EE219 Project 4

304743326 Andrew Lin, 004587761 Wei-Ting Chen

Part 1

We first fetch the data from Computer Technology and Recreational Activity. Then convert the labels into 0 and 1, indicating the two classes respectively.

```
import numpy as np
from sklearn.datasets import fetch_20newsgroups
categories = [ 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware',
               'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']
train_data = fetch_20newsgroups(subset='all', categories=categories, shuffle=True, random_state=42)
X = np.asarray(train_data.data)
y = [0 if x<4 else 1 for x in train_data.target]
y = np.asarray(y)
```

As implemented in project 2, we tokenize all the words in the documents with stop word elimination and stemming, then we transform the tokens into a TFIDF matrix with a size of 7882 × 57042, the number of documents and words respectively.

```
import nltk
import string
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from nltk.stem import PorterStemmer

# stemming words from the same root
stemmer = PorterStemmer()
def tokenize_and_stem(text):
    tokens = nltk.tokenize.word_tokenize(text)
    tokens = [token.strip(string.punctuation)
                for token in tokens if token.isalnum()]
    tokens = [stemmer.stem(token) for token in tokens]
    return tokens

# vectorize documents
vectorizer = CountVectorizer(min_df=1, stop_words='english', tokenizer=tokenize_and_stem)
X_vect = vectorizer.fit_transform(X)
print(X_vect.shape)

# transform into TFIDF
tfidf_transformer = TfidfTransformer(sublinear_tf=True, use_idf=True)
X_tfidf = tfidf_transformer.fit_transform(X_vect)
```

Part 2

Apply K-means clustering directly to the TFIDF matrix with k=2 using a several permutations of rows. The confusion matrices and performance metrics of 5 different permutations of rows are shown below. The model is obviously unstable since it K-means clustering depends on the initial centers. It's unlikely to pick a rather good set of initial centers as only ⅕ chance of achieving a high score.

```
[[3903    0]
 [2250 1729]]
Homogeneity: 0.260456
Completeness: 0.343134
Adjusted Rand-Index: 0.184014
Adjusted Mutual Information: 0.260388
[[3903    0]
 [2265 1714]]
Homogeneity: 0.257699
Completeness: 0.341076
Adjusted Rand-Index: 0.180762
Adjusted Mutual Information: 0.257631
```

```
[[ 111 3792]
 [3781  198]]
Homogeneity: 0.763519
Completeness: 0.763553
Adjusted Rand-Index: 0.849315
Adjusted Mutual Information: 0.763498
[[   0 3903]
 [1729 2250]]
Homogeneity: 0.260456
Completeness: 0.343134
Adjusted Rand-Index: 0.184014
Adjusted Mutual Information: 0.260388
```

```
[[3903    0]
 [2252 1727]]
Homogeneity: 0.260088
Completeness: 0.342859
Adjusted Rand-Index: 0.183579
Adjusted Mutual Information: 0.260020
```

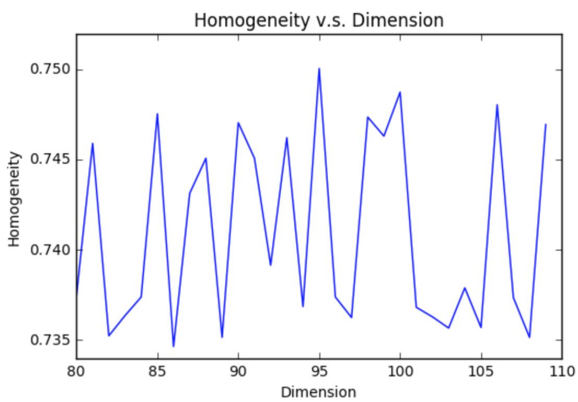
The metrics give a perfect clustering model scores of 1, so the higher the scores are, the more accurately the model performs. Applying K-means clustering directly on the TFIDF matrix actually results in a unstable clustering model. Except for the one that got 0.763519 on homogeneity score, all other 4l scores are below 0.5, which are lower than random guess.

The permutations of rows do affect the performance but only affects slightly. The difference comes from the initial centers. Note that an antidiagonal matrix doesn’t matter because we can make the predictions opposite and make it diagonal. An antidiagonal matrix does not affect the metrics either.

Part 3

Principle component analysis(PCA) is a commonly used technique for dimensionality reduction, but in this case our input is too sparse for the Python PCA module, so we use normalized **latent semantic indexing(LSI)**. Since the metrics above are consistent enough, we choose the dimension that gives highest homogeneity score. After several experiments, we know that the optimal dimension lies between 90~100, so we can reduce the range of finding the optimal dimension to 80~110 to save computational time.

```
95 dimensions give the max homogeneity score
```



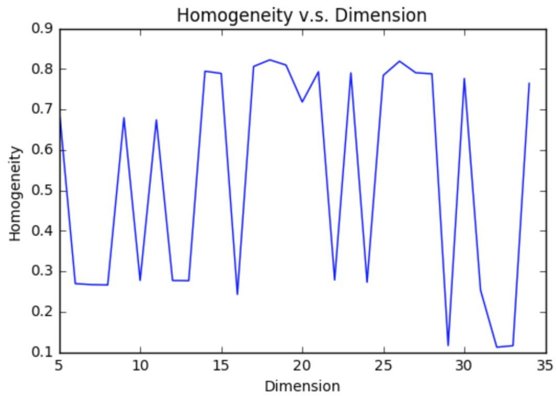

```
[[3759  144]
 [ 191 3788]]
Homogeneity: 0.746947
Completeness: 0.746900
Adjusted Rand-Index: 0.837197
Adjusted Mutual Information: 0.746876
```

The performance of our model has become more stable after applying dimensionality reduction. The confusion matrix also shows that the classification has become more accurate as the diagonal elements are very large. We try to get more improvement so we apply logarithm function to the LSI data, but it didn’t turn out better.

```
[[2991  912]
 [  87 3892]]
Homogeneity: 0.500144
Completeness: 0.518180
Adjusted Rand-Index: 0.557223
Adjusted Mutual Information: 0.500098
```

Another approach to do dimensionality reduction is nonnegative matrix factorization(NMF). We apply NMF to the normalized TFIDF matrix. Since NMF takes more computational time, we aim to find the dimension that gives the highest homogeneity score from dimension=1 to 100. After several experiments, we know that the optimal dimension lies between 5~35.

```
18 dimensions give the max homogeneity score
```

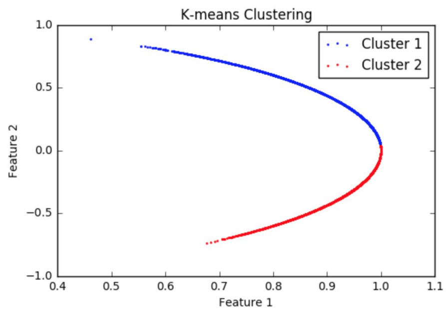



```
[[3786  117]
 [  94 3885]]
Homogeneity: 0.822237
Completeness: 0.822324
Adjusted Rand-Index: 0.895774
Adjusted Mutual Information: 0.822221
```

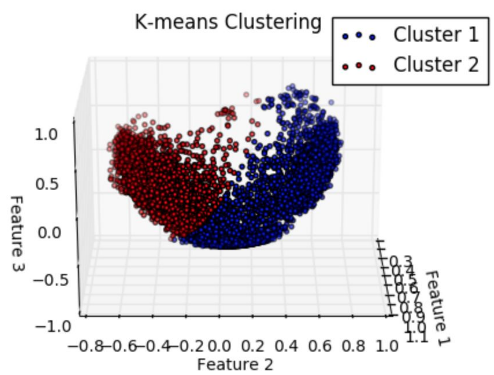
Using NMF actually gives out a better accuracy. Additionally, the confusion matrix also looks better. Different from LSI, higher dimension doesn't guarantee better results. And throughout the experiments, NMF yields a more stable and accurate reduced data.

Part 4

We can visualize the performance of the model by plotting the data points onto a 2D or 3D space. The plots of LSI model are shown below. Since LSI requires higher dimensional data to give better result, it does not perform well when reduced to 2 and 3.

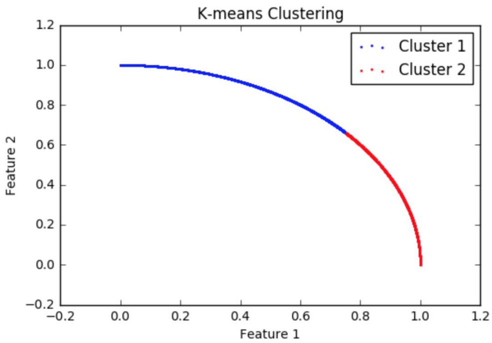



```
[[3813   90]
 [ 788 3191]]
Homogeneity: 0.538872
Completeness: 0.550016
Adjusted Rand-Index: 0.604012
Adjusted Mutual Information: 0.538829
```

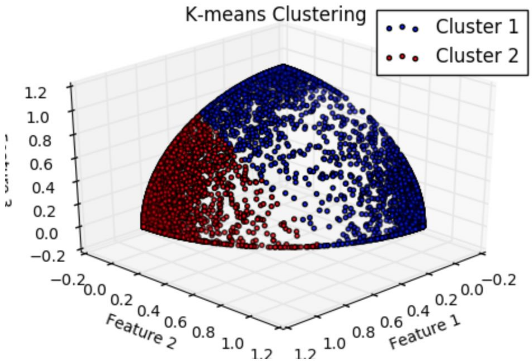



```
[[3677  226]
 [ 188 3791]]
Homogeneity: 0.703164
Completeness: 0.703306
Adjusted Rand-Index: 0.800911
Adjusted Mutual Information: 0.703137
```

For NMF data, the plots are shown below. Reducing dimension down to 2 and 3 is not a suffcient representation of the data. Thus the model does not do well.




```
[[3775  128]
 [ 659 3320]]
Homogeneity: 0.558698
Completeness: 0.565056
Adjusted Rand-Index: 0.640442
Adjusted Mutual Information: 0.558657
```



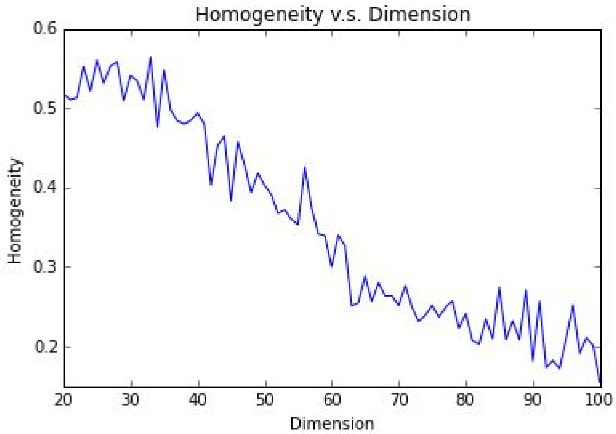

```
[[3524  379]
 [  99 3880]]
Homogeneity: 0.682861
Completeness: 0.686040
Adjusted Rand-Index: 0.772104
Adjusted Mutual Information: 0.682832
```

In general, utilizing nonlinear transformation functions can be useful in mapping the data into a different dimension, which is usually a higher one. Such transformation improves the separability of data since most original data are not directly separable. And after transformation it can be more descriptive such as in terms of decision boundaries. However in our project, just using NMF reduction can achieve 97% accuracy in clustering data into 2 classes.

Part 5

In this part, we want to apply clustering on all the 20 sub-class labels. We used the same transformation method as described above: tokenized the words in all the documents, and then transform the words into a TFIDF matrix. We tried reducing the dimension of the TFIDF matrix using both NMF and truncated SVD. We have chosen K=20 as the parameter k to use in K-means clustering. We again used homogeneity score, completeness score, adjusted rand score, and the adjusted mutual info score as our measures.

For NMF dimension reduction, we try to find the optimal dimension in the range of 20 to 100. The optimal dimension we found is 33. Results are shown below:




```
Homogeneity: 0.572720
Completeness: 0.589891
Adjusted Rand-Index: 0.399455
Adjusted Mutual Information: 0.571340
```

We have also tried applying non-linear transformation on the data vectors. Specifically, we applied logarithm and square functions, but it did not show any improvement in terms of the measure we used.

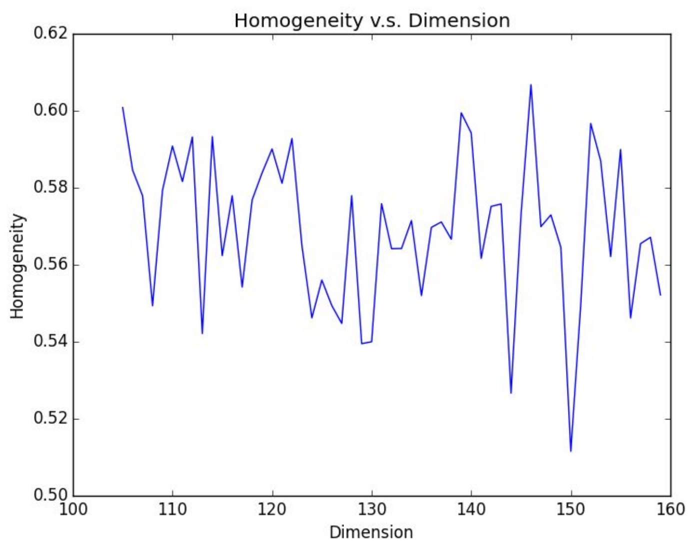
Logarithm transformation

```
Homogeneity: 0.133007
Completeness: 0.133053
Adjusted Rand-Index: 0.054418
Adjusted Mutual Information: 0.130210
```

Square transformation

```
Homogeneity: 0.491192
Completeness: 0.521238
Adjusted Rand-Index: 0.263015
Adjusted Mutual Information: 0.489549
```

For truncated SVD dimension reduction, we try to find the optimal dimension in the range of 105 to 160. The optimal dimension we found is 146. Results are shown below:



```
Homogeneity: 0.571052
Completeness: 0.592505
Adjusted Rand-Index: 0.377079
Adjusted Mutual Information: 0.569667
```

Again, we applied logarithm and square transformation with our optimal dimension, but shows no improvement.

Logarithm Transformation

```
Homogeneity: 0.469533
Completeness: 0.509188
Adjusted Rand-Index: 0.249504
Adjusted Mutual Information: 0.467817
```

Square

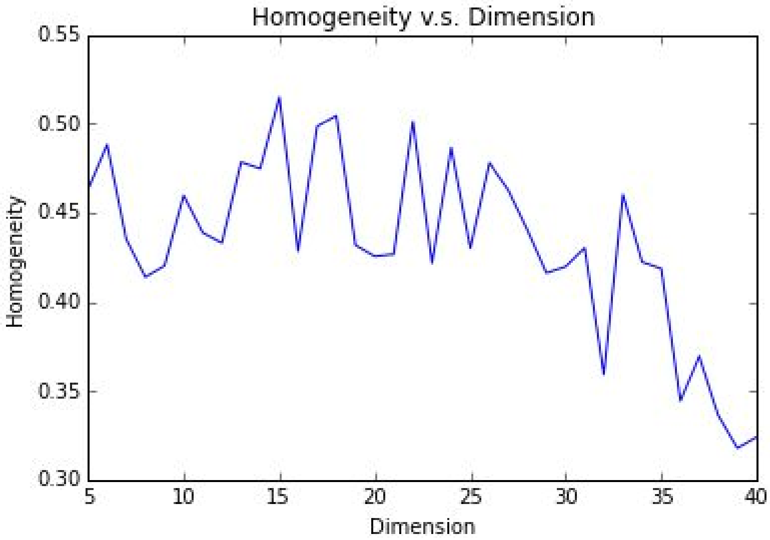
```
Homogeneity: 0.470968
Completeness: 0.530224
Adjusted Rand-Index: 0.251125
Adjusted Mutual Information: 0.469253
```

The best representation we found is applying NMF dimension reduction to reduce it into 33 dimension.

Part 6

In this part, we want to apply clustering on the 6 topic-wise classes. We again used the same transformation method to build a TFIDF matrix. We tried reducing the dimension of the TFIDF matrix using both NMF and truncated SVD. We have chosen K=6 as the parameter k to use in K-means clustering. Homogeneity score, completeness score, adjusted rand score, and the adjusted mutual info score are used as our measures.

For NMF dimension reduction, we try to find the optimal dimension in the range of 5 to 40 after a few experiments. The optimal dimension we found is 15. Results are shown below:




```
Homogeneity: 0.486641
Completeness: 0.493026
Adjusted Rand-Index: 0.346387
Adjusted Mutual Information: 0.486440
```

We applied logarithm and square transformation, but it did not show any improvement in terms of the measure we used. In fact, the purity scores drops drastically when we apply logarithm transformation.

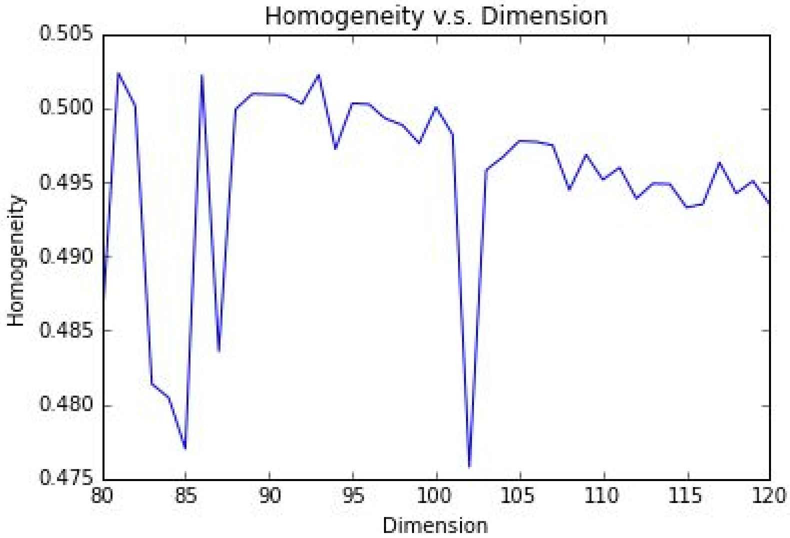
Logarithm Transformation

```
Homogeneity: 0.085640
Completeness: 0.081593
Adjusted Rand-Index: 0.064678
Adjusted Mutual Information: 0.081251
```

Square Transformation

```
Homogeneity: 0.410308
Completeness: 0.441799
Adjusted Rand-Index: 0.254731
Adjusted Mutual Information: 0.410077
```

For truncated SVD dimension reduction, we try to find the optimal dimension in the range of 80 to 120. The optimal dimension we found is 81. Results are shown below:




```
Homogeneity: 0.502393
Completeness: 0.517003
Adjusted Rand-Index: 0.346484
Adjusted Mutual Information: 0.502198
```

Logarithm Transformation

```
Homogeneity: 0.275315
Completeness: 0.312030
Adjusted Rand-Index: 0.132519
Adjusted Mutual Information: 0.275031
```

Square Transformation

```
Homogeneity: 0.313822
Completeness: 0.333878
Adjusted Rand-Index: 0.177972
Adjusted Mutual Information: 0.313553
```

The best representation we found is applying truncated SVD dimension reduction to reduce it into 81 dimension.