SVKM'S NMIMS Deemed to be UNIVERSITY

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING

IBM MINOR PROJECT ON:

Predicting The Success of an Upcoming Movie

PROJECT BY:

Concept Crew

TEAM MEMBERS:

Koustubh Sharma	MBA TECH IT I229 (70411118033)	**Manthan Gandhi**	MBA TECH IT I214(70411118017)
Ravi Sista	MBA TECH IT I233 (70411118037)	**Vishwa Babariya**	BTECH CS B204(70021118004)
Sashank Yeri	BTECH CS B256(70021118064)	**Divya Pokkunuri**	BTECH CS C225(70061118030)

Index

Background

Predicting a movie's opening success is a difficult problem, since it does not always depend on its quality only. External factors such as competing movies, time of the year and even weather influence the success as these factors impact the BoxOffice sales for the moving opening. Nevertheless, predicting a movie's opening success in terms of BoxOffice ticket sales is essential for a movie studio, in order to plan its cost and make the work profitable. We introduce a simple solution for predicting movie success in terms of financial success and viewer recipients. As a result, this approach achieved decent estimations, allowing theatre planning to a certain extent, even for small studios. So, the prediction of movie success is of great importance to the industry. Machine learning algorithms are widely used to make predictions such as growth in the stock market, demand for products, etc. This paper presents a detailed study of Adaboost, SVM Logistic Regression, Naïve Bayes Classifier and K-Nearest Neighbours on IMDbdata to predict movie box office.

Thousands of films are released every year. Since the 1920s, the film industry has grossed more money every year than that of any other country. Cinema is a multi-billion-dollar industry where even individual films earn over a billion dollars. Large production houses control most of the film industry, with billions of dollars spent on advertisements alone. Advertising campaigns contribute heavily to the total budget of the movies. Sometimes the investment results in heavy losses to the producers. Warner Brothers, one of the largest production houses, had a fall in their revenues last year,

despite the inflation and the increased number of movies released. If it was somehow possible to know beforehand the likelihood of success of the movies, the production houses could adjust the release of their movies to gain maximum profit. They could use the predictions to know when the market is dull and when it is not. This shows a dire need for such software to be developed. Many have tried to accomplish this goal of predicting movie revenues. Techniques such as social media sentiment analysis have been used in the past. None of the studies thus far have succeeded in suggesting.

While the development of this project, we have opted Python as the programming language. Python libraries that will be used for the development of this program are:
Numpy, Pandas, Scikit Learn, Seaborn and Random. For the exploratory analysis of the data, we have used Tableau and Python.

We have used Visual Studio as the integrated development environment for the program. Reason for using Visual Studio Code is because its code extensions cover more than just programming language capabilities:

- Keymaps allow users already familiar with Atom, Sublime Text, Emacs, Vim, PyCharm, or other environments to feel at home.

- Themes customize the UI whether you like coding in the light, dark, or something more colourful.

- Language packs provide a localized experience.

We used Tableau for EDA as EDA is the step that is performed before doing the machine learning models and it provides valuable insights on the current state of the data. It may throw light on the code errors in the Data pipeline which resulted in the data and also helps in visualizing the outliers in your data.

There are many ways to perform EDA, the most common way is using libraries such as matplotlib, Seaborn etc. We prefer to use Tableau to perform EDA and then if required use the libraries to get any visualizations which are not available as part of the software.

Requirements

Hardware Requirements: For Visual Studio Code

We recommend:

- 1.6 GHz or faster processor
- 1 GB of RAM

Platforms:

VS Code has been tested on the following platforms:

- OS X Yosemite (10.10+)
- Windows 7 (with .NET Framework 4.5.2), 8.0, 8.1 and 10 (32-bit and 64-bit)
- Linux (Debian): Ubuntu Desktop 16.04, Debian 9
- Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 8, Fedora 24

Additional Windows requirements:

Microsoft .NET Framework 4.5.2 is required for VS Code. If you are using Windows 7, please make sure .NET Framework 4.5.2 is installed.

Additional Linux requirements:

- GLIBC version 2.15 or later
- GLIBCXX version 3.4.21 or later

Hardware Requirements: For Tableau

Windows

- Microsoft Windows 7 or newer (x64)
- 2 GB memory
- 1.5 GB minimum free disk space
- CPUs must support SSE4.2 and POPCNT instruction sets

Mac

- macOS High Sierra 10.13, macOS Mojave 10.14 and macOS Catalina 10.15
- Intel processors
- 1.5 GB minimum free disk space
- CPUs must support SSE4.2 and POPCNT instruction sets

Virtual Environments:

- Citrix environments, Microsoft Hyper-V, Parallels, VMware, Microsoft Azure and Amazon EC2.
- All of Tableau's products operate in virtualized environments when they are configured with the proper underlying Windows operating system and minimum hardware requirements. CPUs must support SSE4.2 and POPCNT instruction sets so any Processor Compatibility mode must be disabled.

Development

Task Description:

The objective of our project is to predict the success rate of a movie based on attributes such as the actors involved, directors, year in which they were released, movie genre, total runtime of movie, user rating, number of votes, total revenue generated by movie, the overall metascore, age of the users watching and recording the votes or rating, the geographical areas where movie was released, any other influences such as political movements, ongoing trends and so on.

Our most important and difficult tasks was to get the dataset for such kind of prediction and analysis. We had to look for datasets available on the web, as we would not be able to collect historical data about past movies for our project. That Following are the steps that we've performed:

- Searched for available datasets to support our idea and thoroughly scrutinized them, to get the most suitable dataset for our idea.

- Shortlisted few datasets, we picked the most suitable dataset for our project.

- Pruned the data which we required, most suitable for our prediction and analysis.

- Performed Exploratory data analysis using both, Python & Tableau.

- Collected ground truth data and saved in the csv format. We also binarized

our attributes and used an additional success column, based on the average revenue, rating and votes received by the movie.

- We used this data as an input to the machine learning and data mining algorithms for prediction of movie success rate.

- We split the data into training and testing data.

- The machine learning algorithms we have used are Logistic Regression, Linear SVM, K-Nearest Neighbor, Naïve Bayes Classifier and Adaboost.

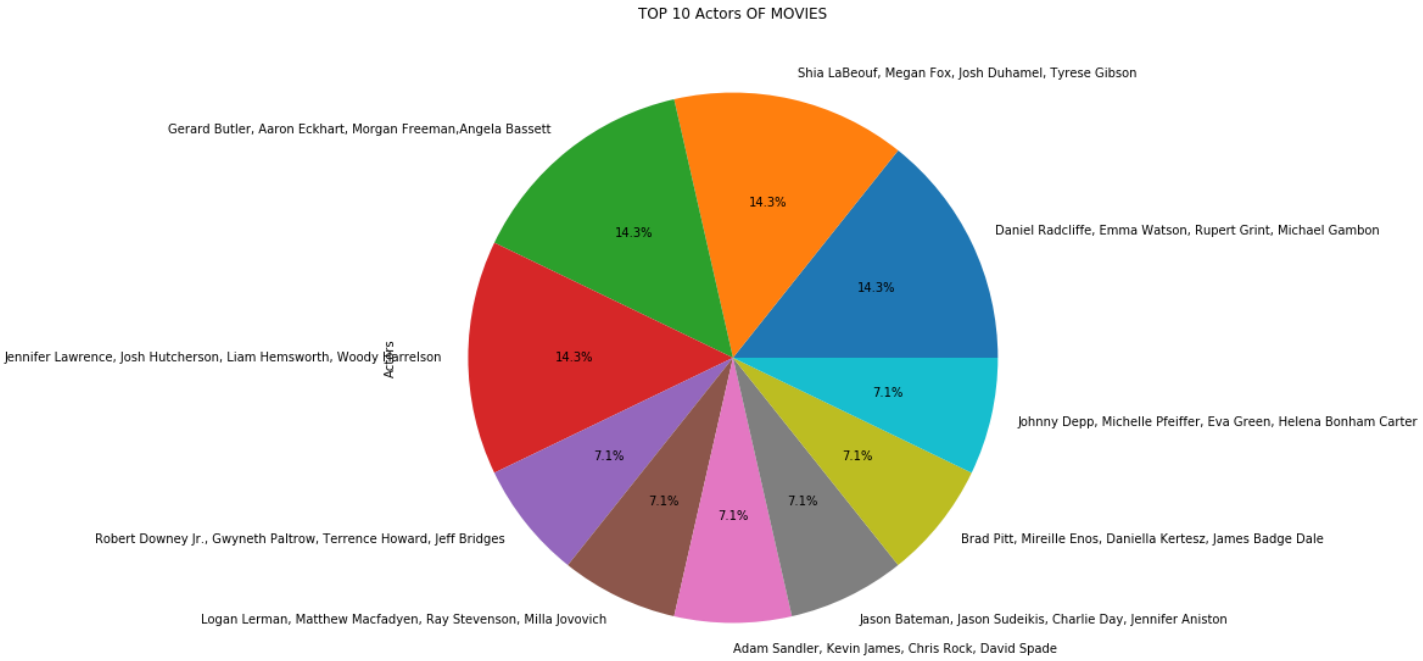
- We have computed the results of our algorithms by means of confusion matrix, accuracy, recall, precision rate and ROC curve.

- We have also used this dataset for analysis of effect of various attributes on the success rate of movie. These attributes include rating, votes, actors, directors, revenue and metascore.

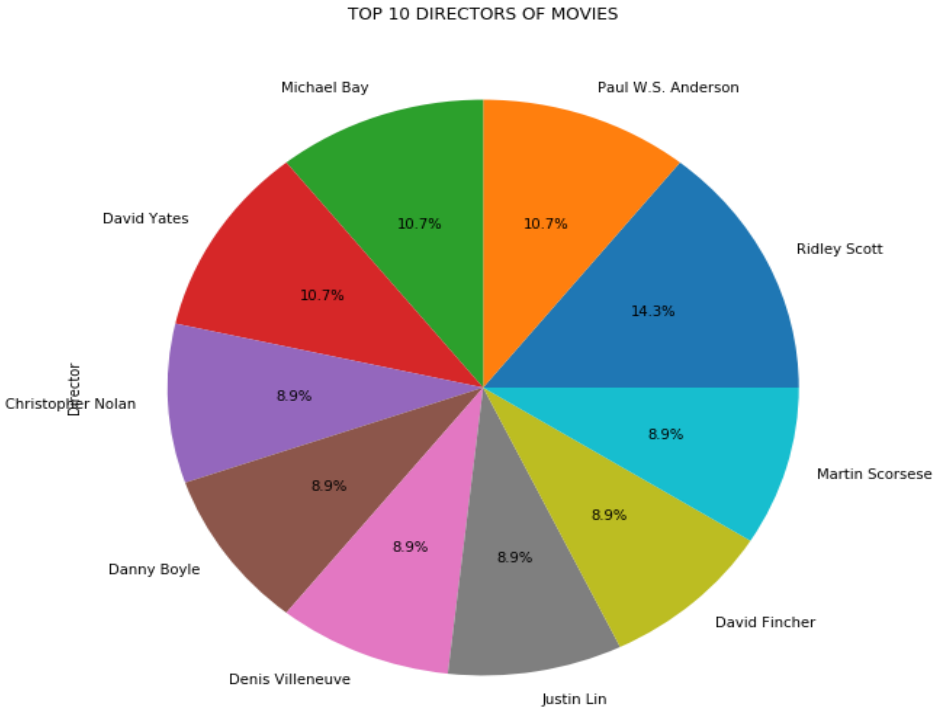
Visualization:

➢ EDA with Python:

- **Top 10 Directors of Movies:**

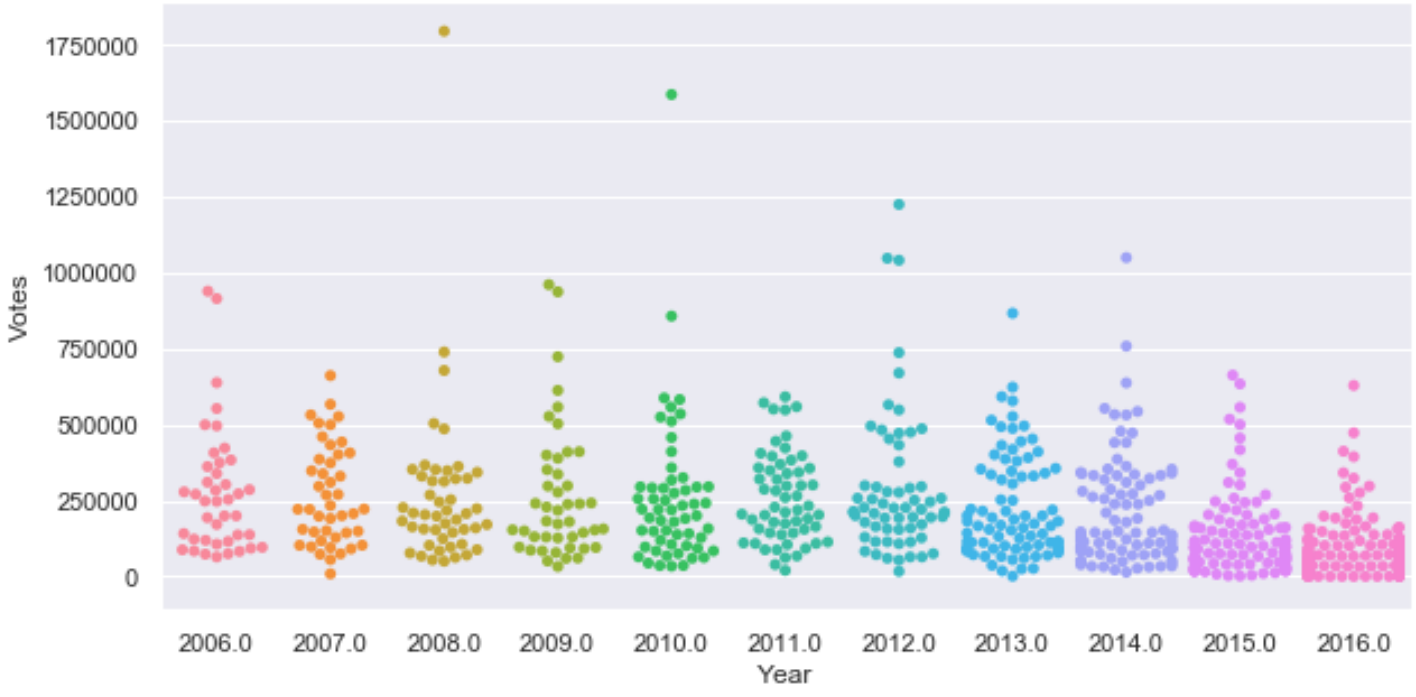


- **Top 10 Actors of movies:**

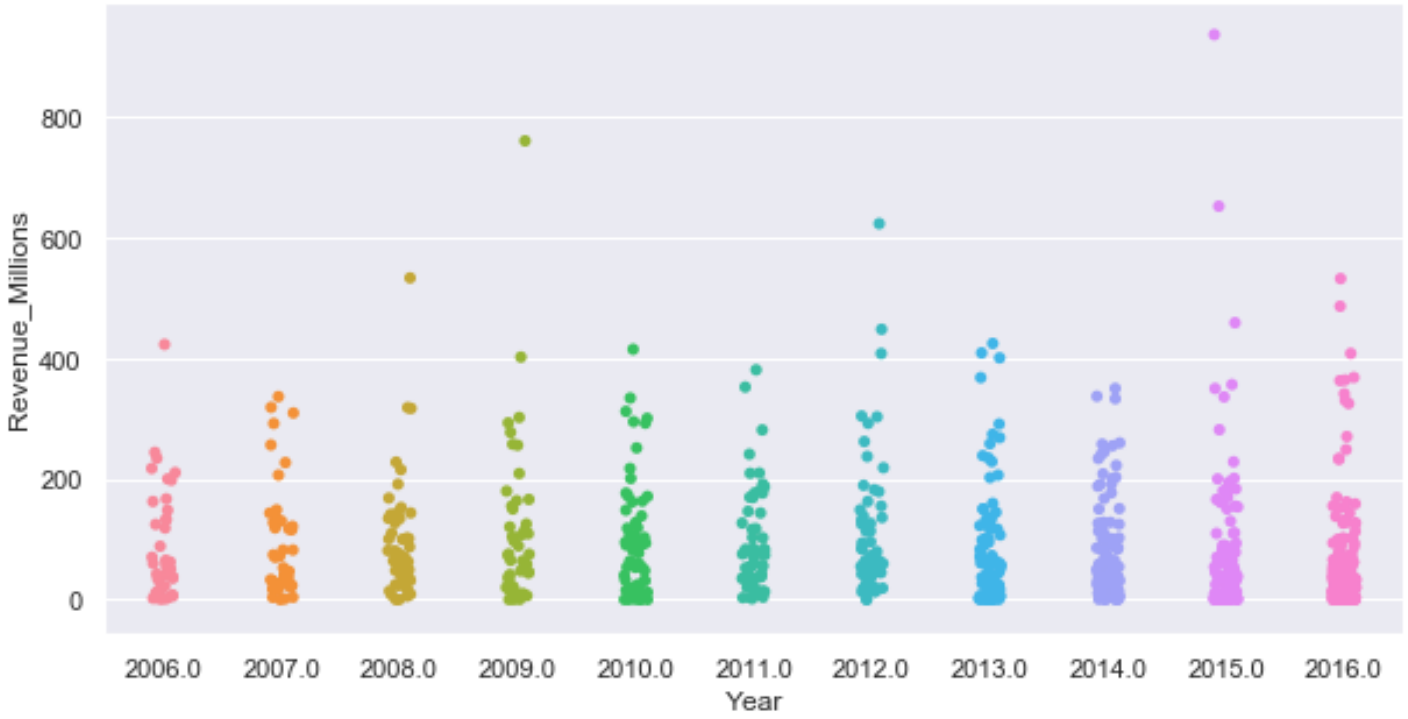


- **Year wise analysis of Ratings:**

- **Year wise analysis of Votes:**

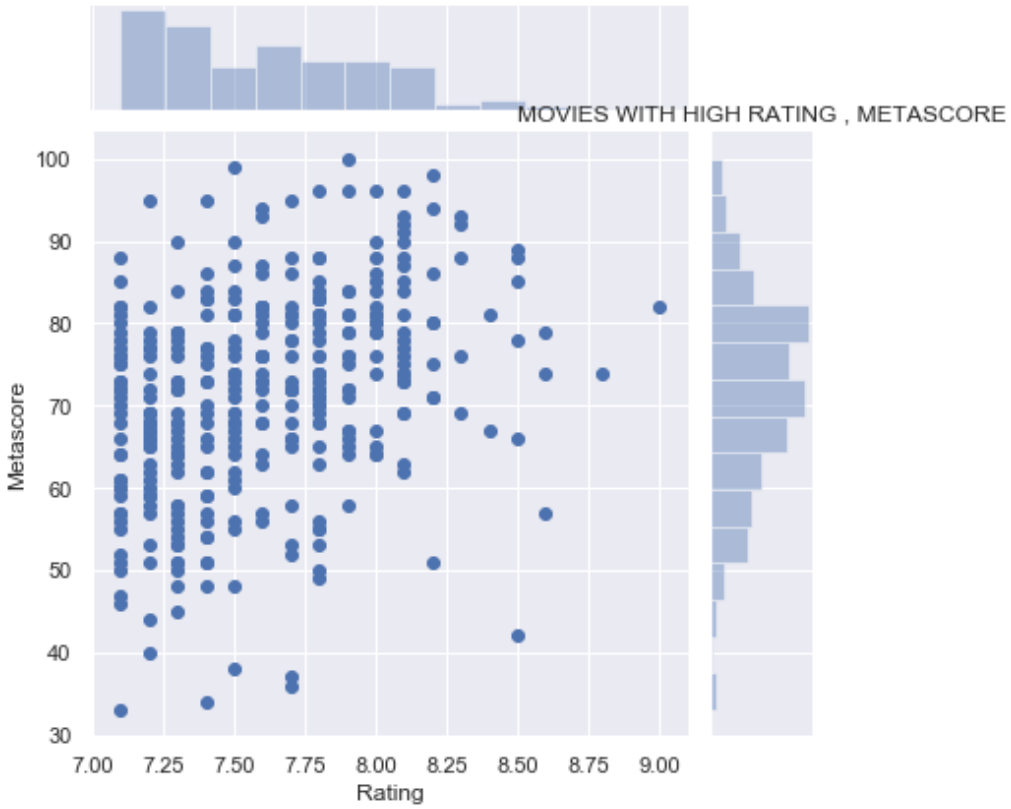


- **Year wise analysis of Revenue:**

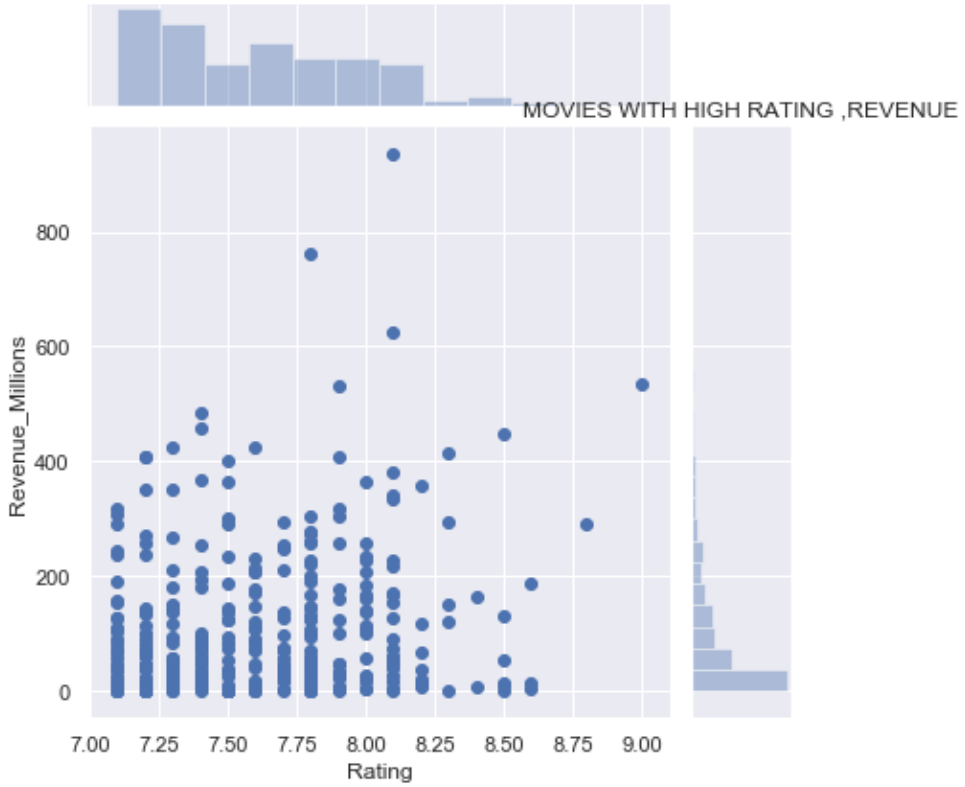


- **Year wise analysis of Metascore:**

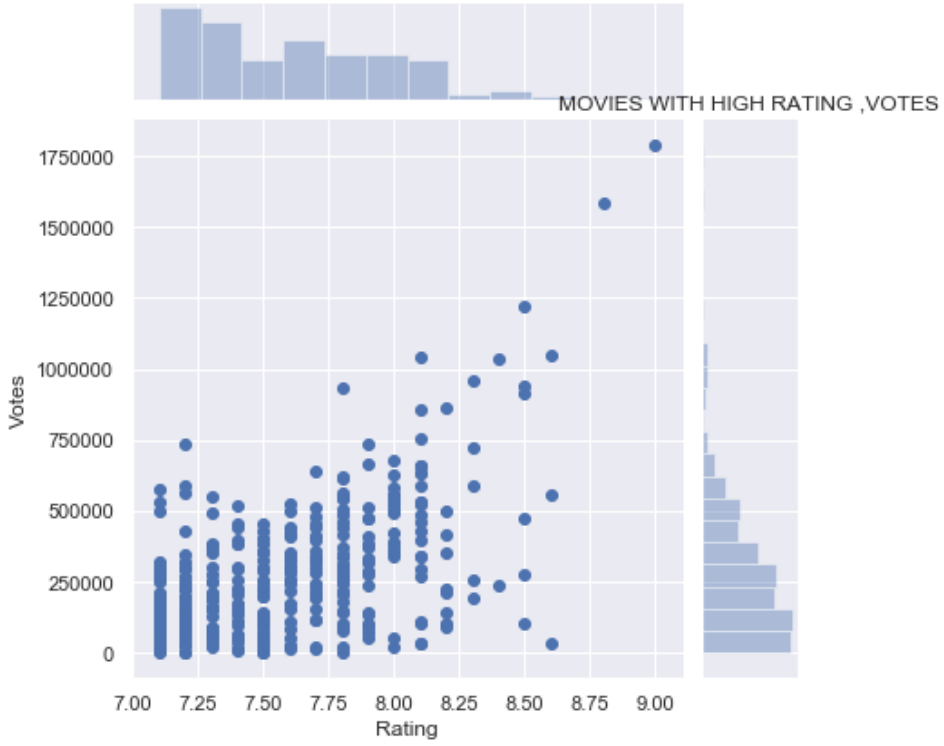
- **Movies with high rating and Metascore:**



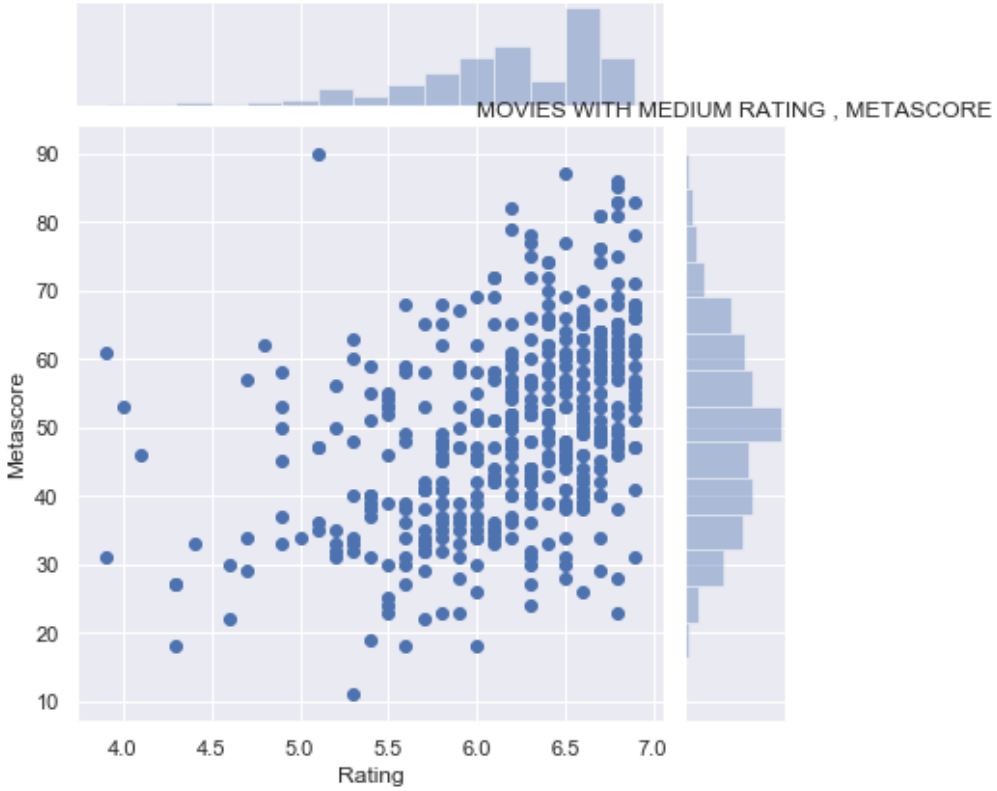
- **Movies with high rating & revenue:**



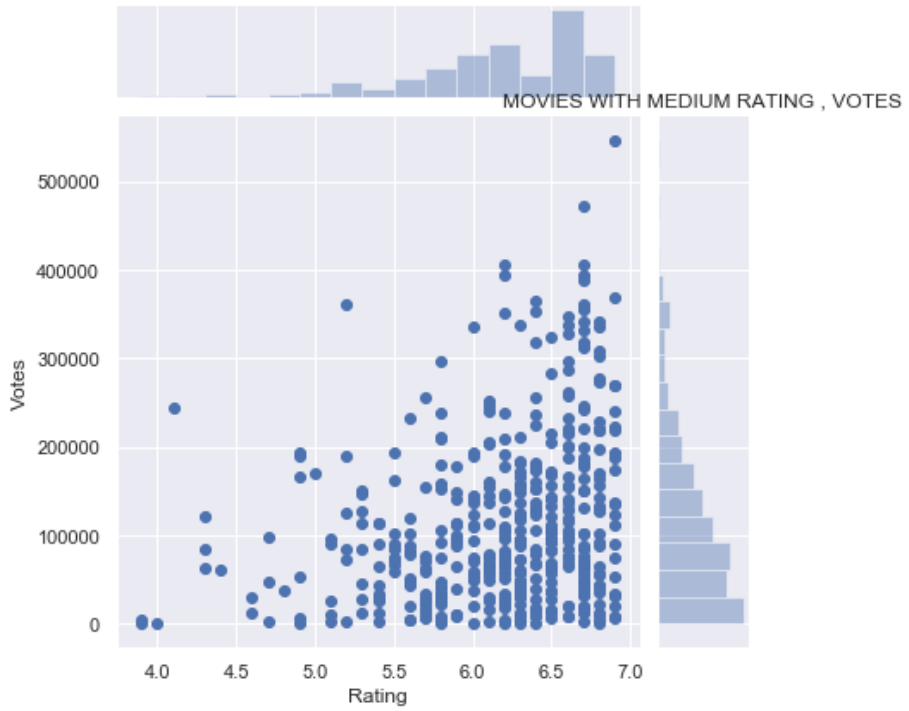
- **Movies with high rating & Votes:**



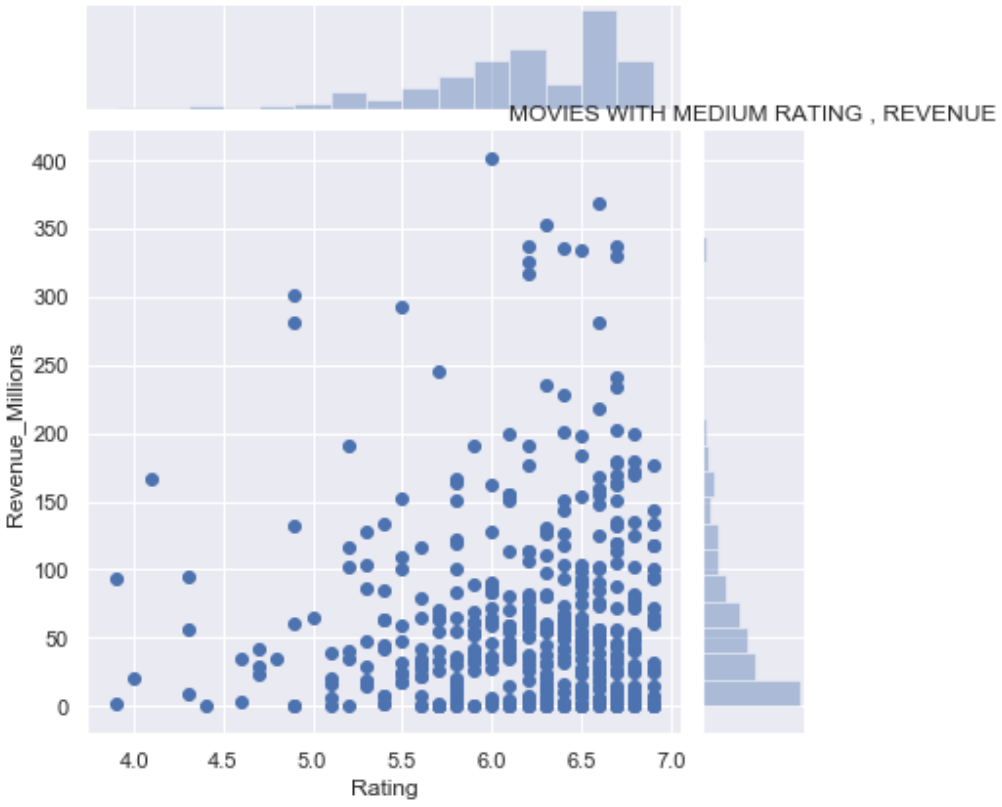
- **Movies with medium rating and Metascore:**



- **Movies with medium rating & revenue:**



- **Movies with medium rating & Votes:**



EDA with Tableau:

Tableau is a Business Intelligence tool for visually analyzing the data. Users can create and distribute an interactive and shareable dashboard, which depict the trends, variations, and density of the data in the form of graphs and charts. The software allows data blending and real-time collaboration, which makes it unique.

We are using this powerful software to create interactive dashboards which provide insights on the data.

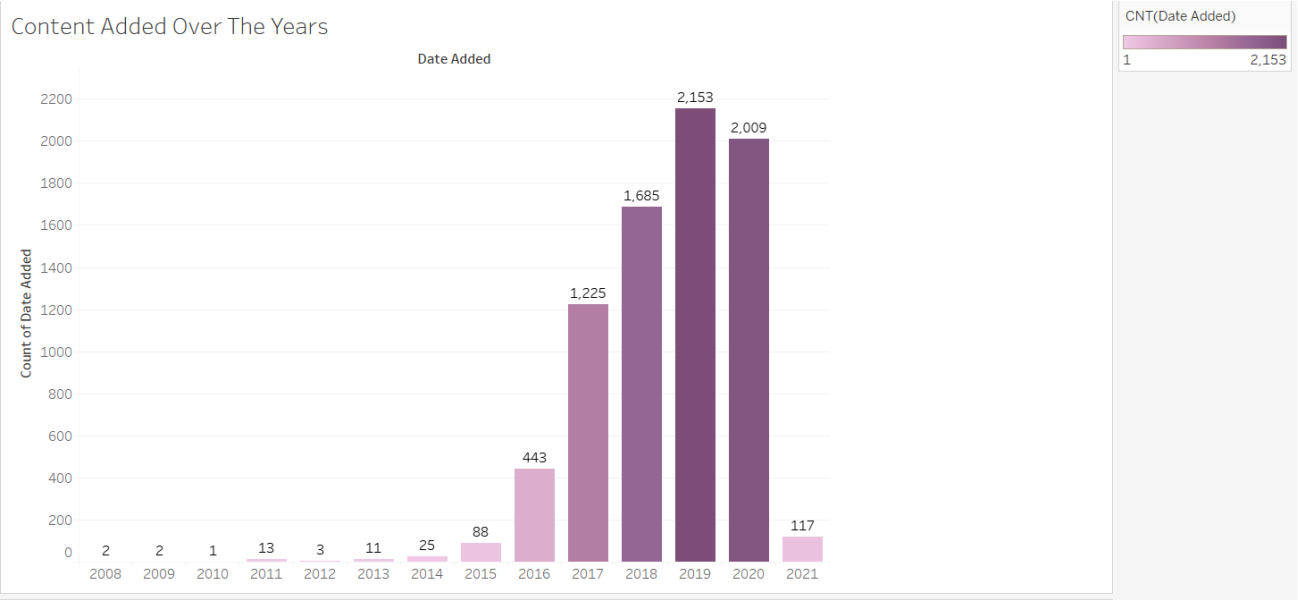
Some of the interesting questions (tasks) which can be performed on our dataset:

- Understanding what content is available in different countries
- Network analysis of Actors / Directors and find interesting insights
- Should applications like Netflix focus more on TV Shows or Movies?
- Identifying similar content by matching text-based features
- Who are the most successful directors/actors in the industry?

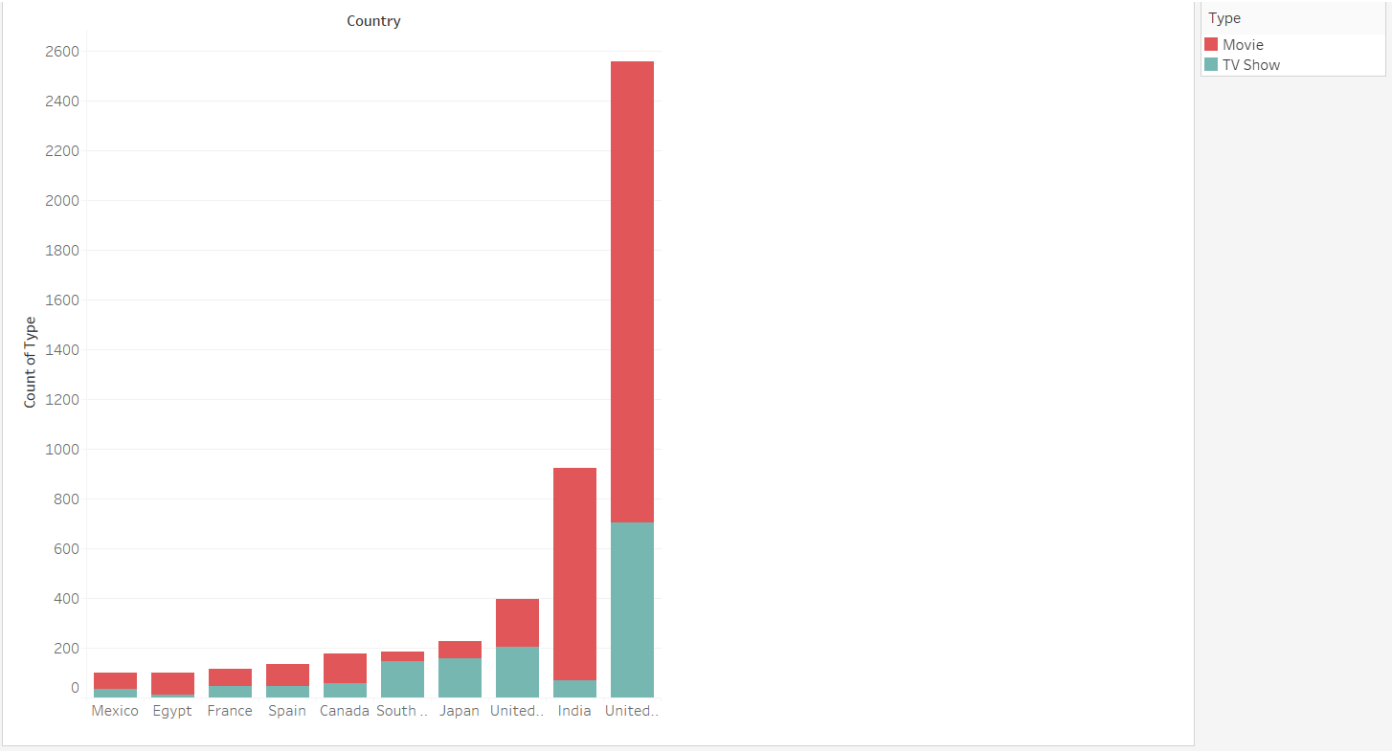
Insights Gained: We have carried EDA on two datasets, one is the dataset used for predictions and the other is Netflix IMdb data which helps know the type of content available on such content platforms. This analysis was done to understand the preferred content nowadays.

- There are almost half the number of TV Shows than there are Movies on applications like Netflix, 69.1% is Movies while 30.9% of content available is of Tv Shows. Further, a huge increase in TV Shows is expected in the coming years.

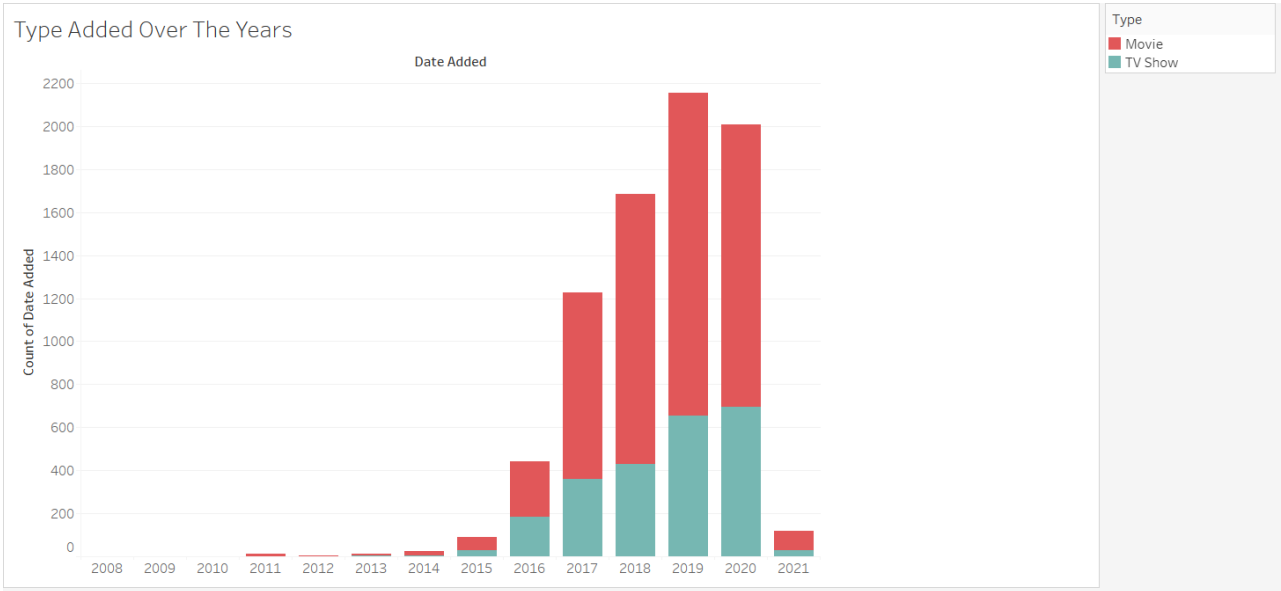
- When we see Trend of Content added over the years, there is a steep increase in movies since 2000. The years 2017-2020 have shown a major increase in content. Year 2019 was the year when the highest content was added on such content platforms. Then it decreased which most probably is due to the pandemic when content production was decreased.



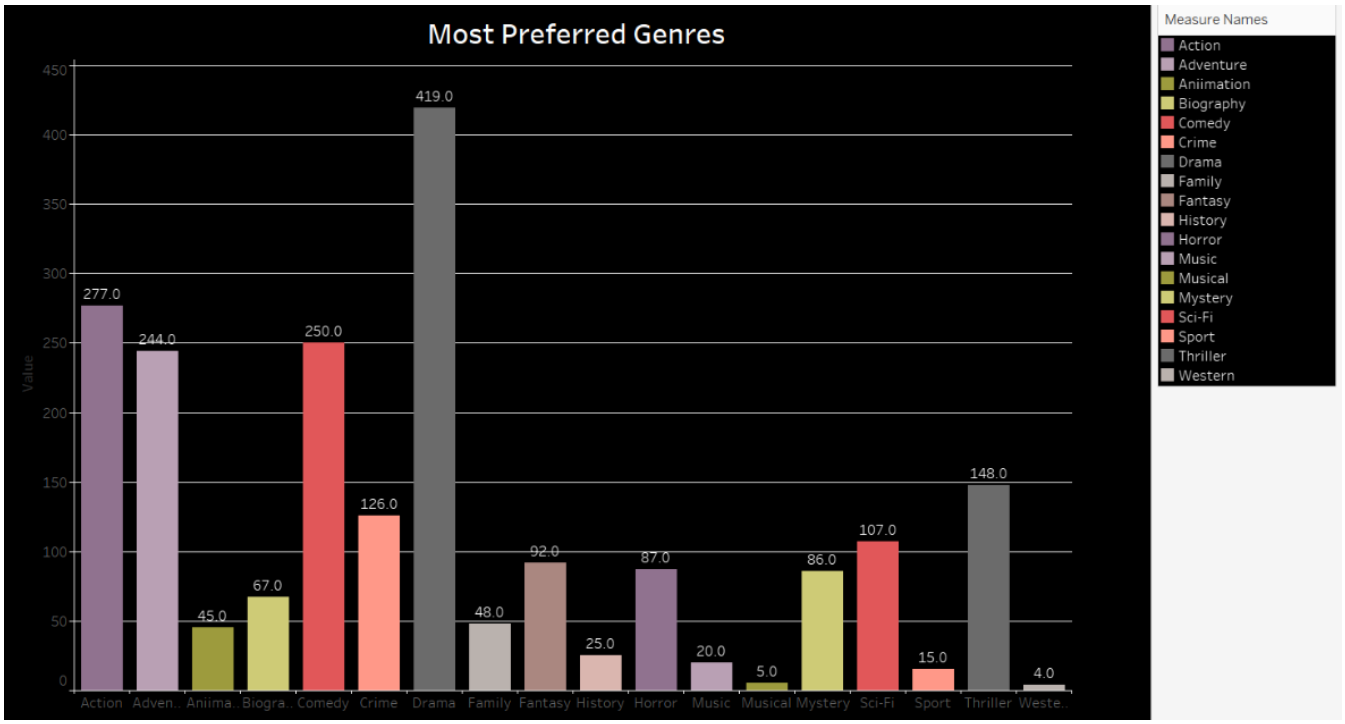
- We have analyzed the top countries with highest content. We have plotted a bar chart for Top 10 countries with the highest content. United States has the highest number of content available followed by India, United Kingdom, and Japan and so on.



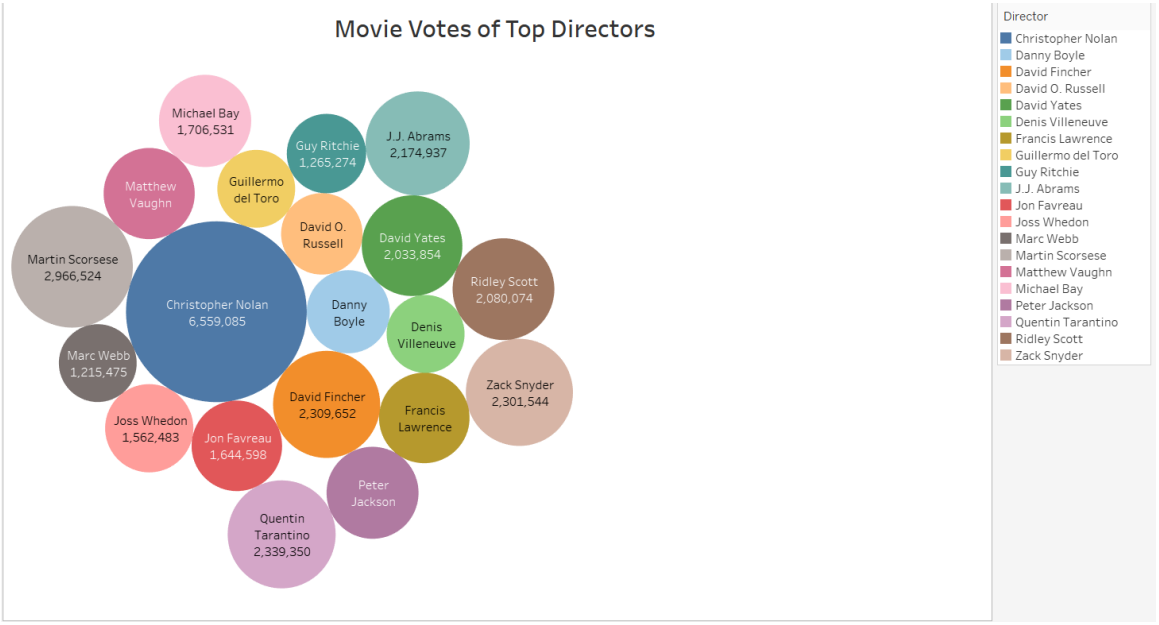
- Year 2019 was the year when people preferred Movies and then the preference shifted to TV Shows, 2020 being the year when most TV Shows were added.



- When analyzed the most preferred genres among the audiences, comedy and drama are on top. A good amount of audience also prefers action and adventure together.

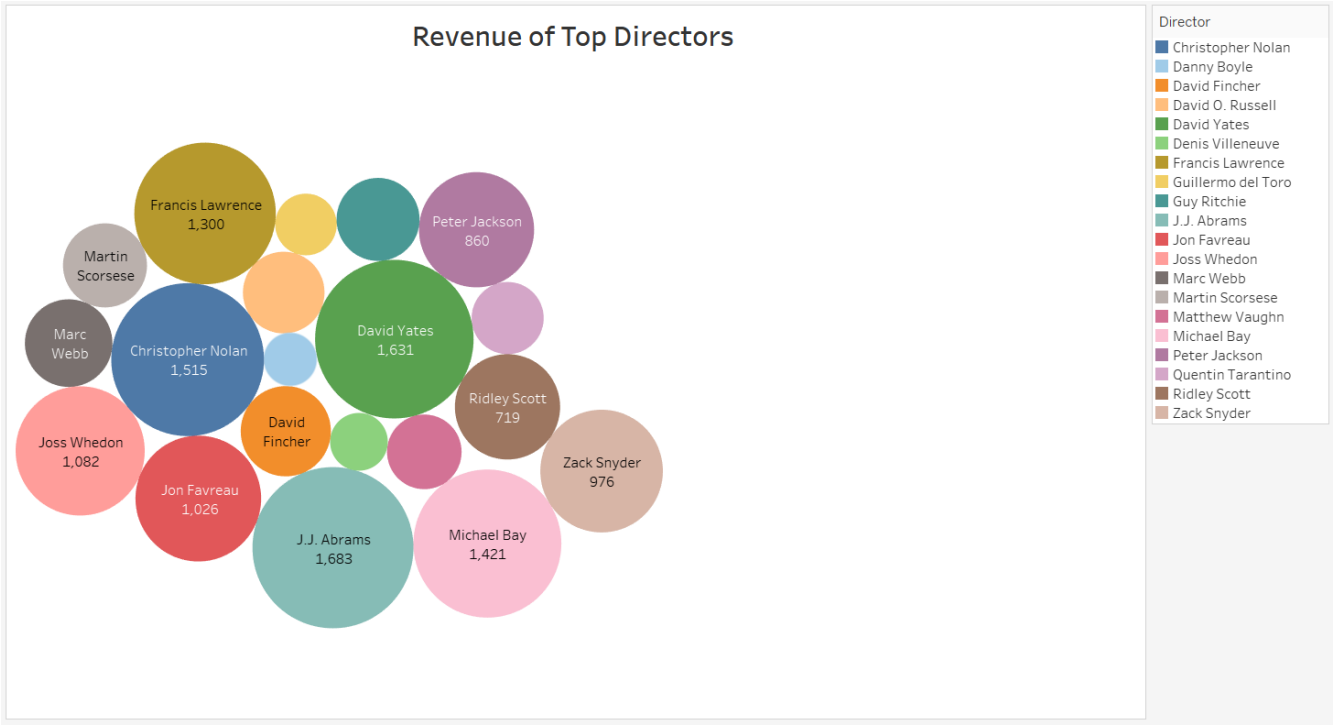


- Analysis of directors is done on the basis of the number of successful movies they had, the revenue generated from their movies, and the number of votes their movies acquired. When it comes the highest votes, we have director Christopher Nolan - 6,559,085 votes respectively. The number of successful movies directed by him is also high.



- Top 30 successful director analysis was done - J.J. Abrams topping the list with, David Yates and so on. The success of a movie depends very much on directors, and for that knowing the trend of the top directors, their movies, revenue, and the ratings their movies acquire need to be considered.

- The director whose movies generated highest revenues is - J.J. Abrams, the top in the list of successful directors.



Major Challenges:

➢ Dataset Description:

- Some key attributes like genre were comma separated values in a CSV file.
- Converting the above data into binary values for the model and other data cleaning process required some serious effort.
- Implementation of cross-validation from scratch without using external libraries to extract all relevant information required several brainstorming sessions.

➢ Visualization

- To get the details of visualization libraries in python a detailed investigation was required.
- To Finalize the key attributes with which movie data needed to be analyzed took a lot of time.
- Most of the attributes in the dataset were too related to each other. Dividing them to separate entities to infer useful information was a challenge.

Experiments:

- **Libraries used:**

  - **For data analysis operation**
    - numpy
    - panda

  - **For splitting dataset into testing and training dataset**
    - train_test_split

  - **Standardizing features**
    - StandardScaler

  - **Adaboost**
    - AdaBoostClassifier
    - DecisionTreeClassifier

  - **KNN**
    - KNeighborsClassifier

  - **LogisticRegression**

  - **Naïve Bayes Classifier**
    - GaussianNB

- **SVM**

- **Implementing various metrics**

    - confusion_matrix
    - accuracy_score
    - precision_score
    - recall_score
    - roc_curve, auc

- **Visualization**

    - matplotlib.pyplot
    - seaborn

- **Random Values**

    - Random

❖ **Dataset to import:**

- IMBD-Movie-Data.csv

❖ **Dataset Description:**

1) Rank - Rank of the movie

2) Title - Title of the movie

3) Genre - Genre of the movie

4) Description - Description of the movie

5) Director - Director of the movie

6) Actors - Actors of the movie

7) Year - Year of the movie

8) Runtime (Minutes) - Runtime of the movie

9) Rating - Rating of the movie

10) Votes - Votes of the movie

11) Revenue (Millions) - Revenue of the movie

12) Metascore - Metascore of the movie

❖ **Evaluation Metrics:**

To visualize the performance of an algorithm, typically a supervised learning confusion matrix is used. Also, known as error matrix, each column of the confusion matrix signifies an instance of a predicted class, and each row signifies an instance of the actual class.

Groundtruth	Prediction $\hat{y}=1$	Prediction $\hat{y}=0$
$y=1$	True-positive	False- Negative
$y=0$	False-positive	True-negative

The above table is an example of a confusion matrix. If the prediction and ground truth are equal, then it is either True-positive or True negative based on the classification labels. If the prediction is not equal to the ground truth, then it is either False- positive or False-Negative based on the classification labels. From the table we are calculating the Accuracy, Precision and Recall. We are also determining the ROC curve to evaluate the performance of the algorithm.

Classification Algorithms:

❖ K Nearest Neighbors:

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

- Ease to interpret output
- Calculation time
- Predictive Power

KNN algorithm fairs across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time.

Breaking it Down: Pseudo Code of KNN:

We can implement a KNN model by following the below steps:

- Load the data
- Initialise the value of k
- For getting the predicted class, iterate from 1 to total number of training data points
  - Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.
  - Sort the calculated distances in ascending order based on distance values

- Get top k rows from the sorted array
- Get the most frequent class of these rows
- Return the predicted class

Accuracy: 0.8952

Precision: 0.8260

Recall: 0.51351

❖ **Logistic Regression:**

Logistic regression is a fundamental classification technique. It belongs to the group of linear classifiers and is somewhat similar to polynomial and linear regression. Logistic regression is fast and relatively uncomplicated, and it's convenient for you to interpret the results. Although it's essentially a method for binary classification, it can also be applied to multiclass problems.

Accuracy: 0.9

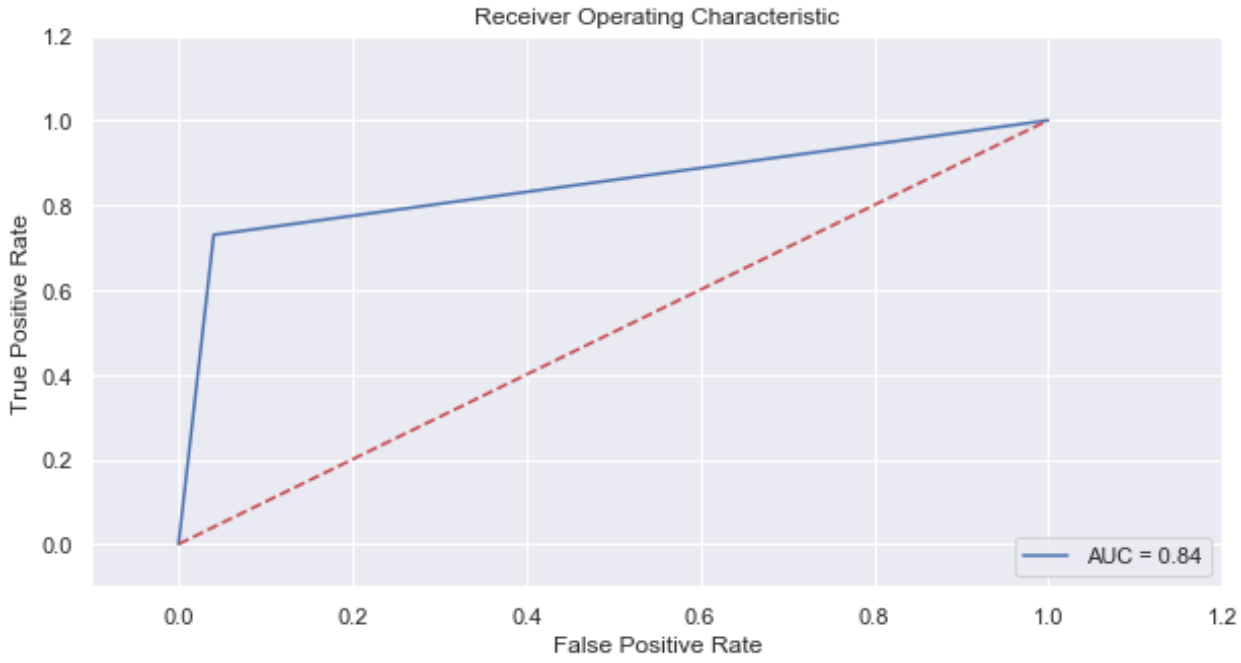
Precision: 0.7105

Recall: 0.7297

❖ **SVM:**

Support Vector Machine or SVM is a supervised and linear Machine Learning algorithm most commonly used for solving classification problems and is also referred to as Support Vector Classification. There is also a subset of SVM called SVR which stands for Support Vector Regression which uses the same principles to solve regression problems. SVM also supports the kernel method also called the kernel SVM which allows us to tackle non-linearity.

The objective of SVM is to draw a line that best separates the two classes of data points. SVM generates a line that can cleanly separate the two classes. How clean, you may ask. There are many possible ways of drawing a line that separates the two classes, however, in SVM, it is determined by the margins and the support vectors.



Accuracy: 0.919

Precision: 0.7941

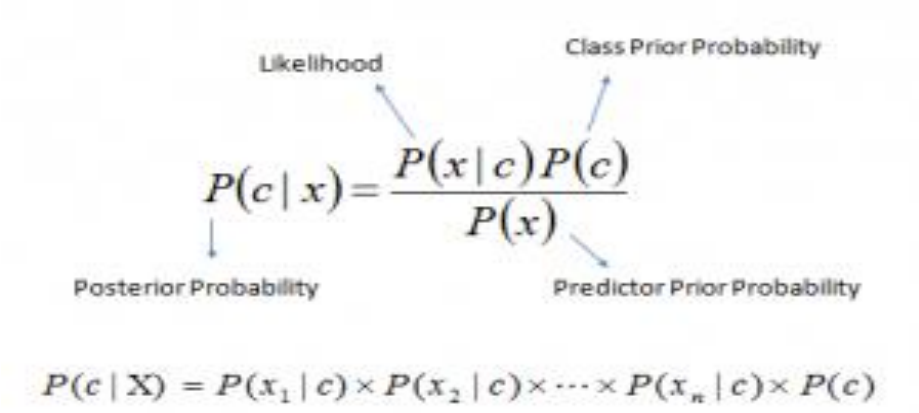
Recall: 0.7297

❖ **Naïve Bayes Classifier:**

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

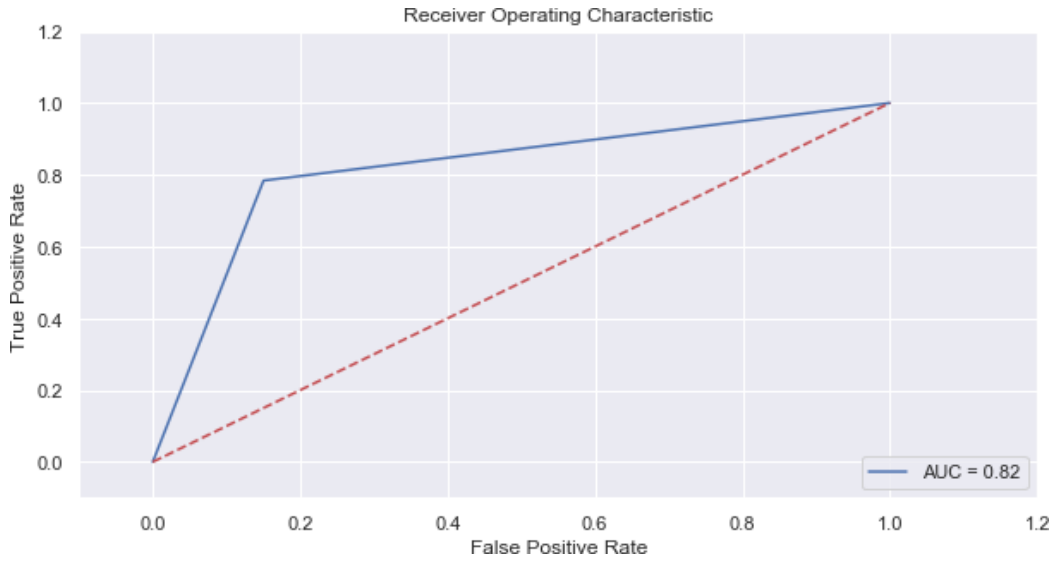
For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



Above, $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).

- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- *P(x)* is the prior probability of predictor.



Accuracy: 0.8380

Precision: 0.5272

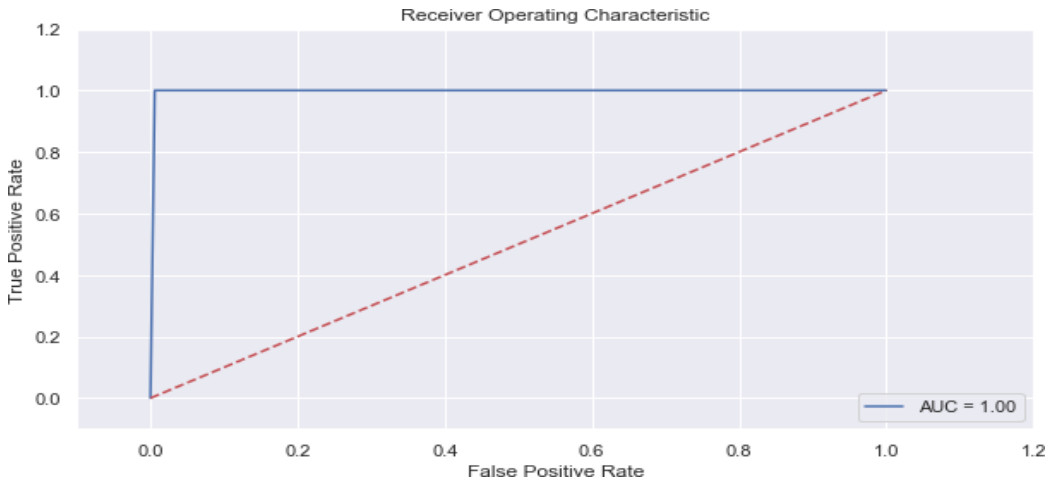
Recall: 0.7837

❖ **Adaboost Classifier:**

When nothing works, Boosting does. Nowadays many people use either XGBoost or LightGBM or CatBoost to win competitions at Kaggle or Hackathons. AdaBoost is the first steppingstone in the world of Boosting. AdaBoost is one of the first boosting algorithms to be adapted in solving practices. Adaboost helps you **combine multiple "weak classifiers" into a single "strong classifier"**. Here are some (fun) facts about Adaboost!

- The weak learners in AdaBoost are decision trees with a single split, called decision stumps.
- AdaBoost works by putting more weight on difficult to classify instances and less on those already handled well.
- AdaBoost algorithms can be used for both classification and regression problem.

After training a classifier at any level, ada-boost assigns weight to each training item. Misclassified item is assigned higher weight so that it appears in the training subset of next classifier with higher probability. After each classifier is trained, the weight is assigned to the classifier as well based on accuracy. The more accurate classifier is assigned higher weight so that it will have more impact in the outcome. A classifier with 50% accuracy is given a weight of zero, and a classifier with less than 50% accuracy is given negative weight.



Accuracy: 0.9952

Precision: 0.9736

Recall: 1.0

Conclusion & Future Development

We conducted Exploratory Data Analysis using Tableau. After building the models we found out that the success percentage for all models were nearly the same however the Adaboost with Decision Tree model had the highest accuracy in our case for predicting the movies success. A larger training set is the key to improving the performance of the model. We need to consider additional features such as geographic location, age of viewers and voters,

```
df = [['K Nearest Neighbor', 0.75, 0.8952, 0.8260, 0.5135],
      ['Logistic Regression', 0.83, 0.9, 0.7105, 0.7297],
      ['Support Vector Machine', 0.84, 0.9190, 0.7941, 0.7297],
      ['Naive Bayes', 0.82, 0.8380, 0.5272, 0.7837],
      ['Adaboost', 1, 0.9952, 0.9736, 1]]
```

```
df = pd.DataFrame(df, columns = ['Algorithm', 'AUC', 'Accuracy', 'Precision', 'Recall'])
```

```
df
```

	Algorithm	AUC	Accuracy	Precision	Recall
0	K Nearest Neighbor	0.75	0.8952	0.8260	0.5135
1	Logistic Regression	0.83	0.9000	0.7105	0.7297
2	Support Vector Machine	0.84	0.9190	0.7941	0.7297
3	Naive Bayes	0.82	0.8380	0.5272	0.7837
4	Adaboost	1.00	0.9952	0.9736	1.0000

current trends, news analysis, movie plot analysis and social networks data analysis could be done, and the information thus obtained could be added to the training set. We can also use Google trends result to improve the result.

```
df.sort_values(by = 'AUC', ascending = False, inplace = False)
```

	Algorithm	AUC	Accuracy	Precision	Recall
4	Adaboost	1.00	0.9952	0.9736	1.0000
2	Support Vector Machine	0.84	0.9190	0.7941	0.7297
1	Logistic Regression	0.83	0.9000	0.7105	0.7297
3	Naive Bayes	0.82	0.8380	0.5272	0.7837
0	K Nearest Neighbor	0.75	0.8952	0.8260	0.5135

```
df.sort_values(by = 'Precision', ascending = False, inplace = False)
```

	Algorithm	AUC	Accuracy	Precision	Recall
4	Adaboost	1.00	0.9952	0.9736	1.0000
0	K Nearest Neighbor	0.75	0.8952	0.8260	0.5135
2	Support Vector Machine	0.84	0.9190	0.7941	0.7297
1	Logistic Regression	0.83	0.9000	0.7105	0.7297
3	Naive Bayes	0.82	0.8380	0.5272	0.7837

```
df.sort_values(by = 'Recall', ascending = False, inplace = False)
```

	Algorithm	AUC	Accuracy	Precision	Recall
4	Adaboost	1.00	0.9952	0.9736	1.0000
3	Naive Bayes	0.82	0.8380	0.5272	0.7837
1	Logistic Regression	0.83	0.9000	0.7105	0.7297
2	Support Vector Machine	0.84	0.9190	0.7941	0.7297
0	K Nearest Neighbor	0.75	0.8952	0.8260	0.5135

References

[1] Darin Im, Minh Thao, Dang Nguyen, Predicting Movie Success in the U.S. market, Dept.Elect.Eng, Stanford Univ., California, December,2011

[2] Jiawei Han, Micheline Kamber, Jian Pei, Data Mining Concepts and Techniques, 3rd ed.MA:Elsevier, 2011, pp. 83 - 117

[3] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, 2nd. NewYork: Wiley, 1973

[4] Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates

[5] Christopher M. Bishop (2006), Pattern Recognition and Machine Learn -ing, Springer, p. 205.

[6] The International Movie Database (IMDb): https://www.kaggle.com/PromptCloudHQ/ imdb -data

[7] Freund, Y.: An adaptive version of the boost by majority algorithm. Machine Learning 43(3), 293 – 318 (2001) [8] Haiyi Zhang, Di Li Jodrey School of Computer Science Acadia University, Canada, Naïve Bayes Text Classifier (2007).