

# Лабораторная работа №4: "Создание рекомендательной модели."

## ИУ5-23 Зорин Арсений

### Задание:

- Выбрать произвольный набор данных (датасет), предназначенный для построения рекомендательных моделей;
- Опираясь на материалы лекции, сформировать рекомендации для одного пользователя (объекта) двумя произвольными способами;
- Сравнить полученные рекомендации (если это возможно, то с применением метрик).

```
In [33]: import numpy as np
import pandas as pd
from typing import Dict, Tuple
from scipy import stats
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from sklearn.metrics import confusion_matrix
from sklearn.metrics.pairwise import cosine_similarity, euclidean_distances, manhattan_distances
from surprise import SVD, Dataset, Reader
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib_venn import venn2

data = pd.read_csv('../datasets/laptop/laptop_price.csv')
data = data.set_index('laptop_ID')
data.head()
```

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
laptop_ID												
1	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	1339.69
2	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	898.94
3	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	575.00
4	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	2537.45
5	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	1803.60

```
In [34]: company = data['Company'].values
company[0:5]
```

```
Out[34]: array(['Apple', 'Apple', 'HP', 'Apple', 'Apple'], dtype=object)
```

```
In [35]: product = data['Product'].values
product[0:5]
```

```
Out[35]: array(['MacBook Pro', 'Macbook Air', '250 G6', 'MacBook Pro',
'MacBook Pro'], dtype=object)
```

```
In [36]: cpu = data['Cpu'].values
cpu[0:5]
```

```
Out[36]: array(['Intel Core i5 2.3GHz', 'Intel Core i5 1.8GHz',
'Intel Core i5 7200U 2.5GHz', 'Intel Core i7 2.7GHz',
'Intel Core i5 3.1GHz'], dtype=object)
```

```
In [37]: gpu = data['Gpu'].values
gpu[0:5]
```

```
Out[37]: array(['Intel Iris Plus Graphics 640', 'Intel HD Graphics 6000',
'Intel HD Graphics 620', 'AMD Radeon Pro 455',
'Intel Iris Plus Graphics 650'], dtype=object)
```

```
In [38]: class SimpleKNNRecommender:

    def __init__(self, X_matrix, X_company, X_product, X_cpu, X_gpu):
        """
        Входные параметры:
        X_matrix - обучающая выборка (матрица объект-признак)
        """
        #Сохраняем параметры в переменных объекта
        self.X_matrix = X_matrix
        self.df = pd.DataFrame(
            {'company': pd.Series(X_company, dtype='str'),
            'product': pd.Series(X_product, dtype='str'),
            'cpu': pd.Series(X_cpu, dtype='str'),
            'gpu': pd.Series(X_gpu, 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 [39]: %%time
tfidf = TfidfVectorizer()
matrix = tfidf.fit_transform(product)
matrix

CPU times: user 7.39 ms, sys: 935 µs, total: 8.33 ms
Wall time: 7.46 ms

Out[39]: <1303x705 sparse matrix of type '<class 'numpy.float64'>'
with 3641 stored elements in Compressed Sparse Row format>

In [52]: data.iloc[0]
```

Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	1339.69

```
In [42]: skrl = SimpleKNNRecommender(matrix, company, product, cpu, gpu)
```

```
In [44]: # 5 похожих моделей
rec1 = skrl.recommend_for_single_object(5, matrix[0])
rec1
```

	company	product	cpu	gpu	dist
1193	Apple	MacBook 12"	Intel Core M 1.1GHz	Intel HD Graphics 5300	484738.745793
794	Apple	MacBook 12"	Intel Core M 1.2GHz	Intel HD Graphics 5300	484738.745793
14	Apple	MacBook 12"	Intel Core M m3 1.2GHz	Intel HD Graphics 615	484738.745793
1069	Apple	MacBook 12"	Intel Core M 1.1GHz	Intel HD Graphics 515	484738.745793
1210	Apple	MacBook 12"	Intel Core M 1.2GHz	Intel HD Graphics 515	484738.745793

```
In [46]: # При поиске с помощью Евклидова расстояния
rec2 = skrl.recommend_for_single_object(5, matrix[0], cos_flag = False)
rec2
```

	company	product	cpu	gpu	dist
1069	Apple	MacBook 12"	Intel Core M 1.1GHz	Intel HD Graphics 515	1.015147e+06
794	Apple	MacBook 12"	Intel Core M 1.2GHz	Intel HD Graphics 5300	1.015147e+06
1210	Apple	MacBook 12"	Intel Core M 1.2GHz	Intel HD Graphics 515	1.015147e+06
1193	Apple	MacBook 12"	Intel Core M 1.1GHz	Intel HD Graphics 5300	1.015147e+06
81	Apple	MacBook 12"	Intel Core i5 1.3GHz	Intel HD Graphics 615	1.015147e+06

```
In [48]: # Манхэттэнское расстояние
rec3 = skrl.recommend_for_single_object(5, matrix[0], cos_flag = False, manh_flag = True)
rec3
```

	company	product	cpu	gpu	dist
635	Asus	VivoBook Pro	Intel Core i5 7300HQ 2.5GHz	Nvidia GeForce GTX 1050	1.514496e+06
241	Asus	VivoBook Pro	Intel Core i7 8550U 1.8GHz	Nvidia GeForce 150MX	1.514496e+06
492	Asus	VivoBook Pro	Intel Core i7 7700HQ 2.8GHz	Nvidia GeForce GTX 1050	1.514496e+06
65	Asus	ZenBook Pro	Intel Core i7 7700HQ 2.8GHz	Nvidia GeForce GTX 1050 Ti	1.515744e+06
474	Asus	ZenBook Pro	Intel Core i7 6700HQ 2.6GHz	Intel HD Graphics 530	1.515744e+06