Machine Learning Movie Based Recommendations: A Mobile-Friendly Approach

Vishal Maisuria

26439978

26439978@students.lincoln.ac.uk

University of Lincoln, School of Engineering and Physical Sciences

College of Health and Science

University of Lincoln

Submitted in partial fulfilment of the requirements for the
Degree of BSc (Hons) Computer Science

Supervisor: Saeid Pourroostaei Ardakani

May 2025

Acknowledgements

I would like to thank my mother and brother for being there for me and supporting me throughout my studies.

Abstract

Here is the abstract for this project report.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Technology is evolving constantly at a rapid pace, and with this comes more options for users to customise and tailor experiences. This is especially true in the entertainment industry, where streaming services are becoming more popular. The increasing complexity and scale of digital content libraries can overwhelm users seeking entertainment that suits them. However, this comes with the problem of choice overload, when the numbers of choice increase, cognitive resources are strained, and this results in decreased decision-making efficiency, and decision fatigue (Chernev Et al., 2015). Although a large assortment might lead to stronger preferences, allowing users to maintain flexibility, an important drawback is a behavioural consequence that can be the increased likelihood of deferring choice. Users with more options are more likely to be overwhelmed and not make a choice altogether, as more choice fuels the expectations to find the perfect satisfaction of needs, which, if not met, leads to the experience of regret and dissatisfaction (Korff and Böhme, n.d.). The relationship between the number of options and the likelihood of making a choice is not linear, as the more options there are, the less likely a user is to make a choice. This is referred to as the "paradox of choice" (Schwartz, 2015). Traditional search and filter methods lack the personalisation that users want to refine the vast options to find a movie that suits their needs. By implementing a machine learning recommendation system in an intuitive mobile application, this project addresses a need for user-centered options. This will not only enhance engagement and satisfaction, but will allow users to enjoy movies that they might not have thought would appeal to them.

1.2 Aims and Objectives

This project aims to create a machine learning system that will recommend movies to users based on their input from a chat-based interface on a mobile application. This input can vary from genre, to more specific requests like directors and actors. To achieve this, the machine learning will be trained on a dataset of movies, to comprehend the genres and cast of the movies. To achieve these aims in an efficient manner, the following objectives were created, so that a systematic approach can be followed:

1. Analyse the dataset chosen, ensuring accessibility and usability.
   a. Ensure movies are present, and the dataset can be used for the training and reference of the machine learning system.
2. Create a machine learning system that can output values with similar properties as the input, and train it to recognise a film's properties, for example genre and cast, so relevancy can be found based of these metrics.
   a. This will use the most appropriate machine learning model for the task, found using evaluation metrics, and will be trained on the dataset chosen.

3. Perform a literature review on ML and user experience.
   a. This will be used to review current literature on machine learning and the efficiency it has on mobile devices, and how this relates to user experience.
4. Create a front-end mobile application.
   a. Consider the accessibility, compatibility and accessible power that mobile devices have.
   b. This will be a chat-based interface, using natural language processing (NLP), where users can input requests and receive recommendations.
   c. Integrate the machine learning system with the application using TinyML, so that the model can be run on the device, and the data processed locally.

1.3 Dissertation Structure

To inform the final outcomes of the project, cutting-edge methods and literature will be explored to understand how to best approach this problem, along with a thorough discussion of the development of the application, analysing the tools required, the process of implementation, challenges encountered and the steps taken to resolve these.

Chapter 2

Literature Review

2.1 Overcoming Choice Overload

There have been multiple theories on combatting this problem. In terms of entertainment, recognising popular genres and user preferences is vital to providing a tailored experience without any overwhelming emotion.

The first method is to have a sequential choice architecture, where users are presented with subsets of options in multiple rounds, retaining one from each round until the final choice is made (Besedeš Et al., n.d.). This architecture was found to be the most effective in improving the decision making quality, by reducing cognitive load. It also reduced bias by deffering the final choice until all the options were considered. This method does not utilise machine learning, but instead focuses on the user making choices through a case of "rounds" until the final choice is made. While this method may be effective, it can be time-consuming, with users being demotivated to process through many films to reach a final decision.

Another way to combat this is to implement a deep-learning based model that captures user-specific context to tailer recommendations directly (Zhou Et al., 2020). This model learns from the user and then based on that, returns appropriate recommednations. However, it is also important to note that user preferences are not static, and can change over time. As such, recommendations that utilise a "first learn, then earn" approach may beocme less effective over time.

2.2 Machine Learning

Machine learning is on the rise in applications, with the ability to learn from data and implement improvements automatically, improving efficiency and effectiveness of applications. Integrating machine learning into an application can provide many benefits when it comes to the user experience.

2.2.1 Machine learning and user experience

Machine learning in applications can offer an increased personalisation and customisation of the user experience, by learning from user preferences and environmental factors (Carmona Et al., 2018).

When it comes to developing an application with machine learning, there are two fundamental roles of UX (user experience) in ML systems: Problem setting and Problem solving. Problem setting is the process of identifying the right ML application by focussing on user-centered problems, and problem solving is the process of designing ML systems that produce outputs that align with the user needs (Yang, n.d.). Problem solving is the most important, as this will need to be tailored to the user, and in my case output recommendations, otherwise this will be unsatisfactory. Developing an application combining both user experience

and machine learning is complex, and there are some challenges to consider, such as the mismatch between the goals of both. Machine learning improves the experience of users through the use of data and algorithms, whereas UX focusses on the contextual relevance. It is also important that users trust the outputs from ML, and having strategies to handle ML errors can help with this.

Alongside the trust of applications, it is important to consider the privacy concerns which ties into the experience and satisfaction of the user, as any confusion with how their data is processed will negatively affect experience. If malicious users were able to recover the data used by these models, they could infer sensitive information about the user, which could be a serious issue (Cristofaro, 2020).

2.2.2 Machine Learning in a Mobile Application

Mobile applications are already a staple in lifestyles, users can access information quickly and easily, as applications are constantly optimised to be lightweight and efficient to run using a cellular data connection and limited power. This is currently a problem when it comes to machine learning, as this is a resource intensive process if you were to run all programs and processes locally on the device. ChatGPT uses a server-based model called server-sent events (SSE). This allows the model to be run on a server, and the mobile application sends requests to the server, which then returns the response. This is a more efficient way to run machine learning models on mobile devices, as it reduces the amount of processing power needed on the device itself (Gitonga, 2024). This is just one of the options that this can allow for efficient machine learning, while maintaining resources.

Another way to having a positive machine learning experience is to utilise TinyML, a paradigm of running machine learning on embedded edge devices with limited computational, memory and power resources (Ray, 2022). The goal of TinyML is to minimise energy consumption, optimise memory usage and achieve acceptable accuracy. This is beneficial for my project, as this will be a mobile application that utilises machine learning. This will allow for the application to run efficiently on the device, without using too much power or memory, and also be beneficial in terms of privacy, as the data will be processed locally on the user's device, as opposed to sending sensitive information to a server. TinyML is scalable as well, as it has the ability to function without constant internet connectivity (Schizas Et al., 2022).

Machine learning is constantly evolving, and TinyML will be cructial to next technologies. For instance, many of the crucial parts of augmented reality glasses are constantly active and powered by batteries. Such devices will benefit greatly from developments in TinyML.

2.3 Data preprocessing in mobile applications

Preprocessing plays a critical role in improving model performance on devices, espeically those with limited computation power, and memory (Haseeb Et al., 2024). Studies using K-Nearest Neighbours, Support Vector Machines (SVM) and other machine learning algorithms combined with various preprocessing techniques have shown that model performance improves with preprocessing,

and time taken is reduces as a result, which is beneficial for power restricted devices and situations.

There are many data preprocessing techniques that I can utilise for my model that will improve performance and result in a high accuracy and execute in little time. A combination of techniques to consider is Random Forest, utilised with Boruta-selected features. This combination on the CICIDS2017 dataset resulted in an accuracy value of 99.99% (Mishra Et al., 2025).

2.4 Deep learning VS Traditional Machine Learning

Deep learning is a subset of machine learning that uses multi-layered neural networks to automatically learn features from data. Unlike traditional machine learning algorithms, which require manual feature engineering, deep learning models can autonomously extract and transform features from raw data, making them effective for unstructured data (Taye, 2023).

Traditional machine learning approaches, including algorithms like Support Vector Machines (SVM) rely on predefined features and are generally more computationally efficient. However, they may struggle with capturing complex patterns in data, such as emotions regarding movies, especially when dealing with large-scale datasets like the movie databases used in this project.

In a comparative study, traditional machine learning algorithms were evaluated against a Convolutional Neural Network (CNN) for text classification tasks (Kamath Et al., 2018). The study utilised two datasets: a proprietary health insurance dataset and the publicly available Tobaccy-3482 dataset. Traditional ML models, including Logistic Regression, SVM, Naive Bayes and Random Forest, were implemented using bag-of-words representations. In contrast, the CNN model employed word embeddings and convolutional layers to capture semantic relationships in the text.

The findings revealed that the CNN outperformed traditional ML models across both datasets. Specifically, the CNN achieved an accuracy of 96% on the processed Tobacco dataset and 89.27% on the processed Health dataset. In comparison, the best-performing traditional model, Logistic Regression, achieved 81% and 77% accuracy respectively.

Despite these advantages however, deep learning models require substantial computational resources, longer training times and careful tuning of hyperparameters. While traditional machine learning mdoels offer simplicity and efficiency, deep learning approaches provide superior performance in handling compelx and unstructured data. The drawbacks that deep learning models have can be offset through the use of TinyML, which allows for the deployment of deep learning models using lightweight modifications. This results in an effective model that is also efficient to be run on devices with resource constraints.

2.5 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence concerned with the interaction between computers and human language. It focusses on enabling machines to understand, interpret and generate natural language text or speech in a meaningful way.

In recent years, the development of deep learning models has drastically improved the performance of NLP systems. One of these is BERT (Bidirectional Encoder Representations from Transformers) (Gardazi Et al., 2025). BERT is a pre-trained transformer model that learns contextual representations of word by processing text bidirectionally (Devlin Et al., 2019). Unlike earlier models that read text in a single direction (left-to-right or right-to-left), BERT's ability to consider the context from both directions has proven highly effective in a variety of language understanding tasks, such as sentiment analysis and question answering.

However, BERT's large size and high computational demands can be impractical for many real-world applications, especially when considering the project aim to have a machine learning chatbot system on a mobile device, which has limited resources. To address this, there are optimised variants of BERT, such as DistilBERT (Sanh Et al., 2020), which retains around 97% of BERT's performance while reducing the model size by 40% and increasing inference speed by 60%. This balance of performance and efficiency makes DistilBERT a compelling option for deployment in resource-constrained environments.

The integration of BERT and its lightweight variants such as DistilBERT into mobile applications represent a major advancement over traditional models. This not only allows for more accurate and context-aware recommendations, but also sentiment analysis, which can be used to gauge user preferences and improve user experience.

2.6 Cross-Platform Development and Use of Expo

Cross-platform development has emerged as a compelling solution to the challenges posed by native mobile app development. Traditional native applications require separate implementations for each target platform - commonly iOS and Android - which leads to increased development time, higher maintenance consts, and platform-specific expertise. In contrast, cross-platform frameworks allow developers to write a single codebase that runs across multiple platforms, streamlining the development lifecycle. For a project being developed by a single person, this approach is particularly advantageous.

In a study, four major cross-platform development models were identified: web, hybrid, interpreted and generated apps (Xanthopoulos and Xinogalos, 2013). Using frameworks such as Expo (Expo, n.d.) aligns with the hybrid approach, where applications are primarlily written using widespread web technologies such as JavaScript, HTML5 and CSS. These apps are wrapped in a native container that provides access to device APIs, simulating a native experience. One of the key benefits of hybrid apps is their ability to leverage a single technology stack - namely JavaScript - to develop applications that are deployable through official app stores like Google Play and Apple App Store. This enables easier onboarding for developers who are already familiar with web technologies.

However, this comes with its limitations. The performance of hybrid applications is generally lower compared to that of fully native apps, especially when handling complex animations or resource-intensive operations. While frameworks like Expo offer access to a broad set of native features through JavaScript APIs, there can still be constraints. For example, if a specific device API is not yet supported

by the framework, developers must either wait for community or official support or eject from the managed workflow to write native code manually. There are also potential issues with app store compliance. Applications that do not adhere to the guidelines of the respective app stores may face rejection, particularly by Apple's App Store, which maintains strict guidelines on user experience.

The use of Expo as a hybrid cross-platform development tool offers several benefits: reduced development overhead, widespread technology adoption, and the convenience of deploying to multiple platforms.

Chapter 3

Requirements Analysis

3.1 Functional Requirements

3.2 Non-Functional Requirements

For this project, several resources, both hardware and software, are required to successfully and efficiently build the desired artefact.

3.3 Hardware Requirements

1. A computer setup with a graphical processing unit (GPU) capable of training a machine learning model on datasets with a size of around 4GB. To make this easier, it may be beneficial to utilise a computer with more power, such as a desktop capable of running 'compute unified device architecture' (CUDA), which helps speed up applications by harnessing the power of GPU accelerators.
2. A mobile device for testing of efficiency and functionality. There is no requirement for a specific device, as this project will be developed using cross-platform frameworks.

3.4 Software Requirements

To develop this project, several software tools and frameworks will be required in order to achieve the desired outcome of an efficient and effective artefact.

3.4.1 Python

Python is a high-level programming language that is widely used in the field of machine learning. There are a large number of open-source and up to date libraries that make the development of machine learning models much more efficient. Python is considered the fastest-growing programming language due to its simplicity, versatility and deep integration with data science workflows (Srinath, n.d.), and as of 2022, it was the second most popular language used on Github (Fig. 1). Part of Python's popularity is due to the fact that it is easier to read compared to other languages. Rather than punctuation, Python uses English keyboards, and line breaks help define code blocks (Scarlett, 2023). In practice, this helps with debugging as it is easier to understand what lines of code are responsible for. This allows for rapid prototyping and iteration, which would be much more difficult in other languages like Java or C.

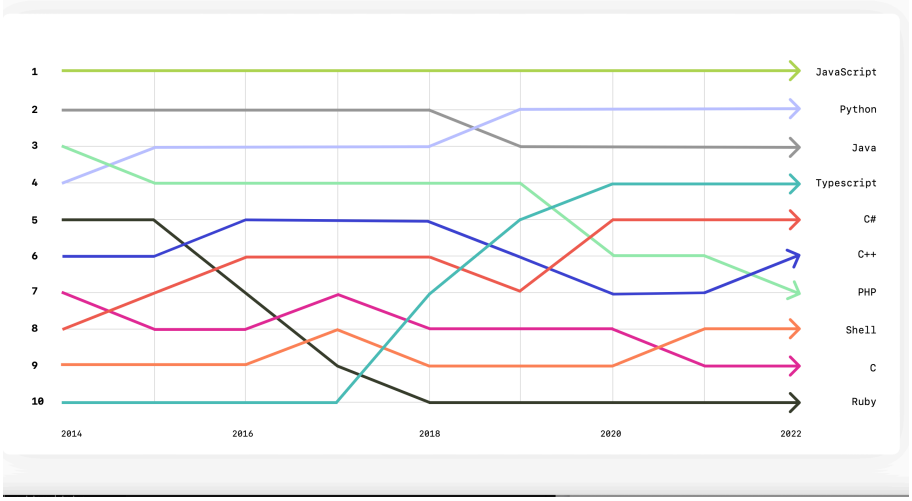

Fig.1. "The top programming languages on Github in 2022", (The top programming languages, n.d.)

3.4.2 Typescript

To develop the mobile application's frontend, TypeScript was selected due to its robust features that enhance code quality, maintainability, and developer productivity. As a statically typed superset of Javascript, TypeScript introduces compile-time type checking, which allows developers to catch errors early, resulting in fewer runtime bugs and more stable code. This is particularly beneficial in mobile development, where user experience can be negatively affected by subtle bugs or crashes. Static typing in TypeScript significantly improves code understandability and defect detection, contributing to overall software reliability and maintainability (Top 6 Benefits of Implementing TypeScript, n.d.).

Moreover, TypeScript integrates seamlessly with React Native, a popular framework used for building cross-platform mobile applications. This allows for reusable components, strong typing and improved development tooling, such as intelligent code completion and refactoring support in modern IDEs (Crudu, 2024). These features contribute to a more efficient development cycle and faster debugging, both of which are crucial for delivering high-performance applications. Additionally, a comparative study highlights that developers using TypeScript experienced fewer regressions and were more productive compared to those using JavaScript alone, especially in large-scale projects, for example Airbnb's migration to TypeScript resulted in a significan reduction in bugs and improved developer productivity (Sekhar Emmanni, 2021).

Chapter 4

Design and Methodology

Chapter 5

Methodology

Chapter 6

Results and Discussion

Chapter 7

Conclusion

Appendices

ReferencesAbbas, A. M. H., Ghauth, K. I. and Ting, C.-Y. (2022) User Experience Design Using Machine Learning: A Systematic Review. *IEEE Access,* 10 51501–51514. Available from https://ieeexplore.ieee.org/document/9770797/?arnumber=9770797 [accessed 21 January 2025]

Besedeš, T., Deck, C., Sarangi, S. and Shor, M. (n.d.) *Reducing Choice Overload without Reducing Choices.*

Carmona, K., Finley, E. and Li, M. (2018) The Relationship Between User Experience and Machine Learning. *SSRN Electronic Journal,* Available from https://www.ssrn.com/abstract=3173932 [accessed 21 January 2025]

Chernev, A., Böckenholt, U. and Goodman, J. (2015) Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology,* 25(2) 333–358. Available from https://onlinelibrary.wiley.com/doi/abs/10.1016/j.jcps.2014.08.002 [accessed 12 January 2025]

Cristofaro, E. D. (2020) *An Overview of Privacy in Machine Learning* Available from http://arxiv.org/abs/2005.08679 [accessed 21 January 2025]

Crudu, V. (2024) *Beyond Web Development Exploring Typescript in Mobile Applications* Available from https://moldstud.com/articles/p-beyond-web-development-exploring-typescript-in-mobile-applications [accessed 22 April 2025]

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019) *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* Available from http://arxiv.org/abs/1810.04805 [accessed 21 April 2025]

Expo (n.d.) Available from https://expo.dev/ [accessed 21 April 2025]

Ganesan, V. (2022) Machine Learning in Mobile Applications. *International Journal of Computer Science and Mobile Computing,* 11(2) 110–118. Available from https://ijcsmc.com/docs/papers/February2022/V11I2202217.pdf [accessed 21 January 2025]

Gardazi, N. M., Daud, A., Malik, M. K., Bukhari, A., Alsahfi, T. and Alshemaimri, B. (2025) BERT applications in natural language processing: a review. *Artificial Intelligence Review,* 58(6) 166. Available from https://link.springer.com/10.1007/s10462-025-11162-5 [accessed 21 April 2025]

Gitonga, M. (2024) *Real-time Web with Server Sent Events* Available from https://medium.com/@maryanngitonga/real-time-web-with-server-sent-events-84a335ac1856 [accessed 21 January 2025]

Haseeb, A., Cleland, I., Nugent, C. and McLaughlin, J. (2024) Optimizing Machine Learning for ResourceConstrained Devices: A Comparative Analysis of Preprocessing Techniques and Machine Learning Algorithms. In: *2024 35th Irish Signals and Systems Conference (ISSC).* June 2024 1–5

Kamath, C. N., Bukhari, S. S. and Dengel, A. (2018) Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification. In: *Proceedings of the ACM Symposium on Document Engineering 2018.* August 2018 Halifax NS Canada: ACM, 1–11

Korff, S. and Böhme, R. (n.d.) *Too Much Choice: End-User Privacy Decisions in the Context of Choice Proliferation.*

Mishra, S., Anithakumari, T., Sahay, R., Shrivastava, R. K., Mohanty, S. N. and Shahid, A. H. (2025) LIRAD: lightweight tree-based approaches on resource constrained IoT devices for attack detection. *Cluster Computing,* 28(2) 140. Available from https://link.springer.com/10.1007/s10586-024-04792-x [accessed 21 January 2025]

Oh, F. (2012) *What Is CUDA?* Available from https://blogs.nvidia.com/blog/what-is-cuda-2/ [accessed 22 April 2025]

Ray, P. P. (2022) A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences,* 34(4) 1595–1623. Available from https://linkinghub.elsevier.com/retrieve/pii/S1319157821003335 [accessed 21 January 2025]

Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2020) *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter* Available from http://arxiv.org/abs/1910.01108 [accessed 21 April 2025]

Scarlett, R. (2023) *Why Python keeps growing, explained* Available from https://github.blog/developer-skills/programming-languages-and-frameworks/why-python-keeps-growing-explained/ [accessed 22 April 2025]

Schizas, N., Karras, A., Karras, C. and Sioutas, S. (2022) TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review. *Future Internet,* 14(12) 363. Available from https://www.mdpi.com/1999-5903/14/12/363 [accessed 21 January 2025]

Schwartz, B. (2015) The Paradox of Choice. Positive Psychology in Practice 121–138.

Sekhar Emmanni, P. (2021) The Role of TypeScript in Enhancing Development with Modern JavaScript Frameworks. *International Journal of Science and Research (IJSR),* 10(2) 1738–1741. Available from https://www.ijsr.net/getabstract.php?paperid=SR24401234212 [accessed 22 April 2025]

Srinath, K. R. (n.d.) *Python – The Fastest Growing Programming Language.* 4(12).

Taye, M. M. (2023) Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers,* 12(5) 91. Available from https://www.mdpi.com/2073-431X/12/5/91 [accessed 21 April 2025]

The top programming languages (n.d.) Available from https://octoverse.github.com/2022/top-programming-languages [accessed 22 April 2025]

Top 6 Benefits of Implementing TypeScript (n.d.) Available from https://strapi.io/blog/benefits-of-typescript# [accessed 22 April 2025]

Tsoukas, V., Gkogkidis, A., Boumpa, E. and Kakarountas, A. (2024) A Review on the emerging technology of TinyML. *ACM Computing Surveys,* 56(10) 1–37. Available from https://dl.acm.org/doi/10.1145/3661820 [accessed 21 January 2025]

Xanthopoulos, S. and Xinogalos, S. (2013) A comparative analysis of cross-platform development approaches for mobile applications. In: *Proceedings of the 6th Balkan Conference in Informatics.* September 2013 Thessaloniki Greece: ACM, 213–220

Yang, Q. (n.d.) *The Role of Design in Creating Machine-Learning-Enhanced User Experience.*

Zhou, T., Wang, Y., Lu, Yan and Tan, Y. (2020) *Spoiled for Choice? Personalized Recommendation for Healthcare Decisions: A Multi-Armed Bandit Approach* Available from http://arxiv.org/abs/2009.06108 [accessed 12 January 2025]