**خلاصه**

هدف این پروژه دریافت اطلاعات (نام فیلم، نام بازیگر، ژانر، نام کارگردان و ... ) از هر کاربر و جستجوی نزدیک ترین فیلم ها بر اساس این اطلاعات می باشد.

**توضیح جزئی پروژه**

ابتدا محیط کاربری به کاربر نمایش داده میشود. هر کاربر نام کاربری خود را وارد کرده و سپس اطلاعات مورد نظر خود را با پر کردن فیلد های مربوطه جستجو میکند. در ادامه با استفاده KD TREE در بانک اطلاعاتی بر اساس فیلم هایی که کاربر قبلا جستجو کرده بود نتیجه جستجویی به ترتیب نزدیکی به جستجو های پیشین این کاربر حاصل میشود که در انتها به کاربر نمایش داده شده و نتیجه جستجو در بانک اطلاعاتی مربوط به همین کاربر ذخیره می شود.

**محیط گرافیکی کاربری (GUI)**

وظیفه نمایش محیط گرافیکی به کاربر را دارد که با استفاده از کتابخانه PyQt5 در زبان پایتون پیاده سازی شده است. در این محیط فیلد ها و محیط جستجو قرار دارند و کاربر با وارد کردن ورودی های مورد نظر جستجو را شروع میکند. این بخش از یک فایل xml که تمام front-end محیط کاربری را شامل میشود داراست. سپس به وسیله کتابخانه pyqt این فایل به یک فایل گرافیکی پایتون تبدیل میشود.

برای هر front-end یک فایل back-end وجود دارد که وظیفه مدیریت بخش گرافیکی را بر عهده دارد مثلا زمانی که کاربر دکمه ی جستجو را انتخاب میکند، بخش پس زمینه قسمت مربوط به جستجو را فراخوانی کرده و وقتی عملیات به پایان رسید نتایج را نمایش میدهد.

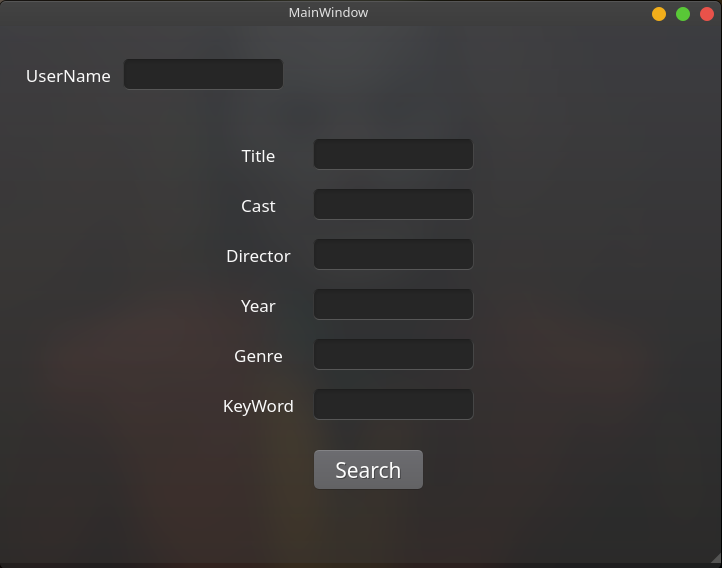
محیط کاربری این پروژه شامل دو بخش است :

1. صفحه اصلی :

شامل فیلد های برای دریافت ورودی از کاربر (mainWindow)

1. صفحه نمایش نتایج :

در این مرحله نتایجی که از جستجو بدست آمده به کاربر نمایش داده میشود. ترتیب نمایش فیلم ها به نزدیکی آن ها بر اساس الگوریتم جستجو میباشد. (Widget.listView)

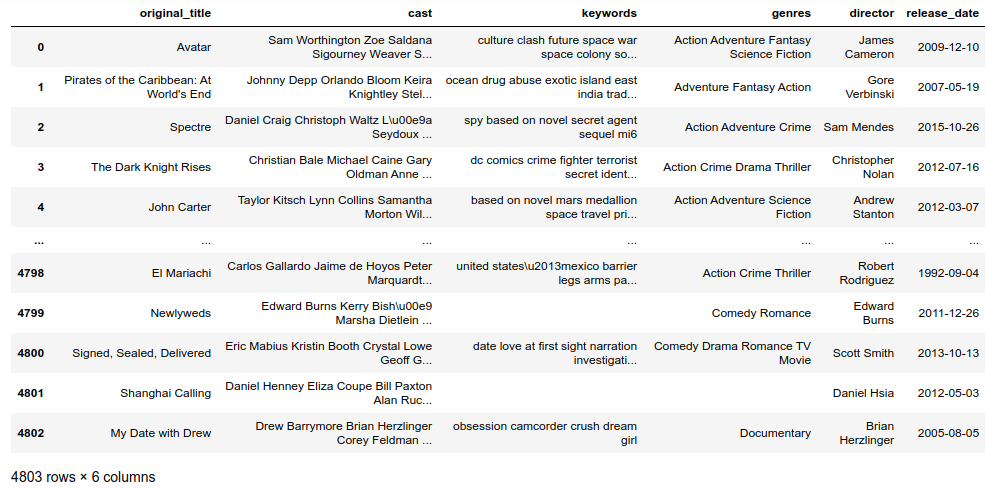


**بانک اطلاعاتی و داده ها**

برای داده های این پروژه از بانک اطلاعاتی سایت imdb استفاده شد که از این [لینک](https://datasets.imdbws.com/) قابل دریافت است.

پس از دریافت داده ها ویژگی هایی که مورد نیاز نبود حذف گردید و همچنین برخی از داده ها مقداری نداشتند (null بودند) که این داده ها مدیریت شده و با مقادیر مناسب پر شدند.

پس از پیش پردازش داده ها، داده های بدست آمده جدید به صورت زیر بودند:



پس از آن این داده ها بر روی یک پایگاه داده ذخیره شدند.

پایگاه داده مورد استفاده برای این پروژه SQLite است که یک مدیر پایگاه داده مناسب و سریع برای نوع اطلاعات مورد نظر میباشد.

**الگوریتم جستجو**

در این بخش با استفاده از الگوریتم جستجوی KD TREE اطلاعات مورد نظر را طبق درخواست کاربر جستجو کرده و به صورت یک آرایه نتیجه را برمیگردانیم.

در واقع یک [ساختمان داده](https://fa.wikipedia.org/wiki/%D8%B3%D8%A7%D8%AE%D8%AA%D9%85%D8%A7%D9%86_%D8%AF%D8%A7%D8%AF%D9%87%E2%80%8C%D9%87%D8%A7) برای سازماندهی و ذخیره‌سازی [مجموعه متناهی](https://fa.wikipedia.org/wiki/%D9%85%D8%AC%D9%85%D9%88%D8%B9%D9%87_%D9%85%D8%AA%D9%86%D8%A7%D9%87%DB%8C) از نقاط در فضای k-بعدی می باشد. این درخت، یک [داده ساختار](https://fa.wikipedia.org/wiki/%D8%AF%D8%A7%D8%AF%D9%87_%D8%B3%D8%A7%D8%AE%D8%AA%D8%A7%D8%B1) مفید برای بسیاری از برنامه‌های کاربردی است، مانند جستجوهایی که شامل کلید واژه‌های جستجوی چند [بعدی](https://fa.wikipedia.org/wiki/%D8%A8%D8%B9%D8%AF) هستند. درخت کی‌دی یک نوع [درخت دودویی](https://fa.wikipedia.org/wiki/%D8%AF%D8%B1%D8%AE%D8%AA_%D8%AF%D9%88%D8%AF%D9%88%DB%8C%DB%8C) از نوع BSP یا [Binary space partitioning](https://en.wikipedia.org/wiki/en:Binary_space_partitioning) است.

هر گره آن یک نقطه K بعدی است و هر گره غیر برگ را می‌توان به عنوان یک مولد جداکننده ابر صفحه، که فضا را به دو قسمت، که به عنوان زیر فضا شناخته می‌شوند، تقسیم می‌کند دانست.

نقاط سمت چپ ابرصفحه، در زیر درخت سمت چپ آن گره و نقاط سمت راست ابرصفحه، در زیر درخت راست نمایش داده می‌شوند .

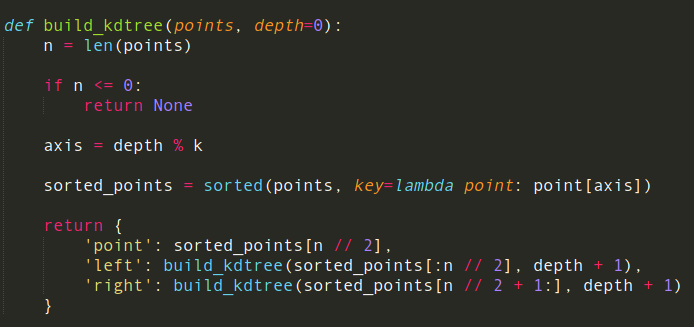
هر گره در درخت با یکی از K بعد به همراه ابرصفحه‌ای که بر محور آن بعد عمود است، مرتبط است. بنابراین، به عنوان مثال، اگر برای یک جداسازی مشخص، محور X انتخاب شده باشد، تمام نقاط در زیر درخت که مقدار X کوچکتری از آن گره دارند در زیر درخت سمت چپ و تمام نقاط با مقدار X بزرگتر، در زیر درخت سمت راست ظاهر می‌شوند . در چنین حالتی، ابر صفحه با مقدار X معین می‌شود و بردار نرمال آن محور X خواهد بود.

برای پیاده سازی این الگوریتم، یک فایل kdtree\_search.py وجود دارد که یک تابع بازگشتی بخش اصلی این کد را تشکیل میدهد.

در این تابع، شرط توقف رابطه بازگشتی رسیدن به گره ی null یا خالی در نظر گرفته شده است.

در هر عمق از درخت یک قابلیت از داده ها در نظر گرفته می شود و سپس تمام داده ها بر اساس آن قابلیت مرتب میشوند.

در انتها به صورت بازگشتی گره اصلی (root) و گره چپ (مقادیر کوچکتر از root) و راست (مقادیر بزرگتر از root) ساخته میشوند.

تابع اصلی برای ساختن درخت kd :

تابع اصلی برای جستجوی نزدیکترین نقطه در درخت ke :



**تبدیل متن به عدد برای درخت**

**Bag of Words**

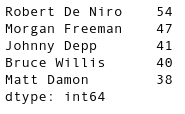
با استفاده از روش Bag of Words، یک متن (مانند یک جمله یا سند) به صورت یک بسته چند مجموعه از کلمات آن، بی‌توجه به دستور زبان و حتی نظم کلمات نمایش داده می‌شود. مدل بسته کلمات معمولاً در روش‌های [دسته‌بندی اسناد](https://fa.wikipedia.org/wiki/%D8%AF%D8%B3%D8%AA%D9%87%E2%80%8C%D8%A8%D9%86%D8%AF%DB%8C_%D8%A7%D8%B3%D9%86%D8%A7%D8%AF) مورد استفاده قرار می‌گیرد که در آن وقوع هر کلمه به عنوان یک ویژگی برای آموزش [طبقه‌بندی آماری](https://fa.wikipedia.org/wiki/%D8%B7%D8%A8%D9%82%D9%87%E2%80%8C%D8%A8%D9%86%D8%AF%DB%8C_%D8%A2%D9%85%D8%A7%D8%B1%DB%8C) استفاده می‌شود.

**TF-IDF**

در این شیوه به هر کلمه‌ای در هر متن یک ‌وزن بر اساس فراوانی آن در متن و فراوانی کلمه در سایر متونِ پیکره متنی داده می‌شود. در واقع هدف این سیستمِ وزن‌دهی، نشان‌دادن اهمیت کلمه در متن است. این مسئله کاربردهای بسیاری در بازیابی اطلاعات و متن کاوی دارد. وزن کلمه با افزایش تعداد تکرار آن در متن افزایش می‌یابد، اما توسط تعداد متونی که کلمه در آن ظاهر می‌شود کنترل می‌شود. به این معنی که اگر کلمه‌ای در بسیاری از متون ظاهر شود احتمالاً کلمه‌ای متداول است و ارزش چندانی در ارزیابی متن ندارد.

از آنجایی که جایگاه کلمات از لحاظ گرامری در این پروژه تاثیری ندارند بنابراین از روش bag of words برای تبدیل متن(کلمات) به اعداد استفاده شد.

همانطور که در تصاویر زیر مشاهده میکنید بر فرض مثال ابتدا تعداد تکرار هر کارگردان در تمام فیلم ها را محاسبه کرده و سپس بر اساس تعداد وزن دهی صورت میگیرد.



**سیستم پیشنهاد دهنده**

یک سامانه پیشنهاد دهنده با تحلیل رفتار کاربر خود، اقدام به پیشنهاد مناسب‌ترین اقلام (داده، اطلاعات، کالا و…) می‌نماید. این سیستم رویکردی است که برای مواجهه با مشکلات ناشی از حجم فراوان و رو به رشد اطلاعات ارائه شده‌است و به کاربر خود کمک می‌کند تا در میان حجم عظیم اطلاعات سریع‌تر به هدف خود نزدیک شوند.

مزایای این روش از جستجو به شرح زیر می باشد :

* کاربر را در اخذ تصمیمی یاری می‌کنند (که مثلاً از میان چندین گزینه پیش رو کدام بهتر است و آن را انتخاب کند و …).
* موجب افزایش آگاهی کاربر، در زمینه مورد علاقه وی می‌شود (مثلاً در حین ارائه توصیه به کاربر موجب می‌شود تا وی با اطلاعات، اقلام و اشیاء جدیدی را که قبلاً آن‌ها را نمی‌شناخته، آشنا شود).

دو روش اصلی برای اینگونه از سیستم ها وجود دارد:

### **پالایش گروهی اطلاعات (Collaborative filtering)**

در این روش بر اساس رفتار و تاریخچه تمام کاربران، پیشنهادات جدید بر اساس شباهت رفتاری کاربران مشابه به کاربر مورد نظر داده می شود.

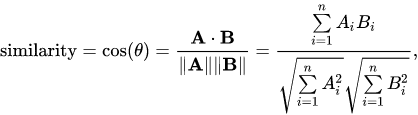
### **پالایش اطلاعات بر اساس محتوا (Content-based filtering)**

### در این روش بر اساس رفتار و تاریخچه کاربر مورد نظر، پیشنهادات مشابه جدید به کاربر داده می شوند.

**معیار شباهت cosine**

روشی آماری که برای سنجش شباهت دو داده عددی مورد استفاده قرار می گیرد. اگر برای هر داده بر اساس ویژگی ها(features) یک بردار در فضای n بعدی در نظر بگیریم، با سنجش کسینوس زاویه بین دو بردار میزان شباهت یا فاصله آن دو بردار در فضای n بعدی به دست می آید.





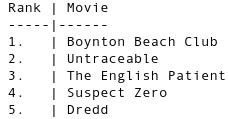
**روش مورد استفاده برای طراحی سیستم پیشنهاد دهنده**

پس از اینکه داده های متنی به عددی تبدیل شدند، یک بار برای تمام داده ها ماتریس شباهت مصاحبه می شود و پس از آن در هر جستجوی کاربر از این ماتریس استفاده میشود.

برای استفاده از این سیستم پیشنهاد دهنده مدل داده ای KD Tree را اساس شباهت فیلم ها می سازیم،‌ به این ترتیب که به هنگام مرتب سازی داده های در هر شاخه از درخت از بردار مرتب شده ی شباهت فیلم ها که بالاتر ساخته شد استفاده میکنیم.

به این ترتیب با هر جستجوی کاربر علاوه بر فیلم اصلی جستجو شده، فیلم های مشابه هم در درخت به راحتی قابل دسترسی و نمایش هستند.

به طور مثال اگر کاربری فیلمی با عنوان Fight Club را جستجو کند، پنج فیلم اول مشابه که سستم به کاربر پیشنهاد می دهد به صورت زیر خواهد بود :



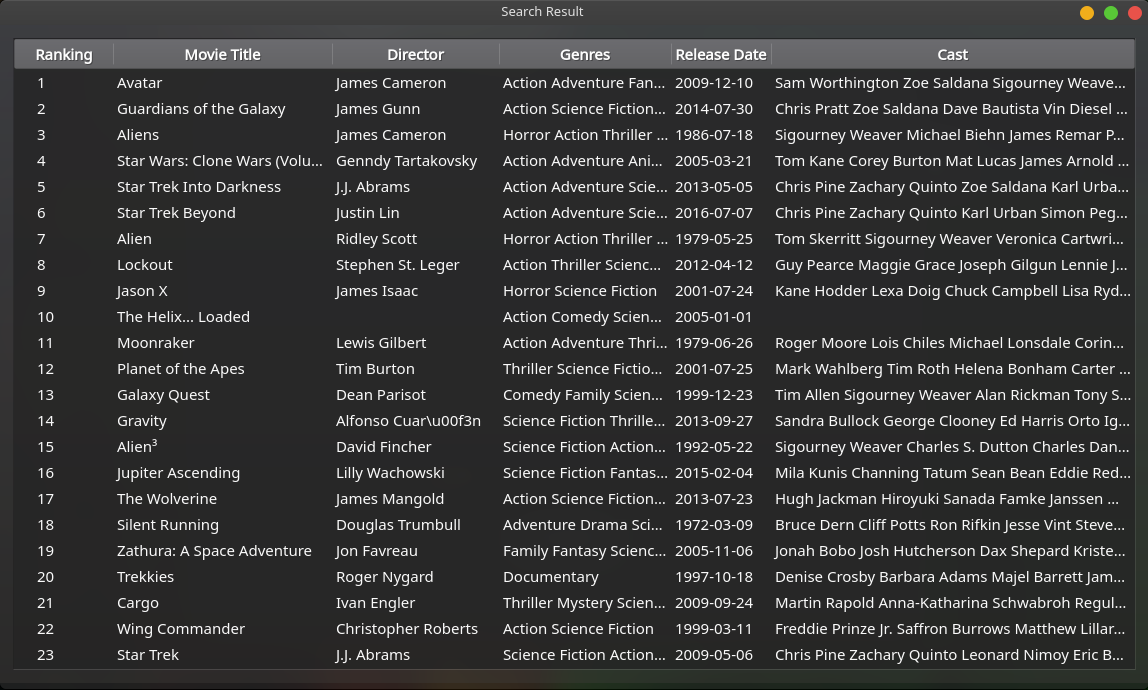
**نتایج نهایی**

با فشردن Search توسط کاربر، داده های ورودی بر اساس مدلی که از پیش متن و کلمات را تبدیل به داده های عددی کردیم،‌ تبدیل به داده های عددی شده سپس در درخت KD جستجو میشوند و علاوه بر فیلم جستجو شده، فیلم های مشابه هم از درخت استخراج می شوند.

پس از حاصل شدن نتایج جستجو، صفحه گرافیکی به صورت زیر به کاربر نمایش داده می شود که شامل عنوان دقیق فیلم جستجو شده به همراه فیلم های مشابه (پیشنهادات) می باشد.

همچنین فیلم جستجو شده توسط کاربر در بانک اطلاعاتی ذخیره شده و در جستجو های بعدی مورد استفاده سیستم پیشنهاد دهنده قرار خواهد گرفت.

به طور مثال نتیجه جستجو برای فیلم Avatar به صورت زیر خواهد بود:



**منابع**