```
!pip install turicreate
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

Retreiving Wikipedia articles

```
import turicreate
```

```
people = turicreate.SFrame('/content/gdrive/My Drive/Turicreate/Week 4/people_wiki.sframe')
```

```
people.head()
```

URI	name	text
<http://dbpedia.org/resource/Digby_Morrell> ...	Digby Morrell	digby morrell born 10 october 1979 is a former ...
<http://dbpedia.org/resource/Alfred_J._Lewy> ...	Alfred J. Lewy	alfred j lewy aka sandy lewy graduated from ...
<http://dbpedia.org/resource/Harpdog_Brown> ...	Harpdog Brown	harpdog brown is a singer and harmonica player who ...
<http://dbpedia.org/resource/Franz_Rottensteiner> ...	Franz Rottensteiner	franz rottensteiner born in waidmannsfeld lower ...
<http://dbpedia.org/resource/G-Enka> ...	G-Enka	henry krvits born 30 december 1974 in tallinn ...
<http://dbpedia.org/resource/Sam_Henderson> ...	Sam Henderson	sam henderson born october 18 1969 is an ...
<http://dbpedia.org/resource/Aaron_LaCrate> ...	Aaron LaCrate	aaron lacrate is an american music producer ...
<http://dbpedia.org/resource/Trevor_Ferguson> ...	Trevor Ferguson	trevor ferguson aka john farrow born 11 november ...
<http://dbpedia.org/resource/Grant_Nelson> ...	Grant Nelson	grant nelson born 27 april 1971 in london ...
<http://dbpedia.org/resource/Cathy_Caruth> ...	Cathy Caruth	cathy caruth born 1955 is frank h t rhodes ...

[10 rows x 3 columns]

Count words for "Elton John"

```
people['word_count'] = turicreate.text_analytics.count_words(people['text'])
elton = people[people['name'] == 'Elton John']
```

```
elton['word_count']
```

```
dtype: dict
Rows: ?
```

```
[{'outside': 1.0, 'concert': 1.0, 'jubilee': 1.0, 'diamond': 1.0, 'composer': 1.0, 'party': 2
```

```
elton_word_count_table = elton[['word_count']].stack('word_count', new_column_name = ['word','count
elton_word_count_table
```

word	count
the	27.0
in	18.0
and	15.0
of	13.0
a	10.0
has	9.0
he	7.0
john	7.0
on	6.0
award	5.0

[255 rows x 2 columns]
Note: Only the head of the SFrame is printed.
You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.

Compute TF_IDF for "Elton John"

```
people['tfidf'] = turicreate.text_analytics.tf_idf(people['text'])
elton = people[people['name'] == 'Elton John']
elton
```

URI	name	text	tfidf
<http://dbpedia.org/resource/Elton_John> ...	Elton John	sir elton hercules john cbe born reginald ken ...	{'movements': 5.030658019760364, ...

[? rows x 4 columns]
Note: Only the head of the SFrame is printed. This SFrame is lazily evaluated.
You can use sf.materialize() to force materialization.

```
elton_tfidf_table = elton[['tfidf']].stack('tfidf',new_column_name=['word','tfidf']).sort('tfidf',a
elton_tfidf_table
```

word	tfidf
furnish	18.38947183999428
elton	17.482320270031995
billboard	17.30368095754203
john	13.93931279239831
songwriters	11.250406447031539
overallelton	10.986495389225194
tonightcandle	10.986495389225194
19702000	10.293348208665249
fivedecade	10.293348208665249
aids	10.262846934045534

[255 rows x 2 columns]
Note: Only the head of the SFrame is printed.
You can use print_rows(num_rows=m, num_columns=n) to print more rows and columns.

Compute cosine distance btw. elton and others

```
beckham = people[people['name'] == 'Victoria Beckham']
paul = people[people['name'] == 'Paul McCartney']
```

```
turicreate.distances.cosine(elton['tfidf'][0], beckham['tfidf'][0])
```

```
0.9567006376655429
```

```
turicreate.distances.cosine(elton['tfidf'][0], paul['tfidf'][0])
```

```
0.8250310029221779
```

Find Nearest Neighbors

```
word_count_model = turicreate.nearest_neighbors.create(people, features=['word_count'], label='name
tfidf_model = turicreate.nearest_neighbors.create(people, features=['tfidf'], label='name', distanc
```

```
Starting brute force nearest neighbors model training.
Validating distance components.
Initializing model data.
Initializing distances.
Done.
Starting brute force nearest neighbors model training.
Validating distance components.
Initializing model data.
Initializing distances.
Done.
```

```
word_count_model.query(elton)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 6.552ms      |
| Done         |         | 100         | 483.029ms    |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Elton John	2.220446049250313e-16	1
0	Cliff Richard	0.16142415258967036	2
0	Sandro Petrone	0.16822542751041114	3
0	Rod Stewart	0.16832716558706107	4
0	Malachi O'Doherty	0.177315545978884	5

[5 rows x 4 columns]

```
tfidf_model.query(elton)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 21.459ms     |
| Done         |         | 100         | 562.389ms    |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Elton John	-2.220446049250313e-16	1
0	Rod Stewart	0.7172196678927374	2
0	George Michael	0.7476009989692848	3
0	Sting (musician)	0.7476719544306141	4
0	Phil Collins	0.7511932487904706	5

```
word_count_model.query(beckham)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 9.553ms      |
| Done         |         | 100         | 471.692ms    |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Victoria Beckham	-2.220446049250313e-16	1
0	Mary Fitzgerald (artist)	0.20730703611504997	2
0	Adrienne Corri	0.21450978278754795	3
0	Beverly Jane Fry	0.21746646874079278	4
0	Raman Mundair	0.21769547499150488	5

[5 rows x 4 columns]

```
tfidf_model.query(beckham)
```

```
Starting pairwise querying.
+--------------+---------+-------------+--------------+
| Query points | # Pairs | % Complete. | Elapsed Time |
+--------------+---------+-------------+--------------+
| 0            | 1       | 0.00169288  | 17.594ms     |
| Done         |         | 100         | 519.667ms    |
+--------------+---------+-------------+--------------+
```

query_label	reference_label	distance	rank
0	Victoria Beckham	1.1102230246251565e-16	1
0	David Beckham	0.5481696102632145	2
0	Stephen Dow Beckham	0.7849867068283364	3
0	Mel B	0.8095855234085036	4
0	Caroline Rush	0.81982642291868	5

[5 rows x 4 columns]

더블클릭 또는 Enter 키를 눌러 수정

✓ 0초 오후 6:15에 완료됨