Implementation Of k-NN Classifier

---

1. **Implementation of an *unweighted* k-NN classifier**

- ✓ 'news_articles.mtx' file is loaded in dictionary
- ✓ Data is divided into two parts training vector or test vector
- ✓ 'news_articles.mtx'' file is loaded in labelled data dictionary
- ✓ Cosine similiarity method is written which except two documents in vector(dictionary) format and returns cosine similarity

  Numerator =sum (frequency of common words in both doc)
  Train = sqrt (sum (frequency of training doc words)
  Test = sqrt (sum (frequency of test doc words)
  Denominator = Train * Test
  Cosine Similarity = Numerator/Denominator
- ✓ With the help of cosine value K similar neighbours is returned with document number and similarity
- ✓ With the help of labelled data, label is given to sorted first k document which having high similarity
- ✓ Voting is done based on count, the class which has highest count in selected k, predicted as a class or label.
- ✓ Based on vote the accuracy is calculated

  Accuracy= (correct prediction/(total test data)) * 100

Random shuffle or random selection of training data is not considered first
First 67 % selected as training and 33 % selected as test data, Accuracy measures.
Split = 0.67 trainset = 67 % testset =33 %

KNN	Weighted KNN
for K:1 KNN Accuracy: 31.301 for K:2 KNN Accuracy: 32.454 for K:3 KNN Accuracy: 32.125 for K:4 KNN Accuracy: 32.454 for K:5 KNN Accuracy: 32.289	for K:1 Weighted KNN Accuracy 31.301 for K:2 Weighted KNN Accuracy: 31.301 for K:3 Weighted KNN Accuracy: 31.960 for K:4 Weighted KNN Accuracy: 32.125 for K:5 Weighted KNN Accuracy: 32.289

Data size
split = 0.80 trainset = 80 % testset=20 %

KNN	Weighted KNN
for K:1 KNN Accuracy: 62.5 for K:2 KNN Accuracy: 36.68 for K:3 KNN Accuracy: 54.07 for K:4 KNN Accuracy: 45.10 for K:5 KNN Accuracy: 53.53	for K:1 Weighted KNN Accuracy: 62.5 for K:2 Weighted KNN Accuracy: 62.5 for K:3 Weighted KNN Accuracy: 57.06 for K:4 Weighted KNN Accuracy: 57.88 for K:5 Weighted KNN Accuracy: 53.804

implementation validation

2. **Weighted kNN classifier using an appropriate weighting scheme.**

- ✓ For weight calculation, I referred the Wikipedia document for distance measure.

  Distance = 1-Similarity

  Weight = 1/ Distance

- ✓ To avoid the case, distance becomes 0 for similarity 1.
  for these case weight will be infinity, so normalize the data by adding 1 in distance
  Weight = 1/(1+Distance)
- ✓ Voting is done based on the weight class which has highest vote will be selected for predicted output

3. Evaluation

Training set and test set are random generated based on random values generated between (1, 1839), if random generated value is less than (1839*split), it is selected as training set else test set.

Split = 0.67 trainset = 67 % testset =33 %

KNN	Weighted KNN
for K:1 KNN Accuracy: 93.94	for K:1 Weighted KNN Accuracy: 94.22
for K:2 KNN Accuracy: 94.37	for K:2 Weighted KNN Accuracy: 95.39
for K:3 KNN Accuracy: 94.09	for K:3 Weighted KNN Accuracy: 96.33
for K:4 KNN Accuracy: 94.17	for K:4 Weighted KNN Accuracy: 94.70
for K:5 KNN Accuracy: 95.269	for K:5 Weighted KNN Accuracy: 97.73
for K:6 KNN Accuracy: 94.037	for K:6 Weighted KNN Accuracy: 97.176
for K:7 KNN Accuracy: 94.599	for K:7 Weighted KNN Accuracy: 94.086
for K:8 KNN Accuracy: 94.32	for K:8 Weighted KNN Accuracy: 94.35
for K:9 KNN Accuracy: 95.52	for K:9 Weighted KNN Accuracy: 95.214
for K:10 KNN Accuracy: 94.15	for K:10 Weighted KNN Accuracy: 95.548

Checked for other splits

split = 0.80 trainset = 80 % testset=20 %

KNN	Weighted KNN
for K:1 KNN Accuracy: 94.38	for K:1 Weighted KNN Accuracy: 95.135
for K:2 KNN Accuracy: 93.35	for K:2 Weighted KNN Accuracy: 93.49

4. Write a report explaining your implementation and detailing the results of your evaluation from Task 3.

Normal split
Case1. First 67 % for training and 33 % for testing:

1. Un-weighted KNN and weighted KNN both are giving almost similar result, there is no big difference in accuracy, and some time unweighted KNN accuracy is higher, so weighting skim for this split not suits well.

Case2. First 80 % for training and 20 % for testing:

1. In this skim, weighted KNN is always higher than un-weighted KNN,
2. for even K values weighted KNN skim perform well and this is also true,
   if, predicted output contains equal number of class, so it is difficult to vote which is better in this case voting is done based on weight. I found significant difference in accuracy. Weighted KNN accuracy is 10 to 20 % higher in some cases.

Random Split (training and test data size may vary based on random numbers.)
Case1. random 67 % for training and 33 % for testing

1. for random data, overall accuracy is increased for weighted KNN and unweighted KNN.
2. Interesting facts are weighted KNN accuracy is higher or equal to un-weighted KNN but not less than to unweighted KNN.
3. For k even values weighted KNN giving better results compare to unweighted KNN
4. Accuracy is increased for k=1 to K=7 then it is decrease and tends to stability and for k=9 and k=10, I found stable results.
5. .

Case2. random 80 % for training and 20 % for testing

This case is considered to find the impact on accuracy for higher split

1. Results are very impressive there is no big impact on accuracy, accuracy does not have big changes in percentage by changing k.
2. No big impact of this split size.

Conclusion: To get higher accuracy, randomized the data and find best split and k values, based on weighted voting. it is impossible to say which split is best for data, it is done only with observation, best split is considered which has small training set and gives better accuracy, and k will not have big impact on accuracy. Results derived

1. Best split is 67% using random selection method.
2. Best K = 5 for weighted KNN and unweighted KNN using random selection.
   but it is not fixed, it is based on training data and test data.
3. Unweighted KNN does not have big impact of k for random data almost constant accuracy.
4. Random selection of data is done to remove the bias from data, initially data splits in two parts first part 80 % as training data and remaining as test data, for this data set accuracy is fixed and at k=1, I found the best accuracy.