Linear Algebra Assignment 4

Anastasia Riana 邱慧莉 109006234

Q1:

Selected spam emails:

5728	Subject: Your secret weapon for winning content It doesn't matter if you're a business owner, or a seasoned professional, you want a tool worth your ir	0	0
5729	Subject: (no subject) safe meet here	1	1
5730	Subject: Hello Darling! I can come to you! Hello. I am in your city. I know you a little. You and I have one mutual friend. If you want to get to know me b	2	2
5731	Subject: (no subject) safe meet here	3	3
5732	Subject: Illegal interview questions an employer cannot ask Feeling stuck in your career? You know you deserve a better job but starting to look for one	4	4

Q2:

(a) TF-IDF is an abbreviation of a text analysis method called *term frequency-inverse document frequency.* TF-IDF is a method in information retrieval that is intended to reflect how important a word is to a document or a corpus. TF-IDF is often used as the weighting factor in information retrieval. Using this method, we can also eliminate stop words.

Suppose we have an English text and we want to rank it according to these words, 'quick', 'brown', 'fox'. We can start out by eliminating the data that don't contain all three words. We can further distinguish them by counting the amount of times how many times those words have appeared called *term frequency*.

In other cases, the word 'the' or 'and' is so common, if we use only *term frequency,* it might reflect an incorrect result. Hence, an *inverse document frequency* factor is put into consideration which might diminish the weight of terms that occur very frequently in the document set and increase the weight for words that rarely appear.

(b) Stop words are often words like 'and', 'the', etc. This group of words are sometimes presumed to be unimportant and uninformative. Hence, using the TF-IDF method, we can eliminate the stop words in order to get a more accurate result of the dataset that we are trying to process. When using the Python package, we have to make sure that the stop word list was processed the same way as the one we have used in the vectorizer, where it would try to identify and warn us if there are any kinds of inconsistencies.

Q3:

- **Fold = 0 as training data ; Fold = 4 as test data**

	Is a spam	Is a ham
Predicted as a spam	213 (TP)	2 (FP)
Predicted as a ham	68 (FN)	860 (TN)

```
213 68
2 860
Prediction = 93.87576552930884
Accuracy = 99.06976744186046
Recall = 75.80071174377224
```

- **Fold = 1 as training data ; Fold = 4 as test data**

	Is a spam	Is a ham
Predicted as a spam	229 (TP)	2 (FP)
Predicted as a ham	52 (FN)	860 (TN)

```
229 52
2 860
Prediction = 95.2755905511811
Accuracy = 99.13419913419914
Recall = 81.49466192170819
```

- **Fold = 2 as training data ; Fold = 4 as test data**

	Is a spam	Is a ham
Predicted as a spam	210 (TP)	4 (FP)
Predicted as a ham	71 (FN)	858 (TN)

```
210 71
4 858
Prediction = 93.43832020997375
Accuracy = 98.13084112149532
Recall = 74.73309608540926
```

Q4:

First, we are going to use *np.linalg.svd* to our spam and ham. From it, we will get our S matrix. We will then use the column space of S to find the distance between the test data to

the column space S (or H). Then, we calculate the distance between the original test data and its projection, using *np.linalg.norm*. Here is the comparison of least-square method and SVD with fold from 0-3 as training data and fold 4 as the test data.

- SVD

```
204 77
1 861
Prediction = 93.1758530183727
Accuracy = 99.51219512195122
Recall = 72.59786476868328
Duration: 0:00:47.666213
```

- Least-Square

```
208 73
1 861
Prediction = 93.52580927384076
Accuracy = 99.52153110047847
Recall = 74.02135231316726
Duration: 0:02:05.055214
```

As can be seen from above, the result of SVD and least square method are just as accurate. However, we can observe that SVD is significantly faster than the least-square method. This can occur due to SVD's ability to reduce the size of the data, making it much easier for the program to process the data. Not only that, the way that SVD works makes computing the matrix very efficiently because in least square method, they will be solving a system of linear equations which is much more time-consuming.

Q5:

(a) **Result (when rank(S)=200 and rank(H)=600):**

```
254 27
1 861
Prediction = 97.55030621172354
Accuracy = 99.6078431372549
Recall = 90.39145907473309
Duration: 0:00:47.958278
```

(b) Latent Semantic Analysis (LSA) is a technique in NLP of analyzing relationships between a set of documents and the terms it contains by producing a set of concepts that are related to the documents and terms. In LSA, it will use SVD concepts to break

down the term-document matrix A into a term-concept U, a singular value matrix S and a concept-document matrix V in the form of A = USV'.

Using LSA, it can improve accuracy as the method allows us to handle sparse data, handle synonyms, reducing the impact of words such as 'the', 'and', etc. Not only that but it can also do dimensionality reduction, without losing any context. As a result, it will reduce the computation power and the time taken for each computation. LSA can help to improve the accuracy

(c) Using cross-validation, we can compare the performance of the program using different rank values. After running the program, we can compute the prediction, accuracy and recall values. If the results yield a higher accuracy value, then it can be considered the best rank to use as it gives the best performance.

Q6:

Using non-negative matrix factorization, a matrix X with mxn size can be factorized into two matrices W and H with mxk size and kxn size respectively. Matrix W and H will only have non-negative values. To compute this factorization, we need to minimize the distance between X and W*H. As it is a NP-hard problem, it cannot be computed analytically and instead approximation is used. Using the multiplicative update rules, we can approximate:

$$A \sim W, H$$

$$W \leftarrow W \,.\, W^T X / WHH^T \qquad\qquad H \leftarrow H \,.\, W^T X / W^T WH$$

In a spam filter, non-negative matrix factorization can be used to detect several topics of messages and residuals. Using that residuals, we can use it as a spam filter as the residuals can be generated into a document-term matrix. We can use non-negative matrix factorization to factorize the term frequency matrix as W and H. In this example, W can represent topics while H can represent the words as a combination of these topics. Then, we can use W and H matrices to classify whether it's spam or ham.