گزارش پروژهٔ علم داده

دکتر نادری

زهرا دهقان 400521333

هلیا شمس زاده 400521486

الف) بخش EDA

- برای رسم نمودارها از plotly استفاده شده است تا فهم و خواندن نمودارها راحت‌تر باشد.

ایجاد توابع کمکی برای ستون‌هایی با مقادیر از نوع دیکشنری

تابع parse_list_column برای تبدیل رشته‌هایی که نمایش‌دهنده یک لیست هستند به یک لیست واقعی استفاده می‌شود. این تابع ابتدا بررسی می‌کند که مقدار موجود در ستون از نوع رشته‌ای باشد و در صورت تأیید، با استفاده از ast.literal_eval آن را به یک لیست

تبدیل می‌کند. تابع extract_dict_field برای استخراج مقدار خاصی از یک دیکشنری که در یک ستون قرار دارد، استفاده می‌شود. این تابع مقدار موردنظر را بر اساس کلید مشخص‌شده (field_name) از دیکشنری استخراج کرده و مقدار آن را بازمی‌گرداند. اگر مقدار موجود در ستون از نوع دیکشنری نباشد، مقدار Noneبرگردانده می‌شود. این روش برای پردازش ستون‌هایی که شامل اطلاعاتی به‌صورت دیکشنری (مانند جزئیات امتیاز فیلم‌ها یا اطلاعات مربوط به کارگردان) هستند، کاربرد دارد.

پیدا کردن تعداد ژانرهای منحصر به فرد

ابتدا، با استفاده از apply و ast.literal_eval، مقادیر رشته‌ای که نمایش‌دهنده یک لیست هستند، به لیست واقعی تبدیل می‌شوند. این کار فقط در صورتی انجام می‌شود که مقدار موردنظر از نوع رشته‌ای بوده و با [شروع شده باشد؛ در غیر این صورت، مقدار به‌صورت یک لیست خالی تنظیم می‌شود تا از بروز خطا جلوگیری شود.

پس از تبدیل رشته‌ها به لیست، با استفاده از تابع explode، مقادیر لیست در هر سطر از DataFrame گسترش داده می‌شوند، به‌طوری که هر ژانر به‌عنوان یک مقدار مستقل در یک ردیف جداگانه قرار می‌گیرد. سپس، مقدار dropna اعمال می‌شود تا مقادیر NaN حذف شوند.

در مرحله بعد، از یک apply دیگر استفاده شده است تا نام ژانرها از دیکشنری استخراج شود. در صورتی که مقدار موجود از نوع دیکشنری باشد، مقدار مربوط به کلید 'name' دریافت می‌شود؛ در غیر این صورت، مقدار None برگردانده می‌شود. مقادیر None نیز با dropna حذف شده و در نهایت، مقادیر منحصربه‌فرد (unique()) استخراج می‌شوند.

---

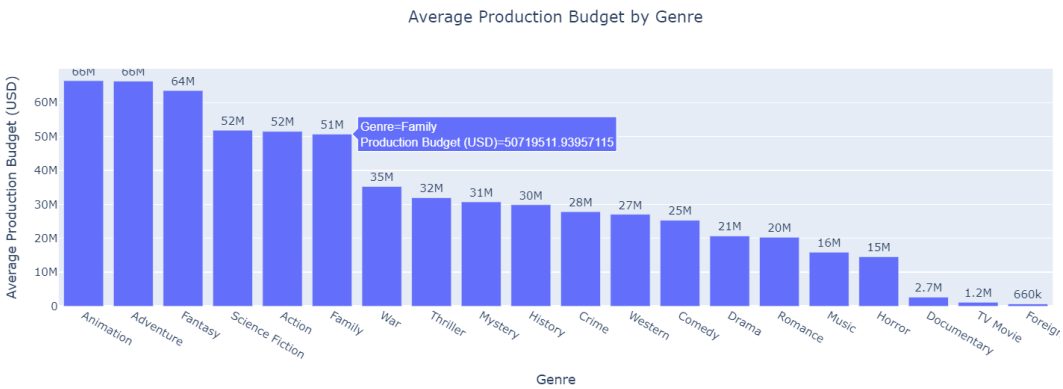
1) متوسط هزینه برای هر ژانر فیلم چقدر است؟

ابتدا، یک کپی از DataFrame اصلی فیلم‌ها (df_movies) با نام df_genres_expanded ایجاد می‌شود تا از تغییر داده‌های اصلی جلوگیری شود. سپس، تابع parse_list_column روی ستون rt_genres اعمال می‌شود تا مقادیر رشته‌ای که نمایش‌دهنده لیست هستند، به لیست‌های واقعی تبدیل شوند.

در مرحله بعد، از متد explode برای گسترش ژانرها استفاده می‌شود. این روش باعث می‌شود که هر فیلمی که دارای چند ژانر است، در چندین ردیف نمایش داده شود، به‌طوری که هر ردیف نمایانگر یکی از ژانرهای مربوط به فیلم باشد. سپس، تابع extract_dict_field برای استخراج نام ژانرها از دیکشنری‌های موجود در ستون rt_genres استفاده می‌شود و مقدار استخراج‌شده در یک ستون جدید با نام genre_name ذخیره می‌شود. پس از آن، تمامی ردیف‌هایی که مقدار genre_name آن‌ها مقدار NaN دارد، حذف می‌شوند.

در نهایت، میانگین بودجه تولید فیلم‌ها برای هر ژانر محاسبه می‌شود. این کار با استفاده از groupby('genre_name') روی ستون rt_production_budget انجام شده و میانگین بودجه با mean() محاسبه می‌شود. نتیجه‌ی این عملیات در قالب یک DataFrame جدید ذخیره می‌شود که دارای دو ستون genre_name و rt_production_budgetاست، جایی که هر ردیف میانگین بودجه تولید برای یک ژانر خاص را نشان می‌دهد.

نمودار:



نتیجه: طبق نمودار می‌توان فهمید پرخرج‌ترین ژانرها به ترتیب انیمیشن، ماجراجویی، فانتزی، علمی-تخیلی، اکشن و خانوادگی هستند. کم‌خرج‌ترین ژانرها هم خارجی، تلویزیون و مستند هستند.

---

2) سهم هر کشور در مجموع هزینهٔ هر ژانر فیلم چقدر است؟ (برای 5 تا از پرخرج‌ترین ژانرها به دست آورید.)

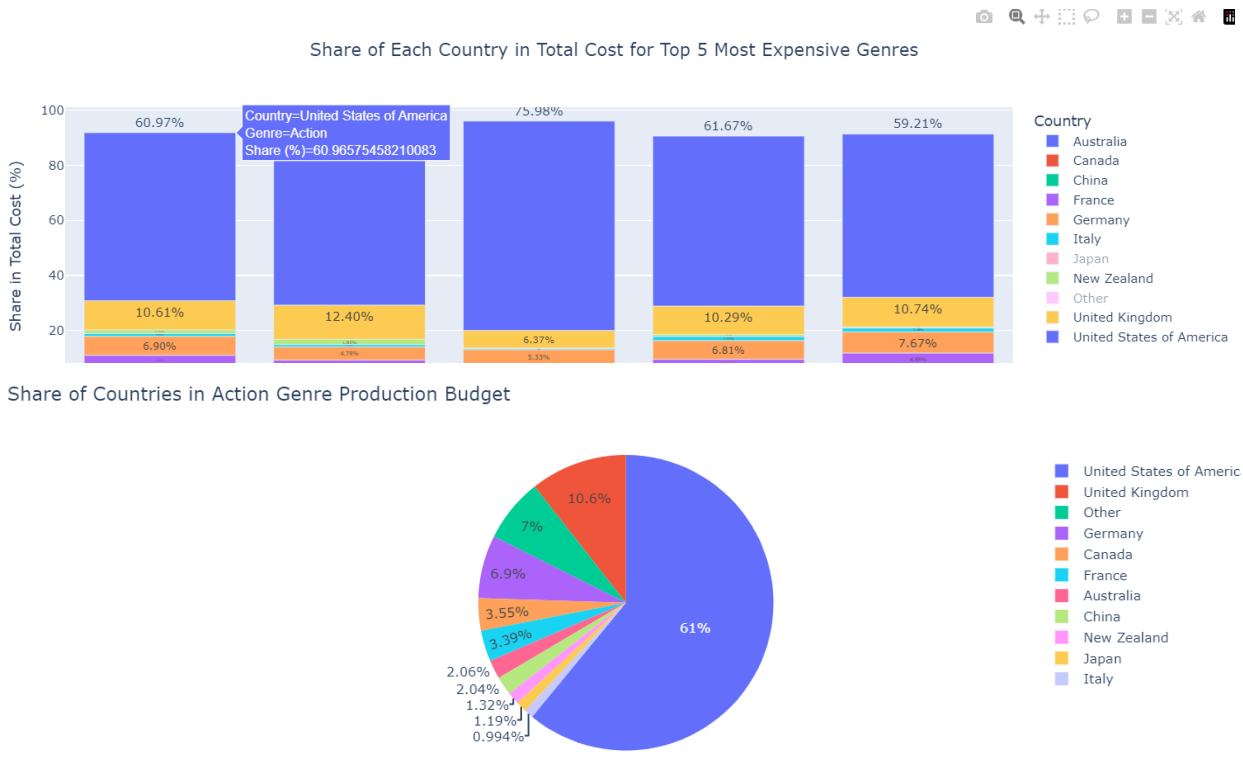
ابتدا، ستون‌های rt_genres و rt_production_countries در DataFrame temp_df با استفاده از تابع parse_list_column پردازش می‌شوند تا مقادیر رشته‌ای که نمایش‌دهنده لیست هستند، به لیست‌های واقعی تبدیل شوند.

سپس، DataFrame با استفاده از explode گسترش داده می‌شود تا هر فیلمی که دارای چند ژانر یا چند کشور تولیدکننده است، در چندین ردیف نمایش داده شود. این باعث می‌شود که هر ردیف فقط یک ژانر و یک کشور را شامل شود. در ادامه، نام ژانرها و نام کشورها از دیکشنری‌های موجود در ستون‌های rt_genres و rt_production_countries با استفاده از تابع extract_dict_field استخراج شده و در دو ستون جدید genre_name و country_name ذخیره می‌شوند. سپس، تمامی ردیف‌هایی که فاقد مقدار معتبر در هر یک از این دو ستون هستند، حذف می‌شوند.

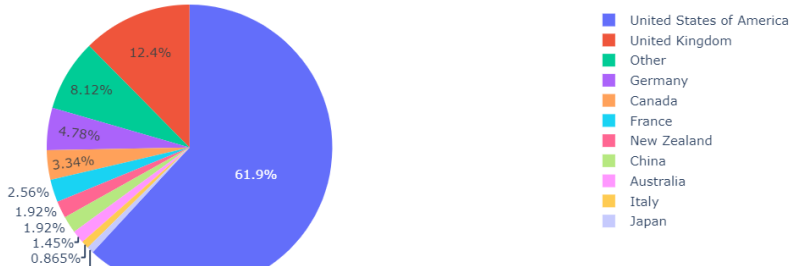
پس از پاک‌سازی داده‌ها، مجموع بودجه تولید فیلم‌ها برای هر ترکیب کشور-ژانر با استفاده از groupby محاسبه می‌شود. نتیجه این گروه‌بندی در DataFrame country_genre_budget ذخیره شده است. همچنین، مجموع بودجه تولید برای هر ژانر در سطح کلی محاسبه شده و در DataFrame total_budget_by_genre ذخیره می‌شود. این اطلاعات به DataFrame اولیه ترکیب کشور-ژانر اضافه می‌شود تا امکان محاسبه سهم هر کشور از کل بودجه تولید ژانر مربوطه فراهم شود. سهم هر کشور برای یک ژانر خاص بر اساس نسبت بودجه آن کشور به کل بودجه ژانر در سطح جهانی، در ستون share به‌صورت درصدی ذخیره می‌شود.

در نهایت، برای هر کشور، پنج ژانر با بیشترین بودجه تولید شناسایی شده‌اند. این کار از طریق مرتب‌سازی داده‌ها بر اساس country_name و rt_production_budget (به ترتیب صعودی برای کشورها و نزولی برای بودجه) انجام شده و سپس با استفاده از groupby و head(5)، پنج ژانر با بیشترین بودجه برای هر کشور استخراج شده‌اند.

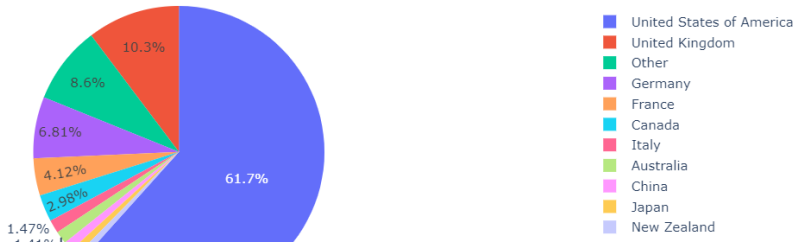
نمودار:



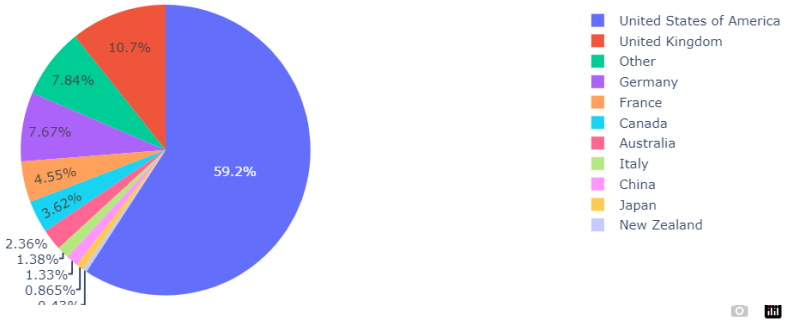
Share of Countries in Adventure Genre Production Budget



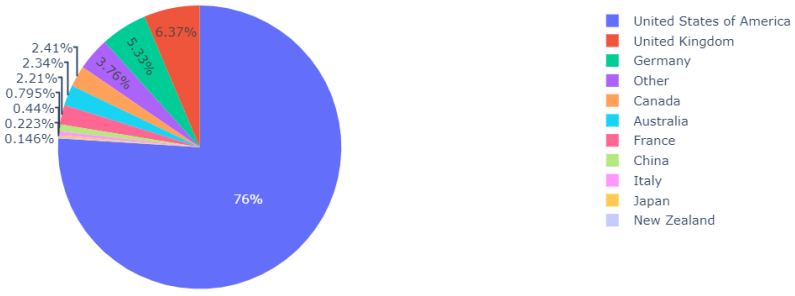
Share of Countries in Drama Genre Production Budget



Share of Countries in Thriller Genre Production Budget



Share of Countries in Comedy Genre Production Budget



نتیجه: در همهٔ 5 تا ژانر، بیشترین سهم برای آمریکا و سپس انگلستان و آلمان می‌باشد.

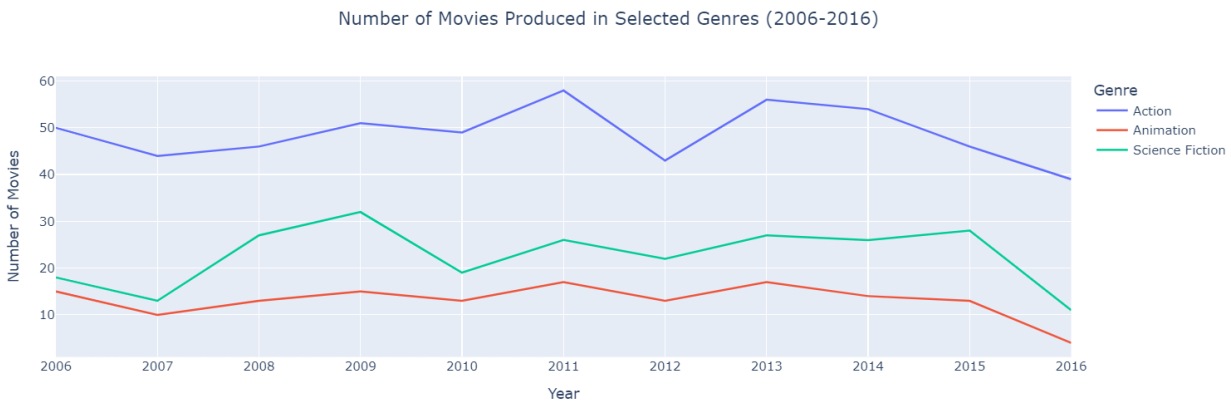
3) تعداد فیلم های ساخته شده در 3 ژانر را در 10 سال گذشته مقایسه کنید.

ابتدا، ستون rt_release_date که شامل تاریخ انتشار فیلم‌ها است، با استفاده از pd.to_datetime به یک نوع داده‌ی زمانی (datetime)تبدیل می‌شود. برای جلوگیری از بروز خطا هنگام تبدیل مقادیر نامعتبر، گزینه errors='coerce' استفاده شده است که مقدارهای نامعتبر را به NaT تبدیل می‌کند. سپس، سال انتشار هر فیلم از این داده‌ی زمانی استخراج شده و در یک ستون جدید با نام release_year ذخیره می‌شود.

پس از آن، یک فیلتر روی داده‌ها اعمال می‌شود تا فقط فیلم‌هایی که بین سال‌های ۲۰۰۷ تا ۲۰۱۶ منتشر شده‌اند، در DataFrame last_10_yearsنگه‌داشته شوند. سپس، ستون rt_genres که شامل لیستی از ژانرهای هر فیلم است، با استفاده از متد explode گسترش داده می‌شود، به‌طوری که هر فیلم در چندین ردیف قرار گیرد، به‌ازای هر ژانری که به آن تعلق دارد. نام ژانرها با استفاده از تابع extract_dict_field استخراج شده و در ستون genre_name ذخیره می‌شود.

در مرحله بعد، از بین تمام ژانرها، فقط سه ژانر اکشن، انیمیشن و علمی-تخیلی انتخاب می‌شوند. این کار از طریق اعمال یک فیلتر روی ستون genre_name و استفاده از متد isin(selected_genres) انجام می‌شود. در نهایت، تعداد فیلم‌های مربوط به هر ژانر در هر سال با استفاده از groupby(['release_year', 'genre_name']) محاسبه شده و تعداد فیلم‌های موجود در هر گروه با size() شمارش می‌شود. نتیجه در قالب یک DataFrame جدید با نام genre_counts ذخیره شده که شامل سه ستون release_year، genre_name و movie_count است.

نمودار:



نتیجه: سه ژانر انتخاب شده اند. در بین این سه ژانر، در تمام این سال‌ها تعداد فیلم‌های بیشتری در ژانر اکشن ساخته شده است. اما هر سه از سال 2013 افت داشته اند. ژانر اکشن در این بازه زمانی پرطرفدار بوده اما از 2011 روند نزولی گرفته است. ژانر علمی تخیلی رشد داشته اما در سال 2016 افت کرده است. تولید فیلم‌های انیمیشنی به‌طور کلی کمتر از سایر ژانرها بوده اما تغییرات محسوسی نداشته است.

---

4) به طور متوسط کدام کشور ها طولانی ترین فیلم ها و کوتاه ترین فیلم ها را می‌سازند؟

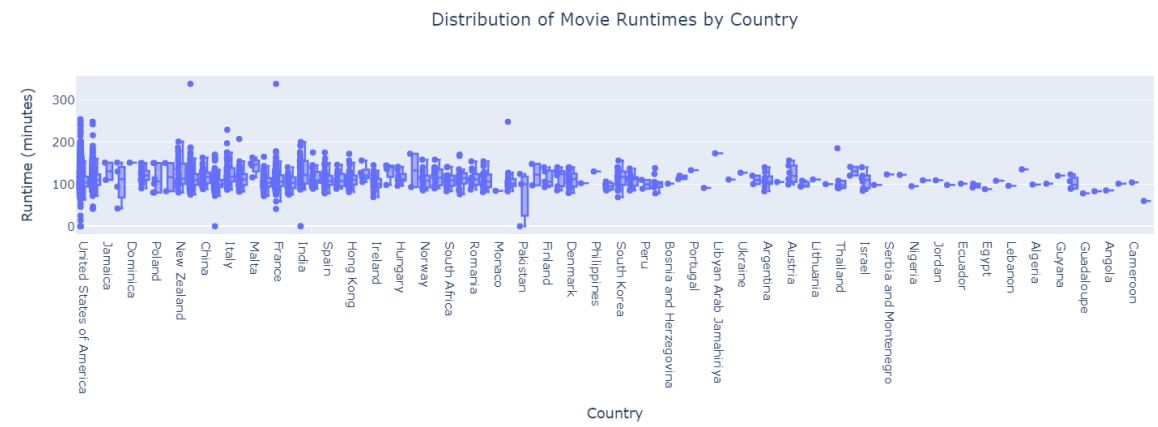
ابتدا، یک کپی از DataFrame اصلی فیلم‌ها (df_movies) در متغیر temp_df ذخیره می‌شود تا داده‌های اصلی بدون تغییر باقی بمانند. سپس، ستون rt_production_countries که شامل اطلاعات کشورها به‌صورت رشته‌ای (لیست دیکشنری‌ها) است، پردازش می‌شود. با استفاده از ast.literal_eval، مقادیر رشته‌ای به لیست‌های واقعی از دیکشنری‌ها تبدیل می‌شوند. این کار فقط در صورتی انجام می‌شود که مقدار موجود از نوع رشته باشد؛ در غیر این صورت، مقدار به‌صورت یک لیست خالی قرار داده می‌شود.

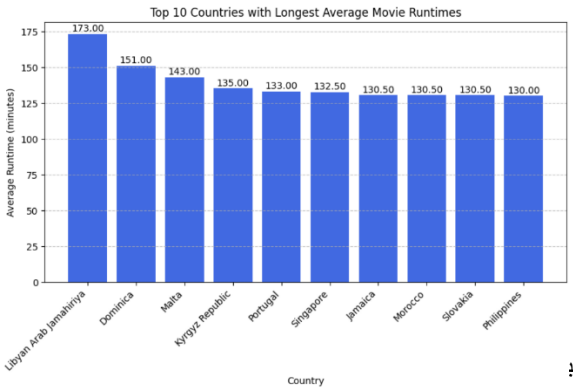
پس از تبدیل، متد explode برای گسترش داده‌ها استفاده می‌شود، به‌طوری که هر فیلم که در چند کشور تولید شده باشد، در چندین ردیف جداگانه نمایش داده شود، به ازای هر کشور یک ردیف. سپس، نام کشورها از دیکشنری‌های موجود در ستون rt_production_countries استخراج شده و در ستون country_name ذخیره می‌شود. اگر مقدار موجود از نوع دیکشنری نباشد، مقدار None در نظر گرفته می‌شود.

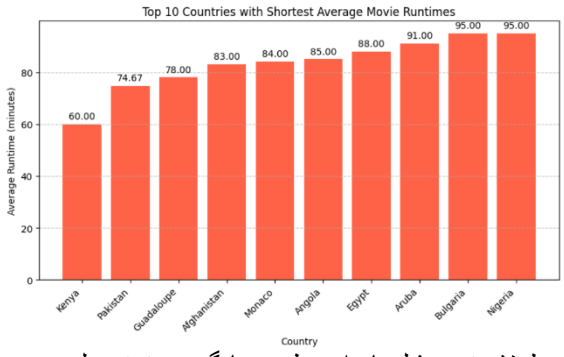
در ادامه، ردیف‌هایی که مقدار country_name یا rt_runtime آن‌ها مقدار نامعتبر (NaN) دارند، حذف می‌شوند تا فقط داده‌های معتبر باقی بمانند. سپس، مدت‌زمان میانگین فیلم‌ها برای هر کشور محاسبه می‌شود. این کار با استفاده از groupby('country_name') روی ستون rt_runtime انجام شده و مقدار میانگین با mean() محاسبه می‌شود. نتیجه این محاسبات در DataFrame average_runtime_by_country ذخیره می‌شود.

در مرحله‌ی آخر، کشور با بیشترین و کمترین میانگین زمان پخش فیلم‌ها شناسایی می‌شود. این کار از طریق idxmax() و idxmin() روی ستون rt_runtime انجام می‌شود که به ترتیب بیشترین و کمترین مقدار میانگین را مشخص می‌کنند. اطلاعات مربوط به این دو کشور در متغیرهای longest_runtime_country و shortest_runtime_country ذخیره می‌شوند.

نمودار:





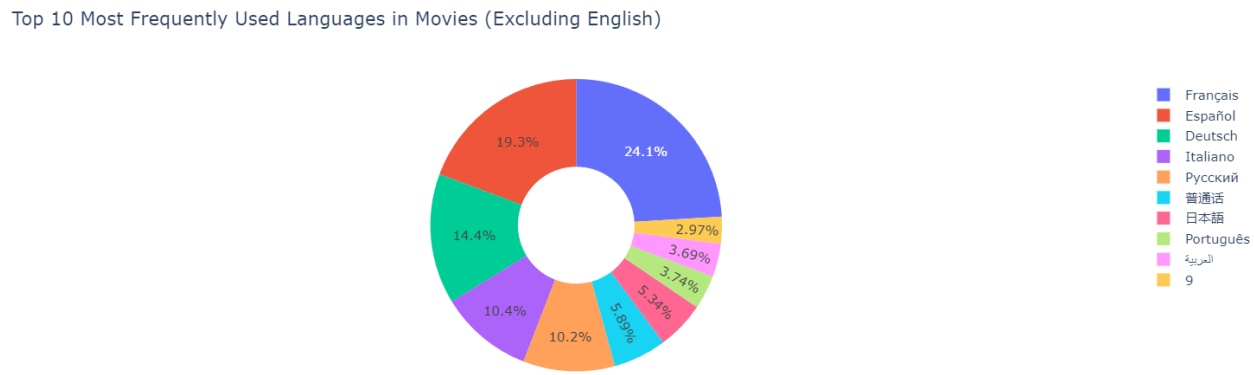


نتیجه: طولانی‌ترین فیلم‌ها را به طور میانگین به ترتیب لیبی، دومینیکن، پاکستان و گوادالوپ ساخته‌اند.

5) به غیر از انگلیسی، پر تکرار ترین زبان ها در فیلم ها چه هستند؟

ابتدا یک کپی از DataFrame اصلی ساخته می‌شود تا داده‌ها بدون تغییر باقی بمانند. سپس، ستون rt_languagesکه شامل لیستی از زبان‌های هر فیلم است، به صورت دیکشنری‌های قابل پردازش تبدیل می‌شود. پس از آن، با استفاده از متد explode، هر زبان به ردیف جداگانه‌ای تبدیل می‌شود و نام زبان‌ها از دیکشنری‌ها استخراج می‌شود. ردیف‌هایی که دارای مقادیر نامعتبر در ستون زبان هستند، حذف می‌شوند. همچنین، فیلم‌های به زبان انگلیسی از تحلیل کنار گذاشته می‌شوند. در نهایت، تعداد فیلم‌هایی که به هر زبان غیر انگلیسی تولید شده‌اند، شمارش شده و در یک DataFrame جدید ذخیره می‌شود.

نمودار:



نتیجه: به غیر از انگلیسی، به ترتیب فرانسوی، اسپانیایی، آلمانی (داچ) و ایتالیایی استفاده شده اند.

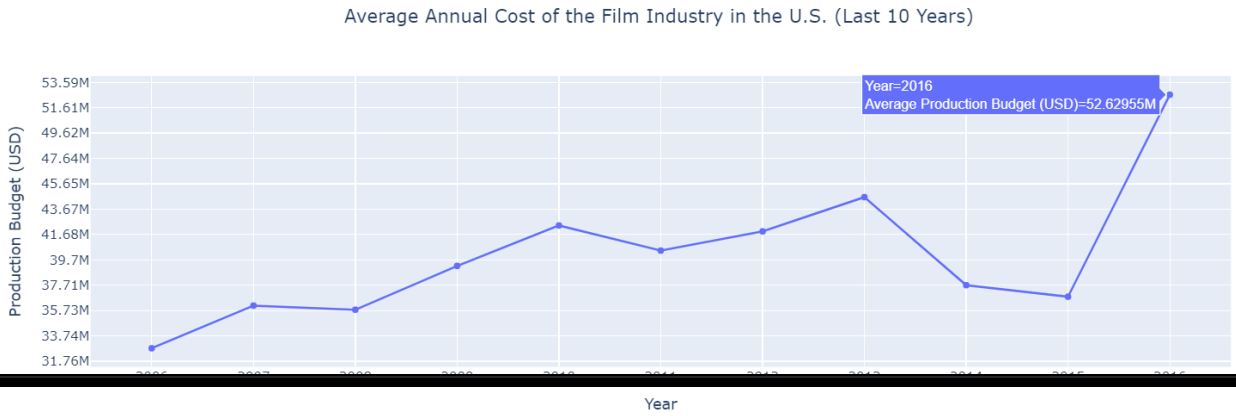
---

6) آمریکا در 10 سال گذشته، به طور متوسط در هر سال چقدر در صنعت فیلمسازی هزینه کرده است؟ (به تفکیک سال)

در ابتدا، تاریخ انتشار فیلم‌ها از ستون rt_release_date به فرمت تاریخ و زمان تبدیل می‌شود، سپس فقط سال انتشار فیلم‌ها از آن استخراج می‌شود و در ستون جدید release_year ذخیره می‌گردد. پس از آن، فیلم‌های تولید شده در ایالات متحده آمریکا فیلتر می‌شوند با استفاده از country_name == 'United States of America'. سپس، برای محدود کردن تحلیل به فیلم‌های ۱۰ سال اخیر، از تابع between استفاده می‌شود تا تنها فیلم‌هایی که در بازه زمانی ۱۰ سال گذشته (از سال جاری تا سال ۱۰ سال پیش) منتشر شده‌اند، در نظر گرفته شوند.

در مرحله بعد، برای هر سال، میانگین هزینه تولید محاسبه می‌شود. این کار با استفاده از groupby('release_year') و محاسبه میانگین rt_production_budget انجام می‌شود. نتیجه این محاسبات در yearly_cost ذخیره می‌شود. در نهایت، برای اطمینان از عدم وجود مقادیر NaN در ستون‌های هزینه تولید، از fillna(0) استفاده می‌شود تا این مقادیر به صفر تبدیل شوند.

نمودار:

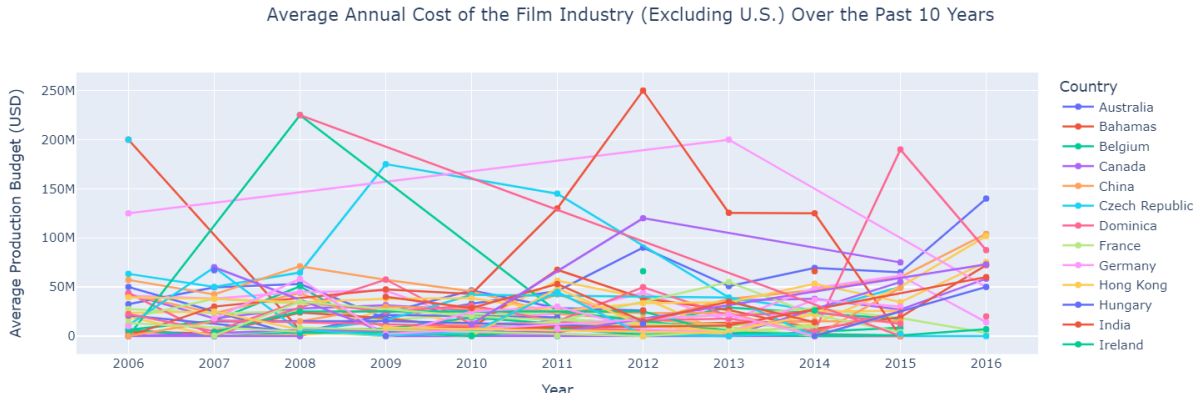


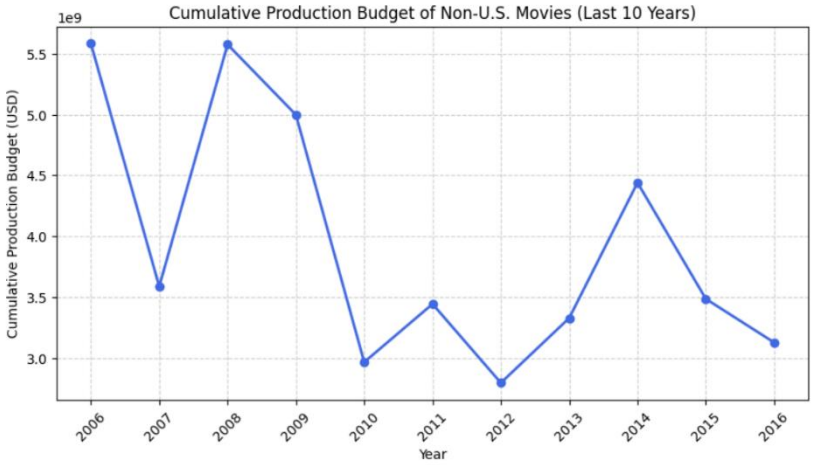
نتیجه: به طور کلی سرمایه‌گذاری آمریکا از سال 2006 روند صعودی داشته است (با کاهش در سال‌های 2014 و 2015) اما در نهایت به بیشترین میزان خود در سال 2016 رسیده است.

7) روند قبلی را بدون در نظر گرفتن کشور برای 10 سال گذشته مقایسه کنید.

در ابتدا، یک کپی از DataFrame اصلی (df_movies) ساخته می‌شود و سپس ستون rt_release_date که شامل تاریخ انتشار فیلم‌ها است، به فرمت تاریخ تبدیل می‌شود. فیلم‌هایی که تاریخ انتشار نامعتبر دارند، با استفاده از dropna حذف می‌شوند. سپس سال انتشار فیلم‌ها در release_yearذخیره می‌شود. در مرحله بعد، ستون rt_production_countries که شامل لیستی از کشورهای تولید فیلم است، با استفاده از تابع parse_list_column به لیست‌های دیکشنری تبدیل می‌شود. سپس با استفاده از explode، لیست کشورهای تولید به ردیف‌های جداگانه تبدیل می‌شود و هر کشور در یک ردیف خاص قرار می‌گیرد. در نهایت، نام کشورهای تولید از دیکشنری‌ها استخراج شده و در ستون country_name ذخیره می‌شود. سپس فیلم‌هایی که در ایالات متحده آمریکا تولید نشده‌اند فیلتر می‌شوند و در non_us_movies ذخیره می‌شوند. این داده‌ها در مرحله بعد فیلتر می‌شوند تا فقط فیلم‌هایی که در ۱۰ سال گذشته منتشر شده‌اند نگه‌داری شوند. در نهایت، تحلیل هزینه تولید برای فیلم‌های غیرآمریکایی انجام می‌شود. برای هر سال و هر کشور، میانگین rt_production_budget محاسبه شده و در ستون average_production_budget ذخیره می‌شود. این تحلیل به بررسی هزینه‌های تولید فیلم در کشورهای مختلف و روند تغییرات آن در طی دهه گذشته کمک می‌کند.

نمودار:





نتیجه:

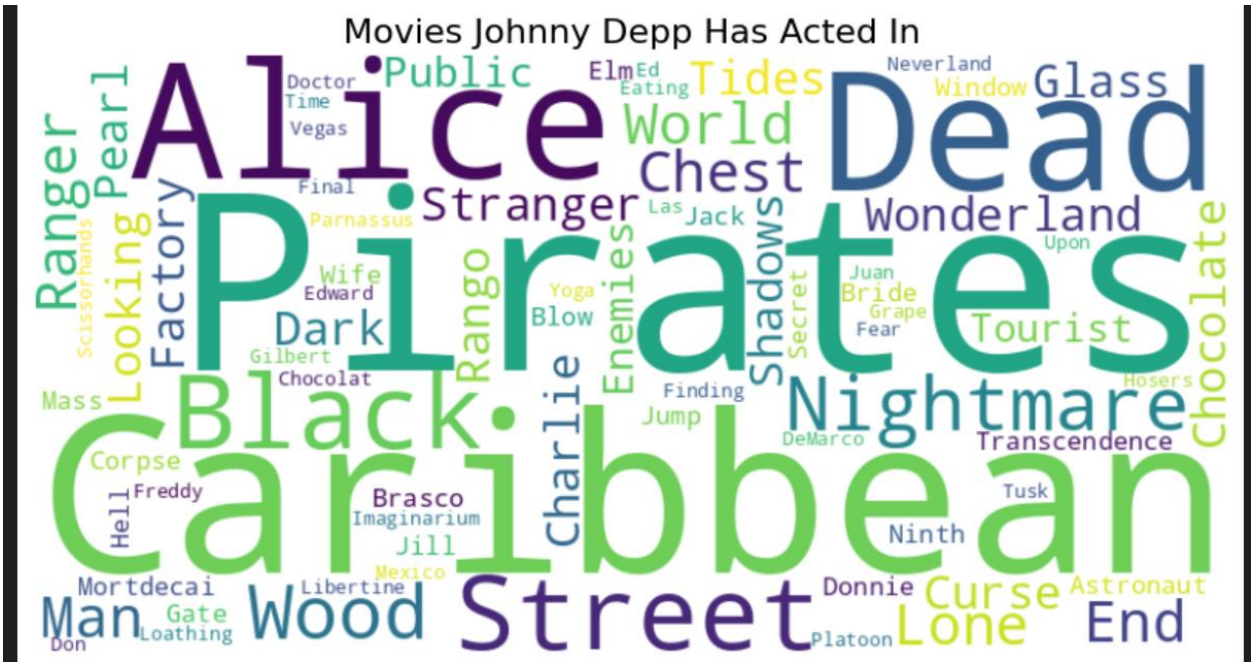
8) Johnny Depp در چه فیلم هایی بازی کرده است؟

ستون rt_actors که شامل اطلاعات بازیگران به صورت لیست دیکشنری است، با استفاده از تابع parse_list_column به لیست‌های واقعی تبدیل می‌شود.

بعد از آن، با استفاده از تابع apply، فیلتر می‌شود که آیا نام جانی دپ در هر کدام از دیکشنری‌های بازیگران موجود است یا خیر. این بررسی با استفاده از دستور any انجام می‌شود که از بین همه بازیگران هر فیلم، اگر حتی یک بازیگر با نام جانی دپ وجود داشته باشد، آن فیلم در نتیجه گنجانده می‌شود.

در نهایت، فقط ستون‌های rt_title (عنوان فیلم) و rt_movie_id (شناسه فیلم) انتخاب می‌شود و برای نمایش نمایش داده می‌شود.

نمودار:



9) به طور متوسط چند درصد نقش اول تا پنجم فیلم ها(به تفکیک برای هر نقش) مرد، و چند درصد زن هستند؟

ابتدا یک کپی از DataFrame اصلی (در اینجا df_credit) گرفته می‌شود تا پردازش‌ها بدون تغییر در داده‌های اصلی انجام شوند. در مرحله بعد، ستون rt_actors که شامل اطلاعات بازیگران به صورت رشته است، با استفاده از تابع parse_list_column تبدیل

به لیست‌های دیکشنری می‌شود. سپس، از متد explode برای تبدیل لیست‌ها به ردیف‌های جداگانه استفاده می‌شود، به طوری که هر بازیگر در یک ردیف جداگانه قرار می‌گیرد.

در ادامه، اطلاعات مربوط به جنسیت و ترتیب بازیگر از هر دیکشنری استخراج می‌شود و در دو ستون جدید gender و order ذخیره می‌شود. سپس، فقط بازیگران نقش‌های اصلی (با ترتیب کمتر از ۵) فیلتر می‌شوند و داده‌ها در DataFrame lead_roles ذخیره می‌شوند.

در نهایت، توزیع جنسیتی بازیگران در این نقش‌ها محاسبه می‌شود و در gender_distribution ذخیره می‌شود. این توزیع به صورت درصدی نشان‌دهنده سهم هر جنسیت از بازیگران نقش‌های اصلی است. برای خوانایی بیشتر، مقادیر جنسیتی به Female، Male و Unknown تبدیل می‌شوند.

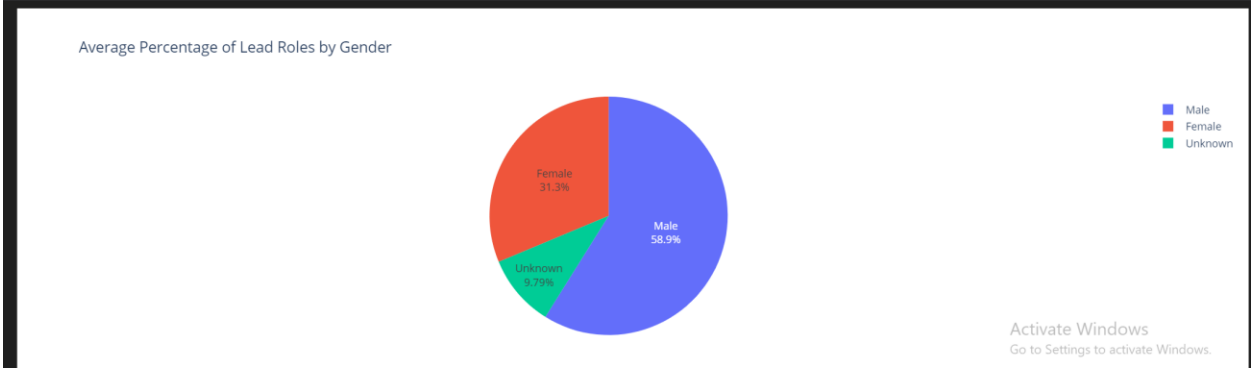
نمودار توزیع جنسیت در نقش اصلی:

```
Average Percentage of Lead Roles by Gender:
gender
Male       58.909232
Female     31.300499
Unknown     9.790270
Name: proportion, dtype: float64
```

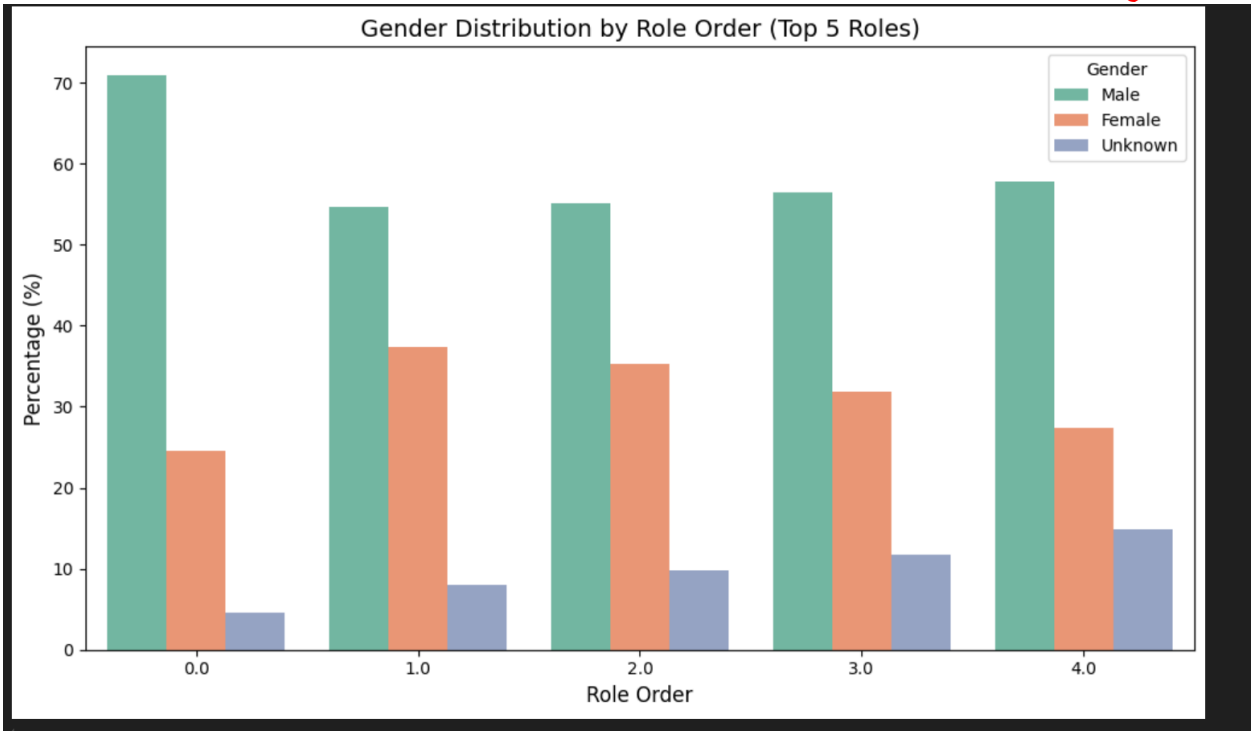
Plot

```
fig = px.pie(
    names=gender_distribution.index,
    values=gender_distribution.values,
    title='Average Percentage of Lead Roles by Gender',
    labels={'value': 'Percentage', 'index': 'Gender'}
)

fig.update_traces(textposition='inside', textinfo='percent+label')
fig.show()
```



نمودار توزیع جنسیت در 5 نقش اول:



نتیجه: بازیگران نقش اول فیلم ها اغلب مرد هستند.

11) محبوب ترین ژانرهای فیلم در 10 سال گذشته به چه ترتیب بوده است؟(یکبار بر اساس تعداد review و یکبار بر اساس critics_score مقایسه کنید)

ابتدا یک کپی از DataFrame اصلی (df_movies) ساخته می‌شود تا تغییرات بدون اثر بر داده‌های اصلی انجام شود. سپس، ستون rt_genres که شامل لیستی از ژانرها به صورت رشته است، با استفاده از ast.literal_eval به لیست‌های دیکشنری تبدیل می‌شود. بعد از آن، با استفاده از explode، لیست ژانرها به ردیف‌های جداگانه تبدیل می‌شود و هر ژانر در یک ردیف جدید قرار می‌گیرد.

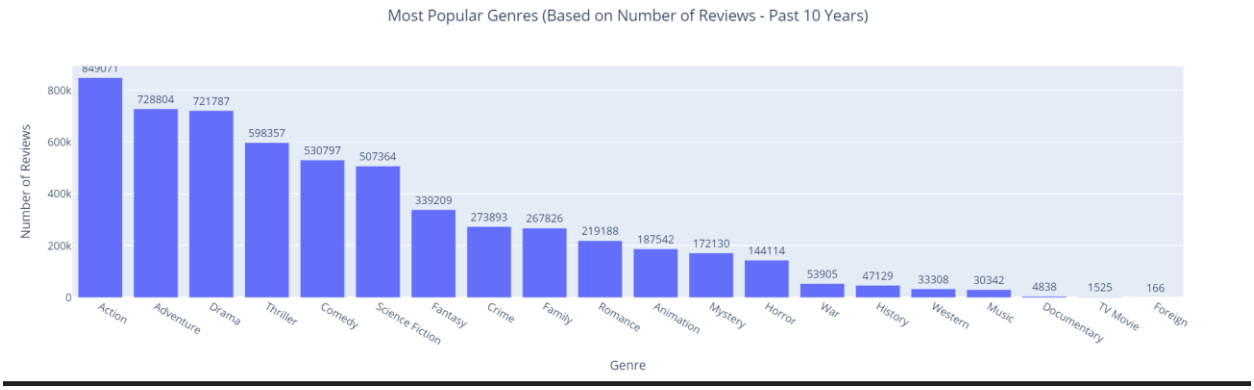
در مرحله بعد، نام ژانرها از دیکشنری‌ها استخراج شده و در ستون genre_name ذخیره می‌شود. همچنین، تاریخ انتشار فیلم‌ها در rt_release_date به فرمت تاریخ و زمان تبدیل شده و تنها سال انتشار (در ستون release_year) استخراج می‌شود. سپس، فیلم‌هایی که در بازه زمانی ۱۱ سال گذشته منتشر شده‌اند (بین سال‌های حداکثر سال انتشار تا یک سال پیش) فیلتر می‌شوند.

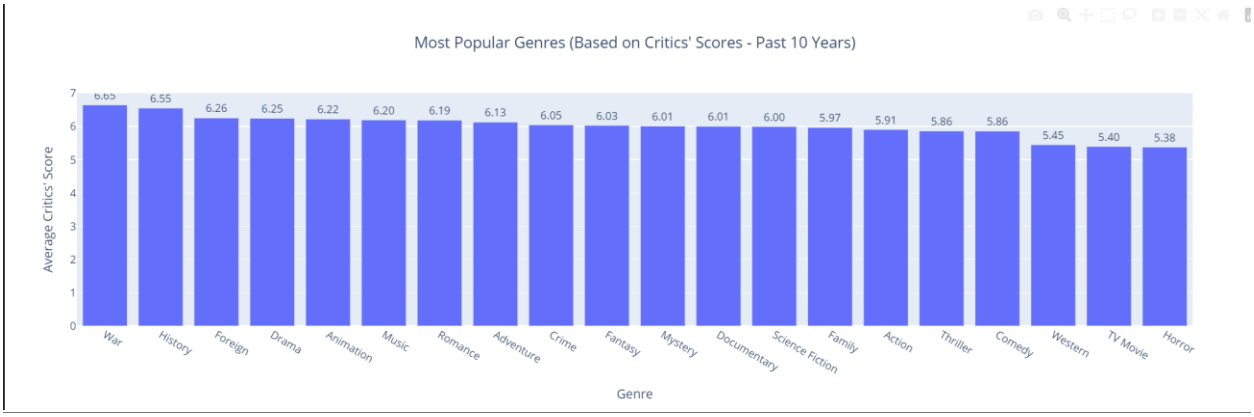
در نهایت، دو تحلیل انجام می‌شود:

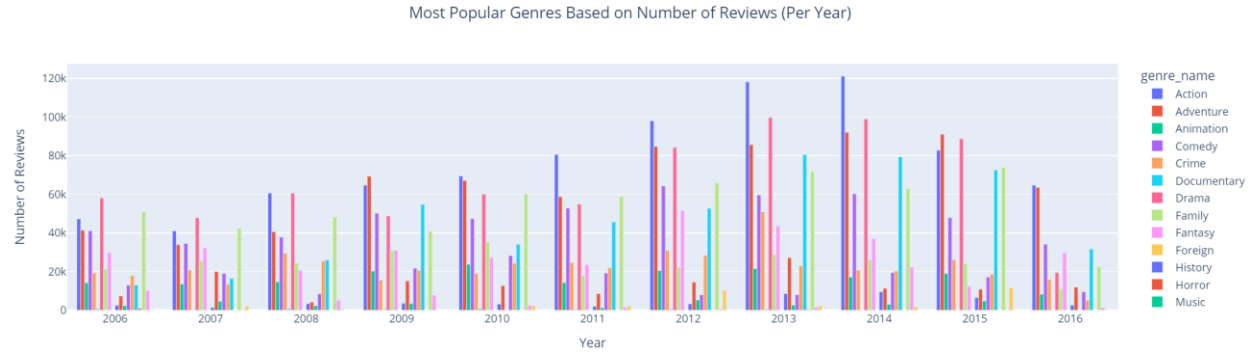
1) محبوبیت بر اساس تعداد نظرات کاربران: برای هر ژانر، مجموع rt_review_count محاسبه شده و بر اساس آن ژانرها مرتب می‌شوند.

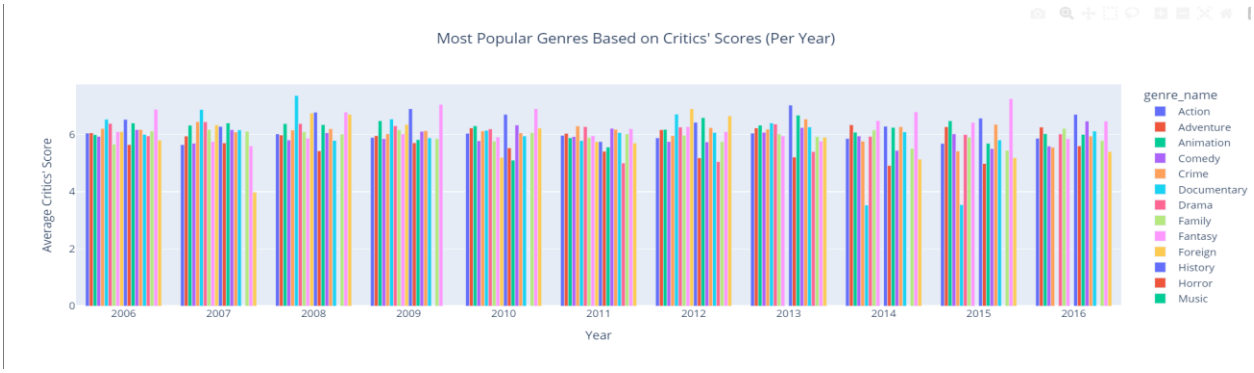
2) محبوبیت بر اساس امتیاز منتقدین :برای هر ژانر، میانگین rt_critics_score محاسبه شده و بر اساس آن ژانرها مرتب می‌شوند.

نمودار:


Most Popular Genres (Based on Number of Reviews - Past 10 Years)
Number of Reviews
800k
600k
400k
200k
0
849071
728804
721787
598357
530797
507364
339209
273893
267826
219188
187542
172130
144114
53905
47129
33308
30342
4838
1525
166
Action
Adventure
Drama
Thriller
Comedy
Science Fiction
Fantasy
Crime
Family
Romance
Animation
Mystery
Horror
War
History
Western
Music
Documentary
TV Movie
Foreign
Genre


Most Popular Genres (Based on Critics' Scores - Past 10 Years)
Average Critics' Score
7
6
5
4
3
2
1
0
6.65
6.55
6.26
6.25
6.22
6.20
6.19
6.13
6.05
6.03
6.01
6.01
6.00
5.97
5.91
5.86
5.86
5.45
5.40
5.38
War
History
Foreign
Drama
Animation
Music
Romance
Adventure
Crime
Fantasy
Mystery
Documentary
Science Fiction
Family
Action
Thriller
Comedy
Western
TV Movie
Horror
Genre


Most Popular Genres Based on Number of Reviews (Per Year)
Number of Reviews
120k
100k
80k
60k
40k
20k
0
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Year
genre_name
Action
Adventure
Animation
Comedy
Crime
Documentary
Drama
Family
Fantasy
Foreign
History
Horror
Music


Most Popular Genres Based on Critics' Scores (Per Year)
Average Critics' Score
6
4
2
0
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Year
genre_name
Action
Adventure
Animation
Comedy
Crime
Documentary
Drama
Family
Fantasy
Foreign
History
Horror
Music

توضیحات بخش دوم (پیش بینی)

1)بارگذاری و پیش‌پردازش داده‌ها

در این مرحله، داده‌ها برای مدل‌سازی به شکلی مناسب پردازش می‌شوند. این کار شامل حذف مقادیر گمشده، تبدیل داده‌ها به انواع مناسب و ایجاد ویژگی‌های جدید برای بهبود دقت مدل‌ها است.

1. **بارگذاری داده‌ها** :ابتدا داده‌ها از فایل CSV که حاوی اطلاعات مختلفی درباره فیلم‌ها است بارگذاری می‌شود. این داده‌ها شامل اطلاعاتی مثل بودجه تولید، درآمد، ژانر، زمان فیلم و تاریخ انتشار است.

2. **حذف مقادیر گمشده** :ردیف‌هایی که مقادیر مهم مانند **بودجه تولید، درآمد فیلم، ژانر فیلم** یا **زمان فیلم** را ندارند، از داده‌ها حذف می‌شوند. این کار برای اطمینان از این است که فقط داده‌های کامل برای پیش‌پردازش و مدل‌سازی استفاده شوند.

3. **تبدیل داده‌ها به نوع مناسب** :ستون‌های **بودجه تولید** و **درآمد فیلم** ممکن است حاوی مقادیر غیر عددی باشند (مثلاً رشته‌های متنی یا نمادهایی که به اشتباه وارد شده‌اند). بنابراین، این مقادیر به نوع عددی تبدیل می‌شوند تا از آن‌ها در مدل‌های رگرسیونی و پیش‌بینی استفاده شود. در صورتی که تبدیل به عدد ممکن نباشد (مثل مقادیر غیر عددی)، این مقادیر به NaNتبدیل می‌شوند.

4. **حذف ردیف‌هایی که مقادیر NaNدارند** :پس از تبدیل داده‌ها به مقادیر عددی، ممکن است برخی از ردیف‌ها همچنان حاوی مقادیر NaNبرای ستون‌های **بودجه تولید** یا **درآمد فیلم** باشند. در این مرحله، ردیف‌هایی که دارای مقادیر NaNدر این ستون‌ها هستند حذف می‌شوند.

5. **پردازش داده‌های ژانرها** :در این مرحله، ستون **ژانرهای فیلم** که به صورت رشته‌ای با فرمت JSON ذخیره شده‌اند، به یک لیست واقعی از ژانرها تبدیل می‌شود. این کار با استفاده از تابع ast.literal_evalانجام می‌شود که داده‌های رشته‌ای مشابه ساختار داده‌های پایتون را به نوع واقعی تبدیل می‌کند.

6. **استخراج ژانر اصلی از داده‌های ژانرها** :از لیست ژانرها، تنها اولین ژانر به عنوان **ژانر اصلی** استخراج می‌شود. این کار به مدل کمک می‌کند تا تنها یک ژانر از هر فیلم را به عنوان ویژگی استفاده کند، که می‌تواند تاثیر بیشتری در پیش‌بینی داشته باشد. اگر لیست ژانرها خالی باشد یا داده‌ها به درستی پردازش نشوند، مقدار 'Unknown' به ژانر اصلی اختصاص داده می‌شود.

7. **ایجاد ویژگی تعامل بین بودجه و ژانر** :در این بخش، ویژگی تعامل بین **بودجه تولید** و **ژانر اصلی** ایجاد می‌شود. برای مثال، اگر ژانر اصلی فیلم 'Action' باشد، مقدار این ویژگی برابر با **بودجه تولید** ضرب در 1 خواهد بود. برای سایر ژانرها، مقدار این ویژگی صفر خواهد بود. این ویژگی کمک می‌کند تا تأثیر خاص ژانر اکشن روی درآمد فیلم‌ها در مدل لحاظ شود.

8. **پیش‌نمایش داده‌ها پس از پیش‌پردازش** :در نهایت، داده‌های پیش‌پردازش‌شده به صورت خلاصه نمایش داده می‌شوند تا از صحت عملیات انجام شده اطمینان حاصل شود. داده‌هایی مثل عنوان فیلم، بودجه تولید، ژانر اصلی، ویژگی تعامل و درآمد فیلم برای اولین چند ردیف نمایش داده می‌شوند.

```python
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
import ast
import pandas as pd



movies_df = pd.read_csv('rotten_tomatoes_5000_movies.csv')

movies_df = movies_df.dropna(subset=['rt_production_budget', 'rt_box_office', 'rt_genres', 'rt_runtime'])

movies_df['rt_production_budget'] = pd.to_numeric(movies_df['rt_production_budget'], errors='coerce')
movies_df['rt_box_office'] = pd.to_numeric(movies_df['rt_box_office'], errors='coerce')
movies_df = movies_df.dropna(subset=['rt_production_budget', 'rt_box_office'])
movies_df['rt_genres'] = movies_df['rt_genres'].apply(ast.literal_eval)

#needed for featur engineering
movies_df['main_genre'] = movies_df['rt_genres'].apply(lambda x: x[0]['name'] if isinstance(x, list) and len(x) > 0 else 'Unknown')

movies_df['budget_genre_interaction'] = movies_df['rt_production_budget'] * movies_df['main_genre'].apply(lambda x: 1 if x == 'Action' else 0)  # Ex
#print data after preprocess
movies_df[['rt_title', 'rt_production_budget', 'main_genre', 'budget_genre_interaction', 'rt_box_office']].head()
```

	rt_title	rt_production_budget	main_genre	budget_genre_interaction	rt_box_office
0	Avatar	237000000	Action	237000000	2787965087
1	Pirates of the Caribbean: At World's End	300000000	Adventure	0	961000000
2	Spectre	245000000	Action	245000000	880674609
3	The Dark Knight Rises	250000000	Action	250000000	1084939099
4	John Carter	260000000	Action	260000000	284139100

2)ادغام داده ها

در این بخش، داده‌های مربوط به فیلم‌ها و داده‌های مربوط به بازیگران و کارگردانان بر اساس شناسه فیلم (rt_movie_id) به هم متصل می‌شوند. سپس ردیف‌هایی که درآمد فیلم آن‌ها برابر با صفر است، حذف می‌شوند.

این دو دیتاست با استفاده از ستون rt_movie_idبه هم متصل می‌شوند. این عمل پیوند دو جدول به یکدیگر است تا داده‌های مختلف مربوط به فیلم‌ها (از جمله اطلاعات کارگردان و بازیگران) به هم مرتبط شوند.

3)پردازش داده‌ها

در این بخش، مجموعه‌ای از عملیات پردازش داده‌ها به منظور آماده‌سازی داده‌ها برای مدل‌سازی انجام می‌شود:

1. حذف ردیف‌های با درآمد صفر :این مرحله شامل حذف فیلم‌هایی است که درآمد نداشته‌اند (یعنی مقدار rt_box_office صفر است). این کار باعث می‌شود که داده‌های نامعتبر از فرآیند پیش‌بینی کنار گذاشته شوند.
2. پردازش ویژگی‌ها:(Feature Engineering)
   - پردازش ژانرها :ژانرهای فیلم‌ها به صورت رشته‌ای در دیتاست ذخیره شده‌اند. در اینجا، با استفاده از تابع parse_genres، ژانرها به لیستی از ژانرهای قابل فهم تبدیل می‌شوند.
   - پردازش بازیگران و کارگردان‌ها :برای استخراج اسامی بازیگران و کارگردان‌ها از داده‌ها، از توابع parse_actorsو parse_directorsاستفاده شده است.
   - ایجاد ویژگی‌های ترکیبی :برای هر فیلم، ویژگی‌هایی مانند ترکیب ژانرها (که شامل ترکیب تمام ژانرهای یک فیلم است) و تعداد بازیگران و کارگردان‌های مشهور (با استفاده از تعداد بازیگران و کارگردان‌های برتر) به ویژگی‌های اضافی اضافه می‌شود.
   - ویژگی‌های مربوط به زمان و بودجه :ویژگی‌های جدیدی مانند log_budget(لگاریتمی از بودجه فیلم) و log_runtime(لگاریتمی از زمان اجرای فیلم) ایجاد می‌شوند تا به مدل کمک کنند تا به طور موثرتر با مقادیر بسیار بزرگ یا کوچک کار کند.
   - ویژگی‌های تعامل :برخی ویژگی‌های تعامل مانند تعامل امتیاز منتقدان و مخاطبان (مجموع امتیازها از دو دسته منتقدان و مخاطبان) و تعامل بودجه و زمان اجرا ایجاد می‌شوند. این ویژگی‌ها می‌توانند نشان دهند که چگونه ترکیب این عوامل ممکن است بر درآمد فیلم تاثیر بگذارد.
3. حذف مقادیر گمشده :ردیف‌هایی که مقادیر گمشده در ویژگی‌های کلیدی مانند rt_box_office، budget_per_minute، و rt_critics_scoreدارند حذف می‌شوند تا مدل به درستی آموزش ببیند.
4. **ویژگی‌های هدف و ورودی:**
   - ویژگی‌های هدف (متغیر yبه صورت لگاریتمی از درآمد فیلم‌ها محاسبه می‌شوند) برای تبدیل داده‌ها به مقیاس لگاریتمی) تا اثرات مقادیر بزرگ درآمدها کاهش یابد.
   - ویژگی‌های ورودی (X) شامل مجموعه‌ای از ویژگی‌های عددی و دسته‌بندی است که شامل بودجه، زمان اجرا، امتیاز منتقدان، تعداد بازیگران و کارگردان‌های مشهور، و بسیاری ویژگی‌های دیگر هستند.

4) پیش‌پردازش داده‌ها

در این مرحله، پیش‌پردازش داده‌ها برای تطبیق با مدل‌های یادگیری ماشین انجام می‌شود:

1. ویژگی‌های عددی :
   - ایمپرشن مقادیر گمشده :از SimpleImputerبرای جایگزینی مقادیر گمشده با میانگین استفاده می‌شود.

- مقیاس‌بندی داده‌ها :از StandardScalerبرای مقیاس‌بندی داده‌های عددی استفاده می‌شود تا همه ویژگی‌ها در یک مقیاس مشابه قرار گیرند.

2. ویژگی‌های دسته‌بندی :

- ایمپرشن مقادیر گمشده :ویژگی‌های دسته‌بندی مانند ترکیب ژانرها و دهه انتشار که مقادیر گمشده دارند، با استفاده از SimpleImputerجایگزین می‌شوند.

- One-Hot Encoding:ویژگی‌های دسته‌بندی با استفاده از OneHotEncoderبه ویژگی‌های عددی تبدیل می‌شوند تا بتوانند در مدل‌های یادگیری ماشین استفاده شوند.

این پیش‌پردازش‌ها به طور خودکار با استفاده از ColumnTransformerانجام می‌شود، به طوری که ویژگی‌های عددی و دسته‌بندی به صورت جداگانه و مستقل پردازش می‌شوند.

5) آموزش مدل‌ها

در این بخش، از چندین مدل یادگیری ماشین مختلف برای پیش‌بینی درآمد فیلم‌ها استفاده می‌شود:

1. Random Forest Regressor:این مدل از ترکیب درختان تصمیم برای انجام پیش‌بینی استفاده می‌کند. مزیت آن در قابلیت به دست آوردن تصمیمات پیچیده و مدیریت داده‌های پرسر و صدا است.

2. Gradient Boosting Regressor:مدل Gradient Boosting یک روش مبتنی بر تقویت مدل‌های ضعیف است که از چندین مدل تصمیم ساده برای ساخت یک مدل قوی استفاده می‌کند.

3. XGBoost Regressor:این مدل مشابه Gradient Boosting است، اما با بهینه‌سازی‌هایی که عملکرد آن را به شدت بهبود می‌بخشد.

4. Linear Regression: رگرسیون خطی به عنوان یک مدل ساده برای پیش‌بینی درآمد فیلم‌ها با استفاده از روابط خطی میان ویژگی‌ها استفاده می‌شود.

در این مرحله، از K-Fold Cross Validationبرای آموزش مدل‌ها و ارزیابی نتایج استفاده می‌شود. در این فرآیند، داده‌ها به K بخش تقسیم می‌شوند و مدل‌ها بر روی K-1 بخش آموزش داده شده و بر روی بخش باقی‌مانده تست می‌شوند.

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)
for model_name, model in models:
    pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('model', model)])
    model_mse, model_r2 = [], []
    for train_idx, test_idx in kf.split(X):
        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
        y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
        pipeline.fit(X_train, y_train)
        y_pred = pipeline.predict(X_test)
        model_mse.append(mean_squared_error(y_test, y_pred))
        model_r2.append(r2_score(y_test, y_pred))
    results.append((model_name, np.mean(model_mse), np.mean(model_r2)))

# **4. Model Comparison**
results_df = pd.DataFrame(results, columns=['Model', 'MSE', 'R²'])
print(results_df)
```

6) مقایسه مدل‌ها

پس از آموزش مدل‌ها، عملکرد آن‌ها با استفاده از معیارهای Mean Squared Error (MSE) و R^2 ارزیابی می‌شود. این نتایج به شکل یک جدول ذخیره شده و برای مقایسه مدل‌ها استفاده می‌شوند. مدل‌هایی که کمترین مقدار MSE و بیشترین مقدار R^2 دارند به عنوان مدل‌های بهتری انتخاب می‌شوند.

7)تنظیم پارامترهای مدل بهترین

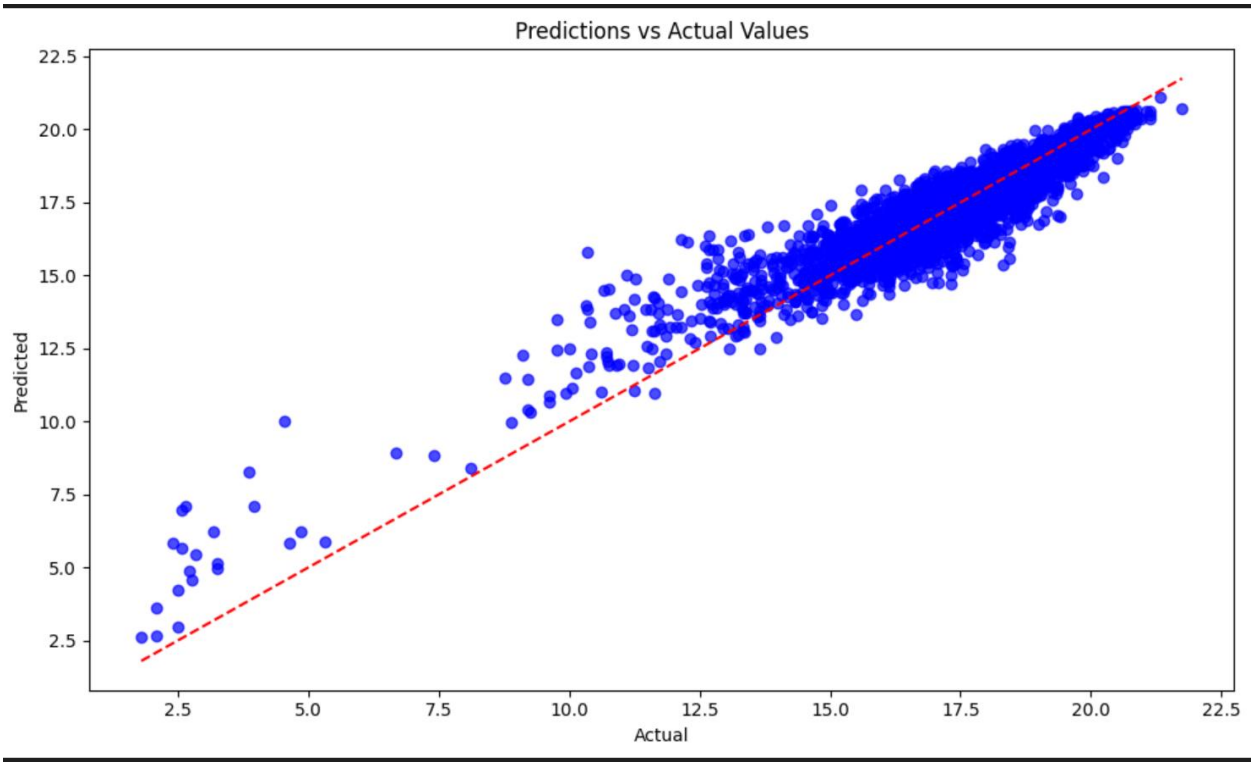
برای مدل انتخابی (مدلی که بهترین عملکرد را از نظر R^2 دارد)، از GridSearchCV استفاده می‌شود تا بهترین هایپرپارامترها شناسایی شوند. این پارامترها شامل تعداد درختان، عمق درخت، نرخ یادگیری و پارامترهای دیگر هستند که می‌توانند عملکرد مدل را بهبود دهند.

8)اهمیت ویژگی‌ها

در این بخش، از Feature Importance برای شناسایی ویژگی‌هایی که بیشترین تاثیر را در پیش‌بینی درآمد فیلم‌ها دارند استفاده می‌شود. این ویژگی‌ها می‌توانند کمک کنند تا ببینیم کدام عوامل (مانند بودجه، امتیاز منتقدان، تعداد بازیگران مشهور) بیشترین تاثیر را بر موفقیت تجاری فیلم‌ها دارند.

9)مقایسه پیش‌بینی‌ها با مقادیر واقعی

در این مرحله، پیش‌بینی‌ها با مقادیر واقعی درآمد فیلم‌ها مقایسه می‌شوند. یک نمودار پراکندگی برای مشاهده این مقایسه ترسیم می‌شود. اگر مدل پیش‌بینی دقیقی انجام داده باشد، نقاط در این نمودار باید در نزدیکی خط رگرسیون (که از مقادیر واقعی و پیش‌بینی‌شده تشکیل شده است) قرار بگیرند. که در نتایج نموداری ما هم این اتفاق افتاده است.



توضیح هایپرپارامترها و نحوه تنظیم و بهینه کردن آنها:

GridSearchCV

این تکنیک جستجوی دقیق و کامل را برای تنظیم بهترین هایپرپارامترها انجام می‌دهد. به این معنی که تمامی ترکیب‌های ممکن از مقادیر مختلف هایپرپارامترها را بررسی می‌کند و بهترین مدل را بر اساس ارزیابی عملکرد (مثلاً Mean Squared Error (MSE) یا R^2) انتخاب می‌کند.

در این پروژه از GridSearchCVبرای تنظیم هایپرپارامترها استفاده شده است. به طور خاص، بعد از انتخاب مدل بهترین (مدلی که بهترین عملکرد را از نظر R^2 داشت)، GridSearchCVبه دنبال مقادیر بهینه هایپرپارامترها می‌گردد.

در اینجا، پارامترهای مختلف برای هر مدل به صورت یک دیکشنری تعریف شده‌اند. برای هر مدل، مقادیر مختلفی برای هایپرپارامترها پیشنهاد شده است، که عبارتند از:

هایپرپارامترهای مورد استفاده برای مدل‌ها

1. Random Forest:
   - model__n_estimators: تعداد درختان در جنگل تصادفی
   - model__max_depth: عمق درخت‌ها
   - model__min_samples_split: حداقل تعداد نمونه‌ها برای تقسیم یک گره درخت،
2. Gradient Boosting:
   - model__n_estimators: تعداد درختان

- model__learning_rate: نرخ یادگیری
- model__max_depth: عمق درخت‌ها

3. XGBoost:

- model__n_estimators: تعداد درختان
- model__max_depth: عمق درخت‌ها
- model__learning_rate: نرخ یادگیری
- model__min_child_weight: وزن حداقل برای تقسیم یک گره

استفاده از GridSearchCV

برای جستجوی بهترین هایپرپارامترها، ابتدا یک پایپ‌لاین (Pipeline)ساخته می‌شود که شامل مراحل پیش‌پردازش داده‌ها با استفاده از ColumnTransformerو مدل مورد نظر است. سپس با استفاده از GridSearchCV، تمام ترکیب‌های ممکن از هایپرپارامترها آزمایش می‌شود.

- cv=kf: این پارامتر مشخص می‌کند که از K-Fold Cross Validationبرای ارزیابی مدل استفاده شود.
- scoring='neg_mean_squared_error': معیار ارزیابی برای انتخاب بهترین مدل است. در اینجا از MSEمنفی استفاده می‌شود تا بهترین مدل از نظر کمترین خطا انتخاب شود.
- param_grids[best_model_name]:این بخش هایپرپارامترهای مربوط به مدل بهترین را از دیکشنری param_gridsبارگذاری می‌کند.

پس از انجام جستجو، بهترین ترکیب هایپرپارامترها از طریق grid_search.best_params_ به دست می‌آید.

RandomizedSearchCV

یک روش دیگر برای تنظیم هایپرپارامترها، RandomizedSearchCVاست. در این تکنیک، به جای بررسی همه ترکیب‌های ممکن، تعدادی ترکیب تصادفی از هایپرپارامترها انتخاب می‌شود و مدل‌ها بر اساس آن‌ها ارزیابی می‌شوند. این روش معمولاً سریع‌تر از GridSearchCVاست، زیرا تمام فضای جستجو را پوشش نمی‌دهد.

با استفاده از RandomizedSearchCVمی‌توان به سرعت هایپرپارامترهایی که بیشترین تاثیر را دارند شناسایی کرد. اما برای پیش بینی این پروژه سرعت برای ما مطرح نبود و دقت مهم بود.

تنظیم هایپرپارامترها در این پروژه

در این پروژه، تنظیم هایپرپارامترها با استفاده از GridSearchCVبرای مدل‌هایی که بهترین عملکرد را دارند انجام می‌شود. پس از انجام این تنظیمات، مدل بهینه (با بهترین هایپرپارامترها) انتخاب می‌شود.

ارزیابی بهترین مدل

پس از انجام جستجو و پیدا کردن بهترین هایپرپارامترها، مدل با استفاده از این تنظیمات بهترین عملکرد را بر روی داده‌ها ارائه می‌دهد که توسط grid_search.best_estimator_انتخاب می‌شود.best model مدل نهایی است که برای پیش‌بینی‌ها استفاده می‌شود.

چرا تنظیم هایپرپارامترها مهم است؟

هایپرپارامترها نقش مهمی در عملکرد نهایی مدل دارند. تنظیم درست این پارامترها می‌تواند منجر به بهبود دقت مدل و جلوگیری از overfitting(پراکندگی زیاد مدل) یا underfitting(کاهش دقت مدل) شود. به همین دلیل، آزمایش‌های متعددی روی این پارامترها انجام می‌شود تا بهترین نتیجه حاصل گردد.

بر اساس نتایج R^2، بهترین مدل انتخابی Gradient Boostingبا R^2معادل 0.537067 است. این مدل از همه مدل‌ها عملکرد بهتری داشته است، به همین دلیل، در مرحله بعدی تنظیمات هایپرپارامترها برای این مدل انجام شده است.

تنظیم هایپرپارامترها(Best Parameters)

با استفاده از GridSearchCV، بهترین تنظیمات هایپرپارامترها برای مدل Gradient Boostingبه شرح زیر به دست آمده است:

- model__learning_rate: 0.1:نرخ یادگیری تنظیم شده به 0.1، که تأثیر زیادی در میزان تغییرات وزن‌ها در طول آموزش دارد.
- model__max_depth: 5:عمق درخت‌های تصمیم 5 تعیین شده است، که نشان می‌دهد درخت‌ها نباید خیلی عمیق شوند تا از overfittingجلوگیری شود.
- model__n_estimators: 200:تعداد درخت‌ها 200 است. این مقدار به تعداد مدل‌های ضعیف که باید با یکدیگر ترکیب شوند، اشاره دارد.

این تنظیمات هایپرپارامترها باعث بهبود عملکرد مدل Gradient Boostingشده است.

ارزیابی نهایی مدل

پس از تنظیم هایپرپارامترها، مدل نهایی دارای R^2معادل 0.8653454261617192است. این مقدار به طور قابل توجهی بالاتر از سایر مدل‌ها است، نشان‌دهنده این است که مدل نهایی توانسته است عملکرد بسیار بهتری داشته باشد و به وضوح از مدل‌های قبلی بهتر عمل می‌کند.

تحلیل نهایی:

- Gradient Boostingبا هایپرپارامترهای بهینه عملکرد عالی داشته است و توانسته است R^2بالای 0.86 را در پیش‌بینی درآمد فیلم‌ها کسب کند.
- Random Forestو XGBoostنیز عملکرد نسبتاً خوبی داشتند، اما Gradient Boostingبرتری محسوسی نسبت به آن‌ها داشت.
- رگرسیون خطی نتایج ضعیفی را ارائه داده است و این نشان می‌دهد که مدل‌های پیچیده‌تر) مانند (Gradient Boosting برای این نوع پیش‌بینی‌ها مناسب‌تر هستند.

برای انتخاب بهترین مدل بر اساس دقت و قابلیت تعمیم (generalization ability)، باید به دو نکته توجه کنیم:

1. دقت مدل در داده‌های آموزشی :این معیار نشان می‌دهد که مدل چقدر در پیش‌بینی داده‌هایی که قبلاً با آن‌ها آموزش دیده است، دقت دارد. اگر مدل در داده‌های آموزشی عملکرد خوبی داشته باشد اما در داده‌های تست یا جدید عملکرد ضعیفی نشان دهد، ممکن است مدل دچار overfitting(هم‌پوشانی بیش از حد با داده‌های آموزشی) شده باشد.
2. قابلیت تعمیم مدل :این معیار نشان‌دهنده توانایی مدل در پیش‌بینی داده‌های جدید است که در زمان آموزش مشاهده نکرده است. این ویژگی برای سنجش قدرت واقعی مدل در مواجهه با داده‌های ناشناخته حیاتی است. اگر مدل در داده‌های تست یا ارزیابی عملکرد خوبی نشان دهد، به این معنی است که مدل قابلیت تعمیم خوبی دارد.

از نتایج موجود، ما به موارد زیر توجه می‌کنیم:

- R^2(ضریب تعیین) و MSE(میانگین مربع خطا) می‌توانند به ما کمک کنند تا دقت و قابلیت تعمیم مدل‌ها را ارزیابی کنیم.
- مدل‌هایی که عملکرد بهتر در مجموعه تست دارند (با MSEکمتر و R^2بالاتر در داده‌های تست) معمولاً مدل‌هایی هستند که قابلیت تعمیم بهتری دارند.

تجزیه و تحلیل مدل‌های مختلف:

1. Random Forest:
   - MSE: 2.311379
   - R^2: 0.520856
   - این مدل دارای دقت متوسط است و عملکرد قابل قبولی دارد، اما نمی‌توان گفت بهترین مدل برای تعمیم است.
2. Gradient Boosting:
   - MSE: 2.230421
   - R^2: 0.537067
   - Gradient Boostingعملکرد خوبی با R^2بالاتر از مدل Random Forestدارد، و این نشان می‌دهد که مدل توانسته است به خوبی داده‌ها را یاد بگیرد و به احتمال زیاد از overfittingکمتری برخوردار است.
3. XGBoost:
   - MSE: 2.319859
   - R^2: 0.516712
   - XGBoostکمی ضعیف‌تر از Gradient Boostingاست اما همچنان عملکرد نسبتاً خوبی دارد.
4. Linear Regression:
   - MSE: 3.084080
   - R^2: 0.360813
   - این مدل ضعیف‌ترین عملکرد را دارد R^2 .کم و MSEبالا نشان می‌دهد که مدل نمی‌تواند به خوبی داده‌ها را مدل‌سازی کند و در پیش‌بینی درآمد فیلم‌ها دقت کمی دارد.

مدل بهترین از نظر دقت و قابلیت تعمیم:

- Gradient Boostingبه نظر می‌رسد که بهترین مدل از نظر دقت و قابلیت تعمیم باشد. این مدل توانسته است عملکرد خوبی در مجموعه‌های آموزشی و تست داشته باشد و بالاترین R^2را در داده‌های تست کسب کرده است. همچنین، مقدار MSEآن نیز نسبت به سایر مدل‌ها کمتر است که نشان‌دهنده دقت بالاتر و قابلیت تعمیم بهتر است.

- XGBoostنیز مدل خوبی است اما به دلیل مقدار R^2کمی پایین‌تر از Gradient Boosting، از نظر دقت و قابلیت تعمیم نمی‌تواند به عنوان بهترین مدل انتخاب شود.

نتیجه‌گیری:

بر اساس نتایج R^2و MSE، بهترین مدل از نظر دقت و قابلیت تعمیم در پیش‌بینی درآمد فیلم‌ها Gradient Boostingاست. این مدل توانسته است به بهترین شکل داده‌ها را یاد بگیرد و توانایی خوبی در تعمیم به داده‌های جدید و ناشناخته داشته باشد.

عوامل کلیدی موثر بر درآمد فیلم ها

با توجه به تحلیل‌های انجام شده، عوامل کلیدی تاثیرگذار بر درآمد فیلم‌ها شامل ترکیب ژانرها، امتیاز منتقدان و مخاطبان، بودجه تولید، مدت زمان فیلم، تعداد نقدها و نقش بازیگران و کارگردان‌های مشهور هستند. این ویژگی‌ها می‌توانند به طور قابل توجهی بر موفقیت تجاری فیلم‌ها تاثیر بگذارند و به مدل‌های یادگیری ماشین کمک کنند تا درآمد فیلم‌ها را با دقت بیشتری پیش‌بینی کنند.

.1ترکیب ژانرها(Genre Combination)

در مدل شما، ترکیب ژانرها یکی از مهم‌ترین ویژگی‌ها برای پیش‌بینی درآمد فیلم‌ها است. ترکیب ژانرهایی که بیشتر مورد توجه مخاطبان قرار می‌گیرند، مانند اکشن، کمدی، درام و تخیلی، معمولاً با درآمد بالاتری همراه است.

- ژانرهای پرطرفدار مانند اکشن و کمدی توانایی جذب مخاطب بیشتری دارند.
- ترکیب‌های ژانری مانند اکشن و درام یا تخیلی و علمی می‌توانند به جذب مخاطب‌های خاص و گسترش بازار کمک کنند.

.2امتیاز منتقدان و مخاطبان(Critic and Audience Scores)

امتیاز منتقدان و امتیاز مخاطبان نقش بسیار مهمی در پیش‌بینی درآمد فیلم‌ها دارند. این دو ویژگی مستقیماً با محبوبیت فیلم‌ها مرتبط هستند.

- امتیاز منتقدان (rt_critics_score)معمولاً بر روی فیلم‌هایی که نقدهای مثبت دریافت می‌کنند تاثیرگذار است و موجب افزایش توجه عمومی به آن‌ها می‌شود.
- امتیاز مخاطبان (rt_audience_score)نیز نشان‌دهنده رضایت عمومی از فیلم است. فیلم‌هایی که امتیاز بالاتری از مخاطبان دریافت می‌کنند، احتمالاً فروش بالاتری خواهند داشت.

.3بودجه تولید فیلم(Budget)

بودجه فیلم یکی از عوامل اصلی در پیش‌بینی درآمد فیلم‌ها است. فیلم‌هایی با بودجه بالاتر معمولاً امکان بیشتری برای تبلیغات و تولید باکیفیت دارند که این امر موجب جذب بیشتر مخاطب و در نتیجه درآمد بیشتر می‌شود.

- بودجه بالا اغلب به معنی توانایی در جذب ستارگان معروف، تبلیغات گسترده و ساخت فیلم‌های با کیفیت بالاتر است که می‌تواند مخاطب بیشتری جذب کند.
- بودجه‌های کلان معمولاً با بازاریابی قدرتمندتر و توزیع گسترده‌تر فیلم در بازارهای مختلف همراه است.

.4مدت زمان فیلم(Runtime)

مدت زمان فیلم تاثیر قابل توجهی بر درآمد آن دارد. فیلم‌هایی که زمان بیشتری دارند، معمولاً بر مخاطبان با حوصله‌تر و یا در ژانرهای خاص که نیاز به توسعه داستان دارند، تاثیرگذارترند.

- فیلم‌هایی که زمان طولانی دارند ممکن است به دلیل پیشرفت داستان و جذب بیشتر مخاطب، درآمد بالاتری داشته باشند.
- در عین حال، باید توجه داشت که در برخی از ژانرها یا بازارها، مدت زمان کوتاه‌تر می‌تواند جذاب‌تر باشد و مخاطب بیشتری جذب کند.

.5تعداد نقدها(Review Count)

تعداد نقدهای منتشر شده و بازخوردهای منتقدان تاثیر زیادی بر میزان توجه به یک فیلم دارند. فیلم‌هایی که نقدهای بیشتری دریافت می‌کنند، معمولاً بیشتر در معرض دید قرار می‌گیرند و احتمالاً درآمد بیشتری خواهند داشت.

- تعداد نقدها می‌تواند نشان‌دهنده محبوبیت یک فیلم باشد. هرچه نقدهای بیشتری در مورد یک فیلم منتشر شود، آن فیلم بیشتر دیده می‌شود و احتمال فروش آن افزایش می‌یابد.

.6تعداد بازیگران و کارگردان‌های معروف(Famous Actors and Directors)

بازیگران مشهور و کارگردان‌های معروف در جذب مخاطب تاثیر زیادی دارند. فیلم‌هایی که توسط کارگردانان برجسته ساخته شده یا بازیگران معروف در آن‌ها نقش دارند، معمولاً از فروش بالاتری برخوردارند.

- بازیگران مشهور می‌توانند توجه مخاطبان را جلب کنند، در حالی که کارگردان‌های معروف به فیلم اعتبار می‌دهند و باعث افزایش علاقه‌مندی به آن می‌شوند.

.7سال انتشار فیلم(Release Year)

سال انتشار فیلم نیز بر درآمد آن تاثیر می‌گذارد. فیلم‌های منتشر شده در سال‌های اخیر معمولاً به دلیل استفاده از فناوری‌های جدیدتر، داستان‌های به‌روزتر و ترندهای جدید، درآمد بیشتری دارند.

- سال انتشار می‌تواند نشان‌دهنده تاثیر زمان و تغییرات فرهنگی و اقتصادی در سینما باشد.

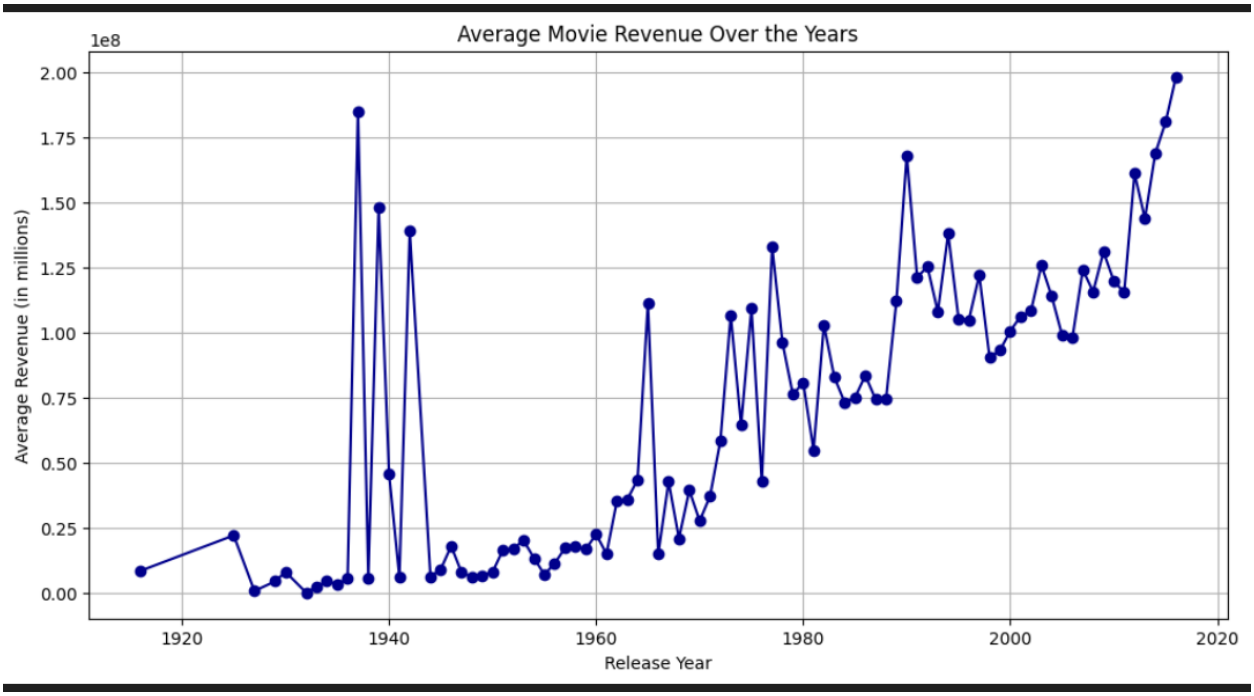
.8تعاملات بین ویژگی‌ها(Interaction Features)

تعاملات بین برخی ویژگی‌ها نیز تاثیر زیادی بر درآمد فیلم‌ها دارند. برخی از ویژگی‌هایی که به‌طور مشترک تاثیر بیشتری دارند عبارتند از:

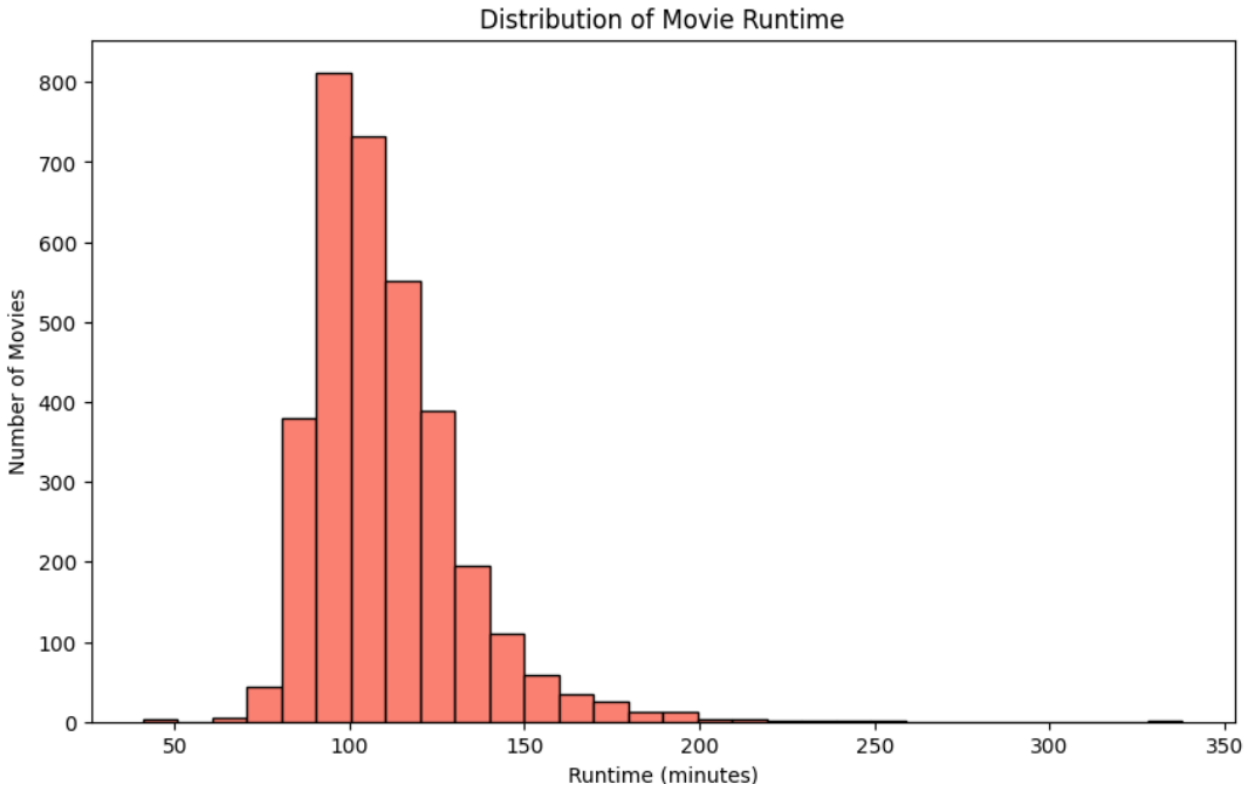
- تعامل بین بودجه و زمان اجرا :(budget_runtime_interaction)فیلم‌هایی که هم بودجه بالایی دارند و هم زمان بیشتری برای تولید صرف کرده‌اند، معمولاً فروش بیشتری دارند.
- تعامل بین امتیاز منتقدان و مخاطبان :(critic_audience_interaction)این ویژگی نشان می‌دهد که ترکیب امتیازهای مثبت از منتقدان و مخاطبان می‌تواند تاثیر مضاعفی در پیش‌بینی درآمد داشته باشد.

نمودار ها)

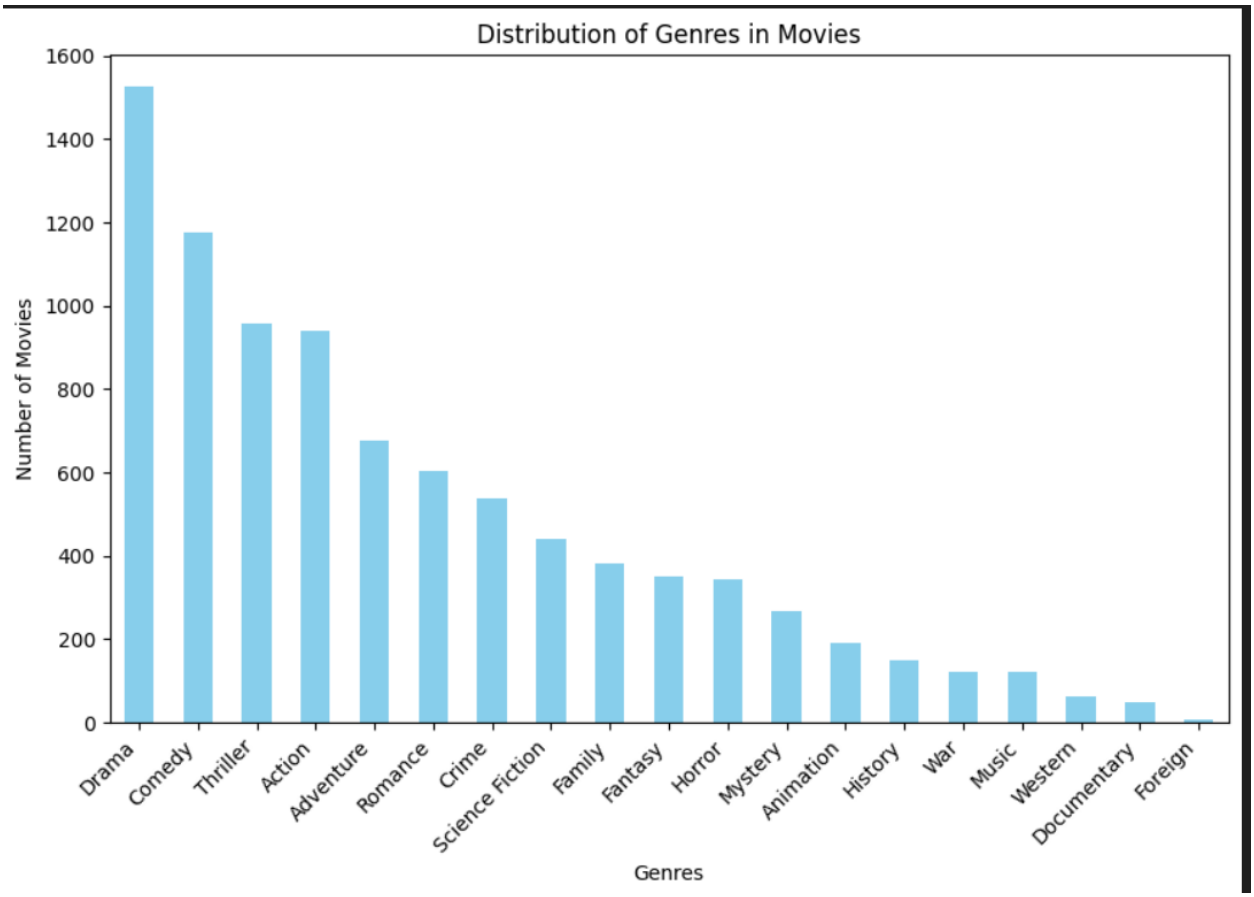
توضیح :این نمودار به ما نشان می‌دهد که متوسط درآمد فیلم‌ها در طول سال‌ها چگونه تغییر کرده است و آیا روند مشخصی در افزایش یا کاهش درآمد وجود دارد یا خیر.

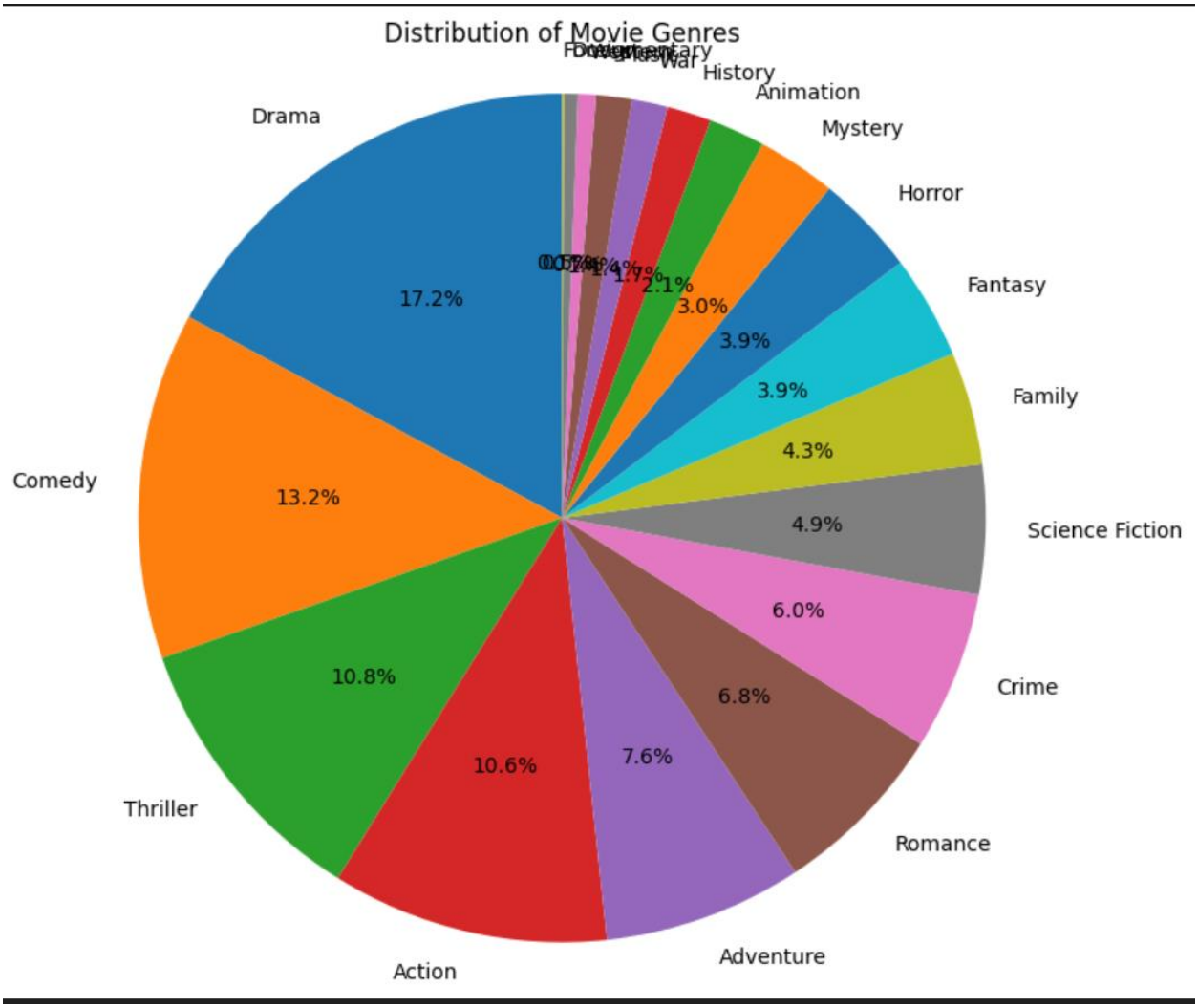


توضیح) این نمودار نشان‌دهنده این است که فیلم‌ها معمولاً چه مدت زمانی دارند و بیشتر فیلم‌ها در چه رنج زمانی قرار دارند.



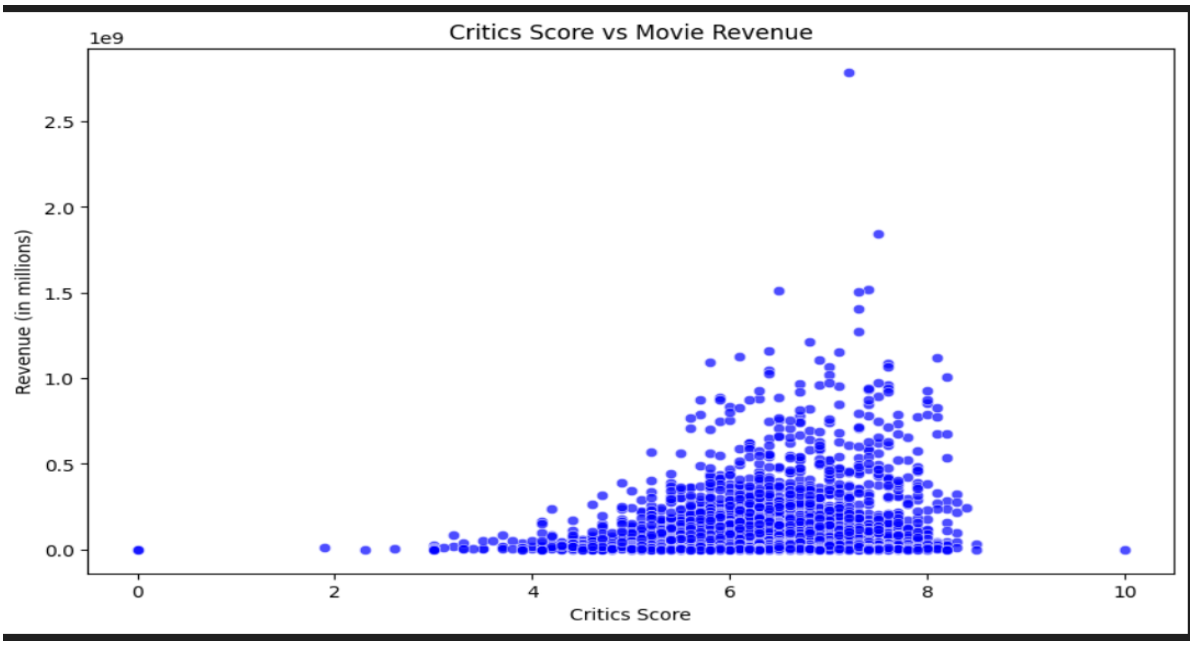
توضیح) نمودارهای زیر نشان می‌دهد که هر ژانر فیلم چقدر در دیتاست پراکنده شده است و کدام ژانر بیشترین تعداد فیلم را دارد.

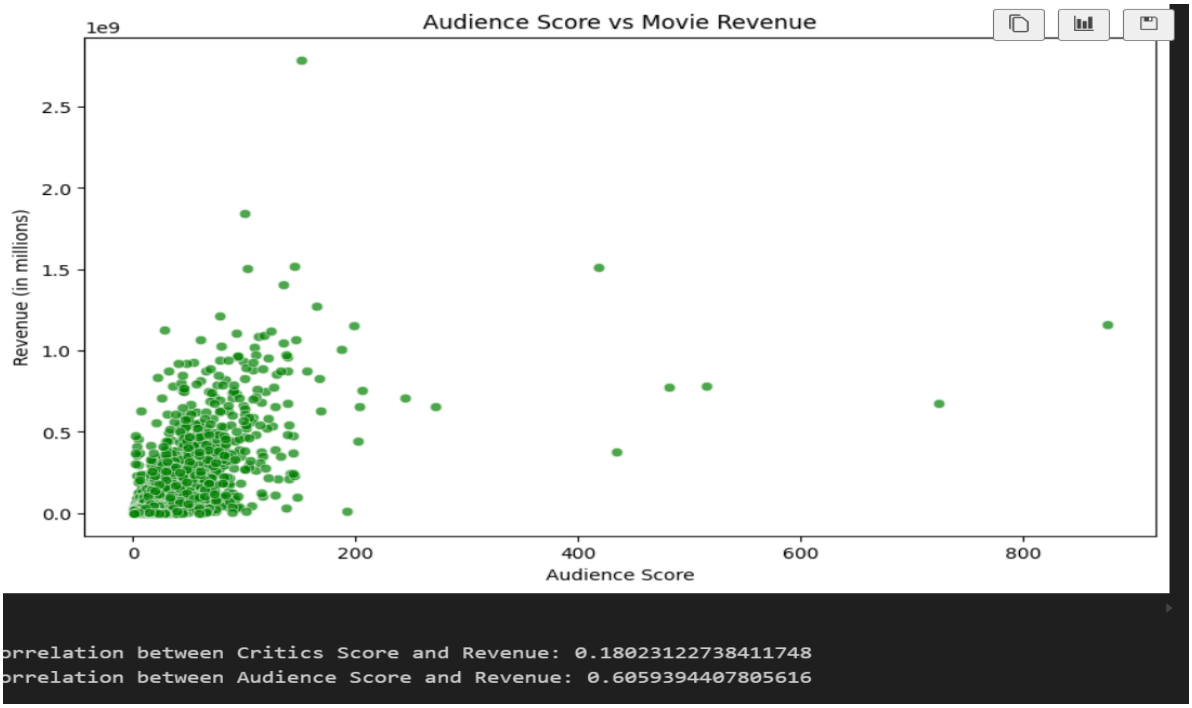




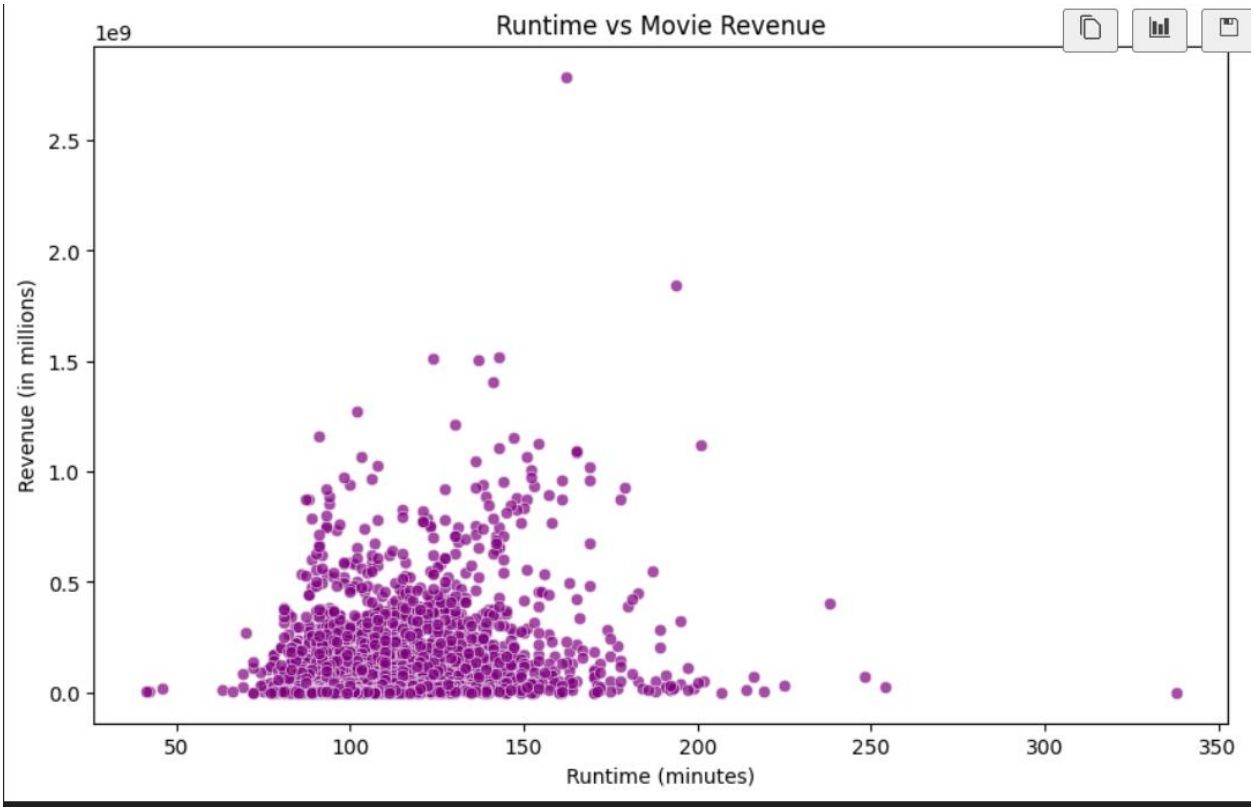
توضیح) **نمودار پراکندگی برای امتیاز منتقدان و درآمد** :این نمودار نشان می‌دهد که آیا امتیاز بالا از منتقدان) محور (x با درآمد بالاتر فیلم‌ها) محور (y مرتبط است یا خیر.

نمودار پراکندگی برای امتیاز مخاطبان و درآمد :این نمودار ارتباط بین امتیاز مخاطبان و درآمد فیلم‌ها را نشان می‌دهد.





توضیح) پراکندگی و ارتباط زمان فیلم با درآمد



پیشنهاداتی برای بهبود عملکرد مدل:

برای بهبود عملکرد مدل‌های پیش‌بینی درآمد فیلم‌ها، می‌توان چندین اقدام مختلف را در نظر گرفت. این پیشنهادات می‌توانند به دقت بیشتر مدل کمک کرده و به قابلیت تعمیم آن در برابر داده‌های جدید و ناشناخته افزوده شود. در ادامه برخی از پیشنهادات بهبود عملکرد آورده شده است:

1. افزایش کیفیت داده‌ها

کیفیت داده‌ها نقش بسیار مهمی در دقت مدل دارد. بهبود داده‌ها می‌تواند باعث بهبود عملکرد مدل‌ها شود.

- پاکسازی داده‌ها :بررسی و اصلاح داده‌های گمشده یا نادرست برای ویژگی‌های کلیدی مانند درآمد فیلم‌ها، بودجه، زمان فیلم، و امتیازهای منتقدان و مخاطبان ضروری است. استفاده از روش‌های ایمپرشن بهینه (به‌ویژه برای داده‌های گمشده در ویژگی‌های حیاتی) می‌تواند دقت مدل را افزایش دهد.
- شناسایی و حذف داده‌های پرت :(Outliers) فیلم‌هایی که دارای ویژگی‌های بسیار متفاوت از دیگر فیلم‌ها هستند می‌توانند تاثیر زیادی بر عملکرد مدل بگذارند. شناسایی و حذف این داده‌ها ممکن است به بهبود پیش‌بینی‌ها کمک کند.
- افزودن ویژگی‌های جدید :ویژگی‌هایی مانند نوع تبلیغات، میزان نمایش در سینماها و بازاریابی آنلاین می‌توانند بر درآمد فیلم‌ها تاثیرگذار باشند و باید به مدل اضافه شوند.

.2مدل‌های پیچیده‌تر و ترکیبی

استفاده از مدل‌های پیچیده‌تر می‌تواند به بهبود دقت پیش‌بینی‌ها کمک کند.

- استفاده از مدل‌های :Ensemble ترکیب مدل‌ها مانند Stacking, Baggingو Boostingمی‌تواند عملکرد مدل را بهبود بخشد. به عنوان مثال، استفاده از Stacking Regressorکه مدل‌های مختلف را ترکیب می‌کند، می‌تواند پیش‌بینی دقیق‌تری ارائه دهد.
- مدل‌های غیرخطی :استفاده از مدل‌های غیرخطی پیچیده‌تر مانند شبکه‌های عصبی (Neural Networks)می‌تواند دقت مدل را در پیش‌بینی درآمد فیلم‌ها افزایش دهد.
- مدل‌های خودآموز :(AutoML) استفاده از ابزارهای AutoMLمانند TPOTیا H2O.aiکه به طور خودکار مدل‌ها و هایپرپارامترهای بهینه را انتخاب می‌کنند، می‌تواند به دقت مدل کمک کند.

.3تنظیم هایپرپارامترها(Hyperparameter Tuning)

تنظیم بهینه‌های هایپرپارامترها می‌تواند تاثیر زیادی بر عملکرد مدل داشته باشد.

- جستجوی دقیق‌تر هایپرپارامترها :استفاده از RandomizedSearchCVو GridSearchCVبا مقادیر گسترده‌تر برای هایپرپارامترهای مدل‌های مختلف می‌تواند به تنظیم بهترین مقادیر و بهبود دقت کمک کند.
- تنظیم پارامترهای الگوریتم‌های :Ensemble برای مدل‌هایی مانند Random Forestو Gradient Boosting، تنظیم پارامترهایی مانند تعداد درخت‌ها، عمق درخت، نرخ یادگیری، تعداد ویژگی‌های انتخابی، و غیره می‌تواند منجر به بهبود عملکرد شود.

.4تعامل بین ویژگی‌ها(Feature Interaction)

افزایش دقت مدل با اضافه کردن تعاملات پیچیده‌تر بین ویژگی‌ها امکان‌پذیر است.

- ساخت ویژگی‌های ترکیبی :به طور مثال، می‌توان ویژگی‌هایی مانند ترکیب ژانرها و امتیاز منتقدان را به صورت ویژگی‌های ترکیبی وارد مدل کرد تا تاثیرات متقابل این ویژگی‌ها بر درآمد بهتر شبیه‌سازی شود.
- استفاده از ویژگی‌های زمانی :اضافه کردن ویژگی‌هایی که به‌طور خاص تاثیر زمان بر درآمد فیلم‌ها را نشان دهند، مانند تاریخ انتشار فیلم، ماه‌های تعطیلات، افزایش یا کاهش رقابت، می‌تواند دقت مدل را بالا ببرد.

.5استفاده از داده‌های بیشتر(Data Augmentation)

- گسترش دیتاست :استفاده از داده‌های بیشتر، مانند میزان فروش در کشورهای مختلف، اطلاعات بیشتر درباره بازیگران و کارگردان‌ها، و واکنش‌های عمومی به فیلم‌ها می‌تواند به مدل کمک کند تا با توجه به ویژگی‌های بیشتری پیش‌بینی‌های دقیق‌تری انجام دهد.
- استفاده از داده‌های تاریخ‌نگاری :اضافه کردن داده‌های مربوط به روند تغییرات درآمد فیلم‌ها در طول زمان می‌تواند به مدل کمک کند تا درک بهتری از بازار و تغییرات آن به دست آورد.

.6ارزیابی مدل در شرایط مختلف(Cross-validation)

برای ارزیابی بهتر مدل و جلوگیری از :Overfitting

- استفاده از :K-Fold Cross Validation برای ارزیابی بهتر عملکرد مدل، استفاده از K-Fold Cross Validationبه جای تقسیم داده‌ها به یک مجموعه آموزشی و یک مجموعه تست، می‌تواند به مدل کمک کند تا بهتر تعمیم یابد و از overfittingجلوگیری کند.
- ارزیابی مدل با مجموعه‌های مختلف داده :استفاده از مجموعه داده‌های مختلف برای ارزیابی و تست مدل می‌تواند به درک بهتر عملکرد واقعی مدل کمک کند.

7.ویژگی‌های بیشتر برای مدل(Feature Engineering)

- اضافه کردن ویژگی‌های جدید :برخی ویژگی‌ها مانند نوع تولید (Studio) یا میزان تبلیغات و بازاریابی آنلاین نیز می‌توانند به دقت مدل کمک کنند.
- ایجاد ویژگی‌های ترکیبی :ترکیب برخی ویژگی‌ها مانند بودجه به مدت زمان (بودجه فیلم تقسیم بر مدت زمان فیلم) می‌تواند در ایجاد ویژگی‌های جدید که به مدل کمک می‌کنند موثر باشد.

8.آموزش با داده‌های خارجی(External Data)

- داده‌های خارجی مانند میزان تماشای آنلاین فیلم‌هااز پلتفرم‌های مانند Netflix, Amazon می‌تواند به پیش‌بینی دقیق‌تر درآمد کمک کند.
- داده‌های اجتماعی :تحلیل‌های رسانه‌های اجتماعی و ترندهای اینترنتی می‌تواند به پیش‌بینی میزان علاقه‌مندی به فیلم‌ها و تاثیر آن بر درآمد کمک کند.