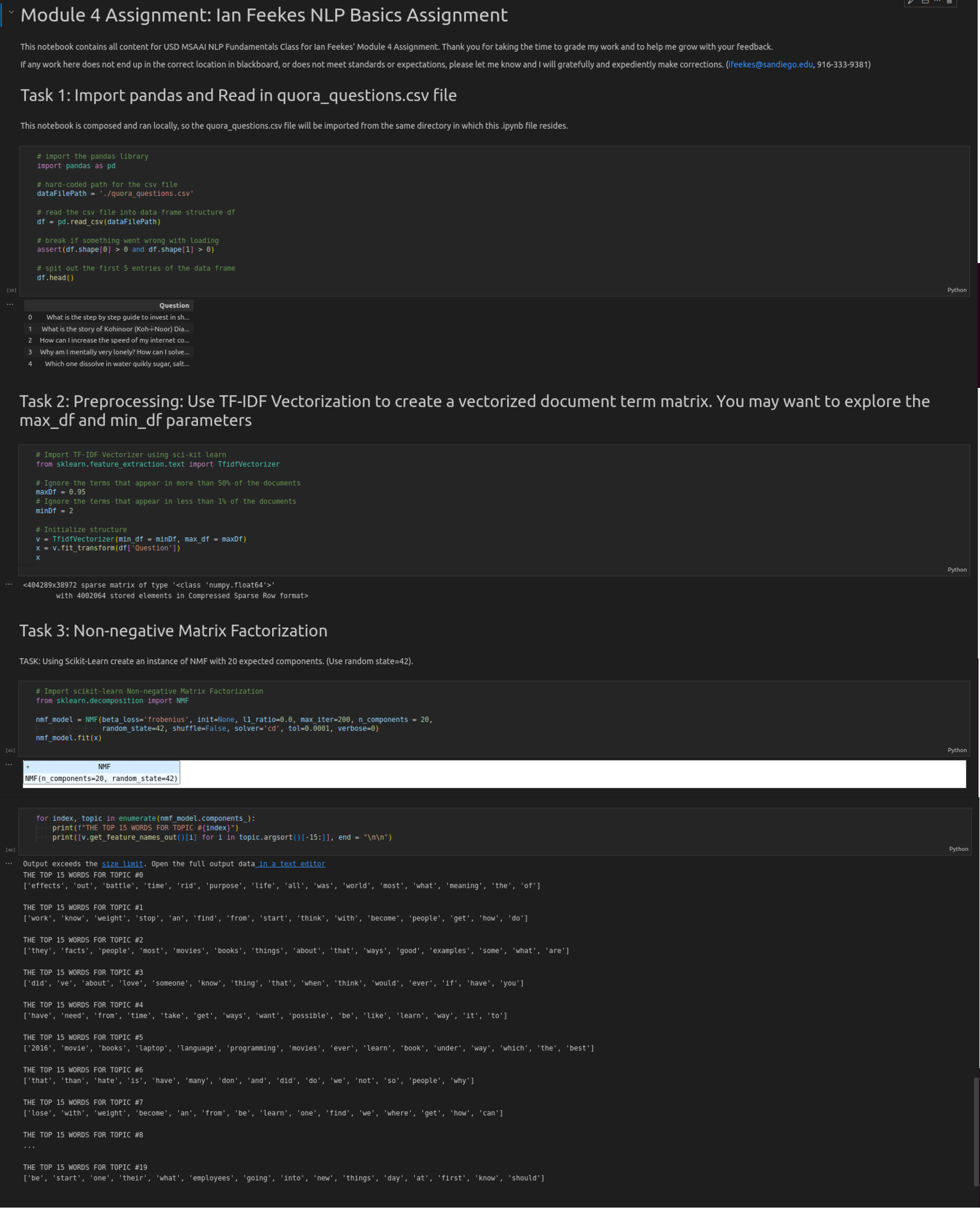


Module 4 Assignment: Ian Feekes NLP Basics Assignment

This notebook contains all content for USD MSAAI NLP Fundamentals Class for Ian Feekes' Module 4 Assignment. Thank you for taking the time to grade my work and to help me grow with your feedback.

If any work here does not end up in the correct location in blackboard, or does not meet standards or expectations, please let me know and I will gratefully and expediently make corrections. (ifeekes@sandiego.edu, 916-333-9381)

Task 1: Import pandas and Read in quora_questions.csv file

This notebook is composed and ran locally, so the quora_questions.csv file will be imported from the same directory in which this .ipynb file resides.

```python
# import the pandas library
import pandas as pd

# hard-coded path for the csv file
dataFilePath = './quora_questions.csv'

# read the csv file into data frame structure df
df = pd.read_csv(dataFilePath)

# break if something went wrong with loading
assert(df.shape[0] > 0 and df.shape[1] > 0)

# spit out the first 5 entries of the data frame
df.head()
```

[39]

	Question
0	What is the step by step guide to invest in sh...
1	What is the story of Kohinoor (Koh-i-Noor) Dia...
2	How can I increase the speed of my internet co...
3	Why am I mentally very lonely? How can I solve...
4	Which one dissolve in water quikly sugar, salt...

Task 2: Preprocessing: Use TF-IDF Vectorization to create a vectorized document term matrix. You may want to explore the max_df and min_df parameters

```python
# Import TF-IDF Vectorizer using sci-kit learn
from sklearn.feature_extraction.text import TfidfVectorizer

# Ignore the terms that appear in more than 50% of the documents
maxDf = 0.95
# Ignore the terms that appear in less than 1% of the documents
minDf = 2

# Initialize structure
v = TfidfVectorizer(min_df = minDf, max_df = maxDf)
x = v.fit_transform(df['Question'])
x
```

```
<404289x38972 sparse matrix of type '<class 'numpy.float64'>'
	with 4002064 stored elements in Compressed Sparse Row format>
```

Task 3: Non-negative Matrix Factorization

TASK: Using Scikit-Learn create an instance of NMF with 20 expected components. (Use random state=42).

```python
# Import scikit-learn Non-negative Matrix Factorization
from sklearn.decomposition import NMF

nmf_model = NMF(beta_loss='frobenius', init=None, l1_ratio=0.0, max_iter=200, n_components = 20,
                random_state=42, shuffle=False, solver='cd', tol=0.0001, verbose=0)
nmf_model.fit(x)
```

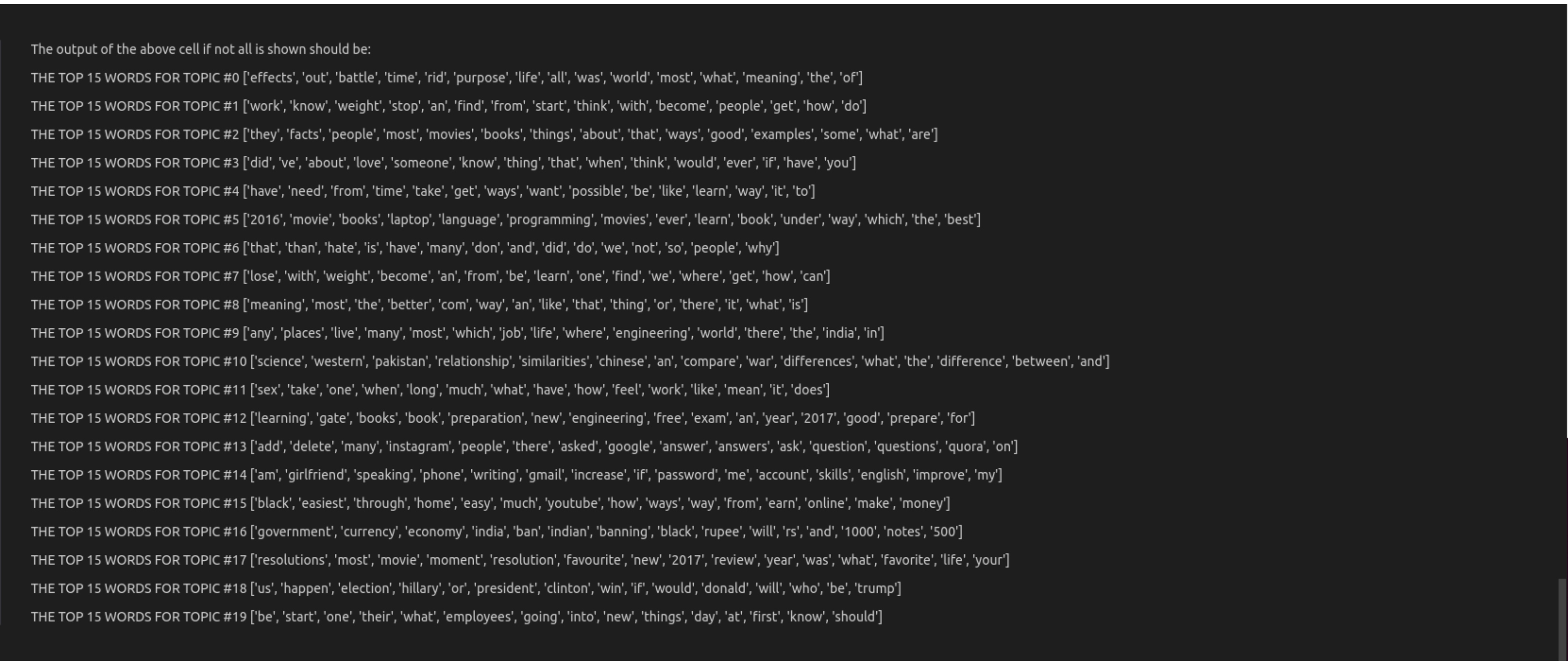
[41]

```
NMF
NMF(n_components=20, random_state=42)
```

```python
for index, topic in enumerate(nmf_model.components_):
    print(f"THE TOP 15 WORDS FOR TOPIC #{index}")
    print([v.get_feature_names_out()[i] for i in topic.argsort()[-15:]], end = "\n\n")
```

[46]

```
Output exceeds the size limit. Open the full output data in a text editor
THE TOP 15 WORDS FOR TOPIC #0
['effects', 'out', 'battle', 'time', 'rid', 'purpose', 'life', 'all', 'was', 'world', 'most', 'what', 'meaning', 'the', 'of']

THE TOP 15 WORDS FOR TOPIC #1
['work', 'know', 'weight', 'stop', 'an', 'find', 'from', 'start', 'think', 'with', 'become', 'people', 'get', 'how', 'do']

THE TOP 15 WORDS FOR TOPIC #2
['they', 'facts', 'people', 'most', 'movies', 'books', 'things', 'about', 'that', 'ways', 'good', 'examples', 'some', 'what', 'are']

THE TOP 15 WORDS FOR TOPIC #3
['did', 've', 'about', 'love', 'someone', 'know', 'thing', 'that', 'when', 'think', 'would', 'ever', 'if', 'have', 'you']

THE TOP 15 WORDS FOR TOPIC #4
['have', 'need', 'from', 'time', 'take', 'get', 'ways', 'want', 'possible', 'be', 'like', 'learn', 'way', 'it', 'to']

THE TOP 15 WORDS FOR TOPIC #5
['2016', 'movie', 'books', 'laptop', 'language', 'programming', 'movies', 'ever', 'learn', 'book', 'under', 'way', 'which', 'the', 'best']

THE TOP 15 WORDS FOR TOPIC #6
['that', 'than', 'hate', 'is', 'have', 'many', 'don', 'and', 'did', 'do', 'we', 'not', 'so', 'people', 'why']

THE TOP 15 WORDS FOR TOPIC #7
['lose', 'with', 'weight', 'become', 'an', 'from', 'be', 'learn', 'one', 'find', 'we', 'where', 'get', 'how', 'can']

THE TOP 15 WORDS FOR TOPIC #8
...

THE TOP 15 WORDS FOR TOPIC #19
['be', 'start', 'one', 'their', 'what', 'employees', 'going', 'into', 'new', 'things', 'day', 'at', 'first', 'know', 'should']
```



The output of the above cell if not all is shown should be:

THE TOP 15 WORDS FOR TOPIC #0 ['effects', 'out', 'battle', 'time', 'rid', 'purpose', 'life', 'all', 'was', 'world', 'most', 'what', 'meaning', 'the', 'of']

THE TOP 15 WORDS FOR TOPIC #1 ['work', 'know', 'weight', 'stop', 'an', 'find', 'from', 'start', 'think', 'with', 'become', 'people', 'get', 'how', 'do']

THE TOP 15 WORDS FOR TOPIC #2 ['they', 'facts', 'people', 'most', 'movies', 'books', 'things', 'about', 'that', 'ways', 'good', 'examples', 'some', 'what', 'are']

THE TOP 15 WORDS FOR TOPIC #3 ['did', 've', 'about', 'love', 'someone', 'know', 'thing', 'that', 'when', 'think', 'would', 'ever', 'if', 'have', 'you']

THE TOP 15 WORDS FOR TOPIC #4 ['have', 'need', 'from', 'time', 'take', 'get', 'ways', 'want', 'possible', 'be', 'like', 'learn', 'way', 'it', 'to']

THE TOP 15 WORDS FOR TOPIC #5 ['2016', 'movie', 'books', 'laptop', 'language', 'programming', 'movies', 'ever', 'learn', 'book', 'under', 'way', 'which', 'the', 'best']

THE TOP 15 WORDS FOR TOPIC #6 ['that', 'than', 'hate', 'is', 'have', 'many', 'don', 'and', 'did', 'do', 'we', 'not', 'so', 'people', 'why']

THE TOP 15 WORDS FOR TOPIC #7 ['lose', 'with', 'weight', 'become', 'an', 'from', 'be', 'learn', 'one', 'find', 'we', 'where', 'get', 'how', 'can']

THE TOP 15 WORDS FOR TOPIC #8 ['meaning', 'most', 'the', 'better', 'com', 'way', 'an', 'like', 'that', 'thing', 'or', 'there', 'it', 'what', 'is']

THE TOP 15 WORDS FOR TOPIC #9 ['any', 'places', 'live', 'many', 'most', 'which', 'job', 'life', 'where', 'engineering', 'world', 'there', 'the', 'india', 'in']

THE TOP 15 WORDS FOR TOPIC #10 ['science', 'western', 'pakistan', 'relationship', 'similarities', 'chinese', 'an', 'compare', 'war', 'differences', 'what', 'the', 'difference', 'between', 'and']

THE TOP 15 WORDS FOR TOPIC #11 ['sex', 'take', 'one', 'when', 'long', 'much', 'what', 'have', 'how', 'feel', 'work', 'like', 'mean', 'it', 'does']

THE TOP 15 WORDS FOR TOPIC #12 ['learning', 'gate', 'books', 'book', 'preparation', 'new', 'engineering', 'free', 'exam', 'an', 'year', '2017', 'good', 'prepare', 'for']

THE TOP 15 WORDS FOR TOPIC #13 ['add', 'delete', 'many', 'instagram', 'people', 'there', 'asked', 'google', 'answer', 'answers', 'ask', 'question', 'questions', 'quora', 'on']

THE TOP 15 WORDS FOR TOPIC #14 ['am', 'girlfriend', 'speaking', 'phone', 'writing', 'gmail', 'increase', 'if', 'password', 'me', 'account', 'skills', 'english', 'improve', 'my']

THE TOP 15 WORDS FOR TOPIC #15 ['black', 'easiest', 'through', 'home', 'easy', 'much', 'youtube', 'how', 'ways', 'way', 'from', 'earn', 'online', 'make', 'money']

THE TOP 15 WORDS FOR TOPIC #16 ['government', 'currency', 'economy', 'india', 'ban', 'indian', 'banning', 'black', 'rupee', 'will', 'rs', 'and', '1000', 'notes', '500']

THE TOP 15 WORDS FOR TOPIC #17 ['resolutions', 'most', 'movie', 'moment', 'resolution', 'favourite', 'new', '2017', 'review', 'year', 'was', 'what', 'favorite', 'life', 'your']

THE TOP 15 WORDS FOR TOPIC #18 ['us', 'happen', 'election', 'hillary', 'or', 'president', 'clinton', 'win', 'if', 'would', 'donald', 'will', 'who', 'be', 'trump']

THE TOP 15 WORDS FOR TOPIC #19 ['be', 'start', 'one', 'their', 'what', 'employees', 'going', 'into', 'new', 'things', 'day', 'at', 'first', 'know', 'should']