Spam Email Classifier with KNN using TF-IDF scores

1. Assignment must be implemented in Python 3 only.
2. You are allowed to use libraries for data preprocessing (numpy, pandas, nltk etc) and for evaluation metrics, data visualization (matplotlib etc.).
3. You will be evaluated not just on the overall performance of the model and also on the experimentation with hyper parameters, data prepossessing techniques etc.
4. The report file must be a well documented jupyter notebook, explaining the experiments you have performed, evaluation metrics and corresponding code. The code must run and be able to reproduce the accuracies, figures/graphs etc.
5. For all the questions, you must create a train-validation data split and test the hyperparameter tuning on the validation set. Your jupyter notebook must reflect the same.
6. Strict plagiarism checking will be done. An F will be awarded for plagiarism.

Task: Given an email, classify it as spam or ham

Given input text file ("emails.txt") containing 5572 email messages, with each row having its corresponding label (spam/ham) attached to it.

This task also requires basic pre-processing of text (like removing stopwords, stemming/lemmatizing, replacing email_address with 'email-tag', etc..).

You are required to find the tf-idf scores for the given data and use them to perform KNN using Cosine Similarity.

Import necessary libraries

In [ ]:

Load dataset

In [ ]:

Preprocess data

In [ ]:

Split data

In [ ]:

Train your KNN model (reuse previously iplemented model built from scratch) and test on your data

1. Experiment with different distance measures [Euclidean distance, Manhattan distance, Hamming Distance] and compare with the Cosine Similarity distance results.

In [ ]:

2. Explain which distance measure works best and why? Explore the distance measures and weigh their pro and cons in different application settings.

3. Report Mean Squared Error(MSE), Mean-Absolute-Error(MAE), R-squared (R2) score in a tabular form

In [ ]:

4. Choose different K values (k=1,3,5,7,11,17,23,28) and experiment. Plot a graph showing R2 score vs k.

In [ ]:

Train and test Sklearn's KNN classifier model on your data (use metric which gave best results on your experimentation with built-from-scratch model.)

In [ ]:

Compare both the models result.

In [ ]:

What is the time complexity of training using KNN classifier?

What is the time complexity while testing? Is KNN a linear classifier or can it learn any boundary?