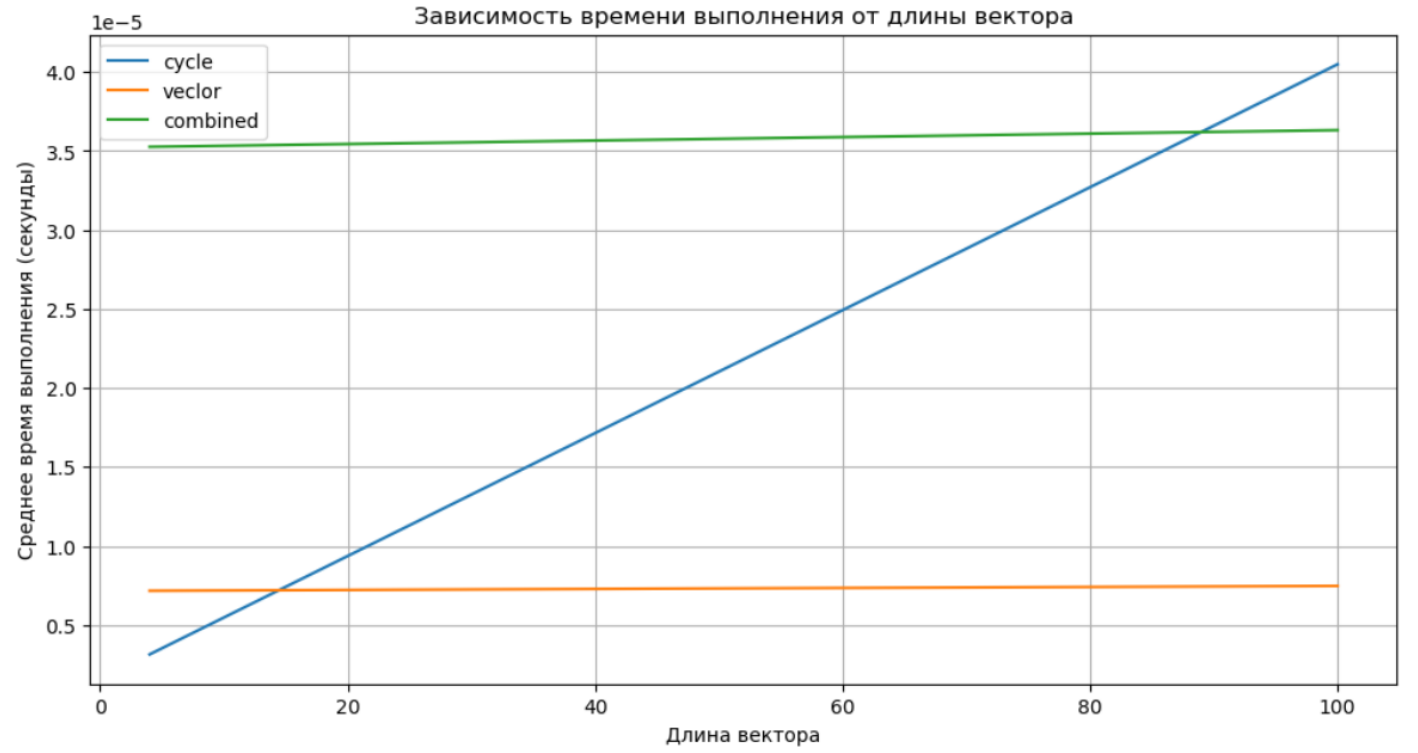
ОБЗОР НА ДОМАШНИЕ ЗАДАНИЯ

БАБЕНКО МАКСИМ ML-12

ДЗ-1: РЕШЕНИЕ ЗАДАЧ ТРЕМЯ МЕТОДАМИ И СРАВНЕНИЕ ИХ ЭФФЕКТИВНОСТИ



ВЫВОД

ПО ИТОГАМ ПРОДЕЛАННОЙ РАБОТЫ МОЖНО СДЕЛАТЬ ВЫВОД, ЧТО ВЕКТОРИЗИРОВАННЫЕ РЕШЕНИЯ С NUMPY ОБЫЧНО ЯВЛЯЮТСЯ НАИБОЛЕЕ ЭФФЕКТИВНЫМИ БЛАГОДАРЯ ОПТИМИЗИРОВАННЫМ ФУНКЦИЯМ ДЛЯ РАБОТЫ С МАССИВАМИ. НА ВСЕХ ТЕСТАХ ТАКИЕ РЕШЕНИЯ ИМЕЛИ МИНИМАЛЬНОЕ И ПРАКТИЧЕСКИ ОДИНАКОВОЕ ВРЕМЯ РАБОТЫ, ВНЕ ЗАВИСИМОСТИ ОТ РАЗМЕРА ПОДАВАЕМЫХ ДАННЫХ. ВЕКТОРИЗИРОВАННЫЕ РЕШЕНИЯ ПОЗВОЛЯЮТ ИЗБЕЖАТЬ ЯВНОГО НАПИСАНИЯ ЦИКЛОВ, ЧТО ПРИВОДИТ К УСКОРЕНИЮ ВЫЧИСЛЕНИЙ, А ТАК ЖЕ ЗНАЧИТЕЛЬНО УМЕНЬШАЮТ КОЛЛИЧЕСТВО КОДА. КОГДА РЕШЕНИЯ С ЦИКЛАМИ РАБОТАЛИ ОЧЕНЬ МЕДЛЕННО И ВЫГЛЯДЕЛИ ГРОМОЗДКО.
КОМБИНИРОВАННЫЕ РЕШЕНИЯ ПОКАЗАЛИ СЕБЯ ЛУЧШЕ ЦИКЛОВ, НО ЭТО СВЯЗАНО С ТЕМ, ЧТО ТАМ ЧАСТИЧНО ИСПОЛЬЗОВАЛИСЬ ВЕКТОРИЗИРОВАННЫЕ РЕШЕНИЯ С NUMPY.

НОВЫЕ ЗНАНИЯ

ДЛЯ МЕНЯ СТАЛО ОТКРЫТИЕМ, ЧТО МАТРИЧНЫЕ ОПЕРАЦИИ В NUMPY НАСКОЛЬКО БЫСТРЫЕ, ТАК ЧТО ТЕПЕРЬ БУДУ СТАРАТЬСЯ ИСПОЛЬЗОВАТЬ NUMPY, ГДЕ ЭТО БУДЕТ ВОЗМОЖНО

ДЗ-2: ПРЕДСКАЗАНИЕ ПОГОДЫ С ПОМОЩЬЮ ПРОСТЫХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import  OneHotEncoder
```

Метод	Accuracy	F1 score
LogReg (самостоятельно)	0.83766	0.53739
LogReg (библиотечная)	0.84984	0.55775
GaussianNB	0.76044	0.56833
KNeighborsClassifier	0.80883	0.38341

ВЫВОД

ПО КАЧЕСТВУ:

ЛУЧШЕ ВСЕГО ПОКАЗАЛА СЕБЯ МОДЕЛЬ SKLEARN.LINEAR_MODEL.LOGISTICREGRESSION, ЧУТЬ ХУЖЕ ОКАЗАЛАСЬ ЛОГИСТИЧЕСКАЯ РЕГРЕССИИЯ РЕАЛИЗОВАННАЯ САМОСТОЯТЕЛЬНО, ДАЛЕЕ ПОКАЧЕСТВУ ИДЕТ SKLEARN.NEIGHBORS И ХУЖЕ ВСЕГО ОТРАБОТАЛА МОДЕЛЬ SKLEARN.NAIVE_BAYES

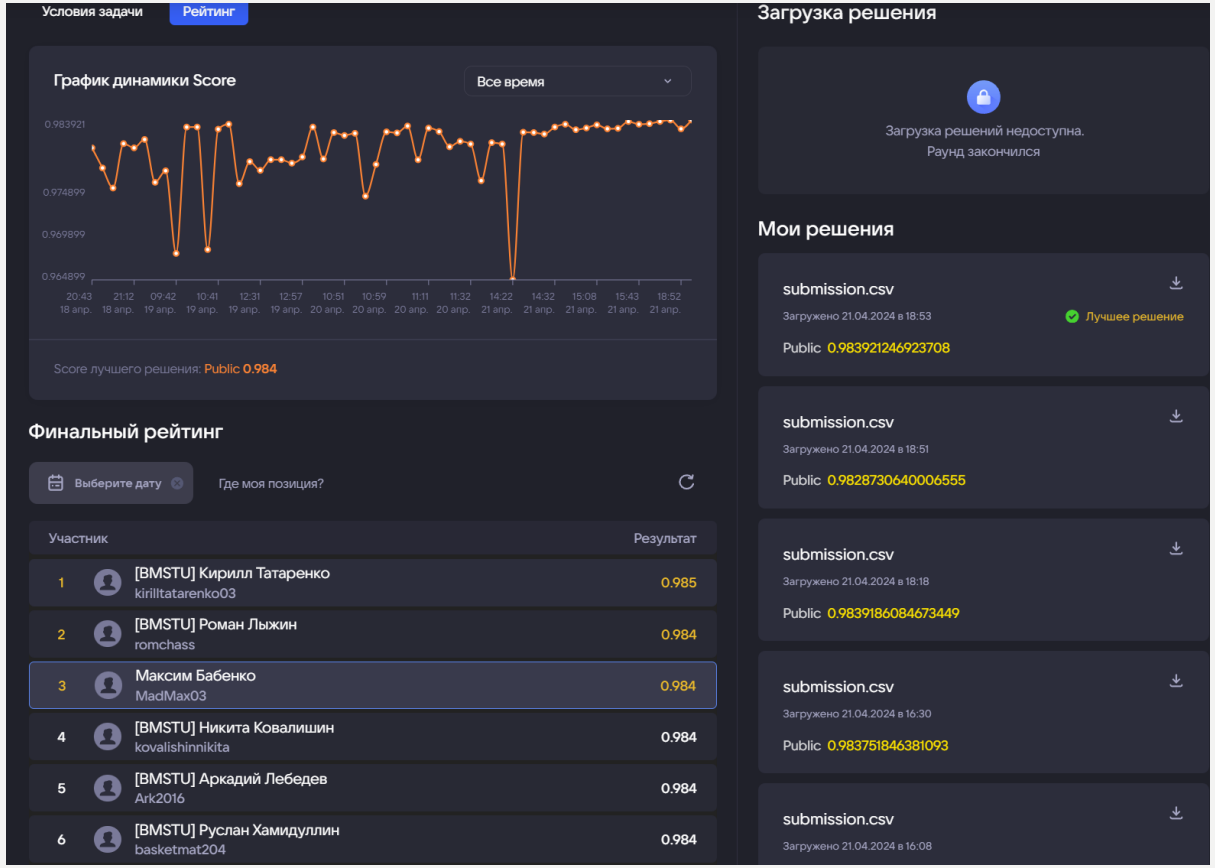
ПО ВРЕМЕНИ РАБОТЫ:

ПРАКТИЧЕСКИ МОМЕНТАЛЬНО ОТРАБОТАЛА SKLEARN.LINEAR_MODEL.LOGISTICREGRESSION, ЧУТЬ ХУЖЕ ОКАЗАЛАСЬ SKLEARN.NAIVE_BAYES, СИЛЬНО ДОЛЬШЕ РАБОТАЛА ЛОГИСТИЧЕСКАЯ РЕГРЕССИИЯ РЕАЛИЗОВАННАЯ САМОСТОЯТЕЛЬНО И ДОЛЬШЕ ВСЕХ РАБОТАЛА МОДЕЛЬ SKLEARN.NEIGHBORS

НОВЫЕ ЗНАНИЯ

В ПРИНЦИПЕ, СО ВСЕМИ АЛГОРИТМАМИ ИЗ ДАННОГО ЗАДАНИЯ Я БЫЛ РАНЕЕ ЗНАКОМ, ТАК ЧТО, ОСВЕЖИЛ ИХ В ПАМЯТИ

ДЗ-3: ДЕТЕКЦИЯ ПОРНОГРАФИИ




```
preprocessor = ColumnTransformer(
    transformers=[
        ('url', TfidfVectorizer(
            analyzer="char_wb", ngram_range=(2, 4),
            min_df=3, max_df=0.75), 'url'),
        ('title', CountVectorizer(
            stop_words=stop_words, min_df=3,
            max_df=0.75), 'title')
    ])
```

```
model = LogisticRegression(
    C=1.5, solver='lbfgs',
    class_weight='balanced', max_iter=1500)

model.fit(X_train_preprocessed, y_train)
y_pred = model.predict(test_preprocessed)
```

ВЫВОД

ПО ИТОГАМ МНОГИХ ЭКСПЕРИМЕНТОВ ВЫЯСНИЛОСЬ, ЧТО ИЗНАЧАЛЬНО ЛУЧШЕ ВСЕГО ПРОСТО ИЗБАВИТЬСЯ ОТ ЛИШНИХ СИМВОЛОВ И ПРИВЕСТИ ВСЕ К НИЖНЕМУ РЕГИСТРУ. УДАЛЕНИЕ СТОП СЛОВ И ЛЕММАТИЗАЦИЯ ТОЛЬКО УХУДШАЛИ РЕЗУЛЬТАТ, А URL ЛУЧШЕ ВООБЩЕ НЕ ТРОГАТЬ

В КАЧЕСТВЕ ПРЕДСКАЗАТЕЛЬНОГО АЛГОРИТМА БЫЛА ВЫБРАНА ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ С ЛУЧШИМИ ПОДОБРАННЫМИ ГИПЕРПАРАМЕТРАМИ (БЫЛИ ПРОТЕСТИРОВАНЫ И ДРУГИЕ АЛОГОРИТМЫ, НО ОНИ ПОКАЗАЛИ СЕБЯ ХУЖЕ)

РЕАЛИЗОВАЛ ГИПОТЕЗУ ОБ ОТНЕСЕНИИ САЙТА К ПОЛОЖИТЕЛЬНОМУ КЛАССУ, ПРИ ВХОЖДЕНИИ В ЕГО TITLE И URL 'ПЛОХИХ СЛОВ' (ПОЛУЧИЛСЯ НЕБОЛЬШОЙ ПРИРОСТ ТОЧНОСТИ)

НОВЫЕ ЗНАНИЯ

ВПЕРВЫЕ ТАК ПЛОТНО НА ПРАКТИКЕ ПОЗНАКОМИЛСЯ С МЕТОДАМИ РАБОТЫ С ТЕКСТОМ: УДАЛЕНИЕ ЛИШНИХ СИМВОЛОВ, СТОП СЛОВ, ЛЕМАТИЗАЦИЯ И ВЕКТОРИЗАЦИЯ (**TFIDFVECTORIZER**, **COUNTVECTORIZER**), А ТАК ЖЕ С ПОДХОДОМ К РЕШЕНИЮ ЗАДАЧИ КЛАССИФИКАЦИИ ТЕКСТА

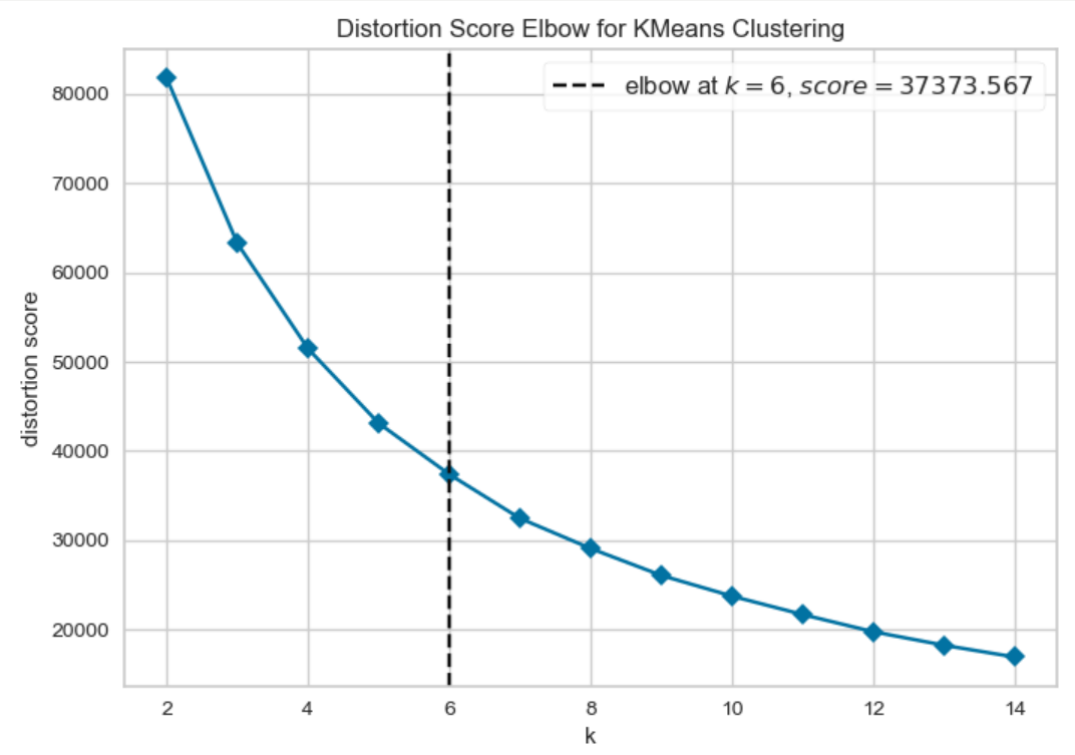
ДЗ-4: КЛАСТЕРИЗАЦИЯ

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import sparse
from tqdm import tqdm

from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering
from sklearn.cluster import DBSCAN
from sklearn.cluster import MiniBatchKMeans
import hdbscan
from sklearn.mixture import GaussianMixture
from sklearn.cluster import SpectralClustering
from yellowbrick.cluster import KElbowVisualizer
from sklearn.metrics import silhouette_score

from sklearn.decomposition import TruncatedSVD
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
import umap

from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MaxAbsScaler
from sklearn.preprocessing import RobustScaler
```

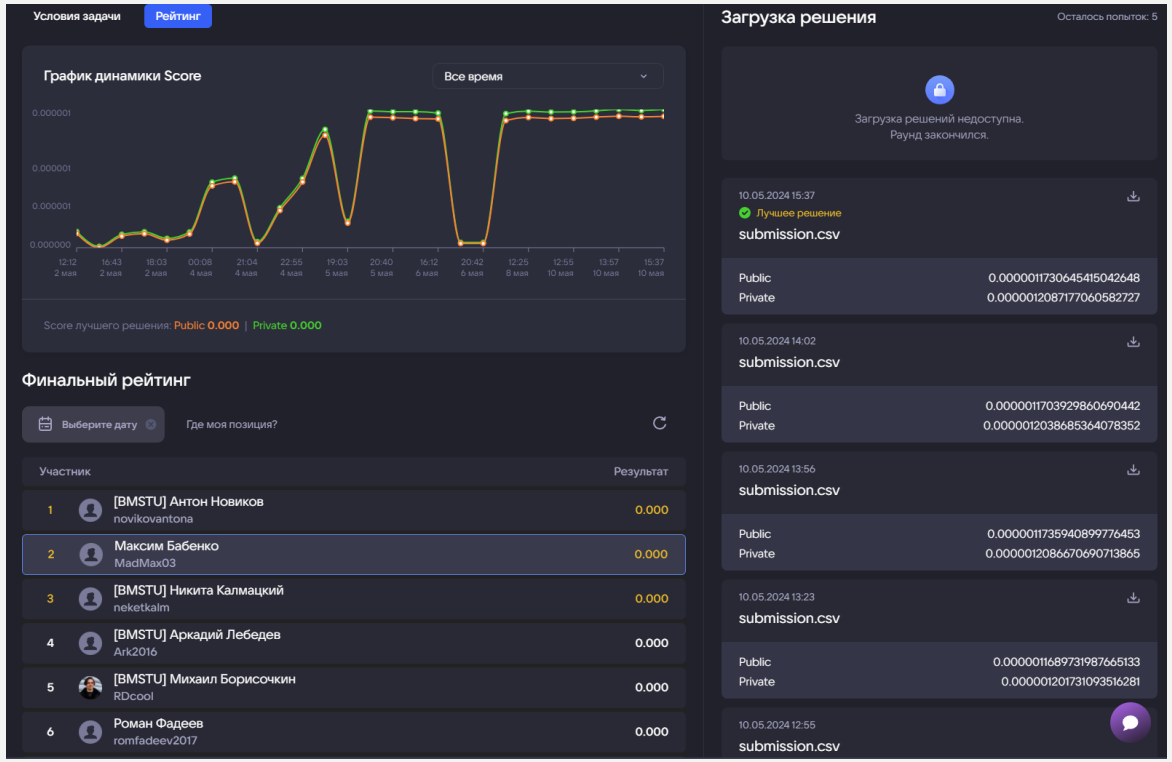


ВЫВОД И НОВЫЕ ЗНАНИЯ

ВПЕРВЫЕ СТОЛКНУЛСЯ С ТАКОЙ ЗАДАЧЕЙ КЛАСТЕРИЗАЦИИ, БЫЛО ИНТЕРЕСНО И ПОЛЕЗНО. В ХОДЕ РЕШЕНИЯ ЗАДАЧИ Я ПРОТЕСТИРОВАЛ МНОЖЕСТВО НОВЫХ ДЛЯ МЕНЯ МЕТОДОВ.

- ДЛЯ НОРМАЛИЗАЦИИ ДАННЫХ: **STANDARDSCALER, MAXABSSCALER, ROBUSTSCALER**
- ДЛЯ ПОНИЖЕНИЯ РАЗМЕРНОСТИ: **TRUNCATEDSVD, PCA, TSNE, UMAP**
- ДЛЯ КЛАСТЕРИЗАЦИИ: **KMEANS, DBSCAN, MINIBATCHKMEANS, GAUSSIANMIXTURE, AGGLOMERATIVECLUSTERING, SPECTRALCLUSTERING**
- ДЛЯ ОПРЕДЕЛЕНИЯ КОЛИЧЕСТВА КЛАСТЕРОВ: МЕТОД ЛОКТЯ, МЕТОД СИЛУЭТА

ДЗ-5: ПРЕДСКАЗАНИЕ ЦЕН НА НЕДВИЖИМОСТЬ




```
best_model =
CatBoostRegressor(iterations=10000,
                  loss_function='MAE',
                  eval_metric='MAE',
                  logging_level='Silent',
                  depth=6,
                  learning_rate=0.05,
                  cat_features=categorical_features,
                  early_stopping_rounds=2000)
best_model.fit(X_train, y_train)
```

ВЫВОД И НОВЫЕ ЗНАНИЯ

УЖЕ НЕ РАЗ СТАЛКИВАЛСЯ С ПОДОБНЫМИ ЗАДАЧАМИ, ТАК ЧТО, В ЭТОТ РАЗ ПРОСТО ОСВЕЖИЛ ЗНАНИЯ В **FEATURE ENGINEERING**, ПОДБОРЕ ГИПЕРПАРАМЕТРОВ И ПРАВИЛЬНОЙ НАСТРОЙКЕ ГРАДИЕНТНОГО БУСТИНГА, КОТОРЫЙ ЛУЧШЕ ВСЕГО СПРАВИЛСЯ С ДАННОЙ ЗАДАЧЕЙ

ДЗ-6: РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ

```
For item:
052165615X
Recommendations:
['0521795028']
```

```
Топ рекомендованных книг: ['1854582747', '0395645662', '0452274427'] (ISBN)
```

```
Реккомендации для пользователя: ['3250100692', '3257060580', '3257062303', '3257205155', '325721837
0', '3257224176', '3257230885', '3100922484', '3100970616', '3123512304']
```

```
User-Based Recommendations for User 276726 : [' 9022906116', '0000000000', '00000000000', '0000001042
283', '0000018030', '000104799X', '0001048759', '000105337X', '0001053736', '0001053744']
```

```
Item-Based Recommendations for User 276726 : [' 9022906116', '0743448162', '0743448642', '074344867
7', '0743448782', '0743448944', '0743449002', '0743449150', '0743450884', '0743451414']
```

ВЫВОД И НОВЫЕ ЗНАНИЯ

С ЗАДАЧЕЙ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ Я ВСТРЕТИЛСЯ ВПЕРВЫЕ, ПОЭТОМУ ВСЕ ИСПОЛЬЗОВАННЫЕ МЕТОДЫ БИЛИ ДЛЯ МЕНЯ В НОВИНКУ. Я ИЗУЧИЛ И РЕАЛИЗОВАЛ:

- ПЕРСОНАЛЬНЫЙ ТОП ПО ПОХОЖИМ ПОЛЬЗОВАТЕЛЯМ
- РЕКОМЕНДАЦИИ НА ОСНОВЕ МЕТОДА КЛАСТЕРИЗАЦИИ ПОХОЖИХ ПОЛЬЗОВАТЕЛЕЙ
- РЕКОМЕНДАЦИИ НА ОСНОВЕ СОВСТРЕЧАЕМОСТИ
- **USER-BASED** РЕКОМЕНДАЦИИ
- **ITEM-BASED** РЕКОМЕНДАЦИИ

ОБЩИЙ ВЫВОД

По итогам курса я получил много новых и полезных знаний в области ML и не только, научился решать разные типы задач новыми и интересными для меня методами. В общем, сильно расширил кругозор в Машинном обучении, а полученные знания станут ценным активом для моей дальнейшей карьеры!