

به نام خدا

موضوع پروژه :

سیستم توصیه گر هوشمند

نام استاد : دکتر عصایی

نام: محمد

نام خانوادگی : علی قنبری

شماره دانشجویی : 40116341054704

گروه دوشنبه 15:30 – 18:15

سکشن ثبت نام : دوشنبه 15 : 13 – 15 : 30

مقدمه :

یک سیستم توصیه‌گر هوشمند داریم که به کاربران کمک می‌کند محصولات، فیلم‌ها یا مطالب آموزشی مناسب را بر اساس سلیقه و رفتار خود پیدا کنند. این پروژه به طراحی و پیاده‌سازی یک سیستم توصیه‌گر هوشمند می‌پردازد که از تکنیک‌های یادگیری ماشین و فیلترینگ استفاده می‌کند.

سناریو :

یک سیستم توصیه‌گر برای یک پلتفرم نمایش محصولات، فیلم و مطالب آموزشی داریم. هدف اصلی این سیستم ارائه پیشنهادات شخصی‌سازی شده براساس سلیقه و رفتار کاربران است. کاربران پس از ورود به وبسایت، می‌توانند محصولات را مشاهده و خریداری کنند. سیستم با استفاده از داده‌های قبلی کاربران (مانند تاریخچه خرید، امتیازدهی به محصولات و جستجوهای اخیر) و همچنین با تحلیل رفتار مشابه سایر کاربران، لیستی از محصولات، فیلم‌ها و مطالب آموزشی که ممکن است کاربر به آن‌ها علاقه‌مند یا به آن‌ها نیازمند باشد را مشاهده می‌کند.

مراحل پیاده‌سازی:

جمع‌آوری داده‌ها :

داده‌های مرتبط با کاربران: نظرات، امتیازات، تاریخچه خرید و مشاهده، سن، جنسیت، مکان جغرافیایی، تاریخچه تماشا
داده‌های مربوط به فیلم‌ها و سریال‌ها: ژانر، بازیگران، سال تولید، رتبه‌بندی کاربران، امتیاز فیلم‌ها و محصولات

پردازش داده‌ها:

پاکسازی داده‌ها و پیش‌پردازش: حذف داده‌های ناقص و تکراری، استاندارد سازی داده‌ها

ایجاد مدل‌های فیلترینگ مبتنی بر محتوا :

استفاده از ویژگی‌های فیلم‌ها (ژانر، بازیگران، کارگردان) برای محاسبه شباهت بین فیلم‌ها

استفاده از ویژگی محصولات برای محاسبه شباهت‌ها

ایجاد مدل‌های فیلترینگ مبتنی بر همکاری: (Collaborative Filtering)

استفاده از رتبه‌بندی‌ها، تاریخچه جستجو و تماشای کاربران برای پیش‌بینی علاقه‌مندی‌های آنها

آموزش مدل‌ها با استفاده از یادگیری ماشین:

استفاده از Python و کتابخانه‌های Scikit-Learn و TensorFlow برای پیاده‌سازی و آموزش مدل‌ها

تست و ارزیابی مدل‌ها:

تقسیم داده‌ها به داده‌های آموزش و تست

ارزیابی دقت پیش‌بینی‌ها با معیارهای مختلف مانند Recall, Precision, RMSE

پیاده‌سازی سیستم نهایی

ارائه پیشنهادات به کاربران در پلتفرم براساس مدل‌های یادگیری ماشین آموزش‌دیده

جدول حالت :

شماره	Percept	Action
1	کاربر جدید ثبت‌نام کرد	درخواست اطلاعات پایه‌ای از کاربر (سن، جنسیت، مکان)
2	کاربر وارد شد	نمایش لیستی از فیلم‌های محبوب برای کاربران جدید
3	کاربر فیلمی را تماشا کرد	ثبت اطلاعات تماشا در پایگاه داده
4	کاربر فیلمی را رتبه‌بندی کرد	به‌روزرسانی مدل‌های فیلترینگ مبتنی بر همکاری
5	کاربر ژانری را انتخاب کرد	نمایش فیلم‌ها و سریال‌های مرتبط با آن ژانر
6	کاربر جستجویی انجام داد	ثبت کلمات جستجو شده و به‌روزرسانی مدل‌های فیلترینگ مبتنی بر محتوا
7	کاربر نظر جدیدی درباره فیلمی نوشت	به‌روزرسانی ویژگی‌های فیلم و مدل‌های توصیه‌گر
8	فیلم یا سریال جدید به پلتفرم اضافه شد	محاسبه شباهت فیلم جدید با فیلم‌های موجود و به‌روزرسانی مدل‌های فیلترینگ
9	کاربر لیستی از فیلم‌های دلخواه ایجاد کرد	پیشنهاد فیلم‌ها بر اساس محتویات لیست دلخواه
10	کاربر از فیلمی صرف‌نظر کرد	ثبت اطلاعات و به‌روزرسانی مدل‌های فیلترینگ
11	کاربر زمان مشخصی را برای تماشا انتخاب کرد	نمایش فیلم‌ها و سریال‌های متناسب با زمان تماشا
12	کاربر پروفایل خود را به‌روزرسانی کرد	به‌روزرسانی اطلاعات کاربر و تأثیر آن بر مدل‌های فیلترینگ
13	کاربر فیلمی را به اشتراک گذاشت	ثبت اطلاعات اشتراک‌گذاری و تأثیر آن بر مدل‌های همکاری
14	کاربر از تماشای فیلمی منصرف شد	ثبت اطلاعات و به‌روزرسانی مدل‌های توصیه‌گر
15	کاربر پیامد رفتارهای گذشته‌اش را بررسی کرد	ارائه پیشنهادات براساس رفتارهای گذشته و پیش‌بینی‌های مدل

در پایین کدی ساده است برای پیاده سازی یک سیستم توصیه گر هوشمند ، که من به دلیل نداشتن دانش کافی در این زمینه آن را به کمک هوش مصنوعی نوشته ام :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.neighbors import NearestNeighbors
import matplotlib.pyplot as plt

# موجود است 'data' به نام DataFrame فرض کنید داده‌ها در یک
data = pd.read_csv('user_item_ratings.csv')

# پیش‌پردازش داده‌ها
data.fillna(0, inplace=True)

# تقسیم داده‌ها به مجموعه آموزشی و آزمایشی
train_data, test_data = train_test_split(data, test_size=0.2)

# مدل Collaborative Filtering
model = NearestNeighbors(metric='cosine')
model.fit(train_data)

# توصیه برای کاربر خاص
user_id = 1 # شناسه کاربر

distances, indices = model.kneighbors(train_data.loc[user_id].values.reshape(1, -1), n_neighbors=5)

# نمایش نتایج
recommended_items = train_data.columns[indices.flatten()]

print("توصیه‌های پیشنهادی برای کاربر", recommended_items)

# گراف نتایج
plt.figure(figsize=(10, 5))
plt.bar(recommended_items, distances.flatten())
plt.title('توصیه‌های پیشنهادی')
plt.xlabel('محصولات')
```

```
plt.ylabel('فاصله')
```

```
plt.show()
```