Projektbericht zum Modul

Information Retrieval und Visualisierung

Sommersemester 2022

Bearbeitetes Thema:

Netflix TV Shows and Movies

Viet-Anh Do

Matrikelnummer: 220228228

GitHub Repository: https://github.com/do46/IR_Netflix_TVShows_Movies_VAD

Inhaltsverzeichnis

1. Einleitung

Streaming-Dienste wie Netflix, Amazon Prime oder Disney+ dominieren heutzutage den TV-Markt. Hinter der Erfolgsgeschichte dieser Video-on-Demand Plattformen stehen nicht nur ihre innovative Programme, sondern auch die Ausnutzung von Technologien, um ihre Kunden zu verstehen und ihnen maßgeschnittene TV Shows bzw. Filme anbieten zu können. Mit weltweit über 200 Millionen Abonnements bis zum 2. Quartal 2022 ist Netflix neben Amazon Prime die Marktführer im Streaming-Dienst-Bereich. JustWatch, eine Firma mit Sitz in Berlin, verfügt inzwischen über eine Datenbank mit über 3.600 Filme und 1.800 Serien [1]. Im Rahmen dieser Arbeit wird dieser genannte Datensatz von JustWatch genauer betrachtet, analysiert und schließlich aufbereitet für die Visualisierung, wobei unterschiedliche Visualisierungstechniken angewendet werden.

Das Hauptziel dieser Arbeit ist es, die individuellen Fragen zum Thema Netflix Filme und Serien von der Zielgruppen dieser Arbeit mithilfe einem Visualisierungsprogramm zu beantworten. Um dieses Ziel zu erfüllen, werden unterschiedliche Eigenschaften des Datensatzes durch zwei- oder mehrdimensionale Darstellungen gegenübergestellt. Zuerst einmal werden lediglich zwei Eigenschaften der Filme und/oder Serien in einem einfachen Scatterplot visualisiert. Weiterhin sollte die Anzahl der Dimensionen nicht nur auf zwei beschränkt werden. Um ein tieferes Verständnis des Datensatzes zu erhalten, werden die Daten mittels Parallelkoordinaten dargestellt, wobei die Achsen ebenfalls beliebig modifiziert werden. Schließlich, um einen Überblick über die gesamte Datenstruktur des Datensatzes zu gewinnen, werden die Daten durch einen Iconplot präsentiert.

1.1. Anwendungshintergrund

Netflix bietet Kunden in Deutschland eine Bibliothek von mehr als 5.000 Filmen und Serien an. Um einen Film oder eine Serie für einen Filmabend zu finden, muss man in der Regel mindestens eine halbe Stunde für die Auswahl verbringen. Oft werden Filme und Serien durch Netflix-Algorithmen empfohlen oder man verwendet zahlreiche Filter von Netflix, um auf das gewünschte Produkt zu kommen. Bewertungen aus namhaften Filmbewertungsdienste wie IMDb oder TMDb bzw. andere Charakteristika der Filmen werden oft nicht angezeigt, sodass Nutzerinnen von Netflix lediglich auf vorgeschlagene Filmen von Netflix zugreifen können. Diese Informationen sind vor allem für langjährige Kunden hilfreich, um etwas ältere Filmen zu entdecken, nachdem alle neue Filme schon angeschaut worden sind.

Des Weiteren scheint Filmproduktion in Westen in Bezug auf Ideen, auf Inhalte oder auf Genres gesättigt zu sein. Heutzutage erleben wir den globalen Aufstieg von Film- und

Unterhaltungsindustrie von Asienländer, insbesondere von Südkorea. Damit die Filmindustrie in Westen nicht im Rückstand liegt, müssen die Filmproduzenten auf Filmstatistiken und -daten achten, um für Kunden maßgeschneiderte Produkte produzieren zu können. Um dieses Ziel zu erreichen kann man die Methode der Visualisierung verwenden.

1.2. Zielgruppe

Im Folgenden werden die Zielgruppen dieser Visualisierungsarbeit ausführlich beschrieben. Diese sind namentlich Verbrauchs- und Medienanalytiker und junge Konsumentinnen und Konsumenten.

Die Verbrauchs- und Medienanalytiker sind die Gruppe, die sich hauptsächlich mit Medienbewertungen im Allgemeinen beschäftigen. Diese Visualisierungen können dieser Zielgruppe einen Einblick in den Filmtrend über Jahre hinweg, in den Zusammenhängen zwischen ihren Eigenschaften sowie einen gesamten Überblick über allen Eigenschaften der gesamten Filmdatenbank von Netflix verschaffen. Erkenntnisse aus dieser Visualisierung könnten sowohl für Filmproduzenten als auch für Streamingplattform vorteilhaft sein, denn sie können aus diesen Entdeckungen oder Trenderkennungen für dem Zuschauern maßgeschneiderte Filme oder Serien produzieren bzw. in den Portfolio aufnehmen.

Die Gruppe der jungen Konsumentinnen und Konsumenten können mithilfe der Visualisierung die Filmwelt selbst erkunden. Für viele junge Leute sind neben brandneue, auch alte Filme und Serien, von Interesse. Hierfür könnten die Bewertungen von beiden Datenbank eine gute Referenz für ihre Filmauswahl sein.

1.3. Überblick und Beiträge

Diese Arbeit ist ein Versuch, die gelernte Visualisierungstechniken auf ein echtes Anwendungsbeispiel anzuwenden. Der verwendete Datensatz befindet sich in der Webseite kaggle.com, eine zur Google gehörende Online-Community, die sich an Datenwissenschaftler richtet. Dieser Datensatz liefert dabei Auskünfte über Netflix Filmen- und Serienportfolio, wie etwa Name, Produktionsjahr, verschiedene Bewertungsmetriken von zwei Filmdatenbanken, also IMDb und TMDb.

Zudem können die entsprechende Zielgruppen die Kategorien auswählen und dementsprechend die Filme auflisten, für die sie sich interessieren. Die Datensätze liefern neben der Kategorisierung der Filmen einen Überblick über die Bewertung der Filme. Insgesamt haben zwei Unternehmen die Filme bewertet. Man kann auf dem ersten Blick erkennen, dass die Bewertung der Filme unterschiedlich ausfallen. Die dargestellten Visualisierungstechniken

ermöglichen den Zielgruppen die Produkte in Filme oder Serien zu unterscheiden. Interessiert man sich nur für Serien, kann man die Daten so filtern, dass man nur die Serien bekommt und je nach dem, was man gerne anschauen möchte, bekommt man die entsprechenden dargestellten Daten. Zudem sind sowohl für die produzierende Filme als auch die Serien das Jahr des Veröffentlichung angegeben. Zudem kann man anhand der Daten herauslesen, wie viele Besucher den Film angeschaut haben.

Der Anwendungshintergrund der ausgewählten Filme und Serien wird in diesem Kapitel angerissen und kurz erklärt und in Kapitel 3 wird den Lesern ein genauer Überblick gegeben. Die Auswahl des Scatterplots für diesen Datensätzen kann so argumentiert werden, dass anhand dessen viele Informationen über diese Streamingdienste gewonnen werden können. Dabei ist zu beachten, dass je nach dem welcher Zielgruppe man angehört, kann man dementsprechend die Achsen auswählen, um besondere Erkenntnisse hinsichtlich der Streaming-Angebote aus einem Blick zu erkennen. Die Parallelkoordinaten ermöglichen die Zielgruppen mehr als zwei Eigenschaften miteinander zu vergleichen, um ein besseres Verständnis hinsichtlich der Datensätze zu bekommen. Diesbezüglich können auch Erkenntnisse bezüglich Trends und Besonderheiten sowohl der Filme als auch den Serien der Interessierten ermöglichen. Mithilfe der dritten Visualisierungstechnik können die Zielgruppen Informationen bei den verschiedenen Merkmalen der Filmen hinsichtlich der zwei angegebenen Bewertungsunternehmen in x- und y-Achse abgebildet werden.

2. Daten

Wie bereits erwähnt stammt der verwendete Daten von einem Nutzer von der Plattform kaggle.com, wobei dessen Ursprung nach seiner Angabe aus der Film- und Seriendatenbank JustWatch zurückzuführen ist. Dieser Datensatz enthält 15 Spalten mit 5.978 Film- und Serieneinträge.

Der Datensatz fängt an mit der Spalte „id“, wobei sich die IDs von Filmen und Serien durch die Präfix „tm“ für Filmen und „ts“ für Serien unterscheiden. Neben „title“ für Titel, „type“ für Typen und „release_year“ für Erscheinungsjahr beinhaltet der Datensatz noch die folgenden Spalten: „description“, „age_certification“, „runtime“, „genres“, „production_countries“, „seasons. Eine kurze Beschreibung erhält man durch die Spalte „description“, während die Spalte „age_certification“ die Auskunft über Altersbeschränkung liefert. „runtime“ gibt die Dauer des einzelnen Film oder die Standarddauer einer Serienfolge an und „genres“ umfasst alle „tags“ von dem Film und Serien, welche sich selbstverständlich auf deren Genres bezieht.

„production_countries“ gibt die Information über Produktionsländer sowohl der Filmen als auch der Serien an, wohingegen in der Spalte „seasons“ nur die Angaben über die Anzahl an Staffeln der Serien eingetragen werden.

Anschließend befinden sich die Informationen bezüglich der zwei Datenbanken, IMDb und TMDb. „imdb_id“ gibt an, mit welcher ID der Film oder die Serie in der IMDb bezeichnet wird. Die IMDb-Bewertung befindet sich in der Spalte „imdb_score“. Registrierte IMDb-Nutzer können für jeden veröffentlichten Titel in dieser Datenbank eine Stimme (von 1 bis 10) abgeben. Einzelne Stimmen werden dann aggregiert und zu einer einzigen IMDb-Bewertung zusammengefasst. Nutzer können ihre Stimmen unbegrenzt aktualisieren. Die neue Bewertung wird die vorherige aber dann überschreibt, sodass eine gültige Stimme lediglich pro Titel und

pro Nutzer ist [2]. Die dazugehörige Anzahl der abgegebenen Stimme findet man in der Spalte „imdb_votes“. Von seitens TMDb enthält der Datensatz zwei Spalten, namentlich „tmdb_popularity“ und „tmdb_score“. Diese beiden Metriken werden anhand zahlreichen Daten kontinuierlich fortgeschrieben und täglich um 7 Uhr UTC aktualisiert. Zu diesen Daten gehören beispielweise die Anzahl der Stimmen pro Tag, Anzahl der Abrufe pro Tag, Bewertung des vorherigen Tages, Anzahl aller abgegebenen Stimmen, usw. [3]

Die Datenqualität dieser Daten eignet sich für beide Zielgruppen der Arbeit gut. Dieser Datensatz verfügt sich über eine klare Struktur, die einheitliche Formatierung des Zellenwertes innerhalb einer Spalte und die Vollständigkeit der Einträge, was die Qualität der Daten auch nach der Datenaufbereitung garantiert. Weiterhin mit über 5.000 Einträge bietet der Datensatz eine angemessene Tiefe sowohl für die Forschungs- als auch für die Selbsterkundungszwecke geeignet. Somit hat der Datensatz die Anforderungen der beiden Zielgruppen erfüllt.

2.1. Technische Bereitstellung der Daten

Die technische Bereitstellung der Daten erfolgt mithilfe des GitHubs Repository, in dem das Visualisierungsprojekt umgesetzt worden ist. Alle Daten, sowohl im Original („titles.csv“) als auch die Aufbereiteten, befinden sich im Ordner „Data“, wobei die letzteren im Unterordner „AufbereiteteDaten“ gespeichert wird. Der Prozess der Datenaufbereitung wird im anschließenden Unterkapitel 2.2 detailliert erklärt.

Das Dateiformat ist „comma separated values“ mit der Dateinamenserweiterung *.csv*, wobei die Werte der nebeneinanderliegenden Spalten mit einem Kommazeichen separiert werden. Da der Delimiter in diesem Fall „Komma“ ist, soll das Dezimaltrennzeichen ein Punkt sein, um

Mehrdeutigkeit zu vermeiden. Leerfelder sind allerdings nicht zu vermeiden und Einträge mit Leerfelder werden auch bei der Datenaufbereitung entfernt.

2.2. Datenaufbereitung

Die originalen Daten werden im Python mit dem Package „pandas“ aufbereitet. Visual Studio Code und Jupyter-Notebook finden auch Einsatz, um das Code besser zu visualisieren. Die Nutzung von Excel wird hier vermieden, da diese möglicherweise die Datenstruktur ändern könnte. Hinzukommt, dass Excel eigentlich für die Interaktion zwischen Maschine und Menschen gedacht wurde. In diesem Fall fungiert das Aufbereitungstool allerdings um eine Schnittstelle zwischen .csv Datei und Elm-Programmierung, darum ist die Nutzung von Excel nicht geeignet.

Aus dem großen Datensatz, die nach der Verarbeitung „titleslesslessdf.csv“ umbenannt wird, entstehen dazu noch zwei kleinere Datensätze, nämlich „moviedf.csv“ und „showdf.csv“. Der erste enthält Daten zu den Filmen, während in dem zweiten die Daten zu die Serien zu finden sind. Somit gibt es insgesamt drei Datensätze, ein gesamter Datensatz, der eine für Filmen und der andere für Serien. Die Unterteilung der Daten wird durch die Spalte „type“ ermöglicht, da diese genau angibt, ob es sich bei dem Eintrag um einen Film („MOVIE“) oder um eine Serie („SHOW“) handelt.

Wie bereits erwähnt sollen alle Einträge mit Leerfelder entfernt werden, damit alle Einträge gleichbehandelt werden können. Nach der Bereinigung reduziert sich die Anzahl der Einträge bei dem großen Datensatz von 5806 auf 5041 Einträge. Auch zu beachten ist die Daten bezüglich der Anzahl des Staffeln, welche bei Filmen immer mit Leerfelder versehen und im Python mit na (*not assigned*) gekennzeichnet werden. Die Einträge sollten aus diesen Grund nicht gelöscht werden. Die Funktion „fillna“ sorgt dafür, dass die Spalte seasons von Filmen durch einen bestimmten Wert ausgefüllt werden (hier den Wert „0“).

Weiterhin wird eine weitere Spalte namens numberTags generiert und am Ende der Tabelle zugefügt. Diese Spalte ist der Spalte genres zurückzuführen, wobei der Wert der generierten Spalte entspricht die Listenlänge der originalen Spalte. Dieser Wert gibt an, an wie viele Filmkategorien einem Film oder einer Serie zugeordnet wird. Filme oder Serien mit vielen Tags könnten beispielweise angeschaut werden. In Betracht kommen nach der Datenaufbereitung nur noch die folgenden Spalten: „id“, „title“, „type“, „release_year“, „runtime“, „genres“, „seasons“, „imdb_score“, „imdb_votes“, „tmdb_popularity“, „tmdb_score“ und „numberTags“.

3. Visualisierung

In diesem Kapitel wird den Lesern einen Überblick verschaffen, wie die Visualisierungstechniken bei den oben genannten Zielgruppen Anwendung finden und ob die ausgewählten Visualisierungen die Zielgruppen aussagekräftige Informationen liefern können.

3.1. Analyse der Anwendungsaufgaben und Anforderungen an Visualisierung

Um die individuellen Fragen der Zielgruppen zu beantworten, bestehen die Hauptaufgaben der Arbeit darin, die Zusammenhänge zwischen unterschiedlicher Merkmalen sowie die gesamte Struktur von dem Datensatz zu erkennen.

Um Fragen zwischen zwei Merkmalen zu beantworten, wurde Technik des Scatterplots in der Visualisierung Eins angewendet, um zwei unterschiedliche Eigenschaften miteinander zu vergleichen. So könnten verschiedene Zusammenhänge zwischen den Filmen- und Serienmerkmalen aufgedeckt und somit ein besseres Verständnis für diese geschaffen werden. Zudem muss man beachten, dass die visualisierte Variablen auch sinnvoll ausgewählt wurden. Aus diesem Grund wurde die zum visualisierenden Variablen-Paare bei dieser Visualisierung vordefiniert. Eine Gegenüberstellung von dem Paar TMDb score und IMDb votes scheint beispielweise nicht sinnvoll zu sein, da die Anzahl der Stimmen (*vote*) bei IMDb offensichtlich kaum Einfluss auf die Bewertung von TMDb-Nutzern hat. Infolgedessen wird dieses Paar ausgeschlossen. Des Weiteren sollten die Datenpunkte so dargestellt werden, dass eine Trendlinie zwischen Merkmale, falls vorhanden, gut erkennbar ist. Zum Schluss sollten auch die tatsächlichen Daten unkompliziert angezeigt werden, etwa wie der Film oder die Serie heißt, zu welchem Genre sie gehören, usw.

Wenn sich die Fragen um mehr als zwei Merkmale beziehen, benötigt man mehr als nur ein Koordinatensystem. Deswegen findet die Visualisierungstechnik Parallelkoordinaten Anwendung. Mithilfe dieser Technik lassen sich mehrere Merkmale auf einem Plot gegenüberstellen. Genau wie bei dem Scatterplot sollten sich die visualisierten Daten auch leicht abrufen lassen, damit die Erkenntnisse nochmal unkompliziert geprüft werden können. Am besten sollten auch Ausreißer ausgeschlossen oder ausgeblendet werden, damit der Trend besser erkennbar wird.

Zum Schluss sollte auch die gesamte Struktur der Daten visuell dargestellt werden. Neue Erkenntnisse könnten möglicherweise mithilfe eines Überblicks über alle Merkmale entdeckt werden. Ein Iconplot wird diese Anforderungen erfüllen, wobei wie bei den anderen

Darstellungen die Datenpunkte möglich ersichtlich, sowie die Werte bestmöglich abrufbar gemacht werden sollten.

3.2. Präsentation der Visualisierung

3.2.1. Visualisierung Eins

Bei der ersten Visualisierung des Projekts handelt es sich um ein Scatterplot (oder auch als Streudiagramm genannt). Mithilfe eines Scatterplots können stets zwei verschiedene Eigenschaften eines Filmes oder einer Serien gegenübergestellt werden, um deren Zusammenhänge bzw. Korrelationen erkennen zu können. Dieses Wertpaar der Eigenschaften wird in ein kartesisches Koordinatensystem eingetragen und bildet sich daraus einen Punkt, welchen in dieser Visualisierung durch einen Kreis repräsentiert wird. Wenn man mit der Maus auf diese Punkte schwebt, werden diese farblich hervorgehoben und deren Informationen werden zeitgleich über dem Scatterplot angezeigt. Die anzuzeigenden Eigenschaften können darüber hinaus auch angepasst werden, indem man ein anderes Eigenschaftspaar aus der Dropdown-Liste auswählt. Diese Liste befindet sich über dem Diagramm, wobei deren Einträge mit den Namen von Eigenschaftspaare beschriftet werden. Durch die Buttons über der Dropdown-Liste kann man auswählt, ob man nur Filmen, nur Serien oder beiden Typen visualisiert. Bei Filmen hat man beispielweise nur eine einzige Filmeinheit, wohingegen haben Serien oft viele Folgen. Als ein weiteres Beispiel haben Serien oft mehr als bloß ein Saison, während der Begriff „Saison“ gar nicht bei Filmen existiert. Aus diesen Gründe sollte man auch die Möglichkeit bekommen, die Filmen und Serien unabhängig voneinander zu beobachten. Das Scatterplot kann hierfolgend in Abbildung 1 zu erkennen.

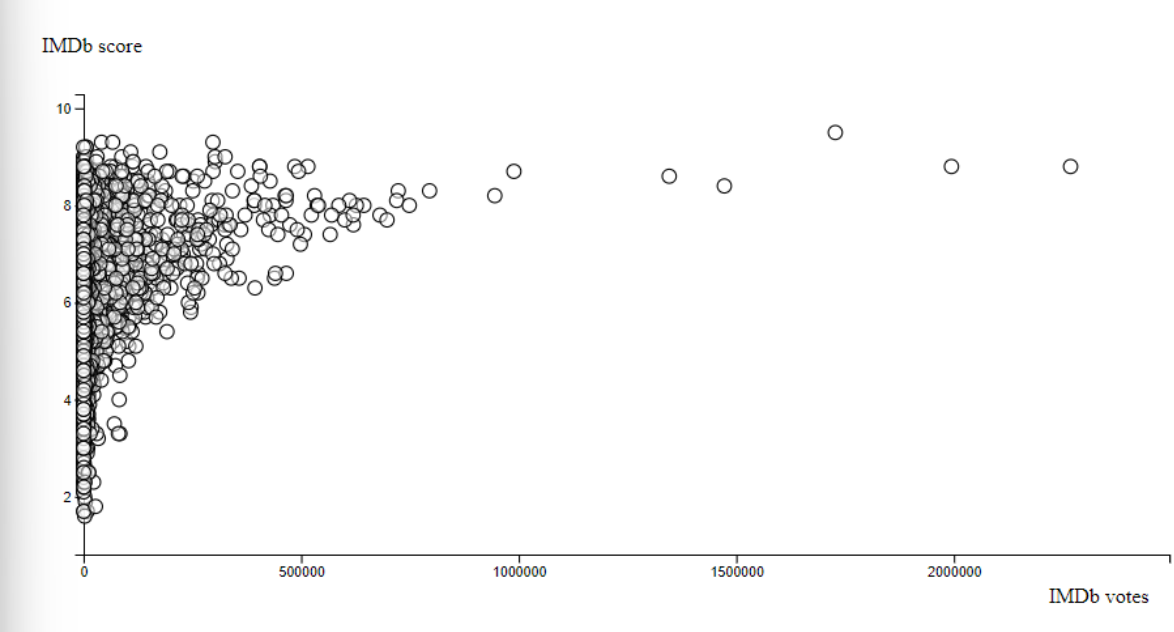

Abbildung 1: Das Scatterplot (Quelle: Eigene Darstellung)

Die Anforderung an das Scatterplot ist somit erfüllt. Man kann durch Auswahl der Optionen die Merkmalen der Filmen oder Serien gegenüberstellen, um mögliche Zusammenhänge zwischen ihnen zu entdecken. Die Seite wird nach der Auswahl nicht neugeladen und die Verzögerung ist nur in Millisekunden zu rechnen. Die Anzahl der Punkten wird ebenfalls angezeigt und die entsprechenden Trends oder Besonderheiten der Daten lassen sich auch erkennen.

3.2.2. Visualisierung Zwei

Die Visualisierung Zwei wird mittels eines Parallelkoordinaten-Diagramm dargestellt. Bei solcher Darstellungsform erhält man einen Überblick über mehreren ausgewählten Eigenschaften des Filmes und der Serien. Die auf den Achsen eingetragenen Datenpunkt werden miteinander durch Linien verbunden, sodass man eine Darstellung wie in Abbildung 2 erhält, wobei jeder Film oder jede Serie durch eine Linie repräsentiert. Darüber hinaus hat man die Möglichkeit, individuell den Eigenschaften zuzuweisen. Die zu beobachtenden Eigenschaften von den Achsen können ebenfalls angepasst werden. Wird ein Button geklickt, so wird auch die Eigenschaft von der entsprechenden Achse auch geändert. Damit die Übersichtlichkeit der anzuzeigenden Daten gewährleistet wird, werden die Linien farblich markiert und die entsprechenden Daten über dem Diagramm angezeigt, wenn man die Maus auf die Linien schwebt.

Die Achsen können dabei individuell den Eigenschaften zugewiesen werden. Dies erfolgt mithilfe von verschiedenen Buttons, welche über der Darstellung zu finden sind. Wenn dabei

ein Button gedrückt wird, nimmt die entsprechende Achse das gewünschte Merkmal an. Zusätzlich werden, wenn mit der Maus über eine Linie gefahren wird, die Eigenschaften angezeigt und die Linie farblich hervorgehoben. Die Parallelen Koordinaten sind in Abbildung 2 zu erkennen.

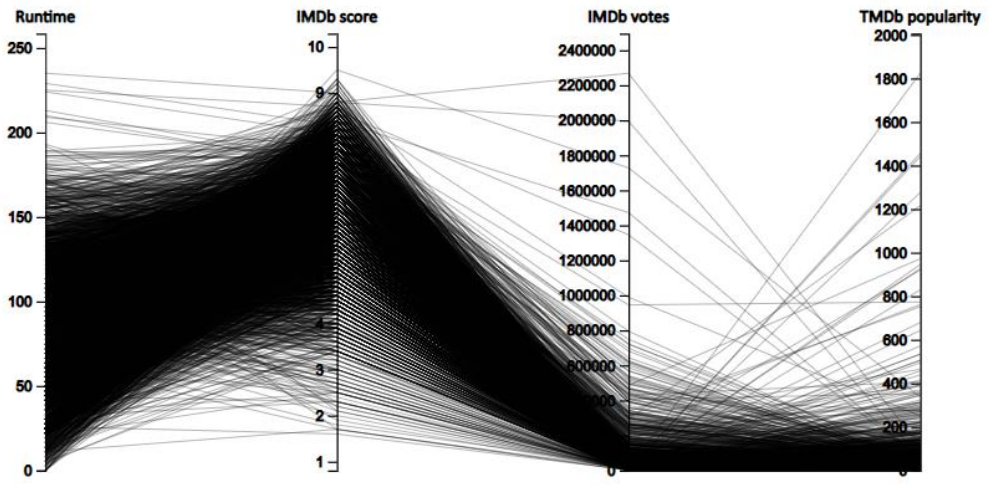

Abbildung 2: Parallelkoordinaten (Quelle: eigene Darstellung)

Weiterhin bietet Parallelkoordinaten noch die Möglichkeit, den Diagramm im X-Ray-Modus wie in Abbildung 3 zu repräsentieren. Die Linien werden verblasst und im weiß anstelle von schwarz dargestellt, während der Hintergrund schwarz gefärbt wird. Mithilfe dieses Modus lassen sich die Linien von Ausreißer ausblenden. Der Trend wird wohingegen deutlich angezeigt. Sobald der Trend erkannt geworden ist, kann man die Attributen genauer betrachten mit der 1. Visualisierung.

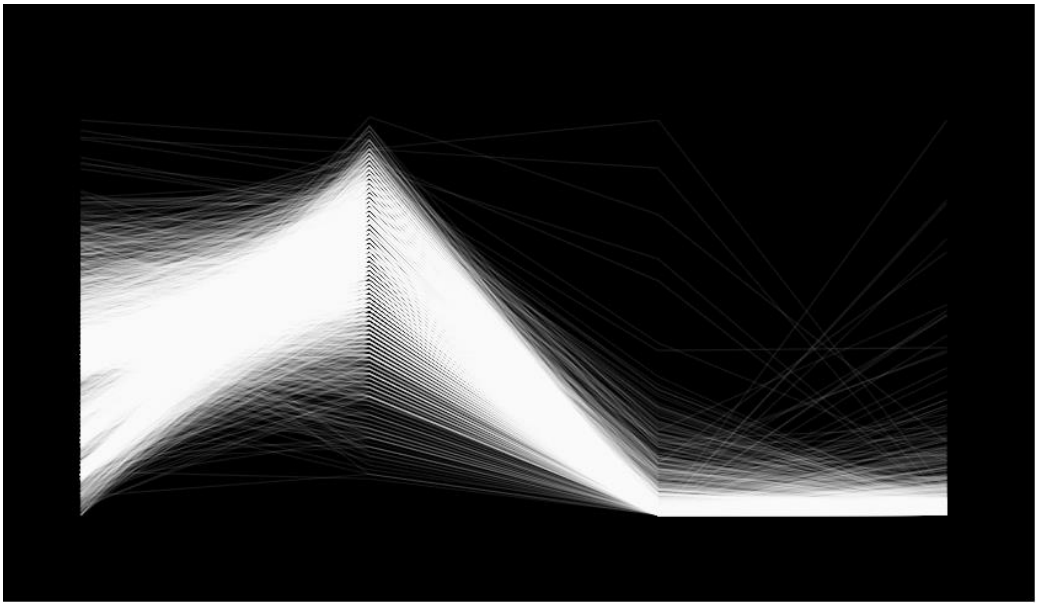

Abbildung 3: Parallelkoordinaten in X-Ray-Modus (Quelle: eigene Darstellung)

Die Anforderungen an diese Darstellung konnten erfüllt werden. So ist es mithilfe dieser Darstellungsform möglich, die verschiedenen Eigenschaften gegenüberzustellen und individuell anzupassen. Weiterhin werden Werte angezeigt, wenn innerhalb dieses Diagramms über eine Linie drübergefahren wird. Somit ist die Verfolgbarkeit und Nachvollziehbarkeit der Daten gegeben. Durch das Erfüllen der Übersichtlichkeit und der Nachverfolgung der Werte sind die Anforderungen an dieses Diagramm erfüllt.

3.2.3. Visualisierung Drei

Für die Dritte Visualisierung werden die sogenannte Stickfigure-Technik angewendet. Im Unterschied zu der ersten Visualisierung werden Datenpunkte mithilfe eines Stickfigure in einer mehrdimensionalen Darstellung visualisiert. Bei dieser Technik werden zwei Attribute verwendet, um das Stickfigure auf das Koordinatensystem platziert. Das Stickfigure selbst wird durch die restlichen Attribute gezeugt, wobei einzelne Attribute durch einen Winkel des Stickfigure abgebildet wird. Wie in Abbildung 3 zu sehen, werden *runtime*, *TMDb popularity*, *release year*, *IMDb votes* und *number of tags* durch den Körper, die linke und rechte Hände bzw. die linke und rechte Füße des Stickfigures dargestellt. Diese Stickfigures werden anschließend auf dem durch *IMDb Score* und *TMDb Score* abgebildeten Koordinatensystem platziert und somit erhält man den Iconplot. Um Daten anzuzeigen, kann man wiederum die Maus auf Stickfigure schwebt. Das angewählte Stickfigure wird gleichzeitig mit vergrößere Länge und dunkler Farbe hervorgehoben. Darüber hinaus ist es möglich, die Default-Größe von Stickfigure anzupassen. Wie bei die anderen Plots kann man mit Buttons die Filmen- und Seriendaten unabhängig voneinander betrachten. Die einzelnen Attributen lässt sich außerdem isoliert anzeigen durch Anklicken auf jeweiligen Buttons. So kann man beispielweise nur die linke Hände anzeigen lassen. Diese werden in diesem Modus rot markiert, wohingegen alle anderen Körperteile ausgeblendet werden.

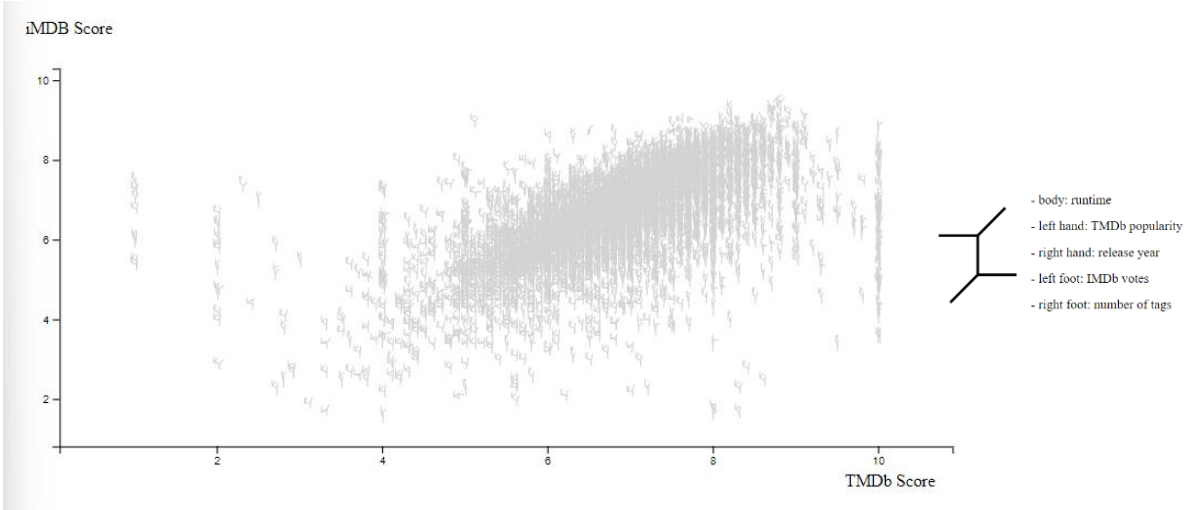

Abbildung 4: Iconplot (Quelle: eigene Darstellung)

Iconplot bietet eine Möglichkeit, den Gesamtdatensatz vollständig durch eine Übersicht zu betrachten. Man kann zwar nicht mit Iconplot den Zusammenhang anschauen, erkennt ihn aber trotzdem durch Bewegungsrichtung oder -tendenz der Körperteilen. Sobald sie erkannt geworden ist, kann man die Attributen mit den ersten und zweiten Visualisierung nochmal genauer betrachten. Schließlich ist die Anforderung an Visualisierung Drei wird wieder erfüllt und Visualisierung Drei wird mit Iconplot repräsentiert.

3.3. Interaktion

Interaktion zwischen Visualisierungen ist notwendig. Der Trend wird zuerst bei der Visualisierung Drei erkannt. Für eine detaillierte Ansicht kann man den Trend mithilfe der Visualisierung Eins und Zwei verdeutlichen. Ein schneller und problemloser Wechsel zwischen Visualisierungen ist erforderlich. Bei dem Homepage erhält man die Möglichkeit, Visualisierungstechniken auszuwählen. Innerhalb der Visualisierung kann sich frei zu den nächsten Visualisierungen oder zu der Hauptseite, ohne die Seite neuzuladen durch Anklicken auf die entsprechenden Hypertexte wechseln.

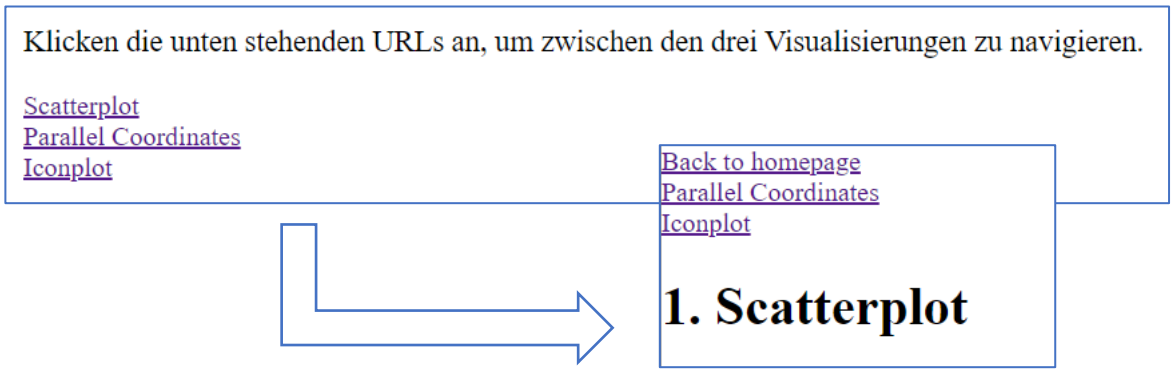

Abbildung 5: Interaktionen zwischen Visualisierungen (Quelle: Eigene Darstellung)

4. Implementierung

4.1. Gesamtaufbau

Das Projekt wird in fünf unterschiedlichen Elm-Modul aufgeteilt. Diese sind nämlich Main.elm, Data.elm, Scatterplot.elm, ParallelCoordinate.elm und Iconplot.elm. Die Daten werden mit Python-Jupyter vorgearbeitet, wobei das Packages Pandas verwendet wird. Die Vorverarbeitungsschritte sind in dem Modul datapreprocessing.ipynb zu finden. Die verarbeitete Daten werden anschließend in dem Ordner AufbereiteteDaten gespeichert. Eine Übersicht über des Aufbaus von dem Projekt befindet sich in der Abbildung 6.

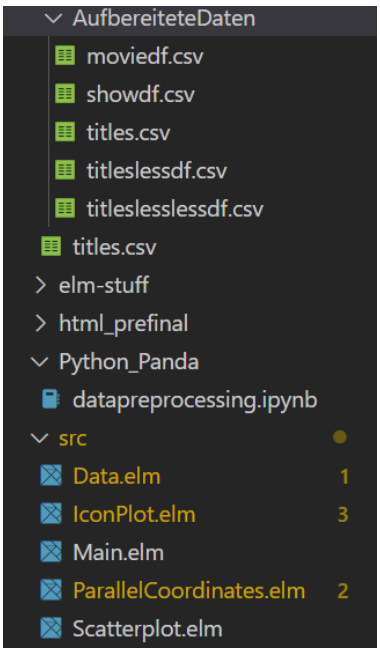

Abbildung 6: Gliederung des Projekts (Quelle: Eigene Darstellung)

Daten wurde zuerst bereinigt durch dem Modul datapreprocessing.ipynb. Werte, die sich nicht visualisieren lassen wie z.B. NaN-Werte (not a number) oder Na-Werte (not assigned) werden entweder gelöscht oder mit 0 ersetzt. Eine neue Spalte (numberTags) wird ebenfalls auch mithilfe dieses Moduls generiert. Eine Funktion wird verwendet, um die Anzahl der „genres“ zu zählen. Diese gelten als die Anzahl des Tags pro Film oder pro Serie. Anschließend wird die Datei wieder als .csv-Datei zurückgegeben.

Im Modul Data.elm werden verschiedene Parameter festgelegt, sodass diese global definiert und somit auch bei den anderen Modul wiederverwendet werden. Diese sind unter anderem der Standardabstand, der Radius von Kreise usw. Das gleiche gilt auch für die Funktionen, die ebenfalls in den anderen Module immer Anwendung findet. Diese sind beispielweise Funktionen, die Datentypen manipulieren, oder die Funktion, die die Länge der Achse automatisch anpassen. Auch bei dem Modul Data.elm werden die Daten aus .csv-Datei eingelesen und dekodiert, sodass diese später auch als Elm-Typen verstanden wurden und somit in der Elm-Umgebung verwendet werden können. Zu guter Letzt werden auch bei diesem Modul Elm-Typen und Elm-Alias-Typen definiert, sodass diese nicht nochmals bei den Visualisierungen definiert werden müssen.

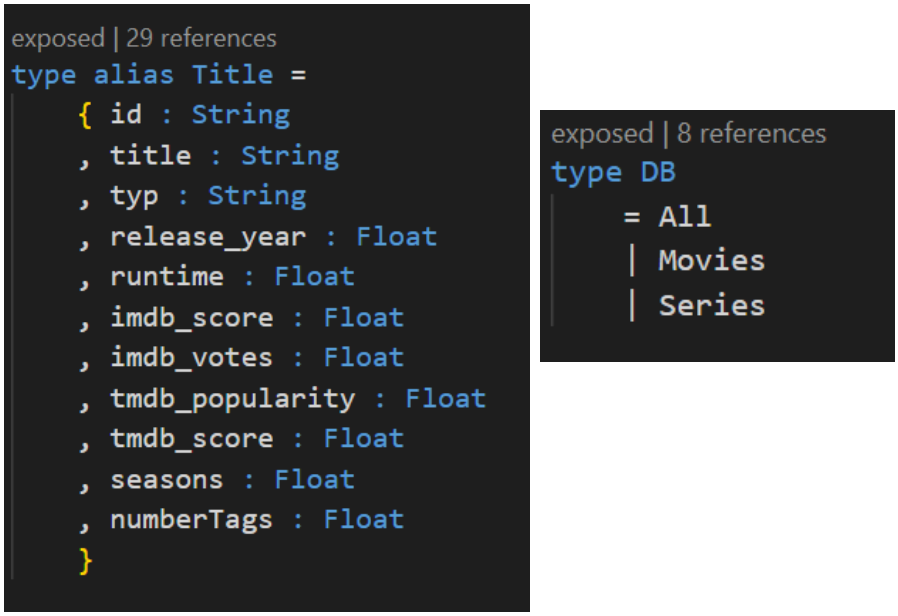

Abbildung 7: Elm-Typen Definition (Quelle: Eigene Darstellung)

4.2. Implementierung Visualisierung Eins

Der Großteil der ersten Visualisierung basiert auf das Quellcode von den Übungsstunden. Da die Anforderung auch darin besteht, dass Filmen und Serien einzeln betrachten lassen, wird ChangeDB-Buttons in allen drei Visualisierungen implementiert. Diese Buttons verleiht den Nutzern die Möglichkeit, zwischen Filmen, Serien bzw. beiden zu wechseln. Die Implementierung dieses Buttons erfolgt durch Namenersetzen der zu visualisierenden Dateien in der Funktion getData, wobei die Namen der csv-Dateien im Data.elm als eine Liste definiert werden.

Weiterhin wird eine Dropdown-Liste in diesem Modul hinzugefügt. Eine Vorteil der Liste gegenüber Button ist, dass die UI-Fläche mit der Liste eingespart werden kann. Die Liste lässt sich zeigen erst beim Betätigen des Dropdown-Pfeils.

Darüber hinaus werden die beim Hovern angezeigten Texte über den Diagramm umgestellt. Diese Änderung sorgt dafür, eine bessere Lesbarkeit des Texts zu gewähren, vor allem wenn sich der betrachtete Punkt in einem dichtbesetzten Bereich befindet. Die Sichtbarkeit des angezeigten Textes wird in diesem Fall stark durch die benachbarten Punkte beeinträchtigt.

4.3. Implementierung Visualisierung Zwei

Basierend auf die erste Visualisierung sowie auf der bearbeiteten Übung wird die Visualisierung Zwei aufgebaut. Um die aktuell gezeigte Achsen zu ändern, werden unterschiedliche Buttongruppe implementiert. Es gibt insgesamt vier Gruppe. Diese gilt jeweils

für eine Achse, sodass sich das bei einer Achse angezeigte Attribut ändern kann, wenn ein Button von derselben Gruppe betätigt wird.

Außerdem wird ein X-Ray-Modus wie bereits erwähnt noch implementiert. Der Diagramm an sich wird inhaltlich nicht geändert, sondern nur deren Eigenschaft. Im diesem Modus werden die Linien in Weiß anstatt in Schwarz dargestellt und die Linienopazität wird zeitgleich auf 0,1 gesetzt. Diese Eigenschaften (stroke, stroke-width, opacity) kann man durch css-style festlegen. Der Hintergrund sollte dabei auch in Schwarz anstatt in Weiß angezeigt. Beide Änderungen werden mit den Buttons verknüpft, sodass sie gleichzeitig geändert wird, wenn man die Buttons betätigt. Schwierigkeit besteht darin, dass sich die Eigenschaften in einer gleichen Stelle befinden, sondern verteilt im Skript. Sie miteinander zu verbinden mithilfe eines Msg (ChangeMode) bei *update* hat bei mir ein bisschen gedauert.

```
[ TypedSvg.style []
    [
        TypedSvg.Core.text """
        .xray {stroke: white; stroke-width:1; opacity:0.1; fill: none}
        .xray text {display: none;}
        .normal { stroke: black; fill:none; opacity: 0.5;stroke-width:0.5}
        .normal:hover {stroke: rgba(4, 244, 251, 1); stroke-width: 2.5}
        .normal text { display: none; }
        .normal:hover text { display: inline; stroke: black; stroke-width: 0.1; font-size: small; font-family: calibri}
        """
    ]
```

Abbildung 8: Änderung der Linieneigenschaften durch css-style (Quelle: Eigene Darstellung)

4.4. Implementierung Visualisierung Drei

Das Quellcode von der ersten Visualisierung wird weiterverwendet für die Implementierung von Visualisierung Drei, da anstelle von Kreisen sollten die Stickfiguren die Datenpunkten repräsentieren. Eine Schwierigkeit besteht darin, dass Stickfigures sollten mithilfe von svg gezeichnet werden und ich mir noch gar keine Erfahrung mit svg gemacht habe. Die Winkel von Körper und Körperteile sollten dabei noch dynamisch sein, d.h. sie ändert sich, wenn die Daten sich ändern. Nach dem mathematischen Hinweis von Herrn Hinneburg wird das Code unter Einsatz von *polyline* zum Erstellen des Stickfigures geschrieben wie in Abbildung 9 zu sehen.

```
[ polyline
        [ TSA.points [ ( lange/2*cos(degrees uDegree), lange/2*sin(degrees uDegree) ),
        ( -lange/2*cos(degrees uDegree), -lange/2*sin(degrees uDegree)) ]
        ][]
, polyline
        [ TSA.points [ ( (-lange/2)*cos(degrees uDegree), (-lange/2)*sin(degrees uDegree) ),
        ( (-lange/2)*cos(degrees uDegree) + lange*cos(degrees vDegree), -lange/2*sin(degrees uDegree) - lange*sin(degrees vDegree) ) ]
        ][]
, polyline
        [ TSA.points [ ( (-lange/2)*cos(degrees uDegree), (-lange/2)*sin(degrees uDegree) ),
        ( -lange/2*cos(degrees uDegree) - lange*cos(degrees pDegree), -lange/2*sin(degrees uDegree) - lange*sin(degrees pDegree) ) ]
        ][]
, polyline
        [ TSA.points [ ( lange/2*cos(degrees uDegree), lange/2*sin(degrees uDegree) ),
        ( lange/2*cos(degrees uDegree) + lange*cos(degrees qDegree), lange/2*sin(degrees uDegree) + lange*sin(degrees qDegree) ) ]
        ][]
, polyline
        [ TSA.points [ ( lange/2*cos(degrees uDegree), lange/2*sin(degrees uDegree) ),
        ( lange/2*cos(degrees uDegree) - lange*cos(degrees zDegree), lange/2*sin(degrees uDegree) + lange*sin(degrees zDegree) ) ]
        ][]
]
```

Abbildung 9: svg-Code für Stickfigures (Quelle: Eigene Darstellung)

Eine andere Schwierigkeiten besteht darin, die Datenwerte mit der Zeichnen-Funktion stickfigure zu mappen. Insgesamt gibt es sechs Variablen, die mit dem Funktion *stickfigure* wie in Abbildung 10 gemappt werden sollen. Eine schnelle Lösung wie List.map6 ist nicht gegeben, sondern nur höchsten bis List.map5. Aufgrund dessen habe ich die Hilfsfunktion namens andMapl verwendet. Das Mapping findet dann sequenziell anstatt gleichzeitig statt durch Nutzen von der Funktion *pipe* (|>). Analog wird auch ein andMap verwendet anstelle von Maybe.map6, was gar nicht existiert.

```
-- plot points and description
    ,g [ transform [ Translate padding padding ] ]
       (List.map (stickfigure xScaleLocal yScaleLocal len)
           uDegree
           |> andMapl vDegree
           |> andMapl pDegree
           |> andMapl qDegree
           |> andMapl zDegree
           |> andMapl model.data


       )
       -- map data with the defined variables
```

```
exposed | 7 references
andMapl : List a -> List (a -> b) -> List b
andMapl = List.map2 (|>)
```

Abbildung 10: andMapl-Funktion und Implementierung (Quelle: Eigene Darstellung)

Weiterhin wird noch Buttons in der Visualisierung hinzugefügt, um die Länge der Linien durch Nutzer anpassen zu lassen. Diese erfolgt durch Hinzufügen eines weiteren Parameter *len* in dem Zeichnen-Funktion. Diese soll die Länge der Linien definieren und wird geändert beim Betätigen des Buttons.

Zum Schluss werden die sechs Buttons in der Visualisierung umgesetzt. Dies erfolgt durch Zuweisung der Linienfarben über If-Schleife beim Anklicken, sodass der angewählte Körperteil mit rot hervorgehoben wurde, während die restlichen komplett ausgeblendet wurden.

4.5. Implementierung der Interaktion

Das Modul Main.elm sorgt für die Interaktionen zwischen den anderen Modulen unter Einsatz von Html. Hypertext wird vor allem verwendet, damit man von einem Modul zu dem anderen Modul, ohne neuzuladen wechseln kann. Bei den anderen Modul wurde ein Hypertext „Back to Homepage“ hinzugefügt, sodass man problemlos zu dem Hauptseite jederzeit zurückkehren kann. In der Hauptseite werden außerdem noch zwei andere Hypertexte dargestellt. Das eine führt zu dem Kaggle-Seite, wo die Daten zu finden ist. Das andere ist eine URL zu diesem Projekt im GitHub.

5. Anwendungsfälle

In diesem Kapitel werden die Visualisierungen durch Anwendungsbeispiele demonstriert.

Einen Zusammenhang zwischen dem Erscheinungsjahr des Filmes und deren Bewertungen könnte durch Nutzung der dritten Visualisierung wie in der Abbildung 11 entdeckt werden. Das Erscheinungsjahr wird in dieser Visualisierung durch den rechten Fuß der Stickfigures angezeigt, wobei dessen Länge für einer besseren Sichtbarkeit auf 9 Pixel angepasst wurde. Die senkrechten Striche repräsentieren in diesem Fall die neuen Filmen. Umso mehr nach Uhrzeigesinn die Striche sich drehen, desto älter sind diese Filme. Die meisten diagonalen Striche befinden sich oben rechts, das heißt, diese gehören meist zu den „guten Filmen“ mit guten Bewertungen.

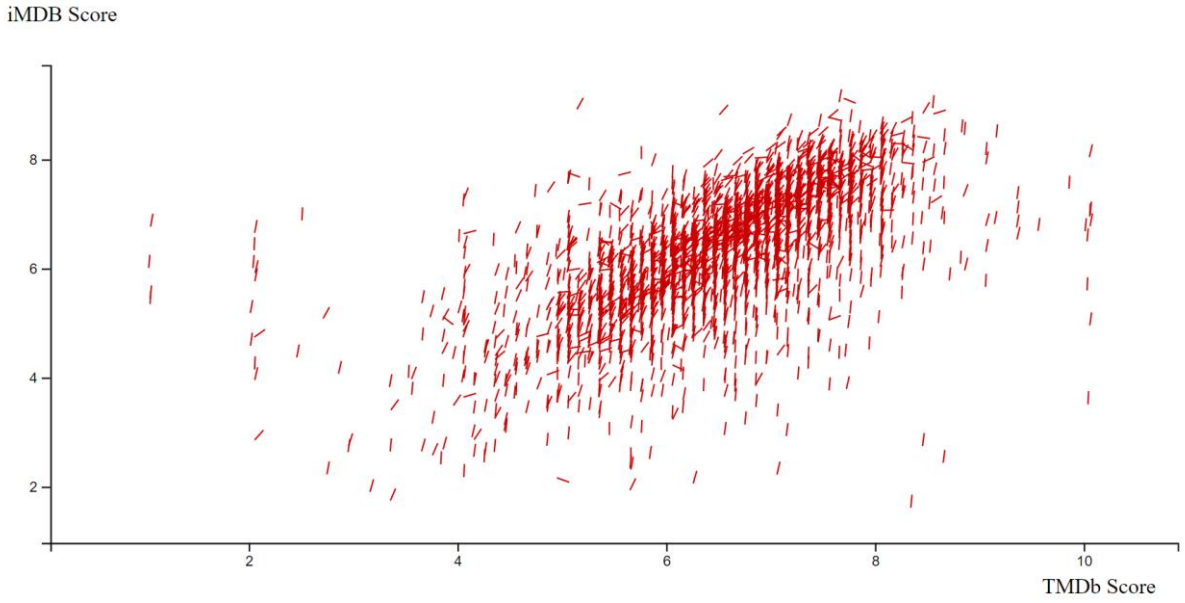

Abbildung 11: Erscheinungsjahr des Filmen wird durch den rechte Fuß dargestellt (Quelle: Eigene Darstellung)

Um diese Erkenntnisse zu prüfen, kommen die ersten zwei Visualisierung zum Einsatz. In der Abbildung 12 werden bei der zweiten Visualisierung die Dimensionen der Attributen IMDb score, Release year bzw. TMDb score nach dieser Reihenfolge durch die erste, zweite bzw. dritte Achse dargestellt. Ergebnisse aus den beiden Modi zeigen, dass ältere Filme sowohl vom Nutzern der IMDb als auch der TMDb oft überdurchschnittlich bewertet werden. Viele neue Filme werden dagegen viel schlechter bewertet aber bleiben trotzdem weiterhin in der Bibliothek von Netflix bestehen.

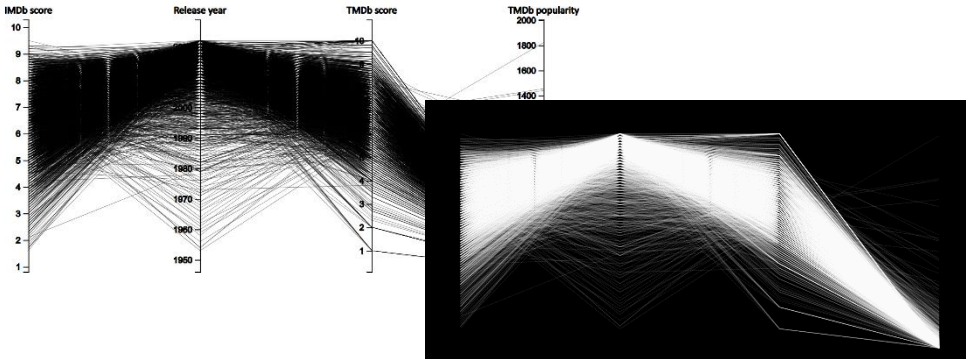

Abbildung 12: Die Zusammenhänge zwischen Erscheinungsjahr und Bewertungen durch Visualisierung Zwei im normalen und im X-Ray-Modus (Quelle: Eigene Darstellung)

Unter Einsatz von der ersten Visualisierung wie in Abbildung 13 lassen sich die Zusammenhänge zwischen Erscheinungsjahr und Bewertungen einzelner Bewertungsdienste darstellen. Ebenfalls aus diesen beiden Ergebnisse (IMDb und TMDb) kann man feststellen,

dass die älteren Filme im Schnitt deutlich besser bewertet werden als ihre Gegenstücke. Die trichterförmige Punktwolke weist darauf hin, dass Netflix nur alte Filme mit überdurchschnittlicher Bewertung anbietet. Netflix und co. sind letztendlich Serviceanbieter, die immer gern mit optimalem Ressourceneinsatz maximalen Gewinn erzielen. In diesem Fall versteht man unter Ressourcen beispielweise Kosten für Filmlizenzen, Speicherplatz beim AWS (Amazon Web Service) sowie Aufwand für Wartungen und Instandhaltungen. Es besteht auch eine Kundengruppe, die gern ältere oder klassische Filme anschauen. Diese entweder gehören zur Minderheit bei Netflix-Community oder sind selbst Senioren Kunden, die in der Regel über konventionelle Kanäle wie TV-Sendungen, DVD oder Blu-ray erreichbar sind. Zumindest hat man eine gute Nachricht für Filmentdecker, und zwar, dass Netflix viele überdurchschnittlich gute Filme besitzt.

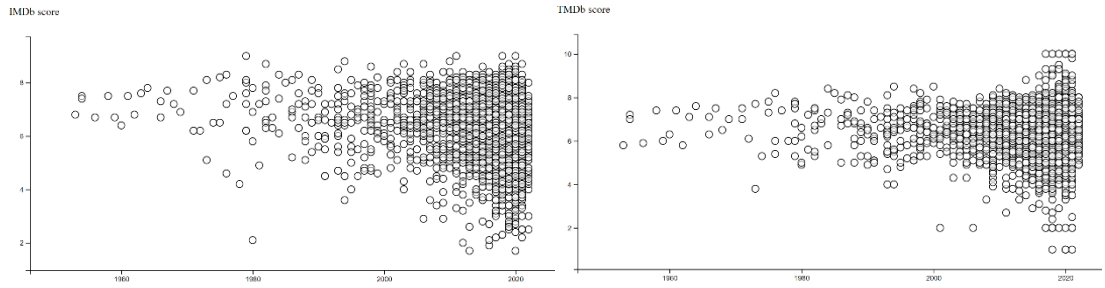

Abbildung 13: Zusammenhänge zwischen Erscheinungsjahr durch Visualisierung Eins (Quelle: Eigene Darstellung)

6. Verwandte Arbeiten

Nach der Literaturrecherche wurden diese drei folgenden, verwandten Artikel hauptsächlich verwendet.

Nguyen et al. schlagt einen Visualisierungsansatz in ihrem Artikel „Evaluation on interactive visualization data with scatterplots“ vor, wobei der Scatterplot durch Nutzer selbst konfiguriert werden kann. Diese Idee wurde in dieser Arbeit umgesetzt, allerdings mit Einschränkungen bei der Paarauswahl. Umgesetzt auf Kriminalitätsdaten der 90 Bezirke in North Carolina, USA, für den Zeitraum von 1981 bis 1987 enthält ihre Visualisierungen noch weitere Features. Zum einen können die Nutzerinnen über die Anzahl der anzuzeigenden Scatterplots entscheiden, was Ihnen beim Entdecken neuer Erkenntnisse die Freiheit gewährt, unterschiedliche Zusammenhänge gleichzeitig darzustellen. Zum anderen bietet die sogenannte *linkable scatterplot* mehrere Möglichkeiten, mit den Informationen zu interagieren, inklusiv Hervorheben, Filtern, Auszuwählen, Verknüpfen und Bürsten, Änderung der visuellen Eigenschaften oder Anzeigen von Regressionslinien [4].

Yuan et al. verwendet in ihrem Artikel „Scattering Points in Parallel Coordinates“ die Technik Parallelkoordinate und schlagt einen Ansatz des sogenannten SPPC (*Scattering Points in Parallel Coordinates*) vor. Diese Methode ist eine hybride Lösung zwischen Scatterplot und Parallelkoordinate, wobei die Punkte in das Parallelkoordinate integriert werden [5]. Weiterhin werden die Linien zwischen adjazente Achsen mit freiförmigen Kurven ersetzt. Der Vorteil von Kurven gegenüber Linien besteht darin, dass das menschliche Auge auf kleine Veränderungen ihrer Form recht empfindlich reagiert [6], sodass man nach diesem Ansatz mehr Informationen gewinnen kann. Interessanterweise demonstrieren die Autoren ihren Ansatz durch Visualisierungen der Autodaten, was wir ebenfalls in der Übungsstunde verwendet haben.

Für Iconplot wird der Artikel „Iconographic displays for visualizing multidimensional data“ von Pickett et al. referenziert [7]. Dieser Artikel werden schon bereits in der Vorlesungsstunde vorgestellt. Es handelt sich bei der Visualisierung Drei im Grunde genommen um einen Versuch, die Grundidee dieses Artikels zu realisieren. Dabei wird die einfachste Variante der Stickfigurefamilie verwendet, welches aus einen Körpern und vier Gliedern besteht.

7. Zusammenfassung und Ausblick

In diesem Bericht wurden drei unterschiedliche Visualisierungstechniken vorgestellt und anhand der Netflix-Daten demonstriert.

Als erstes wurde in der ersten Visualisierung die Technik Scatterplot präsentiert. Diese Technik wurde eingesetzt vor allem, um die Daten paarweise zu visualisieren und deren Zusammenhängen damit zu entdecken bzw. zu validieren.

In der zweiten Visualisierung kamen die Parallelkoordinaten zum Einsatz. Es handelt sich hierbei um eine mehrdimensionale Repräsentation der Datenpunkten, wobei sich die Achsen anpassen lassen. Dies bietet Nutzern die Möglichkeit, bis zu vier Attribute gleichzeitig zu beobachten.

Bei der dritten Visualisierung geht es um die Implementierung von den Iconplot unter Einsatz von der einfachsten Variante der Stickfigurenfamilie. Der Vorteil dieses Ansatzes ist, dass er den Nutzern einen Überblick über mehrere Attributen sowie über die Datenstruktur bietet. Iconplot ist vor allem geeignet für Entdecker, um neue Erkenntnisse zu gewinnen.

Weiterhin wurden auch in allen drei Visualisierungen benutzerfreundliche Steuerungsmöglichkeiten implementiert. Diese sind unter anderem der Wechsel von Attributen, die Interaktionen mit Punkten, Linien und mit Stickfiguren beim Hovern, die

Anpassungsmöglichkeit der Linien und der Stickfiguren sowie Ein- und Ausschalten eines Modus.

Die Anwendung der drei Visualisierungen wurden demnächst durch ein Anwendungsbeispiel demonstriert. Ergebnisse zeigen, dass alte Filmen von Netflix oft überdurchschnittlich bewertet werden, im Vergleich zu deren neueren Gegenstücken.

Andere Visualisierungstechniken wie pixelorientierte Darstellung oder Baumdiagramme finden hier auf jeden Fall Anwendung und könnten auch in dieser auftauchen. Da laut Anforderung der Veranstaltung die Anzahl der Visualisierung auf drei beschränkt wurden, wurden die drei Techniken ausgewählt und implementiert. Die Visualisierungen lassen sich außerdem noch erweitern, um mehr Erkenntnisse zu gewinnen. Als Beispiel könnte bei der zweiten Visualisierung die Linien mit frei förmigen Kurven ersetzt werden, wie bei Yuan et al. Denkbar wäre auch die Möglichkeit, dass Punkte, Linien und Stickfiguren miteinander interagieren könnten.

8. Anhang: Git-Historie

* commit 109150f896ff0d46117938163eef2672ab46c871 (HEAD -> main, origin/main, origin/HEAD)

| Author: do46 <100580004+do46@users.noreply.github.com>

| Date: Wed Dec 14 14:57:10 2022 +0100

|

| Update README.md

|

* commit 66c40f24bd174d5719c677e47732389c4a02dc9a

| Author: do46 <100580004+do46@users.noreply.github.com>

| Date: Wed Dec 14 14:54:42 2022 +0100

|

| Update README.md

|

* commit 722f0252438c0f40baff6205f7edf93eaad8b442

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Dec 12 15:46:45 2022 +0100

|

| Add features to Iconplot

|

*   commit 2f833422d6a8a50fbbde4d0ce106b3e7a5c6803b

|\ Merge: 235c9d8 aee16bd

| | Author: do46 <tyto4694@gmail.com>

| | Date: Mon Oct 10 15:38:33 2022 +0200

| |

| | Merge branch 'main' of https://github.com/do46/IR_Netflix_TVShows_Movies_VAD

| |

| * commit aee16bd4252240d8d8b3281420a70a7b4400264a

| | Author: do46 <100580004+do46@users.noreply.github.com>

| | Date: Wed Jul 27 14:32:02 2022 +0200

| |

| | Update README.md

| |

* | commit 235c9d8a23423b048396eb46479ea6c8b6c3035a

|/ Author: do46 <tyto4694@gmail.com>

| Date: Mon Oct 10 15:38:25 2022 +0200

|

| Update Main, html, interaction

|

* commit 8d30f1896ae3d28f0efaf376d27c11671c0e3de6

| Author: do46 <tyto4694@gmail.com>

| Date: Wed Jul 27 14:27:16 2022 +0200

|

| Prototype

|

* commit 955b3a82df9f5a57da898a7728ab27cbb2f149e6

| Author: do46 <tyto4694@gmail.com>

| Date: Tue Jul 26 19:09:29 2022 +0200

|

| Prototype ready!!!

|

* commit b5f0569501da78c5417f4f2daa7abdde0f4c7df1

| Author: do46 <tyto4694@gmail.com>

| Date: Sat Jul 23 14:35:35 2022 +0200

|

| Iconplot ready

|

* commit ba02bf88460996889be8d5c5668e208c9df03af3

| Author: do46 <tyto4694@gmail.com>

| Date: Sat Jul 23 13:10:52 2022 +0200

|

| add attributes, template for icon, sctp and parcoord prototype

|

* commit 37263030d52bd048d556fabecbc3c2ecff758bad

| Author: do46 <tyto4694@gmail.com>

| Date: Wed Jul 20 22:27:41 2022 +0200

|

| Scatterplot fertig

|

* commit fe89803aab75d24a9b29acca17167739ecc5cab3

| Author: do46 <tyto4694@gmail.com>

| Date: Tue Jul 19 12:53:04 2022 +0200

|

| Update d.dat

|

```
* commit b60373961c9138d7c2a2fe9e151ce3dd94af7277
| Author: do46 <tyto4694@gmail.com>
| Date:   Tue Jul 19 12:49:58 2022 +0200
|
|     add seasons
|
* commit fba4ea8a882ad32ea699f52cec049bb710caa98a
| Author: do46 <tyto4694@gmail.com>
| Date:   Mon Jul 18 15:32:47 2022 +0200
|
|     endlich funktionieren
|
* commit 3bf7f996b59add71be205215e3e4a21e30471a82
| Author: do46 <tyto4694@gmail.com>
| Date:   Mon Jul 18 14:53:15 2022 +0200
|
|     remove index column
|
* commit 127103f7ff5b50fe7b8c8bcab106d53fa5d717ee
| Author: do46 <tyto4694@gmail.com>
| Date:   Mon Jul 18 14:37:12 2022 +0200
|
|     filtered movie and show
|
* commit 74802b8dfa7e4dc6a7b57843188384a498311176
```

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Jul 18 10:35:20 2022 +0200

|

| nochmal test

|

* commit 24ae8936787cd1567e8e1e5e2021c42825b73da6

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Jul 18 10:32:56 2022 +0200

|

| test

|

* commit 67610a50ea9b2f1a1522e4520c59b29dac56d298

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Jul 18 10:22:36 2022 +0200

|

| Update d.dat

|

* commit 9db433fe1034d1f888005ebcf77f63d9d0a19f7c

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Jul 18 10:21:20 2022 +0200

|

| nan eliminate

|

* commit 3f4734f8e8b49016fa4270578ab90a056d2311ba

| Author: do46 <tyto4694@gmail.com>

| Date: Mon Jul 18 10:19:03 2022 +0200

|

| endlich funktioniert

|

* commit d8d775d8ad62cb3c728e1562bd8251e1e4f298cd

| Author: do46 <tyto4694@gmail.com>

| Date: Sun Jul 17 13:30:55 2022 +0200

|

| Update titleslesslessdf.csv

|

* commit 9dcb73db35c615af86c5124f951709ad1eb6c0fa

| Author: do46 <tyto4694@gmail.com>

| Date: Sun Jul 17 13:29:14 2022 +0200

|

| str

|

* commit 24e6ec2ea29fd75bdd03a5f0b464d5a443c02331

| Author: do46 <tyto4694@gmail.com>

| Date: Sun Jul 17 13:18:01 2022 +0200

|

| nan eliminated

|

* commit 531e4adec995bd591f46365ae942e2a8e4e67913

| Author: do46 <tyto4694@gmail.com>

| Date: Fri Jul 15 17:58:34 2022 +0200

```
|
|     nan eliminated
|
* commit 75cad2be9289bbe5a8b421a3d1aeb3580924763a
| Author: do46 <tyto4694@gmail.com>
| Date:   Fri Jul 15 17:22:56 2022 +0200
|
|     test again ...
|
*   commit 4953d5a89558792a44357a1524e2ead27f8183c8
|\  Merge: d0c0262 61e5e5e
| | Author: do46 <tyto4694@gmail.com>
| | Date:   Fri Jul 15 17:02:48 2022 +0200
| |
| |     Merge branch 'main' of https://github.com/do46/IR_Netflix_TVShows_Movies_VAD
| |
| * commit 61e5e5ec023f2a23a3310a216f6d2fc528534621
| | Author: do46 <100580004+do46@users.noreply.github.com>
| | Date:   Fri Jul 15 16:58:47 2022 +0200
| |
| |     with comma
| |
* | commit d0c0262c265a75d850bad5595c2580233ff53c8a
|/  Author: do46 <tyto4694@gmail.com>
|   Date:   Fri Jul 15 17:02:29 2022 +0200
```

```
|
|       Create titleslessdf1.csv
|
* commit 945fb8bb75137883f993f55efd015be9017115d5
| Author: do46 <tyto4694@gmail.com>
| Date:   Fri Jul 15 16:57:17 2022 +0200
|
|     Update d.dat
|
* commit cbe03f5b03c0863750d441d6527adc4a5197cb0c
| Author: do46 <tyto4694@gmail.com>
| Date:   Fri Jul 15 16:54:15 2022 +0200
|
|     csv is csv ...
|
*   commit 704447882d2dd94e63ce5b3cf7d12573674a3934
|\  Merge: 2a6d9ed de7e4ed
| | Author: do46 <tyto4694@gmail.com>
| | Date:   Fri Jul 15 16:50:22 2022 +0200
| |
| |     Update titleslessdf.csv
| |
| * commit de7e4edae1c9cac384843b8e2439995d19fe83af
| | Author: do46 <100580004+do46@users.noreply.github.com>
| | Date:   Fri Jul 15 15:58:07 2022 +0200
```

```
| |
| |     Update titleslessdf.csv
| |
* | commit 2a6d9edfa909e45d09101e7bd71fb446bcb1ca64
|/  Author: do46 <tyto4694@gmail.com>
|   Date:   Fri Jul 15 16:44:00 2022 +0200
|
|       add quote
|
* commit 88ee97725ecba44940d6af5ff11cbc2cbbdee4cb
| Author: do46 <tyto4694@gmail.com>
| Date:   Fri Jul 15 15:51:03 2022 +0200
|
|     Type changed to str
|
* commit e0f38989a024690865a44f23b4d0cb6151ec5401
| Author: do46 <tyto4694@gmail.com>
| Date:   Thu Jul 14 18:28:15 2022 +0200
|
|     Anpassung
|
* commit dba05d8e4606fbae8024dfa62b2e67d62097ebb8
| Author: do46 <tyto4694@gmail.com>
| Date:   Thu Jul 14 17:40:12 2022 +0200
|
```

```
|     csv bearbeitet
|
* commit a8a9651d909382e1e075671e20583942fed5b389
| Author: do46 <tyto4694@gmail.com>
| Date:   Thu Jul 14 13:49:20 2022 +0200
|
|     Daten
|
* commit 3569e3c0708bd7e1c65151c41598bd98860d634d
| Author: do46 <tyto4694@gmail.com>
| Date:   Thu Jul 14 11:27:30 2022 +0200
|
|     csv bearbeitet
|
* commit 9aa83f5ea9730e107e9fd417f5ea242be2547900
| Author: do46 <tyto4694@gmail.com>
| Date:   Wed Jul 13 16:04:41 2022 +0200
|
|     Template from African War repo
|
* commit e3c29178c9852ace2510da770174bcd95544b8e7
  Author: do46 <100580004+do46@users.noreply.github.com>
  Date:   Wed Jul 13 16:01:48 2022 +0200

      Initial commit
```

Literaturverzeichnis

[1] IMDb.com Help Center. Ratings FAQ. https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV#.

[2] Rabe L (2022) Anzahl der zahlenden Streaming-Abonnenten von Netflix weltweit vom 3. Quartal 2011 bis zum 2. Quartal 2022. https://de.statista.com/statistik/daten/studie/196642/umfrage/abonnenten-von-netflix-quartalszahlen/. Zugegriffen: 28. Juli 2022.

[3] themoviedb.org Getting Started. https://developers.themoviedb.org/3/getting-started/popularity.

[4] Nguyen, Q. V., Miller, N., Arness, D., Huang, W., Huang, M. L., & Simoff, S. (2020). Evaluation on interactive visualization data with scatterplots. Visual Informatics, 4(4), 1-10.

[5] Yuan, X., Guo, P., Xiao, H., Zhou, H., & Qu, H. (2009). Scattering points in parallel coordinates. IEEE Transactions on Visualization and Computer Graphics, 15(6), 1001-1008.

[6] Theisel, H. (2000, November). Higher order parallel coordinates. In VMV (pp. 415-420).

[7] Pickett, R. M., & Grinstein, G. G. (1988, August). Iconographic displays for visualizing multidimensional data. In Proceedings of the 1988 IEEE Conference on Systems, Man, and Cybernetics (Vol. 514, p. 519).