**國立中正大學**

資訊管理研究所

碩士論文

結合時間因素與文字探勘技術建立電影推薦系統

Constructing a movie recommendation system by combining time factors and text mining techniques

|  |  |
| --- | --- |
| 研究生 : 史博先 | 撰 |
| 指導教授 : 李珮如 | 博士 |
| 胡雅涵 | 博士 |

中華民國 一一零 年 六 月

**誌謝**

在中正就讀的這兩年，學習到很多關於資料分析及探勘之技術與實務知識，以及認識了許多導師與朋友。首先，非常感謝我的指導教授李珮如博士與胡雅涵博士，於中正的這兩年期間給予我不管是學術上的指導或是實習上的建議，讓我在學術與實務方面都有扎實的基礎，也感謝林勝為博士與許經國博士，給予論文學術和技術上相關的建議，使我的碩士論文能更加完整。

感謝學長姐妤庭、俐麟、育愷、亭儀、凡君、瑜真、沛蓁、翼張，於碩一時的照顧，讓接觸新還進的我們快速融入校園生活，以及分享課程與實習相關的經驗。也感謝實驗室的同學玲蓉、貞樺、庭維、程傑、卓庭、惠婷、雨琦，不管是在論文的撰寫或是實習工作都給予有用的意見，讓這兩年的碩士生活充滿愉快的回憶。

最後，感謝我的父母，給我在求學與生活方面的建議與支持，也感謝所有資管所同學的陪伴，在這短暫的碩士生活充實而快樂，謝謝你們。

史博先 謹致

中正大學資訊管理研究所

中華民國一一零年六月

**摘要**

背景與動機：隨著社會進入高度資訊化的時代，人們使用網路來獲取資料已是生活中的一部份，伴隨著資訊量的快速增長，資訊過載的問題也成了學者們嘗試解決的問題。而作為解決資訊過載的方法之一，推薦系統透過蒐集使用者或物品的相關資訊，協助使用者有效的過濾不必要的資訊，從而解決資訊怪仔的問題。而電影推薦系統為推薦系統研究中的熱門領域，並且隨著文字探勘技術與網路評論的普及，除了電影類別或評分等數據之外，使用者評論文本內容亦可作為描述使用者喜好的重要資訊。因此，本研究使用網路電影資料庫之使用者評論文本，並考量了時間對資訊價值的影響，建立混合式電影推薦系統並加以驗證其效果。希望透過本研究開發之推薦系統提供使用者與企業預測模型，使其能過透此系統準確且有效的預測使用者喜好。

方法：本研究以IMDb網路電影資料庫Feature Film類別2010年至2020年熱門電影為研究資料，共60部熱門電影及836名使用者，以電影的基本資料、劇情關鍵字與使用者評論文本進行分析，將使用者評論文本關鍵字分類至電影不同面相特徵進行計算，透過KeyBERT與NGD實現文本處理，並以不同資料組合對使用者 – 電影評分矩陣補值進行補值，使用Time SVD、Funk SVD與協同過濾來建立預測模型。本研究所使用的工具包含Python3之自然語言處理工具和KeyBERT套件。

結果：加入時間因素的考量於實驗結果中相較於未加入者皆有更佳的表現，而資料組合中採用電影相關資料與使用者評論文本資料進行補值，在不同模型與參數配置下皆獲得最佳的結果。在考量時間因素與採用電影相關資料與使用者評論文本資料組合的情況下，推薦模型獲得之最低RMSE為0.922185，最低MAE為0.515967。

結論：基於本研究之實驗結果，發現（1）加入對時間的考量，推薦模型能更有效的捕捉使用者興趣的改變;（2）不同的捕值方式可能會改變時間資料的分布，從而影響模型預測準確度;（3）使用電影相關資料與使用者評論文本資料，能使推薦模型有效的捕捉使用者與電影在不同面向的特徵。

**關鍵詞**：推薦系統、電影推薦、矩陣分解、自然語言處理、文字探勘、機器學習、KeyBERT

**Abstract**

Background：As living in a society with a highly developed information technology, people use the Internet to obtain information become parts of daily life. With the rapid increase in data volume, the problem of information overload become has become a problem that the scholars are trying to solve. As one of the solution of information overload, recommendation system helps users filter unwanted information via the data from user or item. As the popular field in the research of recommendation system and the popularity of text mining technology and online reviews. Movie recommendation system can use not only the basic data of movie itself, but also the user’s review which content the information describing user’s preferences. Therefore, we apply the user’s review on IMDb website and the consideration of the effect of time. We establish a hybrid movie recommendation system that predict the user’s preference during the different time period. The aims of the recommendation system developed by this study is providing user and enterprise to Predict user preferences accurately and effectively.

Method：We focused the Feature Film data on IMDb website with a study period from 2010 to 2020. There are 60 movies and 836 users which selected according popularity. The dataset includes movie’s basic data, plot keyword and user’s review. The user’s review are used for calculating the user’s degree of preference of the movie feature and implement by KeyBERT and NGD. The next step is to make the imputation of the user-movie rating matrix and build the predict models by Time SVD, Funk SVD and Collaborative Filtering. The tools applied in this study are Natural Language Toolkit Kit and KeyBERT.

Results：We found that the model which takes into consideration of time effect had a better performance compares to others. For the different combination of data matrix, we found that the combination of movie data and user’s review had the best performance in different hyperparameter configurations. The best RMSE of predict model was 0.922185, and the best MAE was 0.515967.

Conclusion：Based on the findings, we conclude that (1) with the consideration of time, the recommendation model can more effectively capture changes in user interests; (2) different ways of capturing values ​​may change the distribution of time data, thereby affecting the accuracy of model predictions; (3) with the combination of movie data and user review, the recommendation model can effectively capture the different characteristics of users and movies.

**Keyword**：Recommendation System, Movie Recommendation System, Matrix Factorization, Natural Language Processing, Text Mining, Machine Learning, KeyBERT

目錄

[圖目錄 iii](#_Toc60187088)

[表目錄 iv](#_Toc60187089)

[第壹章 緒論 1](#_Toc60187090)

[1.1 研究背景 1](#_Toc60187091)

[1.2 研究動機 3](#_Toc60187092)

[1.3 研究目的 5](#_Toc60187093)

[1.4 預期結果與貢獻 6](#_Toc60187094)

[第貳章 文獻探討 7](#_Toc60187095)

[2.1 傳統電影推薦系統 7](#_Toc60187097)

[2.2 情感分析應用於電影推薦系統 9](#_Toc60187098)

[2.2.1 情感分析 9](#_Toc60187099)

[2.2.2 使用情感分析於電影推薦的過往研究 11](#_Toc60187100)

[2.3 時間權重應用於電影推薦系統 12](#_Toc60187101)

[第參章 研究方法 16](#_Toc60187102)

[3.1 研究流程 16](#_Toc60187104)

[3.2 資料收集與說明 17](#_Toc60187105)

[3.3 資料前處理 20](#_Toc60187106)

[3.3.1 評論篩選 20](#_Toc60187107)

[3.3.2 文字前處理 21](#_Toc60187108)

[3.4 矩陣建立 24](#_Toc60187109)

[3.4.1 電影資料矩陣 24](#_Toc60187110)

[3.4.2 定義評論特徵 26](#_Toc60187111)

[3.4.3 TF-IDF 27](#_Toc60187112)

[3.4.4 NGD 28](#_Toc60187113)

[3.4.5 情感分析 29](#_Toc60187114)

[3.4.6 評論時間 30](#_Toc60187115)

[3.5 資料補值 32](#_Toc60187116)

[3.5.1 相似度計算 32](#_Toc60187117)

[3.5.2 使用者 – 電影評分矩陣補值 34](#_Toc60187118)

[3.6 資料探勘技術 35](#_Toc60187119)

[3.6.1 Funk SVD 35](#_Toc60187120)

[3.6.2 Time SVD 36](#_Toc60187121)

[3.6.3 協同過濾 37](#_Toc60187122)

[3.6.4 產生推薦結果 38](#_Toc60187123)

[第肆章 實驗方法與設計 39](#_Toc60187124)

[4.1 實驗建構 39](#_Toc60187126)

[4.2 資料驗證及評估指標 40](#_Toc60187127)

[第伍章 研究計劃 42](#_Toc60187128)

[5.1 目前進度與未來規劃 42](#_Toc60187130)

[5.2 可能遭遇之困難與應變計劃 42](#_Toc60187131)

[參考文獻 44](#_Toc60187132)

# 圖目錄

[圖 3‑1: 研究架構圖 17](#_Toc60186810)

[圖 3‑2: IMDb Feature Film頁面 18](#_Toc60186811)

[圖 3‑3: IMDb使用者評論頁面 19](#_Toc60186812)

[圖 3‑4: IMDb有用評論投票頁面 20](#_Toc60186813)

[圖 3‑5: IMDb劇情關鍵字頁面 25](#_Toc60186814)

[圖 3‑6: IMDb電影基本資料頁面 26](#_Toc60186815)

[圖 3‑7: 眾數補值示意圖 31](#_Toc60186816)

[圖 3‑8:上映日期 + 7日補值示意圖 31](#_Toc60186817)

[圖 3‑9:使用者 – 電影評分矩陣補值示意圖 34](#_Toc60186818)

[圖 4‑1: 實驗一 39](#_Toc60186819)

[圖 4‑2: 實驗二 40](#_Toc60186820)

[圖 4‑3: 實驗三 40](#_Toc60186821)

[圖 5‑1: 已完成之工作及未來工作規畫圖 42](#_Toc60186822)

# 表目錄

[表 2‑1: 電影推薦系統相關文獻整理 14](#_Toc60186823)

[表 3‑1:電影相關資料說明 19](#_Toc60186824)

[表 3‑2:使用者評論資料說明 20](#_Toc60186825)

[表 3‑3: NLTK停用詞表 23](#_Toc60186826)

[表 3‑4: 電影–關鍵字矩陣 25](#_Toc60186827)

[表 3‑5: 電影 – 基本資料矩陣 26](#_Toc60186828)

[表 3‑6:評論特徵種子字列表 27](#_Toc60186829)

[表 3‑7:電影 – 評論特徵矩陣 30](#_Toc60186830)

[表 3‑8: 電影特徵相似度矩陣表 33](#_Toc60186831)

[表 3‑9: 使用者 – 使用者相似度矩陣 38](#_Toc60186832)

# 

# 緒論

## 研究背景

隨著資訊科技的蓬勃發展，現今社會已進入了高度資訊化的時代，網際網路 (Internet) 已經是現代人生活中不可或缺的一部份，並且徹底改變了人們的生活方式，人們不管是在娛樂上或是購物上，都不免使用網路來獲取資料。尤其是在進入Web2.0時代後，隨著社群媒體的異軍突起，使用者既是資訊的消費者，也是資訊的生產者，網路上的資訊量呈指數級增長，而資訊過載的問題則在此時代背景下隨之而來 (Bawden & Robinson, 2009; Hemp, 2009; Soni et al., 2017)。

資訊過載指的是由於資訊過多，遠超出個人的資訊需求、資訊處理和資訊利用能力，以至於無法有效且準確的挑選與運用資訊的情況 (Schneider, 1987; Schick et al., 1990; Edmunds & Morris, 2000)。對使用者來說，對資訊的反應速度遠低於資訊傳播增長的速度，且大量無關的沒用資訊嚴重干擾對資訊的選擇。而對企業來說，如何挖掘潛在使用者以及呈現適合的商品，從而提高營收以及促進經濟成長，成為首要問題。

目前針對資訊過載問題的解決方法普遍有兩種，其一是以搜尋引擎 (Search Engine)為代表的資訊檢索系統，如Google、Yahoo和百度等，透過使用者輸入的關鍵字，系統在後台進行匹配，將與使用者搜尋相關的資訊展示給使用者，但是，若使用者無法精準的描述自身的需求，搜尋引擎將無法有效地提供資訊 (Berghel, 1997; Liang et al., 2006)。

而另一種方法則是以推薦系統 (Recommendation System)為代表的資訊過濾系統，在過載的資訊中找出感興趣的資訊與讓他人注意到自身資訊是件困難的事，而推薦系統良好的解決了上述問題 (Maes, 1995; Aljukhadar et al., 2010; Aljukhadar et al., 2012; Huang et al., 2012)。推薦系統蒐集使用者的資訊需求、歷史資料甚至是與系統互動的隱性特徵，透過資料探勘算法對使用者興趣進行建模，預測出使用者對於給定物品的評分或喜好，從而準確地推薦給使用者其可能會喜歡的物品，以解決資訊過載問題 (Costa & Macedo, 2013; Isinkaye et al., 2015)。而與傳統搜尋引擎不同的是，推薦系統無需使用者提供明確的需求，而是透過系統來發掘使用者的興趣，並且引導使用者發現自身的資訊需求。一個良好的推薦系統不僅能為用戶提供個人化服務，甚至還能與使用者之間建立密切關係，讓使用者對該推薦系統產生依賴。

推薦系統較為典型的應用是在 Business to Customer (B2C) 電子商務領域，擁有良好的發展與應用前景，如大型的電子商務系統Amazon、eBay等，當中Amazon甚至投入高達十年來研究電子商務的推薦系統，透過使用者願望清單、瀏覽紀錄等資料，搭配雲端平台儲存管理與計算，讓推薦準確度幾乎到使用者下單前就出貨的程度。其他領域包括各種影音串流平台如YouTube、Spotify等，皆有在不同程度的使用了各種形式的推薦系統。

而目前推薦系統在電影相關的線上平台如IMDB、Netflix和愛奇藝佔有一席重要地位 (Jugovac & Jannach, 2017; Wang & Lobato, 2019)。電影領域與電商平台一樣品項繁多，使用者無法觀看或購買每部電影，甚至許多電影使用者連名稱都不知道，而線上電影平台則是透過推薦系統推薦使用者有興趣的電影來增加平台收益 (Azaria et al., 2013)。綜上所述，目前推薦系統在電影相關領域得到了廣泛的應用，而本研究將以電影推薦系統為研究主題。

## 研究動機

電影推薦系統是推薦系統研究中的熱門領域，近幾年來也有許多研究證明電影推薦系統能對使用者提供有效的選擇建議 (Lekakos et al., 2008; Rajarajeswari et al., 2019)。而推薦系統的可簡單分為基於內容推薦 (Content-Based Recommendation, CBR)、協同過濾 (Collaborative Filtering, CF)、混合推薦 (Hybrid Recommendation)等類型 (Li et al., 2019；M.-H. Chen et al., 2015；Reddy et al., 2019；Subramaniyaswamy et al., 2017；Uluyagmur et al., 2012； Wei et al., 2016)。

過去基於內容推薦的研究，有學者採用劇情導向的方式，透過主題模型，為使用者推薦與過去觀看紀錄劇情相似的電影 (Bergamaschi & Po, 2014)。也有學者透過分析顯性特徵如電影類別、導演和演員特徵，或是隱性特徵如使用者觀看時間等方式，為使用者更準確的推薦電影 (Uluyagmur et al., 2012; Reddy et al., 2019)。然而基於內容的推薦存在著一些缺點，此種推薦方式需將內容抽取成有意義的特徵，因此通常需要對資料進行複雜的前處理來獲取能代表物品的特徵，且無法獲取使用者不熟悉但具有潛在興趣的物品 (Sharma et al., 2013)。

而過去在協同過濾的相關研究，基於使用者 (User-Based)與基於物品 (Item-Based)的協同過濾為較常見的方式，透過計算使用者或電影之間的相似性來做出推薦 (Suganeshwari & Ibrahim, 2016)。另外，近期的則有學者使用交替最小平方法 (Alternating Least Squares)，並且用矩陣分解的方式使用協同過濾來做電影推薦 (Aljunid & Manjaiah, 2019)。或是透過混合電影特徵與使用者長短期特徵，讓系統更能適應使用者的變化 (Li et al., 2018)。但協同過濾有個較為明顯的缺點，實際上使用者評價過的物品是有限的，以至於使用者對電影的評價矩陣會十分稀疏，因此基於使用者評價所得到的使用者之間的相似性便無法達到較高的準確度，此為協同過濾的稀疏性問題 (Data Sparsity Problem) (Grčar et al., 2005; Hwangbo & Kim, 2017)。

各種推薦方法皆存在優缺點，因此目前多數研究都採用混合式推薦來彌補模型中某種技術存在的缺陷，組合的方法有很多種，在協同過濾中加入基於內容過濾，在基於內容過濾中加入協同過濾或是將協同過濾與基於內容過濾整合到同個推薦系統中，其目的便是截長補短，期望透過多個推薦算法協同合作，避免單個算法存在的問題，更準確地為使用者推薦 (Soni et al., 2017)。

混合式推薦系統依據算法的組合方式可分為整合多種推薦算法到同一個算法中的單體 (Monolithic)混合式 (Walek & Fojtik, 2020)、將多個算法的推薦結果融合生成最終推薦的並行 (Parallelized)混合式 (Tian et al., 2019)與將一個算法的結果當作另一個算法的輸入的流水線 (Pipelined)混合式 (Oh et al., 2019)。而最常見的混合算法為協同過濾加上基於內容推薦，以文字探勘或增加參數等方式補齊協同過濾中的缺失值 (Shah et al., 2017)。使用者的評價通常分為評分與評論兩種形式，評分為使用者對物品給出的等級分數，使用較為方便、簡單；而評論則是用以描述使用者對物品的情感態度與認同度，多由文句或文章組成，能較完整的表達使用者的感受 (L. Chen et al., 2015)。過去學者曾透過文本訊息試圖找出使用者對物品表達之想法與情感態度。透過在協同過濾中加入情感分析 (Sentiment Analysis)，增加社群評論等參數，可有效降低資料稀疏與冷啟動問題對於推薦效能所造成的影響 (Ibrahim et al., 2019)。

另外傳統的協同過濾推薦系統忽略了資訊的價值會隨著時間衰減對推薦質量的影響，這就是資訊過期的問題(Bakir, 2018)。本質上使用者的興趣是會隨著時間的推移產生改變，並非保持不變的。例如使用者A在三年前喜歡科幻類型的電影，為電影《蜘蛛人：返校日》打出五分的評分。但隨著時間的推移，他開始更偏好青春戀愛類型的電影，也為電影《可不可以，你也剛好喜歡我》給出五分的評分。如果我們為現在的使用者A給予推薦時不加入使用者興趣會隨時間改變的考量，兩部電影的權重會在推薦的過程中一樣，因此會以相同的機率被推薦，這違背了目前使用者A對青春戀愛電影偏好程度更高的事實，導致推薦效果不夠準確，使用者體驗變差。因此，在推薦系統中加入對時間的考量能有效的增加推薦的質量與準確度 (Zhang & Mao, 2019)。

而過去研究未同時考量評論內容與時間權重的影響於電影推薦系統中，且多數只使用電影類別或使用者評分等特徵進行預測，考量的特徵還有增加的空間 (Koren, 2009; Bansal et al., 2016)。基於以上考量，本研究提出一個混合式電影推薦系統，透過加入時間權重概念來應對使用者隨時間改變的喜好，並利用情感分析找出使用者評論中使用者對電影潛在特徵的喜好，再結合協同過濾所獲得的使用者評價，形成混合式推薦方法，預期解決推薦系統中資料稀疏問題所造成的影響，並增加推薦的準確度。

## 研究目的

根據上述的研究背景與動機，本研究的主要目的如下：

1. 蒐集網路電影資料庫IMDb的實際數據，建立使用情感分析與考量時間權重的混合式推薦模型。
2. 比較使用情感分析是否增加模型準確度。
3. 比較考慮時間權重是否增加模型準確度。
4. 驗證過往研究建立電影推薦模型之關鍵因子。
5. 透過實驗評估本研究推薦方法對資料稀疏性問題的改善程度。

## 預期結果與貢獻

本研究的預期結果與貢獻分為學術上與實務上兩部分。學術上貢獻的部分分為四點: 第一點為透過加入評論的時間因素，模型可對使用者興趣隨時間的改變作出應對; 第二點為透過情感分析找出評論中使用者對電影潛在特徵的喜好，能更有效地找出使用者喜歡的電影類型; 第三點為本研究使用基於內容過濾與協同過濾兩種算法，期望透過混和模型能有效的減緩資稀疏對於推薦系統效能的影響; 第四點為本研究合併過往研究中未曾同時考量的評論文本特徵與評論時間特徵，期望能提升模型的準確性。

實務上的貢獻分為兩點: 第一點為線上影音播放平台或電影院等企業能透過此系統更有效的應對使用者喜好的變化並予以準確的推薦，增加使用者黏著度; 第二點為使用者透過此系統能更迅速及準確地找到適合的電影。

# 文獻探討



## 傳統電影推薦系統

使用者在選擇電影時，通常都是透過曾經的印象來挑選相似的產品，抉擇的標準通常是基於產品或使用者的某些特徵，而基於內容推薦便是基於此概念。基於內容推薦是建立在產品或使用者資訊上，透過個人資訊、歷史瀏覽紀錄、產品的文字的描述、使用者的歷史購買紀錄等，來定義產品相關的特徵，分析被使用者購買過與沒購買過產品彼此屬性是否相似，來找出產品之間的關聯性，並以使用者偏好的產品的特徵來預測使用者可能的喜好，再將合適的產品推薦給使用者 (Lops et al., 2019)。

在基於內容電影推薦的過去研究中，Bougiatiotis and Giannakopoulos (2018) 提出基於文本、影像與音頻的電影推薦系統，利用這三個面向去歸類出電影的特徵及屬性。在文本方面，對電影的字幕進行主題建模，用以提取可區分電影的不同主題。在影像方面，對語意上有用的特徵進行萃取，如模擬相機的移動方式、顏色和臉部表情等。而對於音頻方面，採用預訓練模型對音頻做分類彙總。將這三個面向與靜態的元資料如導演、演員等作結合，證明使用低級別的多模型資訊可加強基於內容推薦的相似性效能，且預測結果會比單純使用元資料還準確。

Bergamaschi and Po (2014) 提出基於電影情節的電影推薦系統，透過使用者的觀看紀錄，評估電影情節之間的相似性。透過與資料庫中大量的電影情節比較，系統獨立於使用者評分數量外，因此可以有效推薦使用者情節相似的未觀看過電影，包括當下熱門電影、冷門電影或是老電影。實驗過程中比較了潛在語義分析模型 (Latent Semantic Analysis, LSA)與隱含狄利克雷分布 (Latent Dirichlet Allocation, LDA)的效能，結果顯示LSA應用於電影情節上優於LDA。

Zhang et al. (2015) 提出Triple Wing Harmonium (TWH)模型，將文本元資料整合到低維度語意空間中，以應用於基於內容的電影推薦系統，研究中考慮的文本元資料包括電影簡介、演員列表以及用戶評論等。透過TWH模型，將文本特徵轉換為具有不同分佈概率的假設的低維度潛在主題，另外，利用對比散度演算法 (Contrastive Divergence Algorithm) 進行學習與預測。實驗結果顯示此方法透過將文本特徵降維使用能有效提高模型效能。

另一種廣泛應用的推薦技術為協同過濾，透過使用者與物品組成的User × Item矩陣分析使用者或產品之間的相似性，來預測使用者可能感興趣的內容並將此內容推薦給使用者。協同過濾的算法可分為基於使用者的協同過濾、基於物品的協同過濾與基於模型的協同過濾 (Model-Based)三個種類。基於使用者的基本思想為基於目標使用者對物品的偏好找到相鄰鄰居使用者，然後將鄰居使用者喜歡的推薦給目標使用者。基於物品的原理有一個基本的假設：「能夠引起使用者感興趣的物品，必定與使用者之前給予高評分的物品相似」，通過計算物品之間的相似性來代替使用者之間的相似性。而基於模型則是透過歷史數據得到一個模型，再利用此模型進行預測 (Ekstrand et al., 2011)。

在使用協同過濾電影推薦的過去研究中， Deldjoo et al. (2019) 提出基於電影基因組的推薦系統，透過整合電影音頻與視頻描述符 (Visual Descriptors)來從電影內容中構成電影基因組。並延續Deldjoo et al. (2016) 先前的研究中證明的，利用典型相關分析技術有效的融合數據。另外提出了兩步驟混和方法用於訓練協同過濾模型的暖物品 (即使用者觀看過的電影)，並利用電影基因組來推薦冷物品 (即使用者沒觀看過的電影)。研究針對冷啟動和物品冷至暖的過渡期間的推薦數據，以及使用者為中心的在線實驗進行驗證，結果證明此方法能有效降低冷啟動的影響並提高準確度。

Inan et al. (2018) 提出的電影推薦系統結合電影特徵 (演員、導演、電影類別等) 與協同過濾技術。電影之間的相似性分數以基於內容的目標編程模型輔助計算。而協同過濾使用皮爾森積差相關分析，此方法可以基於電影內容的得分來推薦滿足使用者喜好的電影。該研究使用MovieLens資料集測試，結果顯示當電影內容的資訊增加時，整體推薦系統的性能會獲得提升。

Li et al. (2018) 提出混合電影特徵與使用者興趣的電影推薦系統。在提出的算法中，電影特徵向量是根據電影的屬性計算出的，並與使用者評分矩陣組合以生成使用者興趣向量。電影特徵向量和使用者興趣向量以迭代得方式相互更新，便可獲得基於使用者興趣向量的使用者相似矩陣，而這通常會因資料稀疏性而難以取得。此外，在使用者興趣向量中加入長短期的概念，能使推薦的結果隨者使用者的變化做出反應。

## 情感分析應用於電影推薦系統

## 情感分析

情感分析又稱觀點分析、意見挖掘或主客觀分析等，透過算法對文本的情緒、主客觀性、觀點等層面進行挖掘與分析，並根據結果對文本的情感傾向做出判斷歸類 (Yi et al., 2003)。情感分析文本廣義來說可分成主觀性文本與客觀性文本。客觀性文本為使用者對於物品、事件與其屬性的客觀陳述，主觀性文本通常為使用者對物品、事件與其屬性的主觀評價，當中會包含豐富的主觀性意見、情緒、看法、和態度等 (Liu, 2010)。此分類方法能將文本中描述事實的客觀部分與表達意見的主觀部分區分開，並將主觀部分抽取出，過濾掉不帶情緒色彩的文本，為文本情感極性分析提供主觀性資料 (Bakshi et al., 2016)。

根據文本的顆粒度，可以將文本劃分為詞、句子和文章三個層級，詞為情感分析的基礎，也是分析句子和文章的重要前提。基於詞的情感分析有情感詞抽取、情感詞判定、語料庫和情感字典的使用等。而句子為文本的核心，句子包含了情感詞分析的結果，給整個句子的情感分析完整的結果，另外，句子也可以視為短文章，句子在很大程度上決定了文章的情感分析結果。最後，文章的情感分析是最具不確定性的，因為需要綜合詞與句子情感分析的結果，結合上下文和該領域相關知識做出判斷 (Wang et al., 2016)。另外，情感詞的抽取方式主要分為基於語料庫與基於詞典兩種方式。基於語料庫的方法是運用機器學習的相關技術對詞語的情感進行分類，機器學習的方法通常需要先讓分類模型學習訓練數據中的規律，然後用訓練好的模型對測試數據進行預測，其優點在於簡易實行。而基於詞典的情感詞則是利用詞典中的近義、反義關係以及詞典的結構層次，計算詞語正、負級性種子詞之間的語義相似度，根據語義的遠近對詞語的情感進行分類，常用的詞典包括WordNet、General Inquirer等詞典 (Jiang & Conrath, 1997; Stubbs, 2001)。

而情感分析用於推薦系統則主要以使用者評論為主，使用者的評論一般含有比較有價值的反饋，不但可吸引潛在使用者，而且可幫助使用者作出決定。評論中通常含有使用者對產品不同特徵的感興趣的程度，推薦系統可依此構建出更加準確的興趣模型，進而提升推薦的準確度。

## 使用情感分析於電影推薦的過往研究

在使用情感分析電影推薦的過去研究中， Kumar et al. (2020) 提出的混合式推薦系統中，使用了包括使用者與電影相關的推特情感分析數據、電影元資料和社交圖譜 (Social Graph)做出推薦。該研究運用情感分析數據獲取觀眾們對某部電影的反應和這則推文是否有用等資訊。而混合式推薦則採用基於內容的相似性特徵與基於社交圖譜的協同過濾結合出推薦模型。另外，系統採用加權分數融合的方式來加強推薦效能，結果證明若推薦系統加入有關使用者情緒的相關數據，能有效的提升推薦的準確度與效能。

Wang et al. (2018) 認為過去的研究忽略了隨著使用者與推薦項目的增加，計算的成本也隨之增長的問題，因此提出了基於混合推薦合情感分析的電影推薦框架，其中使用了Spark提升系統的效能。混合式推薦方法由基於內容的推薦與協同過濾組成。另外，作者認為情緒的正面或負面對推薦有重大的影響，一般而言人們傾向對正面評價有正面的影響，而負面評價則反之，因此情感分析有助於提高推薦結果的準確度。

Li et al. (2016) 提出能用於社交網路與挖掘使用者部落格中偏好資訊的電影推薦系統。推薦模型能識別部落格中討論的內容與給定的主題是否相關，且通過使用者部落格之間的相關性，使用基於情緒感知的關聯規則演算法在部落格中識別出一定的模式來制定關聯規則，並且通過此方法來克服推薦系統冷啟動 (Cold-Start) 的問題。而研究也證明此模型不僅能使用於純文本，社交網路中的非結構化信息也是通用的。

Bansal et al. (2016) 提出基於使用者推特類別的電影推薦系統。透過使用者的電影相關的推特文章，從上下文中了解使用者的意圖。推薦系統則根據使用者發布的電影相關推特文章和預測的電影類型來向使用者進行推薦。為此，除了對推特進行文字前處理外，應用了潛在語義索引 (Latent Semantic Indexing, LSI ) 來對數據進行奇異值分解 (Singular Value Decomposition, SVD)，而系統依照IMDb的分類標準對電影進行分類與預測。而在未使用電影特徵如演員、電影評分等資料的情況下，系統已能達到70%以上的準確度。

## 時間權重應用於電影推薦系統

在使用時間權重的協同過濾算法中，充分考慮到時間對使用者興趣的影響，即越早期的興趣其重要性將會越小，而為了降低它對推薦結果的影響，在興趣預測中引入時間函數，時間函數會隨時間的推移一直增加，而時間的權重值皆在(0,1)的範圍之間，這也代表，雖然所有的數據都有利於推薦系統，但新的數據的貢獻程度會更大，而舊數據反映了使用者過去的喜好，它在推薦的預測上將占有較小的影響力。因此透過考慮時間因素，推薦系統能更有效的應對使用者興趣的改變 (Zhang & Liu, 2010)。

在應用時間權重電影推薦的過去研究中，Zhang and Mao (2019) 認為概率矩陣分解 (Probabilistic Matrix Factorization, PMF) 雖然在協同過濾算法中表現出色，但並不帶有任何關於概率的解釋，因此使推廣此框架到其他協同過濾問題相當困難。作者基於此基礎上，提出名為馬爾克夫矩陣分解 (Markovian Factorization of Matrix Process, MFMP) 模型，一方面MFMP能捕捉數據中時間的動態模式，另一方面也有較為清楚的概率公式，使MFMP能夠適應不同的協同過濾問題。

Sun and Dong (2017) 考慮了概念漂移 (Concept Drift) 的特性並利用了聚類和時間衰減的推薦模型 (Dynamic Time Drift Model, DTDM)。DTDM透過應用長期、短期、定期影響的時間影響因子矩陣來預測使用者的評分，並用擬反矩陣來進行模型優化。另外，系統透過分析數據集的時間分佈來計算時間衰減函數，且替換掉算法中的最小二乘法來防止過擬合。實驗透過對比電影推薦平台之三個不同資料集MovieLens100K、MovieLens1M和MovieLens10M證明，此模型能有效提高預測的準確度。

蘭艷(2017) 提出時間加權協同過濾算法 (Neighborhood-Based and Improved Time Weighted Collaborative Filtering Algorithm, NTWCF) 。NTWCF算法在基於資訊保持期概念的基礎上，將資訊影響力保持不變的時間窗融合到資訊指數衰減函數中，得出時間加權分數，並將其應用到物品相似度計算和預測評分兩個階段。另外，改良相似度計算與評分預測方式，有效的提高了算法預測的準確度。

Koren (2009) 提出觀念轉移的概念，為最早提出類似概念的學者。建模時變問題是數據挖掘的核心問題之一，模型不僅要實時反應出數據的動態，還必須在暫時性影響和長期內在特徵中找出平衡點，作者稱此問題為觀念轉移。研究中作者透過基於時間的奇異值分解 (Singular Value Decomposition, SVD)模型，模擬使用者與物品之間隨時間變化的方式，以便提取長期趨勢。而在物品鄰域模型中，通過學習使用者評定的兩個物品之間的影響如何隨時間衰減來了解物品之間更基本的關係。實驗結果證明，在分解和鄰域模型中加入時間動態考量能有效提升推薦的質量。

本研究列出過去電影推薦系統相關的研究文獻，並以所使用之推薦技術、資料集、是否考量時間因素與應用情感分析等條件彙整於表2-1。

表 二‑1: 電影推薦系統相關文獻整理

| 學者 | 資料來源 | 文字探勘 | 時間因素 | 推薦技術 |
| --- | --- | --- | --- | --- |
| Bergamaschi and Po  (2014) | IMDb |  |  | Content Based Filtering |
| Zhang　et al.  (2015) | IMDb |  |  | Content Based Filtering |
| Inan et al.  (2018) | MovieLens 100K |  |  | Collaborative Filtering |
| Li et al.  (2018) | MovieLens 100K |  |  | Collaborative Filtering |
| Deldjoo et al.  (2019) | MovieLens 20M |  |  | Collaborative Filtering |
| Bansal et al.  (2016) | IMDb |  |  | Collaborative Filtering SVD |
| Li et al.  (2016) | Netflix |  |  | Association Rule |
| Wang et al.  (2018) | Douban |  |  | Content Based Filtering Collaborative Filtering |
| Kumar et al.  (2020) | IMDb Netflix Twitter |  |  | Content Based Filtering  Collaborative Filtering |
| Zhang and Mao  (2019) | MovieLens 1M |  |  | MFMP |
| Sun and Dong  (2017) | MovieLens 1M |  |  | Collaborative Filtering |
| 蘭艷(2017) | MovieLens 100K |  |  | Collaborative Filtering |
| Koren (2009) | Netflix |  |  | Collaborative Filtering SVD |

參考資料：本研究整理

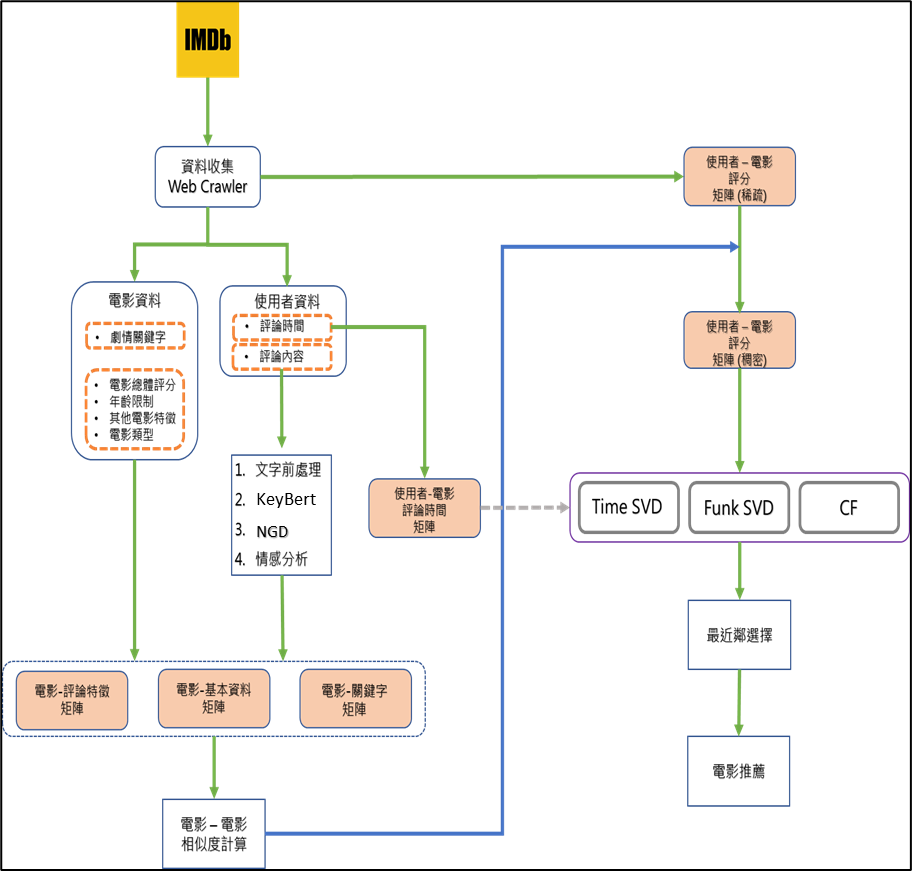
# 研究方法



## 研究流程

研究架構圖如圖3-1。本研究考量了時間與使用者評論等因素，提出基於協同過濾與基於內容過濾的混合式電影推薦系統。首先，透過撰寫網路爬蟲 (Web Crawler)從網路電影資料庫 (Internet Movie Database, IMDb)中抓取研究所需資料，包含電影基本資料、劇情關鍵字與使用者對電影之評論資訊，作為本研究建立電影預測模型所需的資料集。資料集分成電影資料與使用者資料兩部分，經過資料前處理、文字前處理與情感分析等步驟後，分別組成電影－關鍵字矩陣、電影­－基本資料矩陣與電影－評論特徵矩陣，矩陣將用於電影與電影之間的相似度計算。在推薦階段，本研究根據使用者對電影的評分 (1到10分)建立使用者-電影矩陣，並透過使用電影之間的相似度計算結果對使用者－電影評價矩陣進行補值，從而改善資料稀疏性問題。而補值後稠密的使用者－電影評價矩陣將用Time SVD (Koren, 2009)、Funk SVD (Kurucz et al., 2007)與協同過濾等算法進行計算，最終採用Top-K方式產生最佳的推薦列表。

圖 三‑1: 研究架構圖



參考資料：本研究整理

## 資料收集與說明

現今使用者在尋找感興趣的電影時，通常會去與電影相關的網站上瀏覽相關資訊，並且參考電影簡介及電影的評論、評價，再考慮是否觀看此電影。而IMDb為目前全美國最大且最具權威的電影資料庫網站，IMDb上有豐富的電影相關資訊，包含演員、導演、劇情和影評等資訊。本研究為了從使用者評論與電影相關資訊中找出相似電影以進行推薦，使用Python3編寫網路爬蟲程式，預計從IMDb上蒐集前60部排行榜上的熱門電影的實際資料，其中包含IMDb所提供之電影相關資訊、使用者觀看電影後所留下之評論與評論時間等，並將資料儲存於SQL Sever資料庫中。

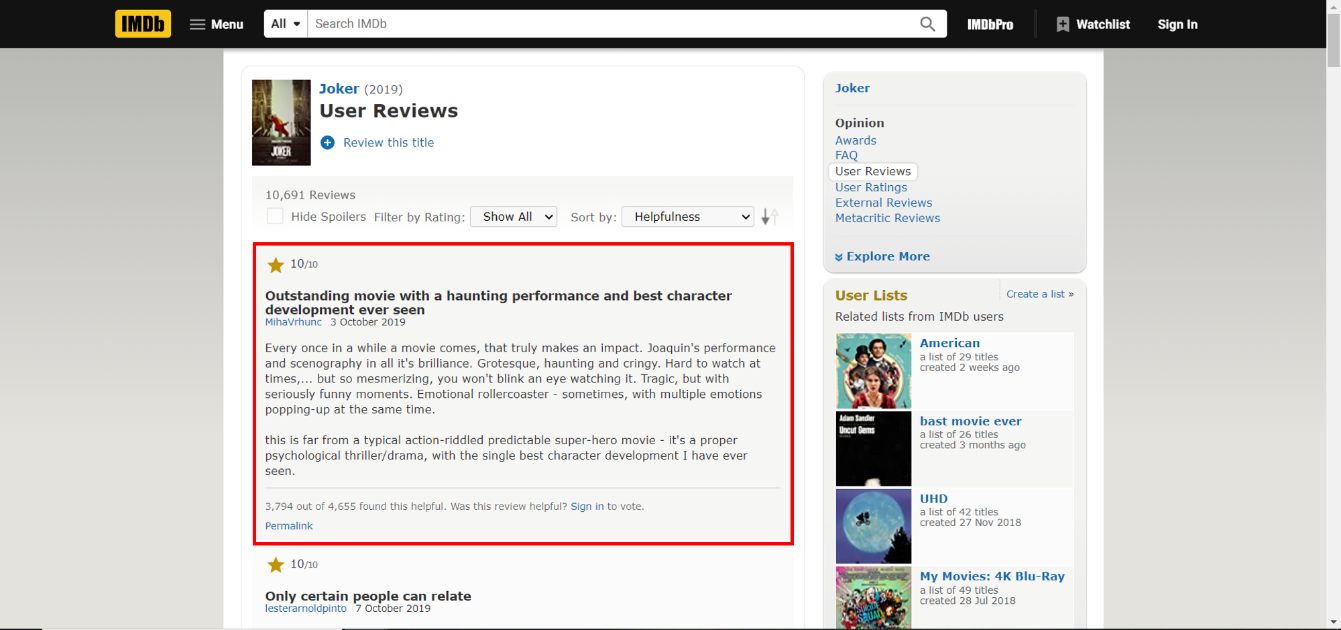
而英文為全世界使用最廣泛的語言，因此本研究以IMDb英文網站資料為主，蒐集英文電影相關資訊與英文使用者評論。資料收集的範圍為2010年1月1號到2020年1月1號期間的IMDb Feature Film前1000部電影，並且依照人氣排序進行蒐集。圖3-2與圖3-3分別為IMDb Feature Film頁面範例與IMDb使用者評論頁面範例。

圖 三‑2: IMDb Feature Film頁面



參考資料：本研究整理

圖 三‑3: IMDb使用者評論頁面



參考資料：本研究整理

經上述步驟所收集的資料類型可分為電影資訊及使用者評論，在與電影相關的資料中收集該電影名稱、播放時常、電影分類、總體評分及電影的劇情關鍵字，如表3-1所示；而表3-2為則為使用者評論的資料說明，收集的資料欄位包含評論的使用者ID、評論時間、有效投票數、總投票數、使用者的評分及使用者對於電影的評論內容。

表 三‑1:電影相關資料說明

|  |  |
| --- | --- |
| 資料名稱 | 說明 |
| Title | 電影名稱 |
| Length | 電影時長 |
| Genres | 電影類別 |
| Overall\_Rating | 電影總評分 |
| Plot\_Keyword | 劇情關鍵字 |

參考資料：本研究整理

表 三‑2:使用者評論資料說明

|  |  |
| --- | --- |
| 資料名稱 | 說明 |
| User\_ID | 使用者ID |
| Review\_Date | 評論時間 |
| Up\_Vote | 有效投票 |
| Total\_Vote | 總投票 |
| User\_Rating | 使用者評分 |
| User\_Review | 使用者評論 |

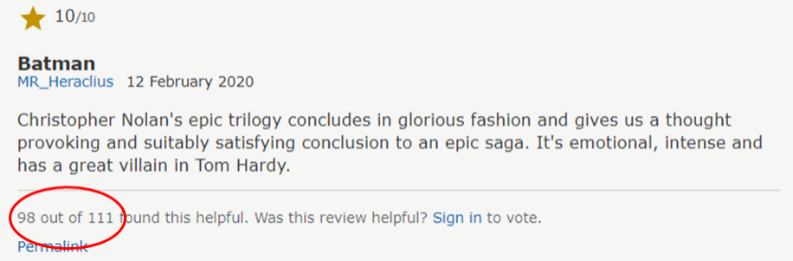
參考資料：本研究整理

## 資料前處理

## 評論篩選

收集到的使用者評論會基於其他使用者給予的評論有用度進行篩選。圖3-4為IMDb有用評論投票頁面範例。IMDb上若使用者想發表電影評論，只需在IMDb上註冊帳號即可開始撰寫，然而網站上無法判別發表該評論的使用者是否觀看過該電影，此問題可能造成收集到的評論無法準確的代表使用者對該部電影的主觀想法。

圖 三‑4: IMDb有用評論投票頁面



參考資料：本研究整理

因此，我們利用評論資訊中包含的其他使用者對於該評論是否有用的投票進行篩選，我們稱此為評論有用度 (Usefulness)，投票資訊中包含了覺得該評論有用的投票量以及總投票量。本研究透過公式(1)的計算方式，以有用的投票量除以總投票量即可得出評論有用度:

(1)

其中 表示覺得評論有效的投票數; 表示投票總數。我們認為評論有用度越高，該評論越能代表使用者對該部電影的主觀看法。本研究參考過去研究將評論有用度小於60%的評論排除，剩下評論有用度大於60%的評論才會被當作實驗資料 (Krishnamoorthy, 2015)。

## 文字前處理

篩選過後的使用者評論將先進行拼寫檢查，改正評論中語法錯誤或拼寫錯誤。使用者在撰寫評論時，時常不經意的發生單詞錯誤。另外，有些使用者會通過重複字詞來強調他們的主觀的感受，例如: “Saaaaad” 這種表示方式為使用者想強調傷心、難過等感覺，但此表示方式並非程式能判斷的標準單詞。且拼寫錯誤將可能影響情感分析中使用者對電影潛在特徵判別的準確度。本研究針對拼寫錯誤使用Python3的拼字檢查套件pyenchant來進行處理，將拼寫錯誤之單詞改正。此步驟幫助提升使用者評論品質，將評論內容修正成正確的表達方式後，再進行後續的文字處理。

在修正拼寫錯誤後，本研究透過Python3自然語言處理工具 (Natural Language Tool Kit, NLTK)來進行文字處理。本研究將使用者電影評論之文字前處理分為五個步驟:包含斷詞 (Tokenize)、詞幹提取 (Stemming)、詞性標記 (Part-of-Speech Tagging, POS)、去除停用字 (Stop Word Remove)及同義詞替換 (Synonym Replace)。

斷詞為自然語言處理的基礎任務，主要是將句子、段落分解為字詞單位的數據結構，方便後續的分析處理。而英文文章的斷詞方式為將句點(.)、問號(?)、驚嘆號(!)等結束符號，作為原始句子斷句的判斷標準，再依照句子中的空格將英文字詞分離，取得獨立的單字。

詞幹提取為去除詞缀得到詞根的過程 (單詞最一般的寫法)，目的為將長相不同，但含意相同的字詞統一，方便後續的分析處理。而常見的詞缀有名詞複數、現在進行式和過去式等，例如：plays、played、playing三者在經過詞幹提取後會等同於單詞play。而本研究採用的詞幹提取算法為NLTK的Snowball算法。

詞性標註為在給定句子中判定每個字詞的語法範疇，確定其詞性並加以標註，使程式能辨識與解析的過程。而英文的詞性基本上可分為動詞 (Verb)、名詞 (Noun) 、形容詞 (Adjective)、副詞 (Adverb)，以及其他類別，例如：代名詞 (Pronoun)、介係詞 (Preposition)、連接詞 (Conjunction)或感嘆詞 (Interjection)等。

停用詞為文章裡出現頻率比較高，幾乎每篇評論中都有出現的詞彙，但對句子沒太大的意義，在不犧牲句子含意的情況下，可以安全的忽略他們。在去除停用字的步驟中會篩選所有的字詞，將停用字去除後，資料集中只會含有較關鍵的單詞。在經過斷詞處理後，使用者的評論文本已經拆解為單詞的集合，而此集合中包含了像是數字、特殊符號或是對評論沒有明顯相關意義的單詞，若不去除這些無意義的單詞，將對後續的情感分析的結果造成影響。本研究使用之停用詞表為NLTK內建，停用詞表如表3-3。

表 三‑3: NLTK停用詞表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | yours | herself | which | was | does | because |
| me | yourself | it | who | were | did | as |
| my | yourselves | its | whom | be | doing | until |
| myself | he | itself | this | been | a | while |
| we | him | they | that | being | an | of |
| our | his | them | these | have | the | at |
| ours | himself | their | those | has | and | by |
| ourselves | she | theirs | am | had | but | for |
| you | her | themselves | is | having | if | with |
| your | hers | what | are | do | or | about |
| against | from | further | any | nor | t | o |
| between | up | then | both | not | can | re |
| into | down | once | each | only | will | ve |
| through | in | here | few | own | just | Y |
| during | out | there | more | same | don | an'tin't |
| before | on | when | most | so | should | aren't |
| after | off | where | other | than | now | couldn't |
| above | over | why | some | too | d | didn't |
| below | under | how | such | very | ll | doesn't |
| to | again | all | no | s | m | hadn't |
| hasn't | weren't | ma | needn't | wasn't | mightn't | shan't |
| haven | won | mustn't | shouldn't | isn't | wouldn't |  |

參考資料：本研究整理

同義詞替換為將兩同義詞轉換為同一個單詞的處理過程。使用者評論文本會與電影潛在特徵種子字進行相似度的計算，來找出評論中與這些特徵有關的關鍵單詞，而為了提高相似度計算的可靠性，本研究採用NLTK內建之WorldNet語料庫進行同義詞的替換。

## 矩陣建立

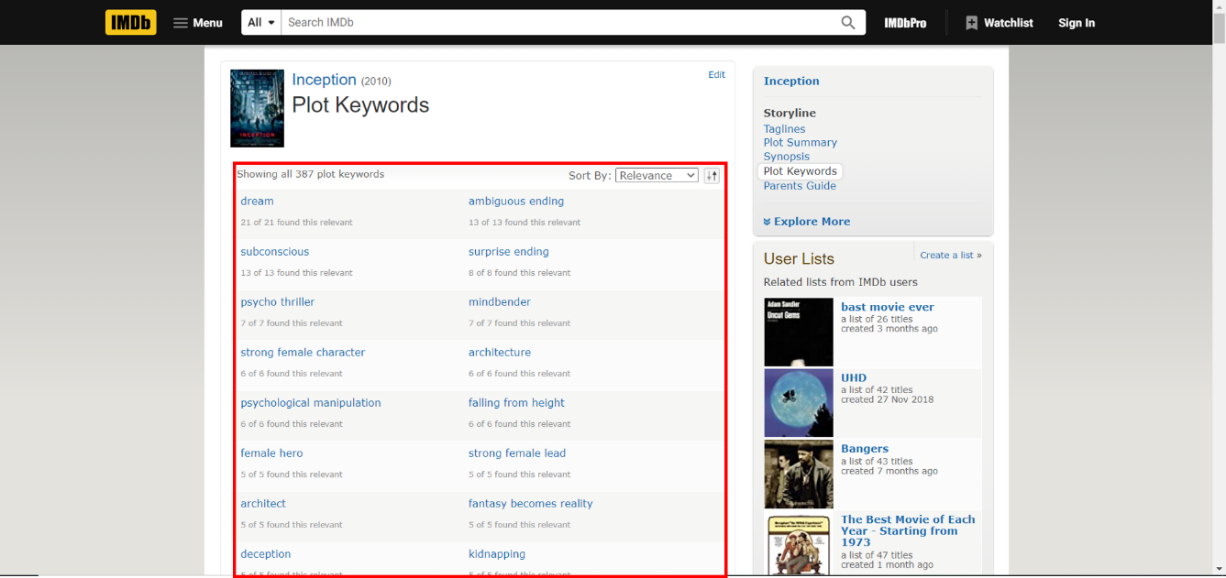
本研究所使用的矩陣包括電影資料矩陣、劇情特徵字矩陣與使用者評論特徵矩陣。本章節將依序從3.4.1節介紹本研究所使用的資料矩陣以及從3.4.2節說明矩陣建立的方式。

## 電影資料矩陣

本研究將從IMDb蒐集的資料分為電影資料和使用者評論資料兩個部分，前者為結構化資料，經過整理排序後可直接用於電影之間的相似度計算，而後者為非結構化資料，需經過文字前處理、KeyBERT、NGD與情感分析等步驟後，儲存成電影 – 評論特徵矩陣使用。此小節將說明電影資料的處理及使用方式，使用者評論之處理方式將於後續小節詳細說明。

電影資料包含電影 – 關鍵字矩陣與電影 – 基本資料矩陣兩部份，關鍵字矩陣為與電影劇情有關的字詞。圖3-5為IMDb劇情關鍵字頁面範例，電影 “全面啟動” 的劇情關鍵字有surprise ending、dream、psycho thriller等。關鍵字從IMDb的Plot Keywords頁面爬取，我們統計了資料集中的電影所含有的關鍵字存放於矩陣中，若電影含有某關鍵字將標示為1，反之為0。表3-4為電影 – 關鍵字矩陣範例，其中代表第部電影。

圖 三‑5: IMDb劇情關鍵字頁面



參考資料：本研究整理

表 三‑4: 電影–關鍵字矩陣

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 電影 / 特徵字 | dream | gun | … | book |
|  | 1 | 1 | … | 0 |
|  | 1 | 0 | … | 0 |
|  | 0 | 0 | … | 1 |

參考資料：本研究整理

而電影基本資料包含電影名稱、電影分類、年齡限制分類、電影片長、總評分資料，本研究以電影 – 基本資料矩陣來計算電影之間基礎屬性的相似性。電影基本資料可從IMDb電影列表頁面爬取，矩陣中如電影類別、年齡限制分類將用標籤編碼 (Label Encoding)方式轉換以方便計算。圖3-6與表3-5分別為IMDb電影基本資料頁面範例與電影 – 基本資料矩陣範例。

圖 三‑6: IMDb電影基本資料頁面



參考資料：本研究整理

表 三‑5: 電影 – 基本資料矩陣

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 電影 / 特徵 | 種類 | 片長 | … | 年齡限制 |
|  | 動作 | 148 | … | PG-13 |
|  | 冒險 | 164 | … | PG-13 |
|  | 科幻 | 135 | … | R |

參考資料：本研究整理

## 定義評論特徵

使用者評論為使用者主觀想法的體現，我們認為在評論中包含了使用者對電影不同面向特徵的喜好，而此類特徵通常不會明確的出現在網站中。學者Jakob et al. (2009)基於維基百科中與電影相關的文章及顯式語義分析 (Explicit Semantic Analysis, ESA) 定義了電影不同面向的特徵，各面向包含種子字用來進行相似度計算。基於此研究，我們將使用者評論中所含的電影特徵分為演技、劇情、製作、配樂與拍攝五個類別，每個類別包含各自獨立的種子字 (Jakob et al., 2009)，用來進行後續的相似度計算與情感分析。表3-6為評論特徵種子字列表。

表 三‑6:評論特徵種子字列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 演技 | 劇情 | 製作 | 配樂 | 拍攝 |
| acting actor role actress filmography co-star act career television theatre | storyline storylines character comic reveal series story appear universe villain | soundtrack song release music album track feature band discography label | cinematography runtime distributor budget min film edit screenplay director star | preproduction Asheville contracted all-animated high-living cash-cow hit-and-miss star-driven singer-actor small-budget |

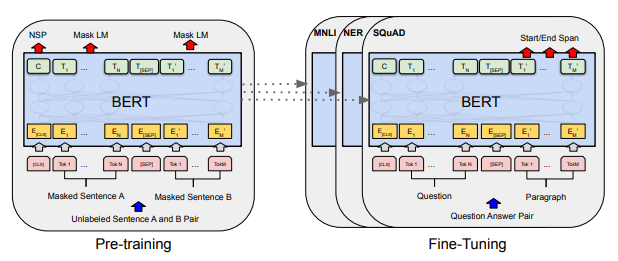
參考資料：本研究整理

## KeyBERT

本研究透過KeyBERT來篩選出評論中較重要的前1500個評論重要單詞，並用於後續與評論特徵種子字進行相似度計算。

KeyBERT為基於Google釋出的BERT(Bidirectional Encoder Representation from Transformers)開發的關鍵字萃取套件，BERT 採用Transformer的Encoder結構，而訓練的部分主要分為預訓練和Fine-Tuning兩個階段，預訓練階段主要透過大型資料集的預訓練任務取得，而Fine-Tuning階段是後續進行其他任務時進行微調，因此BERT不需調整結構也可使用在不同任務上。圖3-7為BERT模型架構圖。

圖 三‑7: BERT模型架構



參考資料：本研究整理

KeyBERT透過BERT提取文本向量以獲得文本級的向量表示。接著，針對N-gram字詞或短語提取詞向量，並透過餘弦相似度來計算每個字詞或短語與該文本的相似程度，相似度最高者則定義成最能描述該文本的詞。

## NGD

本研究使用正常化的Google距離 (Normalized Google Distance, NGD)來測量經由KeyBERT篩選出的評論重要單詞與評論特徵種子字之間的相似度， 並將其用於與評論重要單詞最相似的評論特徵種子字的類別進行歸類，以方便後續情感分析的計算。

而Google距離為一種語義相似性度量的方法，由給定一組關鍵字集合的Google搜尋引擎所返回的命中數量來計算。在自然語言意義上有相同或類似含意的關鍵字往往會在Google距離單元中傾向於緊密，反之不同含意的單詞則往往距離較遠。NGD的公式如下:

(4)

其中是Google搜尋到的網頁總數; *f(x)*和*f(y)*分別是評論重要單詞*x*和電影評論種子字*y*的命中數量; *f(x, y)*是同時出現*x*和*y*的網頁數量。如果*x*和*y*從未出現在同一網頁上，而是單獨出現時，他們之間的NGD值是無窮的。反之，如果兩個單詞總是同時出現，則他們NGD值將會是0，或者是相當於*x*平方和*y*平方之間的係數。

## 情感分析

透過先前步驟，我們利用電影評論特徵的特徵詞找出所有評論中含有特徵詞的文句，本研究透過情感分析計算含有特徵詞文句的情感分數，對加總後的值取平均計算，並經過電影之間相似性計算整合到推薦系統中。根據情感分析計算的結果，我們可以獲取使用者評論中對於電影不同面相特徵的喜好。

本研究使用Python的SentiStrength來計算情感分數，此方法適合處理較低質量或篇幅較短的文本字詞情感提取工具。過去的研究證明，SentiStrength對於電影評論等社交媒體中的文本具有良好的準確性，不管是電影、餐廳或旅館評論，字數都相對簡潔，也因此SentiStrength成為分析使用者評論情感的合適方法之一 (Thelwall et al., 2013) 。

`SentiStrength給予文本中每個字正值與負值兩種分數，以{正值, 負值}表示，字詞正面情緒的正值範圍介於+1到+5之間，其中+5為極度正面之情緒，1代表完全不帶有正面情緒。字詞負面情緒的負值範圍介於-1到-5之間，其中-5為極度負面之情緒，-1則代表完全不帶有負面情緒。例如like的分數為{4, -1，而dislike的分數為{1, -3}。

而分數的選擇方式則參照過去研究，單詞的情感分數等同於其獲得的正值或負值分數，選擇具有最大絕對值的分數作為單詞之情感分數較能具有代表性 (Guzman et al., 2014) 。本研究參照先前研究，選擇具有最大絕對值的分數作為情感分數，並計算含有評論特徵詞之文句的情感分數總和，用以生成電影 – 評論特徵矩陣，如表3-7所示。

表 三‑7:電影 – 評論特徵矩陣

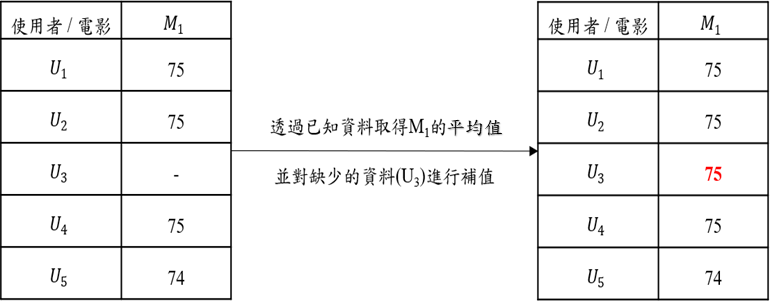
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 電影 / 特徵 | 演技 | 劇情 | … | 配樂 |
|  | 1.9 | 2.4 | … | 0.6 |
|  | -1 | 0.3 | … | -1.2 |
|  | 3.5 | -4.3 | … | 4.7 |

參考資料：本研究整理

## 評論時間

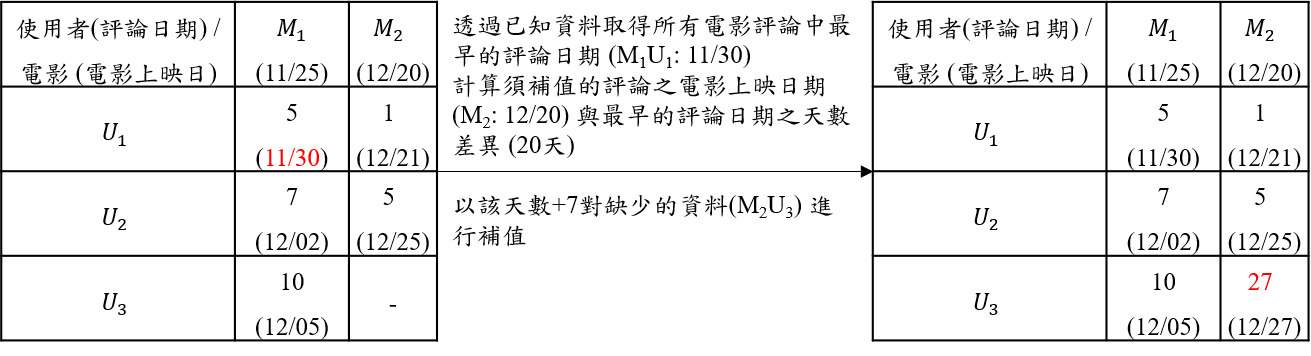
IMDb上的使用者電影評論中包含了評論時間，透過評論時間可獲得評論時間矩陣，而日期的表示方式為資料集中最早的評論日期為起始日，標記為第0天 ，其他評論時間依照起始日往後增加。而為了使本研究基於時間的推薦算法方便計算，除了對使用者 – 電影評分矩陣進行補值，缺少的評論時間也需要補齊。另外，本研究採用兩種不同方式對時間進行補值，第一種為平均數，平均數代表使用多數人評論的平均日期進行補值。第二種為上映日期 + 7天，我們認為上映後第7天為使用者觀看完電影並寫下評論的時間。本研究將在實驗階段比較兩種補值方式對推薦結果的影響，並採用準確度較高者。圖3-8為使用平均數補值範例，假設電影的平均評論時間在資料集中最早的評論日期後的第75天，因此我們對使用者的評論日期補值成第75天。而圖3-9為基於上映日期 + 7天的補值範例，假設電影評論的資料中最早的評論日期為11月30日*(M1U1)*，而需要補值的*M2U3*的電影上映日期為12月20日，本研究先計算出需補值評論的電影上映日期與最早評論的日期之天數差異 (20天)，再以該差異天數加7天 (27天)作為補上的值。

*圖 三‑8: 平均數補值示意圖*



參考資料：本研究整理

圖 三‑9:上映日期 + 7日補值示意圖



參考資料：本研究整理

## 資料補值

在本章節中，不同於過往的協同過濾電影推薦系統，本研究分別從電影基本資料、劇情關鍵字和使用者評論三個方向來進行電影之間相似度的計算，計算結果將用於使用者 – 電影評分矩陣進行補值，來解決資料稀疏性問題。

## 相似度計算

根據先前步驟所獲得的電影 – 關鍵字矩陣、電影 – 基本資料矩陣與電影 – 評論特徵矩陣，本研究以內容導向的方法，尋找相似的電影以補足使用者 – 電影評分中的缺值。而為了衡量電影與電影之間的相似度，我們利用三個矩陣中電影之間的特徵個別進行計算，計算方法採用餘弦相似度 (Cosine Similarity)，公式如下:

(5)

其中，*F*為一組特徵，和表示電影*a*和電影*v* 的特徵頻率。分別計算完三個矩陣電影之間的相似度後，我們可以透過三種不同的矩陣組合包括電影資料 (含劇情關鍵字與電影基本資料)、電影評論特徵、電影資料加電影評論特徵來獲得電影特徵相似度矩陣表3-8，矩陣中的值代表兩兩電影之間的相似度，矩陣相似度的值介於[0, 1]之間，代表電影*a*與電影*v*沒有相同的特徵，反之，若則代表電影*a*與電影*v*有相同的特徵，值越高則兩部電影越相似。

表 三‑8: 電影特徵相似度矩陣表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 電影 / 電影 |  |  | … |  |
|  | 1 |  | … |  |
|  |  | 1 | … | … |
| … | … | … | 1 | … |
|  |  | … | … | 1 |

參考資料：本研究整理

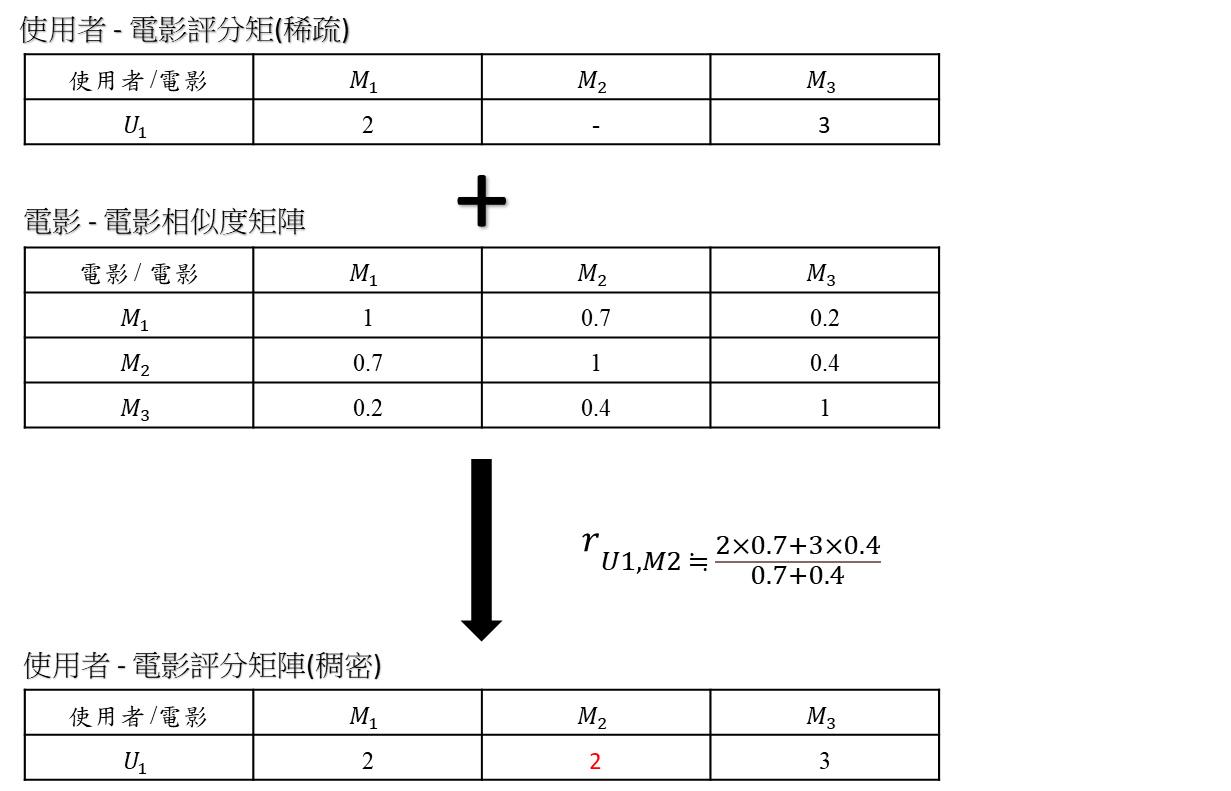
## 使用者 – 電影評分矩陣補值

在計算完電影之間的相似度後，透過電影特徵相似度矩陣，我們可以對稀疏的使用者 – 電影評分矩陣進行補值。補值的計算公式如下:

(6)

其中接著為使用者對電影的評分; 為電影與電影的相似度; 為使用者對電影的評分，計算出的值會取四捨五入作為補值結果。接著，我們能獲得補完值的稠密使用者 – 電影評分矩陣來進行最終的推薦。圖3-9為使用者 – 電影評分矩陣補值示意圖，若要計算使用者對電影的補值分數，可透過使用者對其他電影的評分乘以其他電影與電影的相似性的乘積，除以其他電影與電影的相似性總和並四捨五入取得補值結果。

*圖 三‑10:使用者 – 電影評分矩陣補值示意圖*



參考資料：本研究整理

## 資料探勘技術

本研究採用基於協同過濾的推薦算法，以Python3實現包含Funk SVD、Time SVD以及傳統協同過濾三種算法，最後將說明如何選擇推薦項目。以下小節會依序進行詳細的介紹。

## Funk SVD

Funk SVD的核心思想為認為使用者的興趣只受到少數幾個因素影響，因此將稀疏且高維度的使用者 – 電影評分矩陣分解為兩個低維度矩陣，即為通過使用者和電影評分資訊來學習使用者特徵矩陣和物品特徵矩陣，通過重新構建的低維度矩陣預測使用者對產品的評分。而Funk SVD使用者評分矩陣的預測公式如下:

(7)

α (8)

α (9)

其中為使用者 – 電影評分矩陣;為使用者矩陣投射到低維度矩陣，為*n k*矩陣; 為電影矩陣投射到低維度矩陣 ，為*k t*矩陣; *α*為學習率。而成本函數可被定義成下列公式:

(10)

其中使用者對電影的評分成本函數使用實際值跟預測值相減。接著使用Stochastic Gradient Descent (SGD)最佳化，並找出最佳的矩陣。其中，因此用SGD尋找最小解，因此會將成本函數進行偏微分，找出更新參數的公式。針對使用者矩陣偏微分公式如公式11，公式12則為針對電影矩陣偏微分的公式：

(11)

(12)

由此可知，Funk SVD算法流程主要分為三步驟，首先給定隨機初始矩陣，接著針對每個已知評分計算更新公式直到模型收斂或達到設定之迭代數，最後透過計算結果給予推薦。

## Time SVD

Funk SVD為靜態模型，但現實生活中，物品的流行程度會隨時間變化，使用者的洗好也可能隨時間改變。因此Time SVD基於此架設建模，模型的核心思想是為每個時間段學習一個參數，某個時間段參數使用該時間段數據進行學習，使模型找出潛在的時間模式。而Time SVD使用者評分矩陣的預測公式如下:

*bu(t)* + *bi(t)* + (13)

其中*bu(t)* 和 *bi(t)*分別為使用者和電影偏置隨時間變化的函數; 為使用者隱因子隨時間變化的函數。而對於這些隨時間變化的函數，我們的處理方式是將時間離散化，將整個時間窗按照一定顆粒度進行劃分，顆粒度越小代表時間變化較大，反之顆粒度越大則代表時間變化較慢。例如處理電影*bi(t)*，物品的偏置在短期內部會有很大的改變，因此我們用較大的顆粒度化分，假設資料集涵蓋的時間範圍為一年，整個時間窗口為一年，我們以1個月區段，總共劃分12個區段，每個時間區段賦予一個*Bin(t)* (為1到12整數)，這樣即可將*bi(t)*劃分為靜態及動態兩部分，計算公式如下：

*bi(t)*+ (14)

建模時需對每個時間區段求參數*bi, Bin(t)*，以學習電影流行度隨時間變化的特性。另外，使用者時間數據相較電影較為稀少，因此我們定義關於時間的連續函數來處理使用者偏置，計算方式如公式15所示，*tu*為使用者所有交互數據的平均時間; 代表時間距離; β為超參數，用來衡量時間變化影響程度，越大代表受時間影響程度越大:

*devu(t)* = *sign(t - tu)* (15)

而使用者偏置隨時間變化的函數如公式16，其中α*u*為學習率與時間無關，透過此方法使用者的所有數據皆能進行學習，而時間因素則主要體現在*devu(t)*上:

*bu(t)* = *bu* +α*u devu(t)* (16)

## 協同過濾

協同過濾使用補值後的使用者 – 電影評分矩陣進行相似度計算，相似度計算採用餘弦相似度，計算公式如下:

(17)

其中代表使用者*i*與使用者*j*共同評價過的電影; 使用者*i*對電影*m*的評分; 使用者*i*評分的平均分。經過計算後可獲得使用者 – 使用者相似度矩陣，如表3-9所示。

表 三‑9: 使用者 – 使用者相似度矩陣

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 使用者 / 使用者 |  |  | … |  |
|  | 1 |  | … |  |
|  |  | 1 | … | … |
| … | … | … | 1 | … |
|  |  | … | … | 1 |

參考資料：本研究整理

最後根據使用者 – 電影評分矩陣的使用者之建的相似性選擇了前*k*個最相似的鄰居並且產生預測分數，計算公式如下:

(18)

其中包含和使用者喜好最相似的*K*個使用者;為使用者對電影*m*有過評分的使用者集合; *wuv*為使用者*u*和使用者*v*的喜好相似度; 為使用者*v*對電影*m*的評分。

## 產生推薦結果

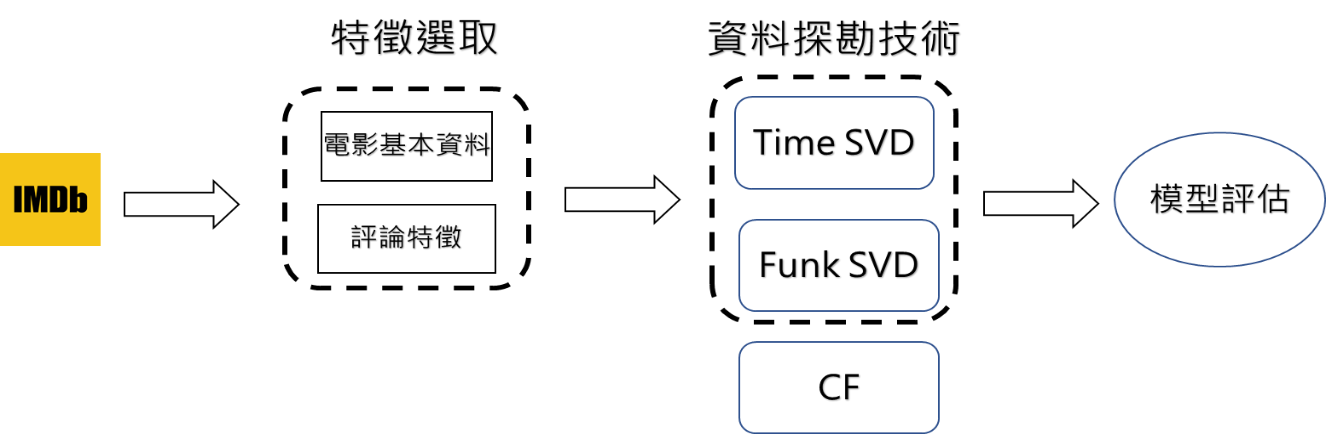
經過三種推薦算法計算過後，我們獲得三種不同算法對玩家的推薦分數結果，最後再分別依照Top-k推薦方式來產生最終的推薦分數。不同算法的推薦準確度將進一步驗證並選擇準確度最高者。

## 實驗設計

基於第三章所說明的不同資料矩陣以及推薦算法，本研究針對時間因素、補值方式以及不同資料組合建構三種實驗。

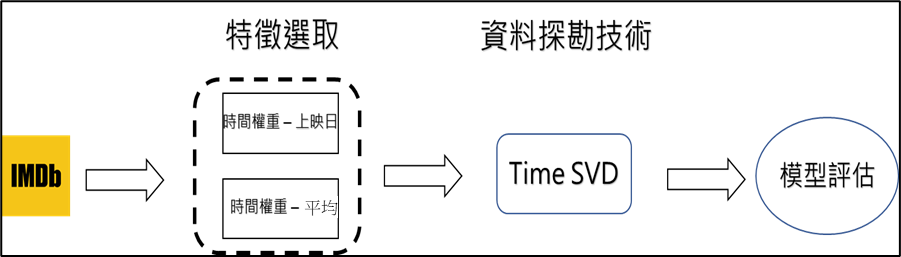
實驗一為使用電影資料及使用者評論資料對稀疏的使用者 – 電影評分矩陣進行補值，並透過三種算法之計算結果來探討加入時間因素的考量是否能增加推薦準確度，如圖4-1所示。實驗二為針對Time SVD所使用之不同補值方式是否對推薦結果有所影響，分為評論時間取平均日期與按照上映日+7天進行時間的補值，如圖4-2所示。而圖4-3為實驗三的架構圖，實驗三使用不同資料組合搭配Time SVD進行推薦，資料組合分為四種，分別為單獨使用電影資料補值、單獨使用使用者評論資料補值、同時使用電影資料及使用者評論資料補值與不使用資料補值。透過所設計之三種實驗方法，我們最後將比較哪種方法有較良好的推薦結果。

圖 四‑1: 實驗一



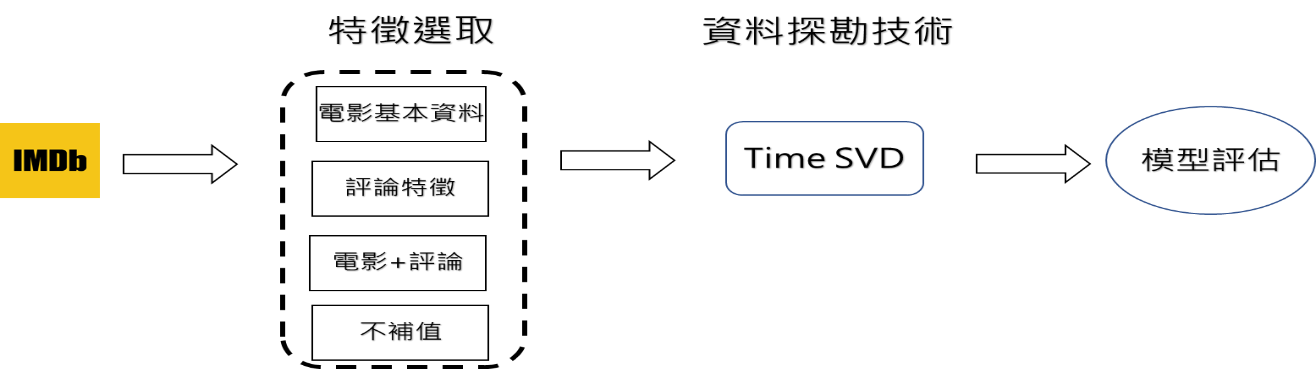
參考資料：本研究整理

圖 四‑2: 實驗二



參考資料：本研究整理

圖 四‑3: 實驗三



參考資料：本研究整理

## 資料驗證與評估指標

本研究使用留一驗證 (Leave-one-out Cross-Validation, LOOCV)方法來對推薦系統進行評估，LOOCV執行方式為每次只使用一筆資料來做測試，而剩下的資料則保留下來用作訓練，此步驟持續重複直到資料集中的每一筆樣本皆被當作過測試資料為止。

透過LOOCV法我們將輪流保留一位使用者的評分資料，其餘使用者評分資料作為訓練使用，以確保資料集中的所有樣本皆有被作為訓練集和測試集且經過驗證，最後將預測結果與真實答案進行比較來評估模型好壞。

本研究採用採用平均絕對誤差 (Mean Absolute Error, MAE)與均方根誤差(Root Mean Square Error, RMSE)作為系統的評估指標，此兩種指標為推薦系統研究領域中最常見的評估指標。MAE為絕對誤差的平均值，用來反映預測值誤差的實際情況。在電影推薦系統中，用來評估使用者對電影的評分與實際使用者對電影的評分之平均絕對誤差，MAE值越低，代表預測結果有較高的準確度。其計算公式如下:

(19)

而RMSE為用來評估預測值和實際值之差的樣本標準差，較低的RMSE代表有較高的推薦準確度，與MAE的主要差別為RMSE在對誤差取平均前做了平方。其計算公式如下:

(20)

MAE與RMSE的計算公式中，T為訓練資料中的評分總數; 預測使用者*i*對電影*a*的評分標記為*Pi,a*; 實際使用者*i*對電影*a*的評分標記為*Aia*; *Pi,a* - *Ai,a*為預測值與實際值的誤差。本研究依照上述的實驗建構步驟，使用MAE與RMSE作為實驗的評估指標來衡量推薦系統之準確度。

# 實驗方法與設計



## 實驗資料

根據本研究之3.2章節資料收集與說明之規範，本研究的實驗資料來源為IMDb英文網站2010年1月1號到2020年1月1號期間的IMDb Feature Film排行前60部電影，其中包含836名使用者對該60部電影的評論資料，每名使用者含有最少20部最多60部電影評論資料。

## 實驗結果與評估

本研究設計三項實驗:實驗一比較三種算法之實驗結果來探討加入時間因素的考量是否能增加推薦準確度；實驗二比較Time SVD模型之時間參數使用不同補值方法的實驗結果，探討兩種不同補值方打對推薦模型的結果的影響；實驗三則是比較Time SVD模型在使用不同資料組合與參數之實驗結果的差異，並找出實驗結果最佳之資料組合與參數配置。

## 實驗一

實驗一實驗結果如圖?與表?所示，其中f為SVD模型引入的隱含空間維度，使用之訓練與驗證資料皆經過電影基本資料、電影劇情關鍵字與使用者評論文本資料進行補值，在模型經過訓練損失值趨於穩定後，加入時間參數的Time SVD模型之訓練結果在評估指標RMSE與MAE皆為最低，而Funk SVD與基於內容的協同過濾模型的訓練結果在兩種評估指標中分別獲得居中以及最高，其中，三種模型之結果在RMSE與MAE最高與最低分別相差大約2.295和1.025。

|  |  |  |
| --- | --- | --- |
| 實驗一 | f = 100 | |
| RMSE | MAE |
| Time SVD | 0.922185 | 0.515967 |
| Funk SVD | 0.976254 | 0.541268 |
| CF | 3.217323 | 1.541268 |

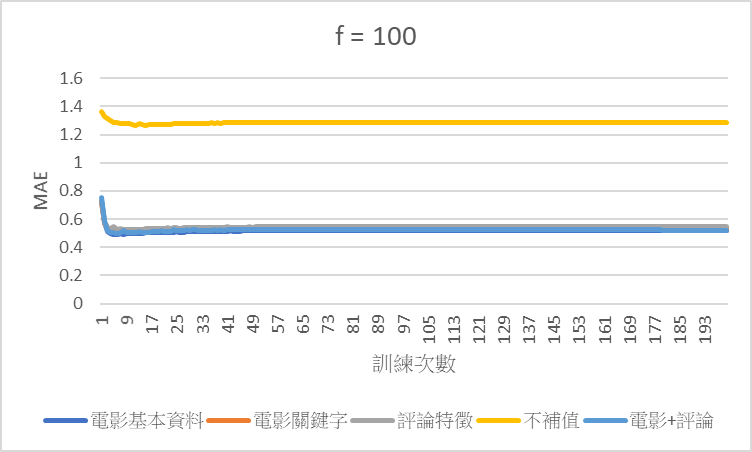
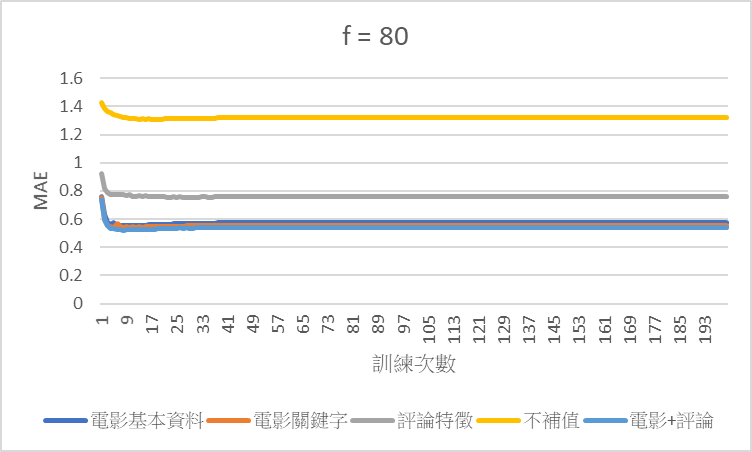
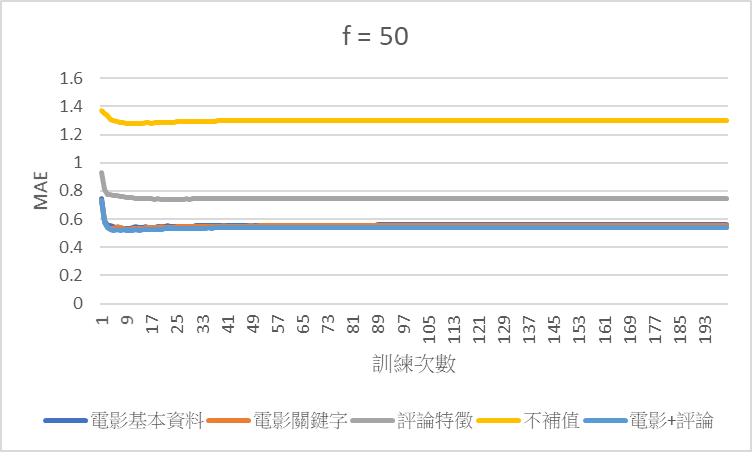
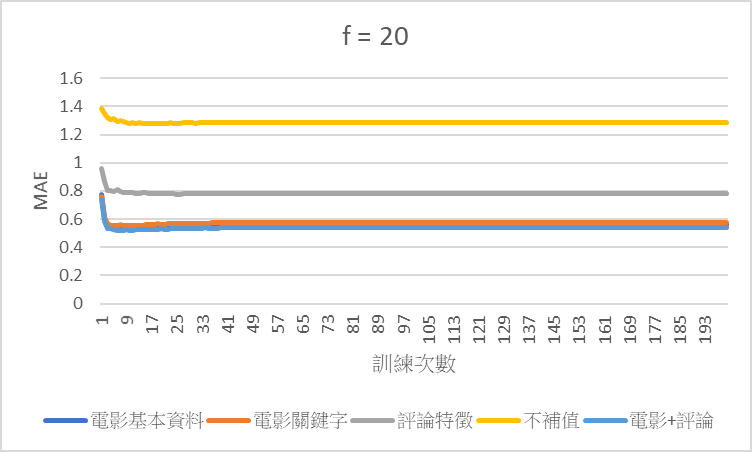
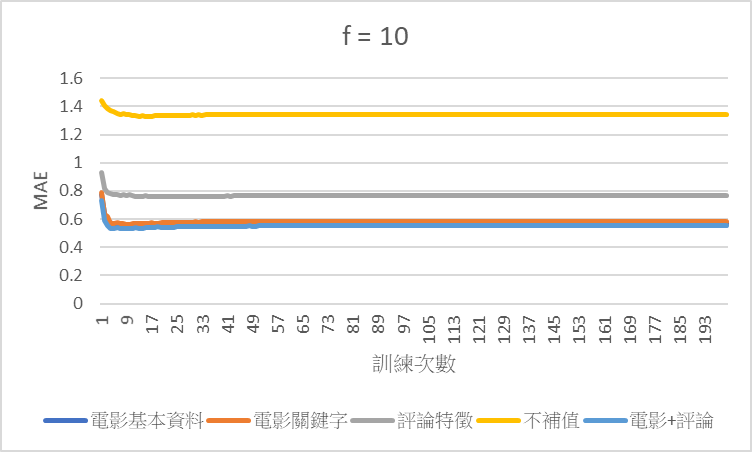
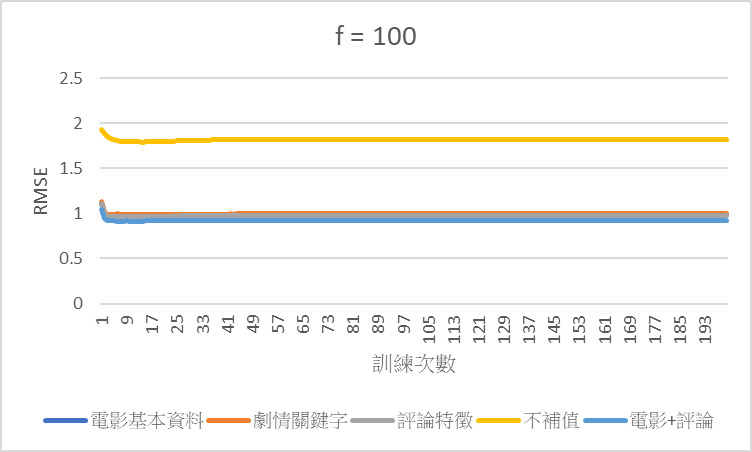
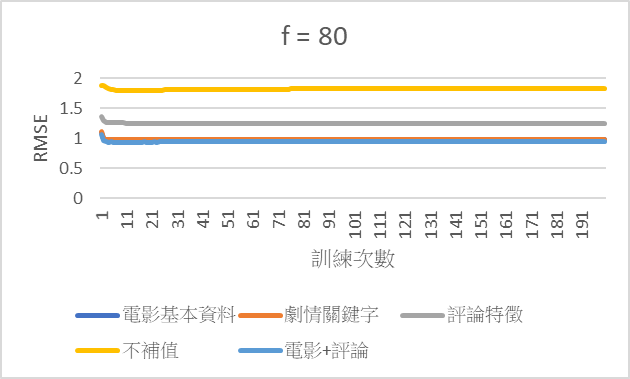
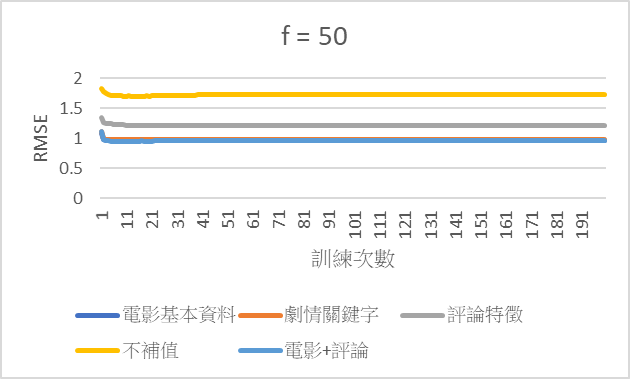
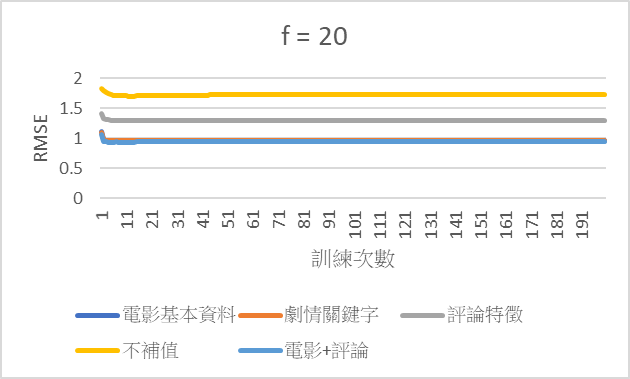
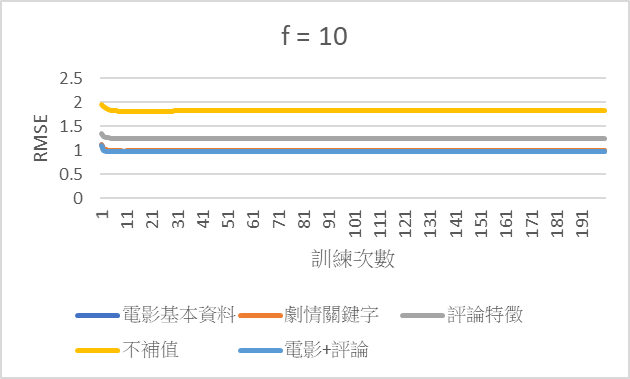
## 實驗二

實驗二實驗結果如圖?與表?所示，使用之訓練與驗證資料皆經過電影基本資料、電影劇情關鍵字與使用者評論文本資料進行補值，評論時間經過上映日與平均值兩種補值後，使用使用者平均評論時間的RMSE與MAE相較於使用上映日+7皆有較佳的結果，其中，兩種不值方式之結果在RMSE與MAE分別相差大約0.037與0.016，故實驗二選出以評論時間取平均值的資料集作為後續實驗三之探討對象。

|  |  |  |
| --- | --- | --- |
| 實驗二 | f = 100 | |
| RMSE | MAE |
| 平均值補值 | 0.922185 | 0.515967 |
| 上映日+7補值 | 0.960043 | 0.532133 |

## 實驗三

實驗三實驗結果如圖?與表?所示， 評論時間接續實驗二皆以評論時間取平均值最為補值方式，並且使用不同的隱含空間維度進行不同資料組合間的比較，在不同數量的隱含空間維度下，使用的隱含空間維度越大模型實驗平均結果起來皆有更低的誤差值。而在不同資料組合的部分，使用所有資料組合進行補值在不同的隱含空間維度下皆有最佳的結果，其次，使用電影基本資料與電影劇情關鍵字之平均結果為次佳，使用者評論資料則在各維度的誤差值大部分較前兩者略高一些，而不對資料進行補值在不同維度的實驗中皆獲得最差的結果。其中RMSE與MAE在維度f為10、20、50、80、100的情況下，最高與最低值差距RMSE大約為0.8565、0.7797、0.7692、0.8796與0.9016，MAE大約為0.7928、0.7496、0.7608、0.7790與0.7707。



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 實驗三 | | f = 10 | f = 20 | f = 50 | f = 80 | f = 100 |
| 電影基本資料 | RMSE | 0.989287 | 0.959643 | 0.965065 | 0.975067 | 0.987367 |
| MAE | 0.565031 | 0.562675 | 0.557987 | 0.576205 | 0.517708 |
| 劇情關鍵字 | RMSE | 0.991512 | 0.970407 | 0.984864 | 0.978366 | 0.994706 |
| MAE | 0.584662 | 0.576795 | 0.55294 | 0.557069 | 0.535667 |
| 評論特徵 | RMSE | 1.245515 | 1.300677 | 1.217323 | 1.250103 | 0.976273 |
| MAE | 0.765196 | 0.779772 | 0.745415 | 0.758044 | 0.545667 |
| 不補值 | RMSE | 1.831434 | 1.72546 | 1.728419 | 1.821797 | 1.823797 |
| MAE | 1.345086 | 1.289368 | 1.301652 | 1.322538 | 1.286756 |
| 電影+評論 | RMSE | 0.974893 | 0.945673 | 0.959204 | 0.942185 | 0.922185 |
| MAE | 0.552266 | 0.539694 | 0.540754 | 0.543524 | 0.515967 |

## 討論

本研究使用IMDb電影資料庫網站中2010年1月1號到2020年1月1號期間的Feature Film前60部熱門電影相關資料，包含電影基本資料、電影劇情關鍵字資料以及使用者評論文本資料用以進行實驗，並針對以下三點進行實驗結果探討：

1. 加入時間參數使否對推薦模型的準確度有影響。
2. 對時間資料採用不同補值方式是否對推薦模型的準確度產生影響。
3. 不同的資料組合以及參數的配置是否對推薦模型的準確度產生影響。

於實驗一討論第一點：加入時間參數使否對推薦模型的準確度有影響。實驗使用含有時間參數的Time SVD模型、不含時間參數的Funk SVD模型以及傳統的基於內容的協同過濾模型。首先，從實驗結果可以發現使用矩陣分解的模型在RMSE與MAE的表現皆遠優於協同過濾模型，表示在使用者 – 電影評分矩陣為稀疏的情況下，採用矩陣分解類型的模型較能有效的去預測使用者的喜好。接著在Time SVD與Funk SVD兩種模型中，含有時間參數的Time SVD有較佳的實驗結果，可推論在使用者的喜好會隨時間有所感變的情況下，使用時間參數能使推薦模型更有效的預測使用者在不同時間段下的喜好。

於實驗二討論第二點：對時間資料採用不同補值方式是否對推薦模型的準確度產生影響。時間資料的補值採用平均日期與上映日+7天兩種方式，兩種補值方式之實驗結果以平均日期補值較佳，推測可能原因為因使用之電影資料為Feature Film十年間的熱門電影，使用者對這些電影的評論熱度在這期間內沒有明顯的下降，因此隨著時間範圍拉長，依然會有新的使用者對這些電影進行評價。因此採用上映日期+7天的方式進行補值可能會造成時間資料分布的偏移，無法維持原有的分布型態，而採用平均日期補值的方式則對資料分布有較小的影響。

於實驗三討論第三點：不同的資料組合以及參數的配置是否對推薦模型的準確度產生影響。實驗三所使用之資料組合以電影相關資料與使用者評論資料作排列組合構成，使用參數配置以比較矩陣分解模型在不同的隱含空間維度下的實驗結果為主。在不同資料組合的實驗結果方面，有使用所蒐集資料對使用者 – 電影評分矩陣進行補值之實驗結果在不同隱含空間維度下皆遠優於不對使用者 – 電影評分矩陣進行補值。而在所使用的補值資料以含有所有電影相關資料與評論資料之資料組合為最佳結果，表示使用者 – 電影評分矩陣在經過補值後更能表現出使用者與電影在不同面向的特徵，推薦模型因此能更準確地給予使用者電影推薦。而在電影基本資料、劇情關鍵字資料與評論特徵資料之中，前兩者在不同隱含空間維度的表現皆較為接近，而評論特徵資料則稍遜前兩者，可能原因為所使用之文字探勘語料庫皆以英文為主，但部分評論關鍵字中所使用之詞彙並非常用之英文單詞，如漫威系列電影《Black Panther》中時常出現之Wakanda一詞則是該電影原創單詞，在對評論關鍵字進行相似性計算與分類至評論特徵類別中會產生不準確的問題。

而在實驗三中不同隱含空間維度的實驗結果中可發現在高隱含空間維度情況下，模型在不同資料組合下的大部分實驗結果皆優於低隱含空間維度，表示在使用者 – 電影評分矩陣維度較大的情況下，矩陣分解模型採用高隱含空間維度較能捕捉使用者與電影的隱含特徵，因此模型能更準確的預測使用者的喜好。

# 研究計劃



## 研究討論

推薦系統是目前解決資訊過載的主流方法之一，而電影推薦系統是推薦系統研究中的熱門領域。過去研究中較常採用電影的類別特徵與使用者評分來建立模型，隨著文字探勘技術與網路評論平台逐漸普及，除了以電影分類等資訊外，許多學者證實了文本資料內也包含了能有效理解使用者興趣的資訊，並且可透過文字探勘與自然語言技術來實現推薦系統的建立。

本研究採用網路電影資料庫IMDb上之電影基本資料、劇情關鍵字資料與使用者評論文本資料，並加入文字探勘與矩陣分解等技術，建立本研究之實驗並進行探討，實驗針對不同面向包含：(1)時間參數的影響、(2)時間的補值方法、(3)不同資料組合與隱含空間維度進行探討。

根據實驗結果，本研究所加入之時間參數在總體表現皆有不錯的結果，並且透過實驗找出較好的補值方式，讓模型在加入時間資料後能更有效的預測使用者不同時間下的喜好。另外，透過不同資料組合與隱含空間維度的實驗結果，模型RMSE與MAE皆有不同程度的提升，但在使用者評論文資料部分實驗結果相較於其他組合結果較不理想，表示系統在分類評論文本特徵時會有誤判的情況，導致RMSE與MAE較其他資料組合高。然而本研究所提出之方法與所使用之資料與參數組合在三種不同實驗中總體表現最佳，表示推薦模型具有預測使用者喜好的價值。

基於本研究之實驗結果，學術上與實務上有下列貢獻：學術方面，本研究藉由整理過去相關文獻，在推薦系統中同時加入對時間的考量與使用者評論文本資料，並透過嘗試不同的資料、參數組合與補值方式，建立混合式的電影推薦系統，並加以驗證其效果；實務方面，本研究所提出之電影推薦系統，使線上影音播放平台或電影院等相關企業能透過此系統更有效給予使用者電影推薦並增加使用者黏著度，而使用者則能透過此系統更迅速及準確的找到適合自己的電影。

## 研究限制

本研究使用電影相關資料與使用者評論文本資料，採用KeyBERT與NGD技術進行文字探勘分析，並使用三種推薦算法建立推薦系統，然而受到以下限制因而尚有不周全之處：

1. 本研究實驗所使用之使用者評論文本資料中包含使用者評論、評論時間與對電影之評分，然而IMDb網站並未強制要求使用者評分時留下評論資料，故在收集使用者評論資料時，為使用者評分與評論文本完整，故加上一定的篩選條件，可能導致部分使用者資料無法使用，以致不能預測每位使用者的喜好。
2. 本研究所採用之使用者評論文本特徵分類方法，若在文本中有非常見英文單詞或非英文單詞，可能會有影響特徵分類結果之疑慮。

## 未來研究方向

於研究資料方面，本研究僅蒐集IMDb網站上的部份資料使用，未來考慮可加入對電影劇照、音訊等特徵進行探討，以更加全面的獲取電影與使用者特徵。此外，在面對方常見單詞部分，可嘗試預先將相關單詞加入語料庫，以增進文本特徵分類之結果。

於資料探勘技術方面，本研究所使用之推薦算法為協同過濾與矩陣分解模型，在未來可以考慮加入神經網路模型，如GRU。

# 參考文獻

**中文文獻**

蘭艷 & 曹芳芳(2017)。面相電影推薦的時間加權偕同過濾演算法的研究。*計算機科學，(4)*，295-301。

**英文文獻**

Aljukhadar, M., Senecal, S., & Daoust, C. E. (2010, September). *Information overload and usage of recommendations* [Conference Presention]. In Proceedings of the ACM RecSys 2010 workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI). Barcelona, Spain. http://doi.org/10.1145/1864708.1864800.

Aljukhadar, M., Senecal, S., & Daoust, C. E. (2012). Using recommendation agents to cope with information overload. *International Journal of Electronic Commerce, 17*(2), 41-70. http://doi.org/10.2753/JEC1086-4415170202

Aljunid, M. F., & Manjaiah, D. H. (2019). *Movie Recommender System Based on Collaborative Filtering Using Apache Spark* [Paper Presention]. In Data Management, Analytics and Innovation (pp. 283-295). Singapore. http://doi.org/10.1007/978-981-13-1274-8\_22

Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., & Netanely, I. (2013, October). *Movie recommender system for profit maximization* [Conference Presention]. In Proceedings of the 7th ACM conference on Recommender systems (pp. 121-128). Hong Kong, China. http://doi.org/10.1145/2507157.2507162

Bakir, C. (2018). Collaborative filtering with temporal dynamics with using singular value decomposition. *Tehnički vjesnik, 25*(1), 130-135. http://doi.org/10.17559/TV-20160708140839

Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016, March). *Opinion mining and sentiment analysis*[Conference presentation]. In 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 452-455). New Delhi, India. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7724305

Bansal, S., Gupta, C., & Arora, A. (2016, August). *User tweets based genre prediction and movie recommendation using LSI and SVD* [Paper Presention]. In 2016 Ninth International Conference on Contemporary Computing (IC3) (pp. 1-6). Noida, India. http://doi.org/10.1109/IC3.2016.7880220

Bawden, D., & Robinson, L. (2009). The dark side of information: overload, anxiety and other paradoxes and pathologies. *Journal of information science, 35*(2), 180-191. http://doi.org/10.1177/0165551508095781

Bergamaschi, S., & Po, L. (2014, April). *Comparing LDA and LSA topic models for content-based movie recommendation systems* [Conference presentation]. In International conference on web information systems and technologies (pp. 247-263). Barcelona, Spain. http://doi.org/10.1007/978-3-319-27030-2\_16

Berghel, H. (1997). Cyberspace 2000: Dealing with information overload. *Communications of the ACM, 40*(2), 19-24. https://doi.org/10.1145/253671.253680

Bougiatiotis, K., & Giannakopoulos, T. (2018). Enhanced movie content similarity based on textual, auditory and visual information. *Expert Systems with Applications, 96*, 86-102. http://doi.org/10.1016/j.eswa.2017.11.050

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction, 25*(2), 99-154. http://doi.org/10.1007/s11257-015-9155-5

Chen, M. H., Teng, C. H., & Chang, P. C. (2015). Applying artificial immune systems to collaborative filtering for movie recommendation. *Advanced Engineering Informatics, 29*(4), 830-839. http://doi.org/10.1016/j.aei.2015.04.005

Costa, H., & Macedo, L. (2013, May). *Emotion-based recommender system for overcoming the problem of information overload* [Paper Presention]. In International Conference on Practical Applications of Agents and Multi-Agent Systems (pp. 178-189). Berlin, Heidelberg. http://doi.org/10.1007/978-3-642-38061-7\_18

Deldjoo, Y., Dacrema, M. F., Constantin, M. G., Eghbal-Zadeh, H., Cereda, S., Schedl, M., ... & Cremonesi, P. (2019). Movie genome: alleviating new item cold start in movie recommendation. *User Modeling and User-Adapted Interaction, 29*(2), 291-343. http://doi.org/10.1007/s11257-019-09221-y

Deldjoo, Y., Elahi, M., Cremonesi, P., Moghaddam, F. B., & Caielli, A. L. E. (2016, September). *How to combine visual features with tags to improve movie recommendation accuracy?* [Paper Presention]. In International conference on electronic commerce and web technologies (pp. 34-45). Portugal, Porto. http://doi.org/10.1007/978-3-319-53676-7\_3

Edmunds, A., & Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. *International journal of information management, 20*(1), 17-28. http://doi.org/10.1016/S0268-4012(99)00051-1

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). Collaborative filtering recommender systems. *Now Publishers Inc*.http://doi.org/10.1561/9781601984432

Grčar, M., Mladenič, D., Fortuna, B., & Grobelnik, M. (2005, August). *Data sparsity issues in the collaborative filtering framework* [Conference Presention]. In International workshop on knowledge discovery on the web (pp. 58-76). Chicago, IL, USA. http://doi.org/10.1007/11891321\_4

Hemp, P. (2009). Death by information overload. *Harvard business review, 87*(9), 82-9.

Huang, Y. M., Liu, C. H., Lee, C. Y., & Huang, Y. M. (2012). Designing a personalized guide recommendation system to mitigate information overload in museum learning. *Journal of Educational Technology & Society, 15*(4), 150-166.

Hwangbo, H., & Kim, Y. (2017). An empirical study on the effect of data sparsity and data overlap on cross domain collaborative filtering performance. *Expert Systems with Applications, 89,* 254-265. http://doi.org/10.1016/j.eswa.2017.07.041

Ibrahim, M., Bajwa, I. S., Ul-Amin, R., & Kasi, B. (2019). A neural network-inspired approach for improved and true movie recommendations. *Computational intelligence and neuroscience, 2019*. http://doi.org/10.1155/2019/4589060

Inan, E., Tekbacak, F., & Ozturk, C. (2018). Moreopt: A goal programming based movie recommender system. *Journal of computational science, 28*, 43-50. http://doi.org/10.1016/j.jocs.2018.08.004

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal, 16*(3), 261-273. http://doi.org/10.1016/j.eij.2015.06.005

Jakob, N., Weber, S. H., Müller, M. C., & Gurevych, I. (2009, November). *Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations* [Conference Presention]. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (pp. 57-64). Hong Kong, China. https://doi.org/10.1145/1651461.1651473

Jiang, J. J., & Conrath, D. W. (1997). *Semantic similarity based on corpus statistics and lexical taxonomy*[Conference Presention]. In the Proceedings of ROCLING X. Taiwan. https://arxiv.org/pdf/cmp-lg/9709008.pdf

Jugovac, M., & Jannach, D. (2017). Interacting with recommenders—overview and research directions. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *7*(3), 1-46. http://doi.org/10.1145/3001837

Koren, Y. (2009, June). *Collaborative filtering with temporal dynamics*[Conference Presention]. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 447-456). New York, United States. https://doi.org/10.1145/1557019.1557072

Krishnamoorthy, S. (2015). Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, *42*(7), 3751-3759. http://doi.org/10.1016/j.eswa.2014.12.044

Kumar, S., De, K., & Roy, P. P. (2020). Movie recommendation system using sentiment analysis from microblogging data. *IEEE Transactions on Computational Social Systems, 7*. http://doi.org/10.1109/TCSS.2020.2993585

Kurucz, M., Benczúr, A. A., & Csalogány, K. (2007, August). *Methods for large scale SVD with missing values* [Paper Presention]. In Proceedings of KDD cup and workshop (Vol. 12, pp. 31-38). https://www.cs.uic.edu/~liub/KDD-cup-2007/proceedings/missing-value-Kurucz.pdf

Lekakos, G., & Caravelas, P. (2008). A hybrid approach for movie recommendation. *Multimedia tools and applications, 36*(1-2), 55-70. http://doi.org/10.1007/s11042-006-0082-7

Li, H., Cui, J., Shen, B., & Ma, J. (2016). An intelligent movie recommendation system through group-level sentiment analysis in microblogs. *Neurocomputing, 210*, 164-173. http://doi.org/10.1016/j.neucom.2015.09.134

Li, J., Xu, W., Wan, W., & Sun, J. (2018). Movie recommendation based on bridging movie feature and user interest. *Journal of computational science, 26*, 128-134. http://doi.org/10.1016/j.neucom.2015.09.134

Li, X., Jiang, W., Chen, W., Wu, J., & Wang, G. (2019, November). *Haes: A new hybrid approach for movie recommendation with elastic serendipity*[Conference presentation]. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 1503-1512). New York, United States. https://doi.org/10.1145/3357384.3357868

Liang, T. P., Lai, H. J., & Ku, Y. C. (2006). Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems, 23*(3), 45-70. http://doi.org/10.2753/MIS0742-1222230303

Liu, B. (2010). *Sentiment analysis and subjectivity* (Second Edition). Handbook of natural language processing, 627-666. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=3B16DA2E9929570680B5FE56E03E32F0?doi=10.1.1.216.5533&rep=rep1&type=pdf

Lops, P., Jannach, D., Musto, C., Bogers, T., & Koolen, M. (2019). Trends in content-based recommendation. *User Modeling and User-Adapted Interaction, 29*(2), 239-249. https://doi.org/10.1007/s11257-019-09231-w

Maes, P. (1995). Agents that reduce work and information overload. *In Readings in human–computer interaction* (pp. 811-821). http://doi.org/10.1016/B978-0-08-051574-8.50084-4

Oh, J., Kim, S., Yun, S. Y., Choi, S., & Yi, M. Y. (2019, September). *A pipelined hybrid recommender system for ranking the items on the display* [Conference Presention]. In Proceedings of the Workshop on ACM Recommender Systems Challenge (pp. 1-5). Copenhagen Denmark. http://doi.org/10.1145/3359555.3359565

Rajarajeswari, S., Naik, S., Srikant, S., Prakash, M. S., & Uday, P. (2019). Movie Recommendation System. *In Emerging Research in Computing, Information, Communication and Applications* (pp. 329-340). http://doi.org/10.1007/978-981-13-5953-8\_28

Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. *In Smart Intelligent Computing and Applications* (pp. 391-397). http://doi.org/10.1007/978-981-13-1927-3\_42

Schick, A. G., Gordon, L. A., & Haka, S. (1990). Information overload: A temporal approach. *Accounting, Organizations and Society, 15*(3), 199-220. http://doi.org/10.1016/0361-3682(90)90005-F

Schneider, S. C. (1987). Information overload: Causes and consequences. *Human Systems Management, 7*(2), 143-153. http://doi.org/10.1016/0361-3682(90)90005-F

Shah, K., Salunke, A., Dongare, S., & Antala, K. (2017, March). *Recommender systems: An overview of different approaches to recommendations* [Conference Presention]. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS) (pp. 1-4). Coimbatore, India. http://doi.org/10.1109/ICIIECS.2017.8276172

Sharma, M., & Mann, S. (2013). A survey of recommender systems: approaches and limitations. *International journal of innovations in engineering and technology, 2*(2), 8-14.

Soni, K., Goyal, R., Vadera, B., & More, S. (2017). A Three Way Hybrid Movie Recommendation Syste. *International Journal of Computer Applications, 160*(9). http://doi.org/10.5120/ijca2017913026

Stubbs, M. (2001). *Words and phrases: Corpus studies of lexical semantics*. Oxford: Blackwell publishers. http://doi.org/10.1075/ijcl.7.2.07ber

Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., & Vijayakumar, V. (2017). A personalised movie recommendation system based on collaborative filtering. *International Journal of High Performance Computing and Networking, 10*(1-2), 54-63. http://doi.org/10.1504/IJHPCN.2017.083199

Suganeshwari, G., & Ibrahim, S. S. (2016, March). *A survey on collaborative filtering based recommendation system* [Conference Presention]. *In Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges* (ISBCC–16’) (pp. 503-518). VIT University, India. https://doi.org/10.1007/978-3-319-30348-2\_42

Sun, B., & Dong, L. (2017). Dynamic model adaptive to user interest drift based on cluster and nearest neighbors. *IEEE access, 5*, 1682-1691. http://doi.org/10.1109/ACCESS.2017.2669243

Tian, Y., Zheng, B., Wang, Y., Zhang, Y., & Wu, Q. (2019). College library personalized recommendation system based on hybrid recommendation algorithm. *Procedia CIRP, 83*, 490-494. http://doi.org/10.1016/j.procir.2019.04.126

Uluyagmur, M., Cataltepe, Z., & Tayfur, E. (2012, October). *Content-based movie recommendation using different feature sets* [Conference Presention]. In Proceedings of the World Congress on Engineering and Computer Science. San Francisco, USA. http://www.iaeng.org/publication/WCECS2012/WCECS2012\_pp517-521.pdf

Walek, B., & Fojtik, V. (2020). A hybrid recommender system for recommending relevant movies using an expert system. *Expert Systems with Applications*, *158*, 113452. http://doi.org/10.1016/j.eswa.2020.113452

Wang, W. Y., & Lobato, R. (2019). Chinese video streaming services in the context of global platform studies. *Chinese Journal of Communication, 12*(3), 356-371. http://doi.org/10.1080/17544750.2019.1584119

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). *Attention-based LSTM for aspect-level sentiment classification*[Conference presentation]. In Proceedings of the 2016 conference on empirical methods in natural language processing. Beijing, China. https://www.aclweb.org/anthology/D16-1058.pdf

Wang, Y., Wang, M., & Xu, W. (2018). A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework. *Wireless Communications and Mobile Computing, 2018*. http://doi.org/10.1155/2018/8263704

Wei, S., Zheng, X., Chen, D., & Chen, C. (2016). A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications, 18*, 83-94. http://doi.org/10.1016/j.elerap.2016.01.003

Yi, J., Nasukawa, T., Bunescu, R., & Niblack, W. (2003, November). *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques* [Conference Presention]. In Third IEEE international conference on data mining. Melbourne, FL, USA. http://doi.org/10.1109/ICDM.2003.1250949

Zhang, H., Ji, Y., Li, J., & Ye, Y. (2015). A triple wing harmonium model for movie recommendation. *IEEE Transactions on Industrial Informatics, 12*(1), 231-239. http://doi.org/10.1109/TII.2015.2475218

Zhang, R., & Mao, Y. (2019). Movie Recommendation via Markovian Factorization of Matrix Processes. *IEEE Access, 7*, 13189-13199. http://doi.org/10.1109/ACCESS.2019.2892289

Zhang, Y., & Liu, Y. (2010, April). *A collaborative filtering algorithm based on time period partition*[Conference presentation]. In 2010 Third International Symposium on Intelligent Information Technology and Security Informatics. Jinggangshan, China. http://doi.org/10.1109/IITSI.2010.161