Hollywood Movie Analysis

Ezigbo Chidozie
2138172

Submitted to Swansea University in fulfilment of the requirements for the Degree of Masters of Science

Department of Computer Science
Swansea University

15 December, 2022

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed Chidozie Chinedu Ezigbo

(candidate) Date 15

December 2022

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed Chidozie Chinedu Ezigbo

(candidate) Date 15

December 2022

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed Chidozie Chinedu Ezigbo

(candidate) Date 15

December 2022

Table of Contents

Acknowledgement

I would like to thank my family, friends and my supervisor and mentor, Dr Daniel Archambault.

Abstract

Since making Hollywood movies is a lucrative business, producers work very hard to make hit movies. However, some movies underperform after being produced, wasting time, money, and resources. This dissertation illustrates the exploration of movie data to outline aspects that affect movie success and demonstrates how these characteristics can all be used together to determine movie success rates. In the bid to achieve this, the thesis explores similar techniques that can adequately achieve this task and proposes a preferred technique based on literature reviews. The proposed visualisation technique was implemented in a python programming language using the Altair library. The resulting end visualisation software is to be employed in analysing movies allowing the production team to evaluate and understand aspects of movies that make them successful and employ the patterns to make more successful movies.

1. Introduction

Hollywood has been known to engage users for decades with great movies as the most profitable and influential in the global film business[1], even with the great success in movie production, there have also been movies that did not meet up to the expected success[2][3][4], so the question is what makes a movie successful? A study on the success of movies shows that the success of movies can be measured by the gross profit of the movie and the viewer's ratings[5], also factors affecting the success of movies can be divided into two categories, classical and social factors[6]. The classical factors include the runtime of the movie, production team, directors and cast influence, the genre of the movie, the script, and the time of release, while the social factors include the movie rating, reviews, and social trends. Movie analysis has become a difficult task following the increase in movie production, which led to a large movie database[7], presenting a problem of how to effectively express complex information to a larger audience, to solve this problem this thesis investigates the most suited technique to interpret the intended task, it achieves this by carrying out a series of literature reviews which investigates existing techniques that can be employed to achieve this task and then proposes the appropriate technique, providing arguments and reasons for this choice. The paper proposes the use of python programming language with Altair library as the most suited technique and then proceeds using this technique to demonstrate in visual illustration, the effect of these factors on the movie's success, combining these factors to show the success rate of already existing movies from data and then using the combined resulting visualisation to implement a predictive visualisation software for intended movies. This thesis attempts to illustrate the most effective technique for interpreting movie data with the users in mind. The resulting visualisation was intended for movie producers to aid them in making more hit movies in future and thus may not be computer literate enough to read complex visualisation, so a simplified efficient technique would be most appropriate for the users. Proceeding from choosing a technique to achieve the task and understanding these movie success measurements with their affecting factors, data was sourced to support the task from Kaggle's "four decades of movie" by Daniel Grijalva. The reason for choosing this data is that it contains the necessary measurement criteria and affecting factors listed above, whilst also accommodating the predictive nature of the end visualisation software, as it contains information accessible to the production team before the production of the movie.

1.1 Aims

- To implement an interactive visualisation technique for the intended task using python programming language and Altair library.
- To show an interactive analysis of Hollywood movies revealing patterns of achieving a successful movie
- To demonstrate the use of multiple coordinated views technique of visualisation.

2. Literature Review

2.1 Similar work involving large dataset

The first challenge I faced in completing my task was how to give meaning to the large movie data in visual representation. The large amount of dataset that was gathered for the project also presented a challenge in managing this dataset in the visual representation, which is why the research was carried out to determine the best approach (or approaches) to take in managing large datasets. An overview of visual analysis techniques for huge datasets is provided in the study by Danilo Montesi[8]. The paper begins by emphasising how useful visualisation is for exploring and analysing enormous datasets, and it then goes on to give an overview of visual analytic tools for doing so. The following steps for gathering and processing data are also listed in the paper: (1) Acquisition, which he defines as the use of crawling and accessing repositories to gather information and process documents; (2) Semantic enrichment, which gathers information from specific domains' semantics from a single document; (3) semantic integration, which combines information from various decentralised information sources information that is provided by each source and prior semantic enrichment, this process is also referred to as data fusion; (4) Selection and aggregation, which is the process of selecting useful subsets and the aggregation of multiple documents to one single object. Then the paper continues to give examples of some tools that can perform the processes described above: (1) Tag cloud, which uses neural language to extract information and show them in visual representation using the web (2) Information landscapes (for example In-SPIRE and InfoSky) approach can be applied to the thematic analysis of big document sets using the geographical landscape concept. The paper then describes different visualisation techniques for different datasets or scenarios. Firstly, for Multidimensional Metadata, the papers suggest Scatterplots and Parallel Coordinates be suited for representing a large amount of dimension, for Space and time the writer suggests Geospatial visualisation, to see relationships in software like PhraseNet and FacetAtlas were recommended by the author, who also mentioned that Cluster Map can be used to visualise relationships between aggregated structures, also Graph visualisation was mention by the writer as a good technique in visualising data with a large amount of relationship as it reduces clusters. Then for an interactive analysis and multiple visualisation interface, the writer suggested Multiple view coordination (CMV) to be a good approach to use. This multiple-view coordination, according to the author, is a method for integrating various visualisation

components into a single, cohesive user interface so that changes brought on by interactions in one component are instantaneously reflected in the other components. This technique described presented the solution to my initial problem which is how to give a meaningful visualisation to my large dataset, as it deals with the management of a large dataset to give an efficient and interactive visualisation. The multiple view and interactive features will support user enquiries better and which is good for user experience, as it will be understandable and users will have to rely on the eye over memory to read the visualisation, this visualisation technique CMV has captured my interest inspired my research on similar technical approaches.

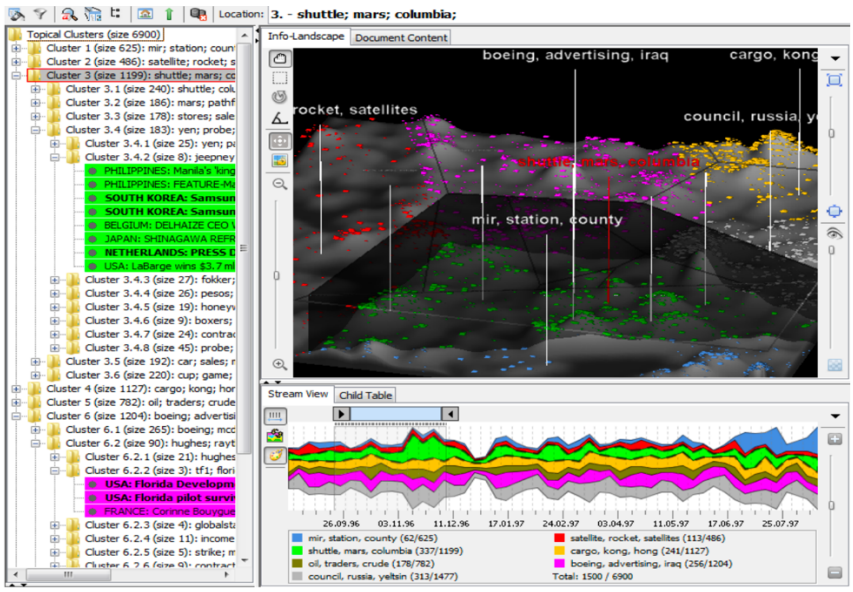

Figure 1 – CMV visualisation technique

In this next paper[9], the author describes a technique for visualising big data sets in aggregated form using symbolic data analysis in this paper, the Visualization of large data sets: the zoom star solution. Intervals are used to represent quantitative data, whereas histograms are used to represent categorical data. Symbolic objects are the name given to these data. The paper describes different methods and tools for visualising these symbolic objects using radial graphs in 2D (zoom star, basic star) or 3D (radial graphs) (3D zoom star, temporal star). Each image depicts only one group in aggregate form, but when they are represented side by side or superimposed, several items can be compared. It's also feasible to visualise the evolution of an object through time, and sound can be utilised to convey features. Zooming, rotating figures, and changing colours and fonts are all interactive elements. Geometric techniques (scatter plots, parallel coordinates, line graphs, polar charts), icon-based

displays(Chernoff faces, stick figures, colour icons), pixel techniques (recursive pattern, spiral technique), Kohonen maps, distortion techniques (perspective wall, Table Lens, Fisheye views), hierarchical (TreeMaps, info cubes, cone trees), hierarchical (TreeMaps, info cubes, cone trees), or graph-based techniques The later techniques are for visualising the data set's structure (hierarchy, graph), whereas the first ones are for visualising the data values. The similarity of this project to mine is that they both deal with large data categorisation and visualisation. However, the visual representation is not suited for my project, because of its complexity, the target audience of my visualisation might not be so computer literate thus understanding and interacting with this visual representation might be difficult for them. The diagram of the papers visualisation could be found in this link: https://www.researchgate.net/profile/Monique-Noirhomme-Fraiture/publication/228615915_Visualization_of_large_data_sets_The_Zoom_Star_solution/links/00b49524be3c16c72a000000/Visualization-of-large-data-sets-The-Zoom-Star-solution.pdf.

This paper looks at how graph-embedding techniques can be utilised to build an interactive movie visualisation and recommendation engine[10]. As the underlying graph structure, it uses a movie actor database and to characterise the movie objects, it uses textual features. The system may get a set of related movies that are depicted and clustered on two dimensions based on a selected pivot movie. The approach is unique in that it captures both neighbourhood and cluster structures. For visualising high-dimensional data, data-embedding techniques have been widely used. The Bourgain embedding, FastMap, BoostMap, and ISOMAP are all examples. Those projection approaches aim to approximate all distances, whereas this approach precisely retains a subset of distances (spanning-tree distances). It starts with the proposed graph embedding to create a movie recommendation interface. This technology enables exploratory visualisation of the movie graph space and includes features such as graph filtering based on multiple criteria, real-time graph exploration, and automatic retrieval of movie trailers from the Internet. This strategy works like this: First, it creates a Minimum Spanning Tree (MST) architecture on the 2D plane that preserves all distances to a user-selected pivot point as well as the MST's neighbourhood distances, for either metric or non-metric distance functions, this architecture carefully examines how to best reflect the original object relationships. Secondly, the dendrogram cluster hierarchy is built in such a way that it can be placed exactly on top of the MST object mapping. To express the multi-granular clusters that are produced, the cluster hierarchy can be frozen at any resolution level

(tomographic view). Objects are accurately mapped on the 2D plane, while the hierarchy of the clustering structure is depicted in the third dimension. They also apply a probabilistic global optimization strategy based on simulated annealing (SA), which intelligently chooses between the two mapping positions to reduce the number of crossed edges. When the search space is discrete, as it is in this case, SA is an excellent optimization strategy. The results of the trials on the movie graph database indicate that the simulated annealing technique is quite effective in improving the layout. This project is related to mine by way of movie visualisation, the embedding technique could be used to distinguish movie sets, but the implementation is challenging and might require a professional. Though it provides an interactive user interface.

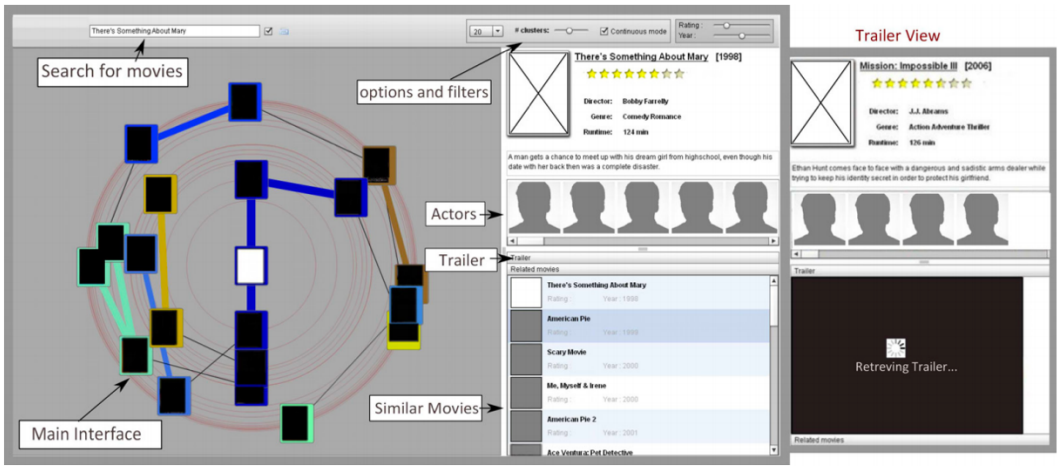

Figure 2 – User Interface for word embedding movie visualisation

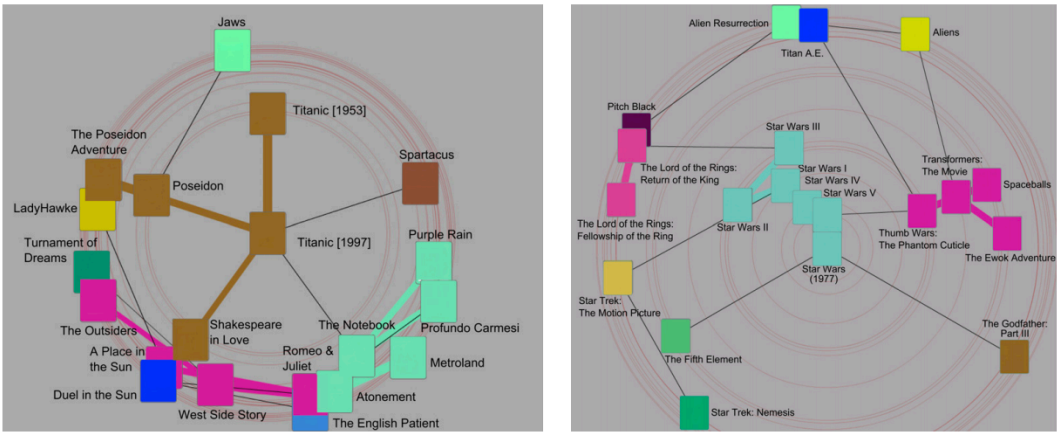

Figure 3–Visualisation output of titanic using word embedding visualisation

2.2 Similar works involving web browser application

This paper [11]describes a technique for visualising movie data by using python's requests, beautifulsoup, jieba and fontools, to collect movie data from 2015 to 2020 then analyses the data and gives a graphical representation of the data using Ajax technology LayUI framework and Echarts. This technique was built to solve the problem of a large movie database, to obtain, analyse and process movie data efficiently, whilst producing effective visualisation for users. This method of visualisation obtains data from two mainstream, the first is by employing the use of APIs, and the second using a web crawler technology, the web crawler can be used to obtain information from web pages and is not restricted by the official API and can access more efficient data and obtain a large amount of data at the same time. Although these two methods of obtaining data show promising results, they also have their disadvantages. Some of them are that APIs have limited permission and the crawler technology must deal with an anti-crawler mechanism. The function of the system is to gain access to movie website data, filter these data and use the data to give a graphical representation through HTML web pages thus allowing users to explore data. This function of the system can further be divided into three stages, firstly data collection, data obtained will contain all information about the movies, the process of obtaining this data has already been explained above, then the second stage is the analysis of the data, which is done in multi-dimension to satisfy the users, here is where the final filtering and processing of data happens, then finally the visual representation. The distinctive function of this technique is that it provides the visualisation to the users promptly whilst providing data storage, the users can obtain visualisation in a matter of 0.5 seconds thus improving user experience. The development of this technique used the LayUI framework for front-end, Echarts for visual representation at the back-end, Ajax for front-end interaction, the back-end uses crawler request and front tools in obtaining data, MySQL for data storage and JIeba library for data analysis. It is developed in the windows operating system, in python through the installation of anaconda and the program written by PyCharm. Below is a diagram showing the user interface of the technique and the visual representation produced.

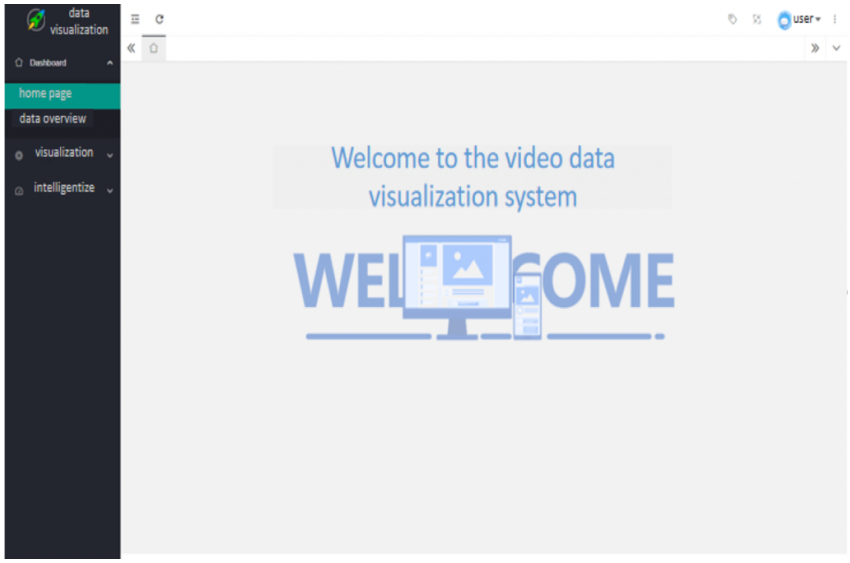

Figure 4 – User Interface Ajax technology LayUI framework and Echarts

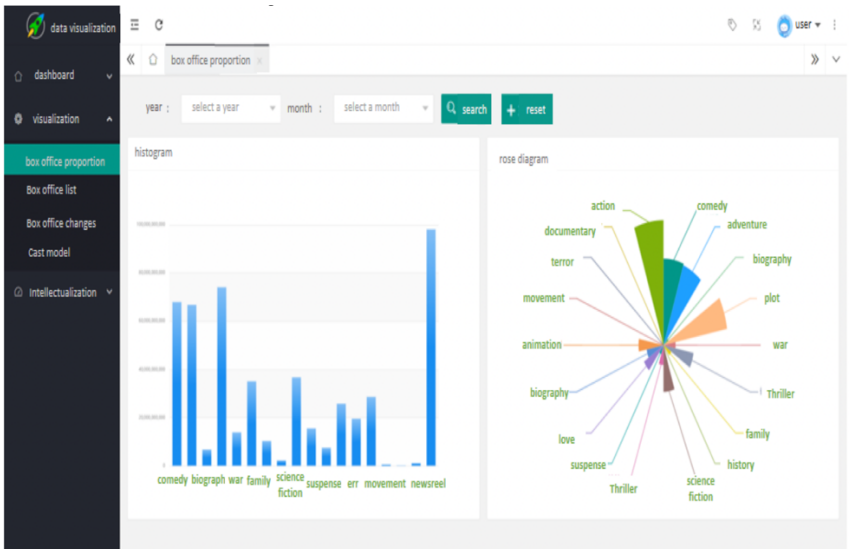

Figure 5 – User Interface Ajax technology LayUI framework and Echarts

Figure 1 above shows the user interface diagram while figure 2 shows the resulting graphical representation of the inquiry for the proportion of movie box office. The paper illustrates similarities in my task as both have to do with movie data and visualisation, although this reviewed paper doesn't show any predictive functions, the data analysis function can be further utilised to accommodate this function. Nevertheless, although the outlook of this paper describes some shortcomings of the use of this technique, such as the failure to realise real-time updates of movie data, and failure to add emotional analysis processing in film reviews,

in line with our task the main reason why this technique cannot be used to achieve our aim is that it requires high-level professionalism to implement, as it is very difficult to write programs. The technique described in this paper shows a positive outcome, the data collection is precise and direct, and the visualisation representation is efficient and intuitive as seen in figure 2, but the implementation shows a challenge as it is difficult to write thus it won't be adopted in our task at hand, this also led me into papers that show lesser challenges in implementation. The second paper reviewed [12], describes a web-based decision support system (web-based DSS) to help Hollywood producers make adequate decisions regarding factors affecting movies, these factors have been listed earlier, and these affecting factors or characteristics are used to build a predictive model that will classify movies into 9 categories from flop to blockbuster. Web-based (DSS) are computerized technological solutions employed to support complex decision-making and for solving problems[13], the existing problem of not being able to predict the success of a movie has motivated researchers to create this model to predict the financial success of a movie before its release or even production. The article accomplishes this by developing and integrating an information-based forecasting engine into a DSS that Hollywood producers may utilise. The Movie Forecast Guru (MFG) system is a web-based DSS that may initially reply to users' requests from a web browser. The web server is where its engine is located, it can use data and models from both local and remote sources and use prior knowledge to execute tasks which is to give the financial success of movie prediction and give an analysis of the affecting factors. to improve the performance of this model, previous outcomes are also stored in the model database. These are some of the reasons behind creating the model as a web-based DSS instead of its counterpart desktop application; Web-based technology has a single point of entry and access, which helps in providing the end users with only relevant information, the web-based system provides a more convenient way of version updating, the system accommodates different platforms and multiple decision-makers which can be manipulated to fit users' task. The uniqueness of this technique is that it uses a neural network to predict the market success of s movie before it is released into the theatre. The system consists of 4 models: neural network, decision trees, ordinal logistic regression, and discriminant analysis, they then used an information fusion mat-data to generate predictions from the output of all four models. The homepage, user interface and visual report are shown in the diagrams below.

Figure 6 - The homepage

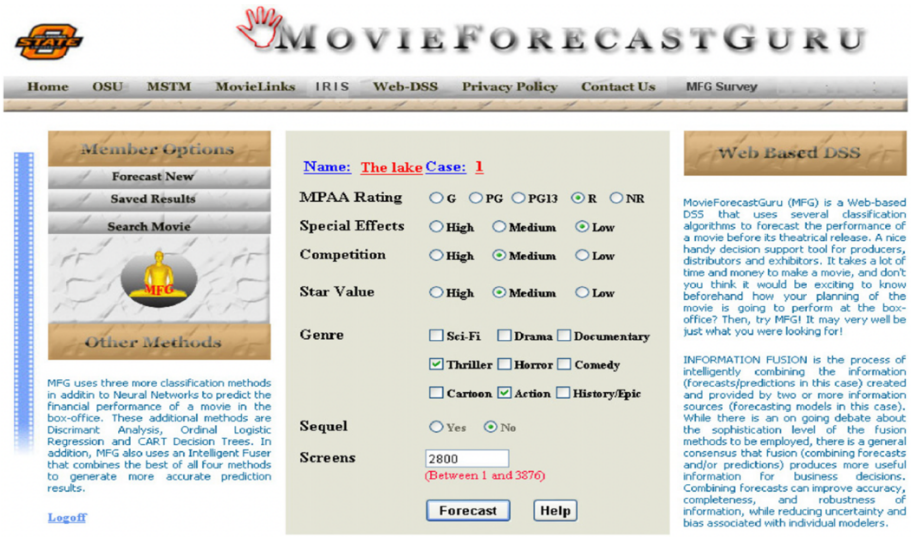

Figure 7 - User interface

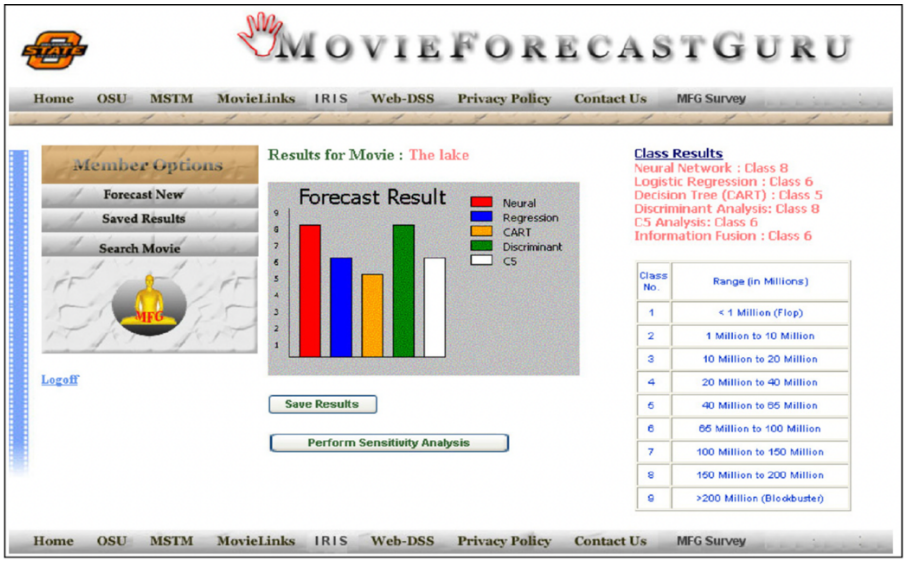

Figure 8 - Visual report

The MFG show a very promising visualisation, although it is a complex system and might pose a challenge in the implementation, the major challenge of this technique is that it shows a low rating for user understandability, from the user assessment result, carried out by the researcher, for this reason, I will say that this is not the best visualisation technique to employ in my task as it does not satisfy the users. The two reviewed papers are both similar as their performance is dependent on the web, to achieve a better visualisation technique that will be able to run, without depending on the web, further review of papers was carried out in this direction.

2.3 Similar works involving desktop software application

The next paper deals with the visualisation software Cinegraph as a graphical visualisation that allows users to explore and analyse data from the Internet Movie Database InfoVis 2007 competition (IMDB) [14]. Cinegraph combines two complementary visual interaction approaches, cross-filtered views, and attribute connection graphs, to provide a wide range of general and highly focused analytic operations. Users can express complex lines of the query as a rapid set of basic interactions. Cinegraph uses ancillary photographs of people, images of movie posters, and icons of movie genres to enhance the experience, providing high-dimensional interactive detailed functionality into the people, genres, awards, release dates, and box office attributes of movies described in the database. Using the Improvise visualisation platform, it was created and produced in just over two days by a single

visualisation designer. Improvises visualisation builder and browser are based on principles used in DEVise and other prior systems like Snap-Together Visualization which combine database searches with coordinated multiple-view visualisation techniques. These methods enable rich data browsing by translating user interactions into multidimensional space navigation and data item selection across several displays. Cinegraph's user interface design organises seven important parts of the cinema database: movies, ratings, release dates, genres, Oscars, individuals, and roles, into coordinated multiple views put out in seven regions. These views integrate two visual techniques for analysing small webs of interactions in high-dimensional data sets between spatial, temporal, and abstract aspects, by turning on and off brushing arbitrary sets of attribute values, cross-filtered views enable drill-down set searches across several tabular data columns. This paper introduced me to the multiple and coordinated view and which I adopted in my visualisation technique, although this software has no predictive nature as it only explores and analysis movie data. It also shows an engaging and interactive visualisation report with the use of multiple and coordinated views as shown in the diagram below.

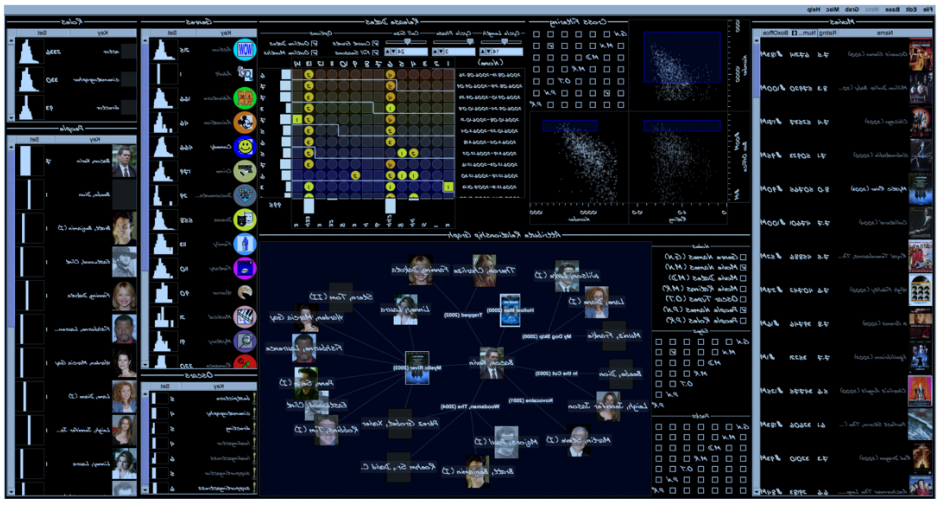

Figure 9 – Cinegraph visualisation software

This poster [15], presents an innovative use of social network visualisation tools in the film business. In analyses of the IMDb co-starring network, which contains over 2.6 million actors displaying over a billion links, with degrees ranging from about 50,000 and above for the most connected actors, the author makes the case and demonstrates with examples of how a k-cores-

based visualisation method relieves potentially intractable memory problems. The authors try to visualise the co-starring network, particularly using the Internet Movie Database (IMDb) network, as well as various attempts to visualise other aspects of movies, such as the network of actors featured in the same scene within a movie, storylines, and so on, are abundant in both published and unpublished reports. The writer states that to their knowledge, nobody has developed a means to represent the whole network of approximately 2.6 million actors owing to the difficulty of fitting the co-starring network of these players with over a billion linkages into memory. Trying to view the whole co-starring network with either the Pajek or k-core approaches utilised in this poster results in memory issues such as segmentation faults or memory locks. Another difficulty is making sense of visualisations, especially since identifying a specific actor in such large representations is challenging. To address this visualisation and analysis difficulty, they propose the k-core technique and compare the visualisations created using the k-core approach to those obtained using previous methods. They showed visualisations of two subsets of the co-starring network: the most linked actors (with 30,000 degrees) and the minority actors. Pajek was used to create the graphs. A k-core is a sub-graph H of a graph C whose degree of all vertices in H is greater than or equal to k, and H is the largest sub-graph with this attribute. If a vertex belongs to a c-core but not a (c+1) c- core, it has the shell index c. All vertices with the shell index c make up a shell of order c. Each cluster of order c is formed by a connected (Inside the original graph C) group of vertices with shell index c. The node's colour corresponds to its shell index. The degree of the nodes in the original network is represented by their size (in logarithmic scale) For vertices with a lower shell index, nodes are represented in concentric rings with a bigger radius. C-clusters (linked - within the original graph - nodes within the same shell) are presented near one another within a c-shell. The two halves of each edge are coloured with the colour of the corresponding extremities in a random sample of edges. Vertices can be placed more internally (if connected to vertices with higher shell indices) or externally (if connected to vertices with lower shell indices) for a given shell diameter. The data shows approximately 13 primary groupings of actors and at least 20 smaller diffuse groups. Only a few actors are found in the graph's maximal shell, which is located at the graph's very centre. As one proceeds from the outside to the inner rings, the size of the nodes does not noticeably grow. The "pie slice" structure of the graphs' sections is fascinating. It denotes that an actor with a high shell index is connected to a chain of related actors from various shells, which in turn is connected to a cluster in an outer shell. They pointed out that the colour range is extremely limited, implying that most nodes only cover a small percentage of the whole range of possible shell

indices (from minimum to maximum). They also pointed out that the outer shell is made up of players who mostly connect to other actors with similar shell indices, rather than actors with different shell indices. Following that, they look at and debate another representation, this time for a network of small actors. The result shows how the k-core representation of the network of fewer connected actors differs significantly from that of the most connected actors. Several circles depict components that are isolated from one another within the k-shells. The k-core technique offers a clearer picture of the network's numerous local communities, which is enabled by the successive "peeling" of the k-cores as the k rises. This poster has made a unique contribution to the field of movie analytics by using network visualisation techniques. Visualization is a key part of understanding network dynamics, but researchers have struggled to find a means to show very huge datasets until now. This suggests one method for the researcher to use to overcome this issue. Investigations into k-core representations of different subsets of interest inside the IMDb co-starring network, notably with the identification of key central actors and groups by genre, director, and so on, are among the other views. The relationship between this poster and my project is that they both deal with movie actors' categorisation, and data filtration to fit the task, although the visualisation technique used in this poster is well-defined and implementable, the resulting visualisation does not show adequate interactivity and might not be suited for my project. Hence technique was not adopted.

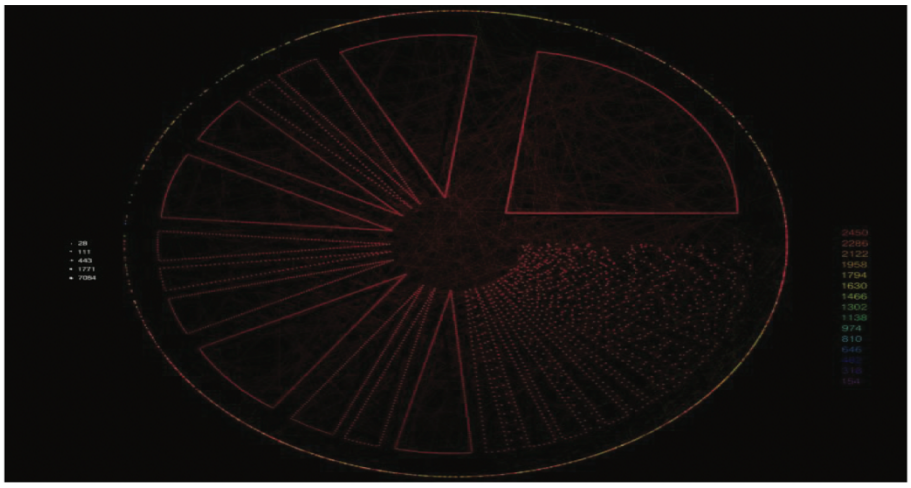

Figure 10 - Pie slice structure visualisation

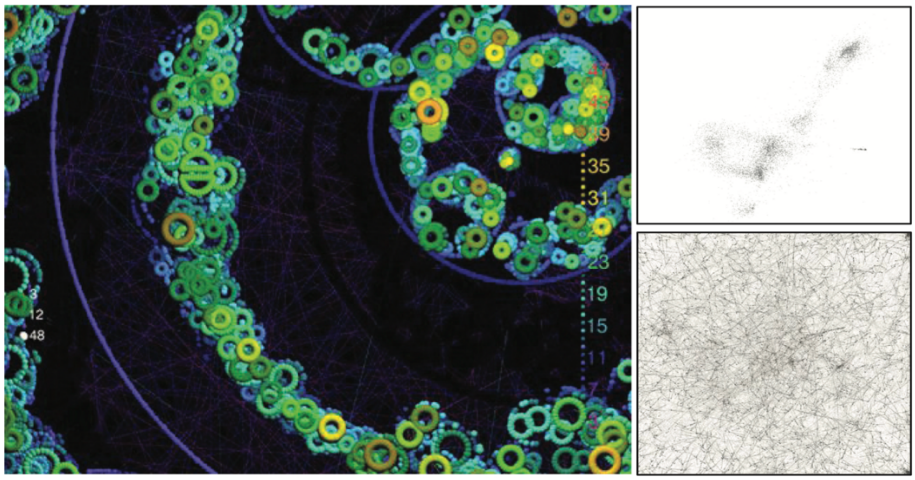

Figure 11 - k-cores-based visualisation

This paper[16], presents a brand-new way of visualising data that will build on currently used techniques, this visualisation is said to provide an understanding of movies by discovering hidden patterns through computer analysis. This kind of visualisation technique delves deeply into the film to examine editing patterns, movement, colour, and other aspects that would have been challenging to detect. The method presents a fusion of computer analysis and visualisation that also enables movie comparison to comprehend each movie's individuality while discovering trends and shared qualities. Cinemetrics generally refers to the statistical study of quantitative data, which is descriptive of the plot and other elements of films that could be considered stylistic. Barry Salt a film historian was the first to be actively involved in this type of technique. The cinemetrics website presents an open-access interactive website that is created to collect store and analyse scholarly data about films, the software is free and can be downloaded directly from the official website, after installation the software works simultaneously as users watch movies, thereby gathering data. Once the user stops watching the information gathered is presented to the user in a statistical format which will include the length of the movie, the number of shots and the average shot length, after the users can decide to upload the information to the websites database server for analysis, storage and publication, this process requires internet to be carried out. The system analyses the data in different ways but the most discussed is the approach by Friederik Brodbek, which he refers to as the movie's fingerprint. The analysis is done on the speech, colour and movements in the movie, the resulting visualisation is represented in several layers, which include the video, audio, subtitle and then the movements and colour mostly used in the movie, to give a final visualisation that is adequate and interactive. The visualisation is shown as a pie shape chart, with the size of

the chart representing the duration of the movie, users can see the colours used in the movie and the breakdown of each part, and the motion segment will show the movement in each scene of the movie, users can also enquire the statistics on each scene. The technique of analysing and visualising movies was designed to analyse movies of the same genre, or different works of the same director and to compare an original movie and its remake. Ultimately the system was designed for film lovers to choose their preferred type of movies based on the colour preference and movement in the movie, it also will suggest to the filmmakers the type of movies that viewers enjoy the most based on these criteria. The paper doesn't go into detail when describing the technicalities and implementation of this approach, but I was interested in it because of the idea and the appearance of the visualisation.

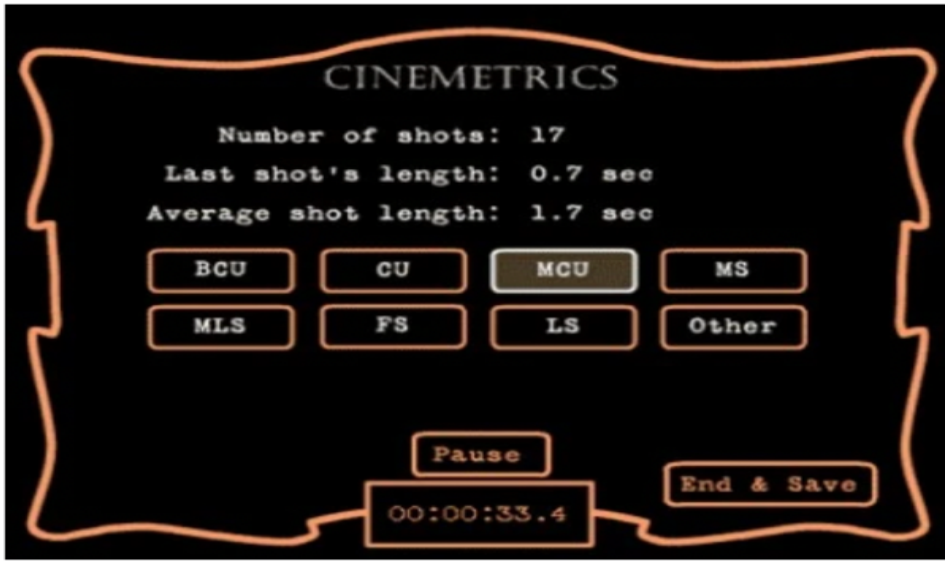

Figure 12 – Cinemetrics user interface

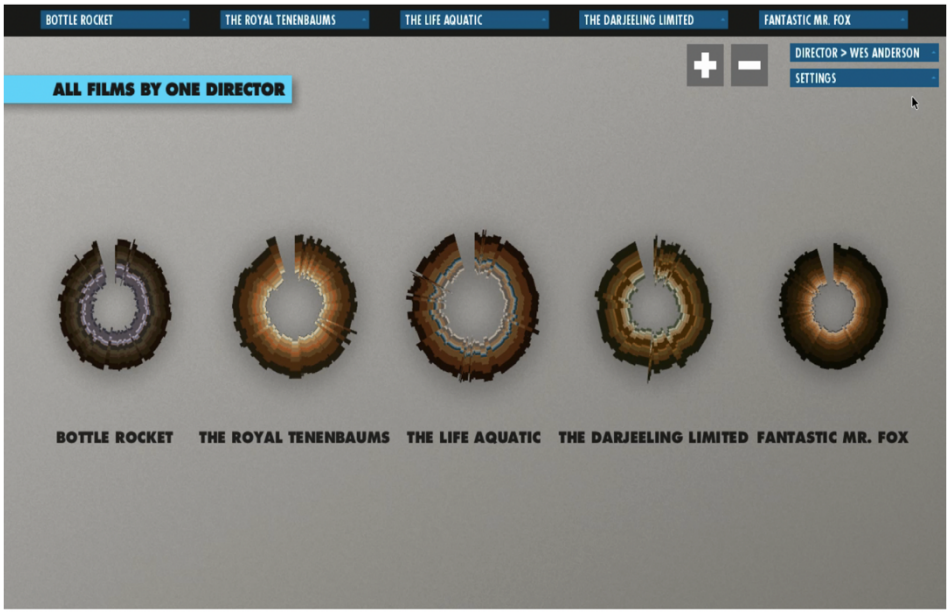

Figure 13 – Cinemetrics visualisation

2.4 Literature review on Python and Altair Library

The literature review gave me an insight into the Multiple Coordinated View, which I adopted in my implementation, then to understand more about the adopted visualisation technique, I had to research more on the technique, and my findings show in detail the implementation of the multiple coordinated view technique. I was able to understand Multiple Coordinated Views is an exploratory visualisation technique that allows users to look at their data in multiple ways. In fact, the technique's overall concept is that consumers will comprehend their data better if they engage with it and perceive it through various forms. Users want to examine extensive and nuanced data they want to investigate and discover things that are not readily available. These complex investigations necessitate the user to consider a variety of scenarios, comparing visualisations generated from multiple datasets, aggregating, and mining the data, possibly fusing data from multiple datasets to generate new information, and easily rolling back to a previous incarnation. Furthermore, multiple specialists may analyse the same data and compare and debate alternative exploration paths and conclusions. As a result, the exploratory tool must have a wide range of features while still being simple to use. I also furthered my exploration of open-source visualisation toolkits that have been developed to enable the production of more interactive visualisations in Python and Jupyter Notebooks.

Which includes Plotly, Bokeh, and Altair as examples. These libraries use web technologies to generate visualisations that can be viewed in web browsers. Plotly and Bokeh are extremely similar to matplotlib in terms of syntax. On the other hand, both libraries have been designed with user interaction in mind, allowing the creation of web-based dashboards with interactive widgets and charts that support multiple user inputs such as click, drag-and-drop, tooltips, and selection, cross filter, and bidirectional communication with Python via call-backs. The way visuals are defined in Altair differs from the other libraries; it employs a declarative specification that transfers VEGA-Lite, a data visualisation grammar, to Python. This library's flexibility comes from expressing a large range of interactive visualisations with only a few Altair primitives. Altair uses a pythonic adaptation of the Vega-Lite specification to facilitate the construction of interactive visualisations. Altair employs a declarative visualisation paradigm. Instead of telling the library how to construct a chart step by step, the programmer simply provides the data and visual encodings, and the library handles the rest. A Pandas DataFrame holding the data to be visualised is required for the developer to generate a chart. Jupyter Notebooks are popular for data exploration because they allow analysts to create documents with software code, computational output, written text, and data visualisations. The efficiency and effectiveness of the Multiple coordinated views and Small Multiple visualisations approaches, in Altair library using jupyter notebook with python programming language, to users, in addition to its interactivity and cooperativeness with other tools, made me choose it as the visualisation technique I will be using for this visualisation task [17]–[24].

At this point in my analysis of the literature, I was certain that using multiple coordinated views would be a smart way to illustrate relationships in graphical form, as shown in earlier research studies. Given my proficiency with Python and my academic background in Visual Analytics, I did additional research on projects that had previously used this technique to better understand the implementation procedure. The first article I came across was a tutorial on using Altair in Python with a movie dataset[25]. The author introduces interactive visualisation in the paper and demonstrates how to do it using the Python library Altair. The author goes on to say that Altair is built on Vega and Vega-lite, which permits interactive visual representation in JSON format. The author continued by saying that what distinguishes Altair from other imperative APIs is its declarative nature, which frees the analyst to spend more time and energy understanding, analysing, and visualising data rather than writing the necessary code. In contrast, with imperative APIs, analysts must concentrate more on writing the necessary codes to produce the visualisation and specify sizes and limits in plotting. In

other words, while Altair requires analysts to concentrate more on what they want to do, imperative APIs require analysts to concentrate more on how to do something. The writer then demonstrates how to install Altair in a Jupyter notebook and explains why importing pandas is crucial because Altair is based on the panda's Data frame. He also points out that there are various methods of entering data into Altair, such as CSV, JSON, etc. the paper then ended with the writer carrying out the step-by-step implementation of the visualisation, this paper, and some other papers [17]–[24], gave me the confidence I needed to implement my visualisation. These papers were informative but didn't show adequate interactivity in their resulting visualisation. The implementation of my design was mainly inspired by the Altair tutorial website, which discussed multiple interactions https://altair-viz.github.io/gallery/multiple_interactions. The tutorial gave me more insight into some interactive tools used in Altair, some of which I adopted in implementing my visualisation, for example, linked highlighting, brush tool, splom charts and drag tool.

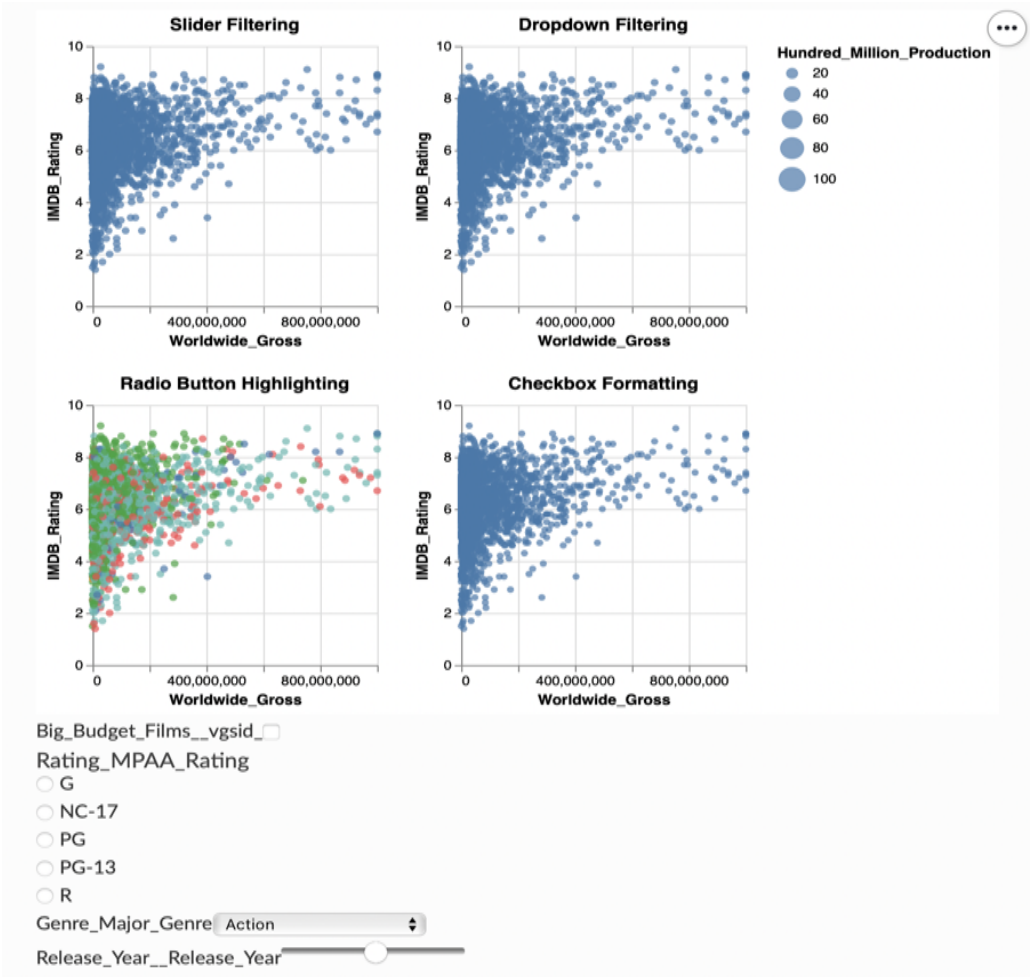

Figure 14 – Altair library visualisation

3. Design

The design of the visualisation implementation is inspired by the Atlas tutorial which discussed multiple interactions https://altair-viz.github.io/gallery/multiple_interactions. This visualisation software users are the movie production team. The data was sourced from IMDb and made available for download on the Kaggle website https://www.kaggle.com/datasets/danielgrijalvas/movies?resource=download, the data contained 6820 movies in the dataset, with each movie having the following attributes:

- Budget: the money spent on making a movie
- Company: the name of the production company
- Country: the country in the movie was made
- Director: the director of the movie
- Genre: the genre of the movie
- Gross: revenue got from the movie
- Name: name of the movie
- Rating: the rating of the movie (R, PG, PG-13, NC-17)
- Release date: date the movie was released YYYY-MM-DD)
- Runtime: duration of the movie
- Score: IMDb user rating
- Votes: number of user votes
- Star: main actor or actress
- Writer: the person that wrote the movie
- Year: the year the movie was released

The design of this visualisation is based on Silvia Miksch's model for interactive visual analytics, which explains interactive visualisation as a system composed of data, goals or tasks and users or audiences, represented in a triangular form, with these components having relationships. The relationship between data and goals or task is expressiveness, which refers to how well the visualisation can express parts of the data related to the goals and the task of the user, the relationship between goals or tasks and user or audience being appropriateness which implies, do the visualisation show the required information to do the task, the relationship between the user or audience and data is effectiveness, which means is the

visualisation effective, does it work with the user's perceptual system and effectively represent the data to the user. Which this in mind I designed my visualisation to meet these requirements.

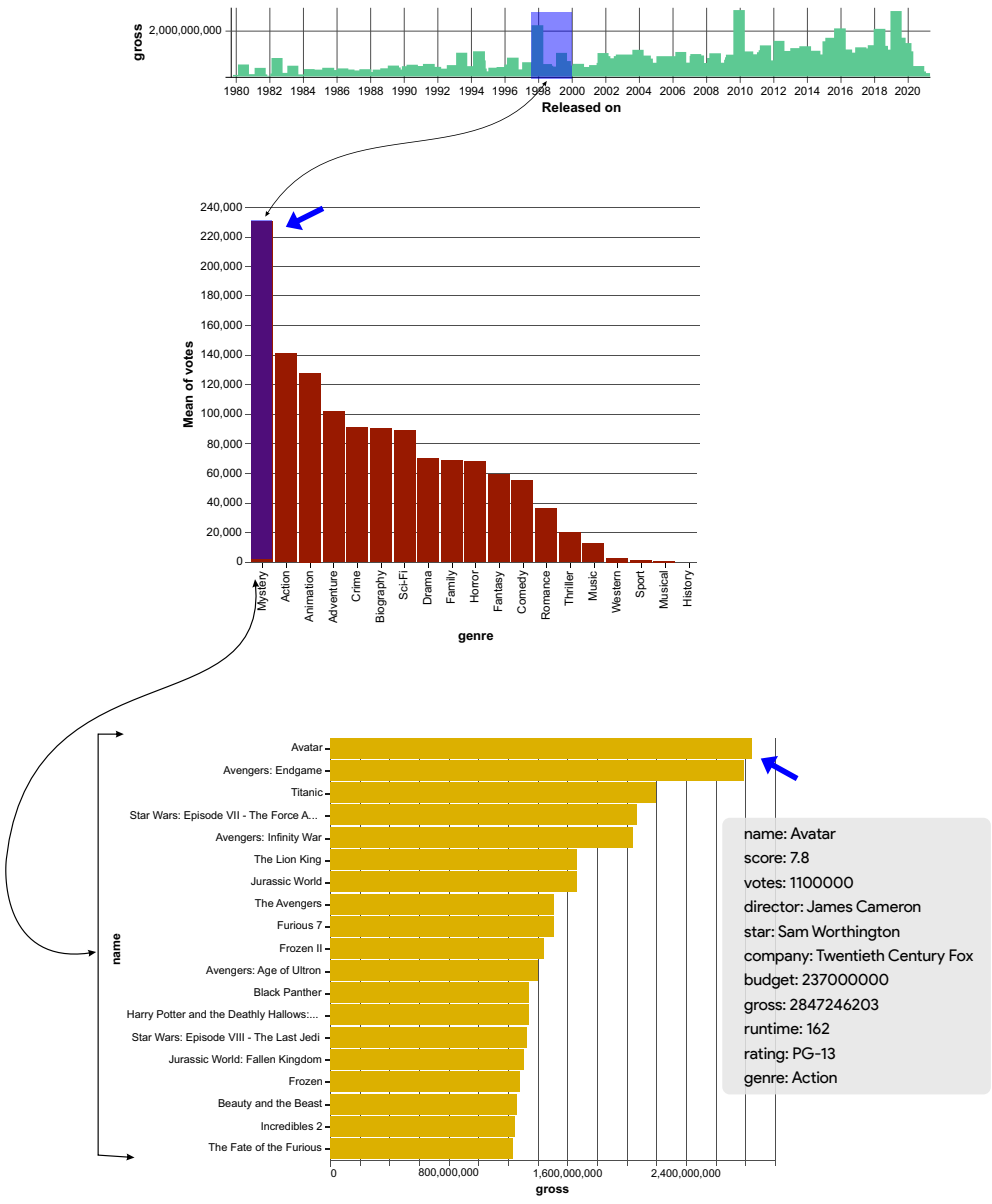

Figure 15- Overview of prototype

The first part of the visualisation is composed of three charts, the first one shows the Y-axis gross of the total movies per year, the X-axis shows the released dates from 1980-2020, and the first graph allows users to drag the release dates to highlight the range of years the user wants to inquire on, an example is shown on the figure above shows an inquiry on 1998-2000

as the years user is enquiring on, after the selection of the year, with the aid linked highlighting, the next graph shows the mean of votes of all the movies in the selected years on their various genre, the genre is on the X axis while the mean of votes on the Y axis, in descending order. Users can now click on the genre of movie that he/she needs to inquire about, on the prototype shown above mystery was clicked, the third chart also linked will then produce the list of 20 top movies in the selected years and genre and is sorted by the gross in descending order. The visualisation also has a tooltip that allows users to hover over the movies to show all information about the movie.

My initial design for this first part of my visualisation was not possible because of the limitation of Altair, the produced visualisation was stacked up and not sensemaking. In the initial prototype, I tried to achieve simultaneous sorting by gross and the votes, but this was not possible in Altair, as the Altair library doesn't allow multiple feeds on the axis. This visualisation would have been more efficient as it saves screen space and would be easier for users to read. The diagram is shown below

The initial prototype for the first part of the visualisation

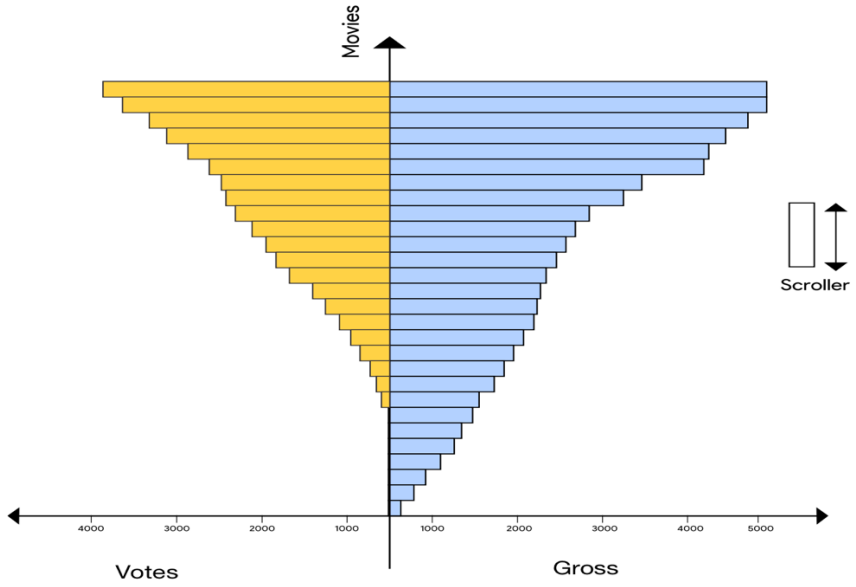

Figure 16- initial visualisation prototype

The second part of the visualisation is made of 16 splom charts which analyse the numeric information in the dataset and are linked to the first part of the visualisation, this part of the visualisation analysis the output of the enquiry, from the first part of the visualisation, which is the graph for year and genre of movies, the charts are also interactive through linked

highlighting, selected movies in one chart will automatically be shown in all splom charts removing the unselected movies, there is also filtering for these charts by rating, the difference rating clicked on will produce a visualisation of only movies in that rating, diagram shown below

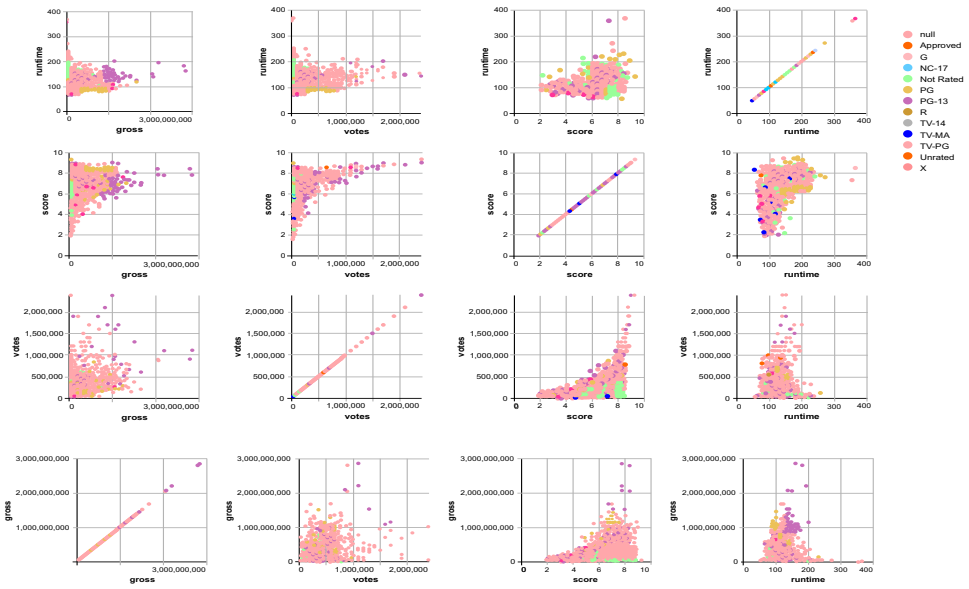

Figure 17- Second part of visualisation prototype

4. Implementation

This project visualises movie datasets using the Altair library in the python programming language in jupyter notebook, the goal of the project is to help movie makers analyse movie data interactively and efficiently, and identify the patterns used previously in making successful movies, to apply these methods while producing intended movies. firstly, since the visualisation is meant for Hollywood movies which are basically American movies, I had to get rid of all the movies that were not produced in the United States, also I needed to get rid of null values as some values were missing in the dataset, at this point pandas was imported, because of its data frame to read the dataset file. Simple python codes were written to achieve this, shown below.

```
import pandas as pd
data_all = pd.read_csv('/users/chineduezigbo/desktop/movies.csv')
data = data_all.loc[data_all['country'] == 'United States'] #To just select the movies from United States
data.dropna(inplace=True) #Drop all null values in any column of the dataset
```

Figure 18- Simple python codes for filtering data

Then I needed to create the base of the visualisation, which will show the genres of movies and their corresponding gross for each month of a calendar year, from 1980-2020, so I created a bar chart with the released dates on the x-axis and gross on the y-axis for each month of a calendar year, added interactive brush to select the time range by drag and release. The codes and visualisation output are shown below.

```
#Creating a base chart with release date and gross chart on top of which a small chart will be encoded
base = alt.Chart(data).mark_bar(size=10).encode( #The bar width is set to size 10
    x = 'Released on:T', #Release date is on x-axis of chart 1
    y = 'gross:Q', #Gross amount date is on y-axis of chart 1
).properties( #Property block controls the height and width of the chart
    width=600,
    height=400
)

#This chart uses the same parameters of base chart but the size of it is small to fit the other charts in the screen
lower = base.properties(
    height=60
).add_selection(release_brush)
```

Figure 15- Python codes for base chart

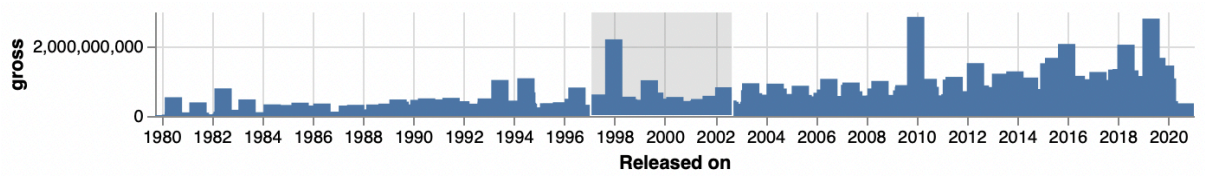

Figure 16- visual output of base chart

After the selection of the time period in the above chart, the next chart will then display all the movie genres and the average votes for each genre, it is sorted in descending order as we are mostly interested in the most successful movies mainly, in this chart the genre is in the x-axis while the mean of votes is on the y-axis, the reason for using gross by release date on the first chart and using mean of votes by genre in this second chart, was because votes and gross are the main criteria for measuring the success of movies as already stated in the introduction of this paper, although my initial prototype would have provided a simultaneous visualisation using both gross and votes, this was not achievable for me in Altair, at this time, so I implemented this design as it also achieves the intended aim, by use of two separate charts. The code snippet below creates a bar chart (before and after the selection of genre) as mentioned above, and the interaction tool is a colour-based selection of the genre. Users can click on the genre they which to inquire about for more information.

```
#This chart shows all Genre with the average votes for each Genre
chart = alt.Chart(data).mark_bar().encode(
    y=alt.Y('mean(votes):Q'), #Y-axis is the average/mean of votes
    x=alt.X("genre:N",sort='-y'), #X-axis is the all Genre
    color=alt.condition(genre_selection, alt.ColorValue("steelblue"), alt.ColorValue("lightgrey")) #Color condition is used for highli
).add_selection(genre_selection #Add Selection block contains all the filters from other charts -> genre_selection is from this chart,
                ).transform_filter(splom_brush).transform_filter(legend_selection).transform_filter(release_brush)
```

Figure 17- Python codes for movie genres by votes chart

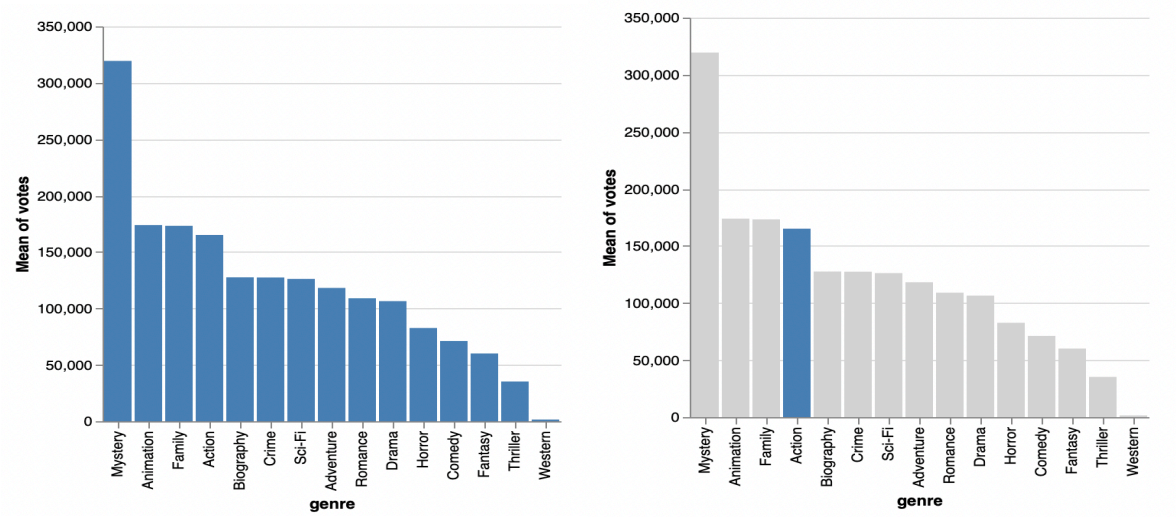

Figure 18- Output of python codes for movie genres by votes

Then to pass the selected genre on to the next level of charts, there is a hidden block of codes that resides at the back of the above chart's code, which helps read the user's selection and pass the exact rows from the dataset onto the next charts.

```
ranked_text = alt.Chart(data).mark_text(align='right').encode( #This is the chart where the data resides at the back of chart 2 to mak
    y=alt.Y('row_number:O',axis=None) #Row number of the data will be Y axis to store the selected data to pass it on
).transform_filter(release_brush  #release brush from chart 1
).transform_filter(
    genre_selection  #release brush from chart 2
).transform_window(  #trnaform window helps to perform required calculations over selected groups of data.
    row_number='row_number()'
).transform_filter(
    'datum.row_number < 20' #Only top20 movies by gross is displayed considering the screen space
).transform_filter(splom_brush).transform_filter(legend_selection) #both filters from SPLOM chart

chart2_bar = ranked_text.mark_bar(width=20).encode( #above selected data from ranked_text is used as base of this chart
    y=alt.Y('name:N',sort='-x'), #Y axis is the name of the movie and it is sorted based on gross in descending order
    x='gross:Q', #x axis is the gross values of the movies
    tooltip = ['name:N','score:Q','votes:Q','director:N','star:N','company:N','budget:Q','gross:Q','runtime:Q','rating:N','genre:N'],
    order=alt.Order("gross:Q"), #ordering the data based on the gross
).transform_filter(splom_brush).transform_filter(legend_selection).transform_filter(genre_selection #filters from all other charts for
).transform_filter(release_brush)
```

Figure 19- Hidden block of codes for selection and passing data

The row number 0, on the Y-axis, takes care of selecting the exact rows from the dataset and passing it to other charts where I used ranked_text chart as the base. Now that I have gotten the date range and the genre, then based on these two selections, the next chart will give the top 20 movies in that genre, and the time period which is sorted based on the gross in descending order. the visualisation has a tooltip added to the chart for users to get all details of the top movies in the genre to give them more insight. This chart is filtered based on the first 2 chart values on the released-on dates and the genre selection.

```
chart2_bar = ranked_text.mark_bar(width=20).encode( #above selected data from ranked_text is used as base of this chart
    y=alt.Y('name:N',sort='-x'), #Y axis is the name of the movie and it is sorted based on gross in descending order
    x='gross:Q', #x axis is the gross values of the movies
    tooltip = ['name:N','score:Q','votes:Q','director:N','star:N','company:N','budget:Q','gross:Q','runtime:Q','rating:N','genre:N'],
    order=alt.Order("gross:Q"), #ordering the data based on the gross
).transform_filter(splom_brush).transform_filter(legend_selection).transform_filter(genre_selection #filters from all other charts for
).transform_filter(release_brush)
```

Figure 20 - Python codes for list of movies

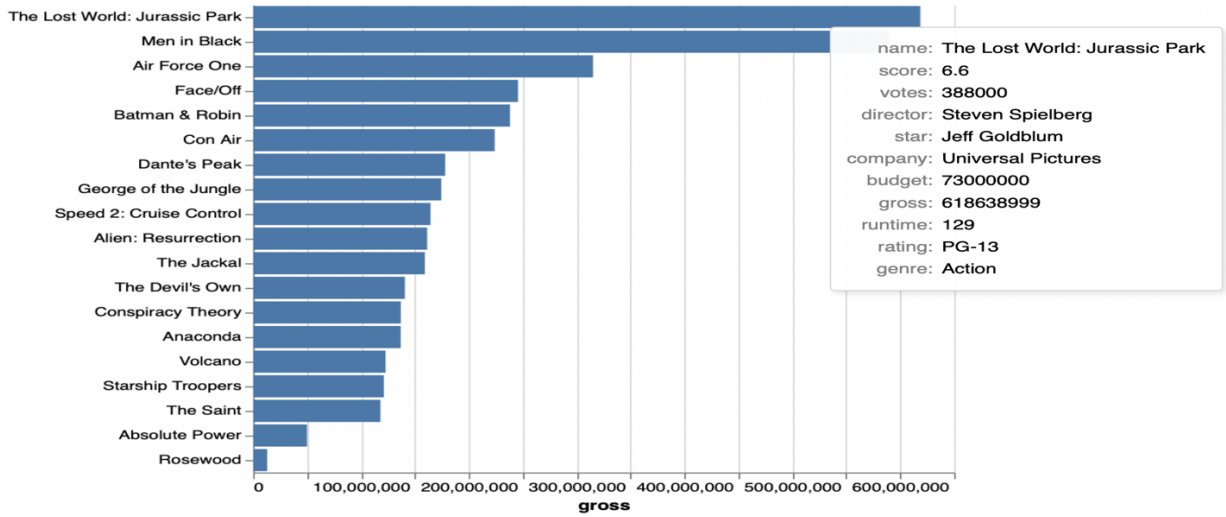

Figure 21- Output of python codes for selection

Now I have reduced the number of records by adding 3 charts with the filter at the top which makes sure that we are left with few records to make it easier for the user to visualise them and identify the relations among them, in the next level of charts, which are shown in SPLOM, which analysis the numerical components of the dataset. There is a block of codes that takes care of the SPLOM generation, and I provided a tooltip on this chart to see the details of the exact movie selected with the help brush tool added to the SPLOM.

```
SPLOM = alt.Chart(data).mark_circle().encode( #SPLOM charts is built with circle chart
    alt.X(alt.repeat("column"), type='quantitative'), #X-axis takes values from column entered below
    alt.Y(alt.repeat("row"), type='quantitative'), #Y-axis takes values from rows entered below
    tooltip = ['name:N','score:Q','votes:Q','director:N','star:N','company:N','budget:Q','gross:Q','runtime:Q','rating:N','genre:N'],
    color='rating:N' #Splom chart is colored based on ratings which is also used in legend selection
).properties(  #Property block controls the height and width of the chart
    width=150,
    height=150
).repeat(  #repeat block take care of the SPLOM design with the rows and columns selected
    row=['runtime', 'score', 'votes','gross'],
    column=['gross','votes','score','runtime'] #columns field order is to be reverse order of the rows
).add_selection(splom_brush,legend_selection #2 different selections are available in SPLOM charts - Brush selection with respect to x
                ).transform_filter(genre_selection
).transform_filter(splom_brush).transform_filter(legend_selection).transform_filter(release_brush #This contains all filters from char
                                                                                    ).interactive() #This chart is interactive
```

Figure 22 - Python codes for SPLOM generation

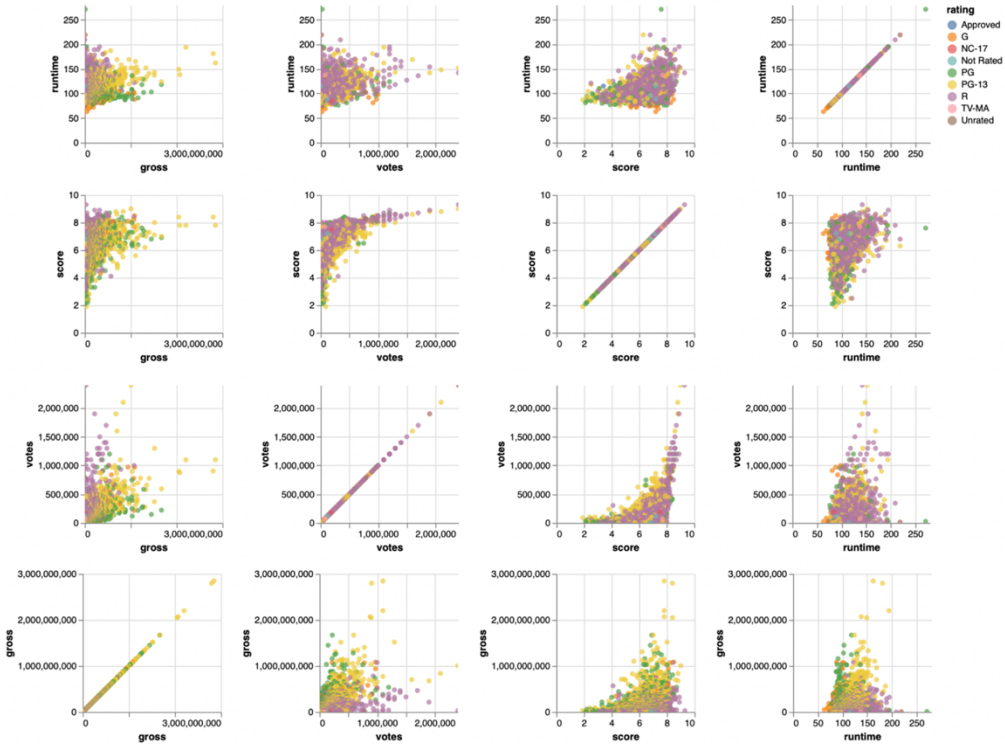

Figure 23 – Output of python codes for SPLOM generation

The other feature added in the SPLOM for interactivity and easy filtration of data is the legend selection, which helps users to filter out data and focus on the exact movie, to get correlations among attributes. This means that if a user selects the rating PG-13 in the legend, the SPLOM will show movies with a PG-13 rating only. Then also another feature in the SPLOM is the linked highlighting which allows users to drag and select some movies on the SPLOM and only the selected movies will be shown in the entire SPLOM.

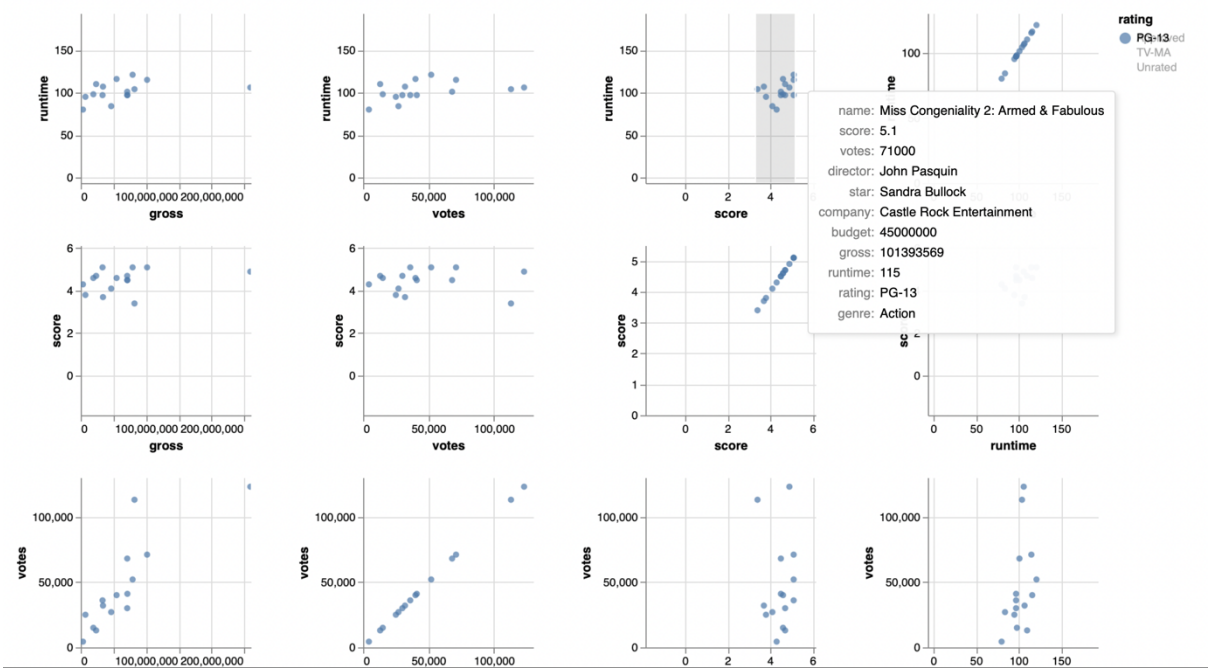

Figure 24 – Output of python codes for SPLOM generation with filtration

Now I have to arrange the entire visualised output in a way that it will serve as 'eye over the memory' for the users and also save screen space, which is vital in making a good visualisation from my knowledge of visual analytics, so I used both horizontal and vertical concentration, the genre and the list of movies are placed next to each other so that users can interpret the changes in the list of movies. then all charts are vertically concatenated with the release date chart at the top followed by the genre and list of movies chart and at the end SPLOM chart. The legends for charts are set to independent to make sure that the legends don't overlap. The code block is shown below.

```
Row_1 = alt.hconcat( #Horizontally concatenating Chart 2 and Chart 3 to place it on Row 2
    chart,
    chart2_bar)

Row_2 = release_chart & Row_1 & SPLOM #All charts are vertically concatenated
Row_2.resolve_scale(
    color='independent' #This make sure that the legends are separate for all charts
).configure_view(strokeWidth=0)
```

Figure 25 – Python codes for legends

4.2 Evaluation by Case study

To show the efficacy of the produced visualisation software, I used some unique cases observed to show that the implemented software is efficient in performing the task that it was intended for. My first case study was to show the criteria that can affect the production of action movies. first, I choose the more recent year from 2010-2020, the visualisation provided me with all the genres of movies produced in these selected years and their corresponding aggregated votes, and then I proceeded to click on the action movie genre which I am interested in analysing. The graphical representation produced on the SPLOM shows a high budget for the movies gives a significantly higher gross the movies, but a high budget doesn't necessarily give a high value for the user's votes. The chart also reveals that runtime has the most impact on the movies produced, as high runtimes of the movie show a significant increase in the gross and the votes of the movie values, as we have discussed earlier votes and gross are the two main criteria in measuring the success of a movie.

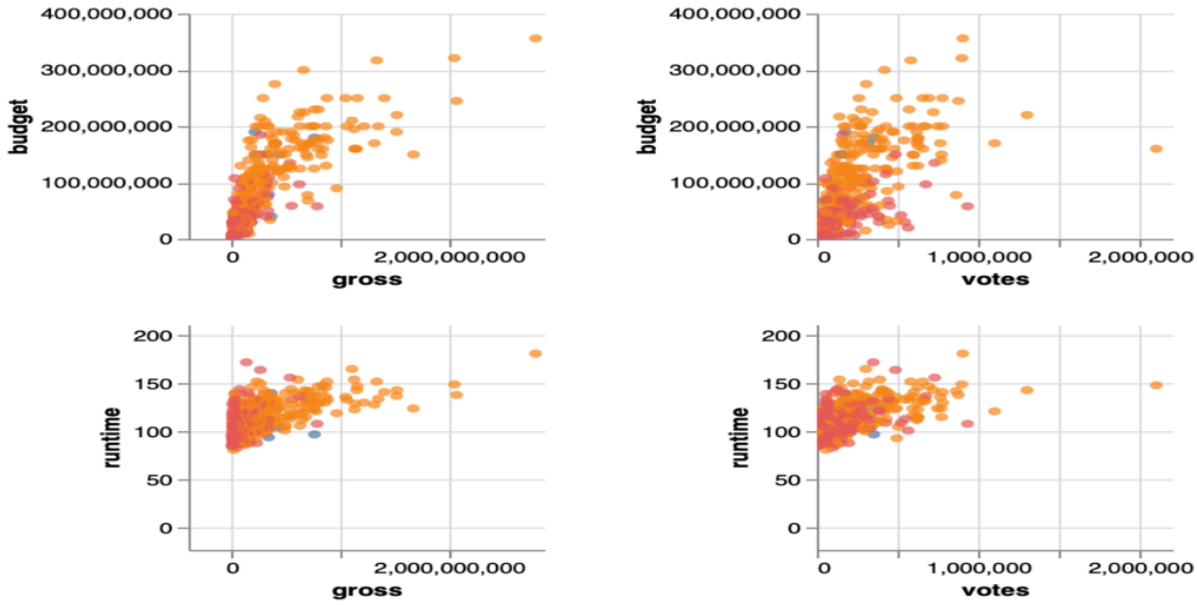

Figure 26 – SPLOM of movies from 2010-2020

Furthermore, I also wanted to get more analysis on the movies produced in the action genre this year, I observed that these movies have three different ratings which are PG-13, PG and R, I wanted to understand how the runtime affect these different rating of movies directly, using the rating filter, to select R rating, I discovered that the movies with rating R show a

less in grossing and votes as the runtime increases, movies with PG rating show less amount of high values as runtime increases but the PG-13 shows the highest impact as high runtime values show massive high values in movies gross and votes as compared to the other ratings. This observation might suggest that viewers enjoy action movies rated PG-13 with long runtimes.

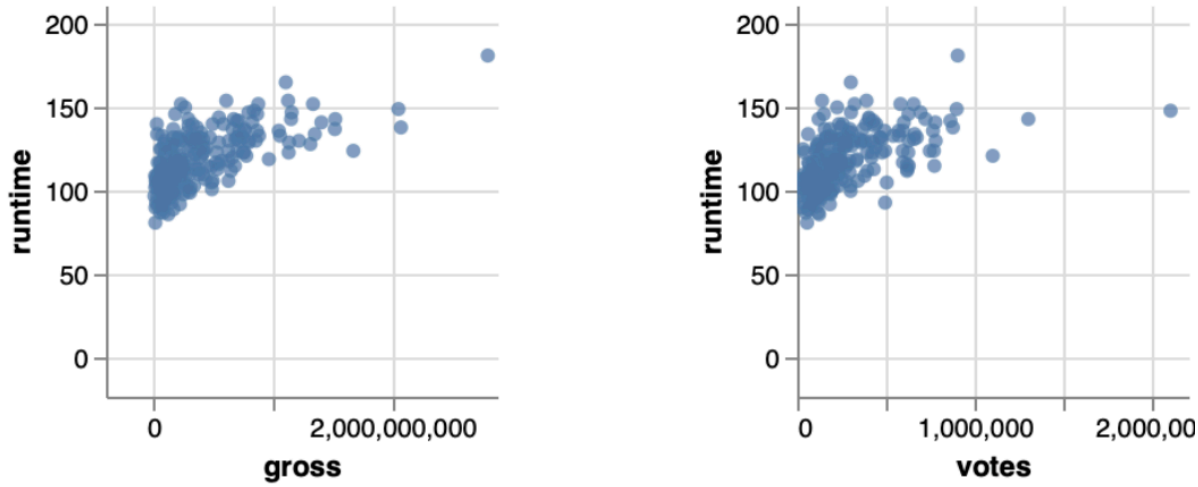

Figure 27 – SPLOM of PG-13 rating

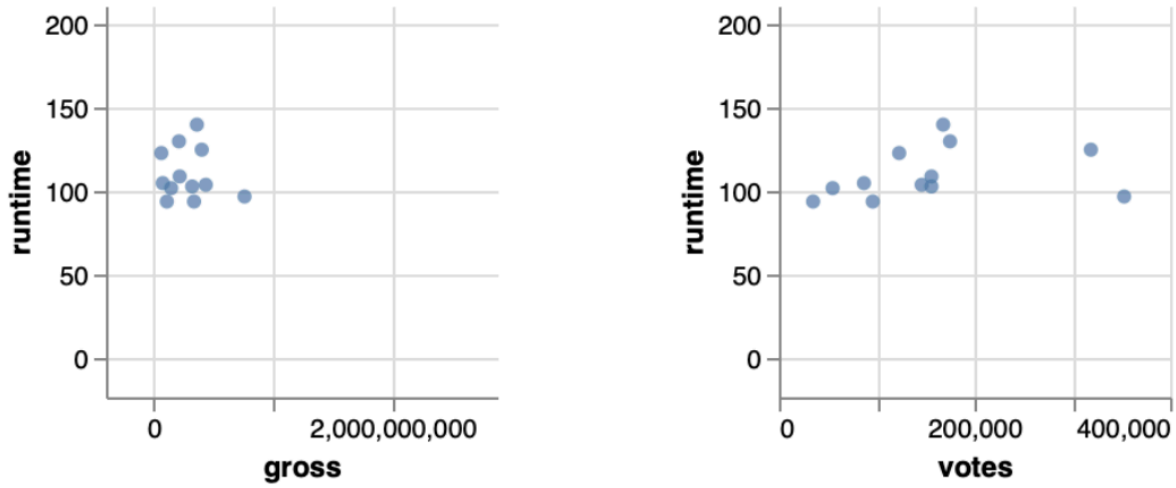

Figure 28 – SPLOM PG rating

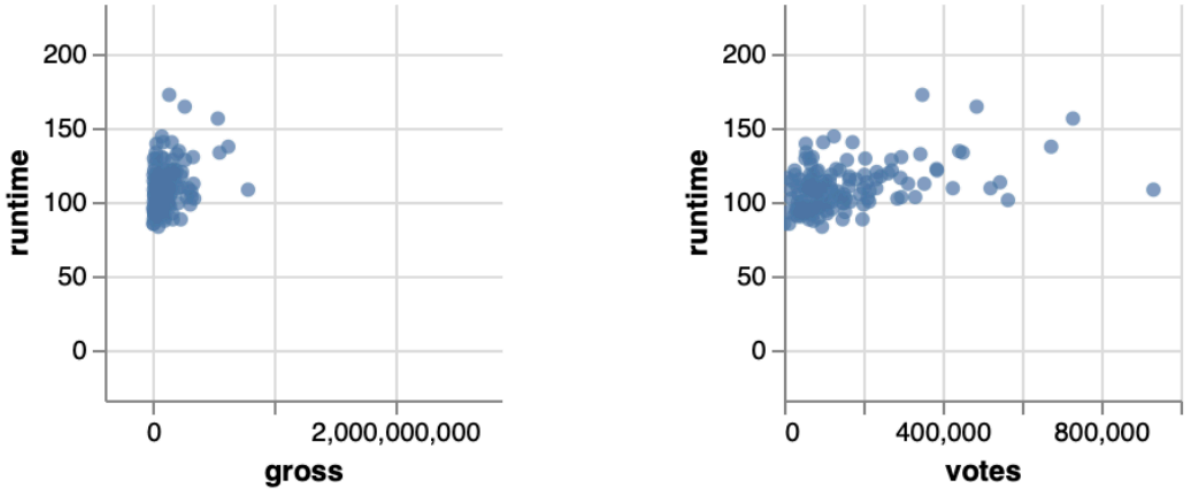

Figure 29 – SPLOM of R rated movies

The second case study shows that the higher budgets of horror movies do not have much impact on the outcome of horror movies, while high values in the runtime for PG-13-rated horror movies show relatively high grossing and user votes. This shows that to make good horror producers do not need to have a big budget, to pay more attention to engaging viewers.

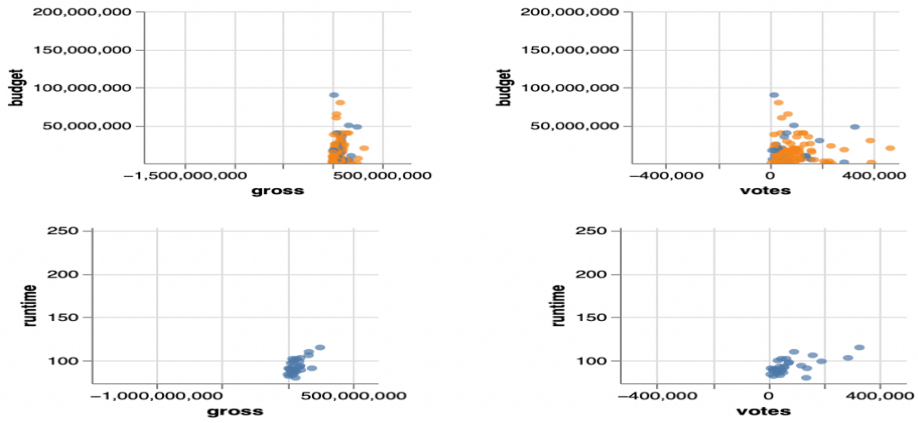

Figure 30 – SPLOM of PG-13-rated horror movies

The third unique observation shows, from the visualisation that R-rated comedy movies have the highest grossing with the highest votes from the year 2010-2020. This will give the producers insight when they are making comedy movies.

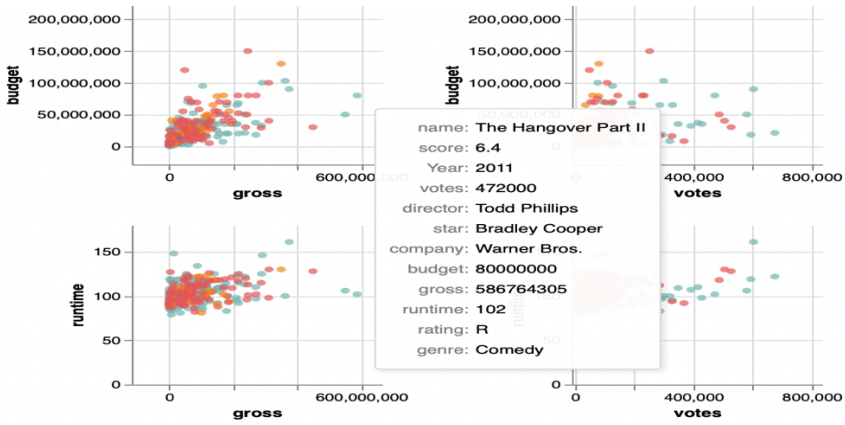

Figure 31 – SPLOM of R-rated comedy movies

5. Project management

In this part of the paper, I will be discussing in detail the sequence of producing the presented visualisation software, the chart below will show the calendrical sequences that lead to the production of the software.

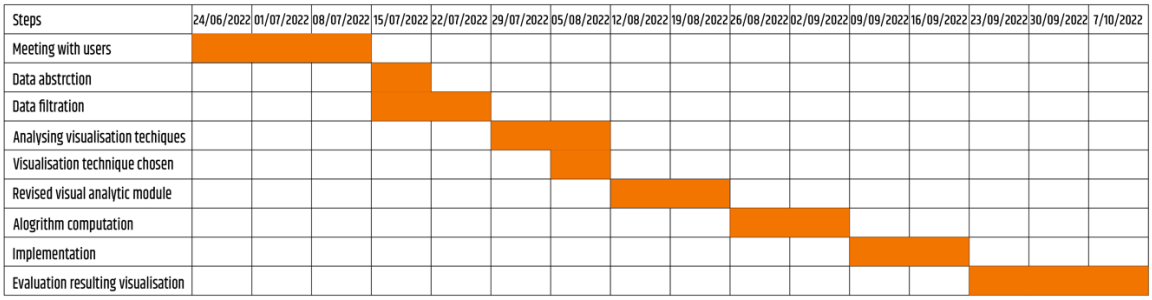

Figure 32 – Project calendrical sequences

Meeting with the users; the users of this visualisation as stated initially are the movie producers, I do not have any immediate movie producers, but I made out time to read articles about movie production to understand the difficulties the movie producers go through when making a movie to understand what movie producers look out for in movie visualisation and what type of information they may be interested in analysing in a movie dataset. This helped me in choosing the direction my implementation will be directed.

Data abstraction: most times it's easy to access data, the only difficulty was choosing the right dataset that will support and satisfy the user's task

Data filtration: some of the values in my dataset were missing and some values were also duplicated to mitigate thigh I wrote a python code that removes all null values, also dataset contained movies that were not produced in the United States as my visualisation involved only Hollywood movies, I also had to implement a code that will filter of the other movies that were not produced in United states.

Visualisation technique chosen: the visualisation technique I used was justified by my literature review, where I researched similar related works and choose the most appropriate for the task.

Revised Visual Analytics Module: I had to go back to my visual analytics module and had a thorough understanding of design and implementation in jupyter notebook using a python programming language with Altair library to be able to carry out the project.

Implementation: the implementation of the project was challenging Altair poses to have many restrictions which stopped me from Implementing my initial prototype, Altair tutorial documents and some other papers on implementation in using Altair helped me to mitigate the challenges.

The evaluation of this visualisation: the evaluation of the visualisation posed many the Altair brush is tough to use sometimes and because the charts are linked legends tend to overlap sometimes when we use interactions. The evaluation is still ongoing, and I will continue when I have more time.

Risk analysis

Personal risk

(P1) Hardware failure: this has a low likely hood of happening but the simple mitigation that I used is to back up my codes and writings in another hard drive, in this case, if there were any crashes of the hard drive the data will not be lost.

(P2) Health: the condition of mental and physical health is important to achieving any goal, to make sure I stay healthy throughout the process of achieving this goal I try to maintain good health conditions and avoid exposure to hazards.

(P3) Natural Disaster: this will be difficult to mitigate if it did occur.

(P4) Internet accessibility: I made sure the Internet was available to me at all times to avoid loss of information updates

Technical Risk:

(T1) Implementation: implementation of this visualisation was difficult mostly because of the restriction that Altair gave, these implementation challenges can be mitigated by using another visualisation technique or library, but I mitigated this by consulting my supervisor and enriching myself in information about Altair through my literature reviews.

(T2) Large data management: this posed a challenge during my implementation, as most of the visualisation was clogged up and could not give a readable visualisation, I was able to mitigate this by starting up my literature reviews with articles that deal with large data management.

(T3) Usability and Acceptability: users might not be able to read visualisation easily, causing them not to adopt the visualisation, to mitigate this I must have close contact with the user before and during the implementation of the task, to make sure I am working in line with their perspective, meetings with clients should be documented and I will have a backup visualisation just in case.

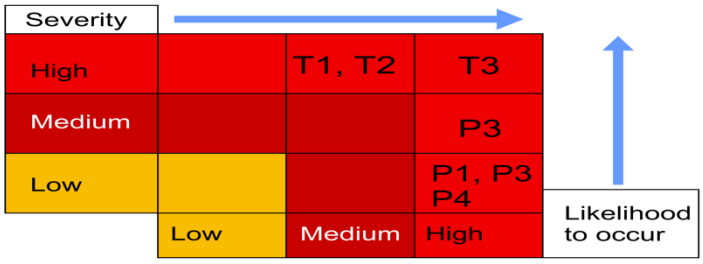

Figure 33 – Risk mangement

Immediately related works:

Some of the immediate works I would like to include in the visualisation software produced if given more time

1 I would try to add more interactive tools in the visualisation software.

2 I would try to implement the visualisation using another python library.

Distantly related works:

1 I would like to include a web-based data mining application in the visualisation software, this software will generate and store users viewing information from movie sites, this way I will not have to download incomplete data from the internet

2 I would like to make a desktop and phone application version of the visualisation software.

3 I would like to combine this visualisation software with a predictive algorithm, to make it possible to predict the outcome of the movie before they are produced.

Other areas that can benefit from this visualisation software:

This visualisation technique can also be applied in sports like football, it can be used to analyse the performance of a team based on their previous matches, and the result can be used to predict the outcome of upcoming matches. Analysis can be based on which stadium team performed best, the weather, formation etc.

6. Discussions

Data analysis has become a difficult task as the increase of data increases in size, this has necessitated the need for different visualisation techniques to serve different tasks, an interactive visualisation will help users to navigate and read the resulting visualisation with ease. This visualisation shows good interaction and navigation, the linking of the visualisation provides a top to bottom, and bottom-to-top linking, by the use of linked highlighting and multiple coordinated views coordinated. The visualisation will provide detailed information about the data provided, by providing a tooltip, the size of the visualisation fits the screen and doesn't accommodate more screen view, the multiple coordinated views of the SPLOM charts provide eye over memory advantage to users.

6.1 Conclusion

In conclusion, this project shows the use of the Altair python library to analyse movie datasets, and understand the pattern of previous successful movies made to help movie makers in making more hit movies, Altair has many interactive tools which were implemented in making this visualisation, the aim of this visualisation is to give an interactive visualisation that will be easy for the users to read and navigate through, in the visualisation I used small multiples or multiple coordinated views as it shows the correlation in data analysing, the tooltip was implemented to help provide details on demand, and also selective linked highlighting will coordinate the relationship through the chart. The SPLOM chart provides an indebt quantitative analysis of data.

[1] "Top 10 Movie Industries in the World | by InPeaks | Medium." https://medium.com/@inpeaksreviews/top-10-movie-industries-in-the-world-5d47cd9df44f (accessed Nov. 13, 2022).

[2] "Biggest movie flops: The 42 biggest box-office bombs." https://www.cbsnews.com/pictures/biggest-movie-flops-box-office-bombs/ (accessed Nov. 13, 2022).

[3] "Hollywood's most interesting blockbuster failures of the past 25 years | Den of Geek." https://www.denofgeek.com/movies/hollywoods-most-interesting-blockbuster-failures-of-the-past-25-years/ (accessed Nov. 13, 2022).

[4] "Great Films That Bombed at the Box Office | IndieWire." https://www.indiewire.com/gallery/best-films-box-office-bombs/wet-hot-american-summer-michael-showalter-christopher-meloni-a-d-miles-2001-c-usa-films-cou-2/ (accessed Nov. 13, 2022).

[5] Z. Gao, V. Malic, S. Ma, and P. Shih, "How to Make a Successful Movie: Factor Analysis from both Financial and Critical Perspectives", doi: 10.1007/978-3-030-15742-5_63.

[6] N. Sood and J. Balamurugan, "Factors Affecting the Success of Movies-A Case Study of Twin Movies," *Int J Innov Sci Res Technol*, vol. 2, no. 11, 2017, Accessed: Nov. 15, 2022. [Online]. Available: www.ijisrt.com

[7] N. Zhang, "Design of Movie Data Visualization System Based on Web Crawler," *J Phys Conf Ser*, vol. 1971, no. 1, Jul. 2021, doi: 10.1088/1742-6596/1971/1/012029.

[8] "(18) Large scale data analytics | Danilo Montesi - Academia.edu." https://www.academia.edu/32436529/Large_scale_data_analytics (accessed Dec. 07, 2022).

[9] M. Noirhomme-Fraiture, "Visualization of large data sets: the zoom star solution," *International Electronic Journal of Symbolic Data Analysis*, pp. 26–39, 2002.

[10] M. Vlachos and D. Svonava, "Graph embeddings for movie visualization and recommendation," in *First International Workshop on Recommendation Technologies for Lifestyle Change (LIFESTYLE 2012)*, 2012, vol. 56.

[11] B. R. Kent -, L. Yang, Z. Ma, and N. Zhang, "Design of Movie Data Visualization System Based on Web Crawler," *J Phys Conf Ser*, vol. 1971, no. 1, p. 012029, Jul. 2021, doi: 10.1088/1742-6596/1971/1/012029.

[12] D. Delen, R. Sharda, and P. Kumar, "Movie forecast Guru: A Web-based DSS for Hollywood managers," *Decis Support Syst*, vol. 43, no. 4, pp. 1151–1170, Aug. 2007, doi: 10.1016/J.DSS.2005.07.005.

[13] "(3) (PDF) Web-Based Analysis for Decision Support Systems." https://www.researchgate.net/publication/324521575_Web-Based_Analysis_for_Decision_Support_Systems (accessed Nov. 26, 2022).

[14] C. Weaver, "InfoVis 2007 Contest Entry: Cinegraph", Accessed: Nov. 28, 2022. [Online]. Available: http://www.personal.psu.edu/cew15/improvise/,

[15] D. Haughton, M. D. McLaughlin, K. Mentzer, and C. Zhang, "Movie analytics: Visualization of the co-starring network," *IEEE Symposium on Large Data Analysis and Visualization 2014, LDAV 2014 - Proceedings*, pp. 115–116, Jan. 2014, doi: 10.1109/LDAV.2014.7013216.

[16] "Cinemetrics: Creative Ways to Measure and Visualize Movie Data." https://www.printfriendly.com/p/g/gqDe5v (accessed Nov. 28, 2022).

[17] "Investigate TMDb Movie Dataset | Kaggle." https://www.kaggle.com/code/deepak525/investigate-tmdb-movie-dataset (accessed Dec. 09, 2022).

[18] "Data Science: Analysis of Movies released in the cinema between 2000 and 2017 | by Nicolas Chen | DataDrivenInvestor." https://medium.datadriveninvestor.com/data-science-analysis-of-movies-released-in-the-cinema-between-2000-and-2017-b2d9e515d032 (accessed Dec. 09, 2022).

[19] "Data analysis from Movie Dataset." https://tichung.com/blog/2017/12/data_analysis_from_movie_dataset/ (accessed Dec. 09, 2022).

[20] "RPubs - Data Visualization Final Project." https://rpubs.com/Nawal_Hasan/865526 (accessed Dec. 09, 2022).

[21] "100 Greatest Movies Data Visualization on Behance." https://www.behance.net/gallery/44767671/100-Greatest-Movies-Data-Visualization (accessed Dec. 09, 2022).

[22] "Exploring IMDB top 10000 movies with visualization." https://rstudio-pubs-static.s3.amazonaws.com/913363_1aa8f527360242238d38da7faf77406b.html (accessed Dec. 09, 2022).

[23] "Movie Dataset Visualization - Coding Ninjas CodeStudio." https://www.codingninjas.com/codestudio/library/movie-dataset-visualization (accessed Dec. 09, 2022).

[24] “Exploring Movie Data with Interactive Visualizations | by Kishan Panchal | Towards Data Science.” https://towardsdatascience.com/exploring-movie-data-with-interactive-visualizations-c22e8ce5f663 (accessed Dec. 09, 2022).

[25] “Altair in Python Tutorial: Data Visualizations<!-- --> | DataCamp.” https://www.datacamp.com/tutorial/altair-in-python (accessed Dec. 09, 2022).