Synapse Task 3.3

OVERVIEW

If the dataset was labelled , this would be a Text Classification NLP problem.

Since the data is unlabelled , we need to use clustering to derive correlations between blocks of texts and place them in their appropriate group.

Step 1 – Preprocessing :

- Convert everything to lowercase
- Remove all punctuations (commas , fullstops , etc.)
- Tokenization (Split everything into separate words)
- Remove the stopwords (words which are only used to make the sentence grammatically correct , eg : the , a , on , am , etc.)
- Carry out lemmatization (reduce words to their root form , eg : typing reduces to type, eating reduces to eat, etc.)

Step 2 – Feature Extraction :

- We will be using Bag of Words.
- Let's say total vocabulary is of size n.
- So every block of text is represented by a vector of size n , and all it's entries will either be 1's or 0's.
- The vector will contain 1 if the corresponding word in the vocabulary is present in the block of text. Eg: If the vocab has "car" at index 58 , can the block of text contains the word "car" , index 58 of the vector representing this text will have a 1.
- Same logic applies for 0's.

- Note : Perhaps TF-IDF can also be used. Consider a block of spam email text , and we want to do feature extraction for the word “free”. Now TF should be high for all such spam emails , since spam emails will contain a lot of words like “free”. IDF should be roughly constant for all spam texts , since vocab size is constant and total number of texts having “free” will be roughly the same as the total number of spam texts. This should be the case for all “spammy” words. So TF*IDF will vary mostly with TF , and it should result in most spam texts having very similar coordinates.

- Note : Word2Vec can also be used. In this case , every word will be a vector and every block of text will be the average of all these vectors. This could be the best method to use.

Step 3 – Clustering :

- Therefore every block of text is now represented by a point in an n-dimensional coordinate system.

- Blocks which have very similar patterns will most likely all be spam , since all spam has the same objective , so they will use the same words and phrases.

- The rest of the blocks should be legitimate , since actual emails can be about a variety of topics so they wouldn’t have very strong relations.

- Hence , the spam text which have similar words and phrases will also have roughly same coordinates.

- So by using clustering algorithms , KMeans in this case , the model can figure out the clusters of similar texts.

- Group the similar blocks together , and the rest of the blocks together.

- A problem that might arise is improper clustering of legitimate emails , since they won’t have strong correlations. Since we will give k=2 in our KMeans Algorithm , one cluster will be the close points , i.e , the spam and everything else will the other cluster , no matter how far apart they will be.

Problems :

1) Legitimate emails can be considered spam if they contain words common to other spam.

2) Non-ASCII characters like emojis might be beneficial for clustering spam together , but their existence in the corpus might create a problem.

3) Number of dimensions are extremely high. Need something to reduce the number of dimensions.