

Weekly progress report – April 3 (7th report)

In the last series:

Next week i'll send the results of implementing a collaborative filtering algorithm for a wine recommendation engine

Sample dataset of user ratings:

User ID	Wine ID	Rating
1	1	5
1	2	4
1	3	4
2	1	4
2	2	5
2	3	3
3	1	3
3	2	2
3	3	5

Now suppose we want to recommend wines for user 1. Here's what the output might look like:

```
Top 3 recommended wines for user 1:
```

```
1. Wine 3 (score: 4.33)
2. Wine 2 (score: 4.00)
3. Wine 1 (score: 3.00)
```

The scores are calculated based on the similarity between each wine and the wines that User 1 has already rated. The more similar a wine is to the ones that User 1 likes, the higher its score will be. The algorithm uses a cosine similarity measure to calculate the similarity between wines.

Dataset for this work was taken from [kaggle](#)

country

The country that the wine is from

US	42%	Valid	130k	100%
		Mismatched	0	0%
France	17%	Missing	63	0%
Other (53374)	41%	Unique	43	
		Most Common	US	42%

description

119955

unique values

Valid	130k	100%
Mismatched	0	0%
Missing	0	0%
Unique	120k	
Most Common	This zesty r...	0%

```
path = '/Users/eseng/winego/winego/'
wine_150k = pd.read_csv(path + 'winemag-data_first150k.csv',index_col=0)
wine_130k = pd.read_csv(path + 'winemag-data-130k-v2.csv',index_col=0)
wine = pd.concat([wine_150k,wine_130k],axis=0)
#wine = wine_150k
print("Number of rows and columns:",wine.shape)
```

Output:

Number of rows and columns: (280901, 13)

I used the nearest neighbor algorithm and it is found using cosine similarity algorithm and it is defined as the cosine of angle between the vectors of two item vectors. The cosine will be larger as the items become closer.

```
col = ['province','variety','points']
wine1 = wine[col]
wine1 = wine1.dropna(axis=0)
wine1 = wine1.drop_duplicates(['province','variety'])
wine1 = wine1[wine1['points'] > 85]
wine_pivot = wine1.pivot(index= 'variety',columns='province',values='points').fillna(0)
wine_pivot_matrix = csr_matrix(wine_pivot)
knn = NearestNeighbors(n_neighbors=10,algorithm= 'brute', metric= 'cosine')
model_knn = knn.fit(wine_pivot_matrix)
```

This algorithm helps me to find top 5 recommendations for a random selection of wine

```
query_index = np.random.choice(wine_pivot.shape[0])
distance, indice = model_knn.kneighbors(wine_pivot.iloc[query_index,:].values.reshape(1,-1),n_neighbors=6)
for i in range(0, len(distance.flatten())):
    if i == 0:
        print("Recommendation for {0}:\n".format(wine_pivot.index[query_index]))
    else:
        print('{0}: {1} with distance: {2}'.format(i,wine_pivot.index[indice.flatten()[i]],distance.flatten()[i]))
```

Recommendation for Mourvèdre:

- 1: Rhône-style Red Blend with distance: 0.4177417062234413
- 2: G-S-M with distance: 0.4867814816359912
- 3: Grenache with distance: 0.504626613980101
- 4: Zinfandel with distance: 0.5155617285620482
- 5: Mourvèdre-Syrah with distance: 0.5252053849854071