Serie 9: Vector Space Model

1. With the slides of the lecture and the code and data available in the folder Vector Space Model (in the large folder Exercises) you can manipulate the three different text genre of the French theatre.

For each genre, one can generate a profile in the form of a single vector. But each play (in a given genre) could be more or less close to this profile (or group average).

Which are the three plays for each text genre (or group) that are the closest to the profile?

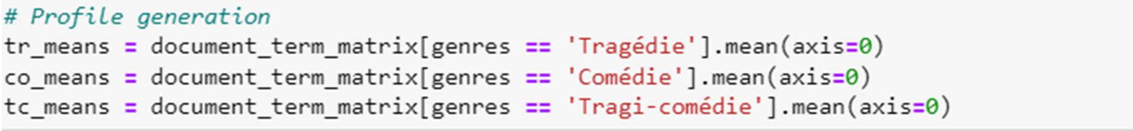
Solution:

In the given folder with all the plays, there are 3 genres

'Comédie', 'Tragédie', 'Tragi-comédie'.

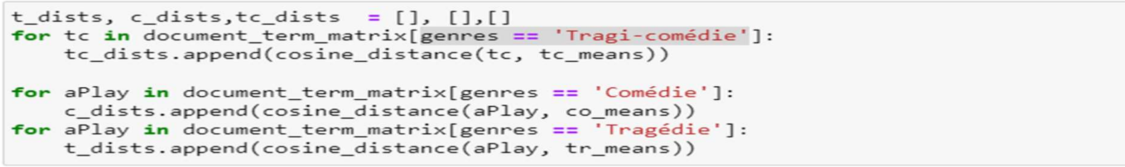
Steps followed to find the 3 plays in each genre:

- First, we extract the vocabulary from the corpus
    - The corpus is tokenized, by preprocessing the words by lowering the word, removing punctuations, and substituting the apostrophes.
- Create the document-term matrix, for the word frequency in each document
- Calculating a profile for each category, using the mean in the document_term_matrix.

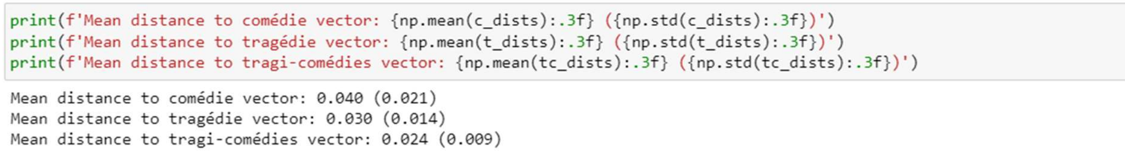


```
# Profile generation
tr_means = document_term_matrix[genres == 'Tragédie'].mean(axis=0)
co_means = document_term_matrix[genres == 'Comédie'].mean(axis=0)
tc_means = document_term_matrix[genres == 'Tragi-comédie'].mean(axis=0)
```

- Calculate the distance between the profile and each play in the given category:



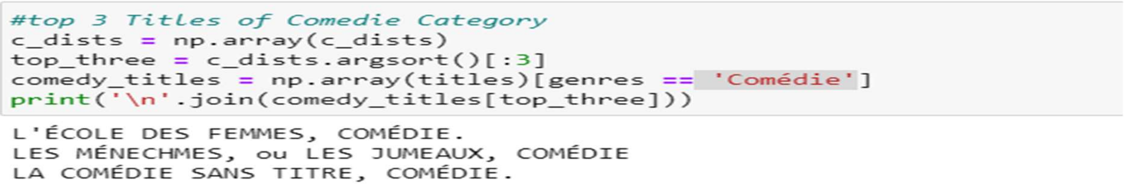
```
t_dists, c_dists,tc_dists  = [], [],[]
for tc in document_term_matrix[genres == 'Tragi-comédie']:
    tc_dists.append(cosine_distance(tc, tc_means))

for aPlay in document_term_matrix[genres == 'Comédie']:
    c_dists.append(cosine_distance(aPlay, co_means))
for aPlay in document_term_matrix[genres == 'Tragédie']:
    t_dists.append(cosine_distance(aPlay, tr_means))
```



```
print(f'Mean distance to comédie vector: {np.mean(c_dists):.3f} ({np.std(c_dists):.3f})')
print(f'Mean distance to tragédie vector: {np.mean(t_dists):.3f} ({np.std(t_dists):.3f})')
print(f'Mean distance to tragi-comédies vector: {np.mean(tc_dists):.3f} ({np.std(tc_dists):.3f})')
```

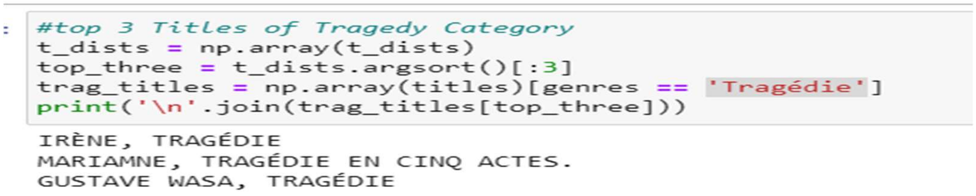
```
Mean distance to comédie vector: 0.040 (0.021)
Mean distance to tragédie vector: 0.030 (0.014)
Mean distance to tragi-comédies vector: 0.024 (0.009)
```

- From these arrays, we can sort the distances to find the plays closed to the category vector



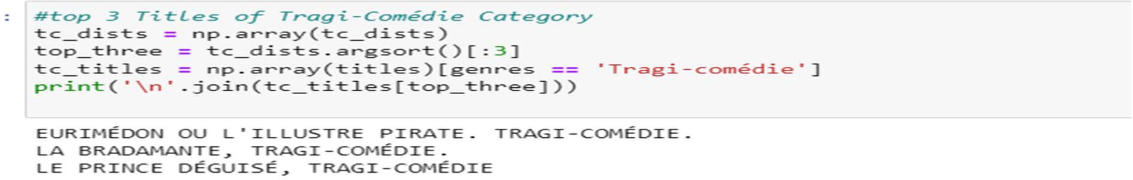
```
#top 3 Titles of Comedie Category
c_dists = np.array(c_dists)
top_three = c_dists.argsort()[:3]
comedy_titles = np.array(titles)[genres == 'Comédie']
print('\n'.join(comedy_titles[top_three]))
```

```
L'ÉCOLE DES FEMMES, COMÉDIE.
LES MÉNECHMES, ou LES JUMEAUX, COMÉDIE
LA COMÉDIE SANS TITRE, COMÉDIE.
```



```
#top 3 Titles of Tragedy Category
t_dists = np.array(t_dists)
top_three = t_dists.argsort()[:3]
trag_titles = np.array(titles)[genres == 'Tragédie']
print('\n'.join(trag_titles[top_three]))
```

```
IRÈNE, TRAGÉDIE
MARIAMNE, TRAGÉDIE EN CINQ ACTES.
GUSTAVE WASA, TRAGÉDIE
```



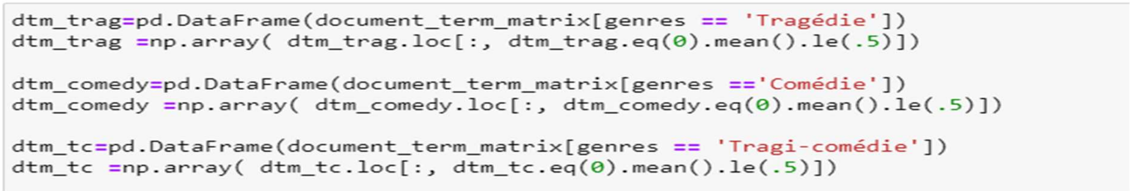
```
#top 3 Titles of Tragi-Comédie Category
tc_dists = np.array(tc_dists)
top_three = tc_dists.argsort()[:3]
tc_titles = np.array(titles)[genres == 'Tragi-comédie']
print('\n'.join(tc_titles[top_three]))
```

```
EURIMÉDON OU L'ILLUSTRE PIRATE. TRAGI-COMÉDIE.
LA BRADAMANTE, TRAGI-COMÉDIE.
LE PRINCE DÉGUISÉ, TRAGI-COMÉDIE
```

Question 2. Each profile was built by averaging over all term frequencies of plays belonging to that group. Do you see another way to generate a profile for a set of documents (or vectors)? Do you think that the profile must include all words appearing at least once in a play of the group? If no, how can we select a subset of the terms that must appear in a profile?

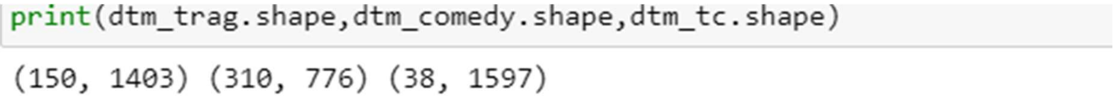
Solution:

Instead of considering the mean of all the terms that appear in the corpus of all the themes, we consider only the terms which appear in a particular category atleast in 50% of the plays.



```
dtm_trag=pd.DataFrame(document_term_matrix[genres == 'Tragédie'])
dtm_trag =np.array( dtm_trag.loc[:, dtm_trag.eq(0).mean().le(.5)])

dtm_comedy=pd.DataFrame(document_term_matrix[genres =='Comédie'])
dtm_comedy =np.array( dtm_comedy.loc[:, dtm_comedy.eq(0).mean().le(.5)])

dtm_tc=pd.DataFrame(document_term_matrix[genres == 'Tragi-comédie'])
dtm_tc =np.array( dtm_tc.loc[:, dtm_tc.eq(0).mean().le(.5)])
```

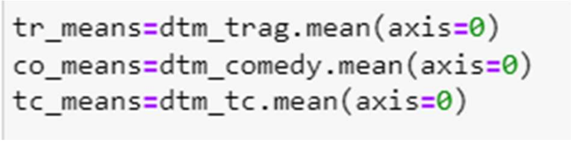
The size of the document term matrix is reduced as the terms considered are reduced:



```
print(dtm_trag.shape,dtm_comedy.shape,dtm_tc.shape)
```

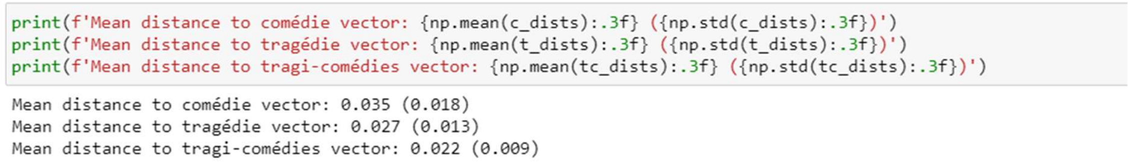
```
(150, 1403) (310, 776) (38, 1597)
```

Then we obtain the mean of the terms to be able to calculate the cosine distance between the plays



```
tr_means=dtm_trag.mean(axis=0)
co_means=dtm_comedy.mean(axis=0)
tc_means=dtm_tc.mean(axis=0)
```

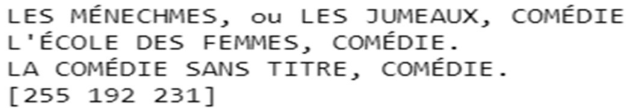
When we calculate the mean distance between the profile we obtain the following statistics:



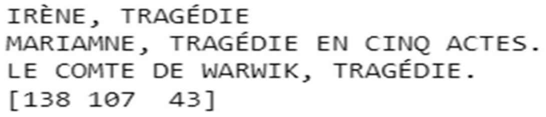
```
print(f'Mean distance to comédie vector: {np.mean(c_dists):.3f} ({np.std(c_dists):.3f})')
print(f'Mean distance to tragédie vector: {np.mean(t_dists):.3f} ({np.std(t_dists):.3f})')
print(f'Mean distance to tragi-comédies vector: {np.mean(tc_dists):.3f} ({np.std(tc_dists):.3f})')
```

```
Mean distance to comédie vector: 0.035 (0.018)
Mean distance to tragédie vector: 0.027 (0.013)
Mean distance to tragi-comédies vector: 0.022 (0.009)
```

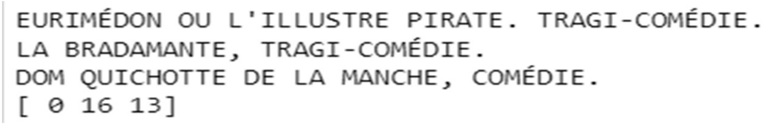
We can observe slight reduction in the deviation of the comedie plays from .21 to .18, using the reduced document term matrix.We can examine the top 3 plays for each category remains almost the same compared to the document term matrix, with all the terms considered.



```
LES MÉNECHMES, ou LES JUMEAUX, COMÉDIE
L'ÉCOLE DES FEMMES, COMÉDIE.
LA COMÉDIE SANS TITRE, COMÉDIE.
[255 192 231]
```



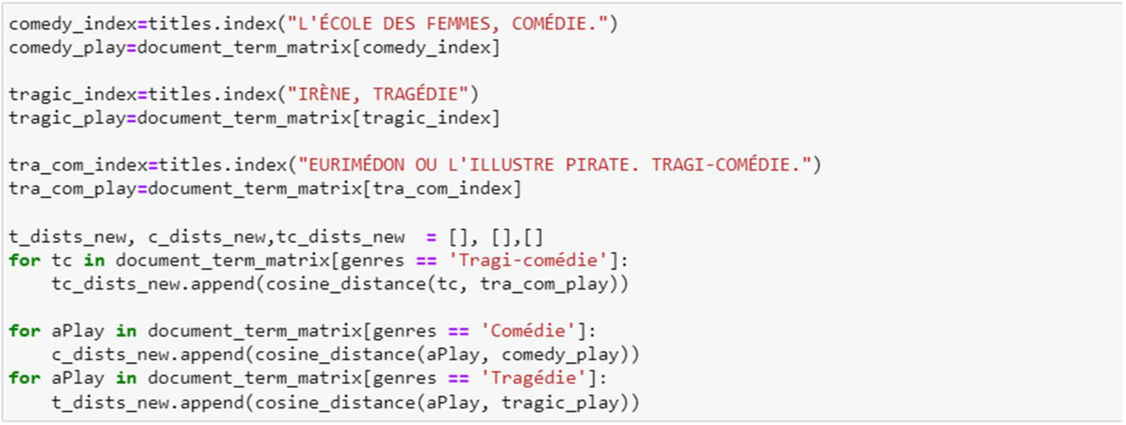
```
IRÈNE, TRAGÉDIE
MARIAMNE, TRAGÉDIE EN CINQ ACTES.
LE COMTE DE WARWIK, TRAGÉDIE.
[138 107  43]
```



```
EURIMÉDON OU L'ILLUSTRE PIRATE. TRAGI-COMÉDIE.
LA BRADAMANTE, TRAGI-COMÉDIE.
DOM QUICHOTTE DE LA MANCHE, COMÉDIE.
[ 0 16 13]
```

Alternate method considered:

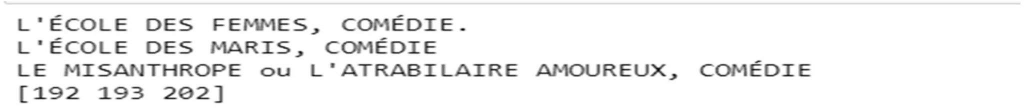
Using the play which is most typical to the category we can obtain the similar plays



```python
comedy_index=titles.index("L'ÉCOLE DES FEMMES, COMÉDIE.")
comedy_play=document_term_matrix[comedy_index]

tragic_index=titles.index("IRÈNE, TRAGÉDIE")
tragic_play=document_term_matrix[tragic_index]

tra_com_index=titles.index("EURIMÉDON OU L'ILLUSTRE PIRATE. TRAGI-COMÉDIE.")
tra_com_play=document_term_matrix[tra_com_index]

t_dists_new, c_dists_new,tc_dists_new  = [], [],[]
for tc in document_term_matrix[genres == 'Tragi-comédie']:
    tc_dists_new.append(cosine_distance(tc, tra_com_play))

for aPlay in document_term_matrix[genres == 'Comédie']:
    c_dists_new.append(cosine_distance(aPlay, comedy_play))
for aPlay in document_term_matrix[genres == 'Tragédie']:
    t_dists_new.append(cosine_distance(aPlay, tragic_play))
```

In this case, we can observe that the plays which we obtain are very similar, for example, in the comedy category

L'ÉCOLE DES FEMMES, COMÉDIE, produces L'ÉCOLE DES MARIS, COMÉDIE as the closest play, from the name we can observe that the topics and terms used must be very common.



```
L'ÉCOLE DES FEMMES, COMÉDIE.
L'ÉCOLE DES MARIS, COMÉDIE
LE MISANTHROPE ou L'ATRABILAIRE AMOUREUX, COMÉDIE
[192 193 202]
```

Conclusion:

Vector Space Model, helps to generate a profile for the documents and try to find similar documents based on the profile.