run").

https://www.shiksha.com/online-courses/articles/text-pre-processing-for-spam-filtering/

5.FEATURE EXTRACTION:

After preprocessing, you need to convert the text data into numerical form that machine learning algorithms can work with.

We can do it by:

1)CountVectorizer

vectorizer = CountVectorizer(lowercase=True, stop\_words='english')

matrix = vectorizer.fit\_transform(data)

2)TfidfVectorisation

vectorizer = TfidfVectorizer(lowercase=True, stop\_words='english')

tfidf\_matrix = vectorizer.fit\_transform(data)

6.TECHNIQUE/MODEL SELECTION

UNSUPERVISED MACHINE LEARNING

Unsupervised learning (UL) is a machine learning algorithm that works with datasets without labeled responses. It is most commonly used to find hidden patterns in large unlabeled datasets through cluster analysis. Its basic idea is to group elements based on their similarity. Machine learning models can evaluate and group similar elements even without the labels.

Let’s say that you have a list of 100.000 subscription emails. They have no labels so you use clustering to group them by the number of recipients and the frequency of response activity

Since the data is unlabelled and you want to identify spam texts, you can approach this problem as an anomaly detection task or an unsupervised clustering task. Here's how:

Anomaly Detection: Train a model to identify instances that are significantly different from the majority of the data. Spam texts are likely to be anomalies in the context of the overall text distribution.

Clustering: Use clustering algorithms (e.g., K-means) to group similar texts together. Spam texts might form distinct clusters due to their content being different from legitimate texts.

num\_clusters = 2

kmeans = KMeans(n\_clusters=num\_clusters, random\_state=42)

cluster\_labels = kmeans.fit\_predict(tfidf\_matrix)

SEMI-SUPERVISED LEARNINGS:

Given that you only have unlabelled data, a semi-supervised learning approach can be used. Start by manually labeling a small subset of your data as spam or non-spam. This labeled data can be used to train a model initially.

Implement active learning strategies where the model can actively query us to label certain examples. The model can be uncertain about the classification of certain texts, and by labeling these examples, you gradually improve the model's performance.

Since you don't have initial labeled data, you can split your data into a training set (a small portion initially) and a validation set. Train your model on the training set and evaluate it on the validation set. As you iteratively label more data, you can increase the size of your training set and update the model accordingly.

Manually inspect the contents of the documents in each cluster. Since we don't have initial labels, this step involves labeling the clusters based on the content you observe. For instance, if one cluster contains messages promoting suspicious products or requests for personal information, you might label it as "Potential Spam."

Use the algorithms of unsupervised learning to simplify your unlabeled data or group it in accordance to your goals. Principles of unsupervised machine learning can be used even for the labeled datasets to preprocess them before supervised learning begins.

Combine the elements of unsupervised and supervised learning in a semi-supervised learning model. This approach will train your AI to maximize the annotation process from the small sample onto a large set of unlabeled data, saving your resources and time while developing more secure and robust AI.

, you could use a simple classifier like Naive Bayes or a more complex model like a Random Forest or a Neural Network.

classifier = MultinomialNB()

classifier.fit(vectorizer.transform(train\_features), train\_labels)

predictions = classifier.predict(vectorizer.transform(validation\_features))

print(classification\_report(validation\_labels, predictions))

Potential Challenges and Shaky Scenarios:

Evolution of Spam: Spam tactics change over time, and a model trained on historical data might not perform well on new forms of spam.

Human Bias: Active learning depends on human labeling, which can introduce bias and errors.

Contextual Understanding: Some spam messages might be context-dependent and require a deeper understanding of cultural or situational nuances.

To mitigate these challenges, it's important to regularly update the model, gather more labeled data, and regularly monitoring the model's performance and adapting to new trends in spam can also help maintain its accuracy over time.

Unlabelled Data: The absence of labelled data means you won't have a ground truth to evaluate your model's performance. You would need to rely on domain knowledge and manual inspection to verify the model's findings.

Mixed Content: Some legitimate messages might share characteristics with spam (e.g., promotional emails). The model might misclassify such instances.