OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

Predicting Movie Success with Data Mining
Interim Report

DT228
BSc in Computer Science

Povilas Kubilius

C16370803

Leo Tilson

School of Computer Science

Technological University, Dublin

09/12/2019

Abstract

The goal of this project is to predict the success of a planned movie, what ratings it will get, how much income it will make in the box office, based on planned input variables such as movie budget, genera and cast. This will be a web application, where users will be able to fill out these planned movie details and my models will make a prediction and show the user what type of rating and revenue the planned movie could get. The data will be gathered about movies from available to download datasets and use web scraping techniques to fill in any gaps in the data and acquire any additional needed information. After cleaning and processing my datasets, the movie data will be used to train an artificial neural network model to make predictions on the success of movies. The project end goal is to host the model online on the interactive web application. Using the entries from the end user, web scrape any necessary additional metadata about the entries, such as how many awards does the actor which was inputted by the user have, or how many movies has a director directed and what ratings those movies had, and then use my pre-trained models to make the predictions.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

____

Povilas Kubilius

Date

Acknowledgements

I would like to thank my family and friends for the support during the development of this project. I would like to acknowledge and thank my project supervisor for the support and guidance during the weekly meet ups. His help and insightful advice were invaluable in completing the work done so far on this project and on this report.

Table of Contents

1. Introduction

1.1. Project Background

What makes a movie successful? We have all watched movies that evoked strong emotions in us. Perhaps it was the relatable characters, or maybe a resonant ideology that was presented, or maybe simply the visuals and art in the movie were so impressive that it left its mark on you. Storytelling is a core aspect of our cultures. We have told stories for as long as we can remember, and we don't even know what was the first story ever told [1]. Even back into prehistory, humans drew pictures on cave walls to tell stories. As humans evolve and develop more advanced tools and civilizations, so do the methods of storytelling developed. Before civilization, we told stories only by word of mouth, then with the discovery of fire we also discovered that charcoal is great for making marks on stone walls, Chauvet Cave is probably one of the most famous examples of prehistoric cave paintings. Later with the discovery of writing, it transformed how we tell stories again, taking on the form of symbols on clay tablets. One of most ancient surviving literary work we have today is the Ancient Sumerian "Epic of Gilgamesh" [2], composed in Mesopotamia around 1800 BC, but only second after the Pyramid Texts in Egypt which have been dated to about 2400–2300 BC [3]. Surely these stories didn't survive thousands of years only because they were written on persistent material, because if that were the case then we would a lot more clay tablets of ancient literature. It is more likely that the Epic of Gilgamesh was such a profound tale to the ancient people of Mesopotamia that they made effort to make copies of the clay tablet and put in effort into keeping them safe for preservation. Today the most advanced form story telling comes in the form of digital media, such as movies, TV series and even video games.

But what makes a story popular, successful and with more chance to survive into the future for future generations to hear or watch these stories? Is it solely dependent the story itself? Do all good stories become successful based on their merit alone or maybe there are more factors involved? Perhaps the way the story is told is more important than the story itself? Two different people can retell the exact same story and one can bore us to death, and the other can grip our attention with such intensity that we sit on the edge of our seats, totally immersed in the plot of the story being told. The same is probably also true of movies. The production, cast and delivery of the movie can have a major impact on the movie's ability to grip us and leave a strong impression on us or leave us bored, forgetting about the movie after a few days. This project aims to explore the relationships between production variables in making a movie and how successful the movie was after its release. How does one measure the success of a movie in the first place? Everyone has their own subjective view on which movie was amazing and which ones disliked, but yet there is a general consensus of the majority on which movies were generally better than others. Main gauge used in determining this are movie ratings and reviews. Ratings by the regular audience and ratings by movie critics. This places a numerical value on movie success. One can even consider the revenue the movie produces as a measure of success. By placing numerical values on production variables, such as the budget to produce the movie, the amount of awards the main actors have, the success of previous movies directed by the director, we can use computational data analytic techniques to spot patterns and correlations between production variables and the variables which dictate the success of the movie.

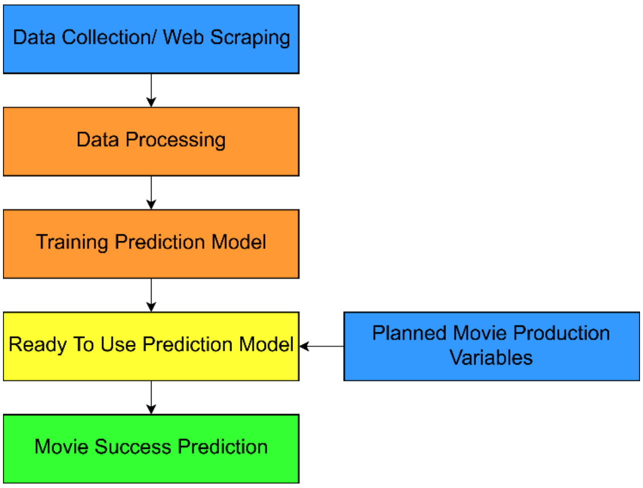
1.2. Project Description

This project will use artificial neural networks and machine learning to learn about the production variables and the success of thousands of movies from the past, and using these models predict the success outcome of planned movies once given the planned production variables, such as the budget of the movie, its runtime, the main actors, writers, directors, genre, release date and other relevant factors.

To be able to use this system, it will be made in the form of an online web application. The end users will have an interactive and friendly user interface to fill out the details of the planned production variables and then get an estimate on what the success of the movie would be based on the user's input.

To make the model in the first place, the key aspect of this project will be data acquisition, which then we can mine to acquire valuable information from. Basic datasets will be acquired online from sites like kaggle.com and IMDb. But these datasets will not be enough, to gain more relevant data it will use web scraping techniques to gather relevant data from movie websites like IMDb and Rotten Tomatoes.

The datasets will then be processed into a single table or file in a format that will be readily acceptable to train the artificial neural network. This will include hot encoding categorical data into binary representations, normalizing highly variant data like budget and runtime and placing the data into formatted arrays. The neural network model will train on the provided dataset to predict movie success and will also be tested for accuracy on a subset of the dataset which the model has not been trained with.


Data Collection/ Web Scraping
Data Processing
Training Prediction Model
Ready To Use Prediction Model
Planned Movie Production Variables
Movie Success Prediction

The end system will have a client-server architecture. The client side will be what the end users will interface will. The processing of input data, the prediction model and the database will be located on the server. The server will also use web scraping to get additional needed data. Even though the server will have a database of movie data, some variables, like how many awards won by an actor, can change over time, so the database may not be up to date. The server will web scrape for that data and use it as input for the prediction model to get an estimate on the movie success and return to the result to be visually viewed by the end users.

1.3. Project Aims and Objectives

The overall aim of the project is to make an estimate prediction of movie ratings with accuracy greater than 50% and to provide a web-based user interface for the system that anyone can use.

In order to have the most accurate possible predictions, a large amount of data is required. Another objective is to compile a full dataset of over 30,000 movies with data about their budget, genre and feature actors. Web scraping will be used to search online movie pages, like IMDB movie pages, for the missing data to create a bigger, completer and more accurate dataset.

The main objective is to create an artificial neural network, that will be trained using data from the database, to make new predictions about planned movie's ratings and box office income, using user inputted data. This will be done by using python libraries which make neural network implementation accessible and flexible.

The end goal for this project will be to make this widely available by having a web application to host this system online and make it easily accessible by everyone on most deceives. This can be broken down into 3 smaller objectives:

- To create web based front-end, that is user friendly, easy to use and is accessible from most deceives, including mobile phones.
- To create a web server which holds all the logic to process the user input and use that input to make movie predictions with the neural network model.
- To create a database back-end that holds data about movies and actors. The database will need to be updated one a week to ensure that the data is current and accurate.

1.4. Project Scope

This project is a probabilistic estimate of the movie's success. It cannot predict with certainty. The project also does not take into account the actual quality of the content of the movie. It doesn't analyse the plot, complexity of the story, character development and quality of acting by the actors. It doesn't analyse the style of the movie or its visuals. It doesn't analyse the content of the movie's scripts, or the title of the movie or its summary. To predict movie success based on its content it would require the movie to have already been filmed and edited for analysis, which at that point is impractical because one would want to know a rough estimate of the movie success before any money is invested into its production. Using sentimental or thematic analysis for things like movie title, plot summary or script doesn't return meaningful numerical data that be used to train an artificial neural network. This project doesn't predict success for TV series. The success of a TV series is highly variable and can depend on season and even episode. First season of a TV series can be highly successful then dwindle over time, so having only initial production data cannot predict the success of a TV series over the long term.

1.5. Thesis Roadmap

Research

This chapter explores the background research done related to the use of datamining and artificial neural networks in analysing data and predicting trends, like movie success. Looking at examples of software that already use artificial intelligence in different areas of the movie production business.

Design

This chapter discusses the planned designed for the overall web application system. Exploring the technologies used and the dataflow between the multi-layered online architecture. This chapter also discusses the software development methodologies that will be used using the development of this project.

Prototype Development

This chapter shows the current progress of the development on this project so far. Explaining the techniques being used with the current approach of web scraping and implementation of artificial neural network, with examples of code.

Testing and Evaluation

This chapter outlines the methods which will be employed in testing the software. It also discusses how the web application will be evaluated by test users and what metrics will be used to measure project evaluation.

Issues and Future Work

This chapter will discuss the current issues and challenges faced in the project so far and planned approaches to solve these issues. Here we also represent the risks associated with the project along with planned approaches on mitigating these risks. Lastly, we discuss the future planned for this project along with a GANNT chart visually representing the scheduled work for the next few months.

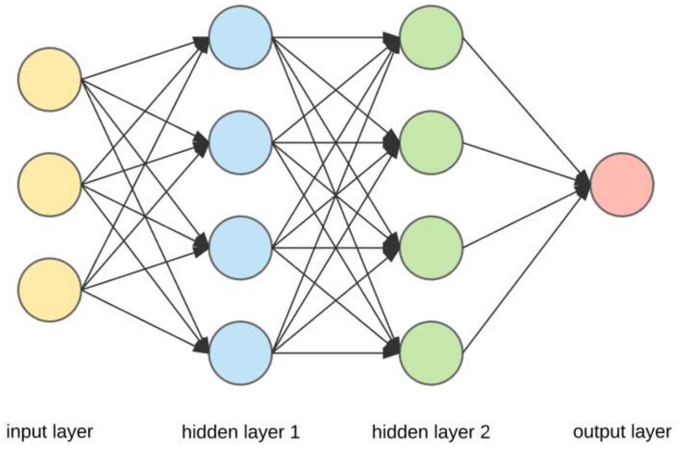
2. Literature Review

2.1. Introduction

This chapter explores the background research related to artificial neural networks, data mining and role of artificial intelligence in movie success prediction. There are a lot of problems in the world and for most of human history it was the humans themselves who could solve their problems. This has worked very efficiently for thousands of years as this is how we developed and progressed our civilisation from the stone age to the modern age of information and technology. As our inventions and civilization became more complex, so have our problems. We now have access a tremendous amount of raw data that we collect with our technology but trying to extrapolate meaningful information about the data is becoming increasing more difficult. Dealing with Big Data is becoming a new problem. Manually analysing Big Data and hoping to find patterns and extracting useful information to make prediction is impractical for humans alone. For example, given 30,000 rows of movies with many columns of metadata about the movie, how would it be possible to predict a planned movie's ratings or box office income? This is where the use of artificial intelligence comes into play. The most powerful aspect of artificial intelligence comes from machine learning techniques such as the use of artificial neural networks. Using the data acquired, we can train the neural networks to find and recognize patterns in the data, creating a model which can be used to make predictions from new observations as inputs. The whole processes of acquiring large data sets, processing them and using artificial intelligence to extract useful information and make future predictions is called data mining.

Artificial Neural Networks

Artificial neural networks are statistical/mathematical models that try to imitate real biological neural networks like the human brain. Artificial neural networks are made up of interconnected nodes, or "neurons", separated out into layers. The connections between the nodes are called synapses, which send signals from one neuron to another. The synapses also have an assigned "weight" to it. Each neuron has an activation function, it calculates the total sum of the weights it received from signals from other neurons, and if the sum is greater than the threshold, it actives and send a single to each neuron it's connected to in the next layer.



What makes Artificial Neural Networks very powerful is backpropagation. When the output of the model does not match the expected output during training, it calculates the degree of error from inputs, expected output and actual output, then iterates backwards through the networks, adjusting the weights and biases of the synapses, in hopes that this will create an output to more closely match the expected output.

Once an Artificial Neural Network has been sufficiently trained, we can input new data into the network to see what the predicted output is. This project will use an artificial neural network, trained on datasets of metadata of previously made movies to then make predictions with new given input.

Data Mining

Data mining is the whole process of finding patterns in large dataset and extracting useful new information. Data mining involves using artificial intelligence, machine learning and statically analysis. Data mining also involves databases and data management, working with pre-existing datasets, pre-processing the data, using machine learning techniques to analyse the data to find new patterns, create statistical prediction models, post-process the raw data from the models then use visualisations to display the data in a meaningful and comprehensive way as to be understood by humans. Data mining can produce models with powerful predictive abilities which can solve business problems and predict trends.

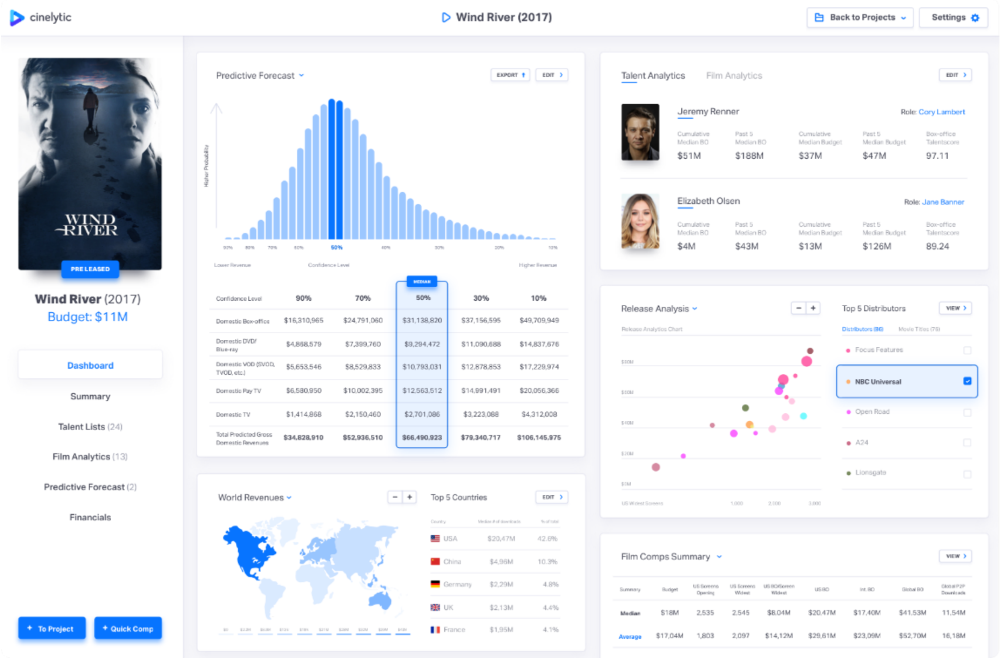
2.2. Alternative Existing Solutions to Your Problem

The use of artificial intelligence in Hollywood's film industry is become more prevalent in the last decade. Making movies can be very expensive, so the production companies want to know if their idea for a movie will be worthwhile making. There are several companies that have started up in the last decade to provide a software solution for this. Companies like Cinelytic, Vault and ScriptBook use data mining and artificial intelligence to make estimate predictions on what will be the movie's box office based on budget, planned actors for the cast and other relevant factors. [4]

Cinelytic

Cinelytic is a software platform which uses artificial intelligence to provide analytic insight for movie studio and independent film makers on possible success and profit for the movie they plan to produce. Cinelytic uses data that is both proprietary and licensed by third parties to create prediction models that allow the users to evaluate the project value and minimised risks. [5]

Cinelytic makes movie success predictions from inputs such as the movie's budget and the main actors cast in the movie. It provides comprehensive visualisation of data and graphs of the results, each after easily understood and business decisions can be made quickly. Cinelytic has been proven to be useful to film makers for "unparalleled accuracy". Cinelytic software doesn't touch on the creative parts of the movie projects. CEO of Cinelytic, Tobias Queisser, said "we want to bring the 'gut' part of the decision down to 60%. The creative part should probably still override, but in order to create a better product, execute and market it better, and find a more financially satisfying outcome, it helps to use a more methodical approach to project evaluation and risk assessment" [6].



Confidence Level	90%	70%	50%	30%	10%
Domestic Box-office	$16,310,965	$24,791,060	$31,138,820	$37,156,595	$49,709,949
Domestic DVD/ Blue-ray	$4,868,579	$7,399,760	$9,294,472	$11,090,688	$14,837,676
Domestic VOD (SVOD, TVOD, etc.)	$5,653,546	$8,529,833	$10,793,031	$12,878,853	$17,229,974
Domestic Pay TV	$6,580,950	$10,002,395	$12,563,512	$14,991,491	$20,056,366
Domestic TV	$1,414,868	$2,150,460	$2,701,086	$3,223,088	$4,312,008
Total Predicted Gross Domestic Revenues	$34,828,910	$52,936,510	$66,490,923	$79,340,717	$106,145,975

Jeremy Renner — Role: Cory Lambert

Cumulative Median BO	Past 5 Median BO	Cumulative Median Budget	Past 5 Median Budget	Box-office Talentscore
$51M	$188M	$37M	$47M	97.11

Elizabeth Olsen — Role: Jane Banner

Cumulative Median BO	Past 5 Median BO	Cumulative Median Budget	Past 5 Median Budget	Box-office Talentscore
$4M	$43M	$13M	$126M	89.24

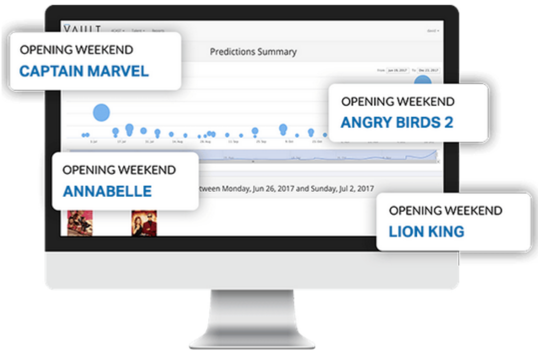
- Focus Features ☐
- NBC Universal ☑
- Open Road ☐
- A24 ☐
- Lionsgate ☐

Country	Median # of downloads	% of total
USA	$20,47M	42.6%
China	$4,96M	10.3%
Germany	$2,29M	4.8%
UK	$2,13M	4.4%
France	$1,95M	4.1%

Summary	Budget	US Screens Opening	US Screens Widest	US BO/Screen Widest	US BO	Int. BO	Global BO	Global P2P Downloads
Median	$18M	2,535	2,545	$8,04M	$20,47M	$17,40M	$41,53M	11,54M
Average	$17,04M	1,803	2,097	$14,12M	$29,61M	$23,09M	$52,70M	16,18M

Vault AI

Vault AI is an Israeli company that uses artificial intelligence that analyses the actual content of movies. Vault "reads" the screenplay and analyses the script, things that happen in the scene, character development and even plot structure. Then it is able to make a prediction of the box office income of the movie [7]. Vault AI have several products that offer different type of analysis of movies. One of them is "RealAudience" which uses artificial intelligence to predict who will be the main global demographic who will watch the movie, and how much tractions the movie will get from these audiences, even suggest improvements for the movie to attract even more people of a specific demographic. [8]

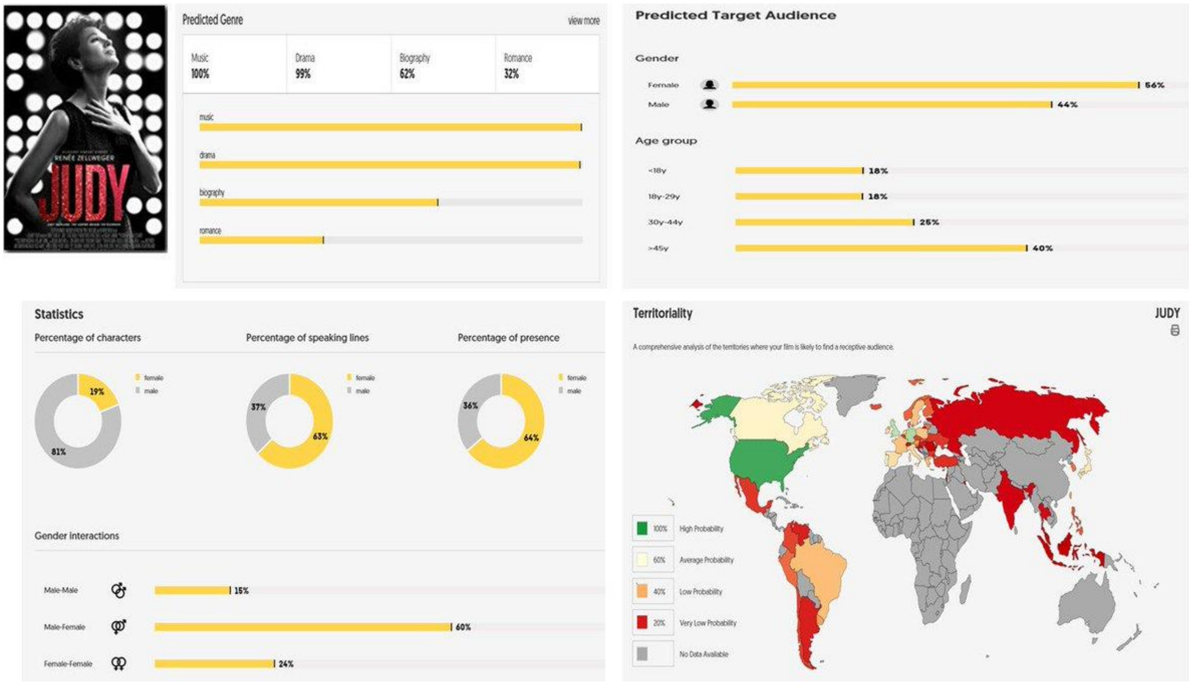


Gain Unprecedented Market Visibility

Vault's RealDemand™ Marketing Intelligence tracks up to 70 theatrical and streaming titles over a 52-week window so you can predict consumer behavior and plan, pivot and evolve your marketing campaigns in real time.

ScriptBook

ScriptBook is a company that uses artificial intelligence with natural language processing to analyse the script and story of the movie. It analyses the characters and story of the movie script and makes predictions of what the audience ratings will be like, which characters are likeable, the genre of the script and even to which audience does the script appeals to most. This doesn't make box office predictions like the previous software do but provides deeper insights into the story itself [9]. ScriptBook have another very interesting product based on artificial intelligence called DeepStory. Having analysed thousands of movie scripts, DeepStory can generate compelling stories and scripts by itself. DeepStory has advanced character awareness, creating characters with their own personalities and traits. [10]



2.3. Technologies you've researched

Machine Learning Technologies

There are several machine learning technologies that implement artificial neural networks. It's is possible to implement an artificial neural network with most object-oriented programming languages such Java and C++. Instead of trying to build an artificial neural network from scratch, it was a better to option to use programming language libraries that allow easy set up and utilisation of artificial neural networks. Python is very popular today due to its simplicity, flexibility and availability of many libraries to be used. This project will use python with third-party python libraries to create, train and use artificial neural networks.

TensorFlow is a python library that provides all machine learning capabilities. TensorFlow is a free and open source library. TensorFlow can be used to implement any type of machine learning technique such as deep learning. Due to this, TensorFlow has a very flexible architecture making this library applicable to most artificial intelligence projects. TensorFlow was chosen for this project due to its wide application and its popularity online means there is a lot of support and documentation for this library.[11]

Keras is another open source python library that implements artificial neural networks. The main advantage of Keras is the ease of use. Keras has a very simple and straight API functions to allow for fast creation of neural networks and testing. Even though Keras is a standalone package, TensorFlow uses Keras as a submodule in its architecture. Keras is the front end of the TensorFlow library. This combination provides a powerful library, with user-friendly API functions to create and train neural networks with the impressive capabilities of the TensorFlow backend giving the library wide application and flexibility. This high level of abstraction allows the focus to remain on the high-level design and implementation of the neural network without having to worry about the low-level technicalities of neural networks. [12]

I have chosen TensorFlow and Keras as technologies to be used in creating and training movie success prediction models.

Web Application Technologies

My goal is creating a web application as the front end for my project. Everyone has access to the internet and hosting my prediction model online gives it wide availability and ease of use. I have chosen python as the programming language of choice for this project, due it the simplicity of use and fast development, this project will also use python-based web framework to host my web application.

Django is a python based free open source web framework. The main advantage of using Django is due to its simplicity of use, plug-and-play architecture and flexibility. This allows for faster design and creation of website without having to deal with the technical low-level details associated with web frame servers. It implements website secure as core of its design allow the developer to avoid the most common security flaws in website design. Django is lightweight and highly configurable. It is compatible with most common sever technologies such as Apache and Nginx. It supports a database backend for most common database technologies as well, such MySQL and MongoDB. Django is highly extensible and allowed for configuration of third-party application and packages. This project will use Django as the web frame to create the web application due to the many benefits stated above, especially ease of use and rapid development.

As part of the website, I will use JavaScript to create an interactive and dynamic website. JavaScript is most widely used and well documented technology, there is no reason to use anything unorthodox.

2.4. Other Research you’ve done

Domain specific research (how many papers should I review? I just reviewed 1 so far, I will come back and add more research reviews later)

There are several research papers online that have experimented with using data mining to predict movie success. They are very similar in approach to this project. This project uses some of the ideas presented in these research papers.

The most recent paper talking about movie success prediction is by Saurabh Kumar from VIT University. In that project, a dataset was downloaded from Kaggle.com that had 651 rows, meaning 651 movies. Additional data about the movie such as its run time was acquired by web scraping IMDb pages with the relevant movie. Movie’s success was measured by IMDb ratings, number of votes, critic and audience ratings, critic and audience scores for the movies. After cleaning the dataset, the data was used to train a naïve Bayes classifier model with supervised learning method like stochastic gradient decent. Using input variables like budget, number of Oscars won by the cast playing in the movie, and using audience score of the movie, made predictions of the IMDb ratings with almost 80% accuracy. [13]

2.5. Existing Final Year Projects

Secure Document Sharing - Owen Kane

This project creates a secure online system to create, edit and share documents over the internet. It uses client-side AES encryption algorithm to encrypt the files before they are sent over the internet. This way the data will never be sent in plain text format for any man-in-the-middle to see the contents of the data in case where they are sniffing and capturing passing packets online.

This is a good approach to file sharing. This increases the privacy and security of data from being access by unauthorized users. The technologies used are also like what I want use, like Python and JavaScript, in a client-server architecture. Any transition of data between the tiers in the architecture use a secure encrypted transfer protocol, SSL/TLS. SSL is used when data is retrieved from the database to the server, and again when data is sent from server to client and vice versa. This a good approach, with I’ll have do the same in my own project.

The project was very well tested. Used multiple types of tests, such as ad-hoc testing, unit testing and integration testing. Testing is vital to any coding project, but more so to project with computer security as possible bugs in the guys can expose vulnerabilities and opportunities for hackers to steal confidential or sensitive data.

Education Tool for Web-Based Vulnerabilities - Cormac Kelly

Interesting project scans your Java files for possible SQ L Injection vulnerabilities. It is designed as an education tool. I like the way it is a web application, making it accessible and easy by the user. It encourages to design code with security in mind and using this tool as quick test for any obvious security flaws pertaining to SQL Injection. I like the idea behind the project, to raise awareness about computer security and encouraging to write secure code.

The project used many technologies and languages. For the code base, Python, Java and JavaScript were used. These are well suited and straightforward languages to use to make a web application and the server back end. These languages also have graphical user interface libraries to make the program easily accessible.

I like this project due to its emphasis on the user interface. It's perhaps the most important aspect of any software because that's all the user is going to see. It's important that is comprehensive and easy to use. As I will also need a user interface for my web application that doesn't look confusing or bland.

2.6. Conclusions

With the current research available, it provides a good guideline of what is a good approach to the problem of trying to predict movie success, and which variables are highly correlated to the movie ratings and which are correlated to movie box office. The research shows that it is indeed possible to have a good estimate of around 70% accuracy what the movie ratings will be just by taking into account production variables such as budget, runtime, genre and the awards and success of the actors in the movie. Based on the successful use of different machine learning techniques, it seems that a neural network would work well and provide similar level of accuracy as the prediction models that did not use artificial neural networks.

Next Chapter will discuss the design of the project based in the researched technologies discussed in this chapter.

3. Design

3.1 Introduction

Following on from the previous chapter, this chapter will discuss how those technologies will be implemented in this project and what the overall design of the project is going to look like. This chapter will discuss software development methodology which will be used during the development of this project and the architecture design of the web application. This will include diagrams of system design, class diagrams of code, diagrams of web application architecture and use case diagrams. The final section in the chapter will discuss the type of testing that will be used and how this projected will be evaluated.

3.2. Software Methodology

Agile Methodology

Agile methodology is a type of software development. Agile uses an incremental approach in developing software. It is a fast approach to writing software whilst remaining flexible to any changes in the requirements of the software. With Agile, the work is split up into time periods called "Sprints", usually about 2 weeks, where the development and testing of the software is done. At the start of each sprint, the team, called a "scrum team" have a meeting to plan out what tasks and features of the software should be developed over the course of sprint. The team discusses the plans with a product manager who relays any requests from the product customers, takes into account any needed changes in the software requirements and plans out the work to be carried out. The team then works to achieves the set-up goals and complete the software development tasks laid out in the meeting. This makes Agile mythology into a cyclical development method, where the work done is reviewed and taking into consideration when planning the next steps in development.

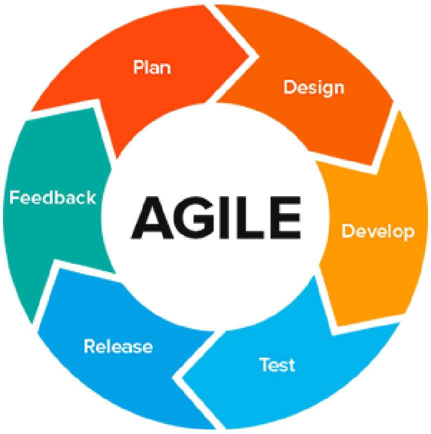


Diagram of Agile development cycle

Feature-driven development

Feature-driven development (FDD) is a type of software development methodology similar to Agile methodology. FDD is also an iterative and incremental approach to software development. FDD main focus is creating feature for the software, focusing one feature at a time before moving on to the next feature. This is where it differs from Agile methodology, Agile do not focus on individual feature as much as they focus on breaking up the project development plan into small to-d0 tasks and plan to implement a certain number of tasks in a Sprint. FDD instead focus on developing fully working individual features in accordance to the principles outlined in the "Agile Manifesto". With this, FDD combined the best industry practices of software development into one cohesive whole. The main advantage of using FDD is that it's simple 5 step process makes it easy to fast develop and deliver tangible results of the software, present working software features to the costumer. Just like agile, FDD also remains very flexible to any change in software requirements and rapidly adapts does changes.

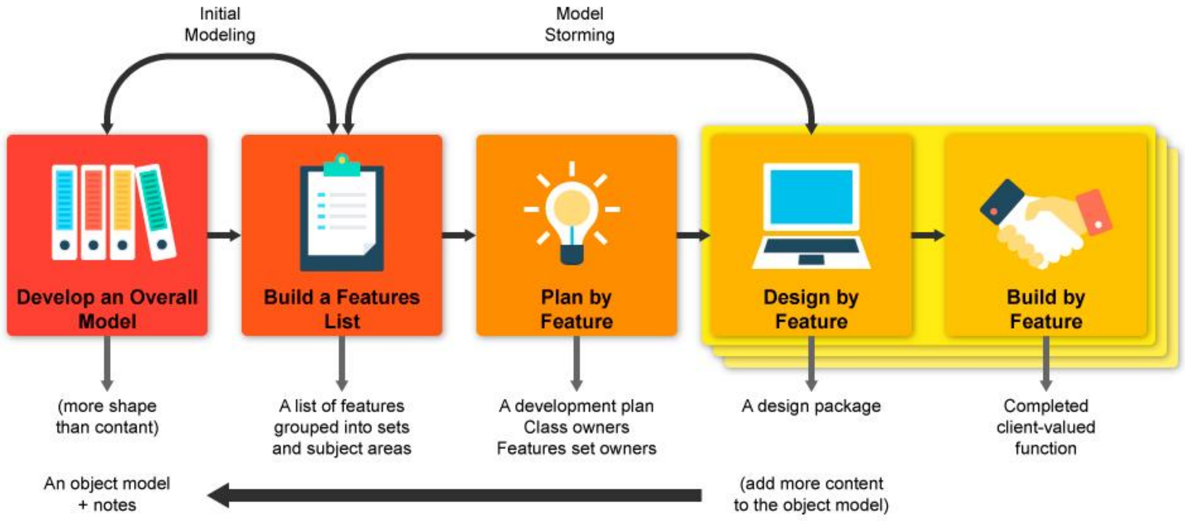
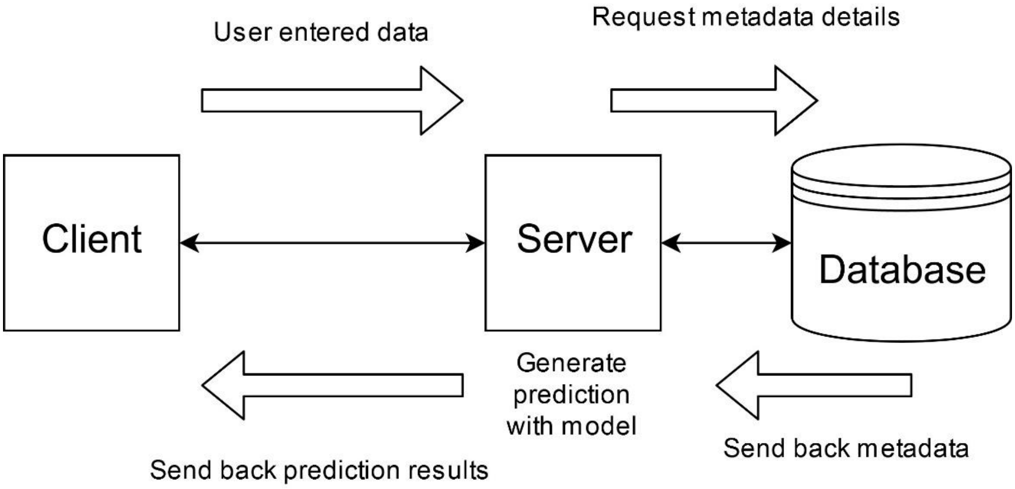

Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than contant)
A list of features grouped into sets and subject areas
A development plan Class owners Features set owners
A design package
Completed client-valued function
An object model + notes
(add more content to the object model)

Diagram of FFD development

3.3. Overview of System

The main system of this project will be in a form of a web application. It will be a 3-teir architecture, client-server with a database in the backend. Python Django will be the web frame used to host and run the web application. The front-end will be a light client web page, made from HTML\CSS and using JavaScript to make it interactive and responsive. The front end will look like a form with fields to fill out with details about the movie they want a prediction on. The data will be sent to the server for processing. The database will store all the metadata, such as details on how many awards an actor has won and how many movies with their ratings has a director directed. The server will get the needed extra metadata and processes into a format ready to be used by the model. The model will analyse the given data and give a prediction of the movie ratings and box office, the server will post process the results from the model into a comprehensible form, make visual representations of the data and send it back to the client to be viewed by the end user.

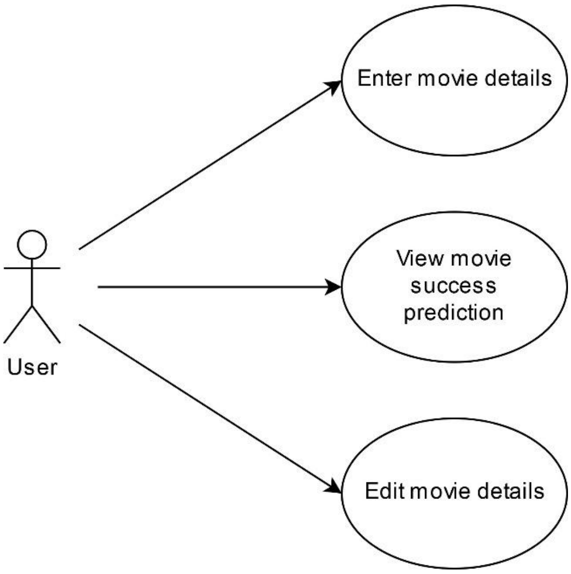


Overview diagram of the web application and data flow

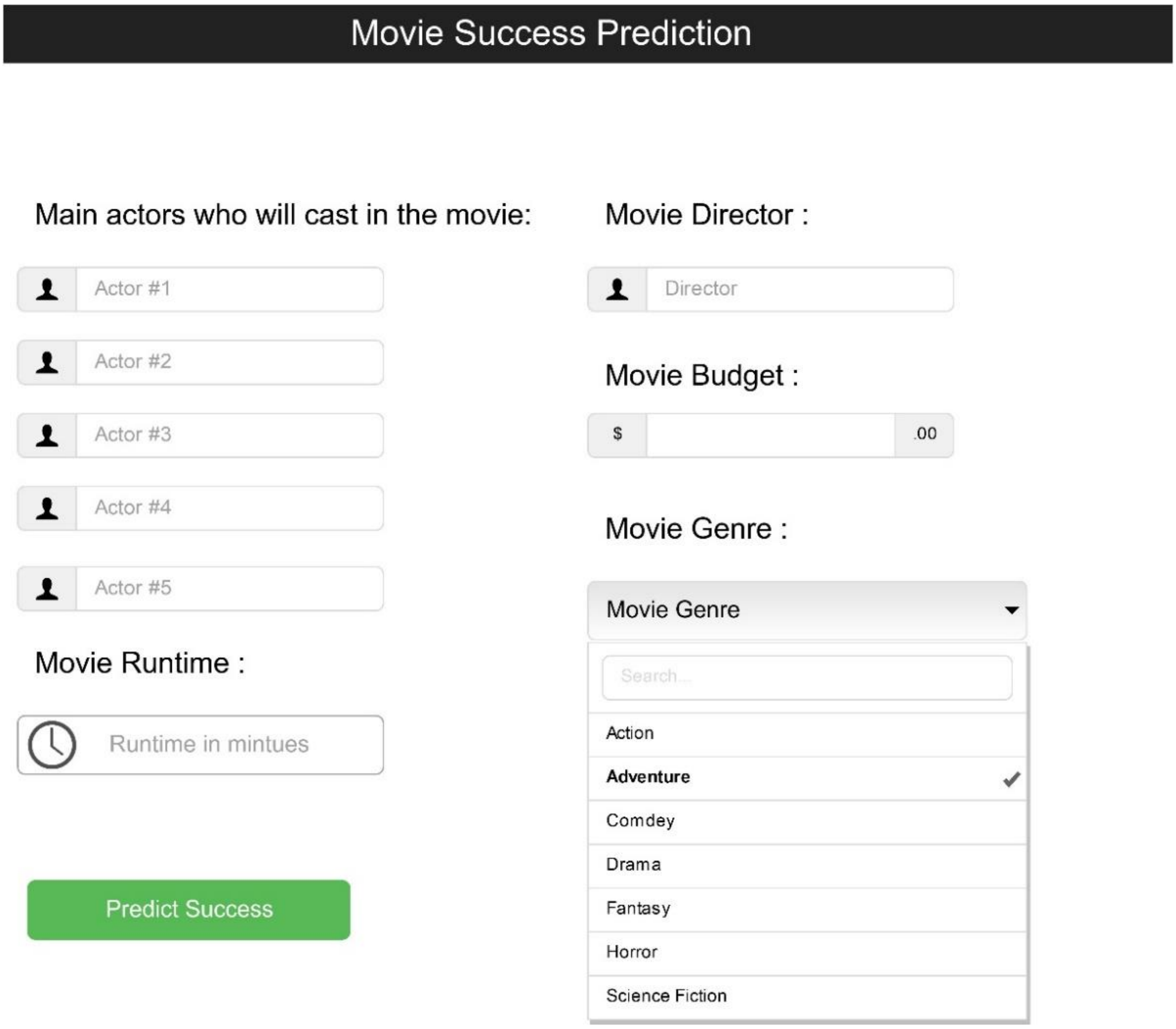
3.4. Front-End

The front-end, the client, will be a simple web page. Written in HTML for basic web elements, and CSS for styling and making the website look good. There will be several web pages, first one to be the form where the users fill out the details about the movie and a results page where the prediction and visual data will be shown to the user. None of the processing of data will be done client side, the web pages will just take user input and send that data to the sever for processing.

The use case for the web app from the user perspective is simple. The user goes on the web page, they will enter the details for the movie and click the button to predict success. The button will take the user to the results page to view the movie predictions. The use than will have the option to edit the movie details, which will bring them back to the original form page, but the old data is still in the fields, so the user only needs to change some fields as they wish instead of filling out the whole again.



User use case diagram



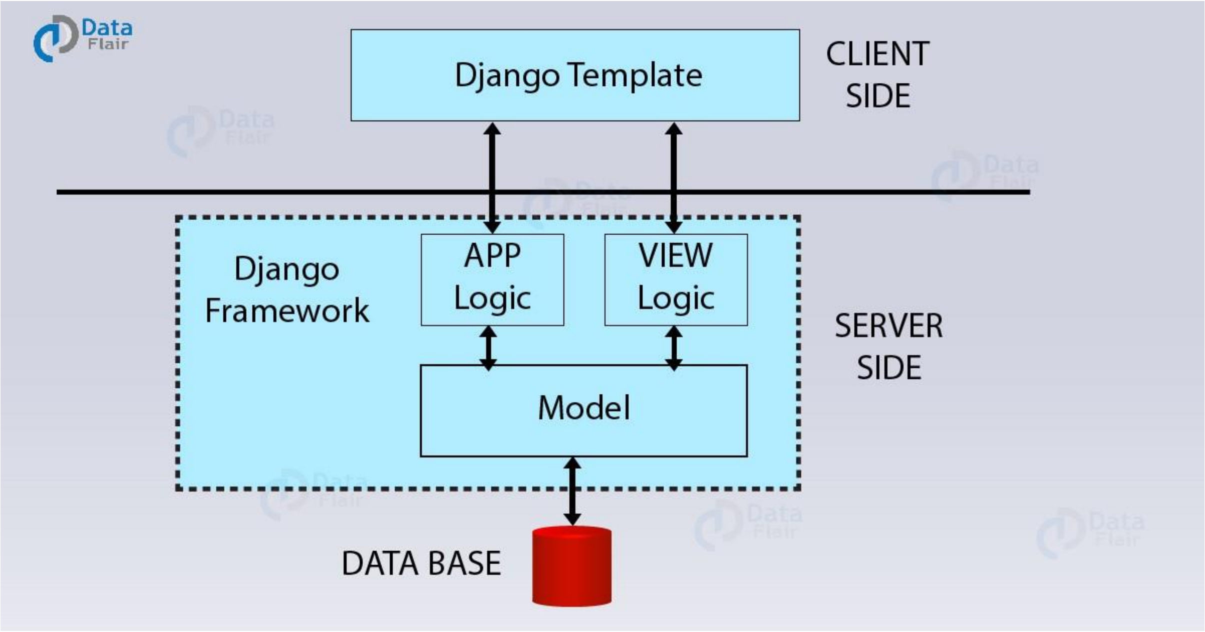
Prototype of from web page

The form will have 5 fields for actors. It will have interactive feature whereas the user types in the names of actors, a small drop down will appear of actors in the database, by searching the database for the actors that match the current user input, like half names and surnames. This is a way to autofill in the field and make the form filling faster, making the website feel responsive and interactive. The same goes for fields for movie director. Have a numerical field for movie budget and movie runtime. A selection for movie genres will have a drop-down list to genres the available movie genres. A button to with "Predict Success" will send all inputted field data to server and then the button will redirect the user to the prediction results page.

3.5. Middle-Tier

Middle-tier will be run on a python Django server. The server will have all the logic behind the web application. The movie success prediction model will be stored on the server. The sever will receive input from the client, movie data to make a prediction on. The server will query the back-end database to get metadata on the data, such as number of awards won by the actors which the user entered. Python scripts on the Django servers will process the data and normalise the data to be used by the prediction model. Run the data through the prediction model to request an output. The project will make several types of predictions, ratings of the movie and box office income. For this there needs to several prediction models, one for each type of prediction. The sever will have to run the processed data through all the prediction models. The results are going to be post processed into a readable comprehensive format. The server will merge all the outputs together into one dataset. Using the dataset with results, the data will be sent back to the client to be displayed. Generate graphs and visual representation of the prediction results.

To keep the data on the database up to date, and to keep the prediction models accurate, once a week, the server will do web scraping to check if any of the data on the database is different on the source websites, like IMDb and Rotten Tomatoes. The database will be updated with the new data and then using the database, re-train the artificial neural networks to create new and up to date prediction models, ready to be used by end users.

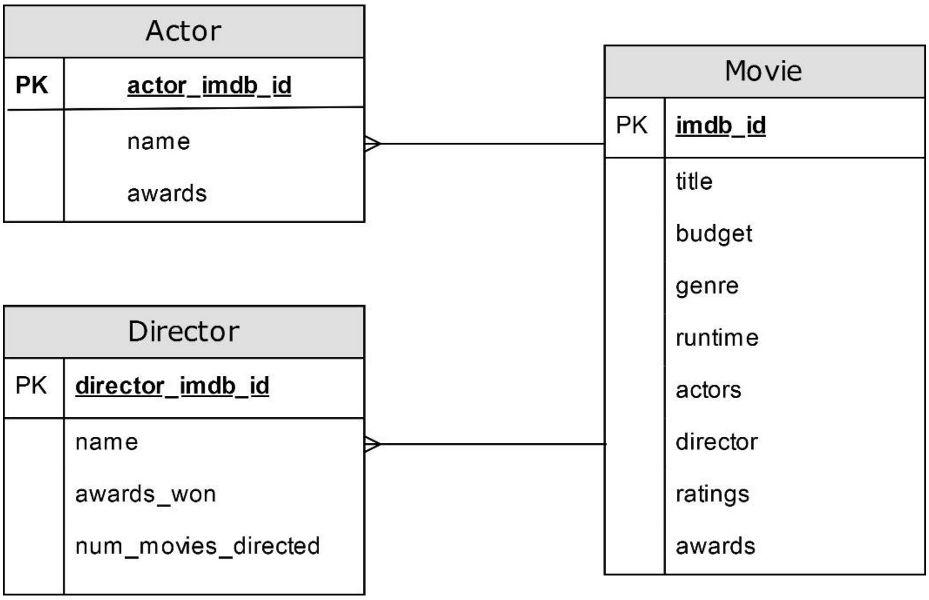


This the model of how the Django server will look in relation to the other tiers

3.6. Back-End

For the back end, there will be a MySQL server holding the all data about the movies, actors and directors. There will be 3 tables in the database, one for movies, actors and directors. The main advantage of using a database is faster performance on processing data. If there was no database, the server would have to web scrape movie websites to get detail metadata on the actors and movies. This can take time, depends on the internet connection and depends on the movie websites to running. If there a lot of users using my web application, this will slow down the application significantly. Using a database solves those issues.

The database will be made with MySQL. It is free and open-source relational database. It has also has a driver, like a library or API, for python making it easily compatible with Django and the all the python scripts running on the server.



ERD of the movie database

3.7. Conclusions

This chapter discussed the overall design of the web application and the technologies used. Discussed the software development methodologies, with their advantages. Showed the dataflow between architecture tiers, from the front end all the way to the back end.

The next chapter will discuss the progress made so far on the project, what has been implemented and what challenges have been encountered so far, and possible solutions to them.

4. Prototype Development

4.1. Introduction

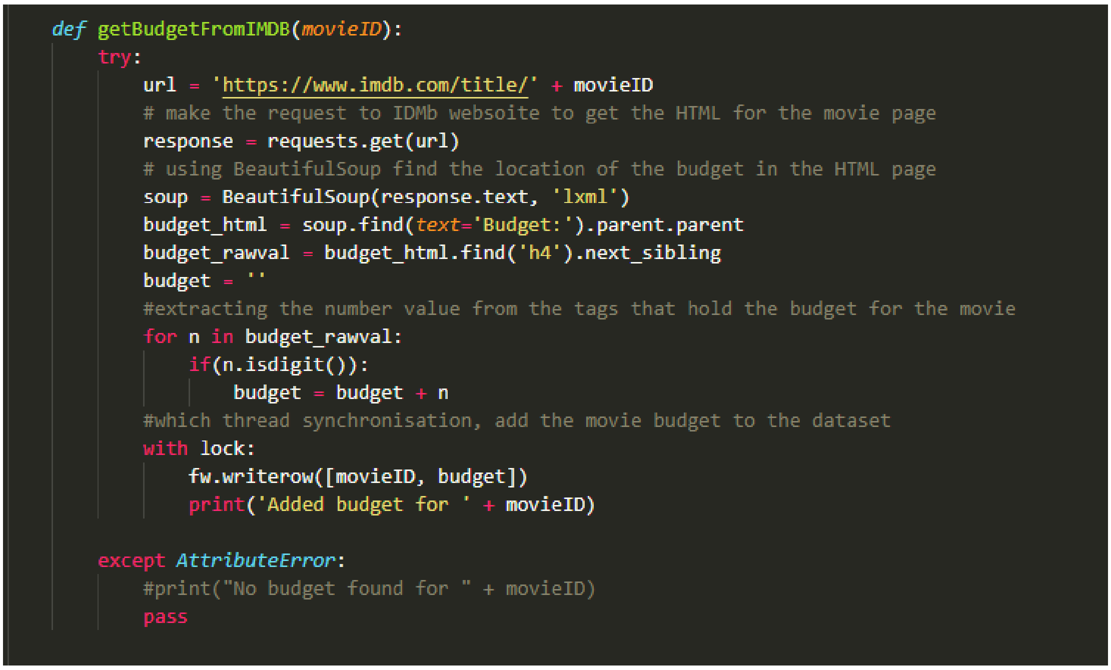
Last chapter we discussed the overall planned design for the system and the web application. So far in the project, the web framework along with the tangle feature like the front-end web pages have not been implemented. Those features are important, but the needed dataset and the artificial neural network which will make movie success prediction are the core of this project, which was the current sole focus of development so far. This chapter will discuss the progress that has been made so far. Here we will discuss how movie data was found online and complied to be more complete. We will also look how the artificial neural network is being implemented and trained with movie dataset.

4.2. Prototype Development

The first part in developing this project was setting up a version control system. This project uses Git with the online GitHub integration. Git was the version control of choice as it's the most popular version control software that's effective and east to use. GitHub allows the project to be stored in online allowing synchronisation of my project across multiple deceives.

4.3. Gathering Data

In order to train an artificial neural network a large amount of data is required. Kaggle.com is an online website that publishes datasets, free to download and use. I found a dataset with 350'000 movies with details like budget, genre and runtime [20]. That dataset also had tables with main casting in the movie. IMDb website also have a dataset available to download [21]. This dataset contains all the movies and tv series on the IMDb website, the movie ratings and titles with their IMDb IDs. The Kaggle dataset was incomplete, there were movies that were missing the budget for example. To get this additional information, I web scraped IMDb website to find movie budget. Using python libraries "Requests", the web scraping script made HTTP requests to the IMDb websites, using IMDb movie IDs as part of the request URL. Using python library "BeautifulSoup", the received HTML DOC was parsed to the budget listed for that movie on the web page, extracted the number value and added the budget to the in the movie dataset row where the movie budget was missing.

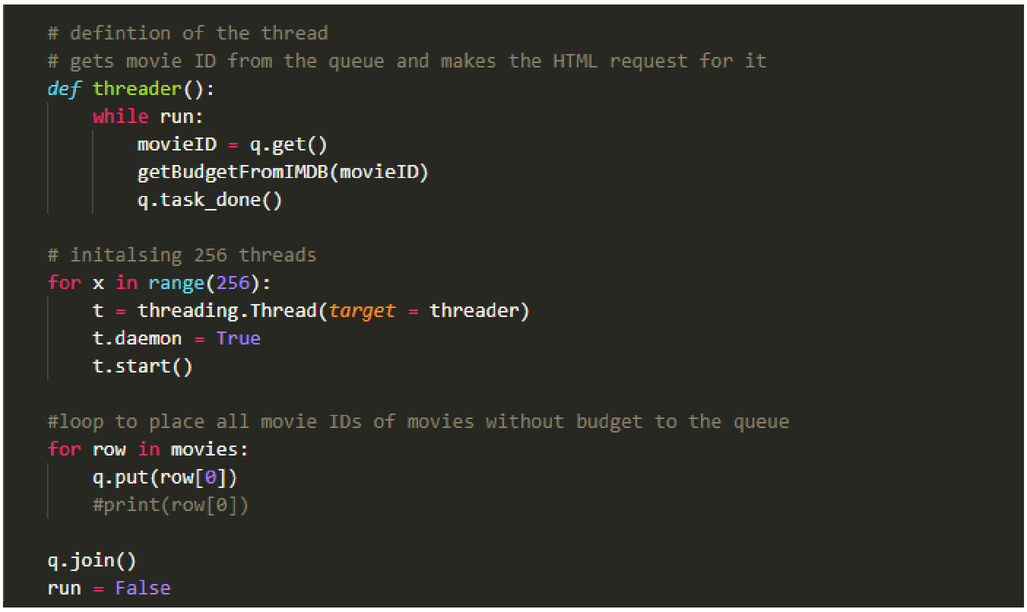


```python
def getBudgetFromIMDB(movieID):
    try:
        url = 'https://www.imdb.com/title/' + movieID
        # make the request to IDMb webseite to get the HTML for the movie page
        response = requests.get(url)
        # using BeautifulSoup find the location of the budget in the HTML page
        soup = BeautifulSoup(response.text, 'lxml')
        budget_html = soup.find(text='Budget:').parent.parent
        budget_rawval = budget_html.find('h4').next_sibling
        budget = ''
        #extracting the number value from the tags that hold the budget for the movie
        for n in budget_rawval:
            if(n.isdigit()):
                budget = budget + n
        #which thread synchronisation, add the movie budget to the dataset
        with lock:
            fw.writerow([movieID, budget])
            print('Added budget for ' + movieID)

    except AttributeError:
        #print("No budget found for " + movieID)
        pass
```

Python code used in the requesting the movie web page and parsing for the budget

From the multiple dataset and filled in I wrote a python script to parse through the datasets and merge the data into a single CSV file. This file will then be read by python script to train the neural network

One challenge in the web scraping was the speed of getting and processing the HTML docs. There were about 175’000 movies which didn’t have budget metadata. Making a request one at a time to check for budget on all the movies would have taken a few days to complete. To speed up the process, I used 256 threads to make 256 requests at a time and parse the received HTML doc. This greatly improved web scraping performance, instead of taking few days to complete, it took a few hours instead.

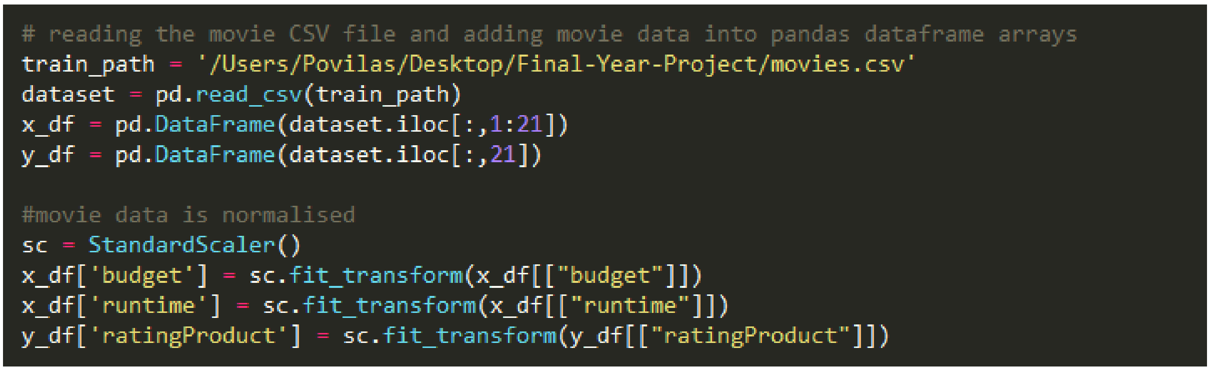


```python
# defintion of the thread
# gets movie ID from the queue and makes the HTML request for it
def threader():
    while run:
        movieID = q.get()
        getBudgetFromIMDB(movieID)
        q.task_done()

# initalsing 256 threads
for x in range(256):
    t = threading.Thread(target = threader)
    t.daemon = True
    t.start()

#loop to place all movie IDs of movies without budget to the queue
for row in movies:
    q.put(row[0])
    #print(row[0])

q.join()
run = False
```

Python code using threads to make HTML requests

4.4. Artificial Neural Network

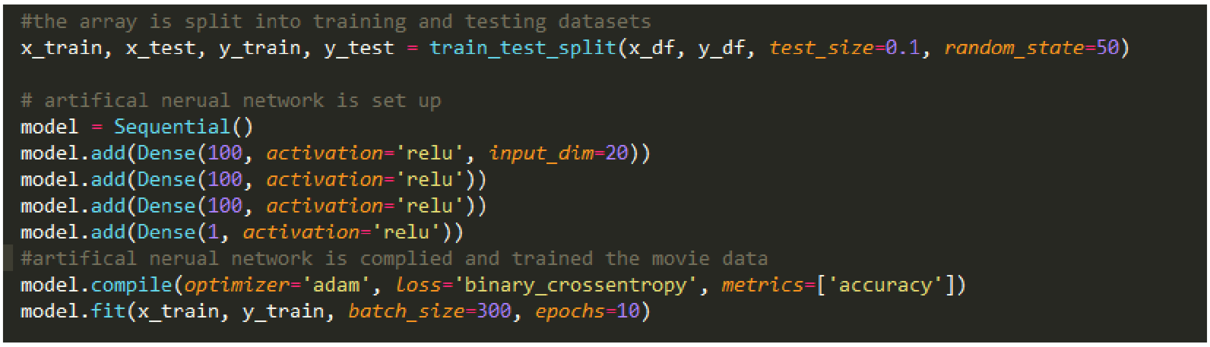
After the necessary data has been acquired, from datasets and web scraping. It is time to train the artificial neural network. First the data is imported from the final csv files that has all the necessary movies with complete metadata. Using python library Pandas, the data is split into 2 “DataFrames”, which are like arrays. First array holds the movie production variables such as budget, genre and run time. The second array holds the ratings for the movie. The first array will be used as inputs for the neural network and the second array is the expected output, so the network can calculate its loss and configure it’s weights. Before the data is inputted into the neural network, the input data needs to be normalised, the genre is represented as binary input into the networking while the budget are disproportionally very large values. Normalising the budget puts it at same level as all other variables making it efficient to be used by the neural network.



```
# reading the movie CSV file and adding movie data into pandas dataframe arrays
train_path = '/Users/Povilas/Desktop/Final-Year-Project/movies.csv'
dataset = pd.read_csv(train_path)
x_df = pd.DataFrame(dataset.iloc[:,1:21])
y_df = pd.DataFrame(dataset.iloc[:,21])

#movie data is normalised
sc = StandardScaler()
x_df['budget'] = sc.fit_transform(x_df[["budget"]])
x_df['runtime'] = sc.fit_transform(x_df[["runtime"]])
y_df['ratingProduct'] = sc.fit_transform(y_df[["ratingProduct"]])
```

Python code used to load the data from local CSV file into DataFrames and the large data being normalised

The neural network is created with the TensorFlow and Keras python libraries. A densely connected sequential neural network is made with 2 hidden layers and 100 neurons in each layer. The nuerual network is compiled and the processed datasets from above are fitted and the neural network is trained to make a movie prediction model to predict movie ratings.



```
#the array is split into training and testing datasets
x_train, x_test, y_train, y_test = train_test_split(x_df, y_df, test_size=0.1, random_state=50)

# artifical nerual network is set up
model = Sequential()
model.add(Dense(100, activation='relu', input_dim=20))
model.add(Dense(100, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(1, activation='relu'))
#artifical nerual network is complied and trained the movie data
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, batch_size=300, epochs=10)
```

Python code used to create, compile and train the neural network using TensorFlow and Keras libraries.

4.5. Conclusions

This chapter had shown the current progress made so far with example of code. Explains how the data is being acquired and cleaned in preparation for training the network. Here it was show how the neural network is being implemented and how the data is pre-processed right before the network is trained. The next chapter will discuss how this system will be tested and how the web application will be evaluated.

5. Testing and Evaluation

5.1. Introduction

Last chapter we discussed the project development progress so far. This chapter will explore how the code will be tested to ensure proper functionally and how the projects web application will be evaluated.

5.2. Plan for Testing

The project application will be tested all the time. Firstly, manually testing by running the code and checking it works as expected. The application will be run on a server, which can be hosted in the local machine. This allows extensive test done immediately with every implementation of a feature or change in code. It is possible to set up automated tested using python libraries like Selenium that automate browser activity.

Whitebox testing will be used to test the source code running the server and data processing code. Unit test will be used to ensure the code internally functions the way it's meant to. Creating test cases will allow for automated testing to be executed.

Web scraping will be tested using Blackbox testing. Testing python web scraping scripts is important as to ensure the HTML pages being returned from online are the right pages containing the right movie data. To test this, a small dataset will be manually complied with reference to the online sites. This is a small dataset with known and expected outputs. Then running the web scraping scripts and comparing the results from the script is the dataset of expected results will ensure web scraping is working as intended and returning accurate data.

The neural network is always tested after training. The dataset is split up into two, one for training the model with, the other for testing. After the model is trained, the test data is run through the model to see if the predictions made are accurate, and to what degree. Instead of configuring the network, it just runs all the movies not used in the training through the model to test the average accuracy in prediction of movie ratings.

This is a form of inbuilt Blackbox testing. Since the model is always tested immediately, this allows for quick development and configuring the neural network to make increase its accuracy.

5.3. Plan for Evaluation

To evaluate this project, in particular the front-end of the application with which the user will interface, is vital. It is important the front-end is easy to use and initiate to as many average users as possible. To evaluate, two groups of volunteers will be needed, second group of tech savvy users, such as other computer science students, second group of regular users.

The aim is to have a decent sample size of at least more than ten people in each group to have a somewhat representable sample size. To evaluate the front-end, everyone will go onto the web application from various devices, like laptops, phone and computers. The tech savvy group should have plenty of experience with technology. Their evaluation of the web app and perhaps even system as whole would be very valuable. The evaluation of the average user will be most important. They reflect how most users would see and interact with the web application. It's important to get their feedback on to their experience in using the app.

Given any suggestions on improvement for the web application, those will be implemented as soon as possible. After the changes have been made, another evaluation from both groups will be needed, to check if there any more improvements that could be done.

This will be based on Nilsen's heuristic evaluation model. This has a list of criteria as a guideline of what makes a computer software to use. Using this approach, the users testing the software will give verbal ratings out of 10 about each section of Nilsen's heuristic. Finding the average rating for each section will provide insight into what needs to be improved upon the most.

This approach reflects the agile software development methodology in that the software developers meet the product managers, or the customers themselves, to show the current progress made and to test or evaluate their progress. The feedback from the meeting is considered when planning for the future sprint.

5.4. Conclusions

This chapter discussed what type of testing will be used and how will the system, the web application, will be evaluated. White box, unit testing, automated testing and black box testing will be used to ensure the system works as expected and securely. For evaluation of the web applications, 2 groups of people will be selected and asked to use the web app, giving verbal ratings on each section according to Nilsen's heuristic. This will provide guidelines on improving the software to make it more user friendly.

6. Issues and Future Work

6.1. Introduction

This chapter will discuss the issues and risks that are currently being faced with the project and possible risks in the future. Along with this, this chapter will discuss the planned solutions to minimize risk and ways to solve current issues being faced now.

6.2. Issues and Risks

Current challenges and issues being faced in the project so far:

- Lack of knowledge in building artificial neural networks and using TensorFlow and Keras APIs and their functions
- Lack of complete datasets for movies which have over 20,000 rows with all the necessary data about the movie such as, budget, genre, actors in the cast and the awards they won
- To complete the dataset, get all the missing column information, the data must get web scraped. The amount of data that needs to get web scraped can very large, which is slow when trying to make a few requests every second. Sometimes can take days to finish 1 web scraping operation
- Currently the neural network isn't making accurate success prediction on the movie

The plans of how to approach and overcome these challenges and issues, respectively:

- Reading books, watching videos and taking part in online course about artificial neural networks will provide the needed knowledge and familiar with this topic to produce a working neural network. The same goes for the python libraries being used in this project, which requires more practice and reading documentation to understand the functions being used.
- Even though no single complete dataset for movies which contains all the information needed to train the neural network exists online to be freely downloaded, there still several datasets available. These have varying amount of information and merging these datasets together will create a more complete dataset to start off with. Any additional data needed will get web scraped from online sources.
- To solve the performance issue with web scraping, there are few approaches. The first approach includes using multiple threads to make the HTTP requests to the website servers instead of one request at a time. The second approach includes using a faster computer machine with a faster interconnect with large bandwidth. Even possible to use multiple computers and split up the list of which sites need to be searched between them.
- This will be solved in time through increasing skill and understanding of how neural networks work. Creating an accurate network is an art sometimes, it will some trial and error before optimally configuring the network to be as accurate as possible.

The risks associated with this project:

- The neural network might not give accurate prediction on ratings and box office income.
- The website layouts may change which will mess up the automated web scraping scripts and return the wrong results, even overwrite the current data with blank data.
- Not being able to keep up with the schedule and falling behind on work ultimately not completing tasks on time.
- Unforeseen computer failure during demo presentation, like a Windows update.

Planned approach to mitigate these risks, respectively:

- Doing research and looking online for help. Perhaps trying a new approach and instead of using a neural network for prediction, can try with alterative models like Support-vector Machines.
- This risk can be mitigated by implementing automated tests for web scraping. Before any writing to the database is done, using black box testing will ensure the data that is being return is accurate and in the right format.
- Breakdown the work into smaller tasks to ensure they are allocated the appropriate amount of time and easier to achieve. Put in effort to complete work in advance in order to have extra time for unseen circumstances like unexpected bugs that take long time to solve or any real-life events.
- Scan for and run any needed updated on the laptop which will be used in the demo. In any case, have the demo pre-recorded and put on YouTube if all else fails.

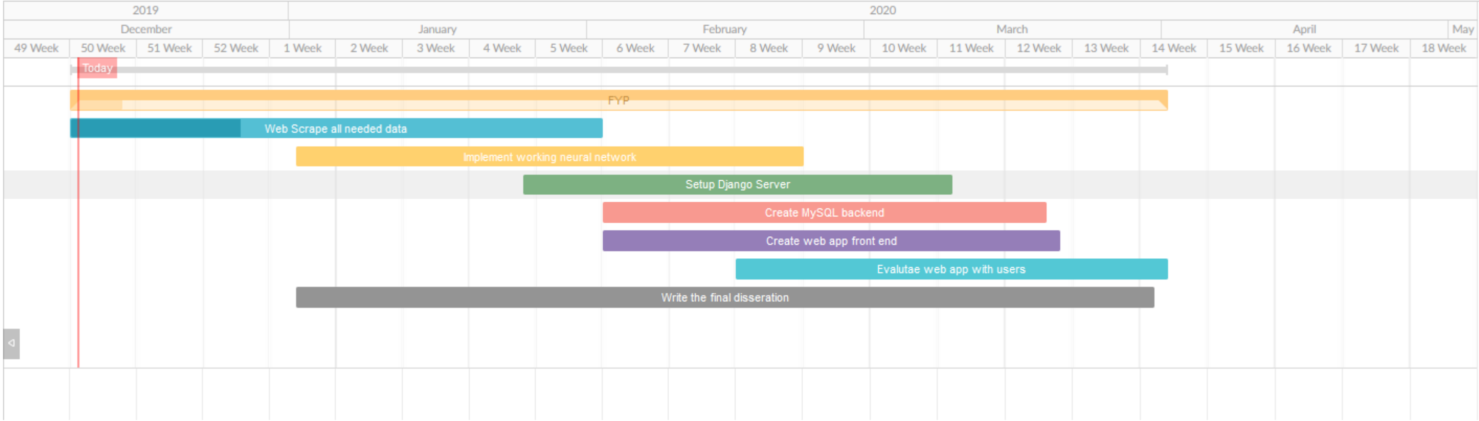
6.3. Plans and Future Work

There is yet a lot to be implemented for this project over the next few months. Most importantly, the entire web frame needs to get implement. Over the next few months, I will focus on completing the dataset which will be used to train the neural network. Then finishing designing and configuring the artificial neural network to make accurate prediction. This the primary core of the project, after both of these have been finished then the work on the frame around the model will begin. The Django web frame along with the MySQL server and client-side web pages will be implemented in the months leading up to the deadline.

Over the course of the next coming months, along with the implementation of the features, the final dissertation will be written up incrementally. After finishing the implementation of a feature, the details of it will be written in the dissertation right after. This way, as soon as the project gets completed, so will the writing of the final dissertation will be completed and ready for finally polishing and submission.

I have included a GANNT chart at the end of this chapter to visually layout of the planned schedule of work on this project.

6.3.1. GANTT Chart



Bibliography

[1] Storytellingday.net. (2019). History Of Storytelling – How Did Storytelling Begin?. [online] Available at: http://www.storytellingday.net/history-of-storytelling-how-did-storytelling.html.

[2] Miller, L. (2019). The oldest story ever written. [online] Salon. Available at: https://www.salon.com/2007/04/24/gilgamesh/.

[3] Mark, J. (2019). The Pyramid Texts: Guide to the Afterlife. [online] Ancient History Encyclopaedia. Available at: https://www.ancient.eu/article/148/the-pyramid-texts-guide-to-the-afterlife/.

[4] Vincent, J. (2019). Hollywood is quietly using AI to help decide which movies to make. [online] The Verge. Available at: https://www.theverge.com/2019/5/28/18637135/hollywood-ai-film-decision-script-analysis-data-machine-learning.

[5] Cinelytic.com. (2019). Cinelytic | Built for a Better Film Business. [online] Available at: https://www.cinelytic.com.

[6] Kay, J. (2019). How data company Cinelytic aims to reduce risk in the film business. [online] Screen. Available at: https://www.screendaily.com/features/how-data-company-cinelytic-aims-to-reduce-risk-in-the-film-business/5136245.article.

[7] En.wikipedia.org. (2019). VaultML. [online] Available at: https://en.wikipedia.org/wiki/VaultML.

[8] Vault-ai.com. (2019). RealDemand™ Market Intelligence. [online] Available at: https://www.vault-ai.com/RealDemand-market-intelligence.html.

[9] ScriptBook. (2019). ScriptBook. [online] Available at: https://www.scriptbook.io.

[10] ScriptBook. (2019). ScriptBook. [online] Available at: https://www.scriptbook.io/#!/deepstory.

[11] TensorFlow. (2019). TensorFlow. [online] Available at: https://www.tensorflow.org/.

[12] Keras.io. (2019). Home - Keras Documentation. [online] Available at: https://keras.io/.

[13] Kumar, Saurabh. (2019). Movie Success Prediction using Data Mining For Data Mining and Business Intelligence (ITA5007) of Master of Computer Application School Of Information Technology and Engineering. [online] Available at: https://www.researchgate.net/publication/332396741_Movie_Success_Prediction_using_Data_Mining_For_Data_Mining_and_Business_IntelligenceITA5007_of_Master_of_Computer_Application_School_Of_Information_Technology_and_Engineering.

[20] Stephanerappeneau. (2017). 350 000+ movies from themoviedb.org. [online] Available at: https://www.kaggle.com/stephanerappeneau/350-000-movies-from-themoviedborg .

[21] IMDb. (2019). IMDb Datasets. [online] Available at: https://www.imdb.com/interfaces/.