Wine Recommendation

The goal of this project is to develop a wine recommendation algorithm based on the attributes of 100,646 different wines, as well as the 21 million ratings given to them by users. The datasets come the X-Wines data from Kaggle, which includes one dataset containing the attributes of each wine, and another dataset containing the rating information given by each user.

In [ ]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as mp
from sklearn.model_selection import train_test_split
from sklearn.decomposition import NMF
from sklearn.metrics.pairwise import cosine_similarity

# for better printing of dataframes
def display_():
    pd.set_option("display.max_rows", None)
    from IPython.core.display import display
    display(df)
```

In [ ]:
```
# read in data
wine_ratings = pd.read_csv("/Users/laurenflemmer/Desktop/wine_project/data/XWines_Test_1K_ratings.csv")
wines = pd.read_csv("/Users/laurenflemmer/Desktop/wine_project/data/XWines_Test_100_wines.csv")

wine_ratings.head()
```

Out[ ]:

	RatingID	UserID	WineID	Vintage	Rating	Date
0	3211	1209683	111478	1959	4.5	2016-08-08 00:50:22
1	27878	1209980	111478	1975	4.0	2018-08-12 17:09:39
2	31227	1258705	111478	1975	5.0	2014-11-16 19:52:38
3	41946	1139706	111478	1979	5.0	2014-12-22 02:30:15
4	61700	1240747	111478	1982	4.5	2019-10-21 02:01:10

In [ ]:
```
wines.head()
```

Out[ ]:

	WineID	WineName	Type	Elaborate	Grapes	Harmonize	ABV	Body	Acidity	Code	Country	RegionID	RegionName
0	100062	Origem Merlot	Red	Varietal/100%	['Merlot']	['Beef', 'Lamb', 'Veal', 'Grilled', 'Pizza', '...	13.0	Full-bodied	Medium	BR	Brazil	1002	Vale dos Vinhedos
1	100191	Reserva Chardonnay	White	Varietal/100%	['Chardonnay']	['Rich Fish', 'Seafood', 'Risotto', 'Poultry',...	13.0	Medium-bodied	Medium	BR	Brazil	1001	Serra Gaúcha
2	101847	Dona Antonia Porto Reserva Tawny	Dessert/Port	Assemblage/Blend	['Touriga Nacional', 'Touriga Franca', 'Tinta ...	['Appetizer', 'Sweet Dessert', 'Blue Cheese']	20.0	Very full-bodied	High	PT	Portugal	1031	Porto
3	102055	Fine Ruby Port	Dessert/Port	Assemblage/Blend	['Tinta Amarela', 'Tinta Barroca', 'Touriga Fr...	['Sweet Dessert', 'Cake', 'Fruit', 'Soft Cheese']	19.5	Very full-bodied	Medium	PT	Portugal	1031	Porto
4	102079	Maré Alta	White	Assemblage/Blend	['Loureiro', 'Alvarinho', 'Arinto']	['Fish', 'Shellfish', 'Vegetarian', 'Appetizer...	10.0	Very light-bodied	High	PT	Portugal	1034	Vinho Verde

Data Cleaning

In [ ]:
```
# check for NA values
print("# of NA values in wine ratings data: ", wine_ratings.isna().sum().sum(), "\n# of NA values in wines data: ", wines.
```

```
# of NA values in wine ratings data:  0
# of NA values in wines data:  0
```

Data Formatting

In order to get the data into a more usable format, I will change the formatting of the datasets, specifically, the wine reviews dataset. Since we care more about understanding specific user's taste in wines, rather than the individual reviews themselves, it's better for the review data to be user-based rather than review-based. In other words, the review data will not be formatted as a sparse matrix, where each row represents a different user, and each column represents a different wine. Each matrix empty (which can be empty) will represent the rating given to each wine by each user. This is the preferable data structure for the collaborative filtering that will be performed later.

In [ ]:
```
# pivot wine_ratings into rating matrix
pivot_ratings = wine_ratings.pivot(columns = "WineID", index = "UserID", values = "Rating")
pivot_ratings.head(10)
```

Out[ ]:

WineID / UserID	100062	100191	101847	102055	102079	102645	102902	103003	103070	103433	...	188472	190621	192515	193488	195300
1000045	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1000064	NaN	NaN	3.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1000196	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1000227	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1000272	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	4.0	NaN	...	NaN	NaN	NaN	2.5	NaN
1000292	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0	NaN	...	NaN	NaN	NaN	NaN	NaN
1000494	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1001050	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
1001862	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	4.0	NaN
1002016	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN

10 rows × 100 columns

In [ ]:
```
print("Size of rating matrix: ", pivot_ratings.shape)
```

```
Size of rating matrix:  (636, 100)
```

Matrix Factorization for Wine Preferences

When recommending wines to a user, it is obviously helpful to use the wine characteristics (e.g. Grape varietal, Body, Acidity) to recommend wines that are similar to a wine a user already likes. However, since the wine ratings of other users are available, it will also be useful to recommend a user wines that another user- one with similar taste- also liked.

Non-Negative Matrix Factorization for Rating Prediction

Since the user-rating matrix is extremely sparse, we need a way to generate predictions for the null entries of the matrix. In other words, to generate wine recommendations for users, we need predictions for the ratings they would give the wines they have not yet tasted. These predictions will be used to better understand what wines a user might like, and help inform the final wine recommendation. I'm choosing to use Non-Negative Matrix Factorization (NMF), because is it better suited for sparse matrices that contain exclusively positive values.

The hyperparameter "n_components" represents the number of latent factors/features in the data. These factors capture underlying patterns, structures, or characteristics that are not directly measured but can explain variations in the observed data. This number of latent factors can also be thought of as the rank of our final ranking matrix. Although tuning this hyperparameter via cross-validation would be ideal, there are few methods tailored for NMF, and those that exist are very computationally expensive. However, domain knowledge can be used to choose a suitable number of latent factors.

I've decided to use some of the features in the wine dataset to help inform my choice of "n_factors". Personally, the wine features that seem the most informative and conducive to a user's rating are:

- Wine Style (Red, White, Rose, etc.)
- Wine Type (100% Varietal, Assemblage/Blend)
- Grapes (Merlot, Chardonnay, etc.)
- Food Pairings (Seafood, Dessert, Fruit, etc.)
- Wine Aging (Type of barrel, Aging time)
- ABV
- Body (Full-Bodied, Light-Bodied, etc.)
- Acidity (Low, Medium, High)
- Wine Region (Italy, France, etc.)
- Wine Vintage (1999, 2022, etc.)

These 10 features seem like the most important factors that might go into a user's rating of a particular wine. Therefore, I am choosing "n_components = 10" as the number of latent factors/features present in the rating data.

In [ ]:
```
# convert df to np array
ratings_matrix = pivot_ratings.to_numpy()

# convert NaN values to 0
# since ratings are 1-5, there won't be any confusion with the existing ratings
missing_entries = pd.isna(ratings_matrix)
ratings_matrix[missing_entries] = 0
```

In [ ]:
```
# specify the number of components (latent factors)
n_components = 10

# NMF
nmf = NMF(n_components=n_components, init='random', random_state=10)

W = nmf.fit_transform(ratings_matrix)  # user matrix
H = nmf.components_  # item matrix
```

Interpreting the results of NMF, we now have the 2 decomposed matrices:

- The "W" matrix contains the user factors (user embeddings)
- The "H" matrix contains the item factors (item embeddings)

Now, the predicted ratings for the wines a user has not yet tasted can be calculated by performing a matrix multiplication of the user and item matrices.

In [ ]:
```
# multiply the user matrix by the item matrix to obtain the prediction matrix
predicted_ratings = W.dot(H)
print(predicted_ratings)

# make recommendations of only wines whose ratings were estimated
recommendation_set = np.zeros(ratings_matrix.shape)
recommendation_set[missing_entries] = predicted_ratings[missing_entries]
```

```
[[4.32216972e-02 2.47643427e-01 0.00000000e+00 ... 5.01087660e-03
  2.40024070e-02 2.59160064e-58]
 [0.00000000e+00 4.69153187e-01 3.67338757e+00 ... 6.12131508e-03
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 3.20758092e-56 3.42843035e-55 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 ...
 [0.00000000e+00 1.59398989e-03 1.00856281e-02 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [3.12014917e-01 2.07917779e-02 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 1.87086142e-57]
 [0.00000000e+00 4.36875867e-43 4.66955790e-42 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]]
```

In [ ]:
```
# convert back to df
user_ids = pivot_ratings.index
wine_ids = pivot_ratings.columns

nmf_df = pd.DataFrame(recommendation_set, columns = wine_ids, index = user_ids)

# from the predicted rating matrix, recommend the 5 wines with the highest rating
final_nmf_recs = pd.DataFrame(columns = user_ids)
```

In [ ]:
```
# convert back to df
user_ids = pivot_ratings.index
user_ids_str = [str(id) for id in user_ids]
wine_ids = pivot_ratings.columns

nmf_df = pd.DataFrame(recommendation_set, columns = wine_ids, index = user_ids)
final_nmf_recs = dict.fromkeys(user_ids_str)

# from the predicted rating matrix, recommend the 5 wines with the highest rating
for user in nmf_df.index:
    # get top 5 ratings
    top_5_nmf = nmf_df.loc[user].sort_values().iloc[-5::][::-1]

    # get corresponding wine names
    top_5_nmf_ids = top_5_nmf.index
    top_5_nmf_names = wines[wines['WineID'].isin(top_5_nmf_ids)].WineName
    top_5_nmf_wineries = wines[wines['WineID'].isin(top_5_nmf_ids)].WineryName

    winery_wine = top_5_nmf_wineries + " " + top_5_nmf_names

    # add to final_nmf_recs dict
    final_nmf_recs[str(user)] = winery_wine.to_list()

# convert final recs to df
final_nmf_recs = pd.DataFrame.from_dict(final_nmf_recs)
final_nmf_recs.index = ['First Recommendation', 'Second Recommendation', 'Third Recommendation', 'Fourth Recommendation',
```

Final Wine Recommendations via NMF Collaborative Filtering

Below are the final wine recommendations for each user using the Non-Negative Matrix Factorization collaborative filtering method.

In [ ]:
```
display(final_nmf_recs)
```

	1000045	1000064	1000196	1000227	1000272	1000292	1000494	1001050	1001862	1002016	...	20
First Recommendation	Bodega Garzón Reserva Cabernet Franc	Castello di Ama Al Poggio Chardonnay di Toscana	Aurora Reserva Chardonnay	Sandeman Fine Ruby Port	Porto Ferreira Dona Antonia Porto Reserva Tawny	Porto Ferreira Dona Antonia Porto Reserva Tawny	Sandeman Fine Ruby Port	Aurora Reserva Chardonnay	Emiliana Coyam	Domaine de Courcel Pommard Premier Cru 'Grand ...	...	R Charc
Second Recommendation	De Angeles Viña 1924 Gran Malbec	Kettmeir Müller Thurgau Alto Adige	Porto Ferreira Dona Antonia Porto Reserva Tawny	Château Suduiraut Sauternes (Premier Grand Cru...	Domaine Fourrier Vieille Vigne Gevrey-Chambertin	Sandeman Fine Ruby Port	Domaine Fourrier Vieille Vigne Gevrey-Chambertin	Porto Ferreira Dona Antonia Porto Reserva Tawny	Doña Paula Los Cardos Malbec	Henri Bourgeois Sancerre Les Baronnes Rosé	...	F A R
Third Recommendation	CHANDON Argentina Extra Brut Cuvée Spéciale Ba...	De Angeles Viña 1924 Gran Malbec	Castello di Ama Al Poggio Chardonnay di Toscana	Domaine Fourrier Vieille Vigne Gevrey-Chambertin	Emiliana Coyam	Castello di Ama Al Poggio Chardonnay di Toscana	Bottega Riserva Prêt-A-Porter	Henri Bourgeois Sancerre Les Baronnes Rosé	CHANDON Argentina Extra Brut Cuvée Spéciale Ba...	Louis Latour Santenay Rouge	...	Cas Charc di T
Fourth Recommendation	Illahe Estate Pinot Noir	CHANDON Argentina Extra Brut Cuvée Spéciale Ba...	Kettmeir Müller Thurgau Alto Adige	King Estate Pinot Noir	Doña Paula Los Cardos Malbec	Emiliana Coyam	Oxford Landing Cabernet Sauvignon-Shiraz	Emiliana Coyam	Oxford Landing Cabernet Sauvignon-Shiraz	Domaine Faiveley Les Fuées Chambolle-Musigny 1...	...	K T Altc
Fifth Recommendation	Matua Pinot Noir	Dashe Dry Creek Valley Zinfandel	Bottega Riserva Prêt-A-Porter	Illahe Estate Pinot Noir	De Angeles Viña 1924 Gran Malbec	Bodega Garzón Reserva Cabernet Franc	Dashe Dry Creek Valley Zinfandel	Dashe Dry Creek Valley Zinfandel	Illahe Estate Pinot Noir	Schloss Gobelsburg Zweigelt	...	B F F

5 rows × 636 columns

Using NMF User-Embeddings for User Similarity

When recommending wines to a user, it is obviously helpful to use the wine characteristics (e.g. Grape varietal, Body, Acidity) to recommend wines that are similar to a wine a user already likes. However, since the wine ratings of other users are available, it will also be useful to recommend a user wines that another user- one with similar taste- also liked. To understand which users are similar to a given user, we can use the individual's user embeddings from NMF, and find the distance to every other users' embeddings via cosine similarity.

In [ ]:
```
# calculate the cosine similarity between each NMF user embedding
temp_arr = np.zeros((len(W), len(W)))

for i in range(len(W)):
    for j in range(len(W)):
        temp_arr[i][j] = cosine_similarity(W[i,:].reshape(1,-1), W[j,:].reshape(1,-1))

cosine_sim = pd.DataFrame(temp_arr, columns = user_ids, index = user_ids)
display(cosine_sim)
```

UserID / UserID	1000045	1000064	1000196	1000227	1000272	1000292	1000494	1001050	1001862	1002016	...	
1000045	1.000000	1.886816e-01	0.000000e+00	0.377767	3.458447e-01	1.425233e-01	0.110170	1.053825e-01	0.917193	0.000000e+00	...	0
1000064	0.188682	1.000000e+00	2.121828e-55	0.084795	6.432243e-01	9.180306e-01	0.123780	5.626621e-01	0.229798	4.506790e-02	...	
1000196	0.000000	2.121828e-55	4.894341e-110	0.000000	1.138014e-55	2.051662e-55	0.000000	1.035788e-55	0.000000	1.257720e-57	...	
1000227	0.377767	8.479493e-02	0.000000e+00	1.000000	3.200696e-01	6.941802e-02	0.359189	1.405918e-01	0.338646	5.373584e-01	...	0
1000272	0.345845	6.432243e-01	1.138014e-55	0.320070	1.000000e+00	6.664842e-01	0.142221	3.799119e-01	0.493216	4.362031e-01	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
2046714	0.181087	1.117549e-01	0.000000e+00	0.323283	1.810437e-02	4.502922e-02	0.921631	8.089014e-01	0.052479	3.222070e-03	...	0
2048967	0.045623	3.550525e-02	0.000000e+00	0.697636	7.105187e-02	6.193794e-02	0.034220	0.000000e+00	0.056411	4.431975e-03	...	0
2049542	0.641835	8.642447e-02	1.860016e-56	0.424669	4.324838e-02	7.797015e-02	0.046443	8.125646e-02	0.430388	4.779765e-04	...	
2052399	0.014236	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000e+00	...	0
2062232	0.000000	2.889952e-42	6.666144e-97	0.000000	1.549986e-42	2.794386e-42	0.000000	1.410754e-42	0.000000	1.713028e-44	...	

636 rows × 636 columns

Now that I have acquired all of the pairwise user cosine similarities, I will find the users that are most similar to each user, and make recommendations based on which wines those similar users enjoyed. Each user's wine recommendations will be gathered from the 3 users that are the most "similar" to them.

In [ ]:
```
# get 3 most similar users for each user
similar_users_dic = dict.fromkeys(user_ids_str)

# for each user, get top 3 cosine similarities (not including themselves)
for i in range(len(W)):
    current_user = user_ids_str[i]
    top_3_cos = cosine_sim.iloc[i].sort_values().iloc[-4::][::-1][1::]
    top_3_users = list(top_3_cos.index)

    similar_users_dic[current_user] = top_3_users

# convert final dict to df
similar_users_df = pd.DataFrame.from_dict(similar_users_dic)
similar_users_df.index = ['Most Similar User', '2nd Most Similar User', '3rd Most Similar User']
display(similar_users_df)
```

	1000045	1000064	1000196	1000227	1000272	1000292	1000494	1001050	1001862	1002016	...	2025327	2032505	2037611	20
Most Similar User	1001862	1119608	1153142	1095942	1392981	1269514	1217479	1218423	1126357	1086669	...	1153142	1281181	1795555	17
2nd Most Similar User	1883175	1222565	1357819	1481948	1705206	1108227	1153860	1137670	1007263	1806447	...	1357819	1858380	1652802	16
3rd Most Similar User	1456446	1055875	1913676	1185372	1186232	1218182	1354311	1007104	1000045	1152133	...	1913676	1172707	1004166	10

3 rows × 636 columns

Now that I've found the 3 most similar users to each user (based on the NMF user embeddings), I will generate each user's wine recommendations based on the 3 highest-rated wines by those 3 most similar users. The recommendations will have the following order:

- From the most similar user
  - First Recommendation: 1st highest rated wine
  - Second Recommendation: 2nd highest rated wine
  - Third Recommendation: 3rd highest rated wine
- From the 2nd most similar user
  - Fourth Recommendation: 1st highest rated wine
  - Fifth Recommendation: 2nd highest rated wine
  - Sixth Recommendation: 3rd highest rated wine
- From the 3rd most similar user
  - Seventh Recommendation: 1st highest rated wine
  - Eighth Recommendation: 2nd highest rated wine
  - Ninth Recommendation: 3rd highest rated wine

Note: due to the sparsity of the original rating matrix, I will be using the rating-imputed NMF matrix as the rating matrix.

In [ ]:
```
test_current_user = user_ids_str[3]

print(similar_users_dic[test_current_user])

print(nmf_df.loc[similar_users_dic[test_current_user][1]].sort_values().iloc[-3::][::-1].index.tolist())
```

```
[1095942, 1481948, 1185372]
[111478, 179958, 162507]
```

In [ ]:
```
final_user_recs_dic = dict.fromkeys(user_ids_str)

# for each user, look at their 3 most similar users
# for each of those 3 similar users, obtain their top 3 rated wines
for i in range(len(W)):
    current_user = user_ids_str[i]
    top_3_wine_temp = []

    for sim_user in similar_users_dic[current_user]:
        # get top 3 wines for sim_user
        sim_user_top3 = nmf_df.loc[sim_user].sort_values().iloc[-3::][::-1].index.tolist()

        # get corresponding wine names
        sim_user_top3_names = wines[wines['WineID'].isin(sim_user_top3)].WineName
        sim_user_top3_wineries = wines[wines['WineID'].isin(sim_user_top3)].WineryName

        sim_user_winery_wine = sim_user_top3_names + " " + sim_user_top3_wineries
        top_3_wine_temp = top_3_wine_temp + list(sim_user_winery_wine)
    final_user_recs_dic[current_user] = top_3_wine_temp


# convert final dict to df
final_user_recs_df = pd.DataFrame.from_dict(final_user_recs_dic)
final_user_recs_df.index = ['First Recommendation', 'Second Recommendation', 'Third Recommendation', 'Fourth Recommendatio
                            'Sixth Recommendation', 'Seventh Recommendation', 'Eighth Recommendation', 'Nonth Recommendat
display(final_user_recs_df)
```

	1000045	1000064	1000196	1000227	1000272	1000292	1000494	1001050	1001862	1002016	...	2025327
First Recommendation	Los Cardos Malbec Doña Paula	Dona Antonia Porto Reserva Tawny Porto Ferreira	Reserva Chardonnay Aurora	Sauternes (Premier Grand Cru Classé) Château S...	Fine Ruby Port Sandeman	Al Poggio Chardonnay di Toscana Castello di Ama	Vieille Vigne Gevrey-Chambertin Domaine Fourrier	Dona Antonia Porto Reserva Tawny Porto Ferreira	Gran Malbec De Angeles Viña 1924	Pommard Premier Cru 'Grand Clos des Épenots' D...	...	Reserva Chardonnay Aurora
Second Recommendation	Extra Brut Cuvée Spéciale Baron B CHANDON Arge...	Los Cardos Malbec Doña Paula	Al Poggio Chardonnay di Toscana Castello di Ama	Pinot Noir King Estate	Coyam Emiliana	Müller Thurgau Alto Adige Kettmeir	Riserva Prêt-A-Porter Bottega	Sancerre Les Baronnes Rosé Henri Bourgeois	Extra Brut Cuvée Spéciale Baron B CHANDON Arge...	Sancerre Les Baronnes Rosé Henri Bourgeois	...	Al Poggio Chardonnay di Toscana Castello di Ama
Third Recommendation	Estate Pinot Noir Illahe	Gran Malbec De Angeles Viña 1924	Müller Thurgau Alto Adige Kettmeir	Estate Pinot Noir Illahe	Gran Malbec De Angeles Viña 1924	Reserva Cabernet Franc Bodega Garzón	Dry Creek Valley Zinfandel Dashe	Dry Creek Valley Zinfandel Dashe	Pinot Noir Matua	Les Fuées Chambolle-Musigny 1er Cru Domaine Fa...	...	Müller Thurgau Alto Adige Kettmeir
Fourth Recommendation	Gran Malbec De Angeles Viña 1924	Dona Antonia Porto Reserva Tawny Porto Ferreira	Reserva Chardonnay Aurora	Sauternes (Premier Grand Cru Classé) Château S...	Coyam Emiliana	Reserva Chardonnay Aurora	Riserva Prêt-A-Porter Bottega	Dona Antonia Porto Reserva Tawny Porto Ferreira	Gran Malbec De Angeles Viña 1924	Pommard Premier Cru 'Grand Clos des Épenots' D...	...	Reserva Chardonnay Aurora
Fifth Recommendation	Pinot Noir King Estate	Los Cardos Malbec Doña Paula	Al Poggio Chardonnay di Toscana Castello di Ama	Coyam Emiliana	Los Cardos Malbec Doña Paula	Al Poggio Chardonnay di Toscana Castello di Ama	Cabernet Sauvignon-Shiraz Oxford Landing	Sancerre Les Baronnes Rosé Henri Bourgeois	Extra Brut Cuvée Spéciale Baron B CHANDON Arge...	Sancerre Les Baronnes Rosé Henri Bourgeois	...	Al Poggio Chardonnay di Toscana Castello di Ama
Sixth Recommendation	Estate Pinot Noir Illahe	Gran Malbec De Angeles Viña 1924	Müller Thurgau Alto Adige Kettmeir	Estate Pinot Noir Illahe	Gran Malbec De Angeles Viña 1924	Müller Thurgau Alto Adige Kettmeir	Dry Creek Valley Zinfandel Dashe	Dry Creek Valley Zinfandel Dashe	Pinot Noir Matua	Les Fuées Chambolle-Musigny 1er Cru Domaine Fa...	...	Müller Thurgau Alto Adige Kettmeir
Seventh Recommendation	Gran Malbec De Angeles Viña 1924	Dona Antonia Porto Reserva Tawny Porto Ferreira	Dona Antonia Porto Reserva Tawny Porto Ferreira	Dona Antonia Porto Reserva Tawny Porto Ferreira	Coyam Emiliana	Fine Ruby Port Sandeman	Fine Ruby Port Sandeman	Dona Antonia Porto Reserva Tawny Porto Ferreira	Gran Malbec De Angeles Viña 1924	Sancerre Les Baronnes Rosé Henri Bourgeois	...	Dona Antonia Porto Reserva Tawny Porto Ferreira
Eighth Recommendation	Pinot Noir King Estate	Los Cardos Malbec Doña Paula	Al Poggio Chardonnay di Toscana Castello di Ama	Fine Ruby Port Sandeman	Los Cardos Malbec Doña Paula	Al Poggio Chardonnay di Toscana Castello di Ama	Riserva Prêt-A-Porter Bottega	Sancerre Les Baronnes Rosé Henri Bourgeois	Extra Brut Cuvée Spéciale Baron B CHANDON Arge...	Les Fuées Chambolle-Musigny 1er Cru Domaine Fa...	...	Al Poggio Chardonnay di Toscana Castello di Ama
Nonth Recommendation	Estate Pinot Noir Illahe	Gran Malbec De Angeles Viña 1924	Müller Thurgau Alto Adige Kettmeir	Sauternes (Premier Grand Cru Classé) Château S...	Gran Malbec De Angeles Viña 1924	Müller Thurgau Alto Adige Kettmeir	Dry Creek Valley Zinfandel Dashe	Dry Creek Valley Zinfandel Dashe	Estate Pinot Noir Illahe	Zweigelt Schloss Gobelsburg	...	Müller Thurgau Alto Adige Kettmeir

9 rows × 636 columns