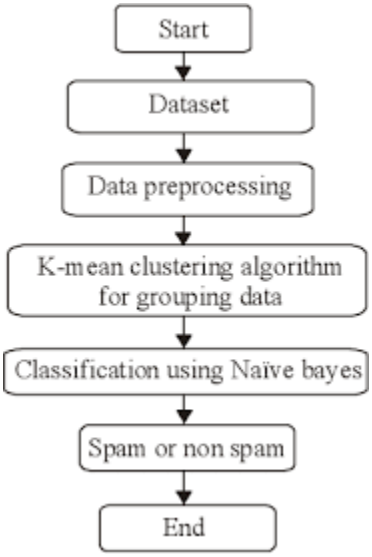
Task 3:

You are given a set of paragraphs, messages and emails. This data is unlabelled. You are to identify which of the given blocks of texts are spam texts. Explain how you would go about it, the preprocessing (removal of certain words and non-ASCII characters like emojis, etc) steps you would take, the method you would use to categorize data into spam or not spam and any scenarios where you think your method is shaky. Explanation of your thought process is a must.

Solution:

Spam filtering models use a wide variety of machine learning models and techniques. A good pipeline to do this is:

1. Pre-Processing
2. Feature Extraction
3. Clustering
4. Classification



Preprocessing:

Data preprocessing is the process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. While preprocessing data for a spam identification model, we must ensure that there aren't any characters, words, etc., that will influence the model counter – productively. Also, it is important to get rid of such words which are not necessary for the model's working. This increases, by far, the speed of the model, which is an important feature.

The various ways that data id formatted during pre – processing:

1. Removing Punctuation: Punctuation marks such as '.', ';', '!', etc., are removed from the data. This is done because these characters do not provide much functionality to the model and hence reduce the speed and efficiency of the model.

2. Removing Non-ASCII Characters: Non-ASCII characters such as emojis, special characters are removed because they are of no use to the compiler as it does not understand such symbols.

3. Lowercasing: All text is converted to lowercase as this helps keep data uniform and prevents errors such as giving higher bias to uppercase letters, etc.

4. Tokenization: The text is split into individual words or tokens. These tokens are integral parts of our model, and the success of the model is dependent on these.

5. Removing Stop Words: Common words like "and," "the," "is," etc., are removed as they typically don't carry meaningful information for classification. Removal of such words reduces the space occupied by the data and makes the classification much more efficient.

6. Stemming: In this process, different forms of the same word are grouped together. For example, plurals, gerund forms, tenses etc. For example, 'group', 'groups', and 'grouped' would all be considered as 'group'.

Feature Extraction:

Feature is the step after pre – processing. Machine learning models do not understand data in the form of words. Thus, we need to encode our data in the form of numerical features that the machine can understand. There are several feature extraction methods that can be used:

1. Bag of Words: Bag of words technique is the easiest to learn and implement. In this technique, the dataset is considered as a "bag" of words. Hence, the encoding of words depends only on frequency of words and not on their order.

We can directly use CountVectorizer class by Scikit-learn to implement bag of words technique.

The main disadvantage of bag of words is that it doesn't consider new words that are from outside the "bag" and that it doesn't consider word order.

2. Term frequency – Inverse Document Frequency (TF-IDF): In this method, a weighting scheme is applied to give more weight to words that are informative and less common in the entire dataset. It does this by calculating the frequency of a word in a particular message relative to a collection of messages. The following mathematical steps are followed to achieve this:

(I) Term Frequency (TF):

The number of times a word appears in a message is divided by the total number of words in that message. 0 < TF< 1

$$tf_{ij} = \frac{\text{Number of times term i appears in the document}}{\text{Total Number of terms in the document j}}$$

(II) Inverse Document Frequency (IDF):

The logarithm of the number of messages in the data frame is divided by the number of messages where the specific term appears.

$$idf_i = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term i in it}}\right)$$

3. Word Embedding: Word embeddings, such as Word2Vec, GloVe, and fastText, encode the text into dense vector representations of words that capture semantic relationships. These embeddings can capture word meanings and contextual relationships, which can be beneficial for spam classification.

   This means that word embeddings capture semantic meaning like, happiness and joy have the same meaning.

Categorizing as Spam:

After the pre – processing of data and feature extraction is completed, we can finally move onto the actual categorization of the data. Since our data has no labels, we must use unsupervised learning instead of supervised learning methods such as classification.

There are two methods which can be applied here: clustering and anomaly detection. Each method has its own pros and cons which play a major role in deciding which method to choose.

Clustering is the process of grouping similar data points together based on features. In spam detection, you could cluster emails into groups and label some clusters as spam based on their characteristics. Clustering can be effective when spam emails share certain common features that differentiate them from non-spam emails.

KMeans clustering is generally used in spam filtration. The syntax of KMeans clustering is as follows:

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters = n, init = 'k-means++', random_state = 42)

y_kmeans = kmeans.fit_predict(X)
```

The disadvantage of clustering is that it only groups together data with similar characteristics. If there is an isolated anomaly, clustering cannot detect it.

Anomaly detection separates out points that deviate from a majority of the data. Hence, if there are spam messages largely differ from the non – spam messages, anomaly detection is excellent at finding them. However, if spam messages are structured similarly to non – spam mails, anomaly detection cannot separate them.

A combined approach is often the most helpful in classifying spam messages. Clustering can be used to find out common spam patterns and anomaly detection can be used to find out the rare outliers.

ANALYSIS OF MODEL:

The analysis of a model is almost as important as the implementation of the model. If we do not know the accuracy and precision of a particular model, we cannot depend on it to solve our task.

One method of analyzing a spam detection model is manually inspecting a random batch of data, however this may be long and tedious, and may also introduce human biases.

We can however make use of a classification model to analyze our model. SVMs (support vector machines) and Naïve Bayes Theorem are often used to this effect. By training the classification model on the data and the models given by our model, we can find out the accuracy of the model on a test set. We can hence find out various metrics such as precision, recall, F1 score, etc.

DRAWBACKS OF MODEL:

1. Limited Context: Unsupervised learning will struggle to capture the contextual differences between spam and legitimate messages. This may lead to errors in outcome due to the presence of certain word, even if the context is legitimate

2. Evolving Spam Patterns: Spammers keep evolving their tactics in order to get past detection models. Hence, if the database the model is built on is outdated, our model will not perform accurately and will fail to recognize new patterns in spam messages.

3. Overfitting Anomalies: Anomaly detection methods might overfit to certain patterns, considering them as anomalies, even if they are not necessarily spam.

4. Inadequate Clustering: Clusters might not align well with spam vs. non-spam categories, leading to ambiguity in outcomes.

While this method can provide initial insights into categorizing spam, it has limitations, particularly due to the lack of labelled data and potential challenges in accurately capturing evolving spam patterns. A supervised or semi-supervised learning method is much better as the presence of preexisting labels makes it much easier for the model.

Resources Used:

1. https://www.globaltechcouncil.org/clustering/k-means-clustering-vs-hierarchical-clustering/#:~:text=The%20two%20main%20types%20of,an%20unknown%20number%20of%20classes.
2. https://pdfs.semanticscholar.org/0856/89349fe236829381c09a43c29278dad74146.pdf?_gl=1*1vvj53w*_ga*MzgxNDMyNDI3LjE2OTM0OTQ4OTE.*_ga_H7P4ZT52H5*MTY5MzQ5NDg5MC4xLjAuMTY5MzQ5NDkwOC40Mi4wLjA.
3. https://ieeexplore.ieee.org/document/8703222
4. https://www.analyticsvidhya.com/blog/2022/05/a-complete-guide-on-feature-extraction-techniques/#:~:text=If%20we%20have%20textual%20data,Feature%20Extraction%20from%20the%20text.