Decision Support System to Predict Movie Success Rate
– Data Mining Approach -

B.C.L Dias

169310D

Faculty of Information Technology

University of Moratuwa

February 2019

Decision Support System to Predict Movie Success Rate – Data Mining Approach -

B.C.L Dias

169310D

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfilment of Degree of Master of Science in Information Technology.

February 2019

Declaration

We declare that is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others, has been acknowledged in the text and a list of references is provided.

Student Details:

Name of Student	Signature of Student
B.C.L Dias	………………………
	Date: 17/02/2019

Supervisor Details:

Name of Supervisor	Signature of Supervisor
Mr S.C. Premaratne	……………………….
	Date: ………………...

Acknowledgements

First of all, I would like to make known my heartfelt appreciation towards my supervisor, Mr Saminda Premaratne, who is a Senior Lecturer at the Faculty of Information Technology, University of Moratuwa for his precious time spent towards the success of this research project, as well as his counsel, guidance and supervision.

Furthermore, I say a big thanks to my M.Sc. Degree Programme in IT Batch Mates for their precious contributions and responses towards the improvement of the results of this research. Many thanks goes to my family for the massive support that I have received from them and specifically for this research project. It is necessary for me to also recognize and appreciate the assistance on editing and encouragement from my sister, without which the completion of this thesis would not have been possible.

Abstract

Featured films are a multibillion-dollar industry. Online movie databases contain rich information about movies and people's preferences. An example is that people often rate and give comments about screened movies.

Selecting the Director, Leading Actor or Actresses is having a major impact on movie success. Other than that, there are many more attributes available which affects the movie success. Movie makers want to see each and every movie they produced to be a success overall. Therefore, to pursue higher success movies, makers and administrators should consider the best feasible selection. To do that, they have to identify major movie attributes in the first place.

In this study, we use data mining methods to identify patterns for predicting the success rate of movies using data collected from online databases. We use historical movie databases (TMDB/OMDB), to derive decision factors to predict the movies success rate.

The models we are about to identify with this research, using bottom-up approach are can be used to de-risk the entire movie industry.

Contents

List of Figures

Abbreviations

TMDB	-	The Movie Database
OMDB	-	The Open Movie Database
AWS	-	Amazon Web Services
HTTP	-	Hyper Text Transfer Protocol
API	-	Application Programming Interface
GUI	-	Graphical User Interface

Chapter 1

Introduction

1.1 Prolegomena

With rapid digitization, the film industry is growing rapidly. The film industry is a capital-intensive industry and one of the fastest developing industries in the world. The average number of movies produced per year is greater than 1,000. From that, the Chinese and Indian box office owns a considerable potion. Even though, very few movies are successful and are ranked high, giving the success rate, models and mechanisms to predict unfailingly the ranking and/or box office pools of a movie can help to de-risk the business ominously and to increase average returns.

Movie box office forecasting is mostly important to investment and financial aspects of the film. Movie budgets spend on cinema advertising, marketing, operations and various aspects. Furthermore, these predictions can be used to make more well-versed decisions. Predictive analytics combines a variety of statistical techniques from modelling, machine learning and data mining that investigate current and chronological facts to make predictions about the future.

Currently the available forecasting models are based on classical factors (producer, cast, stunts, special effects, director etc.) or on the social factors in form of considerations of the audience and social critics on various online platforms.

Even though this approach is accurate up to some level, it is not giving a success rate when the considered time frame is low. As such a better method is required, by using a larger data set to predict movie success.

1.2 Problem Statement

Despite the popularity of movies, prediction of the success of a movie has not been addressed in great detail as in other industries.

Since we have movie screening details, movie credits (Filmographies), movie ratings, movie popularities, and movie budgets/revenue and so on; we decided to introduce a method to predict success or flop of a movie by using model formation in Data mining. Ultimately it can be used to make some strategic decisions in the industry to lower the risk.

1.3 Aims and Objectives

By doing this study, our aim is to predict movie success rates using a bottom-up approach. The project's objectives are as follows:

1. Prepare Scripts/Programs to get movie data from the historical movie databases (TMDB/OMDB).
2. Data Pre-Processing (for null values, redundancies, inconsistent data).
3. Identify major movie attributes which cause a movie a flop or succeed.
4. Build/Identify models (Using Training Data set).
5. Test the built model (Using Test Data set).

1.4 Background and Motivation

Nowadays, with the high number of movie releases, the movie industry has become an extensively competitive industry. Competition and the bad selection of crew make the movie a flop. Because of the capital intensiveness, producers risk the movie by hiring low profile actors, directors etc. and some producers make the wrong decision and use a larger portion of the movie budget to hire one or two.

Even though movie makers select the best crew, there are some other factors which affect the screening movies e.g.: movie trailers, advertising mechanisms, main actor/actress's social life and behaviour, etc.

Still, there is no way of selecting perfect combinations of the crew for a movie. By identifying proper models/combinations using freely available movie related data, we can de-risk the entire industry.

1.5 Problem in Brief

Having movie related data like movie credits (Filmographies), movie ratings, movie popularities, and movie budgets/revenue and so on, no proper involvement has made in movie industry on how this historical information/data can be used for predicting the success/flop of a movie.

1.6 Proposed Solution

We proposed to use historical movie data to predict a movie's success or flop. It may help to make some selection decisions to make a hit movie. We model the movie using the best subset of movie parameters, and then use multiple classification algorithms to select what the best is to predict the success of a movie.

1.7 Thesis Structure

The overall thesis is structured as trails. The First Chapter is for an introduction to the full project with the objectives, background, problem and solution and the Second Chapter critically reviews the literature in the data mining technology, in the movie industry with reference to classification techniques. The Third Chapter is for technologies adopted and data mining technology by showing it is relevant to the movie industry. The Fourth Chapter presents our approach with inputs, outputs, process and features. The Fifth Chapter is the analysis and design of the solution and the Sixth Chapter is for the implementation of the solution. The Seventh Chapter reports on the evaluation of the solution. Finally, Chapter Eight concludes the solution with a note on further work.

Chapter 2

Data Mining Techniques for the Movie Industry

2.1 Introduction

The uses of techniques in data mining to predict the success of a movie, is critically reviewed in this chapter. In this perspective, we first discuss the data mining techniques general usage. Subsequently, we identified unsolved issues and concerns inherent in the techniques for data mining in predicting movie success. Lastly, in this study, we define the research problem meant to be addressed. This chapter also identifies the possible data mining techniques meant to be utilized for extending solutions to the problem.

2.2 Data Mining Techniques

On the basis of diverse tasks, data mining involves several approaches and disciplines as in *figure 2.1*. It can be classified into main two categories: Descriptive and Predictive. Depending on the different methods explored; data mining can be generally divided into machine learning, statistics and neural network. Machine learning is further divided into inductive learning, case-based learning and genetic algorithm, among others. Statistics are further divided in a more detailed way to consist of regression analysis, clustering and discriminant analysis etc. Talking about the neural methods, it is composed of self-organizing neural networks and feed-forward neural networks. The major method used in the database is multidimensional data analysis and online analytical processing. The following figure indicates data mining techniques distribution. Looking at all the requirements, there is no method of data mining that can fulfil all of it. For a particular problem, the data characteristics itself will affect the choice of tools.

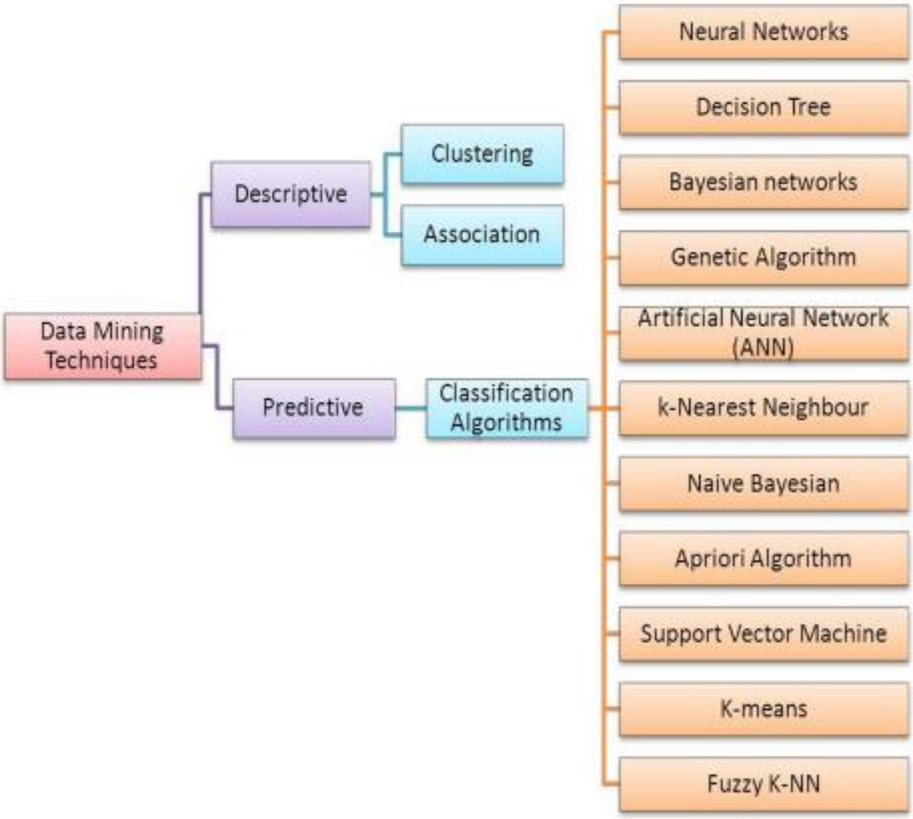


Figure 2.1- Data Mining Techniques

- **Decision Tree**

It is a classification method by modelling a tree-like structure which has representing class labels and branches representing features. This method is so called as "divide and conquer".

- **Logistic Regression**

Logistic regression is the appropriate regression analysis to conduct, when the dependent variable is binary. Like all other regression analyses, the logistic regression is predictive analysis. The outcome is measured with a dichotomous variable (involving the availability of only two possible outcomes).

- **Naïve Bayes**

The classifier called Naïve Bayes works on the basis of the Bayes theorem, with the assumptions of independence between predictors. The Naïve Bayesian model is easy to build, with no complicated iterative parameter estimation that turns it specifically into a useful system for very large datasets.

- **Weka**

Weka is a data mining tool, which helps in the integration of several machine learning tools within a common GUI. Its major data mining tasks are classification, association and summarization. Users can use Weka's API directly in Java programmes to perform tasks on machine learning. Its functions are data pre-processing, association, regression, clustering, classification and visualization. In this study, the Weka GUI is used as the tool for the process of identifying models in the data set.

2.3 Data Mining Process

Generally, data mining process can be categorized into the following phases, as in *figure 2.2* that follows, also depicts the entire process.

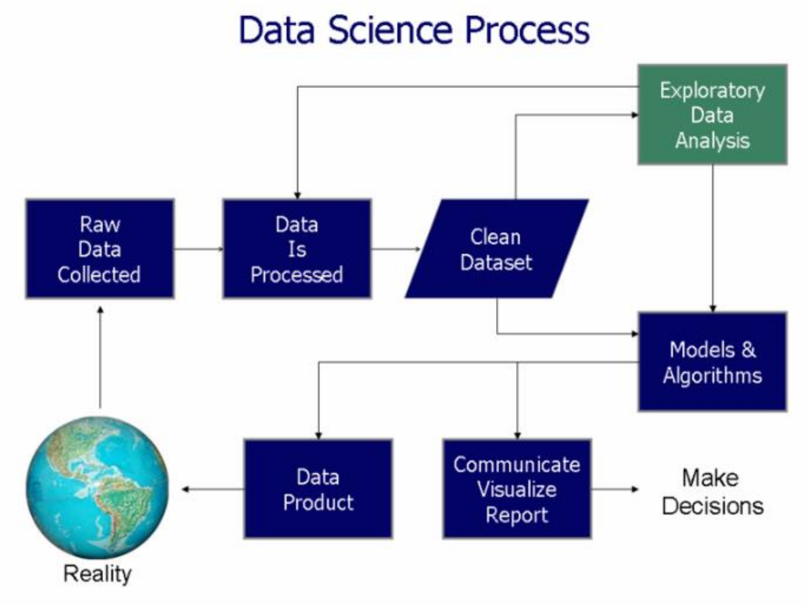


Figure 2.2- Data Science Process Flowchart

- Problem Definition

A right and adequate understanding of the business problem is what the project on data mining starts with. Here, the explanation of the understanding can be given in the objectives as well as the necessities from the perspective of the industry. The objective of the project is then converted into a definition of the data mining problem, which will give direction for the following work. There is no requirement of the data mining tools in the problem definition phase.

- Data Collection and Pre-Processing

Acquiring the data is what is meant by data collection. This can either be extremely simple or very complicated. Data acquisition can be done either manually or automatically. These processes are: data selection, data pre-processing and data conversion. In this study, we used scripts to automate the data collection.

- Modelling

There are several types of techniques available in data mining to resolve differentiated problems. Here in this study, selections of several modelling techniques are used a lot of times, in order to adjust factors to an ideal state, until the best values are achieved. By doing so, we ended up with a perfect model that any movie may fit.

2.4 Review of Previous Work

Linear Regression and Logistic Regression models are used by Jason van der Merwe et al [1]. In linear regression to identify the weight vectors, least mean square method, a specifically stochastic gradient descent was used. In their study, a movie title was given a score to include the movie title in the feature vector. By using K-Means clustering, accuracy was increased to 52%.

In order to identify the impact of a critical review of a movie, Alec Kennedy conducted a study. The author concludes, if a movie is released along with effective marketing strategies and favourably good critical reviews, there is a high chance of getting a success [2].

Jeffrey Ericson et al used only the attributes that are influential in the pre-release phase [3]. They also tried to analyse the impact of the movie title on its success.

Neural networking method is used by Sharda and Delen in their study, to process the movie's pre-release data. In their study, they took the popularity and quality variables of movies. With the use of the aforementioned attributes, authors classify movies into nine categories according to their anticipated income, from ‘‘flop’’ to ‘‘blockbuster’’ [4]. The neural network they built, was able to correctly classify 36.9% of the movies with the remaining 75.2% of the movies, being in one category and deviated slightly away from the correct category.

In the research conducted by Gloom et al [5] which took into account social media's movie feedbacks and responses to get a more accurate estimation in box office collection, he considered social factors as well. Part of the hypothesis of the project, is that the anticipation and social media feedback helps to predict a movie success.

To get an overview of the movie itself, The MooVis tool was used. It is a popular approach developed by Google to predict movie success. This approach uses Google's gigantic pool of search data to forecast box office performance using the volume of queries [6]. YouTube metrics can also be used for predicting the box office performance by influencing the viewer.

Eldar's Sadikov et al implemented a model for analysis using a comprehensive set of extracted features from blogs for prediction of movie sales [7]. The authors used the wide-ranging list of features that deal with movie references in blogs. For this study, the authors used blog data set from spin3r.com.

By using normalized graphs, Jaehoon Lee, Giseop Noh and Chong-Kwon Kim, [8] find the movie's moods. Authors used the site “naver-movie.com” and the Korean film council which contains the number of customers in each movie as data set. In their study, the authors were able to find relations of word-of-mouth effects to the movie's success.

Guijia He and Soowon Lee [9] demonstrate movie metadata can beat social data in some cases. In their paper, the authors utilize EM (Expectation Maximization) algorithm to divide movies into several groups, and then for each group they learn one model to predict movie box-office revenue separately. The study shows accuracy can be earned by using multiple models.

As we previously mentioned, multiple attempts have made by different researchers to identify what makes a movie a hit. As we can see, they tried to achieve it using unique different approaches. If there is a way to predict a movie success in early production stages or in the planning stage of a movie, it would help to de-risk the movie industry. Industries capital intensiveness is increasing day by day to cope with larger crowds, and their different interests.

2.5 Problem Definition

Available forecasting models are based on various factors of a movie. Even though the number of researches done in this field is high, there is no system or methodology to predict the success rate for a movie. Most of the systems use only the top 100 movies and the time span is limited to one or two years and all of them forecast only success or flop. Even though the effort made is high, forecasting a movies success is still a challenging task. As per the literature, all detailed information about movies are available on the internet but there is not one comprehensively analysed and able to predict movie success rate by considering longer durations. Everyone was evaluated by only taking a few years.

2.6 Summary

This chapter presented a comprehensive critical review of data mining techniques with a specific reference to web data mining. The next chapter will discuss the technologies adapted for solving our problem.

Chapter 3

Adopted Technologies

3.1 Introduction

In the previous chapter, we discussed different findings in the area of movie success predictions; its developments, issues as well as future challenges. We defined our research problem and this chapter highlights the effectiveness of selected technology that distinguishes it from the technologies applied in the existing literature.

3.2 What is Data Mining?

In recent times, we come across a large and complex set of data which is generated by computers, networks and humans. Data mining is identifying interesting patterns using these big data sets.

3.3 Technologies We Used

Node.js

Node.js is mainly influenced by Python's Twisted and Ruby's Event Machines, all of those are similar in design. As an asynchronous event-driven JavaScript runtime, Node is designed to build scalable network applications. Node takes the event model a bit further and it uses Google Chrome's V8 JavaScript engine. It is a cross-platform javascript runtime environment and is open source. With the help of Node.js it is possible to run JavaScript outside of web browsers.

Sequelize

Sequelize is a promise-based ORM for Node.js v4 and up. It supports the dialects PostgreSQL, MySQL, SQLite and MSSQL and features solid transaction support, relations and read replications. By nature, it supports synchronisation and validation. The demand for a product like Sequelize is gigantic.

Axios

It is a Promise based HTTP client for the browser and Node.js and is a Javascript library. With the aid of Axios, it is very convenient to perform HTTP requests. It supports all modern browsers and allows users to write Async/Await tasks. Axios comes with methods for all HTTP verbs but still these methods are less popular.

AWS

It is a subsidiary of Amazon to provide on-demand platforms using cloud technology for individuals. It allows users to create and purchase hardware solutions according to their preferences. Hardware which resides in the cloud is highly accessible and it is a very affordable solution which is charged by subscription basis. Server farms are implemented throughout the world as such provides high availability. Along with this solution, it provides high redundancy and security for data. It is the best option available to acquire high computational power and high storage for a limited time. In the year 2017, they offered more than 90 cloud-based solutions. The number of solutions they provide is growing rapidly year by year.

AWS offers reliable, scalable and inexpensive cloud computing services.

Web API

It is a service-based technology, usually limited to a web application's client-side (including any web frameworks being used), and thus usually does not include web server or browser implementation details such as SAPIs or APIs unless publicly accessible by a remote web application.

Weka

Named after a flightless New Zealand bird, Weka can be used for the process of analysing data. It contains tools for big data mining. It is also well-suited for developing new machine learning schemes. Users can use Weka’s API library directly in Java programmes to perform tasks on machine learning. Its main functions are data pre-processing, association, regression, clustering, classification and visualization.

It is open source software issued under the GNU General Public License. It provides a very user-friendly GUI interface for its users.

3.4 Summary

In this chapter, we discussed the technologies we adopted in order to achieve our goal. Technologies that we can use for a task like this, is not limited. There are many other technologies available out there that do the same. In the next chapter, we describe our approach to predict a movies success rate.

Chapter 4

A Novel Approach to Predict Success/Flop of a Movie

4.1 Introduction

The technology used in solving the research problem is presented in chapter three. This chapter describes the approach of addressing the problem. Here, we highlight hypothesis, input, output and the process.

4.2 Hypothesis

We hypothesise that the issue of an unavailable proper mechanism to predict movie success rates can be achieved by using classifier analysis. We are going to use various classification technologies such as Naïve Bayes, Decision Tree, AdaBoost and Bagging and then finally pick up the most accurate classifying techniques based on the data model.

4.3 Input

For conducting this research, we collected movie data from the year 2005 to 2010 which was produced in the USA and already released to theatres. To achieve this, we used the web API provided by TMDB [10] and OMDB [11]. By using multiple Node.js scripts, we collected all the movie related attributes such as IMDB id, year, movie name, IMDB rank, budget, revenue, genre, actors (Cast), directors, writers, camera operators, sound, production, costumes, crew, visual effects, art, editing and lighting. Most of these attributes are multivalued attributes as shown in *figure 4.3*. As an example, for any given movie, it may have hundreds of actors, multiple directors, multiple camera operators, and so on. These values are going through multiple stages of pre-processing prior to mining and used as inputs to Weka miner.

4.4 Output

As the main output of this process, we obtained a combination of major movie success influence attributes. These are the attributes that make a movie a hit or a flop.

We then got prediction accuracy in terms of the used train and test data as in *figure 4.1 and figure 4.2.*

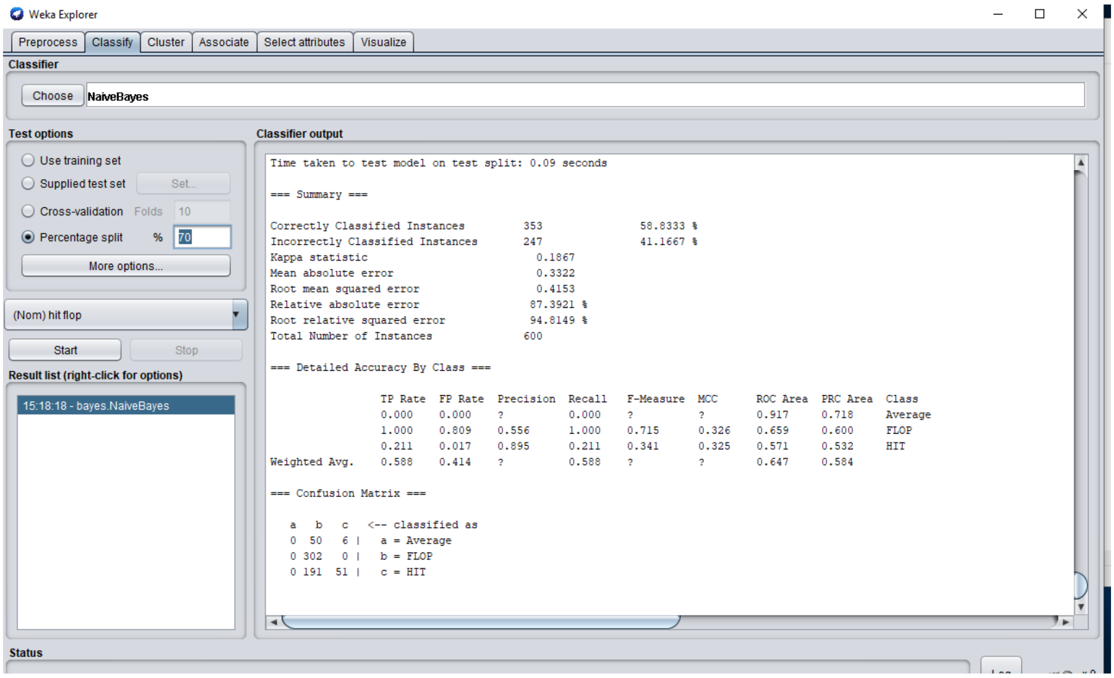


Figure 4.1- Sample Classification Output

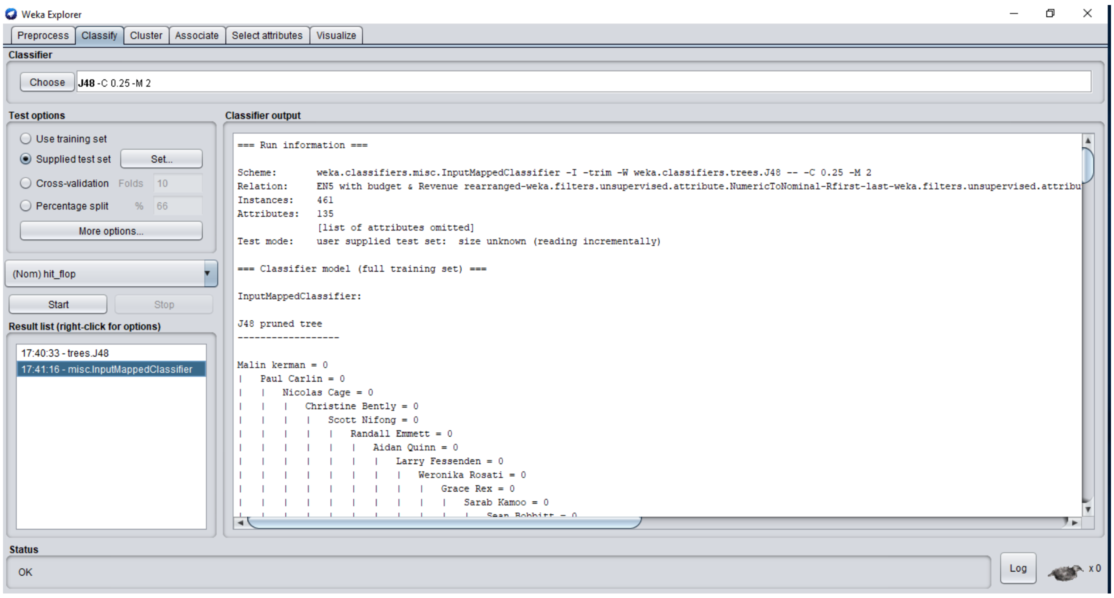


Figure 4.2- Sample Classification Test Set Output

4.5 Process

As explained in the input section, we are using two APIs to do web data scraping. To predict a movie success, we have to take it's budget and revenue details into account. We used OMDB API to collect any missing attribute values, web data scraping is a highly time-consuming task. In order to do that, we used AWS cloud computing services, by renting a server.

Once we acquired all data, we then created separate scripts to merge each of them and we ended up with a larger data set which consisted of thousands of records in the format as shown in *figure 4.3.* For the same data set, the .arff format is shown in *figure 4.4.*

IMDB_ID	Movie Name	Actor1	Actor_2	Actor_3	Direcor_1	Direcor_2	ETC***
T169265	A****	b***	v****	c****	NULL	NULL	*****
T169266	L****	c***	M****	NULL	K***	U***	*****

Figure 4.3- The Structure of Data we got Using Web API Form of a Table

```
@relation '5 Mine ready spcr'

@attribute movie_id numeric
@attribute imdb_id {tt2641186,tt1820402,tt1528081,tt3417334,tt2940482,tt1982882,
@attribute 'HIT FLOP' {Average,FLOP,HIT}
@attribute country {USA}
@attribute genres numeric
@attribute cast_1 {'Eliza Swenson','Craig Beeman','Alexander J. Bonds','Dean Cai
@attribute cast_2 {'Alexandra Turshen','Bob Diven',NULL,'Lawrence Hilton-Jacobs'
@attribute cast_3 {'Ramona Mallory','Joe Duerksen',NULL,'Tamara Goodwin','Ryan G
@attribute cast_4 {'Arron Shiver','Sean Cook','William Katt','Bumper Robinson','
@attribute cast_6 {'Alex Poncio','Lance Henriksen','Lisa Rotondi','Philip Tan','
@attribute cast_5 {'Charlotte Kirk','Jeffery Scott Lando','Luke Goss','Rance How
@attribute cast_7 {'Andrea Tice','Mark Mikita','Tyeisha Gibson','Brigitte Nielse
@attribute cast_8 {'Amanda Fuller','Jennifer Lauren DiBella','Goldie Loc','Willi
@attribute cast_9 {'Trent Haaga','Steven Michael Quezada','Masiela Lusha','Joe S
@attribute cast_10 {'Allyn Rachel','Thomas Jane','Diana Bang','Richard Edson','G
@attribute cast_11 {'Artem Mishin',,'Pim Bubear','Grant Harvey','John Savage','M
@attribute cast_12 {'Robert Patrick','Iulia Verdes','Kate Tomlinson','Paul Wight
@attribute cast_13 {'Joanne Kelly','Ashley Bell','Jeremy Piven','Gregory Jbara',
@attribute cast_14 {'Jodie Moore','Sharon Lawrence','Jacqueline Obradors','Ron H
@attribute cast_15 {'Eidan Hanzei','Patrick McDaniel','Nicole Badaan','Tamas Men
@attribute cast_16 {'Bill Oberst Jr.','Hisham Tawfiq','Tim Connolly','Amy Morris
@attribute cast_17 {'David Kaye','Gregg Berger','James Arnold Taylor','Tom Kenny
@attribute cast_18 {'Mike Winkler','Rami Jaber','Brian Villalobos','Gemma Donato
@attribute cast_19 {'Jen Ray','Thomas Downey','Brian Thomas Smith','Tommy Wiseau
@attribute cast_20 {'Kate Bringardner','Jessica Juarez','Gift Harris','Melissa O
@attribute cast_21 {'Bruce McGill','Jennifer Jason Leigh','Debra Harrison-Lowe',
@attribute cast_22 {'Christopher Meloni','Ronnie Gene Blevins','Steve Kahan','Ri
```

Figure 4.4- The Structure of Data we got using Web API Form of Arff File

We then filtered out movies according to the filters below, to narrow it down:

- Movie - produced in the USA
- Movie - original language English
- Movie – status: released
- Budget and Revenue is greater than $10,000USD

The data we collected using the APIs, are not clean. There was a considerable amount of noise included. This is very normal when collecting data through web APIs. We used a Node.js script to remove this noise.

Then we made statistical decisions and followed serialize steps to make the dataset valid. The next step was, we then created a script to reorganize a large volume of data in order to do mining. By using that script, we made each actor, director, writer etc, as an attribute to every movie in the dataset and ultimately ended up with a much larger dataset which had over twenty thousand+ number of attributes.

IMDB_ID	Movie Name	b***	c***	U***	v****	M****	ETC***
T169265	A****	1	1	0	1	0	*****
T169266	L****	0	1	1	0	1	*****

Figure 4.5- The Structure of Data once Rearranged

We then followed several data mining techniques to predict movie success rate by using the filtered data set.

4.6 Summary

This chapter presented the machine learning approach for conducting our thesis. We discussed learning techniques including hypothesis, input, output and process. We defined the research process and also identified the possible approach for addressing the research problem. The next chapter will present the design.

Chapter 5

Analysis and Design

5.1 Introduction

In the previous chapter we discussed the approach we followed to address the problem and there, we highlight hypothesis, input, output and the process. In this chapter, we describe the analysis and design of the model.

5.2 Design the Solution

Below is the serialised steps we followed to achieve our goal, as shown in *figure 5.1*.

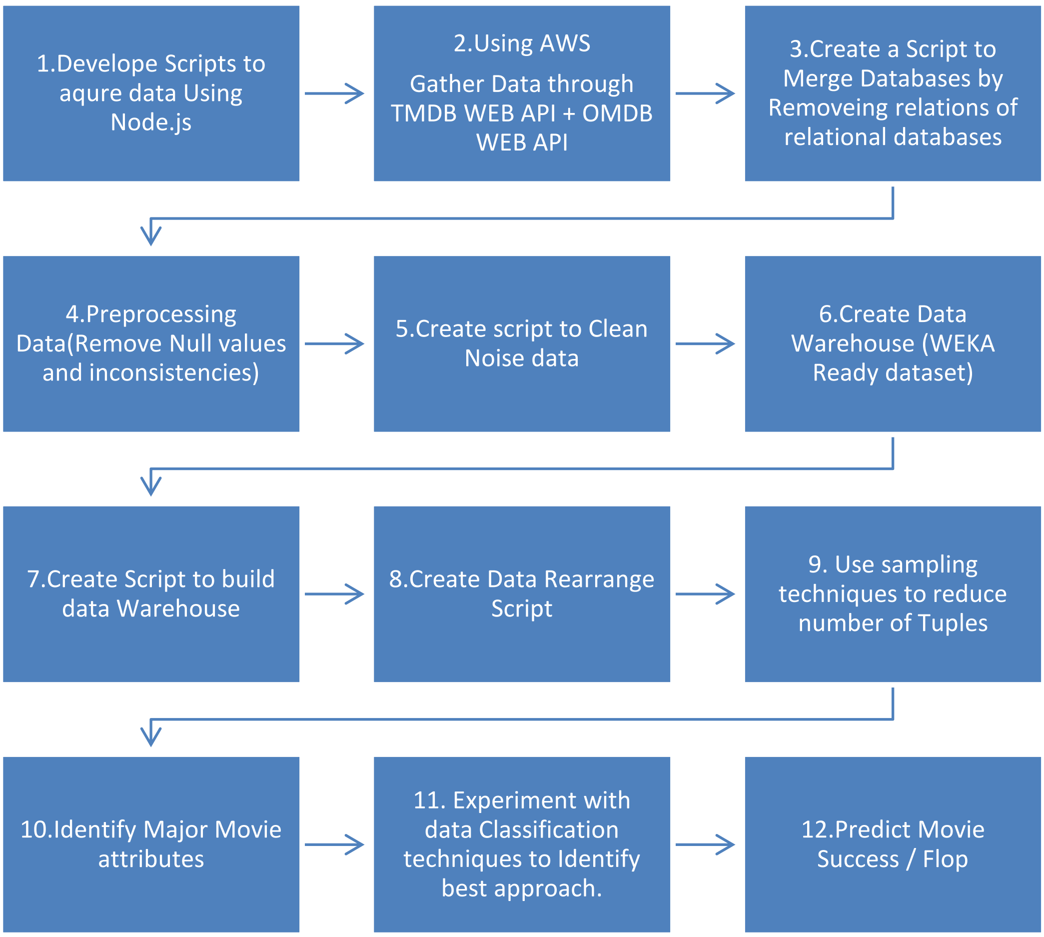


Figure 5.1: Top Level Floor Design

In the first phase, we developed two Asynchronous Scripts to get data from TMDB and OMDB APIs. To do that, we hosted our scripts in the AWS cloud server facility to collect data. *Figure 5.2* shows the top-level architecture of our project.

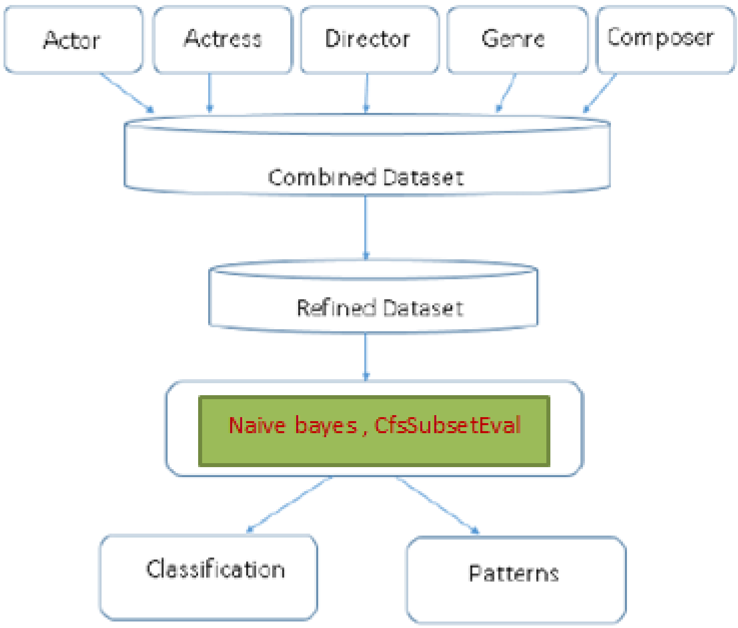


Figure 5.2: Top Level Architecture

For the next step, we create another Asynchronous Script, to merge the collected data and to store them in separate tables. We then pre-processed the dataset and replace all null values by checking its availability on other resources like IMDB [12]. In that stage, we identify data redundancies. Then we created another asynchronous script to rearrange the dataset by removing multi-valued attributes. We then removed the noise that was added at the time of data collection. To identify and remove noise, we developed a small script.

Using Weka's supervised "**CfsSubsetEval**" attribute selector, we reduced the dimensionality (feature selection) of the dataset. We then used several classifiers in order to identify the best fit for the collected dataset.

Once we got the correct classifier, we tried a few predictions by providing random data.

5.3 Summary

In this section, we discussed the top-level architecture of the design and the steps we followed to achieve a prediction of movie success in brief. In the next chapter, we explain the overall implementation.

Chapter 6

Implementation

6.1 Introduction

In the previous chapter, we discussed the top-level architecture of the design and approach. Here in this chapter, we discuss overall implementation.

6.2 Data Collection

For the analysis, data is collected from www.themoviedb.org *(Appendix A)* and from www.omdbapi.com *(Appendix B)*. For that, we are using Node.js scripts with axios and sequlizer libraries. We created web data scraping scripts to gather movie related data. The collected data was stored in another couple of tables since it has a lot of missing values and noise. Other than that, redundant data is a major problem when scraping data from the web. We identified redundancies using movie's ImdbId, and then removed it.

TMDb is a community-built movie and TV database. Every piece of data has been added by the online community dating back to 2008. Over 150,000 developers and companies are using the TMDB platform, hence TMDb has become a premier source for metadata for movies and TV shows.

TMDB provides a stable web API to extract data. We use AWS *(Appendix C)* to host Node.js script. This process consumes major computational power, and as such, we rented an AWS server to collect data.

6.3 TMDB - API Functionality

TMDB, one of the main web APIs we used to collect data, is shown in *figure 6.1*

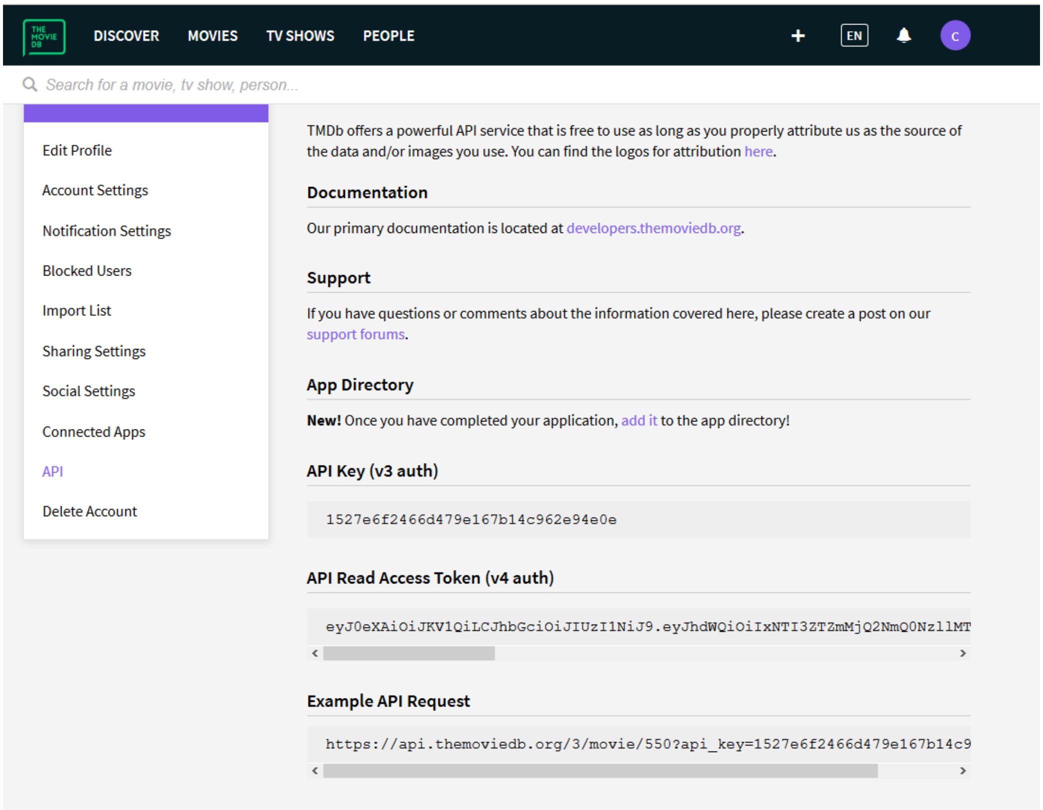


Figure 6.1: TMDB API

6.3.1 Get By Year

Firstly, we needed to get movies by using its released year. To achieve this, we used TMDB API's Discover method as in *figure 6.2*. For any given year TMDB has more than 5000 released movie details. Once the API call is made, the received dataset is shown in *figure 6.3*.

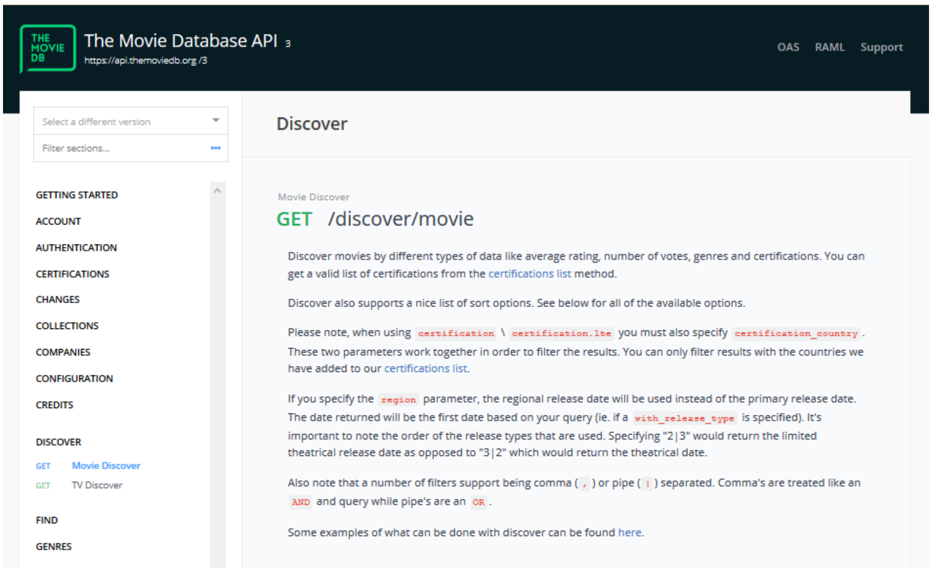


Figure 6.2- TMDB API- Get by Year

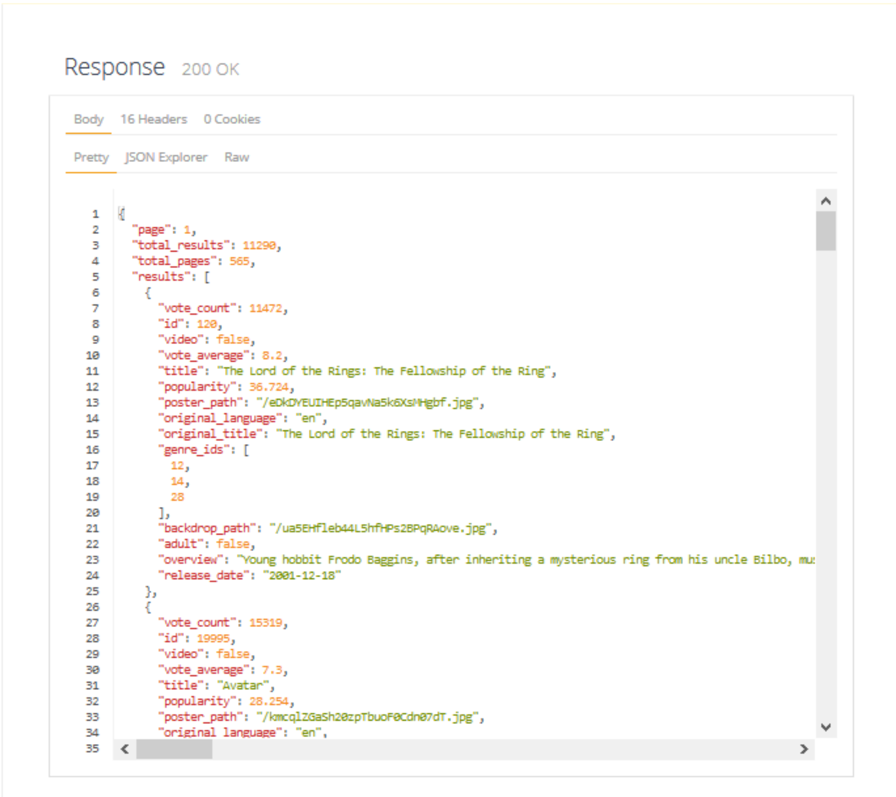


Figure 6.3- TMDB API-Results - Get by Year

6.3.2 Get Movie Details By ID

We then collected movie-related details such as budget, revenue and popularity using the API method below as in *figure 6.4*. Once the API call is made, the received dataset is shown in *figure 6.5*.

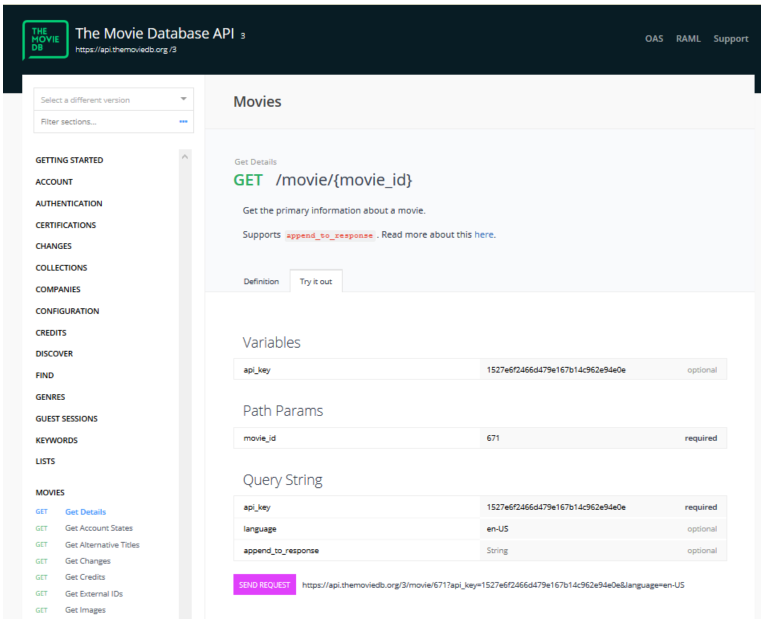


Figure 6.4- TMDB API- Get Details by ID

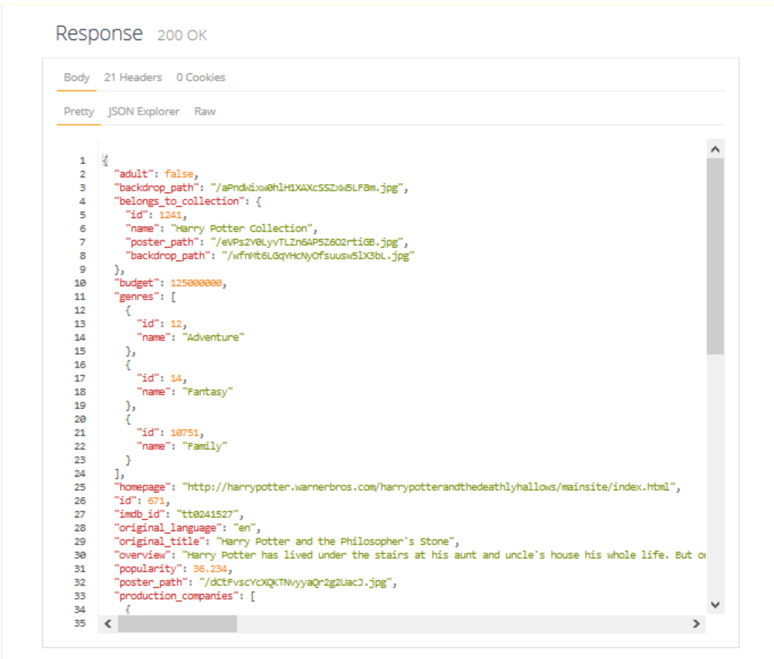


Figure 6.5- TMDB API-Results - Get Details by ID

6.3.3 Get Movie Credits By ID

We then used API's get movie credits by ID method as *in figure 6.6*, to collect details about actors, directors, writers etc. Once the API call is made, the received dataset is shown in *figure 6.7.*

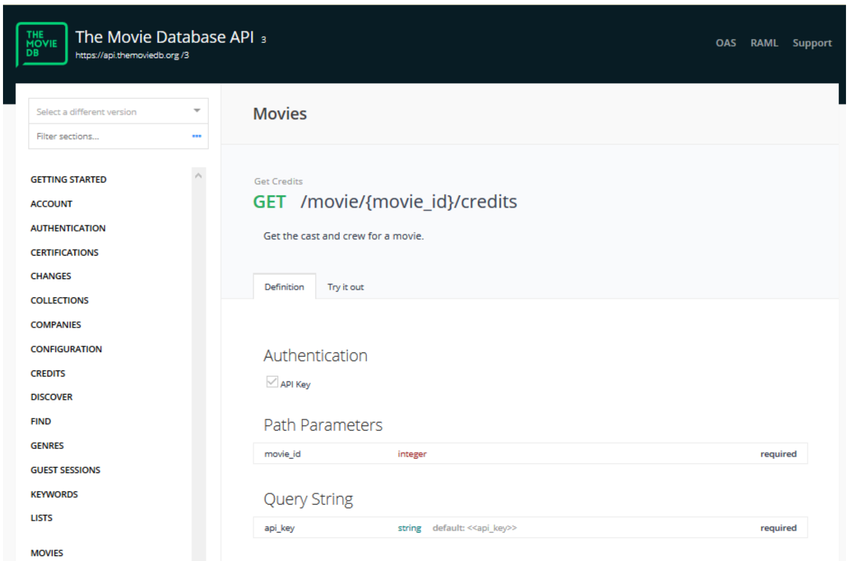


Figure 6.6- TMDB API- Get Credits by ID

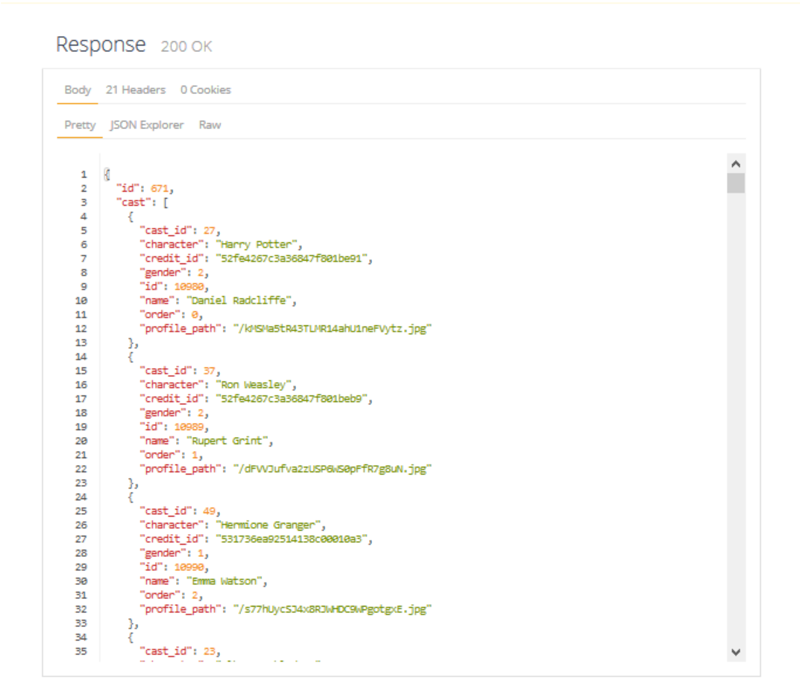


Figure 6.7- TMDB API-Results - Get Credits by ID

6.4 Scripts Developed to Web Data Scripting

We used Asynchronous Node.js scripts to collect data over time using AWS servers. Figure 6.8 shows the script we created to collect data from TMDB, and Figure 6.9 shows the script we created to collect data from OMDB.

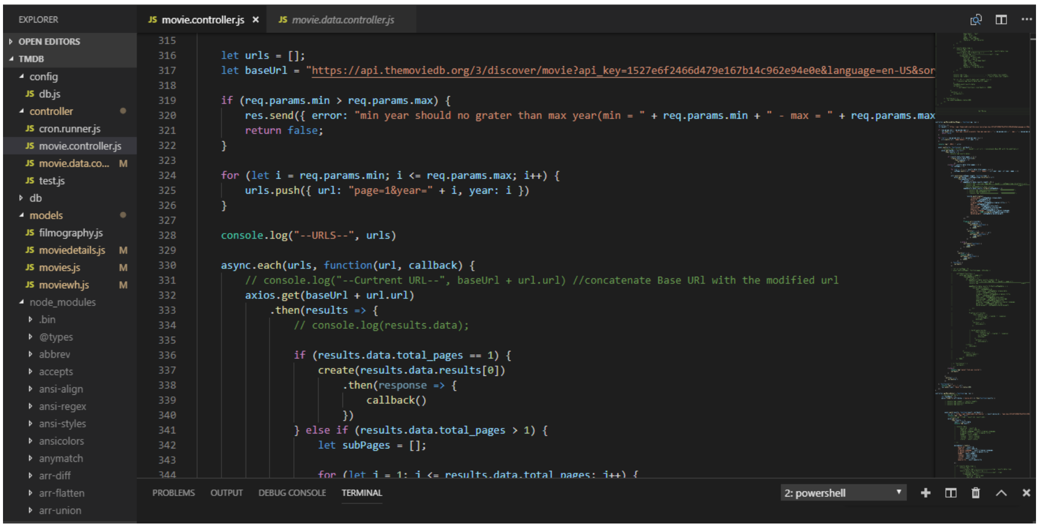


Figure 6.8- Web Data Scraping Script to use with www.themoviedb.org.

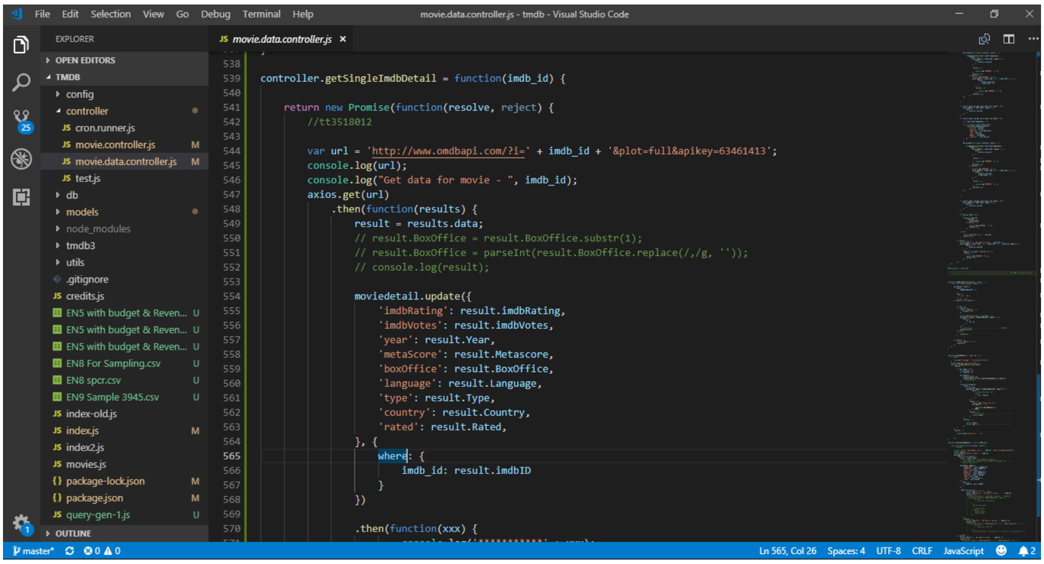


Figure 6.9- Web Data Scraping Script to use with www.omdbapi.com

6.5 Merging Data Tables

Then the database tables are merged using another script, by removing relationships. In selected movies, attribute count is diverse from one to another. As an example, 30 actors are involved in a movie but 45 may be involved in another, two producers may be involved in one movie but for another movie there are four producers. Hence, to merge tables without any data loss, we have to identify the maximum count for any movie. Separated scripts were created to serve the difficulty of merging tables. The script we created to merge, is shown in *figure 6.10.*

Figure 6.10- Script for Merge Datasets.

6.6 Data Pre-Processing

We filtered the required dataset by taking produced country = USA, original language = EN (English), movie status = released, to narrow it down further and we considered only the movies that had a budget and revenue greater than $10,000USD. Then we checked for null values, inconsistencies and redundancies and removed these. If the data is available in another API we used, we replaced nulls with that data. We also replaced missing IMDB rating values from the mean value.

When collecting data over web APIs, there is always a chance of adding noise to the data. We created another script to remove this noise as in figure 6.11.

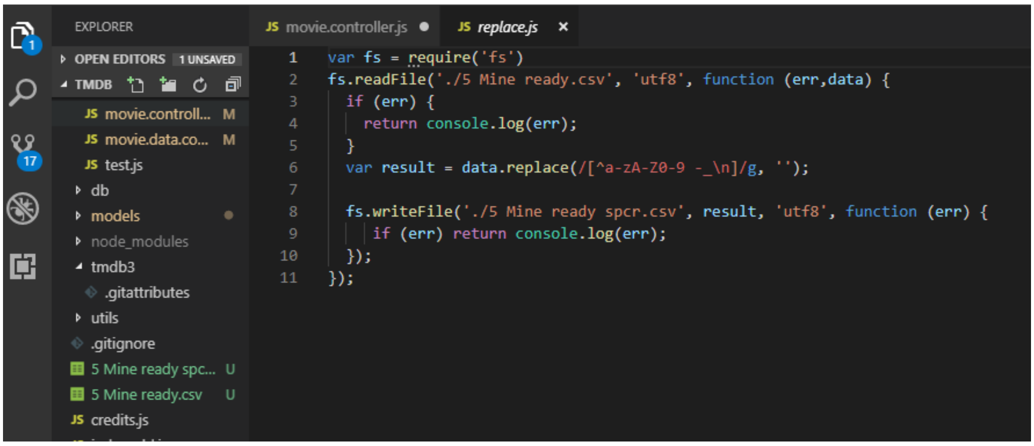
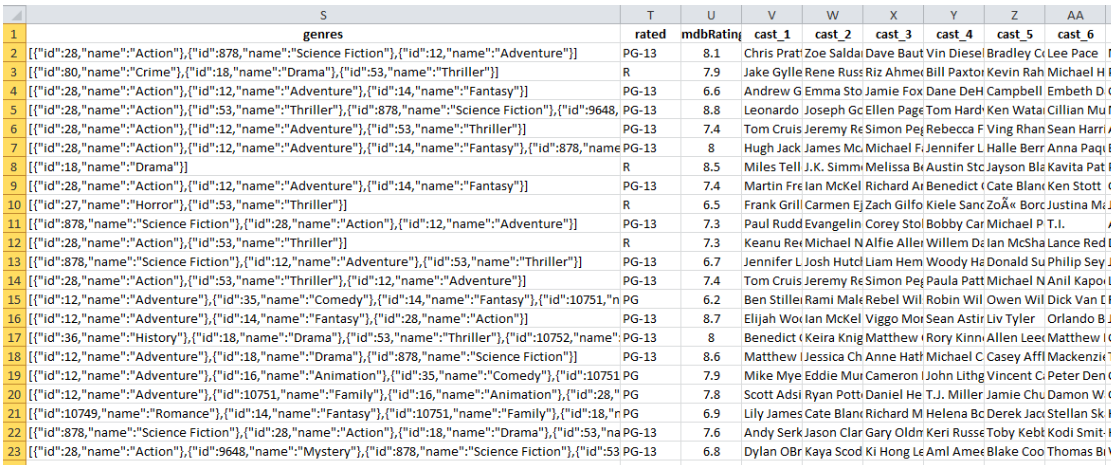


Figure 6.11- Script to Remove Noise

6.7 Building Logic and Rearrange Data to Predict Success

We had to remove most of our data, because of the unavailability of revenue and budget details. It is almost 85% of the dataset. To build logic for that 85% of data, we carried out a few studies to identify the movie revenue, positively correlated with the IMDB rating using Microsoft Excel and identified it is not correlated. As such, we decided to proceed with the remaining dataset. We took a class variable as HIT_FLOP. If a movie's revenue was greater than its budget, we took it as a HIT. Others, we labelled as FLOPS.

The data we acquired using web data scraping, are not arranged in a suitable way to use WEKA for mining as in *figure 6.12*.



	S	T	U	V	W	X	Y	Z	AA
1	genres	rated	mdbRating	cast_1	cast_2	cast_3	cast_4	cast_5	cast_6
2	[{"id":28,"name":"Action"},{"id":878,"name":"Science Fiction"},{"id":12,"name":"Adventure"}]	PG-13	8.1	Chris Pratt	Zoe Saldai	Dave Baut	Vin Diesel	Bradley C	Lee Pace
3	[{"id":80,"name":"Crime"},{"id":18,"name":"Drama"},{"id":53,"name":"Thriller"}]	R	7.9	Jake Gylle	Rene Russ	Riz Ahmed	Bill Paxtoi	Kevin Rah	Michael H
4	[{"id":28,"name":"Action"},{"id":12,"name":"Adventure"},{"id":14,"name":"Fantasy"}]	PG-13	6.6	Andrew G	Emma Sto	Jamie Fox	Dane DeH	Campbell	Embeth D
5	[{"id":28,"name":"Action"},{"id":53,"name":"Thriller"},{"id":878,"name":"Science Fiction"},{"id":9648,	PG-13	8.8	Leonardo	Joseph Gc	Ellen Page	Tom Hard	Ken Watai	Cillian Mu
6	[{"id":28,"name":"Action"},{"id":12,"name":"Adventure"},{"id":53,"name":"Thriller"}]	PG-13	7.4	Tom Cruis	Jeremy Re	Simon Peg	Rebecca F	Ving Rhan	Sean Harri
7	[{"id":28,"name":"Action"},{"id":12,"name":"Adventure"},{"id":14,"name":"Fantasy"},{"id":878,"name	PG-13	8	Hugh Jack	James Mc	Michael F	Jennifer L	Halle Berr	Anna Paqu
8	[{"id":18,"name":"Drama"}]	R	8.5	Miles Tell	J.K. Simm	Melissa B	Austin Stc	Jayson Bla	Kavita Pat
9	[{"id":28,"name":"Action"},{"id":12,"name":"Adventure"},{"id":14,"name":"Fantasy"}]	PG-13	7.4	Martin Fre	Ian McKel	Richard Ar	Benedict	Cate Blanc	Ken Stott
10	[{"id":27,"name":"Horror"},{"id":53,"name":"Thriller"}]	R	6.5	Frank Grill	Carmen Ej	Zach Gilfo	Kiele Sanc	ZoÃ« Borc	Justina Ma
11	[{"id":878,"name":"Science Fiction"},{"id":28,"name":"Action"},{"id":12,"name":"Adventure"}]	PG-13	7.3	Paul Rudd	Evangelin	Corey Stol	Bobby Car	Michael P	T.I.
12	[{"id":28,"name":"Action"},{"id":53,"name":"Thriller"}]	R	7.3	Keanu Ree	Michael N	Alfie Aller	Willem Da	Ian McSha	Lance Red
13	[{"id":878,"name":"Science Fiction"},{"id":12,"name":"Adventure"},{"id":53,"name":"Thriller"}]	PG-13	6.7	Jennifer L	Josh Hutcl	Liam Hem	Woody Ha	Donald Su	Philip Sey
14	[{"id":28,"name":"Action"},{"id":53,"name":"Thriller"},{"id":12,"name":"Adventure"}]	PG-13	7.4	Tom Cruis	Jeremy Re	Simon Peg	Paula Patt	Michael N	Anil Kapo
15	[{"id":12,"name":"Adventure"},{"id":35,"name":"Comedy"},{"id":14,"name":"Fantasy"},{"id":10751,"n	PG	6.2	Ben Stille	Rami Male	Rebel Wil	Robin Wil	Owen Wil	Dick Van D
16	[{"id":12,"name":"Adventure"},{"id":14,"name":"Fantasy"},{"id":28,"name":"Action"}]	PG-13	8.7	Elijah Wo	Ian McKel	Viggo Mor	Sean Astir	Liv Tyler	Orlando B
17	[{"id":36,"name":"History"},{"id":18,"name":"Drama"},{"id":53,"name":"Thriller"},{"id":10752,"name":	PG-13	8	Benedict	Keira Knig	Matthew	Rory Kinn	Allen Lee	Matthew
18	[{"id":12,"name":"Adventure"},{"id":18,"name":"Drama"},{"id":878,"name":"Science Fiction"}]	PG-13	8.6	Matthew	Jessica Ch	Anne Hath	Michael C	Casey Affl	Mackenzie
19	[{"id":12,"name":"Adventure"},{"id":16,"name":"Animation"},{"id":35,"name":"Comedy"},{"id":10751	PG	7.9	Mike Mye	Eddie Mur	Cameron	John Lithg	Vincent C	Peter Den
20	[{"id":12,"name":"Adventure"},{"id":10751,"name":"Family"},{"id":16,"name":"Animation"},{"id":28,"	PG	7.8	Scott Adsi	Ryan Pott	Daniel He	T.J. Miller	Jamie Chu	Damon W
21	[{"id":10749,"name":"Romance"},{"id":14,"name":"Fantasy"},{"id":10751,"name":"Family"},{"id":18,"n	PG	6.9	Lily James	Cate Blanc	Richard M	Helena Bc	Derek Jac	Stellan Sk
22	[{"id":878,"name":"Science Fiction"},{"id":28,"name":"Action"},{"id":18,"name":"Drama"},{"id":53,"na	PG-13	7.6	Andy Serk	Jason Clar	Gary Oldn	Keri Russe	Toby Kebl	Kodi Smit-
23	[{"id":28,"name":"Action"},{"id":9648,"name":"Mystery"},{"id":878,"name":"Science Fiction"},{"id":53	PG-13	6.8	Dylan OBr	Kaya Scod	Ki Hong Le	Aml Amee	Blake Coo	Thomas B

Figure 6.12-Acquired Data Structure

As such, there was a requirement to rearrange the dataset to ensure suitability when using WEKA. To do this, we created another Node.js script as *in figure 6.13*. By doing so, we were ended up with 20,000+ attributes as in *figure 6.14*.

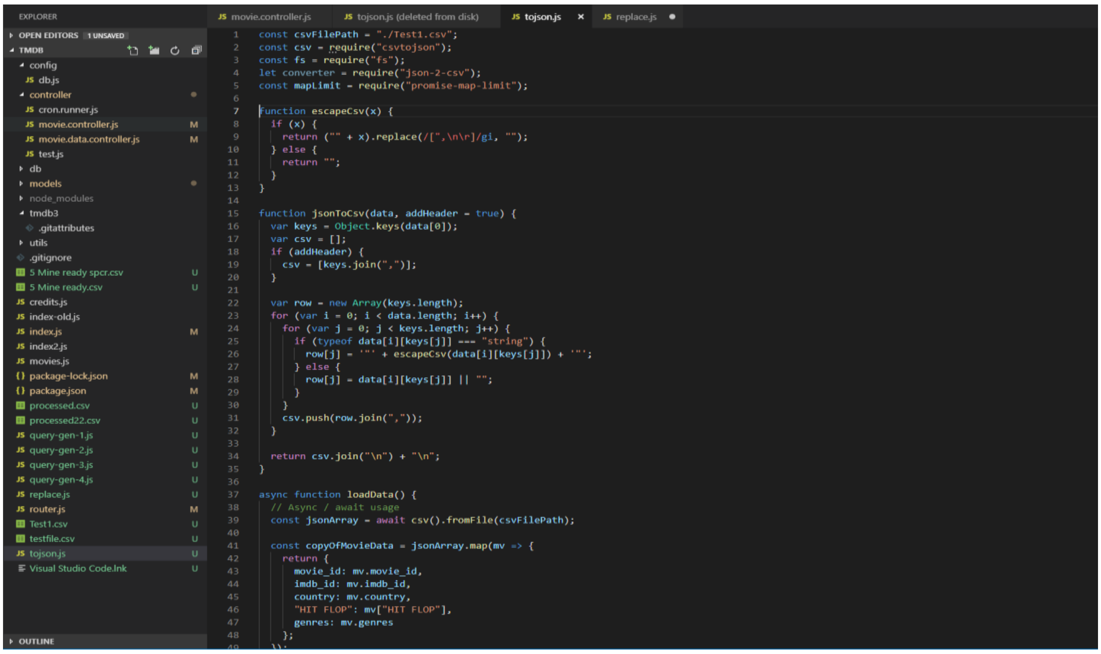


Figure 6.13 –Script used to Rearranged Data

	A	B	C	D	E	F	G	H	I	J	K
1	movie_	imdb_i	country	hit flop	genres	50 Cent	A Martinez	A. A. Milne	A. Ali Flores	A. Arnold Gillespie	A. Demetrius Brown
336	174645	tt2083231	USA	FLOP	89	0	0	0	0	0	0
337	469856	tt4768794	USA	FLOP	89	0	0	0	0	0	0
338	25602	tt1182921	USA	FLOP	89	0	0	0	0	0	0
339	25594	tt1393000	USA	FLOP	89	0	0	0	0	0	0
340	70586	tt1748197	USA	FLOP	89	1	0	0	0	0	0
341	139567	tt1925431	USA	FLOP	89	1	0	0	0	0	0
342	80188	tt1456060	USA	HIT	89	0	0	0	0	0	0
343	21948	tt0076637	USA	HIT	89	0	0	0	0	0	0
344	360203	tt3593046	USA	HIT	89	0	0	0	0	0	0
345	22907	tt1135084	USA	HIT	89	0	0	0	0	0	0
346	949	tt0113277	USA	HIT	89	0	0	0	0	0	0
347	4597	tt0913354	USA	HIT	90	0	0	0	0	0	0
348	407874	tt1829654	USA	HIT	91	0	0	0	0	0	0
349	187799	tt1742682	USA	FLOP	92	0	0	0	0	0	0
350	28019	tt1258137	USA	FLOP	93	0	0	0	0	0	0
351	66193	tt1130969	USA	Average	94	0	0	0	0	0	0
352	76640	tt1549920	USA	Average	94	0	0	0	0	0	0
353	37860	tt1569369	USA	FLOP	94	0	0	0	0	0	0
354	208242	tt2663744	USA	FLOP	94	0	0	0	0	0	0
355	150230	tt1928335	USA	FLOP	94	0	0	0	0	0	0
356	169298	tt2544734	USA	FLOP	94	0	0	0	0	0	0
357	259138	tt0835775	USA	FLOP	94	0	0	0	0	0	0
358	346592	tt2134170	USA	FLOP	94	0	0	0	0	0	0
359	380754	tt4287348	USA	FLOP	94	0	0	0	0	0	0

Figure 6.14 -Rearranged Data set having over 20,000+ Attributes

These scripts are high memory consuming scripts. We used server hardware with 32GB of ram to run these scripts in the early stages. Then, we were able to optimise the script to run on a local PC. Movie genre is also a multivalued attribute. A particular movie can have multiple genre labels.

In order to use it in an informative way, we replace genre by giving a categorical value as in *figure 6.14*. Categorical genre values are assigned by referring a distinct genre table, which we derived from the original dataset as in *figure 6.15*.

	A	B
1	***genres***	***Category***
2	Animation:Action:Comedy:Family	1
3	Thriller:Drama	2
4	Horror:Thriller	3
5	Drama:Fantasy:Music	4
6	Comedy:Family	5
7	Action:Adventure	6
8	Drama:Science Fiction:Mystery	7
9	Action:Crime:Drama:Thriller	8
10	Horror	9
11	Comedy	10
12	Drama:Music	11
13	Drama	12
14	Action:Western:Drama:Fantasy:Thriller:	13
15	Romance:Drama:Family	14
16	Drama:Horror:Mystery:Thriller	15
17	Comedy:Animation:Family	16
18	Mystery:Horror:Thriller	17
19	Action:Science Fiction:Thriller	18
20	Music:Drama	19
21	Action:Comedy:Horror:Mystery:Science Fiction:Thriller	20
22	Comedy:Drama:Romance	21
23	Adventure:Fantasy:Action	22
24	Drama:Thriller	23
25	Thriller:Mystery	24

Figure 6.15 – Categories of Distinct Movie Genre Values

As explained earlier, once we rearranged the dataset, there were 20,000+ attributes in total. We then used a supervised feature selection technique to identify class influence attributes. To do that, we used Weka's supervised "CfsSubsetEval" filter and we were ended up with a limited attribute set as in *figure 6.16*. The dataset is now statistically ready to experiment with Weka's data classification techniques.

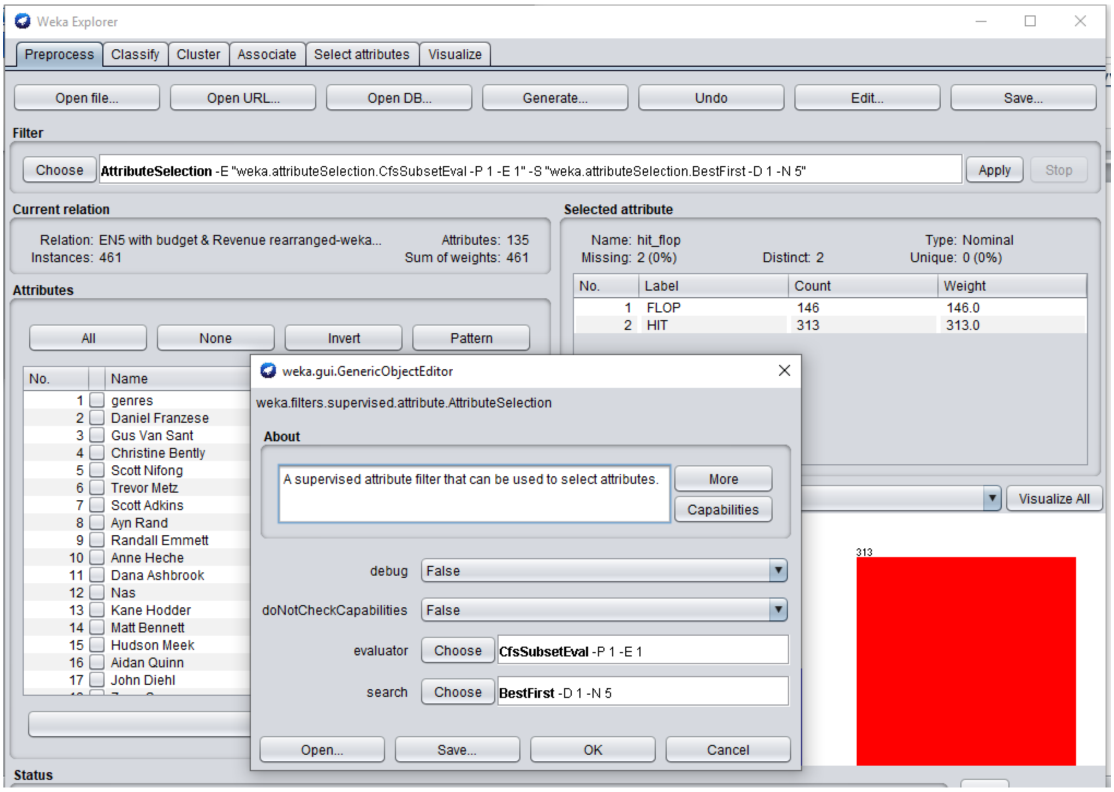


Figure 6.16.-Feature selected Data set with 100+ Attributes

6.8 Summary

This chapter provided the overall implementation details of each module of the proposed solution. Moreover, it mentioned software and data mining techniques for the model's development with aligning it to design. The next chapter evaluates all the modules implemented in the solution.

Chapter 7

Evaluation

7.1 Introduction

In the previous chapter, we discussed module implementation in detail. This chapter justifies and evaluates the overall solution.

7.2 Evaluation of Classification Techniques

With the aid of the Weka tool, we evaluated different classification techniques. The techniques we evaluated were: Naïve Bayes, J48, Bagging and Random Forest. For evaluating a classifier's accuracy, recall and precision, we used a confusion matrix *(Appendix C)*. Formulas used to calculate the confusion matrix are shown in figure 7.1.

Measurement	Formula	Description
Precision	TP/ (TP + FP)	Correct Positive predictions percentage
Recall Sensitivity	TP / (TP + FN)	The percentage of positive labelled instances that were predicted as Positive.
Specificity	TN / (TN + FP)	The percentage of negative labelled instances that were predicted as Negative.
Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions those Are correct.

Figure 7.1.-Identifying Confusion Matrix Parameters and Formulas

Using the aforementioned measurements, we can deduce the TP-rate, FP-rate, F-measure and ROC area. The TP-rate describes sensitivity, while the FP-rate is equal to 1. A perfect test has an area of 1.00. Usually the best models have a higher TP rate, lower FP rate and ROC close to 1.00.

For this evaluation, we used the below-mentioned classifiers. For classification, we used cross-validation techniques with ten folds. *Figure 7.2* shows the results we got by using different classification techniques.

Technique	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC**
Naïve Bayes	0.739	0.553	0.791	0.739	0.673	0.923
Decision Tree	0.758	0.515	0.814	0.758	0.704	0.626
AdaBoost	0..682	0.682	N/A	0.682	N/A	0.482
Bagging	0.660	0.619	0.599	0.660	0.599	0.561
Random Forest	0.671	0.654	0.592	0.671	0.578	0.695

Figure 7.2-Result Evaluation – Different Classification Techniques

According to the results, Naïve Bayes and Decision Tree gave almost equal accuracy, but the Decision Tree classification gave higher accuracy value than Naïve Bayes. So, we can say that Decision Tree produced the best results in predicting movie success rates for this dataset. Other classification techniques as shown in the above table, did not perform significantly well in our research. The Weka model creation summary is attached to *(Appendix D)*.

7.3 Summary

In this section, we did a detailed evaluation of our build models. We experimented and evaluated different types of classifiers. Out of all discussed classifiers, Decision Tree performs well for this dataset. The next chapter is reserved for conclusions of the project and future expansions.

Chapter8

Conclusion and Further Work

8.1 Introduction

In this research, we have addressed the problem of predicting movie success rates for theatre released English movies in the USA. By analysing movie attributes, we have been able to predict a movies success/flop.

The main goal of this research is to identify a model for predicting a movies success rate. There are two main segments to our prediction. One is by taking the movie attributes like actors, directors etc and the other one is by taking the multivalued genre details of movies. To do this, we replaced genre details with categorical values (ex: 1-for action films, 2 -for comedy films, Etc.). This way it helps us to simulate the movie properly. We selected the Decision Tree algorithm over others to predict a movies success rate. Then we evaluated our model for quality and accuracy by using an extensive set of experiments on real movie data. The main contribution of our work can be listed as follows:

- Comparison of machine learning techniques which revealed that classification is the best approach to solve the problem.
- Evaluation of various classifiers over real data to prove that the Decision Tree (Weka-J48) works best for the selected data set.
- Multi-valued Genre details considered when predicting a movies success rate.

8.2 Limitations

By default, movie success is unpredictable. There are some situations we cannot imagine. As an example, with the death of “Stan Lee” (American comic book writer 1922 - 2018), Marvel movies became very popular. In addition to that, a new actor may cause a movie to become a hit. Events and news about celebrities may make movies more attractive. Those social factors and their impact on a movie’s success, cannot be predicted using this approach.

8.3 Future Developments

As future work, we are planning to expand our analysis by taking more attributes into account (Box office, screened duration etc).

In this research, we did not address any special situations. We did not consider the role (: the director is an actor of a given movie). It is open for research. Even though movie makers select the best crew, there are other factors that affect the screening movies e.g.: movie trailers, advertising mechanism, main actor/actress‘s social life and behaviour etc. Still there are gaps available to do further researches.

8.4 Summary

This chapter concludes the thesis by describing the solution given with data mining to analyse the movies success prediction.

7. References

[1] J. van der Merwe and B. Eimon, "Predicting Movie Box Office Gross," *Stanford University*, 2013.

[2] A. Kennedy, "Predicting box office success: Do critical reviews really matter?," *Berkeley Projects*, 2008.

[3] J. Ericson and J. Grodman, "A Predictor for Movie Success," *CS229, Stanford University*, 2013.

[4] R. Sharda and D. Delen, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, pp. 243-254, 2006.

[5] P. A. Gloor, J. Krauss, S. Nann, K. Fischbach, and D. Schoder, "Web science 2.0: Identifying trends through semantic social network analysis," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, 2009, pp. 215-222.

[6] R. Panaligan and A. Chen, "Quantifying movie magic with a google search," *Google Whitepaper—Industry Perspectives+ User Insights*, 2013.

[7] E. Sadikov, A. G. Parameswaran, and P. Venetis, "Blogs as Predictors of Movie Success," in *ICWSM*, 2009.

[8] L. Jaehoon, N. Giseop, and K. Chong-Kwon, "Analysis andamp; visualization on movie's popularity and reviews," in *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, 2014, pp. 189-190.

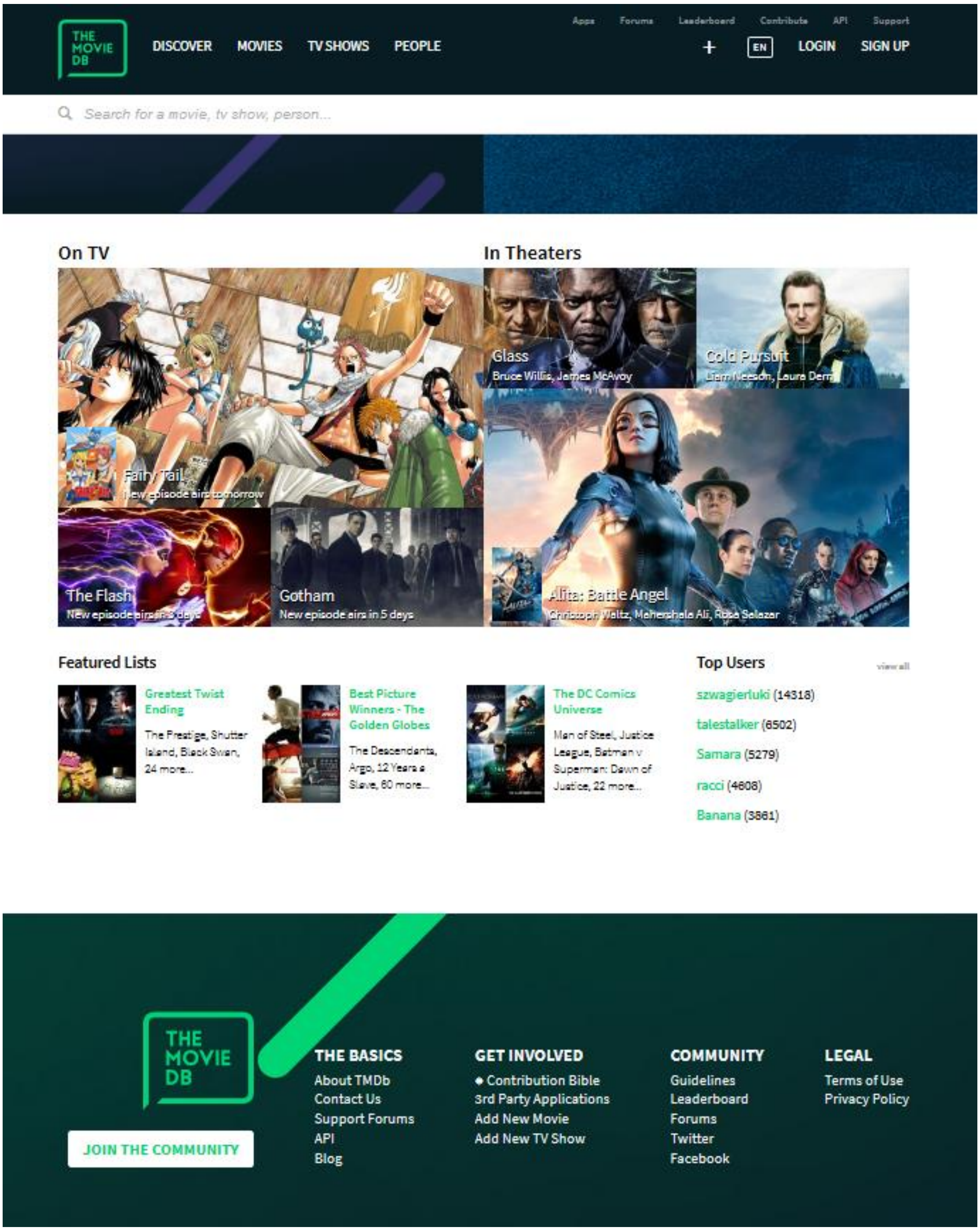
[9] G. He and S. Lee, "Multi-model or Single Model? A Study of Movie Box-Office Revenue Prediction," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 321-325.

[10] www.themoviedb.org.

[11] www.themoviedb.org.

[12] www.imdb.com.

Appendix A - The Movie Database - TMDB



Appendix A: 1 Web Site: The Movie Database (TMDB)

Appendix B - OMDB API

OMDb API Usage Parameters Examples Change Log API Key Become a Patron Contact

OMDb API

The Open Movie Database

The OMDb API is a RESTful web service to obtain movie information, all content and images on the site are contributed and maintained by our users.

If you find this service useful, please consider making a one-time donation or become a patron.

Poster API

The Poster API is only available to patrons.

Currently over 280,000 posters, updated daily with resolutions up to 2000x3000.

Attention Users

01/20/19 - Supressed adult content from search results.

01/20/19 - Added Swagger files (YAML, JSON) to expose current API abilities and upcoming REST functions.

Become a Patron

Sponsors

Emby, Trakt, Fullscreen, FileBot, WMSIWT, Galvanize, Direktpoint, Cordcutting.com, rrbone, Free TV Mov, Frontend Masters, NetflixReleases, Reelgood, Xirvik Servers, ALG Websites, MassFlix, Yidio, Indexed, mi.tv, Couchpop, Key Video, What's on Netflix, CelebrityHow, iFlicks, MyIPTV, Datantify

Usage

Send all data requests to:

```
http://www.omdbapi.com/?apikey=[yourkey]&
```

Poster API requests:

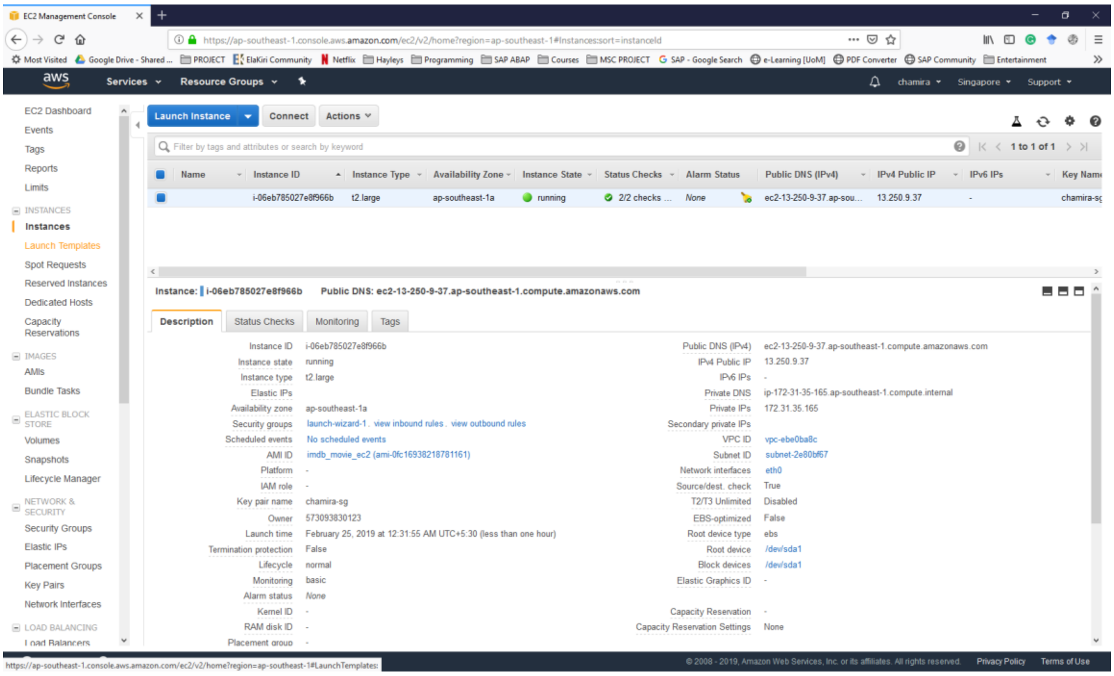
```
http://img.omdbapi.com/?apikey=[yourkey]&
```

Parameters

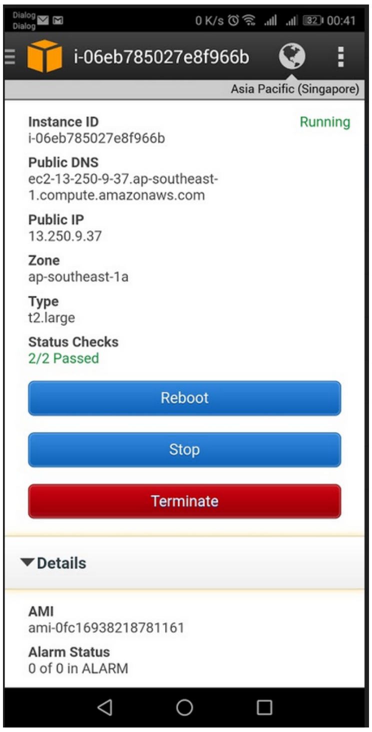
By ID or Title

Parameter	Required	Valid Options	Default Value	Description
i	Optional*		<empty>	A valid IMDb ID (e.g. tt1285016)
t	Optional*		<empty>	Movie title to search for.
type	No	movie, series, episode	<empty>	Type of result to return.
y	No		<empty>	Year of release.
plot	No	short, full	short	Return short or full plot.
r	No	json, xml	json	The data type to return.
callback	No		<empty>	JSONP callback name.
v	No		1	API version (reserved for future use).

Appendix B: OMDB API with Parameters

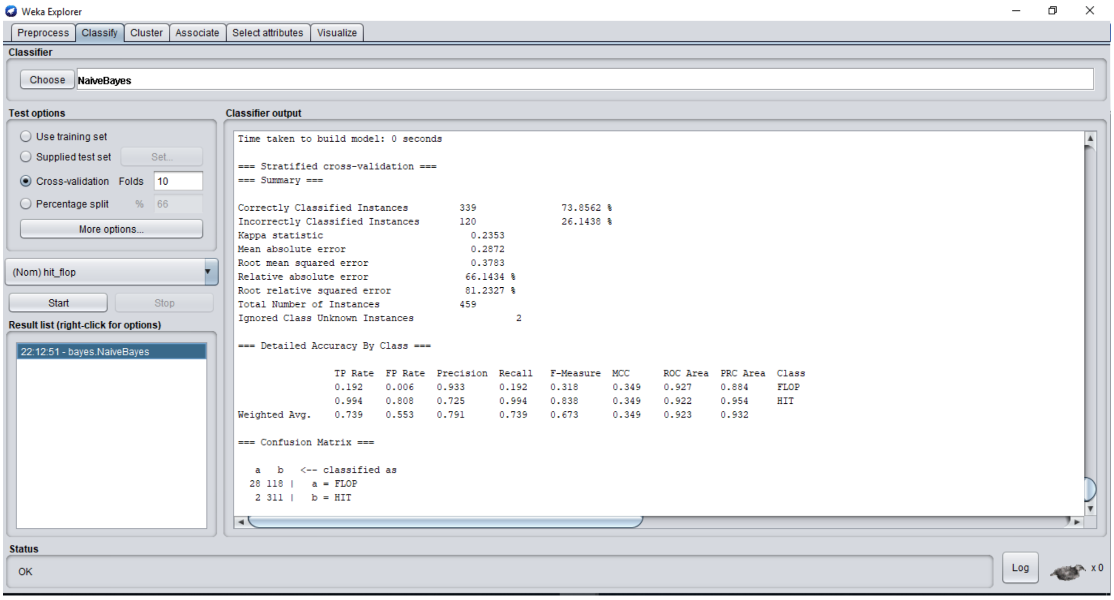


Appendix C: 1 Amazon Web T2. Large Server Instance

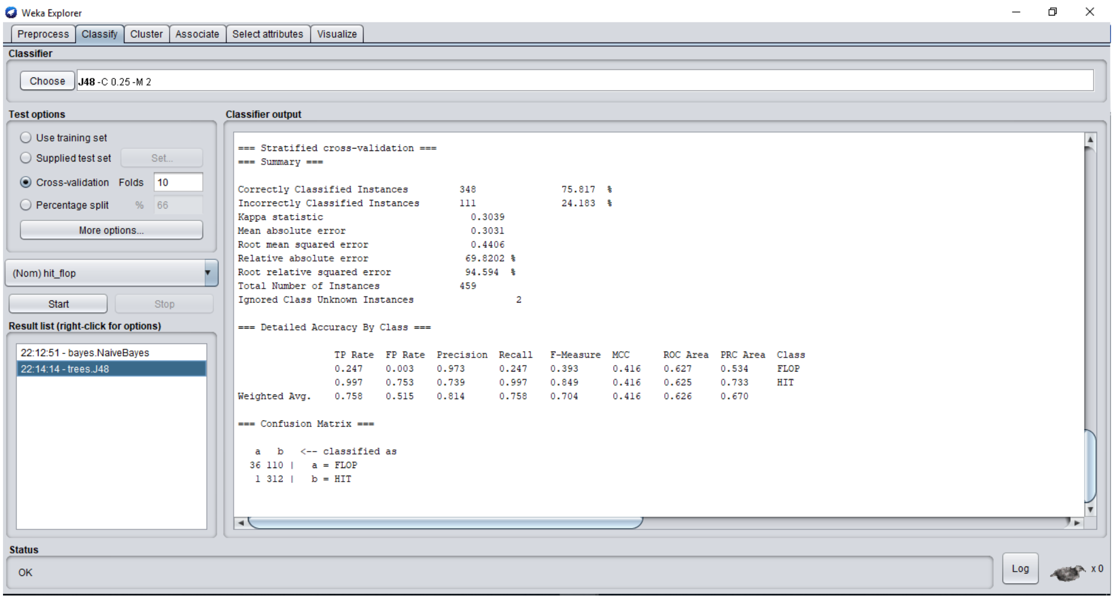


Appendix C: 2 Amazon Server Administration Console - Mobile

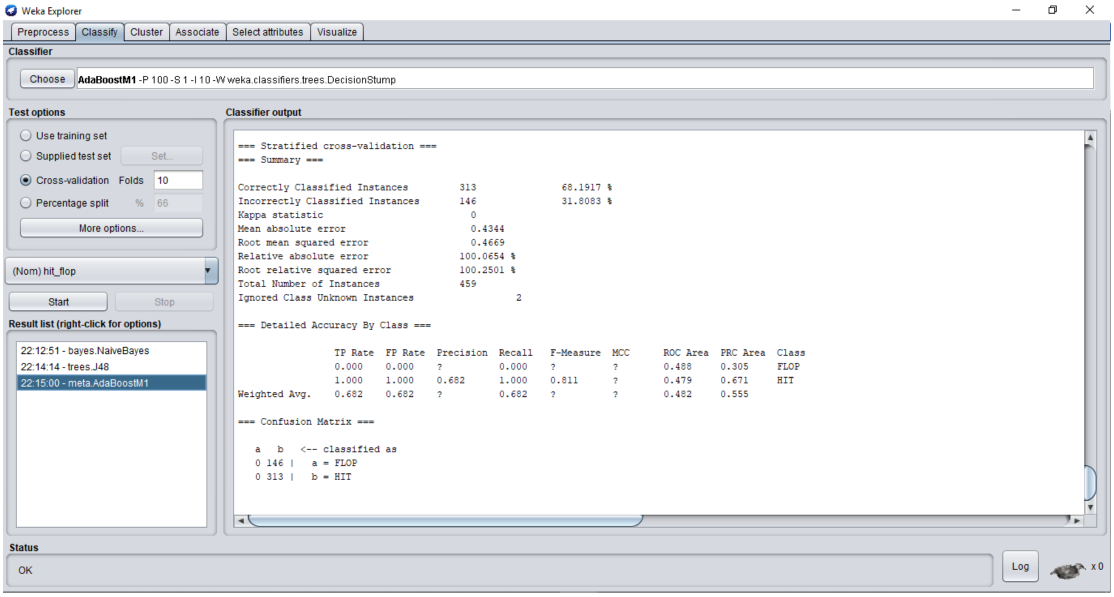
Appendix D - Model Evaluation Summary



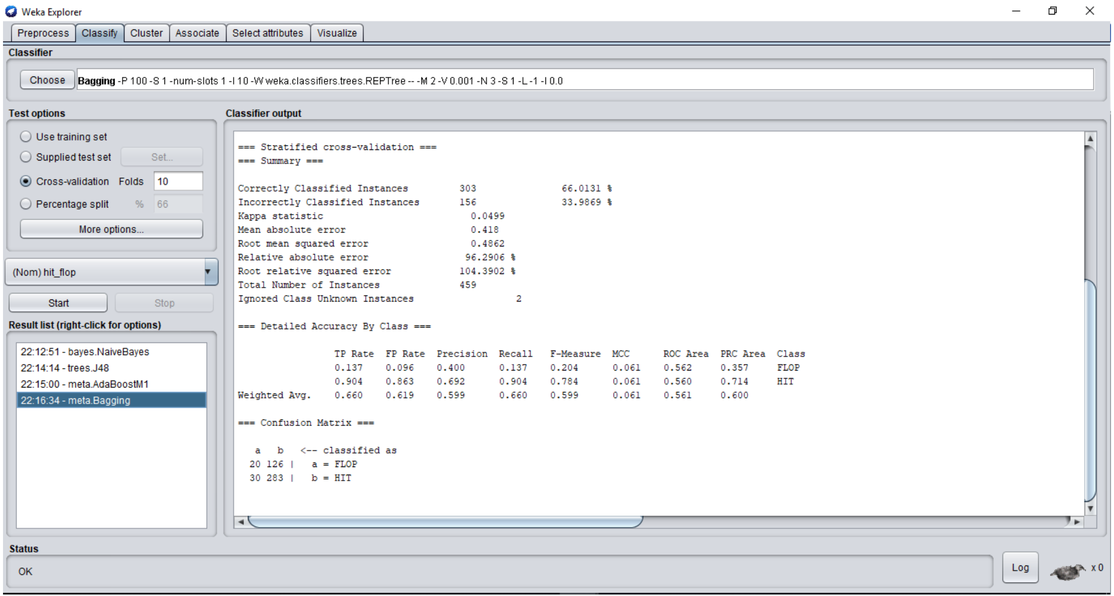
Appendix D: 1 Naive Bayes Model Creation Summary



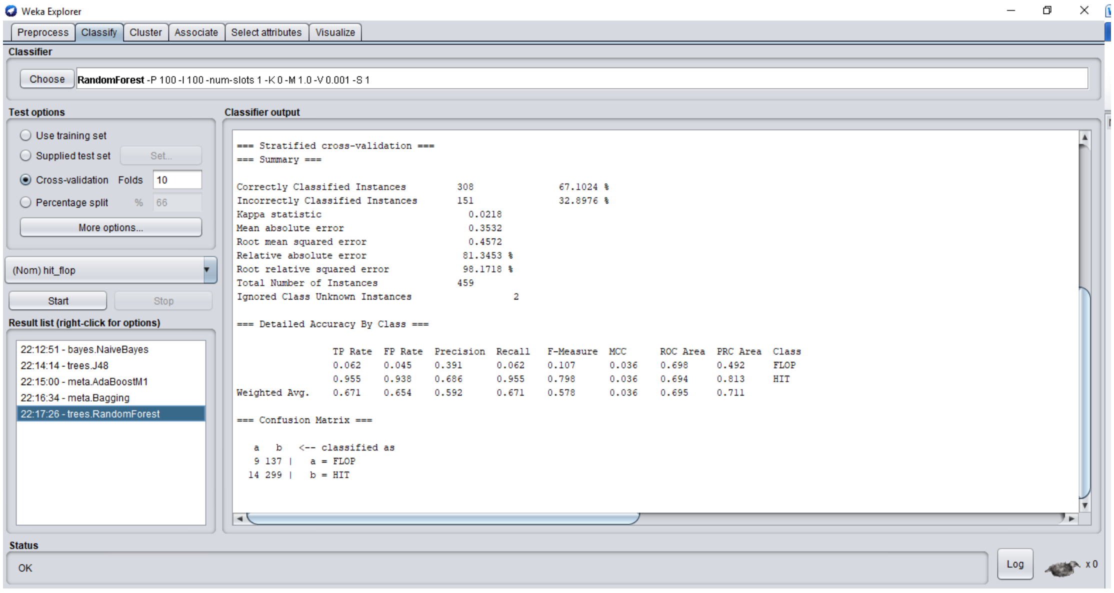
Appendix D: 2 Decision Tree Model Creation Summaries



Appendix D: 3 AdaBoost Model Creation Summaries



Appendix D: 4 Bagging Model Creation Summaries



Appendix D: 5 Random Forest Model Creation Summaries