CIS: 563 INTRODUCTION TO DATA SCIENCE

PROJECT REPORT

Neil Malu, 635692900 *Vindhya Ravi Prakash, 777803438*

1. INTRODUCTION

The movie industry has been hit hard by COVID, and social distancing has caused theatres to be shut down. Further, online streaming studios such as Netflix and Amazon Prime are threatening to greatly impact the way traditional movie studios and production companies work. Now more than ever, it is important for film studios to be cautious with their investments and try to maximize profits using new modern data-driven methods. The key in maximizing profits for any budget is to hire valuable and reliable personnel in order to minimize losses. The metric that this project aims to analyze is Return on Investment. We calculate this metric as (Revenue/Budget), although it is also sometimes considered in terms of the percentage of profit a movie makes.

This project aims to create a classifier that identifies the main reasons behind a successful movie with a high Return on Investment so that studios can try to produce new movies using that model. Obviously, studios cannot control the external factors that may determine how a movie performs (such as other similar movies releasing at similar times, further COVID lockdowns or consumer fatigue). So we aim to create a model based on the personnel that production studios can hire (such as actors and directors) and the types of movies they can produce to ensure reliable profits.

The resulting problem statement begs the question - what is considered a good return on investment? The answer is based on the datasets we used (which are attached to this project) - and we have split the ROI into four classes by discretizing it, so that our classifier can predict whether a movie's returns are bad, ok, good or excellent. This project uses linear regression to fix missing values, removes outliers outside the 0.05 quantile range, classifies the dataset using a Random Forest Classifier and the LeaveOneOut k-Fold cross validation method to evaluate the classifier. The evaluation metric used is accuracy.

2. PRIOR WORK

Projecting the revenue of an upcoming film is not a new problem, and has been the key factor governing financial decisions made by studios. According to [1] however, most of these predictions have been based on statistical models which can at best, only provide a rough estimate of film revenue. It's also a known and obvious fact that increasing the budget of a movie tends to increase the revenue, which is why the metric this project tries to maximize is ROI. Paper [1] tries to create a computational model using a two layered neural network for predicting revenue based on MPAA ratings (G, PG-13 or R), release date and genre. However, this paper uses a dataset of size 1.2 million with only 3,593 of those films having valid box office data. In contrast, this project uses a much smaller data set and tries to estimate these missing values to improve the training of our model.

This project also builds on Link [2]. This link is a very good exploration of the kind of data that we are working with, and highlights the importance of ROI as a metric and some interesting findings. However, it doesn't analyze the reasons for ROI and does not use any prediction algorithm, but simply provides good visualizations of how our data features correlate to ROI.

3. IMPLEMENTATION

We have implemented our project using Python and its data libraries (panda, numpy, pyplot) as well as it's machine learning libraries from scikit learn. The classifiers have been implemented with guidance from the sklearn_examples provided on Blackboard, as well as the official scikit learn documentation.

3.1 DATASET

We have used and combined two datasets for this project. Both of them are available on Kaggle.
Dataset 1 - https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

- We mainly focused on IMDb movies.csv to get all the details of movies like
- IMDb id, title, year, genre, duration, director, actor, budget, world wide gross income (revenue), language, country and avg votes (IMDb score)
- We also took total votes(number of user votes) from IMDb ratings.csv which is part of the same dataset. The other two files were ignored.
- Some of the English movie titles are in Italian, but since titles are not used in our prediction algorithm, this was ignored.

Dataset 2 -https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

- We renamed movies_metadata.csv to dataset2.csv and only took columns like movie imdb link, title, year, genre, duration, director, actor name, budget, gross, country language, IMDb score and number of user votes.
- From the movie_imdb_link we extracted the IMDb id and used this id as the row index for both our datasets.

While extracting and reading both datasets, we filtered out non English movies and only considered movies produced in the USA and UK since so that the currencies for budgets and revenues matched. We then renamed the columns and merged the dataset to have an initial size of 53,163 rows and 13 columns.

3.2 DATA PREPROCESSING

3.2.1 Data Cleaning

Since dataframes in python are read as strings, we started by converting the 'year' column to numeric type (integer) and removed the resultant null values if the conversion was not possible.

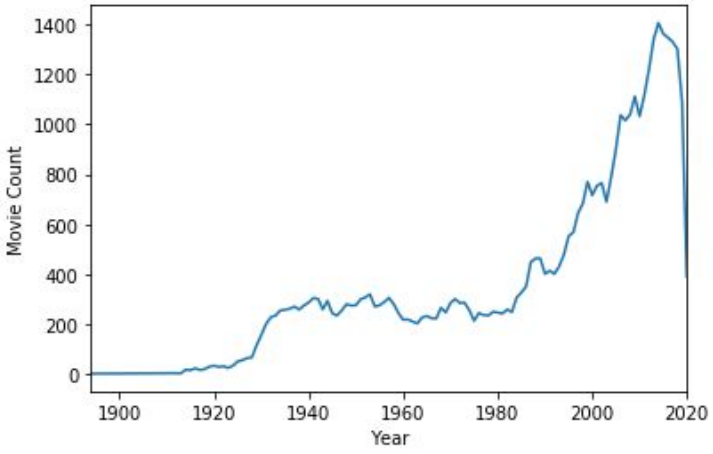


Fig 1: Movies by year

As seen in Fig 1, there are a few inflection points in the count of movies in our dataset. The first inflection point is around 1930 and then another one around 1980. If we ignored movies before 1980 we would lose a lot of information. But if we considered movies between 1930-1950 our prediction would bias actors and directors who are retired or not alive anymore, so we only considered movies after 1950 - leaving 47,160 rows.

Special characters such as Å, Æ were removed from movie titles. Currencies like $, INR, EUR and GBP and punctuations (.,) and leading zeros were also removed from budget and revenue information. These columns were then converted to floating point type.

Then duplicates based on the IMDb ids were removed - we kept the ones with the highest profit margins among all the duplicates. This brought our data size to 42,770.

3.2.2 Handling Missing Values

Most of the missing values were present in budget and revenue columns.
A lot of these values were also zeros. These were first converted to null type.
Then movies with unreasonably low budgets and revenues for today's market were removed (below $1,000,000) as well as movies without both budget and revenue data were removed - which left us with 12,570 records.

We then had several options to deal with these missing values.

- **Drop rows that have null values** for both budget and revenue - this reduced the accuracy as a lot of information was lost. Data size was reduced to 4434.
- **Imputing mean/median to the missing value**. This was a good option but biased high budget and high revenue movies and therefore lifted the ROI unreasonably for the majority of the films. It also ignores the correlation between these two columns and features like directors, actors, and year released.
- **Algorithms to handle missing values** - Linear Regression was used to input missing values for budget and revenue columns so that the correlation between these columns was taken into account. However, when predicting revenues, we got negative values. To deal with this, we added a constant to each predicted row to maintain the correlation and get positive ROIs.

3.2.3 ROI calculation

After the entire dataframe had no null values, we calculated ROI using budget and revenue.

$$ROI = \frac{revenue}{budget}$$

On calculating the ROI for the 12,570 records left, we got ROI's in the range of 0 and 94. Since such high ROI's are unrealistic and are probably a function of the regression error and/or outliers, such outliers were removed from the extreme 5% quantiles on either end (less than 0.05 and greater than 0.95). This resulted in 11,312 records left with ROIs ranging from 0 to 13.94.

3.2.4 Features (Dimensionality Reduction)

Then we try to find out which features affect ROI and to what degree. If we select enough features with a reasonably strong positive or negative correlation it would improve the accuracy of the classifier.

Budget: It became pretty obvious that high budget movies tend to have higher revenues, so having a massive budget was obviously a big factor in returns, but since budgets are inversely proportional to ROI, these often cut the gains.

Runtime: Since the runtime of a movie has a strong positive correlation with the budget, we expect that higher runtimes will probably reduce the ROI of a movie. Some exceptions include epics and genres like Film-Noir, War and old Westerns.

IMDb Ratings and votes: IMDb ratings and votes both had an interesting correlation to the profitability of a film. We found out that most movies with very low, or very high ratings had a very high profitability. This is probably explained by low rated movies having low budgets and therefore boosting the ROI. On the other hand, high rated movies had higher budgets, and as shown before, on occasion, boosted revenues significantly.

Year released: Since the overall popularity of movies is increasing, we expect that more recent movies will have a higher profitability. However old classics that had low budgets will also have a high ROI.

Genres: To get an estimate of how much impact a particular genre has on the profitability of a movie, first the genres had to be converted to unique strings and added to a set. Then for each movie, since there are multiple genres, it would be better if each genre was scored based on the ROI. So for example, if a movie had genre's Animation, Comedy and Drama, and the ROI of the movie was 6 - each of these 3 genres would have 6 added to their ROI sum. Then based on the count of these genres, the average ROI for each genre is calculated.

$$Genre\ Score\ = Avg\ of\ Avg\ ROI\ for\ each\ movie$$

Directors: Directors are expected to have a high impact on the movie's profitability. For each director, we did a similar calculation as genre, and calculated the average ROI for each director. Since there were a lot of directors that had only directed one movie (at least in our dataset), they automatically received a very high ROI. So we assigned these directors a score of 0, and only considered directors that had made multiple movies. This raised another issue: For a pair of directors, if they had similar average returns but one of them produced those returns with a much lower budget, that director was more valuable. So the director score was calculated as the ratio of the average ROI to the average budget of each director.

$$Director\ Score\ = \frac{Avg\ ROI\ of\ director}{Avg\ budget\ of\ director}$$

Actors: Actor scores were calculated similar to director scores, but with a caveat. Initially, we considered all the actors in each movie, but we decided to go only with the start of the movie so that the correlation between actor score and director score was maintained (Reasons why are explained in the Results section.)

$$Actor\ Score\ = \frac{Avg\ ROI\ of\ actor}{Avg\ budget\ of\ actor}$$

3.3 ALGORITHMS USED

The columns left at the end which were used in the classifier are Ratings, Budget, Runtime, Year, Genre Score, Director Score, Actor Score and Number of Votes. Feature scaling was used to normalize the values for the classifier. Then we discretized the ROI based on the following split points:

- **0 - 1**: Indicates that the movie was a loss making movie - so this category is **BAD**
- **1 - 3**: Indicates that the movie made a small profit - this category is **OK**
- **3 - 8**: Indicates that the movie made a good amount of profit - this category is **GOOD**
- **> 8**: Indicates that the movie was very profitable - this category is **EXCELLENT**

We experimented with the following classifiers using Leave one out k-fold cross validation to evaluate the accuracy of the model by selecting a subset of the features:

- Decision Tree Classifier
- Support Vector Machine
- Naive Bayes
- Neural Network MLP
- **Random Forest Classifier**

4. RESULTS

After experimenting with various subsets of the features, and several classification models, these are the results we observed. The best performing classifier was the Random Forest Classifier.

4.1 OBSERVATIONS

Accuracy with revenue as a feature - 0.95

- Initially, we included Revenue as a column in our classifier, but we realized that if both budget and revenue are present, it is really easy for the classifier to predict the ROI and doesn't make sense for our model, because the revenue is not going to be available to the film studios as a metric for prediction.

Accuracy with Genre Score - 0.66

- It was observed that the genre score had almost no correlation to the profitability of the film. In fact, when we removed it as a column, the accuracy improved to 0.69. As noted in Fig 2 - apart from Film-Noir and News all the genres are pretty much in the ROI range of 2-3. This shows that genre really does not have a high impact on a movie's ROI and can be ignored.

Accuracy without Genre Score - 0.68

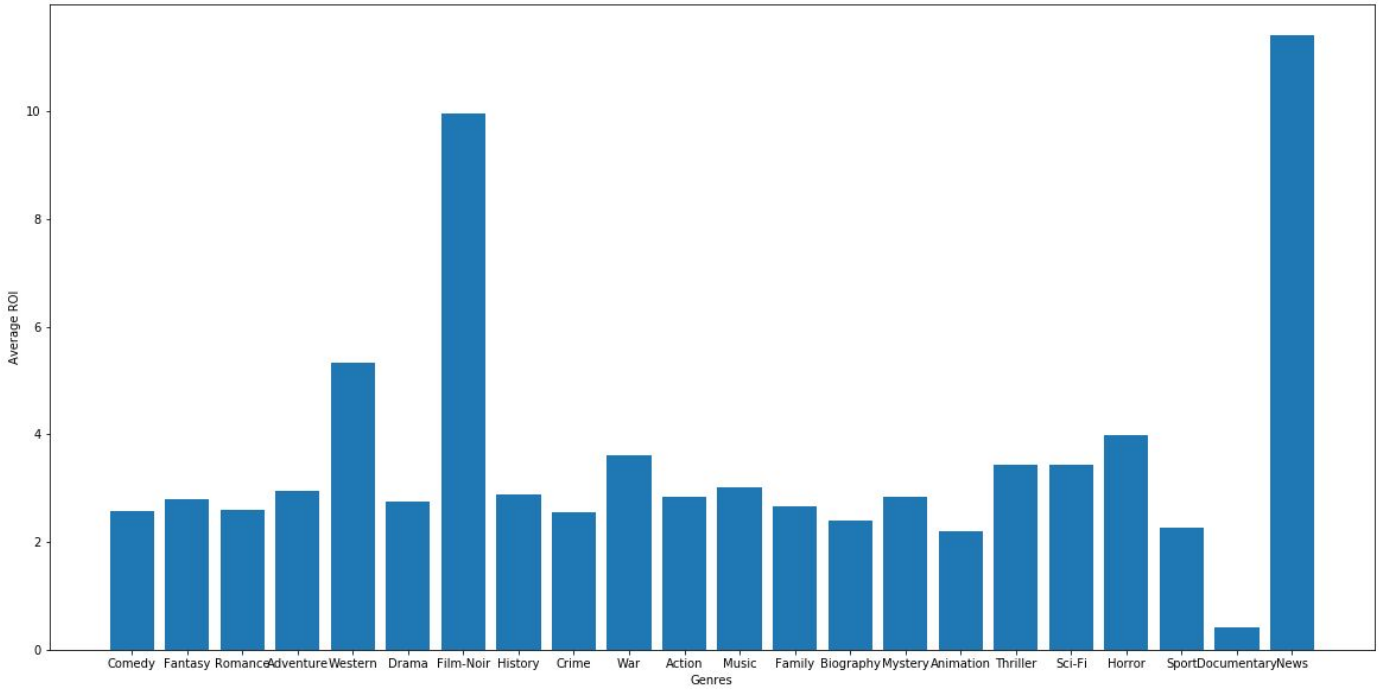


Fig 2: Average ROI for each genre

The list of most important features to the classification model was as follows:

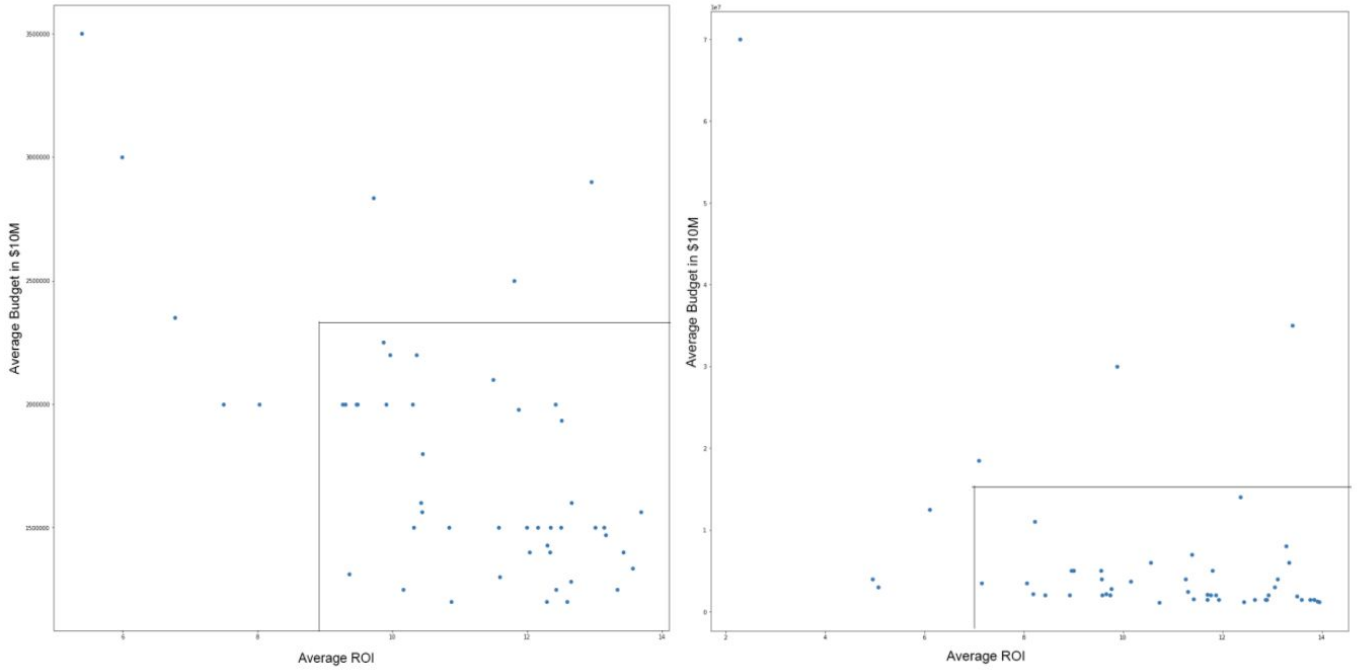
1. **Budget** - Had a strong negative correlation with ROI and was the most important feature. Removing budget from the columns gave an accuracy of 0.55

2. **Director Score** - Had a weak positive correlation and a decent impact on the ROI of the movie. Removing the director score dropped the accuracy to 0.62
3. **Number of IMDb votes** - Movies with higher number of votes tend to be more indicative of a commercially successful movie, regardless of the critical acclaim (which is indicated by the IMDb ratings). Removing the number of votes dropped the accuracy to 0.62
4. **Runtime** - As expected, runtime was similar to budget and had a negative correlation to the movie's profitability. Removing the runtime also dropped the accuracy to 0.62
5. **Actor Score** - Surprisingly, actor score was not indicative of a movie's profitability. This is counter intuitive because they are usually the face of a movie and always a strong consideration for a consumer to watch a particular movie. We can infer that the gains received by hiring a super star actor are probably undercut by the consequent substantial increase in the budget, thus not affecting the actual profitability of the movie. The obvious exceptions to this case are when the budget of the movies are incredibly high. Removing the actor score did not change the accuracy of the model.

4.2 CONCLUSIONS

Our final model using Random Forest Classifier, after performing LeaveOneOut k-fold cross validation had an accuracy of 68%.
The classifier itself was very quick because the final size of our dataset was 11,310. This is also why we decided to implement the LeaveOneOut algorithm, because it wouldn't take too long to perform it (about 20 minutes).

- Based on the findings, Fig 3 shows the initial scatter plot of the top 50 most reliable directors, with Fig 4 showing the more reliable ones (Higher average ROI and lower average budget means the lower right quadrant).
- The top 50 recommended directors and actors are stored in the top_50.txt file.
- Fig 5 and Fig 6 show a similar scatter plot for actors, although based on the findings, they do not make much of an impact on profitability.
- Genres do not have a significant impact on the ROI of a movie, however the top 3 genres with highest average ROI, after removing noise were Film Noir, Western and Horror.



Average ROI and budgets: Fig 3 (directors) and Fig 5 (actors)

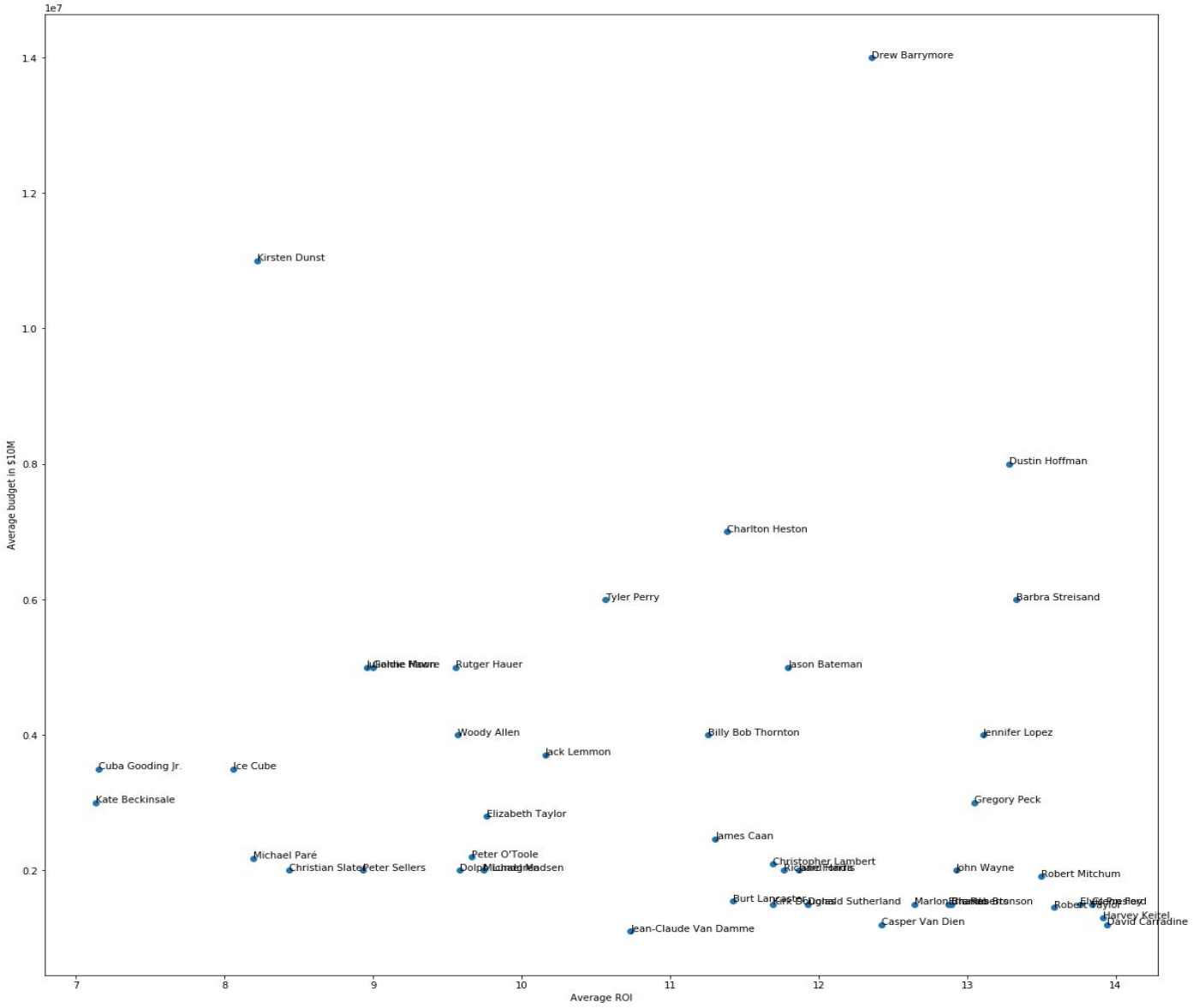


Fig 4: Most reliable actors

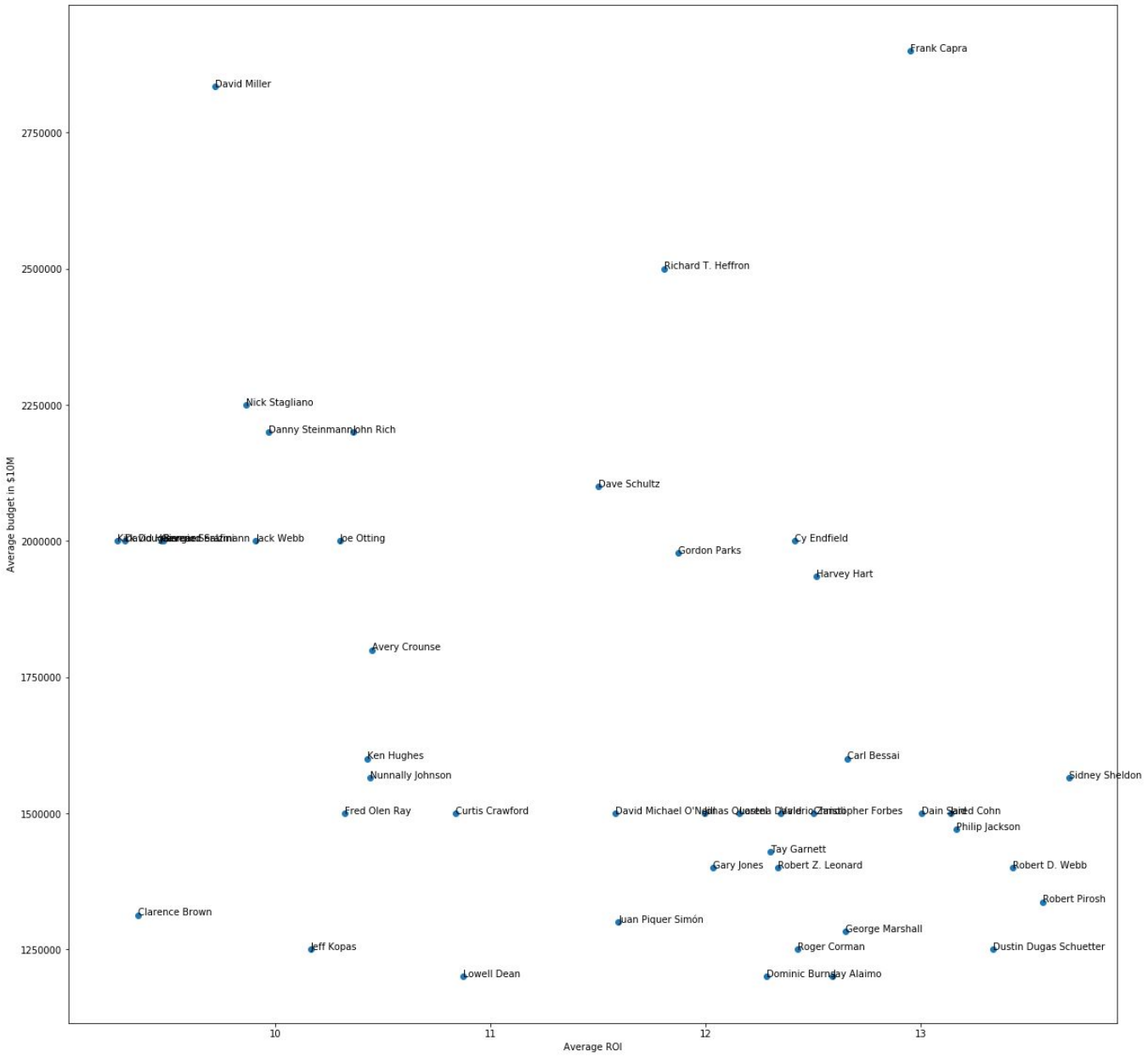


Fig 6: Most reliable directors

5. REFERENCES

[1] Vitelli, Matt. "Predicting box office revenue for movies." http://snap.stanford.edu/class/cs224w-2015/projects_2015/Predicting_Box_Office_Revenue_for_Movies.pdf

[2] Movie Exploration - https://rpubs.com/mm-c/movies