Springboard
Capstone Project 1

Film Data Study Final Report

Outline

- Problem
- Data Sources
- Data Wrangling and Processing
- Exploratory Data Analysis
- Data Storytelling
- In-Depth Analysis

Problem

Every movie consists of thousands of different parts and factors. There are of course the actors and directors, producers and writers. And there are consultants, agents, crews like the cinematographers, sound engineer, lighting expert and even caterer, the list goes on.

To everyday movie goers, the public facing aspect of these “parts” are the main determinant whether average joe or ordinary jane decides that they might like the movie or not. By public facing parts, this will include the actors, directors, producers, production studio or even genre. And with around 600 movies being made every year, we see definite trends in a specific genre or film produced by a certain studio or director or all of those combined. For example, Marvel studios with Kevin Feige as executive producer making superhero films starring Robert Downey Jr. as Iron Man has been dominating the box office for almost a decade now. But as we all know, this wasn’t the case 20 years ago.

The question we are trying to answer is: are there patterns or even cycles of genres that popularize? Are there combinations of movie “parts” that will produce a blockbuster?

Data Sources

All data to be used in this project has been provided by IMDb website. The website contains film related data from the 1890's to present and is updated everyday. The full data consists of 7 different tsv files. The description for each data file can be found on https://www.imdb.com/interfaces/ . Because of the scope of this project being specific to movies, we will not be using the data files: **title.episode.tsv** (contains the tv episode information) and **title.akas.tsv** (contains regions specific information like languages).

Data files used

title.basics.tsv : Contains basic information about the film such as title and year released

title.crew.tsv : Contains the director and writer information

title.principals.tsv : Contains the principal cast/crew for titles

title.ratings.tsv : Contains the IMDb rating and votes information for titles

name.basics.tsv : Contains further information about the cast/crew

These files are available to download in the link: https://datasets.imdbws.com/

Data Wrangling

All five data files used in the project needed to be processed to account for null values and select the important features to be used for analysis.

title.basics.tsv

Contains basic information about the film such as title and year released.

Findings:

- This is the main dataset where important data is kept such as title, year released, runtime and genres. It was very important that these above features were clean and available.
- Title showed missing entries for 7 records. These were parsing errors occurring from tsv read. Pandas was not distinguishing '/t' as a tab and included the contents after '/t' as part of the previous data.
- On read, empty data was written as '/N'. This was translated to NaN during reimport.

title.crew.tsv

Contains the director and writer information.

Findings:

- This dataset contains valuable director data that we will need to use later. Aside from the director and possibly writer data, there is not much to work with.
- There were no missing entries for this dataset.
- Similar behavior was seen for null values. Reimport to account for null values as '/N' translated them to NaN.

title.principals.tsv

Contains the principal cast/crew for titles.

Findings:

- There were no entries missing for any columns for this dataset.
- Reimport to account for '/N' as null was done.
- This dataset was eventually decided not to be used during analysis as only the director data was chosen to move forward with.

title.ratings.tsv

Contains the IMDb rating and votes information for titles.

Findings:

- This dataset was also very important as it contains the average rating the movie has received and the number of votes which would determine whether a movie was well received or not.
- There were no entries missing for any columns for this dataset.

name.basics.tsv

Contains further information about the cast/crew.

Findings:

- This dataset contains the actual names of the crew. We are only interested in directors. In the previous dataset, the individuals in this dataset were given identifier codes that could be used in this dataset to find out further information about the person.
- Reimport to account for '/N' as null was done.
- There were quite a lot of null entries (1.7M out of 9.4M) for the primaryProfession column which gives us the actual position of the individual. This was also another reason we left out the principal's dataset above. Most of the director data was present but a lot of crew data was missing.

Data Processing

Master Dataframe

After successfully importing and cleaning the datasets to use, it was important that there was a master dataset that contains all the information required.

Most important features were title, genres, averageRating, numVotes, directors, startYear, runtimeMinutes. These data were scattered across three different datasets and the three datasets (basic, name, rating) all had different numbers of movies.

Since we are dealing with movie popularity, it was imperative that all movie data has a rating. So the final master dataset was centered around the ratings dataframe and were merged using inner join.

As a result, the initial creation of the master dataset reduced the amount of data from 523864 records to 523864 to 236124 initially.

This dataframe was further reduced if there were any records with any of the above 7 features missing.

Feature Engineering

Genres column contained up to three different genres out of 28 different genre types. Due to this, there were 1230 different genre combinations that made it difficult to categorize. Two approaches were used to account for this.

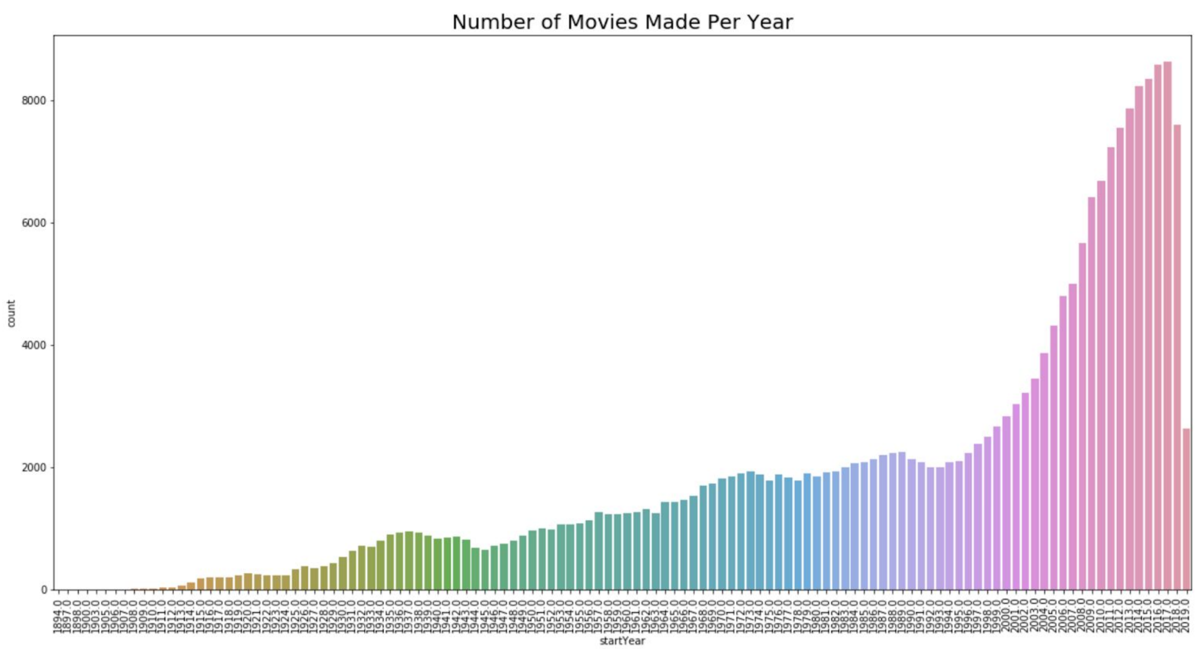
- For ease of genre component analysis, 28 columns comprising each genre were added to the dataframe with 1 or 0 representing the inclusion of the genre for movies.
- For ease of movie categorization, top 5 genres were found and stripped other irrelevant genres from the list of genres so there were 23 genre combinations to work with instead of 1230.

Exploratory Data Analysis

Because almost all of the data used in this project were categorical, in depth inferential statistics was not applicable.

startYear

- Range: 1894 ~ 2019
- Steady increase in the number of movies were seen from 1890's until 1990's.
- From 2000's, the number of movies released has increased significantly.



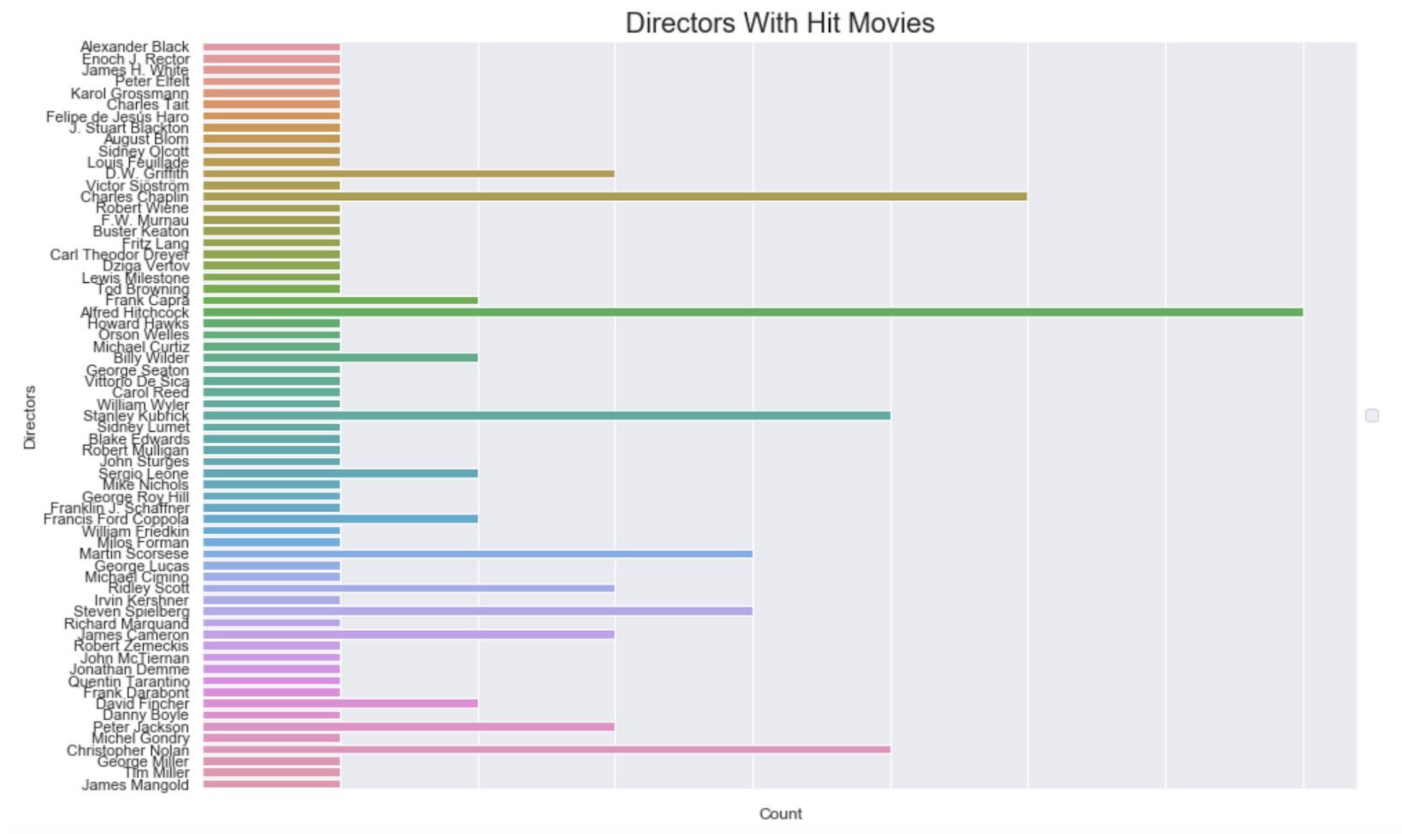
- 2018 and 2019 showed a decrease due to data being collected in Q2 of 2019 where not all movie data were fully entered.

runtimeMinutes

- Range: 1.0 ~ 51420.0
- It was very strange to see there are entries of movies of length 1 minute to movies over 800 hours. There were quite a few of these absurd runtime movies and were considered outliers.
- To account for outliers, the 1.5 IQR method was used with lower and upper bound of 57.5 minutes to 125.5.
- Further analyzed the runtime changes throughout the history by slicing the dataframe to contain 20 years of movie runtime data only and compared the statistics.
    - Average runtime of movies has increased by 12% since the 1890's to present. (81 to 92)

Directors

- Number of directors in the master dataframe: 94577
- One of the points of this project is to find characteristics in movies that would make it popular. Although there can be exceptions, directors who have only directed or produced one movie are not likely to be considered as popular movie directors.
- After removing one-time directors: 30857

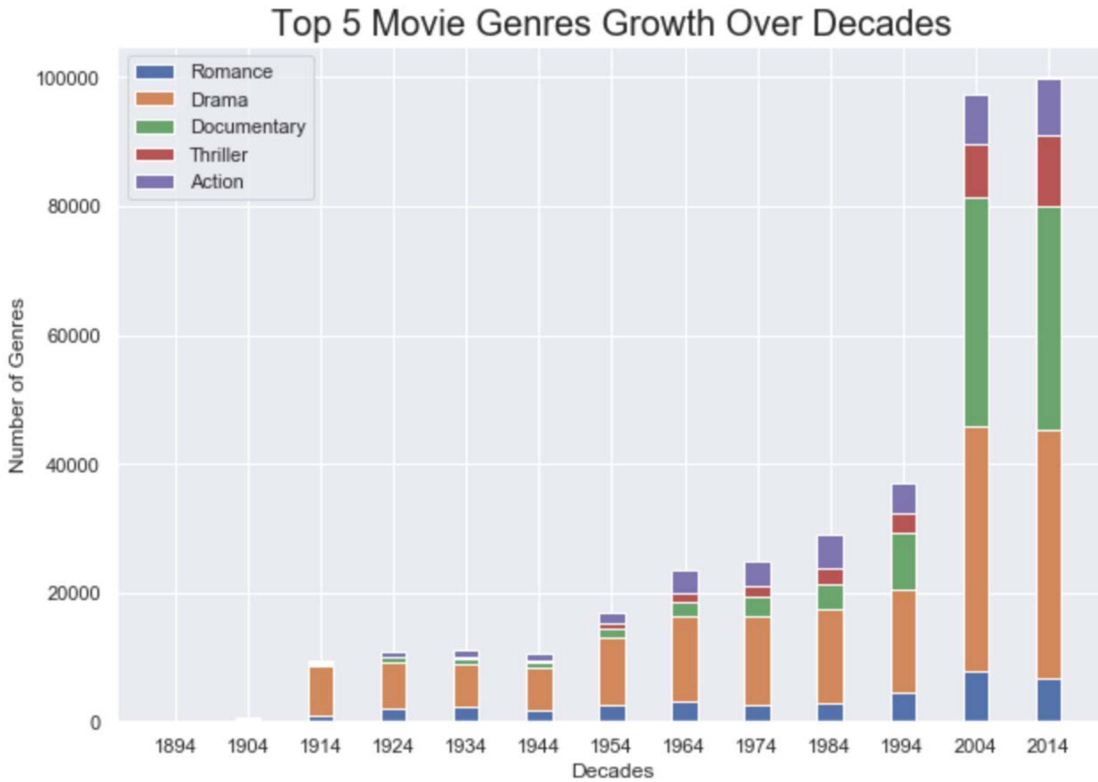


Popularity Metric

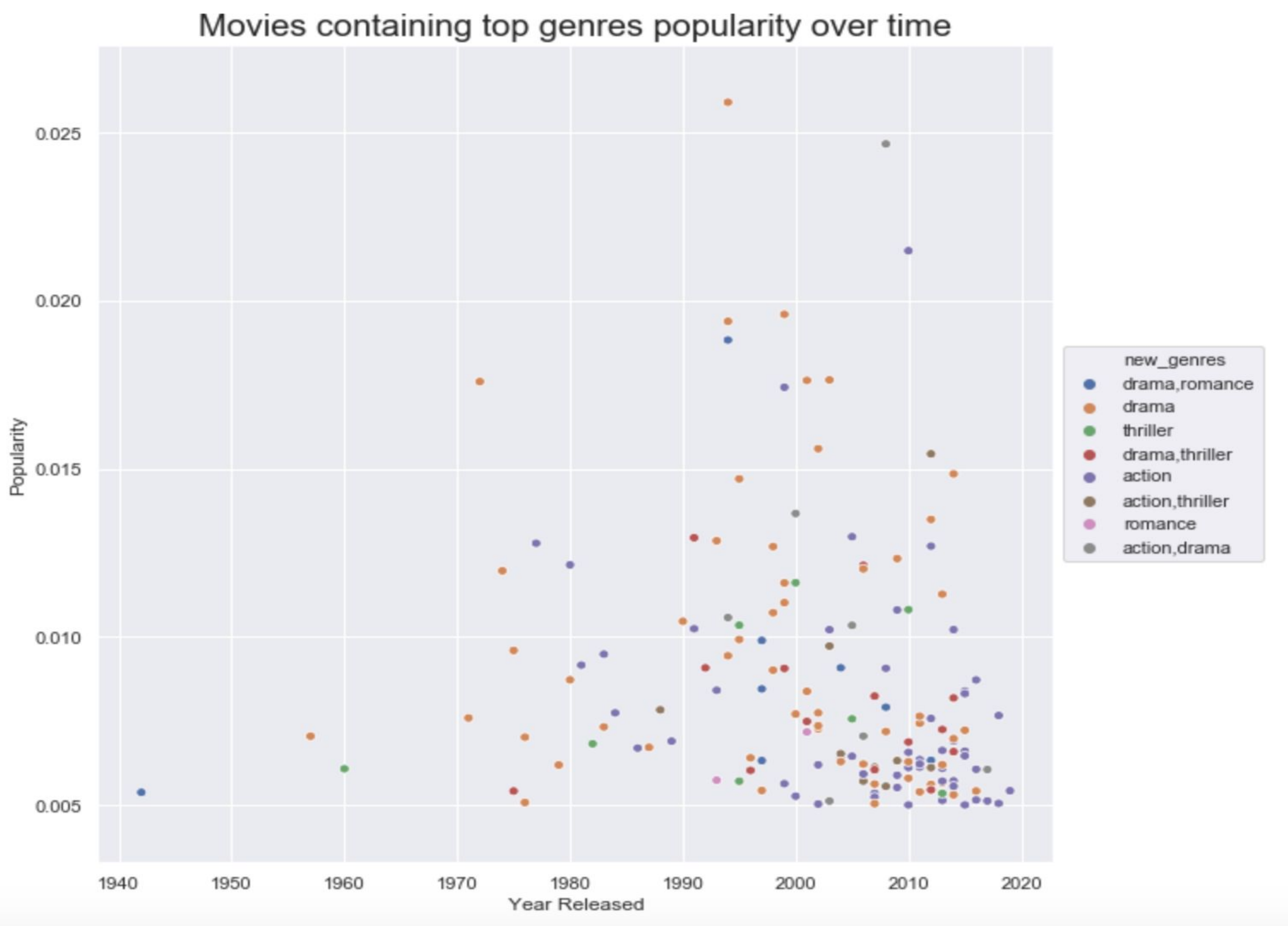
- Our goal for this project was not necessarily to find the highest rated movie but to find the most popular movies. Popularity was determined heavily by combining both the number of votes the movie has gotten and the average ratings.
- The outcome is the 'popularity' column in the dataset. We follow the below simple calculation to get the metric.
    - **popularity metric = averageRating * numVotes/totalVotes**
- This metric gave us key insight as to which movies to consider and which to ignore. Even if a movie had averageRating of 9.5, if only 5 people voted for that movie, it scored very low as this reflects that the movie, although well made, was not a box office hit.

Data Storytelling

With the top 5 genres derived, it became easier to find out how often these genres were used throughout history.



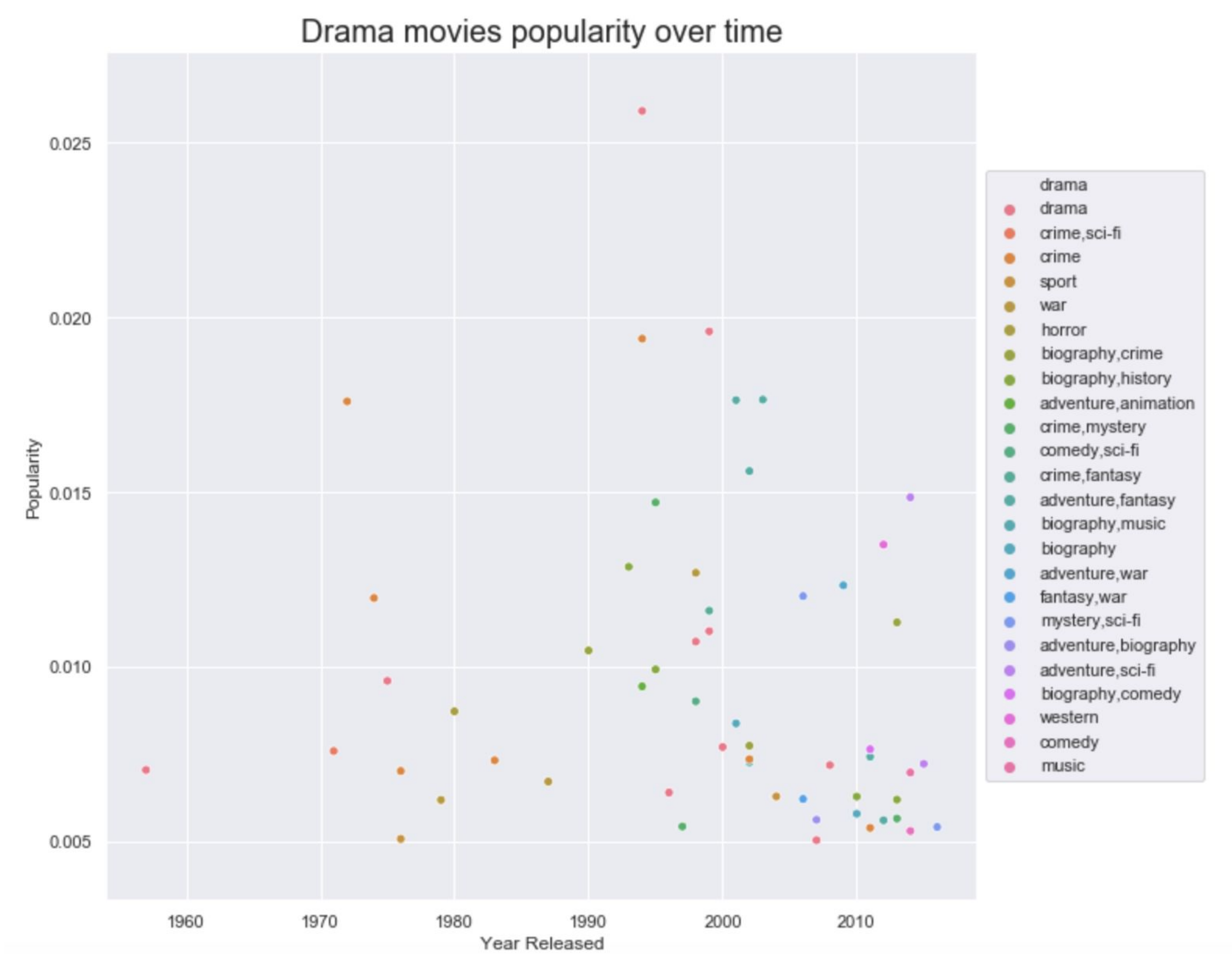
Adding popularity metrics on these genres, we are able to see which combination of these genres proved to be more popular.



Drama is the leading genre component that is embedded in almost all of the popular films. Quite a lot of orange data points that are just 'Drama' but this is due to our feature engineering done just before. So these orange Drama data points can be a mixture of Drama genre and any other lesser used genre components like Fantasy or Sci-Fi.

Next, we can see the increase in the amount of popular action films. The Drama genre distribution is very consistent throughout so we can assume that the Drama genre is a somewhat 'backbone' of good film production.

However, we have not been seeing a lot of popular action films until the late 1970's. And since then until 2019, we can see it has become the 2nd most included genre in our 'popular' films. This is represented by the purple, gray and brown data points that we can see a lot more from 2000-2020. By the late 2010's, we are even seeing more action type films over drama type films.
Below is a chart of only movies containing ‘Drama’ as genre. The colors represent genres other than Drama.



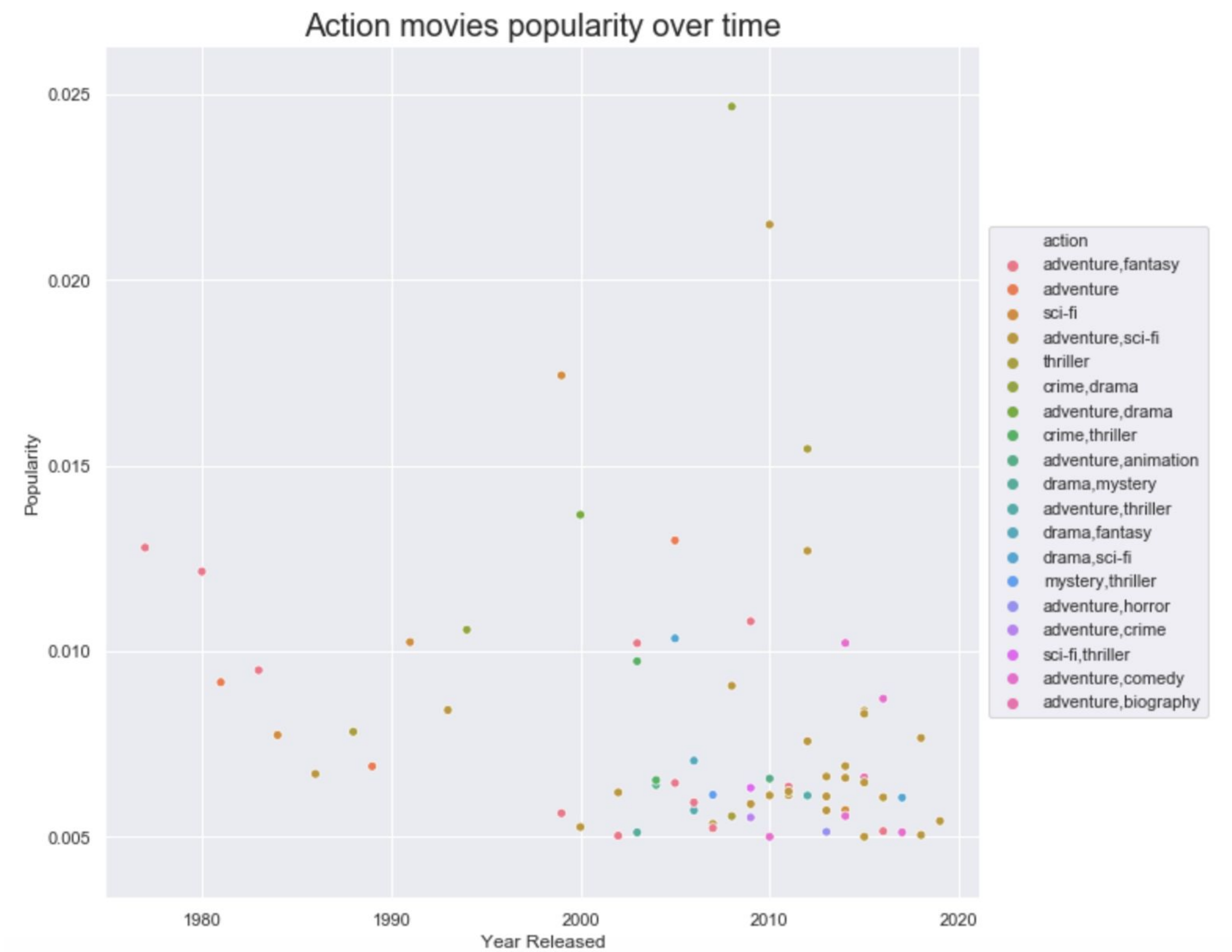
It is noticeable that there were more movies with ideas of War, Crime, Biography and History related popular movies until the 1980's. Then the trend changes to more of Crime, Fantasy, Adventures and Mystery until the latter half of 2000's. From then on until now, we are seeing a lot more of Sci-fi, Fantasy, Adventure and Comedy.

In the early years of film production, the CGI technology was not readily available and very expensive. The first movie to adopt CGI was in 1973's Westworld. In this era, practical effects were mainly used and because of the limited CGI capabilities, popular movies were mostly based on very real or historical events or storylines that did not involve a lot of computer generated images.

Near the 2000's, CGI capabilities were enhanced by a lot and this was an era where there was a good mix of both practical effects and CPI. This allowed production companies to create vast scale images or other worldly scenes and adding CGI on top of practical effects were mostly used to create very real looking effects. This opened the door for directors to bring the great fantasy novels to life.

Nowadays, CGI is used in almost all movies to a point it is difficult to find movies that do not utilize it. Some movies are shot almost exclusively in front of green screens. With these technological advances, what the audience sees on screen could be whatever the director imagined. This created a boom of sci-fi films that dive into crazy scientific theories and introduction of highly advanced techs on screen all taking place in environments all up to the director's imagination.

Below is a chart of only movies containing 'Action' as a genre. The colors represent genres other than Action.



They are almost dominated by adventure and sci-fi in the modern filmmaking era which is what we expected from the previous drama only movies investigation. It also looks like action movies are likely to be paired with Adventure genre which makes sense as for an action film as it is important for characters to express these "actions" in different places.

primaryName	primaryTitle	genres
Frank Darabont	The Shawshank Redemption	drama
Christopher Nolan	The Dark Knight	action,crime,drama
Christopher Nolan	Inception	action,adventure,sci-fi
David Fincher	Fight Club	drama
Quentin Tarantino	Pulp Fiction	crime,drama
Robert Zemeckis	Forrest Gump	drama,romance
Peter Jackson	The Lord of the Rings: The Return of the King	adventure,drama,fantasy
Peter Jackson	The Lord of the Rings: The Fellowship of the Ring	adventure,drama,fantasy
Francis Ford Coppola	The Godfather	crime,drama
Peter Jackson	The Lord of the Rings: The Two Towers	adventure,drama,fantasy
Christopher Nolan	The Dark Knight Rises	action,thriller
Christopher Nolan	Interstellar	adventure,drama,sci-fi
David Fincher	Se7en	crime,drama,mystery
Ridley Scott	Gladiator	action,adventure,drama
Quentin Tarantino	Django Unchained	drama,western
Christopher Nolan	Batman Begins	action,adventure
Jonathan Demme	The Silence of the Lambs	crime,drama,thriller
Steven Spielberg	Schindler's List	biography,drama,history
George Lucas	Star Wars: Episode IV - A New Hope	action,adventure,fantasy
Joss Whedon	The Avengers	action,adventure,sci-fi

The above shows the top 20 popular movies' director list. We can see Christopher Nolan having the most movies in the top 20 with 5 movies. Next, Peter Jackson takes the 2nd place in most movies in top 20 with his Lord of the Rings trilogy. Next, we have David Fincher and Quentin Tarantino for Fight Club, Se7en and Pulp Fiction, Django Unchained respectively.

From this list, we can see that Christopher Nolan's movies, which are all relatively new movies within a decade or slightly over, along with Peter Jackson, contains the combination of genres: Action, Drama, Sci-fi and Fantasy.

In-Depth Analysis

In this section, we will finally utilize the processed data to build a model to predict when given the characteristics of a movie, will it be popular or not.

First question is... How do we measure that a movie is successful or not?

We have been using the 'popularity' field as a guideline for measuring success. This field is calculated from two features in the original dataset, averageRating and numVotes.

popularity metric = averageRating * numVotes/totalVotes

In preparation for actual modelling, we need to drop columns like primaryTitle, averageRating, numVotes and popularity. PrimaryTitle is unique to all movies and does not have weight to whether the movie will be successful. AverageRating, numVotes were used to calculate the popularity metric so keeping those fields would be redundant and lastly, popularity metric will be converted to categorical columns to indicate normal (0), popular (1) and classic (2).

After encoding the numeric variables into categorical variable ranging from 0 to 2, we were left with this disbritution :

0 :   220072

1 :   2718

2 :   68

The distribution for the above three different ground truth results are very skewed. Because of this reason, our normal test_train_split may not be the best way to split the data frame as this is done randomly. We will first try this method and later try with a different method of test and training a dataset with reasonably distributed prediction values.

SMOTE Sampling

We have a heavily skewed dataset. Our goal is to correctly identify the movies that might become popular but our dataset only consisted of around 70 movies categorized as 2 or 'popular'. If we are building a model that identifies the 0's well, we would have less issues but we are mainly focused on finding 2's. Having less that 0.1% of 2's in the data frame will not produce a model with high accuracy.

Therefore, we are using SMOTENC to equalize the three categories' distribution.

Downsampling

After SMOTE sampling, we are left with almost three times the size of our dataframe. Having a dataset of size 462151 may prove to be difficult when attempting models like SVC where it creates a hyperplane between the perceived clusters of data, this will take a long time. More so because we are not dividing the data cluster into two but three.

We will downsample the SMOTE sample to about 15000, our original number of records, and test for modelling.

Random Forest Classifier

Testing out-of-box model with downsampled dataset:

```
Accuracy: 0.96549617549794 26
different predictions:  {0.0, 1.0, 2.0}
```

Predicted True	0.0	1.0	2.0	All
0	63283	2738	15	66036
1	2978	62210	837	66025
2	5	261	65738	66004
All	66266	65209	66590	198065

Already out of the box, our model is doing very well in identifying the '2' which are the most popular movies and our goal of this project. Out of the three classifications, '2' has the highest precision rate.

We can see that we have mis-predicted 5 potential popular movies to a '0' which is the identifier for bad movies. It is better that we over-predict some non- popular movie to it becoming a popular movie than under-predict a potentially huge movie to a mediocre movie.

This means we are focused on Recall than Precision at this time.

In the following steps, we will also test the out of box Random Forest Classifier with the full X_t dataset to see if the large training dataset will have further impact on the accuracy than move on to Randomized Search for hyperparameter optimization.

Testing out-of-box model with full dataset:

```
Accuracy: 0.9805669855855401
different predictions:  {0, 1, 2}
```

Predicted	0	1	2	All
True				
0	64596	1426	14	66036
1	1822	63716	487	66025
2	2	98	65904	66004
All	66420	65240	66405	198065

Random Forest Hyperparameter Optimization

In order to find out the hyperparameter that best fits our problem, we will first conduct RandomizedSearch to test different variations of hyperparameter.

Randomized Search Parameter Grid:

{'bootstrap': [True, False],

'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], 'max_features': ['auto', 'sqrt', 1, 2, 4, 6],

'min_samples_leaf': [1, 2, 4, 6, 8, 10],

'min_samples_split': [2, 4, 6, 8, 10, 12],

'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}

During this process and further hyperparameter optimization process to follow, we are comparing the models with recall_score as we are interested in identifying all movies that may fall into category '2' rather than of the predicted 2's, how many are right.

Best RandomizedSearch Hyperparameter:

{'n_estimators': 400, 'min_samples_split': 8, 'min_samples_leaf': 1, 'max_features': 6,

'max_depth': 100, 'bootstrap': False}

```
Accuracy: 0.9848383106555929
different predictions:  {0, 1, 2}
```

Predicted	0	1	2	All
True				
0	64795	1233	8	66036
1	1241	64339	445	66025
2	6	70	65928	66004
All	66042	65642	66381	198065

The crosstab shows very promising results. We are reducing the number of mis-predicted '2' down from 99.848% to 99.885% recall score. The Randomized search has definitely worked in favor of recall score. The difference may not be great as our original model already did a nice job of predicting '2'.

We will take this one more step and perform GridSearchCV to finetune the parameters from RandomizedSearchCV.

GridSearch Parameter

{ 'bootstrap': [False], 'max_depth': [80,100,120], 'max_features': [4,6,8],

'min_samples_leaf': [1, 2, 3], 'min_samples_split': [6,8,10] 'n_estimators': [300,400,500] }

Best GridSearch Hyperparameter:

{'bootstrap': False, 'max_depth': 80, 'max_features': 8, 'min_samples_leaf': 1,

'min_samples_split': 10, 'n_estimators': 300}

Predicted	0	1	2	All
True				
0	64850	1178	8	66036
1	1148	64435	442	66025
2	6	78	65920	66004
All	66004	65691	66370	198065

We got back a very similar set of hyperparameters compared to our prior randomized search attempt. This is not the first attempt in getting back best parameters for GridSearchCV. The best parameters returned from previous iterations always resulted in a very different set of best hyperparameters. After increasing the cv value from 3 to 5 and providing a full dataset (instead of the downsampled 1/3 dataset) yielded a much better result.

The overall accuracy has increased ever slightly. But technically, we have mis-predicted '2' more than our randomized search cv. The reason this combination was selected is likely due to the overall recall score being higher than our randomized search hyperparameters. Although we mis-predicted '2' 8 more records, we have a better recall score on the remaining '0' and '1' prediction pushing our overall recall score up.

Features sorted by their score

[(0.3026, 'topDirectors'), (0.2987, 'runtimeMinutes'), (0.1812, 'startYear'), (0.0348, 'Adventure'), (0.0291, 'Comedy'), (0.019, 'Drama'), (0.0175, 'Action'), (0.0164, 'Documentary'), (0.0148, 'Biography'), (0.0134, 'Thriller'), (0.011, 'Mystery'), (0.0107, 'Romance'), (0.0099, 'Crime'), (0.0089, 'SciFi'), (0.0088, 'Family'), (0.0087, 'Horror'), (0.0076, 'Fantasy'), (0.0069, 'History')]

It looks like a primary determinant if predicting great movies by the top directors. More so than not, the runtime of the movies also claim a big proportion when it comes to important features. This is most likely the more eccentric movie runtime has a significantly lower popularity rating than the traditional movie configurations. Year releases also seem to have a big impact and then we move on to the genres. As expected a lot of the top 5 genres we've seen so far are included as the biggest decision makers out of genres (Adventure, Comedy, Drama, Action, Thriller).

Testing Other Models

Aside from RandomForestClassifier, other classification models like SVM and LogisticRegression were also used to test for prediction. However, even after hyperparameter optimization, RandomForestClassifier yielded the best results.

Logistic Regression Model after RandomizedSearch

```
Accuracy: 0.8091182187665665
different predictions:  {0, 1, 2}
```

Predicted	0	1	2	All
True				
0	52170	13494	372	66036
1	12888	45678	7459	66025
2	1	3593	62410	66004
All	65059	62765	70241	198065

Support Vector Machine Model after GridSearchCV

```
Accuracy: 0.9361219801580289
different predictions:  {0.0, 1.0, 2.0}
```

Predicted	0.0	1.0	2.0	All
True				
0	59342	6677	17	66036
1	5191	60143	691	66025
2	6	70	65928	66004
All	64539	66890	66636	198065

Conclusion

Throughout various model testing, RandomForestClassifier proved to be the best model for this job. After hyperparameter optimization and cross validating, the model accurately predicted the popular movies to 99.3%.

In real life, this is a ridiculously accurate model which means, this model is either overfitting or our dataset has become too simple where just a few columns determine whether a movie will be popular or not.

The feature importance plays a big role here:

[(0.3026, 'topDirectors'), (0.2987, 'runtimeMinutes'), (0.1812, 'startYear'), (0.0348, 'Adventure'), (0.0291, 'Comedy'), (0.019, 'Drama'), (0.0175, 'Action'), (0.0164, 'Documentary'), (0.0148, 'Biography'), (0.0134, 'Thriller'), (0.011, 'Mystery'), (0.0107, 'Romance'), (0.0099, 'Crime'), (0.0089, 'SciFi'), (0.0088, 'Family'), (0.0087, 'Horror'), (0.0076, 'Fantasy'), (0.0069, 'History')]

We can see that we have a categorical variable ‘topDirectors’ having almost 30% weight in determining the popularity followed by similar weight from ‘runtimeMinutes’ and lastly, 18% for the ‘startYear’. These account for almost 80% of the importance ratio. And speaking outside of the modelling, we know that movie runtime and year released is something actual viewers do not think about or consider when deciding whether a movie is liked or not.

This means that the actual determinant of a popular movie is the director. It makes sense in a way and we can also explain how runtimeMinutes and startYear come to have this much importance if we start thinking this way. Whether a movie was directed by one of the top directors or not also depends on the startYear. As the data collects movies since the 1890's, most of our topDirectors are involved in these movies. Also since the beginning of film age, a lot of different runtimeMinutes have been tried in movies.

The model is likely using the runtimeMinutes column to filter out any movies that are too short or long running (as the popular movies used for training follows strict 1.5hr to 2.5hr runtime) and startYear to further filter the movies released during top directors' career and finally, whether the movie was actually directed by one of the top directors.

Issues

This creates an issue if we use the model later down the road when the model is not aware of the new up and coming directors. Until those movie lists are updated, even if in the future, we input a movie released that was known to have been successful, our model would not be able to predict that it would be popular as it has a limited knowledge of top directors and the model is highly focused on the director list and its effects in runtime or start year.

To fix this issue, we may try to remove the start year and runtime minutes out of the modelling process in the hopes of giving the actual genres a bit more importance.