pca-operator-notebook

July 13, 2020

[1]:
```
import pandas as pd
import numpy as np
import spacy
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import altair as alt

alt.renderers.enable('notebook')
alt.renderers.enable('html')
spacy_nlp = spacy.load('en_core_web_sm')
```

[2]:
```
df = pd.read_csv("test_reviews_small.csv")
df
```

[2]:
```
                                                   Id  \
0   tripadvisor_review_0###0###usa_san francisco_f…
1   tripadvisor_review_0###1###usa_san francisco_f…
2   tripadvisor_review_0###2###usa_san francisco_f…
3   tripadvisor_review_0###3###usa_san francisco_f…
4   tripadvisor_review_0###4###usa_san francisco_f…
..                                                …
68  tripadvisor_review_6###8###usa_san francisco_f…
69  tripadvisor_review_6###9###usa_san francisco_f…
70  tripadvisor_review_6###10###usa_san francisco_…
71  tripadvisor_review_6###11###usa_san francisco_…
72  tripadvisor_review_6###12###usa_san francisco_…

                                               review  Unnamed: 2  Unnamed: 3  \
0   We stayed here for 8 nights on our trip to San…         NaN         NaN
1   From our arrival and check-in, to check-out, s…         NaN         NaN
2   We got a 2 bedroom apartment, with 2 bathrooms…         NaN         NaN
3   All appliances were state of the art, and our …         NaN         NaN
4   The apartment itself was very spacious, with l…         NaN         NaN
..                                                …           …         …
68  Note - the towncar is now a Cadillac v. a Mase…         NaN         NaN
```

```
69                                           Not biggy.         NaN         NaN
70                                 Still plush and comfy.         NaN         NaN
71  1. Have the driver drop you off on the far sid…         NaN         NaN
72                                   You won't regret it.         NaN         NaN

    Unnamed: 4  Unnamed: 5
0          NaN         NaN
1          NaN         NaN
2          NaN         NaN
3          NaN         NaN
4          NaN         NaN
..         …         …
68         NaN         NaN
69         NaN         NaN
70         NaN         NaN
71         NaN         NaN
72         NaN         NaN

[73 rows x 6 columns]
```

```
[3]: def basic_tokenizer(sentence):
         doc = spacy_nlp(sentence)
         tokens = [token.text for token in doc]
         return tokens
```

```
[4]: def clean_text(text):
         doc = spacy_nlp(text)
         toks = [token.text for token in doc if not (token.is_stop or token.
      ↪is_punct)]
         return " ".join(toks)
```

```
[5]: df['review'] = df['review'].str.lower()
     df['review'] = df['review'].map(clean_text)
     df['review']
```

```
[5]: 0     stayed 8 nights trip san francisco australia a…
     1     arrival check check service faultless faciliti…
     2     got 2 bedroom apartment 2 bathrooms fully equi…
     3     appliances state art room bosch washer separat…
     4     apartment spacious large windows automatic bli…
                                  …
     68                note towncar cadillac v. maserati
     69                                            biggy
     70                                      plush comfy
     71    1 driver drop far crissy field walk golden gat…
     72                                        wo regret
     Name: review, Length: 73, dtype: object
```

[6]:
```
vectorizer = TfidfVectorizer(tokenizer=basic_tokenizer)
tfidf_vectors = vectorizer.fit_transform(df['review'])
#df['review-tfidf'] = [v.toarray() for v in tfidf_vectors]
#df['review-tfidf'] = [list(v) for v in tfidf_vectors.A]
df['review-tfidf'] = list(tfidf_vectors)
df['review-tfidf']
```

[6]:
```
0       (0, 228)\t0.2936957192815969\n  (0, 13)\t0.3…
1       (0, 405)\t0.3423848832382839\n  (0, 139)\t0…
2       (0, 210)\t0.2584561161687453\n  (0, 130)\t0…
3       (0, 193)\t0.3363408767210485\n  (0, 125)\t0…
4       (0, 25)\t0.20129970891521648\n  (0, 223)\t0…
                                  …
68      (0, 386)\t0.46861135437172535\n  (0, 70)\t0…
69                                         (0, 53)\t1.0
70      (0, 89)\t0.7388456925735822\n  (0, 281)\t0.6…
71      (0, 397)\t0.3103007566254149\n  (0, 62)\t0.3…
72      (0, 299)\t0.7071067811865475\n  (0, 404)\t0…
Name: review-tfidf, Length: 73, dtype: object
```

[13]:
```
from scipy.sparse import vstack
print(df['review-tfidf'].shape)
tfidf_2d = vstack(df['review-tfidf'])
tfidf_2d = [list(v) for v in tfidf_2d.A]
tfidf_2d = np.stack(tfidf_2d, axis=0)
print(tfidf_2d.shape)
print(tfidf_2d)
```

```
(73,)
(73, 409)
[[0.         0.         0.         … 0.         0.         0.        ]
 [0.         0.         0.         … 0.         0.         0.        ]
 [0.         0.         0.         … 0.         0.         0.        ]
 …
 [0.         0.         0.         … 0.         0.         0.        ]
 [0.28301419 0.         0.         … 0.         0.         0.        ]
 [0.         0.         0.         … 0.         0.         0.        ]]
```

[31]:
```
pca = PCA(n_components=10)
pca_vectors = pca.fit_transform(tfidf_2d)
df['review-pca'] = pca_vectors.tolist()
# shape should be (73,10) because taking only first 10 pca components
print('shape of pca vectors is: ', pca_vectors.shape)
print('first row of pca vectors is: ', df['review-pca'][0])
```

```
shape of pca vectors is:  (73, 10)
first row of pca vectors is:  [0.48833997249634137, -0.08888450160382981,
-0.4323899543481765, 0.1212443411922488, -0.14307674857654296,
-0.1283073736805461, 0.13614033401407682, -0.027222870040782044,
-0.05891490437499333, -0.07853873268245114]
```

[32]:

```
df['pca_0'] = df['review-pca'].map(lambda x: x[0])
df['pca_0']
```

[32]:

```
0     0.488340
1    -0.088979
2    -0.035418
3    -0.151678
4    -0.090126
        …
68   -0.043464
69   -0.032528
70   -0.045832
71   -0.015672
72   -0.032528
Name: pca_0, Length: 73, dtype: float64
```

[33]:

```
df['pca_1'] = df['review-pca'].map(lambda x: x[1])
df['pca_1']
```

[33]:

```
0    -0.088885
1    -0.238910
2    -0.125668
3     0.002303
4     0.147720
        …
68   -0.035507
69   -0.025314
70   -0.069091
71   -0.004836
72   -0.025314
Name: pca_1, Length: 73, dtype: float64
```

[34]:

```
df.iloc[0]
```

[34]:

```
Id                tripadvisor_review_0###0###usa_san francisco_f…
review            stayed 8 nights trip san francisco australia a…
Unnamed: 2                                                     NaN
Unnamed: 3                                                     NaN
Unnamed: 4                                                     NaN
Unnamed: 5                                                     NaN
review-tfidf      [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
review-pca        [0.48833997249634137, -0.08888450160382981, -0…
pca_0                                                      0.48834
pca_1                                                   -0.0888845
```

```
Name: 0, dtype: object
```

[35]:
```
pca_data = []
c = 0
for index, row in df[['pca_0', 'pca_1']].iterrows():
    if c < 10:
        print(index, row['pca_0'], row['pca_1'])
    c += 1
    pca_data.append([row['pca_0'], row['pca_1']])
pca_data[:5]
```

```
0 0.48833997249634137 -0.08888450160382981
1 -0.08897918232101187 -0.238909796451915
2 -0.035417844859262755 -0.12566797059471962
3 -0.15167764607473028 0.0023026003254821743
4 -0.09012648842940259 0.14772010381609113
5 0.5956148741759717 0.02999107151817785
6 0.07064850434708214 0.04899340710106217
7 0.09380581396327828 0.06524188819261688
8 -0.10281067430332978 0.028270569884454477
9 0.08898698380053201 0.066923111857221
```

[35]:
```
[[0.48833997249634137, -0.08888450160382981],
 [-0.08897918232101187, -0.238909796451915],
 [-0.035417844859262755, -0.12566797059471962],
 [-0.15167764607473028, 0.0023026003254821743],
 [-0.09012648842940259, 0.14772010381609113]]
```

[36]:
```
kmeans = KMeans(n_clusters=6, n_init=10, verbose=0).fit(tfidf_2d)
cluster_preds = kmeans.predict(tfidf_2d)
print(cluster_preds)
print(cluster_preds.shape)
```

```
[5 1 1 1 2 5 4 4 0 4 1 5 2 1 0 3 2 1 1 0 0 1 1 0 0 0 5 0 5 3 4 0 5 0 3 0 0
 0 0 0 1 0 1 0 1 5 0 1 3 1 0 0 3 1 0 2 1 1 1 5 0 1 0 1 2 3 0 0 0 0 0 4 1]
(73,)
```

[37]:
```
df['review-clusters'] = [str(v) for v in cluster_preds]
df.iloc[0]
```

[37]:
```
Id                  tripadvisor_review_0###0###usa_san francisco_f…
review              stayed 8 nights trip san francisco australia a…
Unnamed: 2                                                        NaN
Unnamed: 3                                                        NaN
Unnamed: 4                                                        NaN
Unnamed: 5                                                        NaN
review-tfidf        [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, …
```

```
review-pca         [0.48833997249634137, -0.08888450160382981, -0…
pca_0                                                         0.48834
pca_1                                                      -0.0888845
review-clusters                                                     5
Name: 0, dtype: object
```

[73]:

```
print(kmeans.cluster_centers_.shape)
kmeans.cluster_centers_
```

```
(6, 409)
```

[73]:

```
array([[-1.73472348e-18,  1.30104261e-18,  2.16840434e-18, …,
         0.00000000e+00,  0.00000000e+00,  2.60208521e-18],
       [ 1.27774318e-02,  8.67361738e-19,  8.92827091e-03, …,
         2.08906696e-02,  0.00000000e+00,  0.00000000e+00],
       [-8.67361738e-19,  4.41415694e-02,  0.00000000e+00, …,
        -8.67361738e-19, -8.67361738e-19,  0.00000000e+00],
       [-8.67361738e-19,  0.00000000e+00,  0.00000000e+00, …,
         0.00000000e+00,  0.00000000e+00,  7.97823592e-02],
       [ 5.66028375e-02, -4.33680869e-19,  0.00000000e+00, …,
        -8.67361738e-19, -8.67361738e-19,  0.00000000e+00],
       [-1.73472348e-18, -4.33680869e-19,  0.00000000e+00, …,
        -8.67361738e-19,  6.81698717e-02, -8.67361738e-19]])
```

[38]:

```
alt.Chart(df).mark_circle().encode(
    alt.X('pca_0'),
    alt.Y('pca_1'),
    color='review-clusters'
)
```

[38]:

```
alt.Chart(…)
```

[42]:

```
# apply sentiment operator to review column
sid = SentimentIntensityAnalyzer()
df['review-sentiment'] = [sid.polarity_scores(r)['compound'] for r in␣
 ↪df['review']]
df['review-sentiment']
```

[42]:

```
0     0.6361
1     0.7717
2     0.0000
3     0.4215
4     0.0000
       …
68    0.0000
69    0.0000
70    0.0000
```

```
71   -0.2732
72   -0.4215
Name: review-sentiment, Length: 73, dtype: float64
```

[68]:

```
alt.Chart(df).mark_circle().encode(
    alt.X('pca_0'),
    alt.Y('pca_1'),
    color=alt.Color(field='review-sentiment', type='quantitative', scale=alt.
 ↪Scale(range=["crimson", "blue"])),
)
```

[68]:

```
alt.Chart(…)
```