[](https://www.iss.nus.edu.sg/)

**Machine Learning:**

***Movie Success Predictor***

**Submitted By: Team #7**

|  |
| --- |
| **Student Name** |
| Aung Myo |
| Cheng Hao |
| Lim, Hean Soon |
| Lin, Wei |
| Liu, Yakun |
| Nyi Nyi Zin |

Table of Contents

[1 Introduction 3](#_Toc498077091)

[2 Business Use Cases 3](#_Toc498077092)

[2.1 Challenge 3](#_Toc498077093)

[2.2 Solution 3](#_Toc498077094)

[3 Machine Learning Techniques 3](#_Toc498077095)

[3.1 Decision Tree 3](#_Toc498077096)

[4 Data 4](#_Toc498077097)

[4.1 Data Collection 4](#_Toc498077098)

[4.2 Data Cleansing and Improvement (Challenges) 4](#_Toc498077099)

[4.3 Tool used to validate the datasets 5](#_Toc498077100)

[Attempt #2 Result: 6](#_Toc498077101)

[Attempt #3 Result: 9](#_Toc498077102)

[Attempt #4 Result: 12](#_Toc498077103)

[5 JAVA (With Weka) Program 19](#_Toc498077104)

[Train Model: 20](#_Toc498077105)

[Predict New Movie: 21](#_Toc498077106)

[View Movie Data: 22](#_Toc498077107)

[6 Conclusion 24](#_Toc498077108)

[7 References 24](#_Toc498077109)

# 1 Introduction

This program attempts to capture what is perfect recipe to increase the chances for a movie to be successful. In our current industry we have all sorts of movies, some are destined to be a box office success, others, are meant otherwise. Hence this program’s main purpose is to look into thousands of data, to identify right combination of talents (actors and director) and the trends (genre). However, there are times that even if we put the right talents, during the prime time, however without the right genre, the movie will still be a flop. It is not always very clear cut on how these attributes actually interact with each other to form the successful movie recipe. This project attempts to identify the factors through the machine learning techniques of Decision Tree.

There are total of 6 attributes (Main Actor, Secondary Actor, Director, Genre, Country of Origin as well as the Budget) that we use to measures the success of a movie, excluding the secret factor number 7th, which is “luck” (which this project will not be able to cover). Many of the successful movie’s attributes are only obvious upon the launching of the movie (trends of the time the movie launched, current events and etc). The project’s measurement will rely heavily on the 6 differentiators which is within the control of the producers.

# 2 Business Use Cases

## 2.1 Challenge

To predict whether there will be success or failure for a potential movie production, in terms of financial gains, i.e., whether the production’s gross earnings is to be estimated to a bigger number than its budget by a good margin. However, not all the data used for prediction have the required attributes of interest, and those which do have the full tuple, need further pre-processing to group/segregate the attributes of the movie datasets, so that the predictive model developed by the team can have more effective/meaningful prediction.

## 2.2 Solution

Based off J48 decision tree algorithm from Weka machine learning library, we built a predictive model to forecast whether the new movie will be a production of success or failure. Existing movie data are collected from sources open/available to public. Some tools and measures, which shall be discussed in detail in subsequent sections, were utilized to pre-process the data and attributes of interest.

# 3 Machine Learning Techniques

Since the business use case we are trying to solve is to predict the success or failure of a potential movie production, it is best suit to use classifier machine learning algorithm. Among many classier algorithms such as Bayesian, SVM, Decision tree, Logistic regression and etc., we chose Decision tree considering below factors:

* fairly transparent and easy to understand
* training data has label (success or failure)
* able to achieve the desired level of accuracy as per model testing in Weka Explorer and hence it serves the requirements.

## 3.1 Decision Tree

We decided to use J48 decision tree algorithm from Weka machine learning library after accessing it via Weka Explorer. Based on training data, the decision tree algorithm will build a predictive model which is mapped to tree structure and it will be used to predict/classify the new data (in this case, a new movie).

# 4 Data

## 4.1 Data Collection

To get the project started, our very first step is to gather all the past movie data. We were fortunate that movie industry is very well established. There is countless amount of data, however with these data there is a downside of it, which is over complicated. If we were to take all the data into consideration, it will definitely overfit the model. Our main source of data came from IMDB, Facebook, Rotten Tomatoes, Wikipedia and Kaggle. Most of the main reference site contain existing API that we can leverage to extract the data.

Here is some breakdown of the type of data that we extracted. From IMDB [1], we get the data for the movie title, main actor name, secondary actor name, director name and financial information. From Rotten Tomato [2], we get the genre information (by matching the name of the title). For Facebook [3], we get the actor/director’s Facebook likes (we take this as the popularity). For Kaggle [4], we extract some of the extra data. All the data are then ported into .csv file, where we will use Microsoft Excel for our data cleansing and filtering.

With the data collection exercise, we managed to harvest 5044 unique movie titles (5044 rows of movies) from all the resources above. However, not all the movies contain all the important information that we needed to study and determine whether the movie is a success or not. On top of that, some of the movie’s title, actor names and directors name contain some non-standardized form of characters. We will need to proceed to the next steps which is to normalized the data before we can feed it into our learning algorithm.

## 4.2 Data Cleansing and Improvement (Challenges)

The data that was freshly extracted will need to be normalized, so that all the data can be referenced with each other. To start off the normalization, we first lowercased all the character, and removed all the non-alphanumeric characters of the movie titles.

By using Microsoft Excel, we managed to parse and cleaned up all the data that we have gathered. Through the excel script, we are able to merge the row based on the key identifier (Movie’s name).

We use the Movie’s title as the key identifier as this information is found across various source. Through the Movie’s sheet, we are able to extract the main actor, secondary actor and director’s name. At this point, our row consists of *[Movie Title, Main Actor Name, Secondary Actor Name and Director Name]*

Subsequently we managed to get the popularity of the main actor, secondary actor and directors from the Facebook likes (the assumption here we made is, that Facebook likes will contribute directly to one’s popularity). At this point, our row consists of *[Movie Title, Main Actor Name, Main Actor Facebook Page Likes, Secondary Actor Name, Secondary Actor Facebook Page Likes, Director Name Facebook Page Likes].*

For the financial data points (Budget and Gross] of a particular movie, we extract it from IMDB sheets. [Note: As our Machine Learning’s main criteria for a movie to be successful, the Gross Profit must be at least 20% more than the Budget. Hence, we will only take in movie data that contains the Budget and Gross earnings.] With the financial information, we will create additional column, for Gross that is 20% more than Budget, the particular movie will consider as ‘Success’, otherwise it’s a ‘Fail’. To further get the movie’s characteristic, we extract the movie’s top 3 Genres and the movie’s Country of Origin. At this point, our row consists of *[Movie Title, Main Actor Name, Main Actor Facebook Page Likes, Secondary Actor Name, Secondary Actor Facebook Page Likes, Director Name Facebook Page Likes, Budget, Gross Profit, Genres, Country of Origin and Results].*

With the row of data formed, next we look into filtering off the rows that do not have complete tuple of information (in particular, all the key differentiator that we intend to use as the filtering/learning criteria). Upon the data cleansing, we are left with 1,559 rows of data.

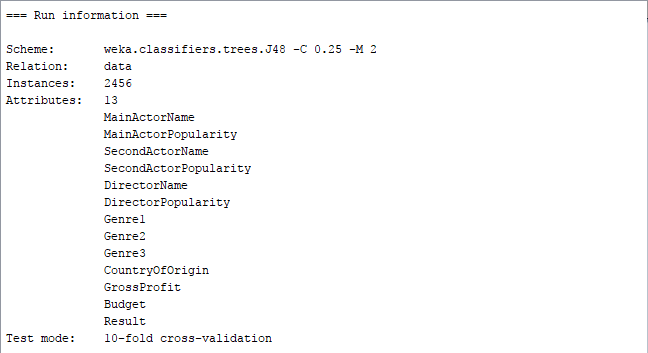
## 4.3 Tool used to validate the datasets

To try out (validate) our datasets, we use the Weka Explorer (v3.8.1). By feeding the dataset into the Weka Explorer, and selecting the algorithm Decision Tree-J48, we will be able to see what’s the results. With this tool, we are able to quickly validate our datasets before we perform actual implementation (using Java).

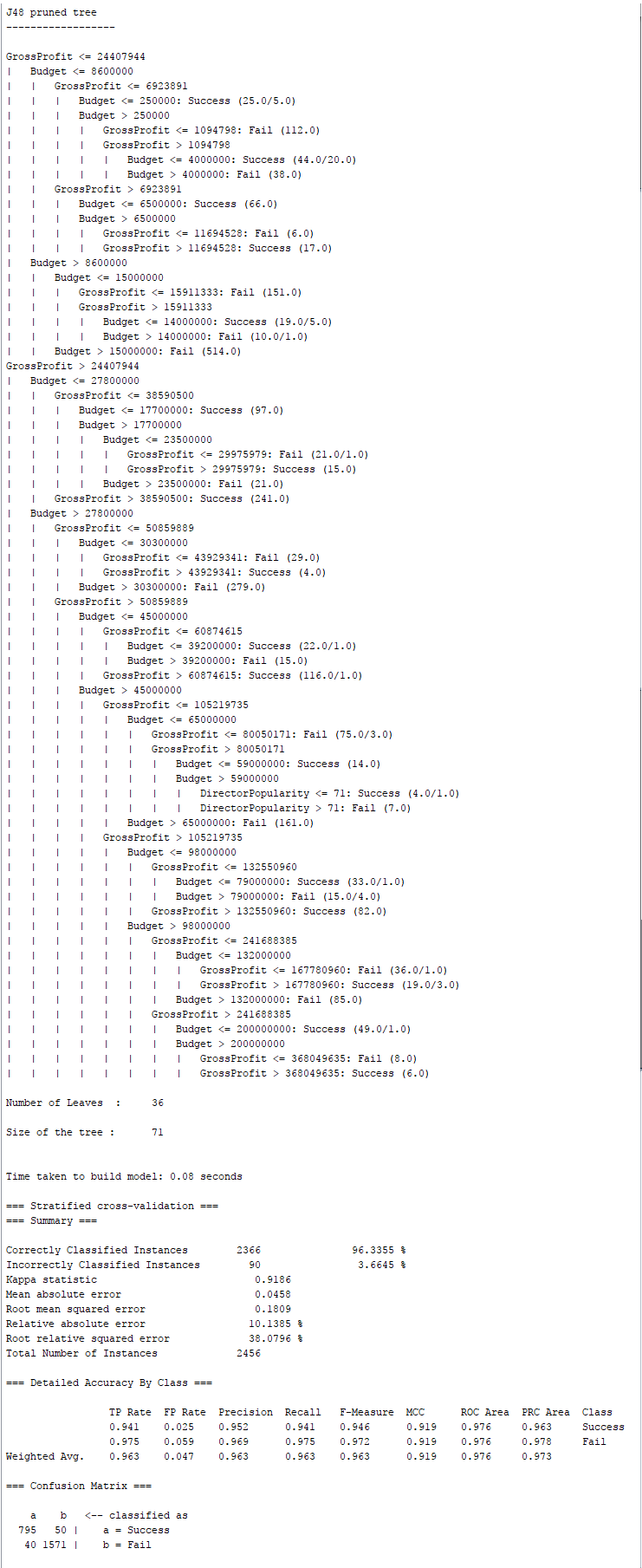
In our first attempt, we tried experimenting with few more attributes, such as Plots, additional Actors, Content Rating, Movies Popularity and Durations, but we found that these are overfitting the algorithm and it’s giving a very low accuracy as we are facing with overfitting. The result is totally off the chart. Hence, we decided to reduce the number of features in our datasets.

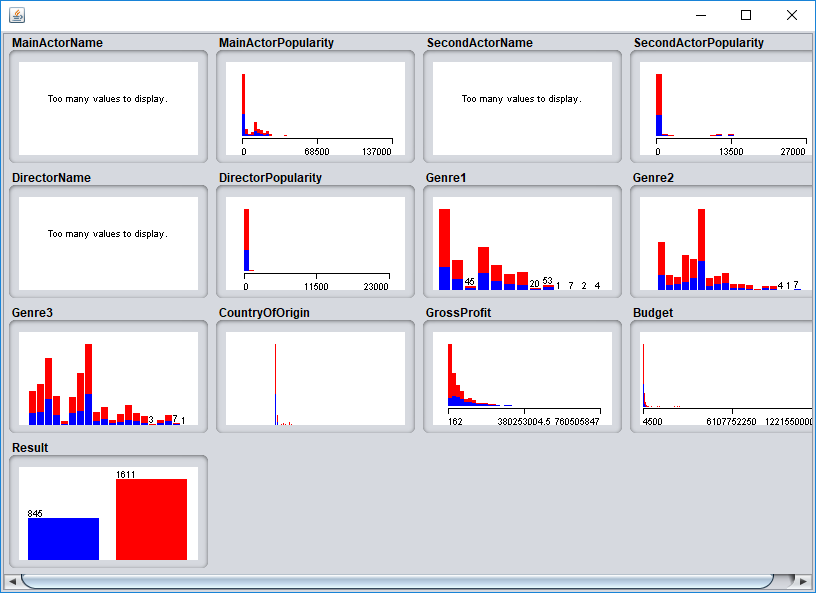
Our second attempt of the datasets [*Movie Title, Main Actor Name, Main Actor Facebook Page Likes, Secondary Actor Name, Secondary Actor Facebook Page Likes, Director Name Facebook Page Likes, Budget, Gross Profit, Genres, Country of Origin and Results]*, we are seeing that we progress towards the right direction, the accuracy hits **96.3762**%, then we look back at the decision tree we realized that the machine is putting high emphasis towards the Budget and Gross Profit to determine the success criteria, hence the high 96.3762% of accuracy. We realized that we are not supposed to put Gross Profit as one of the criteria for the decision tree, as that feature will not be available before the movie launched. Hence our next set of data, we removed the Gross Profit feature.

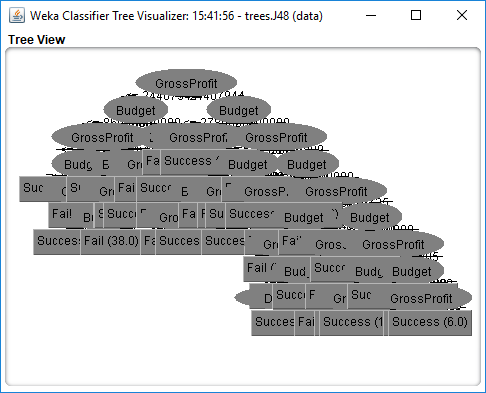
### Attempt #2 Result:



For attempt #2, we have 13 attributes that we use for the learning, and also for this model, we uses the J48 decision tree algorithm (with 10 folds Cross-validation).



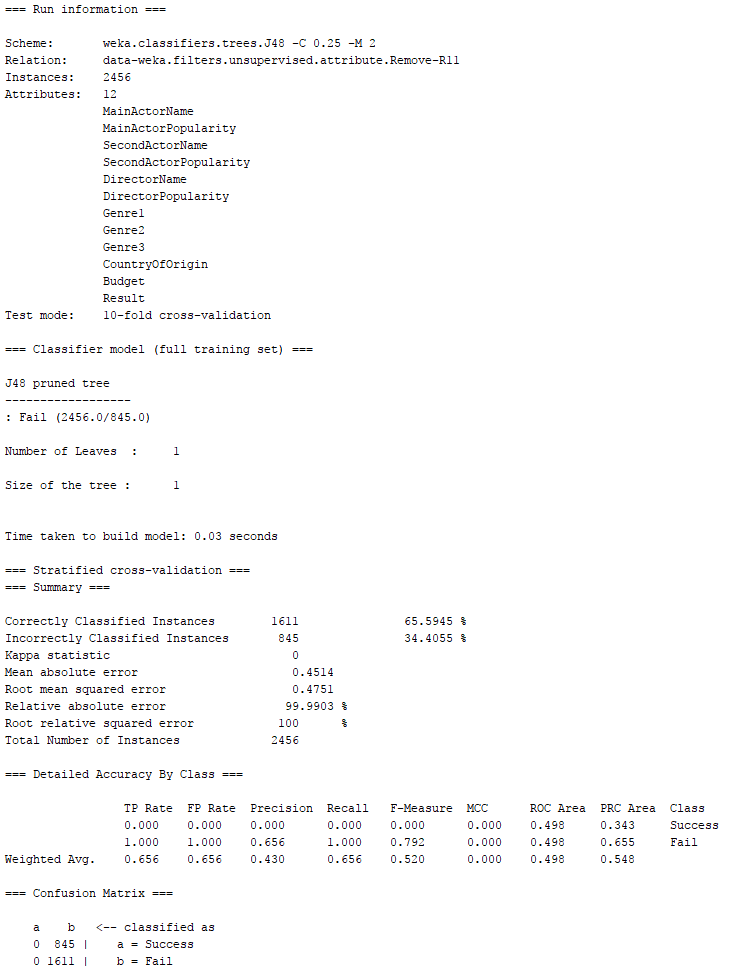




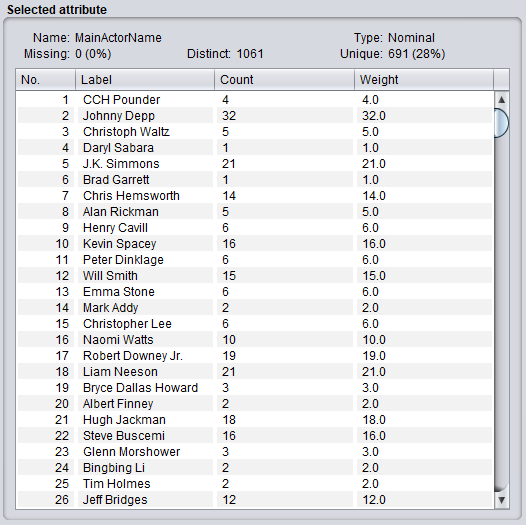
As we can see that the GrossProfit and Budget are too closely related to determine the ‘Success’ or ‘Fail’, in actual case when we were to run the Estimation, we will not have the ‘GrossProfit’ information and it will not be accurate. This leads to our third attempt.

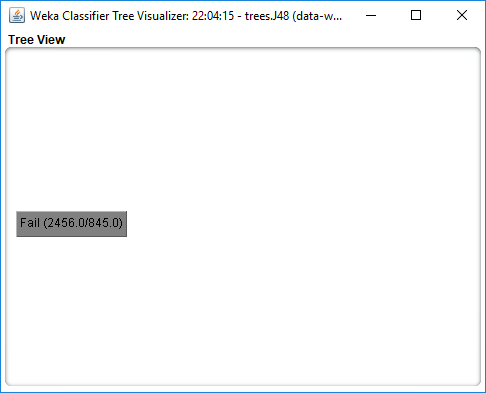
In our third attempt, we remove the ‘Gross Profit’ feature, and this time, we are seeing that the accuracy rate dropped tremendously **65.60%**. When we look into the decision tree, we realized that the success rate was too low due to the fact that we have too many unique entries for the Names (Main Actor, Secondary Actor as well as the Director). Due to some of the names are only appearing once or twice throughout the 3k datasets, the accuracy was vastly impacted by these. Hence our next step is to look into improving these features. We are not able to dropped these features, due to the fact that they are consider as the main criteria.

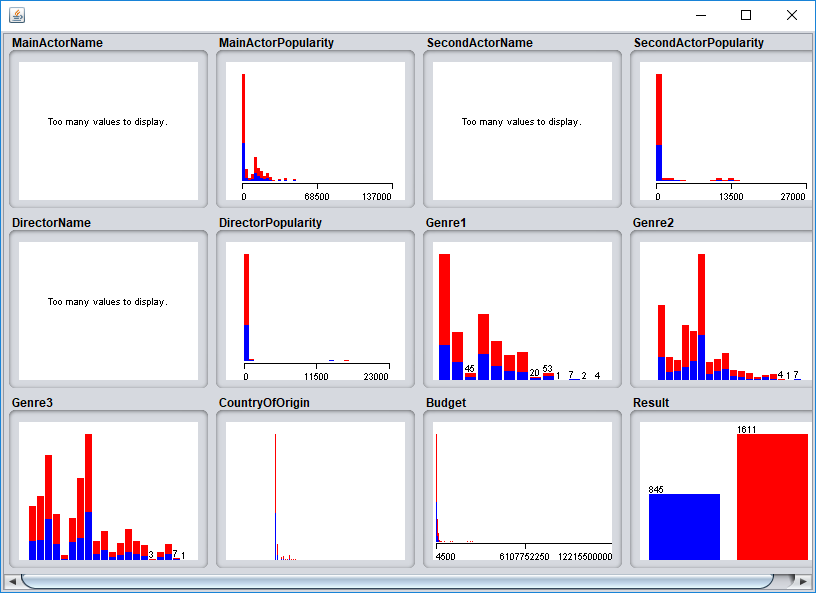
### Attempt #3 Result:



From this exploration we can see that the machine is losing a strong pattern. The current features are spreading across to thin on the percentage due to too many variances of Main Actor’s Name, Secondary Actor’s Name and Director’s Name.



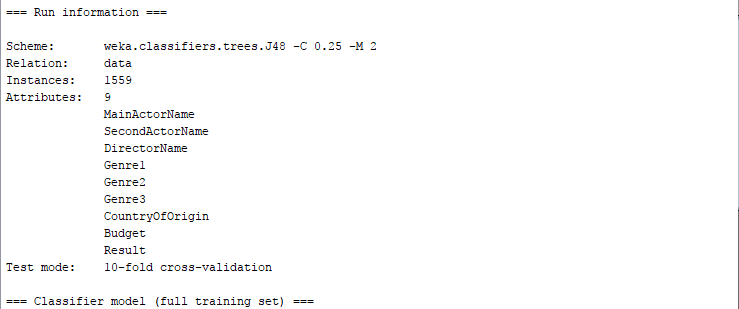


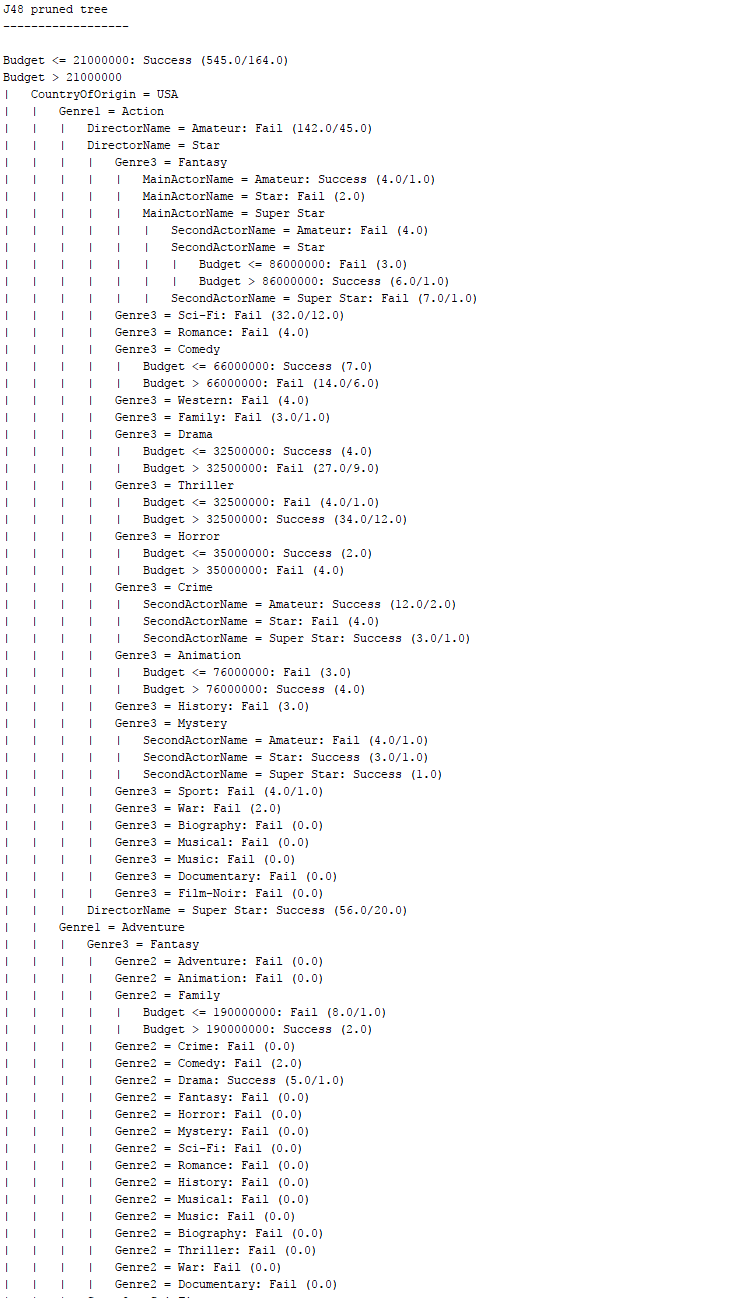


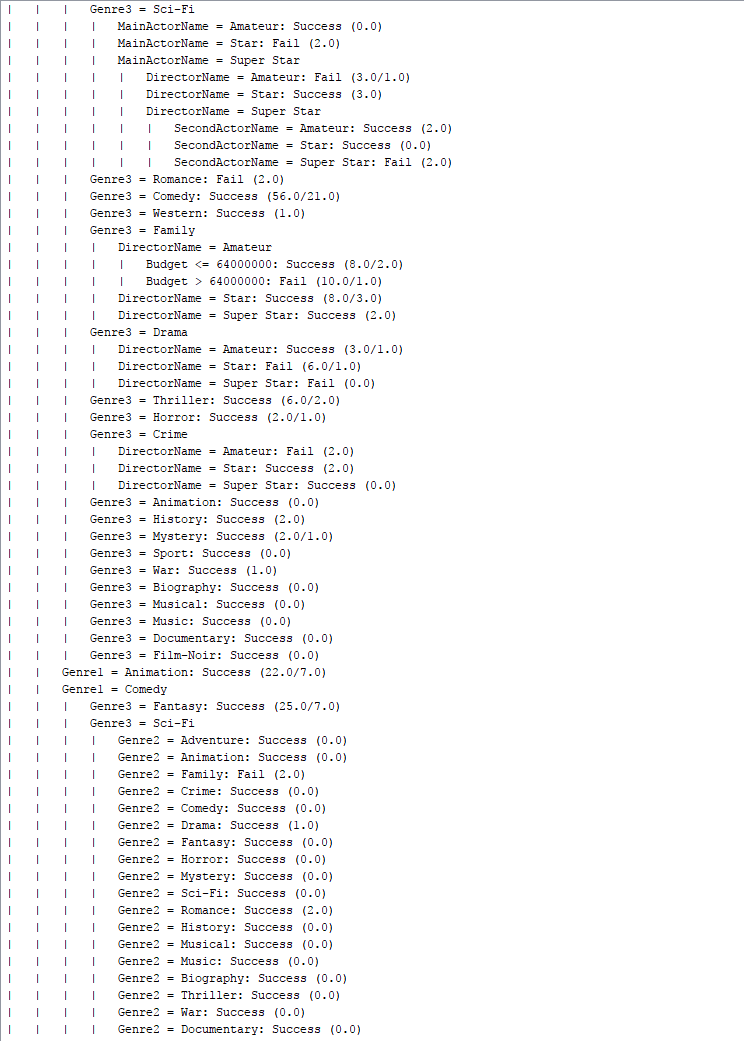
For our fourth attempt, we decided to group the Main Actor, Secondary Actor and the Director into 3 categories. For the Main and Secondary Actor that has <x> Facebook Likes we will classify them as ‘Super Star’, if they have <y> number of Facebook Likes will classify them as ‘Star’ if their Facebook like were to be lower, he or she will be classify as ‘Amateur’. Same goes to the Director, if their Facebook Likes were to be above 500 Facebook Likes, they will be considered as ‘Super Star’, whilst for Facebook Likes above 200 will be classify as ‘Star’ and other than that will be classify as ‘Amateur’.

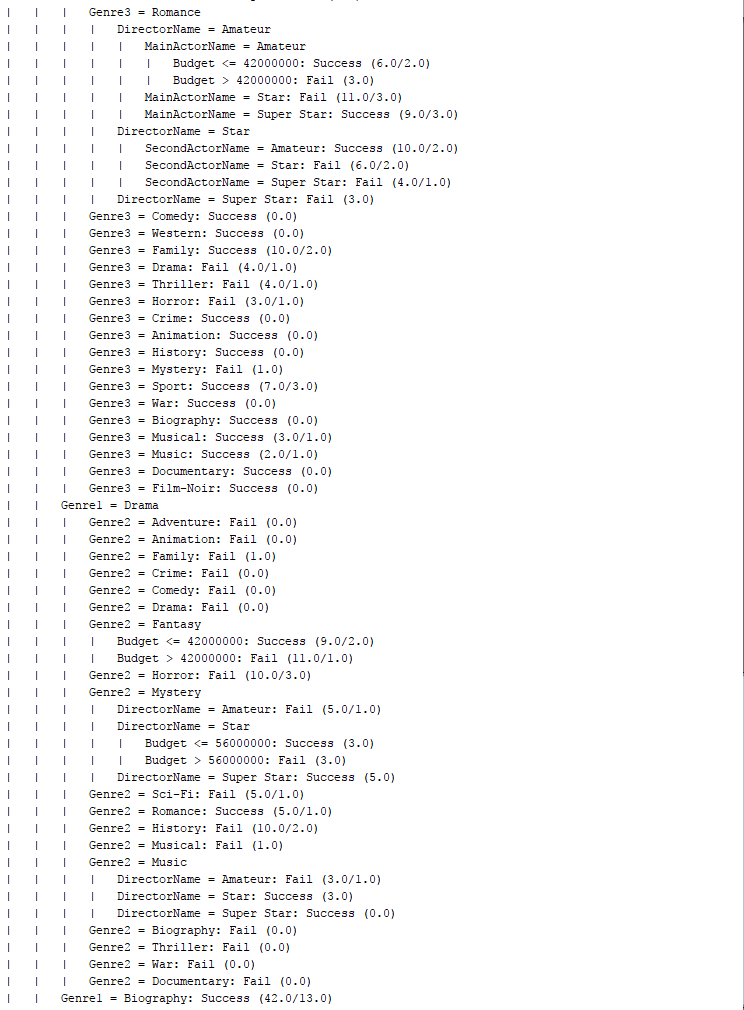
With this classification, when we train the model, we can see that the accuracy rate is still low at 62.16%. However, when we look at the size of the tree and the number of leaves, we believe that we are heading towards the right direction.

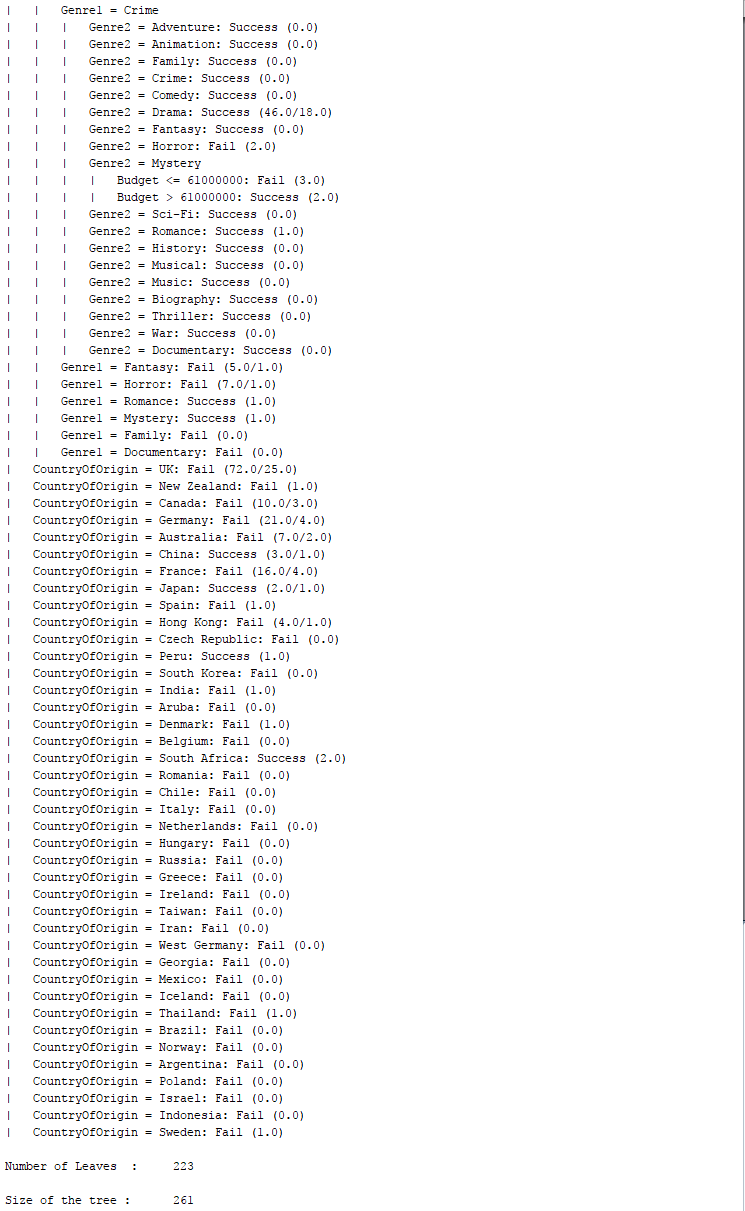
### Attempt #4 Result:

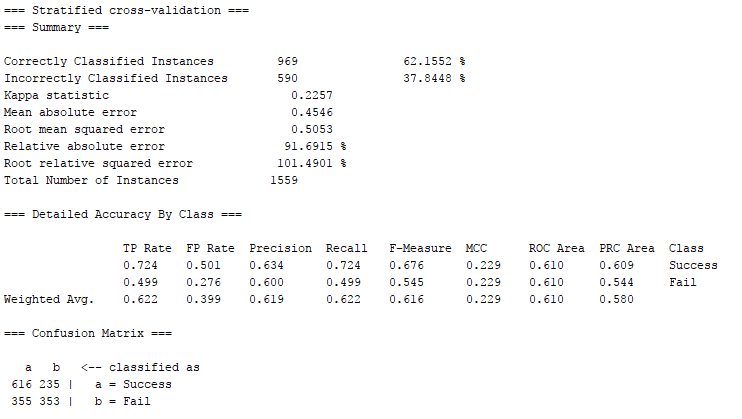


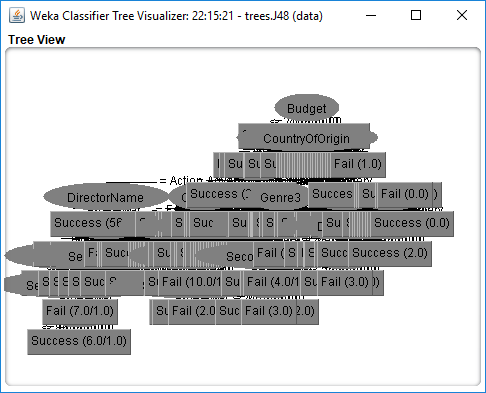


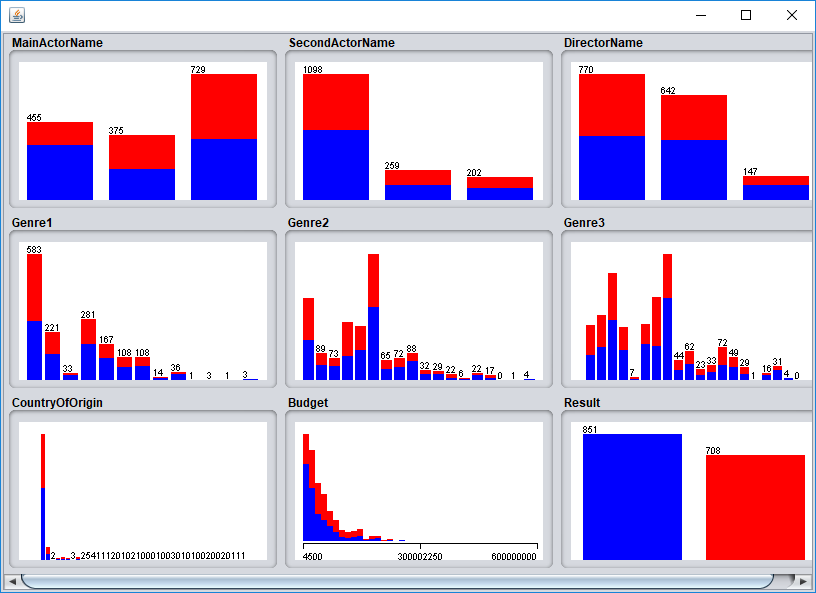












From the visualization, we can see a more reasonable model that we are heading towards.

Through the ‘Resampling’ of the data once, we can see that the percentage improved to 72.35%. With this accuracy, we decided to stick with it and proceed with our program development (using Java with Weka library).

# 5 JAVA (With Weka) Program



The main page, where we allow the user can select ‘Menu’ and being presented with few options such as:

‘Train Model’ – where it trains the model based on the latest dataset

‘Predict New Movie’ – where it allows the user to input the features and the output of the program will tell if the movie will likely be a ‘success’ or ‘failure’

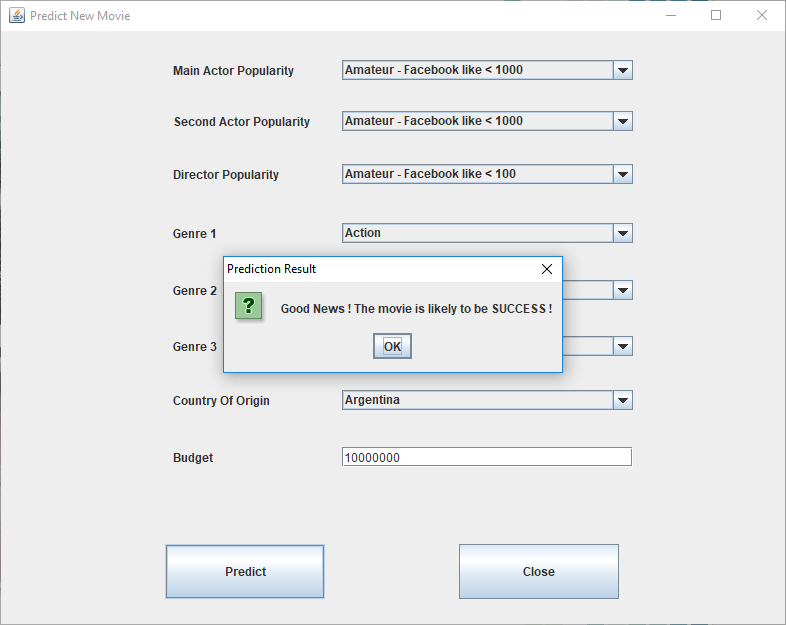
‘View Movie Data’ – this allow the user to view the current moviedata.arff, and it also give the user the capability to ‘add’ and ‘remove’ movie data from the moviedata.arff. [*Note: Upon adding/removing of movie data, we must reinitiate the ‘Train Mode’ for the machine to train a new model’*]

## Train Model:



When we select train model, the system will automatically load the moviedatasets.arff, and train it. This will generate a model which the ‘Predict New Movie’ will use.

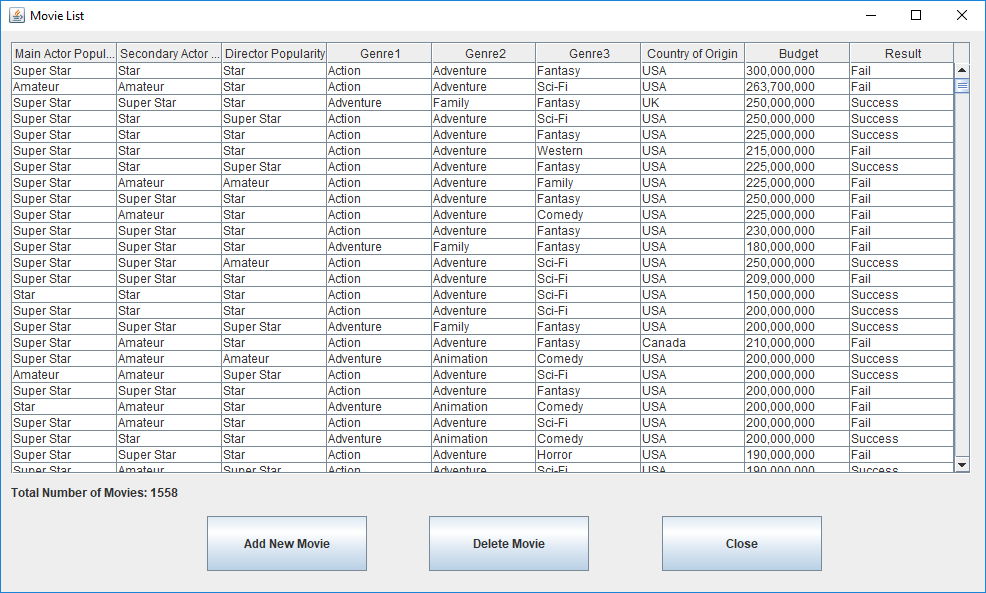
## Predict New Movie:



In the predict new movie option, this allow the user to select the features from the drop down and also input the Budget.

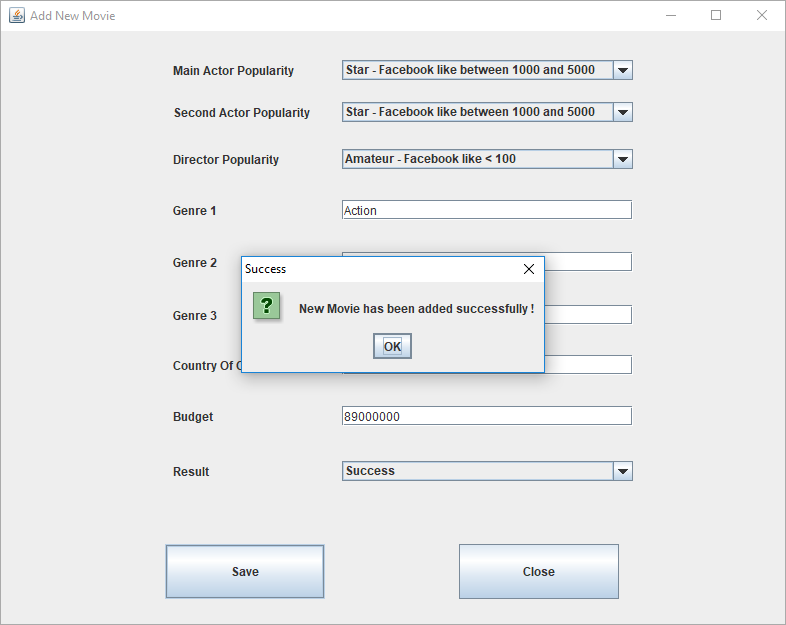
By clicking on ‘Predict’ the system will perform the checking and gives the info such as, will this movie be a success or not.

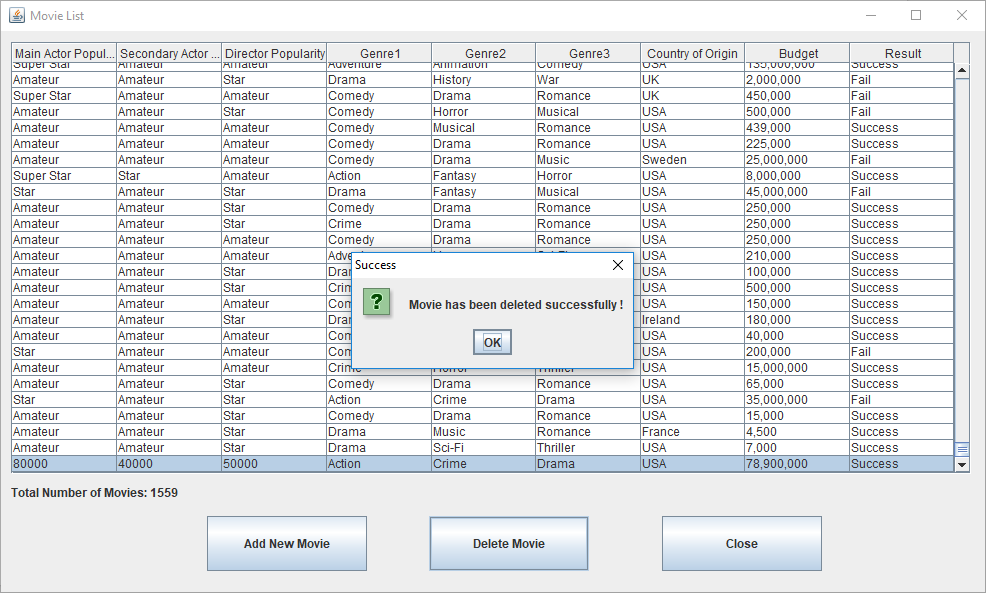
## View Movie Data:



This will load the Movie Data, allowing the user to go through the list of data that is used to train the model. The user has the option to ‘Add New Movie’ or ‘Delete Movie’.

With these feature, we allow the user to continuously increase their dataset coverage.





# 6 Conclusion

Through the use of the J48 decision tree, our classification accuracy rate reaches 72.35% (with the condition of re-sampling the data once). Though the accuracy may not be high enough (ready to be used in financial analysis), our learning results provide us some insight of what makes the movie successful. For instance, even though we have an Amateur Main character, but with a Super Star Secondary Actor and Directors pair them with the right Genres, the movie can still be a success. That goes vice versa too, even though with the Super Star Main Actor, Secondary Actor and the Director, if we pair it with the wrong Genres, the movie will still be a flop.

The above model study is also inclusive of few assumptions, such as, each movie must have at least 3 main Genres. This is to simplify the learning algorithm and allow the model to have equal number of features. To simplified to learning further, our datasets are not taking into consideration of the year attribute of the movie.

# 7 References

[1] IMDB: http://www.imdb.com/

[2] RottenTomatoe: https://www.rottentomatoes.com/

[3] Facebook: https://www.facebook.com/

[4] Kaggle: https://www.kaggle.com/

[5] Coursera Course