Recommendation System Implementation

Abduhakimov Fayzullo 201-23 KTL
Batirov Jumaniyoz 201-23 KTL
Jo'rayev Xudoyshukur 201-23 KTL

https://github.com/Khudoyshukur/RecommerSystemTUIT

We are proposing a recommendation system for e-commerce platforms. The recommendations will be based on collaborative filtering (user rating based). For this custom datasource has been created as follows:

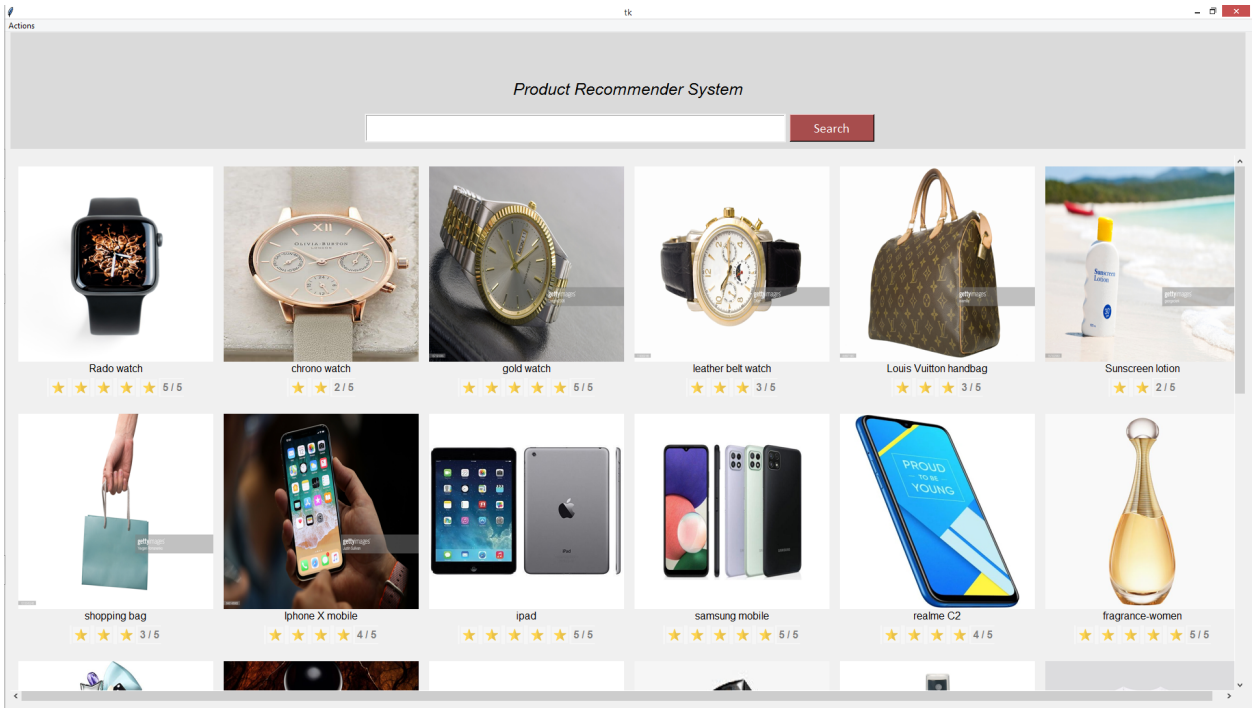
product_id	name	cetagory	images	average_rating
321732944	Rado watch	watch	Rado watch.jpg	5
439886341	chrono wacth	watch	chrono watch.jpg	2
511189877	gold watch	watch	gold watch.jpg	5
528881469	leather belt watch	watch	leather belt wacth.jpg	3
558835155	Louis Vuitton handbag	bad	Louis Vuitton handbag.jpg	3
594012015	Sunscreen lotion	lotion	Sunscreen lotion.jpg	2
594017343	tropical lotion	lotion	tropical lotion.jpg	1
594017580	shopping bag	bag	shopping bad.jpg	3
594033896	Iphone X mobile	mobile	iphone x mobile.jpg	4
594033926	ipad	mobile	ipad.jpg	5
594033934	samsung mobile	mobile	samsung mobile.jpeg	5
594202442	realme C2	mobile	realme C2.jpg	4
594287995	fragrance-women	perfume	fragrance-women.jpg	5
594296420	daisy perfume	perfume	daisy perfume.jpg	5
594450209	marc-jacobs-perfect	perfume	marc-jacobs-perfect.jpg	5
594450705	tofrod perfume	perfume	tofrod perfume.jpg	5
594451647	mouse-keyboard	electronics	mouse-keyboard.jpg	4
594477670	speaker system	electronics	speaker system.jpg	5
594478162	olive oil 1	lubricants	olive oil 1.jpg	4
594481813	T-shirt	cloth	T-shirt.jpg	4
594481902	Navy blue T shirt	cloth	Navy-blue T shirt.jpg	4
594482127	local shirt	cloth	local shirt.jpg	4
594511488	Girls T shirt	cloth	Girls T shirt.jpg	5
594514681	Koren T shirt	cloth	Koren T shirt.jpeg	5
594514789	Dell XPS 15 laptop	laptop	Dell XPS 15 laptop.jpg	5
594549507	hp pavilion gaming	laptop	hp pavilion gaming.jpg	4
594549558	battery 1	electronics	battery 1.jpg	5
743610431	dell precision battery	electronics	dell precision battery.jpg	4
777700018	mac book pro	laptop	mac book pro.jpg	5
840017677	jeans	cloth	jeans.jpg	4

Products datasource

user_id	product_id	rating
A2CX7LUOH	321732944	5
A2NWSAGRI	439886341	1
A2WNBOD3	439886341	3
A1GI0U4ZRJ	439886341	1
A1QGNMC6	511189877	5
A3J3BRHTDF	511189877	2
A2TY0BTJOT	511189877	5
A34ATBPOK	511189877	5
A89DO69P0	511189877	5
AZYNQZ94U	511189877	5
A1DA3W4G	528881469	5
A29LPQQDG	528881469	1
AO94DHGC7	528881469	5
AMO214LNF	528881469	1
A28B1G1MS	528881469	4
A3N7T0DY8	528881469	3
A1H8PY3QH	528881469	2
A2CPBQ5W	528881469	2

Partial datasource of product ratings

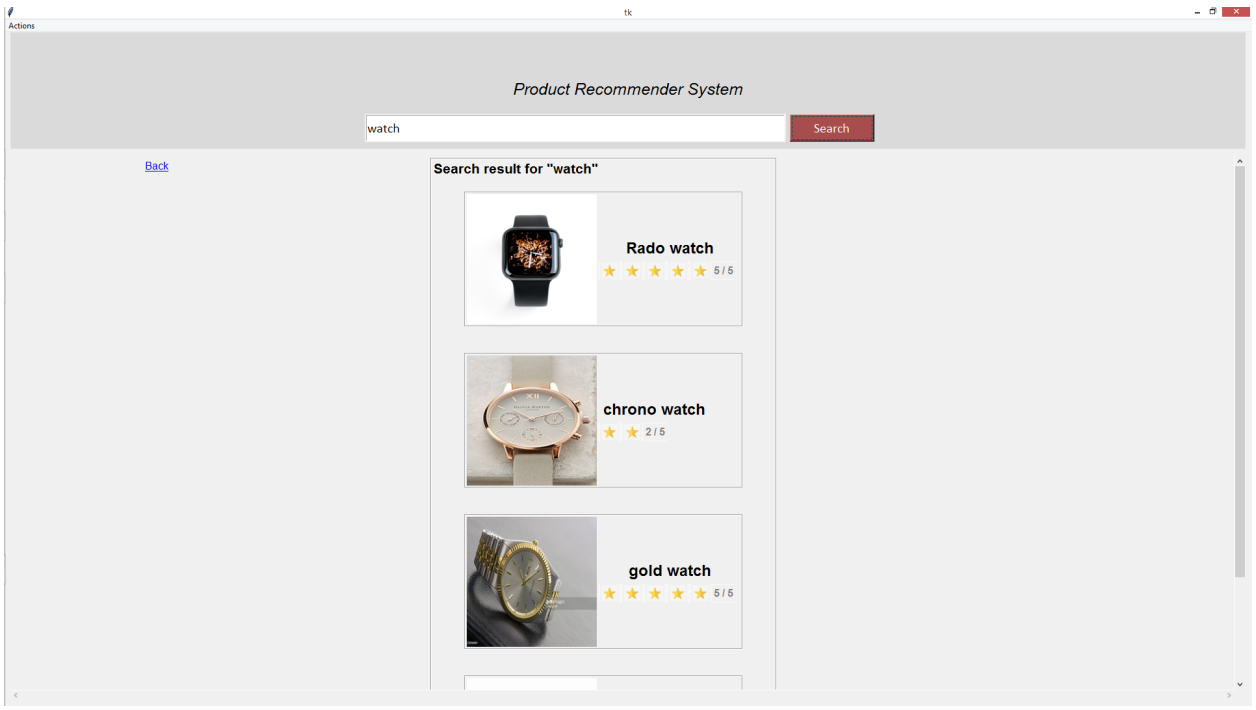
In the UI part we are showing the products as follows:



The code below represents the products above:

```
import time
import tkinter as tk
from Frames import panels
from tkinter.constants import *
from PIL import Image, ImageTk
from data import prodDisplay, starfilled
from doubleScroll import DoubleScrolledFrame
from productpage import ProductPage, RecommendationPanel
from recommendation import recommendKNN
from searchresult import SearchResult

2 usages
class Catalogue(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.__config()
        self.render()
        self.pack(side=LEFT, pady=10, fill=BOTH, expand=1, padx=10)

    1 usage
    def __config(self):...

    3 usages (3 dynamic)
    def hidemain(self):...

    2 usages (2 dynamic)
    def showmain(self):...

    7 usages (6 dynamic)
    def render(self):...

1 usage
class Product(tk.Frame):...
```

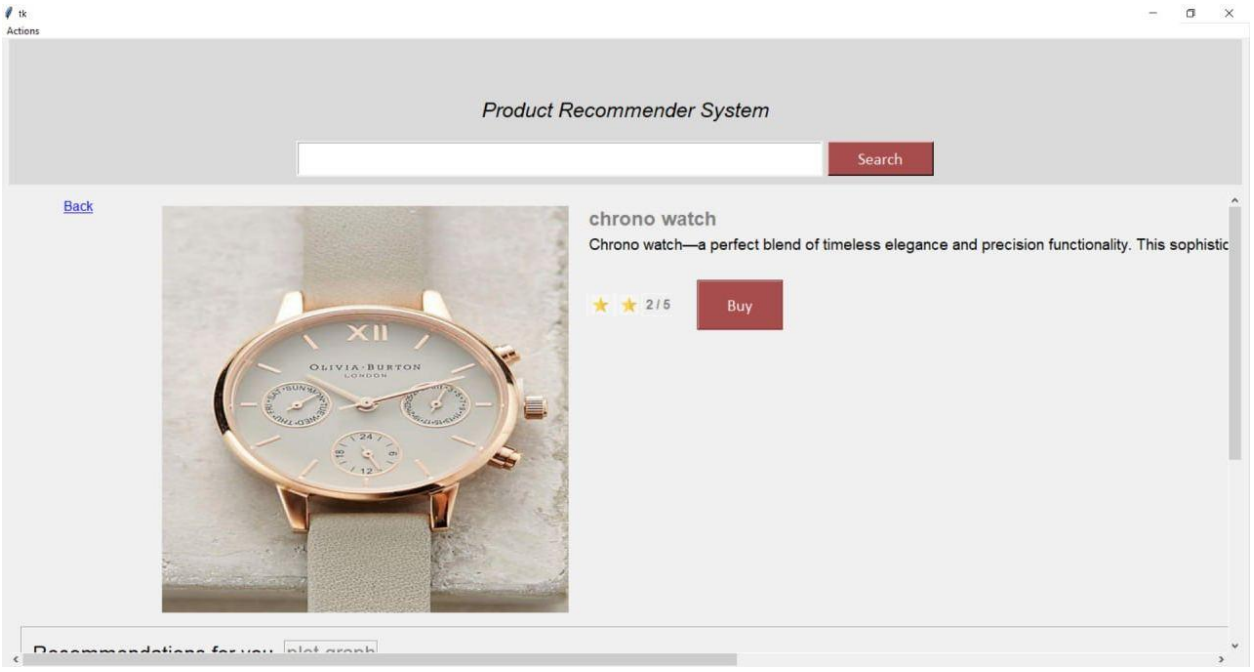
When the user enters a query in the search bar, we show a search recommendations.



The code below represents this search results page:

```
2 usages
class SearchResult(tk.Frame):
    def __init__(self, master=None):...

    1 usage (1 dynamic)
    def setSearchResult(self, res):...

    1 usage (1 dynamic)
    def setSearchItem(self, item):...

    1 usage
    def __config(self): pass

    5 usages (4 dynamic)
    def hide(self):...

    5 usages (5 dynamic)
    def show(self):...

    5 usages (5 dynamic)
    def clear(self):...

    def __renderRating(self, master=None, rating=5):...


    6 usages (6 dynamic)
    def render(self):...
```

When one product is clicked, the Product page is opened. Also, similar products are shown.



The code below represents this screen:

```
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

import time
from Frames import panels

from PIL import Image, ImageTk

2 usages
class ProductPage(tk.Frame):
    def __init__(self, master=None, name=str, price=0, desc="", ratiing=int):...

    4 usages (4 dynamic)
    def setProps(self, name, img, rating, gdata):...

    5 usages (5 dynamic)
    def show(self):...

    5 usages (4 dynamic)
    def hide(self):...

    5 usages (5 dynamic)
    def clear(self):...

    1 usage
    def __config(self):...

    1 usage
    def __renderRating(self, master=None):...

    6 usages (6 dynamic)
    def render(self):...

1 usage
def get_value_from_dict(key):...


2 usages
class RecommendationPanel(tk.Frame):...
```

The recommendations will be based on collaborative filtering:

```
4 usages
def recommendKNN(_id):
    df_data = pd.DataFrame(ratings)
    features = df_data.pivot(index="product_id", columns="user_id", values="rating").fillna(0)
    n = list(features.index).index(_id)
    cm = csr_matrix(features.values)

    d = NearestNeighbors(metric="cosine", algorithm="brute")
    d.fit(cm)
    dis, ind = d.kneighbors(features.iloc[n, :].values.reshape(1, -1), n_neighbors=7)
    _products = []
    prodName = []
    distance = []

    for i in range(1, len(dis.flatten())):
        pro = list(filter(lambda pr: pr["product_id"] == features.index[ind.flatten()[i]], products))[0]['name']
        prodName.append(pro)
        _products.append(features.index[ind.flatten()[i]])
        distance.append(dis.flatten()[i])

    m = pd.Series(_products, name='product')
    d = pd.Series(distance, name='distance')

    pr = pd.Series(prodName, name="productName")
    di = pd.Series(distance, name="p_distance")
    prdi = pd.concat( objs: [pr, di], axis=1)
    prdi = prdi.sort_values( by: "p_distance", ascending=False)

    recommend = pd.concat( objs: [m, d], axis=1)
    recommend = recommend.sort_values( by: 'distance', ascending=False)
    recommendedP = []

    for i in range(0, recommend.shape[0]):
        pro = list(filter(lambda pr: pr["product_id"] == recommend["product"].iloc[i], products))[0]
        recommendedP.append(pro)

    return recommendedP, (prdi)
```