МГТУ им. Н. Э. Баумана, кафедра ИУ5
курс “Методы машинного обучения”

Лабораторная работа №4

«Создание рекомендательной модели»

ВЫПОЛНИЛ:

Сергеев М.К.

Группа: ИУ5-22М

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва 2022

Задание:

1. Выбрать произвольный набор данных (датасет), предназначенный для построения рекомендательных моделей.
2. Опираясь на материалы лекции, сформировать рекомендации для одного пользователя (объекта) двумя произвольными способами.
3. Сравнить полученные рекомендации (если это возможно, то с применением метрик).
4. Сформировать отчет и разместить его в своем репозитории на github.

Выполнение работы:

In [1]:

```
import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from IPython.display import Image
from IPython.display import Image
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.datasets import load_iris, load_boston
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import accuracy_score, balanced_accuracy_score
from sklearn.metrics import precision_score, recall_score, f1_score, classification_report
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, export_graphviz
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.ensemble import ExtraTreesClassifier, ExtraTreesRegressor
from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.metrics import mean_absolute_error, mean_squared_error, mean_squared_log_error, median_absolute_err
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics.pairwise import cosine_similarity, euclidean_distances, manhattan_distances
from collections import defaultdict
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib_venn import venn2
%matplotlib inline
sns.set(style="ticks")
```

Чтение и обработка данных

In [2]:

```
data = pd.read_csv('winemag-data-130k-v2.csv')
data.head()
```

Out[2]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	P
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	St. Julian 2013 Reserve Late Harvest Riesling ...	
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Sweet Cheeks 2012 Vintner's Reserve Wild	

In [3]:

```
data.shape
```

Out[3]:

```
(129971, 14)
```

In [4]:

```
description_data = data[data['description'].notnull()]
description_data.shape
```

Out[4]:

```
(129971, 14)
```

In [5]:
```python
title = description_data['title'].values
title[0:5]
```

Out[5]:
```
array(['Nicosia 2013 Vulkà Bianco  (Etna)',
       'Quinta dos Avidagos 2011 Avidagos Red (Douro)',
       'Rainstorm 2013 Pinot Gris (Willamette Valley)',
       'St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)',
       "Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)"],
      dtype=object)
```

In [6]:
```python
descriptions = description_data['description'].values
descriptions[0:5]
```

Out[6]:
```
array(["Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive, offer
ing unripened apple, citrus and dried sage alongside brisk acidity.",
       "This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with j
uicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better f
rom 2016.",
       'Tart and snappy, the flavors of lime flesh and rind dominate. Some green pineapple pokes through, with cr
isp acidity underscoring the flavors. The wine was all stainless-steel fermented.',
       'Pineapple rind, lemon pith and orange blossom start off the aromas. The palate is a bit more opulent, wit
h notes of honey-drizzled guava and mango giving way to a slightly astringent, semidry finish.',
       "Much like the regular bottling from 2012, this comes across as rather rough and tannic, with rustic, eart
hy, herbal characteristics. Nonetheless, if you think of it as a pleasantly unfussy country wine, it's a good com
panion to a hearty winter stew."],
      dtype=object)
```

In [7]:
```python
description_data.keys()
```

Out[7]:
```
Index(['Unnamed: 0', 'country', 'description', 'designation', 'points',
       'price', 'province', 'region_1', 'region_2', 'taster_name',
       'taster_twitter_handle', 'title', 'variety', 'winery'],
      dtype='object')
```

In [8]:
```python
wine_ids = description_data['Unnamed: 0'].values
wine_ids
```

Out[8]:
```
array([     0,      1,      2, ..., 129968, 129969, 129970])
```

In [9]:
```python
%%time
tfidf = TfidfVectorizer()
description_matrix = tfidf.fit_transform(descriptions)
description_matrix
```

```
CPU times: user 5.75 s, sys: 110 ms, total: 5.86 s
Wall time: 6.87 s
```

In [10]:
```python
description_matrix
```

Out[10]:
```
<129971x31275 sparse matrix of type '<class 'numpy.float64'>'
	with 4475479 stored elements in Compressed Sparse Row format>
```

Фильтрация на основе содержания. Метод k-ближайших соседей

In [11]:
```python
class SimplerKnnRecomender:
  def __init__(self, X_matrix, X_ids, X_title, X_overview):
        """
        Входные параметры:
        X_matrix - обучающая выборка (матрица объект-признак)
        X_ids - массив идентификаторов объектов
        X_title - массив названий объектов
        X_overview - массив описаний объектов
        """
```

```
        #Сохраняем параметры в переменных объекта
        self._X_matrix = X_matrix
        self.df = pd.DataFrame(
            {'id': pd.Series(X_ids, dtype='int'),
            'title': pd.Series(X_title, dtype='str'),
            'overview': pd.Series(X_overview, dtype='str'),
            'dist': pd.Series([], dtype='float')})

    def recommend_for_single_object(self, K: int, \
                X_matrix_object, cos_flag = True, manh_flag = False):
        """
        Метод формирования рекомендаций для одного объекта.
        Входные параметры:
        K - количество рекомендуемых соседей
        X_matrix_object - строка матрицы объект-признак, соответствующая объекту
        cos_flag - флаг вычисления косинусного расстояния
        manh_flag - флаг вычисления манхэттэнского расстояния
        Возвращаемое значение: K найденных соседей
        """

        scale = 1000000
        # Вычисляем косинусную близость
        if cos_flag:
            dist = cosine_similarity(self._X_matrix, X_matrix_object)
            self.df['dist'] = dist * scale
            res = self.df.sort_values(by='dist', ascending=False)
            # Не учитываем рекомендации с единичным расстоянием,
            # так как это искомый объект
            res = res[res['dist'] < scale]

        else:
            if manh_flag:
                dist = manhattan_distances(self._X_matrix, X_matrix_object)
            else:
                dist = euclidean_distances(self._X_matrix, X_matrix_object)
            self.df['dist'] = dist * scale
            res = self.df.sort_values(by='dist', ascending=True)
            # Не учитываем рекомендации с единичным расстоянием,
            # так как это искомый объект
            res = res[res['dist'] > 0.0]

        # Оставляем K первых рекомендаций
        res = res.head(K)
        return res
```

In [48]:

```
test_id = 11
print(title[test_id])
print(descriptions[test_id])
```

```
Leon Beyer 2012 Gewurztraminer (Alsace)
This is a dry wine, very spicy, with a tight, taut texture and strongly mineral character layered with citrus as well as pepper. It's a food wine with its almost crisp aftertaste.
```

In [49]:

```
test_matrix = description_matrix[test_id]
test_matrix
```

Out[49]:

```
<1x31275 sparse matrix of type '<class 'numpy.float64'>'
        with 25 stored elements in Compressed Sparse Row format>
```

In [50]:

```
skr1 = SimplerKnnRecomender(description_matrix, wine_ids, title, descriptions)
```

In [51]:

```
# 15 вин, наиболее похожих на Leon Beyer 2012 Gewurztraminer (Alsace)
# в порядке убывания схожести на основе косинусного сходства
rec1 = skr1.recommend_for_single_object(15, test_matrix)
rec1
```

Out[51]:

	id	title	overview	dist
24045	24045	Domaine Michel Thomas et Fils 2015 Rosé (Sance...	The wine is textured and tight with crisp acid...	633624.990866
90700	90700	Henri de Villamont 2014 Morgeot Premier Cru (...	This wine is still tight and crisp. It has ple...	442624.176096
58330	58330	Schröder & Schÿler 2013 Chartron la Fleur (Bo...	The wine is tight and nervy, very fresh, crisp...	432556.705703
66081	66081	Maison Champy 2014 Viré-Clessé	This taut and structured wine has weight as we...	430242.028148
78572	78572	Domaine Olivier Merlin 2014 Mâcon La Roche Vi...	This wine is tight, structured and taut. Still...	428504.458538
105230	105230	Domaine Nigri 2013 Pierre de Lune (Jurançon Sec)	This rich and ripe wine is full of apricot and...	425886.605501
25907	25907	Louis Max 2014 Mâcon-Villages	Tight and structured, this wine has minerality...	424385.444731

99011	99011	Joseph Drouhin 2013 Les Clos (Macon-Bussières)	This crisp wine offers plenty of acidity as we...	423757.525560
5406	5406	Aveleda 2015 Alvarinho (Vinho Verde)	Ripe Alvarinho gives a wine that is rich as we...	421592.529700
22652	22652	Maison Malet Roquefort 2012 Léo de la Gaffeliè...	Very herbaceous in character, this is a wine t...	418388.507228
129715	129715	Boeckel 2012 Vieilles Vignes Sylvaner (Alsace)	Intensely peppery as well as fruity, this is a...	416866.789965
119482	119482	Boeckel 2012 Vieilles Vignes Sylvaner (Alsace)	Intensely peppery as well as fruity, this is a...	416866.789965
21920	21920	Moncigale 2014 Frais et Délicat Rosé (Coteaux ...	This is crisp, fruity with apple and citrus fl...	411434.544994
96505	96505	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	408987.116976
92292	92292	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	408987.116976

In [52]:

```
# При поиске с помощью Евклидова расстояния получаем такой же результат
rec2 = skr1.recommend_for_single_object(15, test_matrix, cos_flag = False)
rec2
```

Out[52]:

	id	title	overview	dist
24045	24045	Domaine Michel Thomas et Fils 2015 Rosé (Sance...	The wine is textured and tight with crisp acid...	8.560082e+05
90700	90700	Henri de Villamont 2014 Morgeot Premier Cru (...	This wine is still tight and crisp. It has ple...	1.055818e+06
58330	58330	Schröder & Schÿler 2013 Chartron la Fleur (Bo...	The wine is tight and nervy, very fresh, crisp...	1.065311e+06
66081	66081	Maison Champy 2014 Viré-Clessé	This taut and structured wine has weight as we...	1.067481e+06
78572	78572	Domaine Olivier Merlin 2014 Mâcon La Roche Vi...	This wine is tight, structured and taut. Still...	1.069108e+06
105230	105230	Domaine Nigri 2013 Pierre de Lune (Jurançon Sec)	This rich and ripe wine is full of apricot and...	1.071553e+06
25907	25907	Louis Max 2014 Mâcon-Villages	Tight and structured, this wine has minerality...	1.072953e+06
99011	99011	Joseph Drouhin 2013 Les Clos (Macon-Bussières)	This crisp wine offers plenty of acidity as we...	1.073539e+06
5406	5406	Aveleda 2015 Alvarinho (Vinho Verde)	Ripe Alvarinho gives a wine that is rich as we...	1.075553e+06
22652	22652	Maison Malet Roquefort 2012 Léo de la Gaffeliè...	Very herbaceous in character, this is a wine t...	1.078528e+06
119482	119482	Boeckel 2012 Vieilles Vignes Sylvaner (Alsace)	Intensely peppery as well as fruity, this is a...	1.079938e+06
129715	129715	Boeckel 2012 Vieilles Vignes Sylvaner (Alsace)	Intensely peppery as well as fruity, this is a...	1.079938e+06
21920	21920	Moncigale 2014 Frais et Délicat Rosé (Coteaux ...	This is crisp, fruity with apple and citrus fl...	1.084957e+06
92292	92292	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	1.087210e+06
96505	96505	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	1.087210e+06

In [53]:

```
# Манхэттэнское расстояние дает несколько иные результаты поиска
rec3 = skr1.recommend_for_single_object(15, test_matrix,
                                        cos_flag = False, manh_flag = True)
rec3
```

Out[53]:

	id	title	overview	dist
24045	24045	Domaine Michel Thomas et Fils 2015 Rosé (Sance...	The wine is textured and tight with crisp acid...	3.865262e+06
22652	22652	Maison Malet Roquefort 2012 Léo de la Gaffeliè...	Very herbaceous in character, this is a wine t...	5.251729e+06
35502	35502	Château de Piote 2012 Perles (Crémant de Bord...	Tight and sharp, this is an herbaceous wine wi...	5.312967e+06
58330	58330	Schröder & Schÿler 2013 Chartron la Fleur (Bo...	The wine is tight and nervy, very fresh, crisp...	5.316624e+06
25907	25907	Louis Max 2014 Mâcon-Villages	Tight and structured, this wine has minerality...	5.354298e+06
21920	21920	Moncigale 2014 Frais et Délicat Rosé (Coteaux ...	This is crisp, fruity with apple and citrus fl...	5.452536e+06
97201	97201	Ravoire et Fils 2013 Domaine la Rabiotte Rosé ...	Tight, zingy and crisp, this wine has fresh, c...	5.535851e+06
70762	70762	Château du Seuil 2015 Domaine du Seuil (Borde...	The wine is tight and mineral in character. It...	5.564448e+06
128577	128577	Ravoire et Fils 2014 Domaine Bel Eouve Rosé (C...	This is a tangy, spicy wine, a character that ...	5.628584e+06
78572	78572	Domaine Olivier Merlin 2014 Mâcon La Roche Vi...	This wine is tight, structured and taut. Still...	5.644448e+06
92292	92292	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	5.653916e+06
96505	96505	Domaine Alban Roblin 2014 Rosé (Sancerre)	This is a fresh wine with caramel as well as r...	5.653916e+06
108912	108912	Quinta do Portal 2012 Colheita Rosé (Douro)	This rosé is almost as rich as a red wine, the...	5.701024e+06
66081	66081	Maison Champy 2014 Viré-Clessé	This taut and structured wine has weight as we...	5.734040e+06
88898	88898	Markus Huber 2009 Hugo Grüner Veltliner (Niede...	Very crisp fruit, with light acidity and a tau...	5.751297e+06

Коллаборативная фильтрация. Метод на основе сингулярного разложения

In [18]:

```
data.head()
```

Out[18]:

	Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	P
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	St. Julian 2013 Reserve Late Harvest Riesling ...	
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Sweet Cheeks 2012 Vintner's Reserve Wil d	

In [19]:

```
data3 = data[30000:55000]
```

In [20]:

```
# Количество уникальных дегустаторов
len(data3['taster_name'].unique())
```

Out[20]: 20

In [21]:

```
# Количество уникальных вин
len(data3['title'].unique())
```

Out[21]: 24517

In [22]:

```
# Сформируем матрицу взаимодействий на основе рейтингов
# Используется идея из статьи - https://towardsdatascience.com/beginners-guide-to-creating-an-svd-recommender-sy
def create_utility_matrix(data):
    itemField = 'title'
    userField = 'taster_name'
    valueField = 'points'

    userList = data[userField].tolist()
    itemList = data[itemField].tolist()
    valueList = data[valueField].tolist()

    users = list(set(userList))
    items = list(set(itemList))

    users_index = {users[i]: i for i in range(len(users))}
    pd_dict = {item: [0.0 for i in range(len(users))] for item in items}

    for i in range(0,data.shape[0]):
        item = itemList[i]
        user = userList[i]
        value = valueList[i]
        pd_dict[item][users_index[user]] = value

    X = pd.DataFrame(pd_dict)
    X.index = users

    itemcols = list(X.columns)
    items_index = {itemcols[i]: i for i in range(len(itemcols))}
```

```python
    return X, users_index, items_index
```

In [23]:

```python
%%time
user_item_matrix, users_index, items_index = create_utility_matrix(data3)
```

```
CPU times: user 987 ms, sys: 14.4 ms, total: 1 s
Wall time: 1.01 s
```

In [24]:

```python
user_item_matrix
```

Out[24]:

	Marqués de Cáceres 2015 Satinela Semi Dulce White (Rioja)	Saint K 2015 Grenache Blanc (Paso Robles)	Latah Creek 2005 Cabernet-Syrah (Washington)	Keating 2012 Beckstoffer Georges III Cabernet Sauvignon (Rutherford)	Cave du Château des Loges 2015 Prestige (Beaujolais-Villages)	Citari 2012 Sorgente (Lugana)	Sanguis 2009 Ode to Sunshine Bien Nacido Vineyard Chardonnay (Santa Maria Valley)	Pellegrini 2005 Olivet Lane Estate Reserve Chardonnay (Russian River Valley)	Blue Rock 2006 Cabernet Sauvignon (Alexander Valley)	Flora Springs 2012 Flora's Legacy Chardonnay (Napa Valley)	...	Pr P
NaN	0.0	0.0	0.0	0.0	0.0	0.0	91.0	90.0	89.0	92.0	...	
Anna Lee C. Iijima	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Sean P. Sullivan	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Roger Voss	0.0	0.0	0.0	0.0	85.0	0.0	0.0	0.0	0.0	0.0	...	
Virginie Boone	0.0	0.0	0.0	91.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Paul Gregutt	0.0	0.0	87.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Michael Schachner	85.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Lauren Buzzeo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Alexander Peartree	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Carrie Dykes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Joe Czerwinski	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Jim Gordon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Jeff Jenssen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Anne Krebiehl MW	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Susan Kostrzewa	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Mike DeSimone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Fiona Adams	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Christina Pickard	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Matt Kettmann	0.0	90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Kerin O'Keefe	0.0	0.0	0.0	0.0	0.0	85.0	0.0	0.0	0.0	0.0	...	

20 rows × 24517 columns

In [25]:

```python
# Выделение тестовой строки
user_item_matrix__test = user_item_matrix.loc[['Kerin O’Keefe']]
user_item_matrix__test
```

Out[25]:

Marqués de Cáceres 2015 Satinela Semi Dulce	Saint K 2015 Grenache Blanc (Paso	Latah Creek 2005 Cabernet-Syrah (Washington)	Keating 2012 Beckstoffer Georges III Cabernet Sauvignon	Cave du Château des Loges 2015 Prestige (Beaujolais-	Citari 2012 Sorgente (Lugana)	Sanguis 2009 Ode to Sunshine Bien Nacido Vineyard Chardonnay (Santa	Pellegrini 2005 Olivet Lane Estate Reserve Chardonnay (Russian River	Blue Rock 2006 Cabernet Sauvignon (Alexander	Flora Springs 2012 Flora's Legacy Chardonnay (Napa	...	Babc 2 Precoci Pinot (S Bar

	White (Rioja)	Robles)		(Rutherford)	Villages)		Maria Valley)	Valley)	Valley)	Valley)		Cou
Kerin O'Keefe	0.0	0.0	0.0	0.0	0.0	85.0	0.0	0.0	0.0	0.0	...	

1 rows × 24517 columns

In [26]:

```
#taster_names = description_data['taster_name'].unique()
taster_names = np.delete(data3['taster_name'].unique(), 0)
taster_names = np.delete(taster_names, 7)
taster_names
```

Out[26]:

```
array(['Jim Gordon', 'Michael Schachner', 'Matt Kettmann',
       'Sean P. Sullivan', 'Roger Voss', 'Virginie Boone',
       'Joe Czerwinski', 'Paul Gregutt', 'Mike DeSimone', 'Jeff Jenssen',
       nan, 'Anna Lee C. Iijima', 'Susan Kostrzewa', 'Lauren Buzzeo',
       'Alexander Peartree', 'Fiona Adams', 'Carrie Dykes',
       'Christina Pickard'], dtype=object)
```

In [27]:

```
# Оставшаяся часть матрицы для обучения
user_item_matrix__train = user_item_matrix.loc[taster_names]
user_item_matrix__train
```

Out[27]:

	Marqués de Cáceres 2015 Satinela Semi Dulce White (Rioja)	Saint K 2015 Grenache Blanc (Paso Robles)	Latah Creek 2005 Cabernet-Syrah (Washington)	Keating 2012 Beckstoffer Georges III Cabernet Sauvignon (Rutherford)	Cave du Château des Loges 2015 Prestige (Beaujolais-Villages)	Citari 2012 Sorgente (Lugana)	Sanguis 2009 Ode to Sunshine Bien Nacido Vineyard Chardonnay (Santa Maria Valley)	Pellegrini 2005 Olivet Lane Estate Reserve Chardonnay (Russian River Valley)	Blue Rock 2006 Cabernet Sauvignon (Alexander Valley)	Flora Springs 2012 Flora's Legacy Chardonnay (Napa Valley)	...	B Prec Pin B C
Jim Gordon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Michael Schachner	85.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Matt Kettmann	0.0	90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Sean P. Sullivan	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Roger Voss	0.0	0.0	0.0	0.0	85.0	0.0	0.0	0.0	0.0	0.0	...	
Virginie Boone	0.0	0.0	0.0	91.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Joe Czerwinski	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Paul Gregutt	0.0	0.0	87.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Mike DeSimone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Jeff Jenssen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
NaN	0.0	0.0	0.0	0.0	0.0	0.0	91.0	90.0	89.0	92.0	...	
Anna Lee C. Iijima	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Susan Kostrzewa	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Lauren Buzzeo	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Alexander Peartree	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Fiona Adams	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Carrie Dykes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
Christina Pickard	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

18 rows × 24517 columns

In [28]:

```
%%time
U, S, VT = np.linalg.svd(user_item_matrix__train.T)
V = VT.T
```

```
CPU times: user 40.5 s, sys: 4.98 s, total: 45.5 s
Wall time: 29.1 s
```

In [29]:

```
# Матрица соотношения между дегустаторами и латентными факторами
U.shape
```

Out[29]:

```
(24517, 24517)
```

In [30]:

```
# Матрица соотношения между объектами и латентными факторами
V.shape
```

Out[30]:

```
(18, 18)
```

In [31]:

```
S.shape
```

Out[31]:

```
(18,)
```

In [32]:

```
Sigma = np.diag(S)
Sigma.shape
```

Out[32]:

```
(18, 18)
```

In [33]:

```
# Диагональная матрица сингулярных значений
Sigma
```

Out[33]:

```
array([[6328.37615756,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        , 6214.00788753,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        , 4603.41568838,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        , 3880.90866797,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
        3683.95055254,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        , 3004.09187609,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        , 2830.86117639,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        , 2714.05324192,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
```

```
           0.        ,    0.        ,    0.        ,    0.        ,
        2451.15054617,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        , 2416.46258391,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        , 1559.43034471,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        , 1283.41926119,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
         922.26406197,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,  825.2545062 ,    0.        ,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,  745.06040024,    0.        ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,  390.0179483 ,
           0.        ,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
         177.03107072,    0.        ],
       [   0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,    0.        ,    0.        ,    0.        ,
           0.        ,  124.45882853]])
```

In [34]:

```python
# Используем 3 первых сингулярных значения
r=3
Ur = U[:, :r]
Sr = Sigma[:r, :r]
Vr = V[:, :r]
# Матрица соотношения между новым дегустатором и латентными факторами
test_user = np.mat(user_item_matrix__test.values)
test_user.shape, test_user
```

Out[34]:

```
((1, 24517), matrix([[ 0.,  0.,  0., ..., 87.,  0.,  0.]]))
```

In [35]:

```python
tmp = test_user * Ur * np.linalg.inv(Sr)
tmp
```

Out[35]:

```
matrix([[ 3.78394162e-04,  4.35827216e-06, -2.92221682e-18]])
```

In [36]:

```python
test_user_result = np.array([tmp[0,0], tmp[0,1], tmp[0,2]])
test_user_result
```

Out[36]:

```
array([ 3.78394162e-04,  4.35827216e-06, -2.92221682e-18])
```

In [37]:

```python
# Вычисляем косинусную близость между текущим дегустатором
# и остальными дегустаторами
cos_sim = cosine_similarity(Vr, test_user_result.reshape(1, -1))
cos_sim[:10]
```

Out[37]:
```
array([[ 9.99999728e-01],
       [-1.53344496e-18],
       [ 6.94212130e-35],
       [-9.35884452e-33],
       [-4.12491330e-04],
       [ 9.99999975e-01],
       [-1.45343196e-36],
       [-1.04994959e-03],
       [ 0.00000000e+00],
       [ 0.00000000e+00]])
```

In [38]:
```
# Преобразуем размерность массива
cos_sim_list = cos_sim.reshape(-1, cos_sim.shape[0])[0]
cos_sim_list[:10]
```

Out[38]:
```
array([ 9.99999728e-01, -1.53344496e-18,  6.94212130e-35, -9.35884452e-33,
       -4.12491330e-04, 9.99999975e-01, -1.45343196e-36, -1.04994959e-03,
        0.00000000e+00,  0.00000000e+00])
```

In [39]:
```
# Находим наиболее близкого дегустатора
recommended_user_id = np.argsort(-cos_sim_list)[0]
recommended_user_id
```

Out[39]:
```
5
```

In [40]:
```
test_user
```

Out[40]:
```
matrix([[ 0.,  0.,  0., ..., 87.,  0.,  0.]])
```

In [41]:
```
# Получение названия вина
wine_list = list(user_item_matrix.columns)
def film_name_by_movieid(ind):
    try:
        wine = wine_list[ind]
        #print(wineId)
        #flt_links = data3[data['movieId'] == wineId]
        #tmdbId = int(flt_links['tmdbId'].values[0])
        #md_links = df_md[df_md['id'] == tmdbId]
        #res = md_links['title'].values[0]
        return wine
    except:
        return ''
```

In [42]:
```
# Вина, которые оценивал текущий дегустатор:
i=1
for idx, item in enumerate(np.ndarray.flatten(np.array(test_user))):
    if item > 0:
        film_title = film_name_by_movieid(idx)
        print('{} - {} - {}'.format(idx, film_title, item))
        if i==20:
            break
        else:
            i+=1
```

```
5 - Citari 2012 Sorgente  (Lugana) - 85.0
21 - Sesta di Sopra 2011  Brunello di Montalcino - 90.0
24 - Florio 2010  Passito di Pantelleria - 90.0
67 - Tasca d'Almerita 2012 Regaleali Nero d'Avola (Sicilia) - 88.0
74 - San Felice 2013  Chianti Classico - 88.0
75 - Lornano 2012  Chianti Classico - 86.0
92 - Feudi di San Gregorio NV Dubl Brut Falanghina (Campania) - 90.0
101 - Michele Chiarlo 2011 Cerequio  (Barolo) - 94.0
109 - Masottina 2014 Rive di Ogliano Contrada Granda Brut  (Conegliano Valdobbiadene Prosecco Superiore) - 88.0
114 - La Lastra 2012 Riserva  (Vernaccia di San Gimignano) - 87.0
119 - La Mozza 2013 I Perazzi  (Morellino di Scansano) - 87.0
141 - Marchesi de' Frescobaldi 2015 Bianco Benefizio Riserva Chardonnay (Pomino) - 90.0
154 - Castello di Meleto 2013  Chianti Classico - 87.0
182 - Cantine del Notaio 2012 La Firma  (Aglianico del Beneventano) - 93.0
208 - La Farra 2014 Rive di Farro di Soligo Extra Dry  (Valdobbiadene Prosecco Superiore) - 89.0
216 - Cormòns 2013 Friulano (Collio) - 87.0
220 - Pietroso 2010  Brunello di Montalcino - 88.0
245 - Conterno Fantino 2011 Sorì Ginestra  (Barolo) - 92.0
268 - CarlindePaolo 2015  Moscato d'Asti - 87.0
271 - Contucci 2009  Vino Nobile di Montepulciano - 94.0
```

In [43]:

```
# Вина, которые оценивал наиболее схожий дегустатор:
i=1
recommended_user_item_matrix = user_item_matrix.loc[['Roger Voss']]
for idx, item in enumerate(np.ndarray.flatten(np.array(recommended_user_item_matrix))):
    if item > 0:
        film_title = film_name_by_movieid(idx)
        print('{} - {} - {}'.format(idx, film_title, item))
        if i==20:
            break
        else:
            i+=1
```

```
4 - Cave du Château des Loges 2015 Prestige  (Beaujolais-Villages) - 85.0
12 - Château Moncontour 2014 Sec  (Vouvray) - 88.0
14 - Hugel 2005 Vendange Tardive Gewurztraminer (Alsace) - 90.0
16 - Duval-Leroy NV Brut Rosé  (Champagne) - 91.0
17 - Château Haut Prieur 2012  Blaye Côtes de Bordeaux - 83.0
19 - Wines & Winemakers 2008 Aguia Moura Em Vinhas Velhas Reserva Red (Douro) - 92.0
22 - Fischer 2006 Klassik Fasangarten Zweigelt (Thermenregion) - 88.0
28 - Domaine Lathuilière Gravallon 2015 Corcelette  (Morgon) - 92.0
29 - Domaine François Schmitt 2011 Bollenberg Sylvaner (Alsace) - 86.0
30 - Les Maîtres Vignerons de la Presqu'île de Saint-Tropez 2014 Domaine Aureillan Rosé (Côtes de Provence) - 86.
0
31 - Château des Antonins 2014  Bordeaux Blanc - 84.0
34 - Ruhlmann 2011 Cuvée Jean-Charles Riesling (Alsace) - 84.0
39 - Château Lamothe 2015  Bordeaux - 87.0
40 - Château de Cénac 2007 Eulalie Malbec (Cahors) - 88.0
58 - Domaine des Comtes Lafon 2007 Clos des Chênes Premier Cru  (Volnay) - 93.0
65 - Domaine du Haut Bourg 2016 Sur Lie  (Muscadet Côtes de Grandlieu) - 87.0
72 - Jaffelin 2010 Les Grandes Vignes Premier Cru  (Givry) - 88.0
76 - Château Suau 2010  Côtes de Bordeaux - 90.0
91 - Quinta do Passadouro NV Ruby Reserva  (Port) - 86.0
94 - Domaine Cauhapé 2010 Noblesse du Temps  (Jurançon) - 93.0
```

Как видно, фильтрация на основе содержания и коллаборативная фильтрация показывают различные результаты работы в рамках рекомендательных систем