Homework 02 — Part II — amc1354 & ads798

1. **Exercise 1**

(a) i. For k=1,
Predicted label = 1
for "it leaves little doubt that kidman has become one of our best actors ."

ii. Confusion matrix on the test set for $k = 1$:

$$\begin{bmatrix} Predicted & 0 & 1 & All \\ True & & & \\ 0 & 75 & 152 & 227 \\ 1 & 56 & 217 & 273 \\ All & 131 & 369 & 500 \end{bmatrix}$$

TN=75
FP=152
FN=56
TP=217

iii. For k=1,
$Accuracy = 0.584$.
$TPR = 0.5880758807588076$,
$FPR = 0.6696035242290749$.

iv. For k=5,
Predicted label = 1
for "it leaves little doubt that kidman has become one of our best actors ."

v. Confusion matrix on the test set for $k = 5$:

$$\begin{bmatrix} Predicted & 0 & 1 & All \\ True & & & \\ 0 & 81 & 146 & 227 \\ 1 & 61 & 212 & 273 \\ All & 142 & 358 & 500 \end{bmatrix}$$

TN=81
FP=146
FN=61
TP=212

vi. For k=5,
$Accuracy = 0.586$.
$TPR = 0.5921787709497207$,
$FPR = 0.6431718061674009$.

vii. Accuracy for k=5 was already reported in (a)vi.

viii. Accuracy using zero-r = 54.6

(b) Because of the likelihood of having similar words increases substantially for the longest sentences, the shortest documents will find it difficult to make it in the

top k-ranked NN, even if they are actually examples that are closer to the test data. For example, let's look at these sentences:

1. "Julia Roberts coupled with Hugh Grant have such a chemistry that makes Notting Hill a great movie!" - label 1.

2. "Hugh Grant making the goofy Britishman, and Julia Robert as an American Holliwood star have such a chemistry that makes this movie very funny. However, Notting Hill isn't great overall. I prefer Roberts in parts when she acts parts like My best friend's wedding, bouncing between a romantinc and fragile person falling in love with her best friend and the tough, ectic city girl trying to hide her sweet spot!" - label 0.

3. "Anna Scott (J. Roberts) and William Thacker (H. Grant) have a perfect chemistry. Notting Hill will always be my favourite." - label 1.

1. and 3. have much more semantic similarity but less less words matching. So in a 1-NN, for instance, 1. will have a high number of common words with 2., and will be erroneously predicted as label 0.

(c) i. The CV accuracy for k=3 is: 0.5726666666666667

The CV accuracy for k=7 is: 0.5533333333333333

The CV accuracy for k=99 is: 0.544

ii. Confusion matrix on the test set for $k = 3$:

$$\begin{bmatrix} Predicted & 0 & 1 & All \\ True & & & \\ 0 & 81 & 146 & 227 \\ 1 & 59 & 214 & 273 \\ All & 140 & 360 & 500 \end{bmatrix}$$

TN=81
FP=146
FN=59
TP=214

For k=3,
$Accuracy = 0.586$.

(d) New distance function. We chose to adopt the cosine similarity on the bag of words matrix with TF-IDF weights.

i. First, we pre-processed data to remove stopwords, punctuation and we stemmed the words. Stemming is used to reducing inflected (or sometimes derived)

words to their word root form. E.g., "loving"'s stem is "lov" so that we can find a match between "loving this movie" and "I love that film". Then we calculated for each words, their term frequency multiplied by the inverse document frequency (TF-IDF = $TF \cdot IDF$).

TF is the bare count of hoe many times a word appears in a document.

IDF is a measure of how much information the word provides, i.e., if it's common or rare across a collection of documents, in this case the collection of the reviews in the training set. It is the logarithmically scaled inverse fraction of the documents that contain the word.

$$IDF(t, D) = log \frac{N}{|\{d \in D : t \in d\}|}$$

where

N: total number of documents in the corpus,
$|\{d \in D : t \in d\}|$: number of documents where the term t appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Each sentence becomes than a vector representing all the words present in the TRAINING corpus (bag-of-word model), and will have 0's for the words not used by that particular sentence and TF-IDF weights for each words which is used by the review. So, for each test sentence, we transform it to one such vector and perform the dot-product with each vector-sentence in the training corpus (a "corpus" is defined as a collection of documents). The result is the so-called cosine similarity between two vectors. We rank in decreasing order (high dot-product means high similarity) the similarities between the test example and all the training examples, and we pick the first k ranked positions. From here, we then follow the same logic as the distance function introduced in (a).

For example, for a training corpus composed by ["hi how are you", "are you ok", "i am not bad"], and test example "how are you doing", each vector representing a sentence will look at the presence of these words: "hi", "how", "are", "you", "ok", "i", "am" "not" "bad".
Each vector's element will have the IF-IDF weight if the word appears in the review or 0 if it doesn't.
The cosine similarity of the vector representing the test example, with each word in the corpus will be: [0.819215, 0.597969, 0], and we can see that the highest similarity is with the first sentence in the corpus, i.e. "how are you doing" is mostly similar to "hi how are you".

ii. First, stemming words helps computing similarities between words used in

different tenses, plural/singular names, etc. Removing punctuation would clear from similarities of sentence that don't have anything in common apart for punctuation. The same is valid for stop-words. Regarding TF-IDF, we expect to have better prediction by assigning a weight that emphasizes the importance of a term in a given context, in this case movie reviews and the training corpus. Finally, regarding the cosine measure, it is a good metric because it is large when the vectors point in the same direction, i.e. when there is a convergence of TF-IDF values, hence information.

iii. Confusion matrix on the test set for $k = 1$:

$$\begin{bmatrix} Predicted & 0 & 1 & All \\ True & & & \\ 0 & 139 & 88 & 227 \\ 1 & 60 & 213 & 273 \\ All & 199 & 301 & 500 \end{bmatrix}$$

TN=139
FP=88
FN=60
TP=213

iv. For k=1,
$Accuracy = 0.704$.
$TPR = 0.707641196013289$,
$FPR = 0.3876651982378855$.

v. Confusion matrix on the test set for $k = 5$:

$$\begin{bmatrix} Predicted & 0 & 1 & All \\ True & & & \\ 0 & 141 & 86 & 227 \\ 1 & 36 & 237 & 273 \\ All & 177 & 323 & 500 \end{bmatrix}$$

TN=141
FP=86
FN=36
TP=323

vi. For k=5,
$Accuracy = 0.756$.
$TPR = 0.7337461300309598$,
$FPR = 0.3788546255506608$.

vii. Yes, it did and much higher than the original distance metrics in both cases. Given the combination of factors: pre-processing, TF-IDF and cosine similarity, we expected a significant improvement.