DATA MINING TECHNIQUES BASED BUILDING INTELLIGENT SHOPPING FOR WEB SERVICES

MUHAMMAD AMIR BIN ABDUL RAZAK

BACHELOR IN COMPUTER SCIENCE (Honours)
MANAGEMENT AND SCIENCE UNIVERSITY

For office use only (Leave this blank)

☐ Approved ☐ Rejected

☐ Approved with changes

Evaluated by:

Name:

Date:

ABSTRACT

The world's technology is rapidly growing and advancing, and with that, new problems arise. The idea of this intelligent shopping system based on data mining is inspired by the problems that have arisen over the years, both from the consumer's side and from the supplier's side. The main objective of this project is to increase the supplier's side profit from selling through an online shopping platform, and to increase the quality of the user experience from the consumer's side. Furthermore, with the findings of this project, this research can contribute to the general advancement of areas of study such as machine learning and data science in general. By using multiple technologies ranging from supervised learning to web scraping, this project aims to optimise the general experience of online shopping in general by implementing new technologies into play. Using IDEs such as PyCharm and text editors such as Atom, this product is mainly developed through hand-made code. By using the Agile Development Method, this project aims to deploy and seamlessly enter the market in an efficient manner.

ACKNOWLEDGEMENTS

First of all, I am grateful to Allah the almighty for the opportunity to complete this Final Year Project titled *Data Mining Techniques Based Building Intelligent Shopping for Web Services*.

Therefore, I would like to express my gratitude to my supervisors Prof. Dato' Dr. Md Gapar Mohd Johar and Assoc. Prof. Dr. Muhammad Hazim Alkawaz for guiding me throughout this entire journey by giving valuable advice that helped realign this project and add value to it. I would also like to thank Dr. Omar Ismail Ibrahim for evaluating this project initially and giving valuable feedback for the short time that he has seen the project so that the thesis would progress as planned. I would also like to give appreciation to Sir Kevin Loo Teow Aik for his insightful teachings on UML diagrams to aid the process of designing the system.

In addition, huge thanks to all the lecturers, family, and friends who were willing to give their opinions throughout the construction of this project. For the final piece, I send all my gratitude towards those who have helped directly and indirectly to this project.

DEFINITION OF TERMS

ML - Machine Learning

DS - Data Science

DM - Data Mining

SEM - Structural Equation Modelling

HTTP - Hypertext Transport Protocol

XML - Extensible Markup Language

JSON - JavaScript Object Notation

SOA - Service Oriented Architecture

AI - Artificial Intelligence

CRO - Conversion Rate Optimization

IoT - Internet of Things

RNN - Recurrent Neural Network

OSS - Online Shopping System

GUI - Graphical User Interface

IDE - Intelligent Development Environment

UML - Unified Modified Language

UCI - University of California Irvine

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter explains the project background, problem identification, project goals and objectives, the project scope, assumptions of the outcomes, limitations of it, and the significance of the project.

1.1 Project Background

E-commerce and supermarket shopping is an ideal environment to deploy pervasive computer assistance and to explore its applications. As e-commerce is a mix between traditional business models and network technology, it gives room for a lot of opportunities but also challenges (Fu et al., 2020). Every week, shoppers from all scopes of the internet order millions of items, on Amazon alone, where ultimately the customer gets away with a few items each purchasing session. Several studies have discovered potential implications of intelligent shopping on physical environments (Bohnenberger et al., 2005; Cumby et al., 2005, Islam et al., 2019).

Intelligent shopping for web services, in its simplest term, is aided online shopping behaviour using an intelligent model. Intelligent shopping is a heavily branched subject, with multiple studies exploring the implications of intelligent shopping through analysing demographic factors on online purchases, having a study in the context of online shopping use this base to measure the predictive power of online shopping through online purchases (Chang, Cheung & Lai, 2005; Mittal, 2021) for web services. Online intelligent shopping is directly related to web services as they act as a medium to execute the action.

A web service is a web technology similar to HTTP, capable of sharing file formats such as XML and JSON. Object-oriented web-based interfaces connected to a database server are a commonly provided utility by a web service. Due to the potential of branching out using web services, it is now widely used nowadays everywhere on the internet. Quoting Alistair Barros , more and more organisations are turning into Service-Oriented-Architecture (SOA) as a strategy to repurpose legacy applications and combine them with new ones. With the rise of web service infrastructures, web service providers are interconnecting their offerings, giving room for web service ecosystems to emerge (Barros et al., 2006).

Due to the abundance of information taken from multiple web services when conducting an intelligent shopping system, data mining (DM) becomes an important asset. In general, DM is the process of extraction of discovered data that is deemed to be potentially useful narrowed down using different techniques including knowledge rules, constraints, abnormalities, and regularities from the data sets using pattern recognition technologies as well usage of statistical and mathematical principles (Frawley et al. 1992; Katz, 1997).

DM is crucial nowadays and they provide amazing features such as allowing analysers to sift through the repetitive and chaotic noise within the large volume of data. At the current stage, the problem only lies in the inability to generate useful information from the large data sets given, and not in lack of data. Due to this phenomenon, DM is gaining traction in terms of importance inside multiple research

areas (Frawley et al. 1992; Katz, 1997; Weiss & Indurkhya, 1998) and is heavily correlated with machine learning (ML) and artificial intelligence (AI).

1.2 Problem Statements

The average amount of decisions made by an American adult is roughly 35,000 on a given day (Sollisch, 2016; Pignatiello et al., 2020). Known as decision fatigue, it occurs from carrying out multiple acts of decision-making and results in impaired ability to make trade-offs and decisions, and often seem impulsive and irrational in making choices (Tierney, 2011; Pignatiello et al., 2020). Due to the information overload on the internet, online shopping becomes a tedious task resulting in decision fatigue from the information overload (Matthew & Joseph, 2014). Other solutions such as Decision Support Systems have been implemented (Kumar, 2020), but in this project, other areas of study will be explored.

In e-commerce, a successful conversion refers to when a customer completes the transaction and fulfils the check out process (Saleem et al., 2019). Low conversion rates are a problem that every web service based e-commerce suffers, where a decent amount of traffic does not translate to a successful conversion (Saleem et al., 2019; Saleem et al., 2019).

The internet has grown immensely from its first recorded user count in December 1995, growing from 558 million users in April 2002 to 5.168 billion users worldwide recorded in March 2021 (Miniwatt Marketing Group, 2022), which led to the growth of online shopping. Statistics have predicted that by 2025, e-commerce will have dominated 24.5% of total global retail sales (Daniel, 2022). Kamber states

that the society currently lives in an era where data is rich but information is poor (Kamber, 2011), and the growth of the internet currently surpasses the growing rate of current technologies on data science and data mining is always growing with increasing frequency everyday (Mughal, 2018).

1.3 Project Objectives

This system aims to use DM as a tool to convert data to information by using techniques from multiple areas of studies such as ML, statistical analysis, and data science (DS). This way, noisy and unorganised data could be converted into information and useful utilities can come out of it.

This project will address the problem surrounding decision fatigue from customers by recommending) relevant products so that users can get their desired product with the least noise as possible from other unrelated products.

Due to the problem regarding low conversion rates of web services in general, this project will try to address this by increasing the Conversion Rate Optimization (CRO) using an intelligent shopping model for web service to entice the customer. As it is relatively hard to predict effectiveness, the aim is to increase the CRO by at least 10%.

1.4 Project Significance

According to statistics, 75.2% of customers only spend time browsing without putting anything in the cart nor ordering any item (Zhao et al., 2019). This project can

contribute by reducing time spent when browsing items which can contribute to less internet usage for society as a whole.

By allowing the user to make decisions quicker, the company can expect a market value increase for the web services that implement this as a result of the intelligent shopping model.

By combining multiple areas of studies and different methodologies for intelligent shopping, this project will contribute to general advancements for DM, ML and intelligent shopping.

1.5 Project Scope

This project revolves around ML and DS to support the intelligent shopping model and caters to two main users, the web service and the users who browse the web services.

The main target would be the web services that would be interested in implementing this online shopping model and would want to enhance their system altogether. The end users who browse through the web are also in high priority as this project will attempt to detail the system to make e-commerce more efficient for them.

The intelligent shopping system that is going to be developed for web services proposes a general recommendation system for the user. The web service will display relevant recommendations on their page to the user and will send notifications

about relevant products and promotions. Users can also query in search items to search for items and the web service will display the relevant search results.

As this project is based on web services, the location would be defined entirely to the demographic of the users of the web service. The system will be operating on web services and are browser independent, meaning you can run it on safari, google chrome and even opera without the functionality being interrupted.

This study will explore areas such as DM, implications of it by integrating other areas as well such as ML, statistical studies, DS, and dive deeper into DM, an example being data preprocessing and clustering within DM methodologies. It also will touch Machine Learning and dive deep into Recommendation Systems using Linear Algebra calculations.

1.6 Assumptions and Project Limitations

This project bases its value upon the assumption that current online shopping web services localised in Malaysia have not integrated any sort of intelligent model, namely with data mining approaches. Success metrics are also dependent upon an assumption that recommendations can be predicted at a high accuracy to the point where the results are feasible in integration within an intelligent shopping ecosystem.

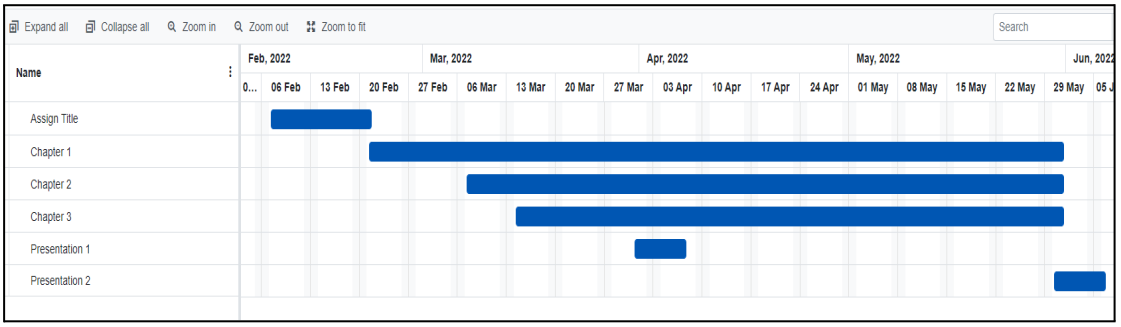
Due to limitations of deep mastery in DM, this project will not be covering advanced areas such as data streams, graphs, time-series and the like which can boost accuracy and gather more information. Due to privacy policies, the proposed project

will not cover areas that intrude on private information such as gender and age, and will rely mainly on previous searches, purchase history, product ratings, and location.

Because of hardware limitations, only a small portion of data will be used for calculation. From the 178 Million Lines of data stored from Amazon's product reviews, only 10,000 reviews for 362 unique products is used for calculation.

1.7 Gantt Chart

Figure 1.1: Gantt Chart for Presentation 1



CHAPTER 2

LITERATURE REVIEW

This section compares five different past products relevant to the project. These projects will feature areas of studies such as web scraping, internet of things (IoT), simple data mining (DM) processes such as association algorithms, machine learning (ML) aspects such as neural networks, and mathematical statistics.

2.1 Introduction

In recent years, due to the inception of the first lockdown order due to the COVID-19 outburst, a certain shift of focus has been set towards online shopping where multiple areas have been integrated into online platforms. Multiple research studies have been performed on online shopping, specifically on shopping behaviour studies where the objective lies on optimising the general experience from the user side and to maximise profit for the vendor's side. However, there have not been much studies on utilising the results of said shopping behaviour studies. This literature review section will give insights upon related products relevant to the proposed study titled *Data Mining Techniques Based Intelligent Shopping for Web Services*.

2.2 Data Mining Techniques

Data mining is a process of mining data and turning it into knowledge from data sources such as databases and data warehouses. This knowledge can be categorised into different areas using different rules and different patterns used for predicting future outcomes or summarising past information. Data mining is a method where organisations detect patterns from rule sets to extract information from data. In

Mughal's study on this, web data mining techniques are explored. Web data mining is related to data mining that is performed on the internet where data is extracted from web pages. Machine Learning supports the process of web data mining and improves the accuracy and is more efficient than traditional methods (Mughal, 2018).

To understand data mining, users need to understand the three main players in the area, Data, Information and Knowledge. In a general sense, data is an unrefined version of unfiltered information. Information on the other hand would be a refined version for data and is useful for analysis. While knowledge resides in the user, and only exists once data and information is applied to human experience (Liew, 2007).

In data mining, several general steps have been set as a baseline to navigate through the complex subject. The first one that would be considered mining would be the pre-processing, where data mining tasks such as preparing and transforming the data into a suitable form takes place. This section aims to reduce data size by removing the noise, normalise the data found, find relations between attributes, and extract features. In this section itself, some techniques such as data cleaning, integration, transformation, and reduction are present (Alasadi et al., 2017).

After preprocessing and transformation takes place algorithms relevant to the subject are applied such as clustering, classification, regression and the like (Alasadi et al., 2017). In the context of classification algorithms, three major steps are involved, these are exploration, pattern identification, and deployment. Data mining algorithms inside this step can go into one of three categories, supervised, unsupervised, or semi-supervised. Most classification algorithms fall into the

supervised technique as labels are mostly known and they have to be classified as such (Neelamegam, 2013).

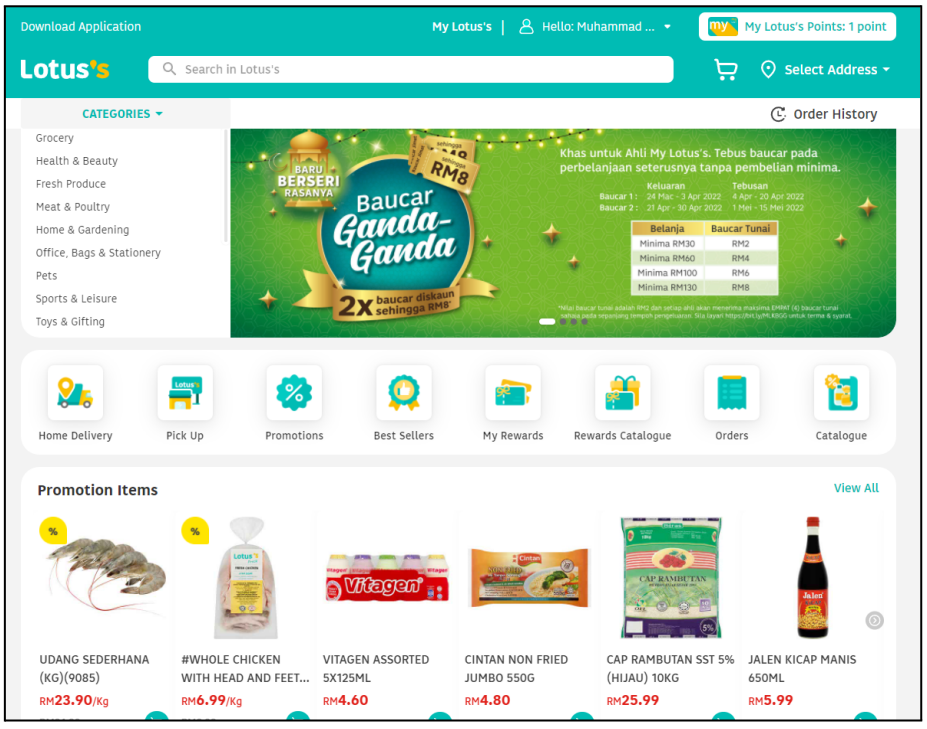
2.3 Online Shopping

In recent times, many traditional physical shops and businesses have transitioned into virtual borderless transactions over the web, referred to as online shopping. Online shopping has been on the rise lately due to the convenience it brings, and due to this companies have invested in this area of technology in forms of brand awareness, customer perception and attitudes (Duffett, 2017; Naseri et al., 2021). Multiple studies on online shopping studies have been performed, showing the significance that it has alongside other growing technologies.

2.3.1 Comparison of Related Products

2.3.1.1 Lotus

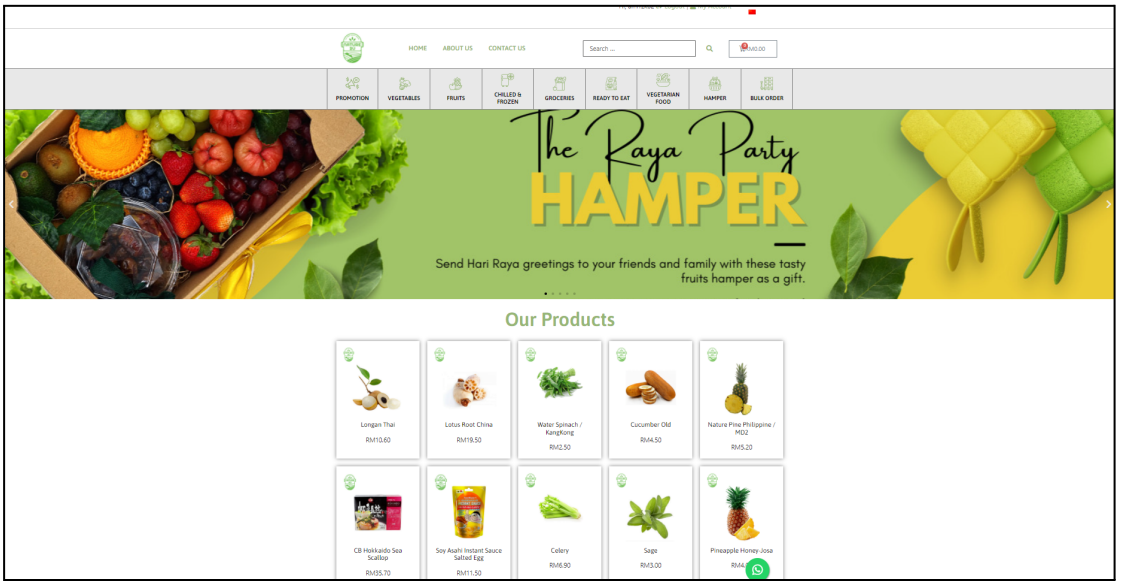
Figure 2.1: Lotus Front Page



Lotus is a local shopping website found in Malaysia, selling grocery goods. Focusing on their website, the current Lotus website focuses on pushing out current promotions and recommendations on the front page based on the best selling products. They have not used that many "smart" implementations inside the Lotus website, and instead they have general marketing tactics to put general promotions and products that are selling well without personalising it to each customer. There are other systems such as rewards catalogue, assumed to be inspired by other systems that offer the same, and are there to attract returning customers.

2.3.1.2 Nature2u

Figure 2.2: Nature2u Front Page



Nature2u is a website selling fresh fruits and vegetables and offers delivery to their customers who live in their country of operation, which is Malaysia. Moving on to the layout of the website and the marketing strategies that it uses, the website seemingly does not have any intelligent shopping implementations other than their promotions tab. Nature2u's promotion tab is situated in a different page, whilst the

main page only features the products without any specific categorisation or customer personalisation.

There are no notification settings present except their email letterlist when a promotion comes up, and there are barely any popup notifications to engage and entice the customer to initiate a purchase. The customer might sense some sense of urgency and be pushed into buying when they see a promotion is ending, but other than that the website does not engage and personalise their catalogue displayed to each customer contrary to how an intelligent shopping website would.

2.3.1.3 MR.DIY

Figure 2.3: MR.DIY Front Page

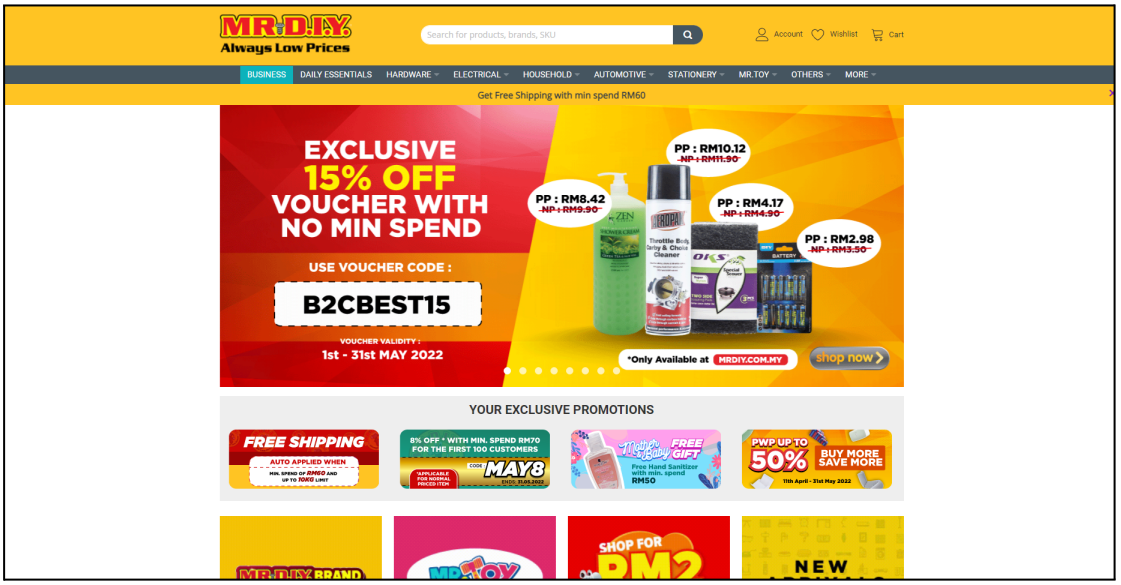
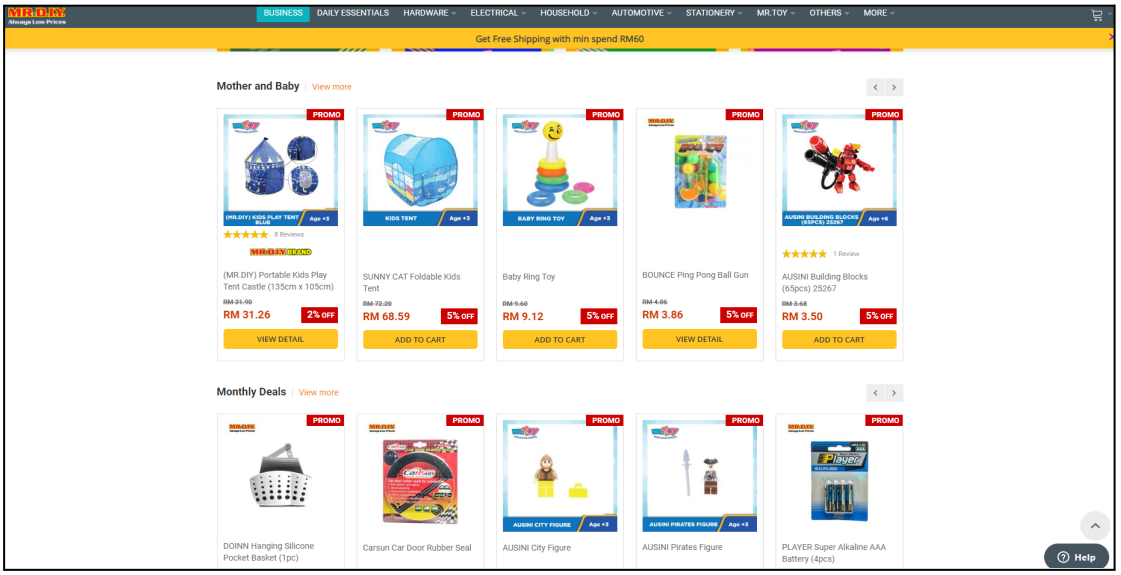


Figure 2.4: MR.DIY Products Page



MR.DIY is a website selling daily goods including mechanical goods such as car parts, tool parts, toilet goods and also even stationeries. The user is greeted with a promotion slider, where you can click on the slider to get to the respective product set on display. The website then follows up with generalised promotions of the month and other products that are on promotion, in this case it's mother and baby products.

There are some flaws with this layout design, the elephant in the room would be the huge promotion slider. MR.DIY has put around two screens worth of slider promotion, and the direct link to promotions and product catalogue is under all the sliders. Instead, a better approach would be putting the promotion sliders on the sides and putting the promotions on top.

The promotions catalogue itself could also use some intelligent mechanism. They could read the demographics of a signed in user and find behavioural patterns, and in this case, they could display mechanical goods instead of mother and baby products, due to the demographic that the website is working with, which is a male

signing in on their website. Other than that, this website is a lot more aggressive on their promotions approach, where their main page is only filled with promotions.

2.3.1.4 Harvey Norman

Figure 2.5: Harvey Norman Front Page

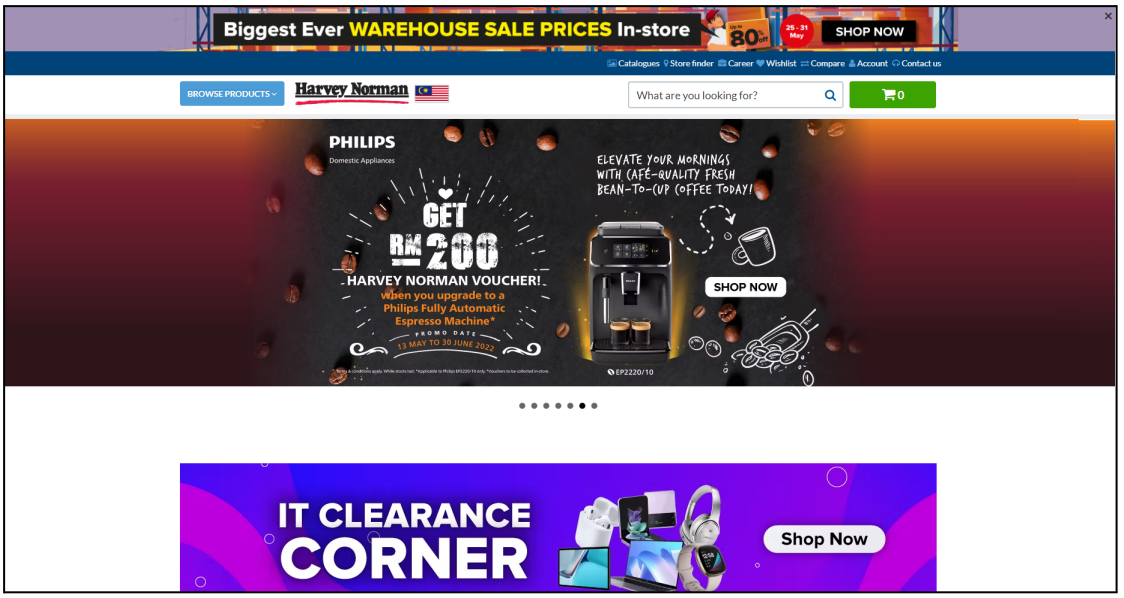
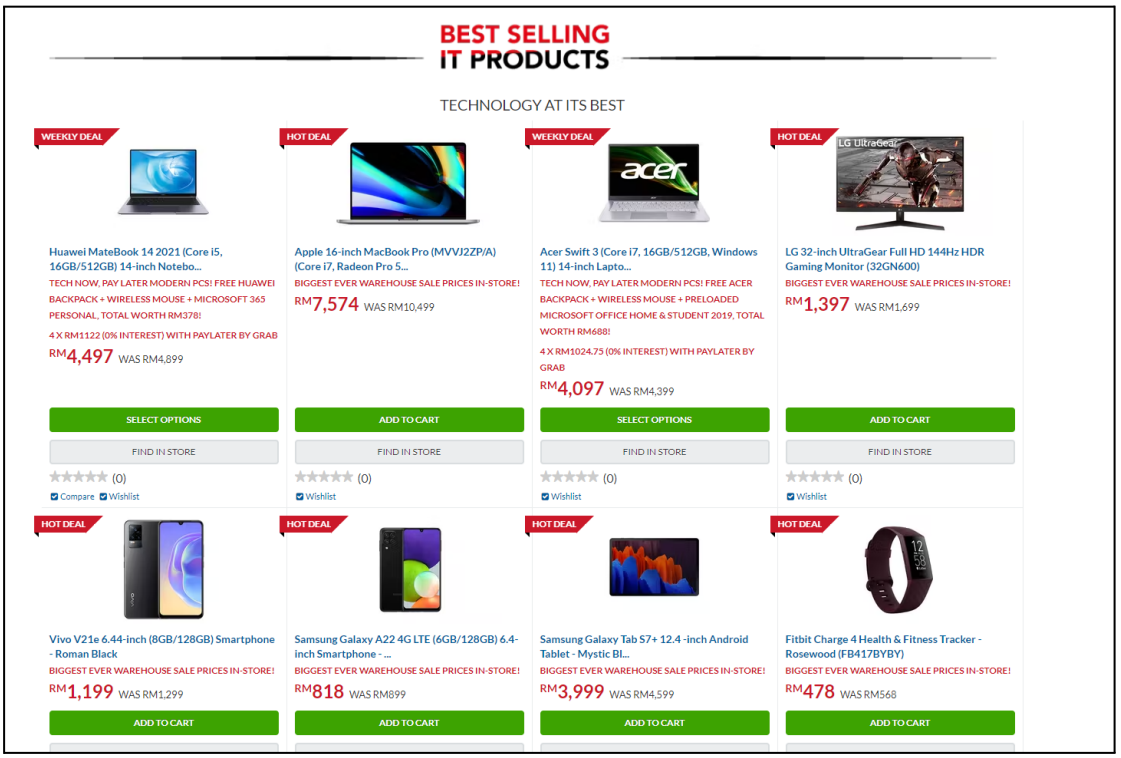


Figure 2.6: Harvey Norman Products Page



Harvey Norman is a company responsible for selling electronic products. The layout of the website page starts with a promotion slider, similar to how MR.DIY operates, then follows with a catalogue of its best selling product. This layout has relatively better user experience than MR.DIY's current layout, having the catalogue page higher and having less clutter with less promotion sliders and banners. It also entices the customer with promotions and best selling products. Other than that, the website does not give out coupons or promotions when the user trajectory is heading onto leaving the website.

2.3.1.5 Shopee

Figure 2.7: Shopee Front Page

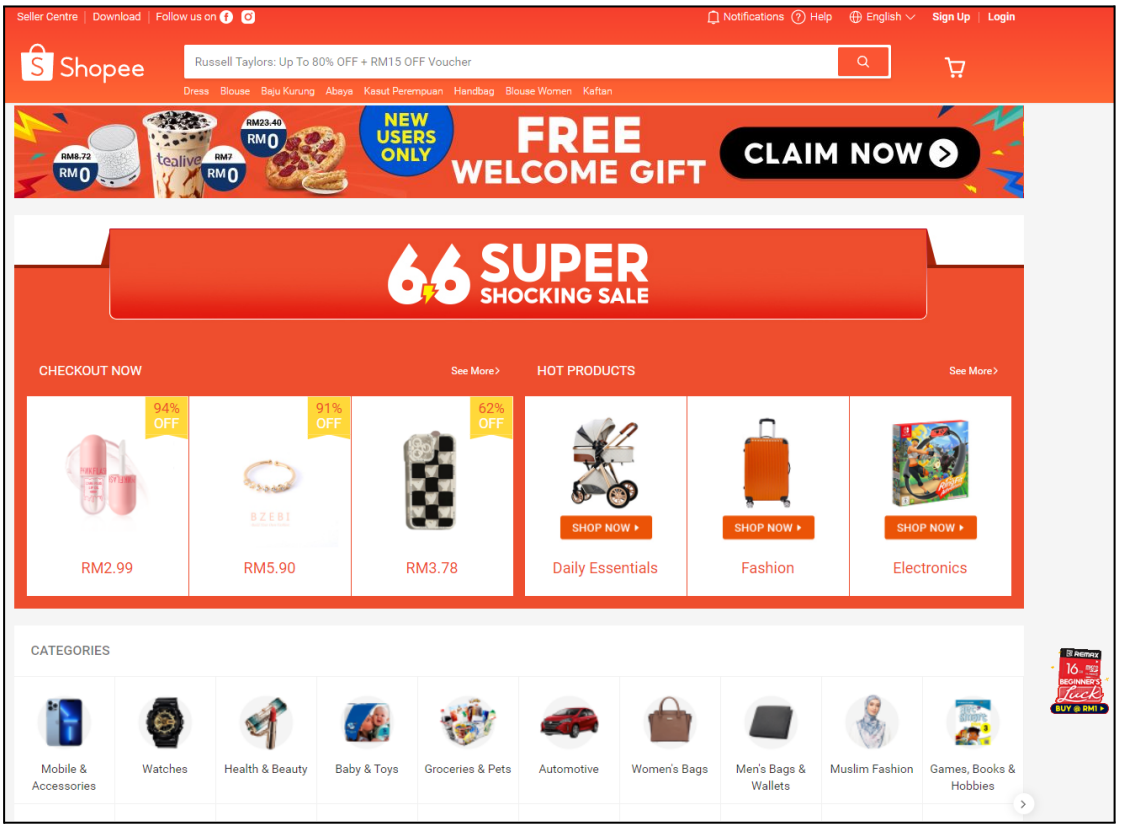
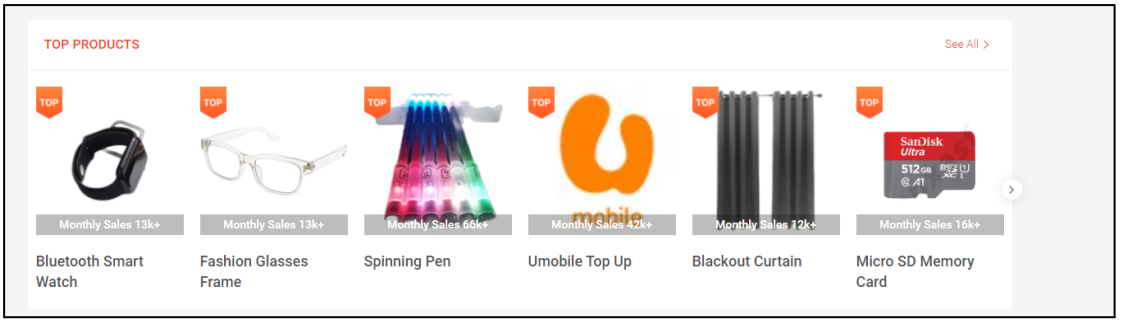


Figure 2.8: Shopee Shocking Sale Tab

Figure 2.9: Shopee Products Tab



Shopee is an online e-commerce platform that sells products of all kinds, and is mainly situated and catered towards the East Asia and SouthEast Asia population. It is a very established website, clocking in at around 54 million users monthly (Statista, 2022). Shopee is utilising a lot of marketing tactics, such as promotion sliders, shocking deals with limited time to incite urgency for the customer to buy it, and daily discovery catalogues based on product activity.

There is no doubt that the intelligent system used by shopee is very effective as the users and customers reflect it. They are using past data to pull out top products that are selling well and aggressively promote it to the customer. Shopee also uses urgency inducing methods such as limited offers which could increase conversion rate optimisation. From my observation, the intelligent shopping system that Shopee has implemented revolves around Apriori algorithms, or any kind of association or classification algorithm under data mining.

Table 2.1: Online Shopping websites feature comparison.

Website Name	**Website Type**	**Regular Promotions**	**Implement Marketing Strategies**	**Intelligent Shopping Recommendations**
Lotus (Lotus, 2022)	Groceries and Daily Goods	✓	×	×
Nature2u (Nature2u, 2022)	Fruits and Vegetables	×	×	×
Mr.DIY (Mr.DIY, 2022)	Hardware Goods	✓	✓	×
Harvey Norman (HN, 2022)	Electronic Items	×	✓	×
Shopee (Shopee, 2022)	All-purpose E-commerce website	✓	✓	✓

2.4 Comparison and Review of Related Methods

2.4.1 Web scraping online shopping system

Mehak et al. and has proposed a tool to exploit filtering processes for smart online shopping usage. In their paper titled *Exploiting Filtering approach with Web Scraping for Smart Online Shopping*, they explore implications of web scraping on smart online shopping. The system searches and scrapes data through HTML DOM-based architecture, using python plugins such as Beautiful Soup 3 and Selenium Web Driver. The system that they built gets data from 5 different websites, where the web content gets scraped from on demand when a user queries their input. To test the credibility of their system, they measured it using an accuracy test through an algorithm and achieved 93% success rate (Mehak et al., 2019).

2.4.2 IoT shopping behaviour analysis system

Fu et al. (2020) have explored a different approach in the same area of study using IoT, instead of web scraping. Under the title *Intelligent decision-making of online shopping behaviour based on internet of things*, this study by Fu et al. (2020) explores public perception on drinking recycled water. Using IoT systems such as grip force and eye tracking sensors, the study used IoT to capture information on college students as college students are more familiar with online shopping and are more willing than adults. Study has set parameters and made the grip limit 15 times as students experience fatigue past that limit to assure unbiased grip strength data gets passed through. For the study's eye tracking model, they measured gaze duration to measure positive response from the subjects. This research can contribute to being used as a guide to avoid distraction of useless information. It is discovered that users

rely more on perception than concrete fact when making decisions about high human contact degree items (Fu et al., 2020).

2.4.3 Machine learning shopping behaviour prediction system

A study by Toth et al. (2017) has explored implications of machine learning on predicting shopping behaviour and online shopping in general, specifically using a mixture of recurrent neural networks (RNNs). Similar to R. Mittal's paper (Mittal, 2021), Toth et al.'s (2017) study titled *Predicting Shopping Behavior with Mixture of RNNs* also focuses on predicting a customer's shopping behaviour. The three possible outcomes this paper explores are: *purchase*, *abandoned shopping cart*, and *browsing only*. However, instead of only focusing on attribute relationships and finding the probability of a customer to stop browsing in general, this paper trains RNN models to actively detect outcomes early dynamically in response to the customer's action throughout their shopping process. Experimenting with mixtures of high-order Markov Chain Models (MCMs) and mixtures of RNNs that use the Long Short-Term Memory (LTSM) architecture, each model is compared to report on F-measures, recall and precision. Clickstream data recorded across two weeks consisting of 1, 560, 830 sessions is considered. Through analysing this clickstream data, an additional constraint is added to this model where the classifier must work for incomplete sequences without using future data (Toth et al., 2017).

Exploring further on Toth et al.'s (2017) study on predicting shopping behaviour, the machine learning concepts in this study are to be defined. The first model contained a mixture of High-Order Markov Chains. Markov Chains are essentially matrices to predict an event happening based on the previous event. This

model fits the problem's constraints because the predictions can be done without using "future" clicks. The second model used a mixture of RNN models. As opposed to neural networks, RNN models are the only networks that allow memory storage and retainment. With neural networks, items are given weight and bias and the outcome is decided after the data gets passed through a hidden layer, but in RNN, data is retained and past weight and bias can also be modified using backpropagation. RNN has some flaws and that is why this paper implements LSTM to fix the flaws of the original RNN structure. Markov chain models are outperformed by LSTM RNNs, as LSTM RNNs generalise the classifications better with less data. Using the results from this study, websites can choose how to display their data according to their trajectory. For example, a website can show a discount offer when the trajectory of a user aligns with *non purchase* or retrieve more options and mitigate decision fatigue from the customer's side (Toth et al., 2017).

2.4.4 Shopping behaviour analysis through data mining techniques

A study by Mittal explores the concept of statistical modules based on data mining techniques in online shopping ecosystems. As opposed to a web scraping approach proposed by Mehak et al. which scrapes data directly from websites (Mehak et al., 2019), R. Mittal's paper titled *Using Automated Predictive Analytics in an Online Shopping Ecosystem* is based on a UCI dataset analysis which includes Google analytics data. Data mining is performed on this dataset to describe shopping intent of the customers and the likelihood to stop browsing mid-way. Five attributes have been selected out of 18 that were generated inside the UCI dataset for this model. All attributes are numeric and given weight. A dependent variable to represent a website's average value before a transaction is completed is declared, named

pagevalue. Seven studies were tested through structural equation modelling (SEM) using AMOS Version 18 on a UCI machine learning repository. To preface, SEM allows relationships between independent variables and dependent variables to be examined using a collection of data analysis techniques. Relationships of the five attributes were discovered and as a result can be used to improve online shopping systems for better customer conversion rates (Mittal, 2021).

2.4.5 Customer satisfaction analysis using data mining classification algorithms

Moon et al. (2021) with their study titled *An advanced intelligence system in customer online shopping behavior and satisfaction analysis* explores the concept of using DM and ML to predict customer satisfaction on a product. Their proposed system is aimed to forecast product quality and cost analysis through customer behaviour analysis. The end result would be a visualisation and precision of the result. Similar to Toth et al.'s (2017) approach on predicting classified behaviour trajectory (Toth et al. 2017), Moon et al. uses methodologies from DM and ML approaches for this system. From the experimentation that they did, two methods stood out the most performance-wise. Their approach on this using the Apriori algorithm scored an 88% on precision and their approach using Naive Bayes algorithm scored an 87%. Apriori algorithm is one of the most used algorithms in frequent pattern mining under the DM subject. This technique is used to find relationships between attributes through association rules. Naive Bayes algorithm on the other hand is a classification algorithm used to solve tasks revolving around classification and can be used in the same conjunction as an Apriori algorithm would. The results of this are detailed through visualisations using data graphs using a lot of different types of data. The result of this experiment can be used to measure the product quality and show the

current situation of intelligent shopping and customer satisfaction where enhancements on certain products can be made based on positive scores coming from the results of this experiment (Moon et al., 2021).

Table 2.2: Method effectiveness comparison.

Author & Year	**Method/Algorithm Used**	**Efficiency/ Effectiveness**	**Area of Study**
Mehak et al. (2017)	Web Scraping (Used by Google on Genius Lyrics)	93% Search Accuracy	Data Scraping
Fu et al. (2020)	Grip Force and Eye Tracking Sensor	N/A, Only Conclusion Was Drawn	Internet of Things
Toth et al. (2021)	Markov Chain Models (MCM) (Used by Weather Statistics), Recurrent Neural Networks (RNN)	MCM: 0.42 Precision RNN: 0.82 Precision	Machine Learning
Mittal (2021)	Structural Equation Modelling	N/A, Only Conclusion Was Drawn	Data Mining
Moon et al. (2021)	Apriori Algorithm, Naive Bayes Algorithm (Used by E-Commerce websites)	Apriori: 87% Precision Naive Bayes: 87% Precision	Data Mining

2.5 Machine Learning Algorithms In Recommendation Systems

In this section, we will be exploring various methods for recommendation systems present in bigger and more prominent contributors to the data science world. We will be diving into methods used by companies namely Amazon and Netflix. We will also be exploring some famous recommending methods which will be KNN with Cosine Similarity, User-based Singular Value Decomposition and Item-based Singular Value Decomposition.

2.5.1 Recommendation systems in existing infrastructure

2.5.1.1 Amazon

At Amazon, the model used inside their website for recommending items to users is an Item-Item Collaborative Filtering method, and it has stood the test of time. Having been implemented around 2003, it has continued to stay reliable and scale well throughout time until today, which is almost two decades of standing (Smith & Linden, 2017) . Before the introduction of Amazon's Item-Item collaborative filtering, most other collaborative filtering based recommendation systems were user based. This meant that it had to be scoured through the internet and grind between all the users' preferences which does not scale well with millions of users everyday. The advantage of an item-item model is that most of the computation can be done online without having to continuously update itself based on users.

The specific algorithm that is used for Amazon's website is not disclosed but it does disclose the variables that are being taken in consideration. Firstly, they take in consideration how items are related by defining through the customers who bought

the item and finding a common ground. Here they implement Fubini's Theorem to find the correlation between items through users and an emphasis on perceived quality is placed on the algorithm. Some factors that had to be neglected due to limitations such as scalability and performance include item compatibility and false positives that are introduced with it. Amazon has also discovered that purchasing behaviours can be emergent from relations between the item and the user. This refers to the correlation between user and items can be different when an item is cheap compared to when it is expensive. When an item is cheap, all users tend to buy any item with less correlation. But when an item is expensive, certain behaviours can emerge based on it which can influence items that are bought after it.

2.5.1.2 Netflix

According to researchers from Netflix, their algorithm consists of more than one algorithm serving different purposes to achieve a single goal which is to make Netflix a streamlined experience (Gomez-Uribe, & Hunt, 2015). Netflix offers a wide choice of options, too many that decision fatigue can be present. Humans are surprisingly bad at making decisions between many options and tend to get overwhelmed easily by choosing "None of the above" and making poor decisions (Schwartz, 2015). Consumer research suggests that customers tend to lose interest after 60 to 90 seconds of browsing, which includes reviewing 10-20 titles that are given to them. Within that time frame, the user either finds something of interest or abandons the site entirely which can be detrimental to the business.

The Netflix recommendation problem was initially solved by predicting the rating a user would give towards a movie. But technology has advanced and a lot more data can describe behaviour patterns. These can be what each member watches,

how each member watches (device,time within a day, area), the page and specific area within a screen the product was discovered, and also recommendations that were shown but were not touched upon could bring in extensive valuable data. Now the Netflix experience consists of many algorithms working together, most of which are present in the main page, which contribute to roughly 65% of the discovered shows and the hours watched.

2.5.2 Widely-used recommendation system algorithms

2.5.2.1 K-Nearest Neighbour

First introduced by Fix and Hodges in 1951, in an unpublished report from the US Air Force School of Aviation Medicine report, the method gained popularity a decade later around the 1960s where computing power increased. It became widely used for pattern recognition and classification and is an unsupervised algorithm. It learns by comparing various sets of tuples and classifies them by closest neighbour. It has stood the test of time and is still being used 60 years later. K-Nearest Neighbour either uses Euclidean distances or Cosine Similarity where it calculates through tuples (Adeniyi et al., 2016). Given a graph, this would mean that the tuples can be a set of coordinates of two different points, where the more items in the tuples, the more dimensions that it has.

2.5.2.2 K-Means Cluster

Clustering refers to the unsupervised classification approach for recognizing patterns, in recommendation systems this is essentially feeding unlabeled data and the neural networks trying to find a common ground between the items to establish relation. K-Means clustering refers to having all the data be plotted on a 2 or more

dimensional graph, and then setting a value of a mean for them to cluster by. They provide random seedings for initial clusters to form. However the main drawback of this is that having random clusters prior to seeding is detrimental to the calculation with numerous drawbacks that can be taken from it thus making it inappropriate for recommendation system use (Zahra et al., 2015), and are more suitable used in grouping genres together (Ahuja, Solanki, & Nayyar, 2019). Because of the drawbacks, to make it feasible in recommendation systems, many studies have been done to make hybrid versions of this algorithm to cover for the drawbacks (Zahra et al., 2015; Chen et al., 2005; Duwairi & Abu-Rahmeh; 2015).

2.5.2.3 Matrix Factorization Model

Matrix factorization model refers to recommendations that are given by calculating the latent features of each model. It uses collaborative filtering and recommends products based on the assumption that a user will always agree on their past tastes (Batmaz et al., 2019).

The actual model involves constructing the latent features of the model by initialising three matrices. Then it is expected to obtain the objective function, this usually involves using gradient descent method to achieve the optimal solution. Gradient descent method refers to a method where an input is run multiple times with calculated epoch adjustment through a neural network or through a deep network before hitting the point of equilibrium. Values of features are then updated to the matrices and adjusted based on the feature modelling. Unknown ratings will then be predicted and then evaluated using root mean squared error or any similar evaluation method.

2.5.2.4 Singular Value Decomposition

Because of data sparsity accompanied with high dimensions of matrices and problems related to it are present in recommendation systems, scalability becomes an issue. Due to this, SVD becomes a powerful technique that allows dimensionality reduction where multiple matrices can be decomposed into fewer dimensions.

2.6 Proposed Solution

Based on previous papers and studying the different problems and potential improvements possible in this area, a conclusion was made for the solution to be a data mining based intelligent shopping system, with support from machine learning. A recommendation system will be created through the use of Singular Value Decomposition with Item-Item correlation for items to be recommended through the online shopping system.

Using Mehak et al.'s method for web scraping, the current system will scrape the target online shopping web service, Amazon, from methods described inside said paper to populate the database with product information.

The features that a user can expect from the online shopping system would be normal shopping features such as buying products, filtering products, checking promotions, searching for products, recommended products, changing shipping time and such. The main difference is that everything that the user receives on the page is curated to the user based on the user's preference using intelligent models applying techniques from data mining and machine learning.

CHAPTER 3

RESEARCH AND DESIGN METHODOLOGY

This section explains about the methodology that this system will take. Using Agile Development Methodology, this section will explore the methodology used through five steps: *Inception*, *Elaboration*, *Construction*, *Development*, and *Evaluation*.

3.1 Introduction

The research methodology is based on two main systems, case study approaches and literature review approaches. Case study is used to give the big picture and find out the current problems that reside within current online shopping ecosystems. With the title being handed, the focus is shifted straight into localised online shopping systems namely Amazon for baseline, and is streamlined within those areas. Once ideas have been outlined, literature review is performed within the relevant areas. Literature reviews are done and they give deeper insights into the problems within the studied area, which is data mining for web services. They help outline the methods past projects have undergone and allow possibilities on adapting said past projects and merging multiple systems into one streamlined system to be used in this project. Literature review in areas such as past data mining methods and the current online shopping insights were done. On top of that, five different products were compared:

- Web scraping online shopping system
- IoT shopping behaviour analysis system
- Machine learning shopping behaviour prediction system
- Shopping behaviour analysis through data mining techniques

- Customer satisfaction analysis using data mining classification algorithms

The development methodology will be based on Agile Development Methodology, which refers to iterative implementations and gradual improvements over time. Based on the book *Object Oriented Systems Analysis and Design Using UML* by Simon Bennett, Steve McRobb, and Ray Farmer, this project references the systems and uses the book as guidance for the development lifecycle and designing the items present inside the project. The data set that we will be using for this machine learning is from a public machine learning repository, UCI (UCI, 2022) and Stanford's Amazon data repository (Stanford, 2018).

3.2 Inception

Exploring within the constraints of online shopping, this stage is all about finding potential optimizations for online shopping on web services. This project has been chosen to use data mining (DM) and machine learning (ML) aspects as they have shown to fit within the constraints of this project.

Multiple areas of an intelligent shopping structure are planned for this project. Recommendation systems through the use of Singular Value Decomposition will be used for product recommendations. Notification systems based on past searches will be implemented, and web scraping plugins, namely Beautiful Soup 3 will be utilised to scrap the data from the target website, which is Amazon.

3.3 Elaboration

Requirements for the Online Shopping System (OSS) is listed as below:

Table 3.1: Table for User Requirements.

No.	Requirement Description	Type (Functional/ Non-Functional /Usability	Stakeholder
1	**Users query products:** Users can query products and the system will find appropriate products and give it to the user based on relevance.	Functional	End User, OSS
2	**Users can rate the product:** Users can give a rating to the product after purchasing it.	Functional	Customer, Seller
3	**Users can purchase the product:** Users can purchase the product through the website, the checkout system will be through lotus.	Functional	Customer, OSS, Seller
4	**Users can check history of checked products:** Users can check last visited products by user.	Functional	End User, OSS
5	**Users can check purchase history:** Users can check and find details about previously bought products by the user.	Functional	Customer, OSS
6	**Users can sort product by criterias:** Users can sort products by name, popularity, price, relevance.	Functional	End User, OSS
7	**Users can browse products by category:** Users can check a product page by shop, category such as appliances, electronics, etc.	Functional	End User, OSS
8	**Users can see most popular products:** Users can have a section for most	Functional	End User, OSS

	popular products		
9	**Users can login:** Users can use their google accounts or Lotus accounts to login through google.	Functional	End User, OSS
10	**System can send emails about promotions:** System can send email to users through python's SMTP.	Functional	OSS, End User
11	**System can send promotions:** Based on customer behaviour, the system can send promotions if a customer seems to be leaving the site.	Functional	OSS, End User
12	**System can recommend items:** System uses results from the Apriori algorithm to associate results and use them for recommendations.	Functional	OSS, End User
13	**Bank system verifies payment:** System can go through the bank to verify the payment to get the receipt.	Functional	Bank, OSS, Customer
14	**Users can check an "Other's have also checked this" section:** System puts out a section for a user to check which products are most relevant to them.	Functional	End User, OSS
15	**Users can manage Payment Methods:** Users can choose payment methods such as buying through cards or through e-wallets.	Functional	End User, OSS
16	**Users can opt out from promotional emails:** Users can opt to not receive emails from the system.	Non-Functional	End User
17	**System can store user behaviour patterns:**	Non-Functional	OSS, End User

	System can store data such as dwelling time, websites visited and the like inside the database for algorithm use.		
18	**System is secure and can handle mild security threats:** Due to the nature of online shopping, websites would want to prevent getting hacked and accidentally send out products for free or get paid less than they should. So the system would opt for greater security to combat this problem.	Non-Functional	End User
19	**System can handle load from many users**	Usability	OSS
20	**Algorithms are efficient**	Usability	OSS
21	**Algorithms are high accuracy and reliable**	Usability	OSS

The system is described as a flowchart, this is the main flow of the website without getting into details of the deep algorithms that will be discussed in other use cases.

Figure 3.1: Flowchart for Main Page.

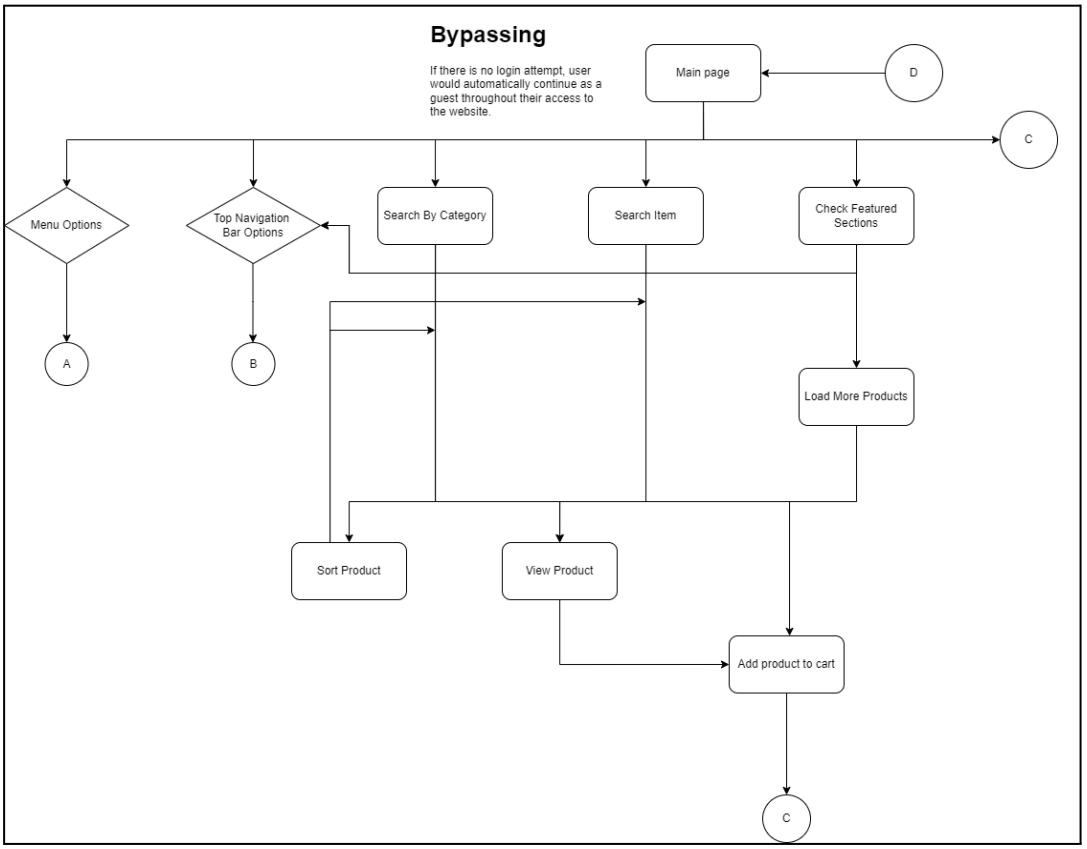


Figure 3.2: Flowchart for Menu.

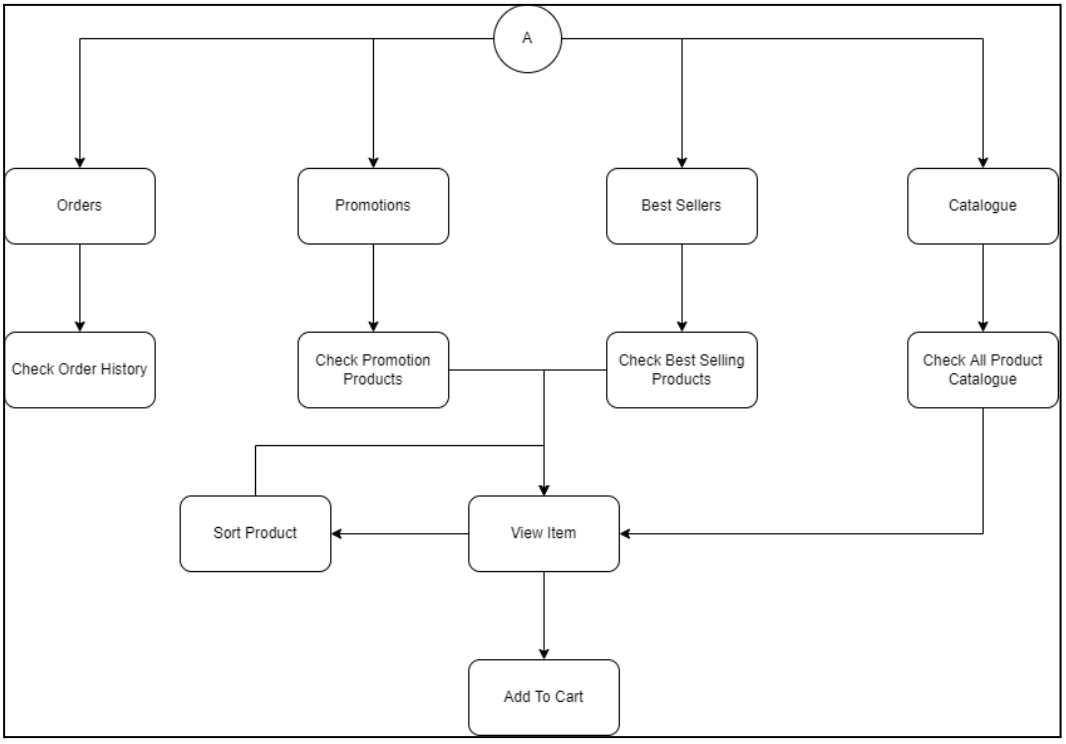


Figure 3.3: Flowchart for Top Navigation

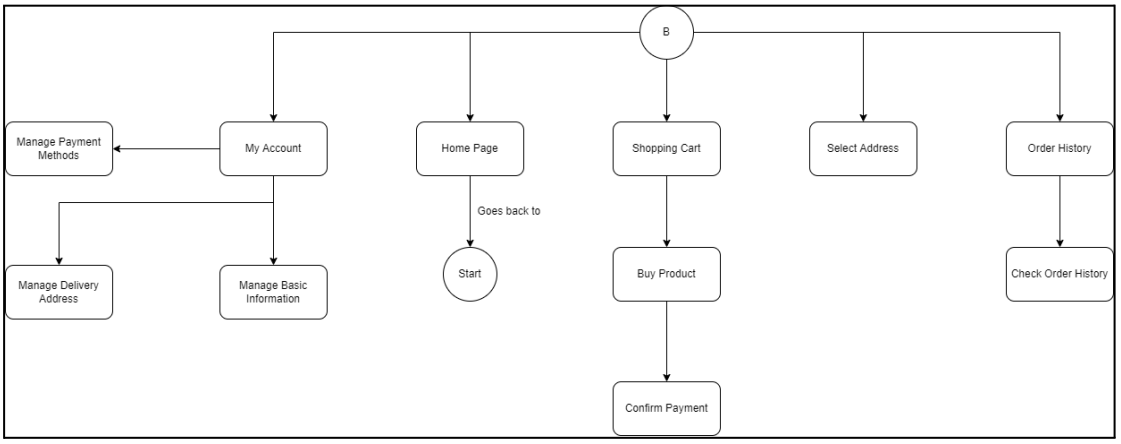


Figure 3.4: Flowchart for Login Verification

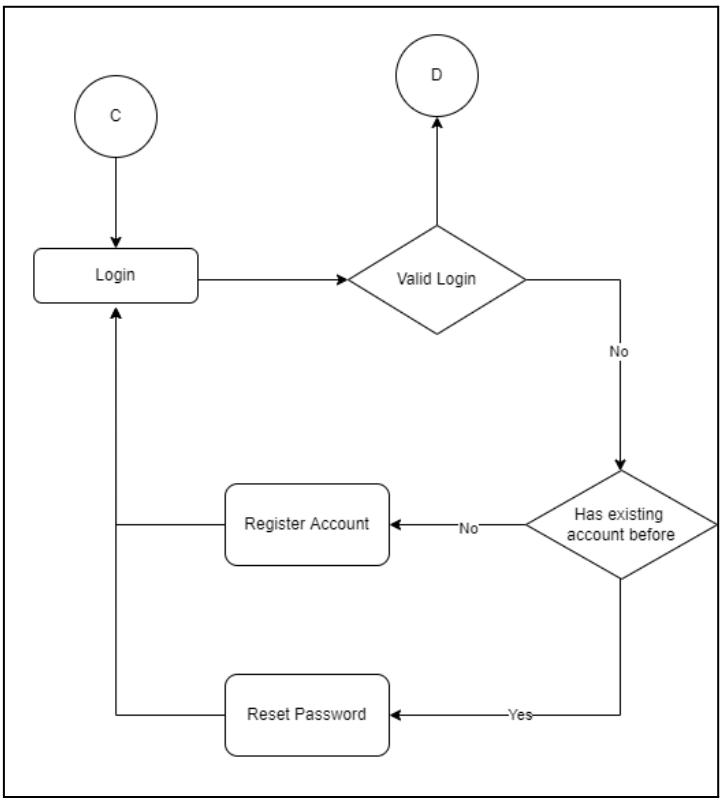


Figure 3.5: Use Case Figure for User Side

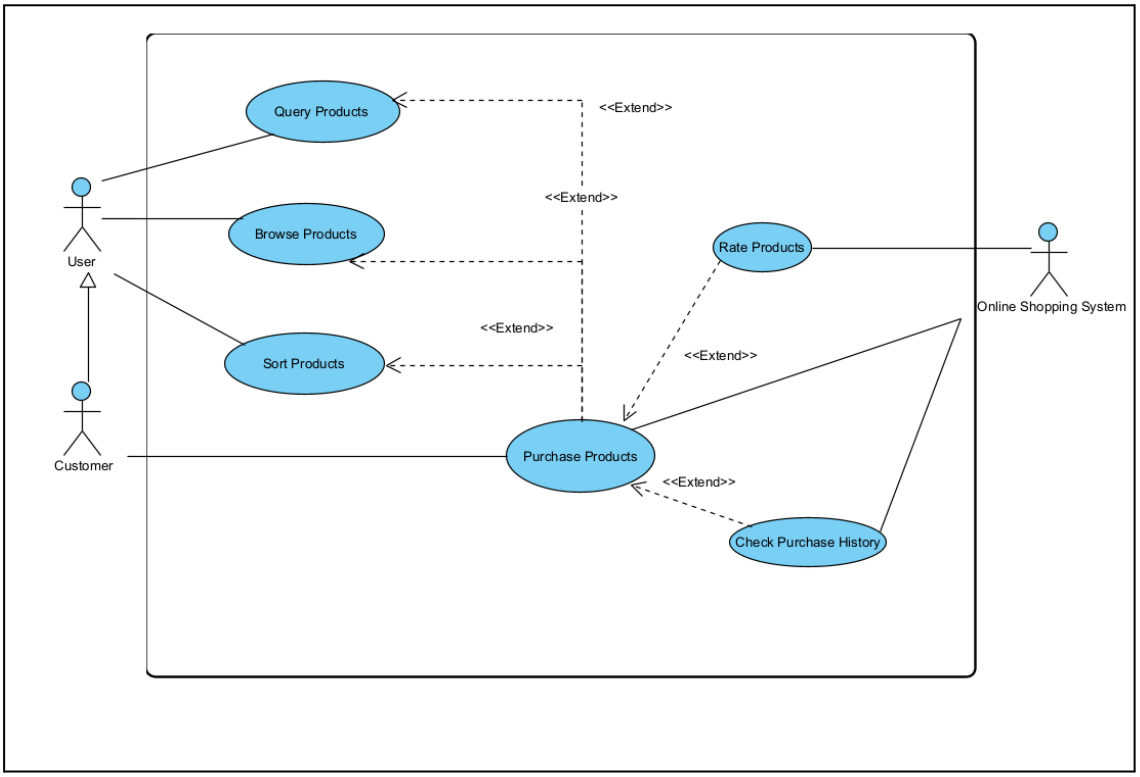


Figure 3.6: Use Case Figure for Online Shopping System Side

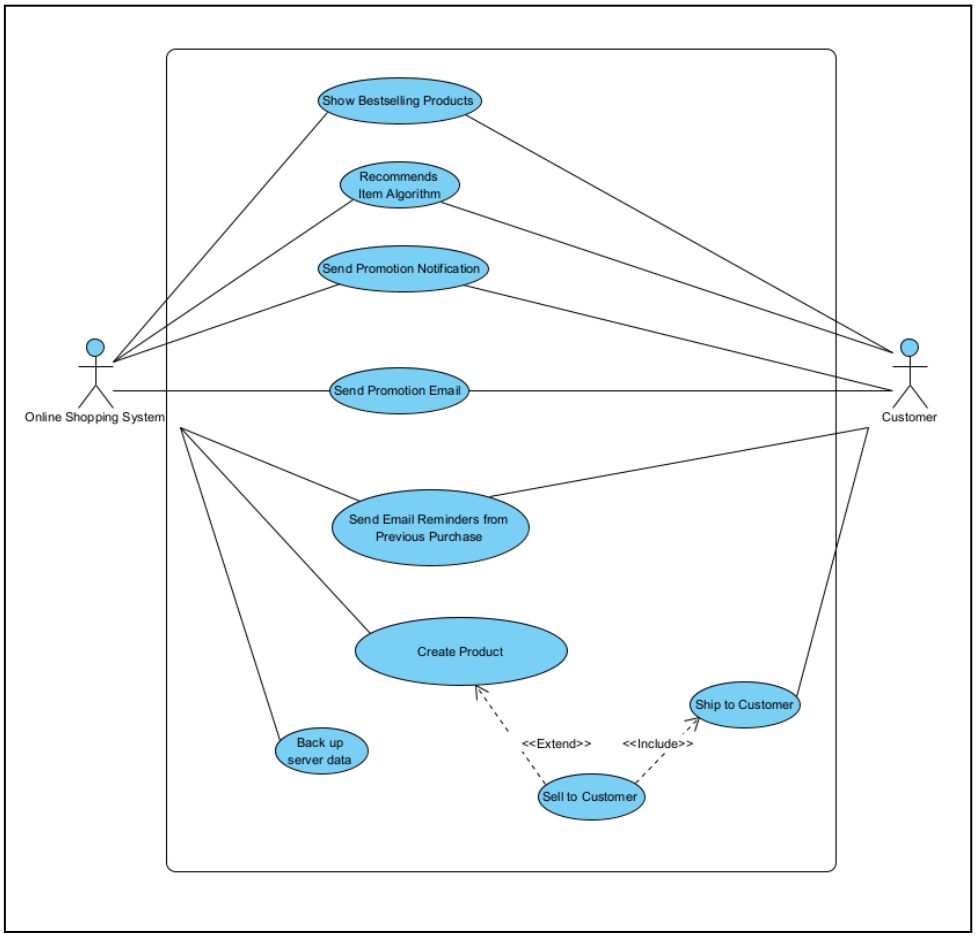
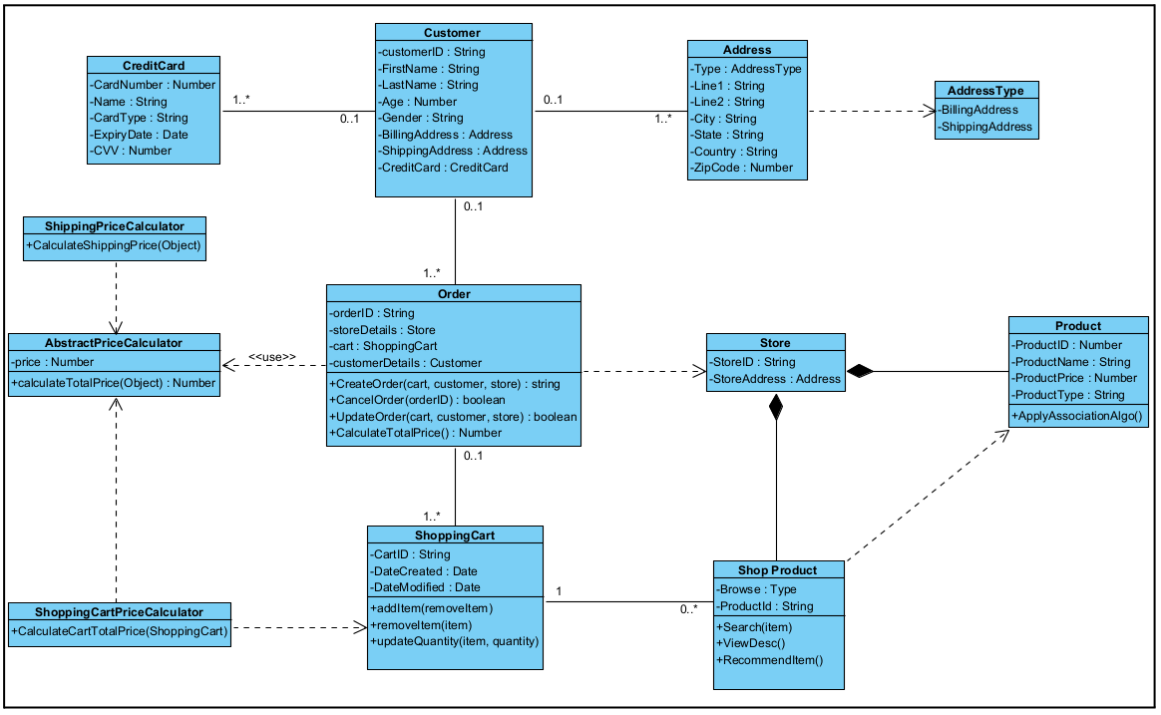


Figure 3.7: Class Figure for Online Shopping System



As shown above, these are the use cases that are crucial for this project, these help to serve as a guideline for the project to base it's requirements of.

3.4 Construction

In the construction phase, the system will be divided into several parts to streamline workflow on them. The main parts will consist of: the recommendation system, web scraping from Amazon, UI display, frontend and backend, and email notifications about products.

For the recommendation system, three types of calculations were tested to see which one had the best results and was easy to incorporate into a web server. The three different methods were, Cosine Similarity, Traditional Singular Value Decomposition, and Truncated Singular Value Decomposition.

Cosine Similarity is a user to item memory based algorithm where you track correlation of each user and their rating predictions by calculating the cosine distance of it. This is a common collaborative filtering method. This method is fairly computationally light as you can scan millions of data in a reasonable amount of time. But because this is memory based, the output footprint is very huge. Using 100,000 rows of data, the output of the calculation neared 8 Gigabytes which is expensive for the storage.

Figure 3.8: Cosine Similarity Diagram

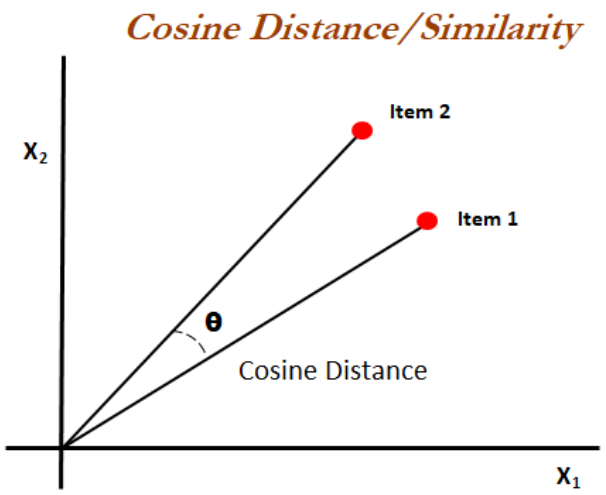


Figure 3.9: Cosine Similarity Code

```
from sklearn.metrics.pairwise import pairwise_distances
user_similarity = pairwise_distances(train_sales,
metric='cosine')
item_similarity = pairwise_distances(train_sales.T,
metric='cosine')
def predict(ratings, similarity, type='user'):
    if type == 'user':
    mean_user_rating = ratings.mean(axis=1)

    #you use np.newaxis so that mean_user_rating has
same format as ratings
    ratings_diff = (ratings - mean_user_rating[:,
np.newaxis])
```

```
    pred = mean_user_rating[:, np.newaxis] +
similarity.dot(ratings_diff) /
np.array([np.abs(similarity).sum(axis=1)]).T
    elif type == 'item':
    pred = ratings.dot(similarity) /
np.array([np.abs(similarity).sum(axis=1)])
    return pred
```

Another method uses the Traditional Singular Value Decomposition Method where it is a user to item model based algorithm where three matrices are decomposed.

$A = USV^T$

Matrix U: singular matrix of (user*latent factors)

Matrix S: diagonal matrix (shows the strength of each latent factor)

Matrix U: singular matrix of (item*latent factors)

Figure 3.10: Traditional Singular Value Decomposition Code

```
import scipy.sparse as sp
from scipy.sparse.linalg import svds


#get SVD components from train matrix. Choose k.
u, s, vt = svds(train_sales.to_numpy(), k=20)
s_diag_matrix = np.diag(s)
X_pred = np.dot(np.dot(u, s_diag_matrix), vt)
pprint(X_pred.shape)
X_df = pd.DataFrame(X_pred, index=train_sales.index,
columns=train_sales.columns)
pprint(X_df)
print('User-based CF MSE: ' + str(rmse(X_pred,
test_sales.to_numpy())))
```

Because of how computationally expensive the traditional SVD, truncated SVD will be used. Truncated SVD refers to having only a specified number of rows of output instead of n-rows, where n is the original sample size. Only items that have more than 500 sales and reviews would be included in this calculation.

Figure 3.11: Truncated Singular Value Decomposition Code

```
X = sales.T

X_index = X.index

# print(X.shape)


from sklearn.decomposition import TruncatedSVD


SVD = TruncatedSVD(n_components=20)

decomposed_matrix = SVD.fit_transform(X)

# print(decomposed_matrix.shape)


correlation_matrix = np.corrcoef(decomposed_matrix)

df = pd.DataFrame(correlation_matrix, columns = X.index)

df.head()
```

The information and products were scraped through Amazon, mainly because the API was paid and there were budget constraints. Because of Amazon's thin layer of security over web scraping and botnets, it was possible to bypass it by passing in headers. Product names were asin identification tags that were being aimed to be scrapped. Manual sleeps were implemented to not overload the website requests and trigger anti-botnet responses.

Figure 3.12: Amazon Web Scraping Code

```
headers = {
     'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:106.0) Gecko/20100101 Firefox/106.0',
     'Accept-Language': 'en-US,en;q=0.9'
}

with open('product_names.txt', 'r') as f:
     product_names = f.readlines()

product_asin = [prod_name.strip('\n') for prod_name in
product_names]

products = []

count = 0
total = len(product_asin)

for asin in product_asin:

     count += 1

     main_URL = f"https://www.amazon.com/dp/{asin}"
     r = requests.get(main_URL, headers=headers)
```

```
     soup = BeautifulSoup(r.content, "lxml")
     result = soup.find('div', attrs={'class':
's-main-slot s-result-list s-search-results sg-row'})

     try:
     img_link =
soup.select_one('img.a-dynamic-image[src]')['src']
     except:
     img_link = 'Does not exist'

     try:
     title =
soup.select_one('span.product-title-word-break#productTi
tle').text.strip()
     except:
     title = 'Does not exist'

     try:
     productPrice =
soup.select_one('input#attach-base-product-price')['valu
e']
     except:
     productPrice = "Out of Stock"
```

```
        product_dict = {
        'asin': asin,
        'url': img_link,
        'title': title,
        'price': productPrice
        }

        products.append(product_dict)

        print(f'Successfully retrieved product details for
{asin} ({count} out of {total})')
        sleep(1)



print ('Operation Completed, Retrieved all needed data
from Amazon')
with open('product_details.json', 'w') as f:
        json.dump(products, f)
```

The result of the scraped Amazon details were inputted in a PostgreSQL provisioned database that is inside Railway.app. It is uploaded through an API within the website. Moving on, the frontend and backend will be written in NextJS, a framework built on top of ReactJS. There are many advantages of using NextJS, the main one being SEO and fast page loading because of how the data flow is set. With the release of NextJS 13, pages can be cached correctly according to the use of each

web application or web service. Because of the complex architecture, dozens of files were needed. In this report, we will only be covering the recommendation system API because that is most relevant.

The API consists of a few sections. The first code block calls the database through Prisma's Object Relational Mapping technology for typesafe queries and calls all the previous orders and products and selects the item ids.

Figure 3.14: Prisma Invocation for Recommendation API

```
const invoiceRes = await prisma.invoice.findMany({
     where: {
          customer_id: '123456',
     },
     select: {
          OrderHistory: {
               select: { item_id: true }}}
});
```

The second code block essentially maps the array of invoices which then maps from the order history table to return the item ids. Because the maps were nested, it was then flattened to return a single array of strings that could easily be accessed. Then it is spliced so that it only returns the 10 most recent products.

Figure 3.15: Accessing Nested Items and Splicing Mapping Code

```
const prevOrders = invoiceRes
     .map((inv)   =>   inv.OrderHistory.map((order)   =>
order.item_id))
     .flat()
     .splice(0, 10)
```

The Supabase API is then called in the third code block which returns an array of correlations of products that were in the 10 most previous orders.

Figure 3.16: Invocation of Supabase API

```
const { data } = await supabase
     .from('Recsys')
     .select('*')
     .in('asin', prevOrders);
```

The response from Supabase is then mapped and filtered to only return correlations that are above 0.65 and are not the correlation of 1, which is the product itself. It then returns a tuple which are pairs of ids and values.

Figure 3.17: Product Threshold Filtering Code

```
const prodFilter: [string, number][] = data!
     .map((recs: number) =>
          Object.entries(recs).filter(([asin,    value]:
[string, number]) => {
               if (asin != 'asin' && value > 0.65 &&
value != 1) {
                    return {
                         [asin]: value,
                    };
               }
          })
     )
     .flat();
```

The filtered products which returned all correlations over 0.65 were then sorted in descending order in the following code block which only returned the ids of the items and left the values behind.

Figure 3.18: Sort By Correlation Code

```
const recSorted = prodFilter
     .sort((a, b) => {
          const x = Object.values(a)[0];
          const y = Object.values(b)[0];
```

```
          return x && y && typeof y === 'number' &&
typeof x === 'number'
               ? y - x
               : 0;
     })
     //                          eslint-disable-next-line
@typescript-eslint/no-non-null-assertion
     .map((rec) => Object.keys(rec)[0]!);
```

Then the Prisma ORM is invoked again to get all the product information inside all highly correlated items.

Figure 3.19: Invocation of Prisma for Product Information Code

```
 const recProds = await prisma.item.findMany({
     where: {
          id: { in: recSorted },
     },
     select: {
          id: true,
          name: true,
          price: true,
          image_url: true,
     },});
```

3.5 Deployment

Deployment will be on multiple platforms. The main frontend and backend will be deployed on Vercel, where it can support all the innovative NextJS features such as incremental static regeneration. The main database where it will hold database records will sit in Railway.app, where the main drawback is that the free version can only run for 500 hours a month before stopping, which is equivalent to around 20 days of running and 10 days of not running. But this should be good enough for the time period that is given. The main recommendation system data will be deployed on Supabase because of ease of use. This is because the REST API in Supabase is automatically set up and there is not a need to sync Prisma with it, where syncing Prisma means that it is needed to type check every single column and their type check. It is possible to introspect it through Prisma and auto generate types, but that will inherently delete the already uploaded items inside the database.

3.6 Evaluation

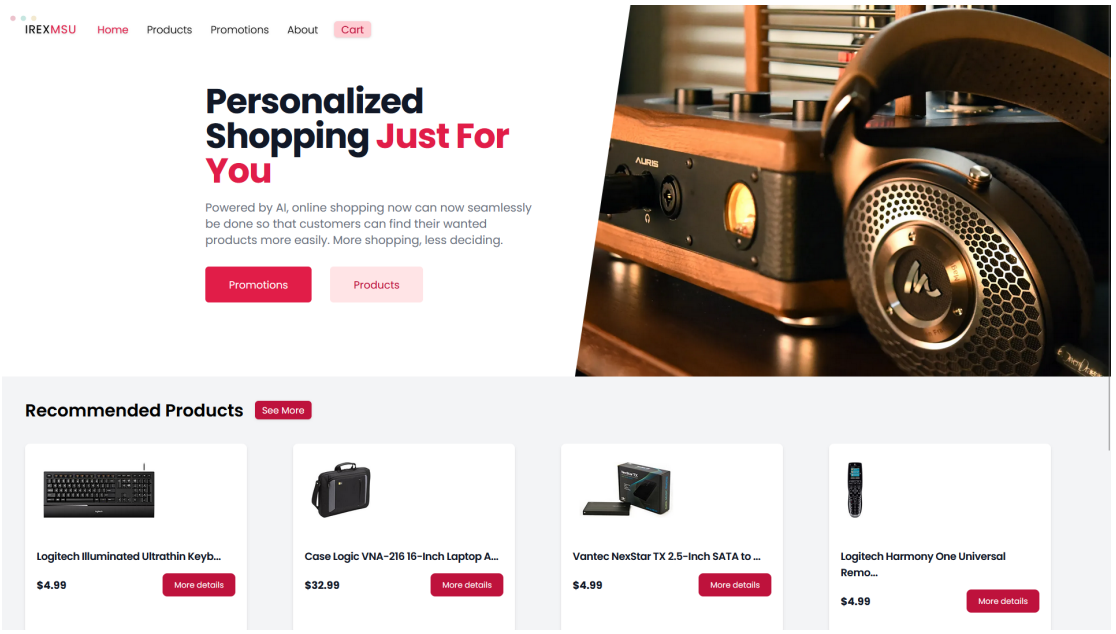
In an agile development setting, the evaluation step is the most important one as it decides which area will be focused on next and will ultimately decide which direction of the project will head. For this project, the evaluation phase would feature evaluating the accuracy of predicting the user behaviour and evaluating outcome of the actions. The next set of objectives will be set and will be moved into the sprint backlogs as a result of this evaluation step.

CHAPTER 4

DISCUSSION

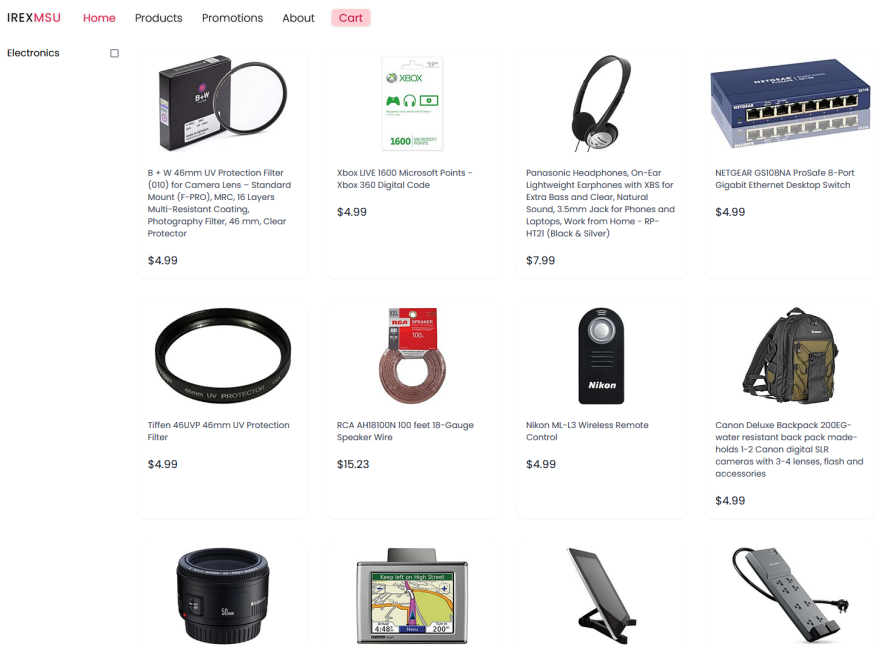
4.1 Website Result

Figure 4.1 Main Page



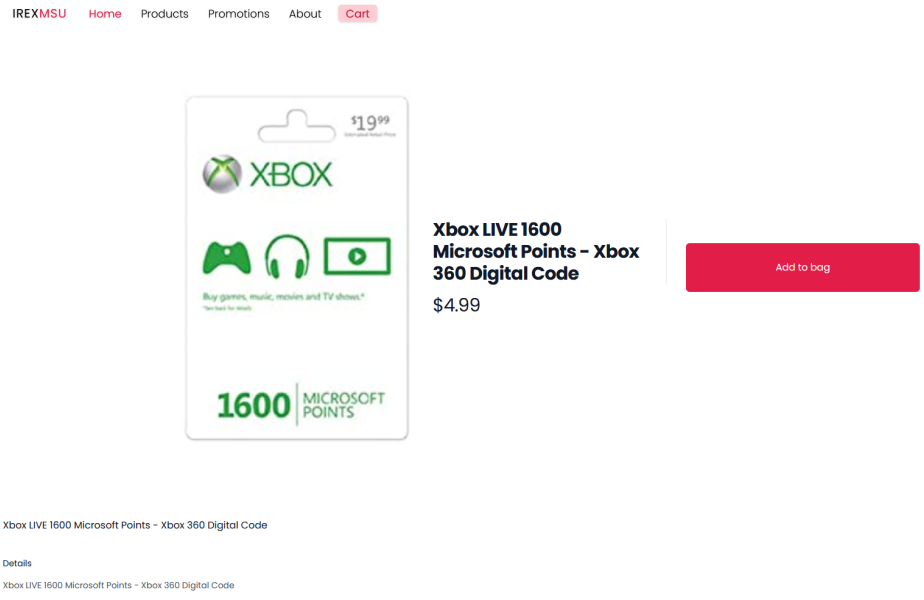
The main page consists of the hero page, the navigation bar, which is present in every page, and the recommendation page snippet. This is the initial page that a user will be able to visit once they enter the website.

Figure 4.2: Products Page



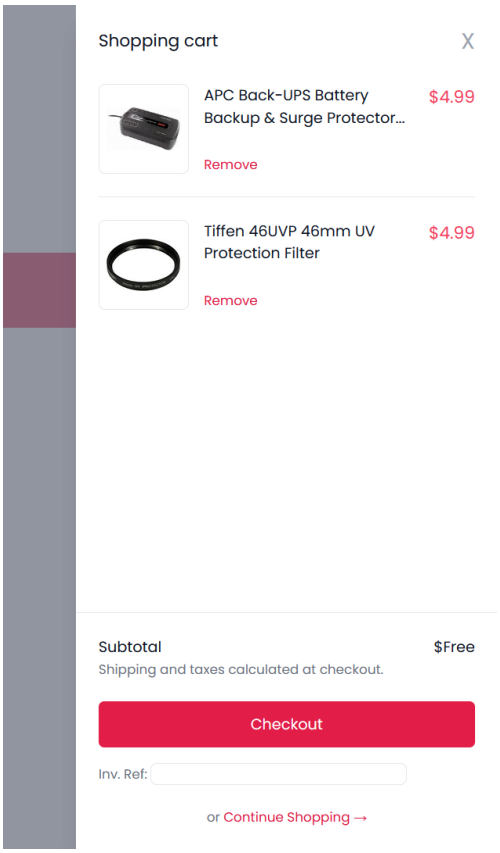
This is the products page where users will be able to browse through products and be able to click on each product to go into their respective pages.

Figure 4.3: Product Information Page



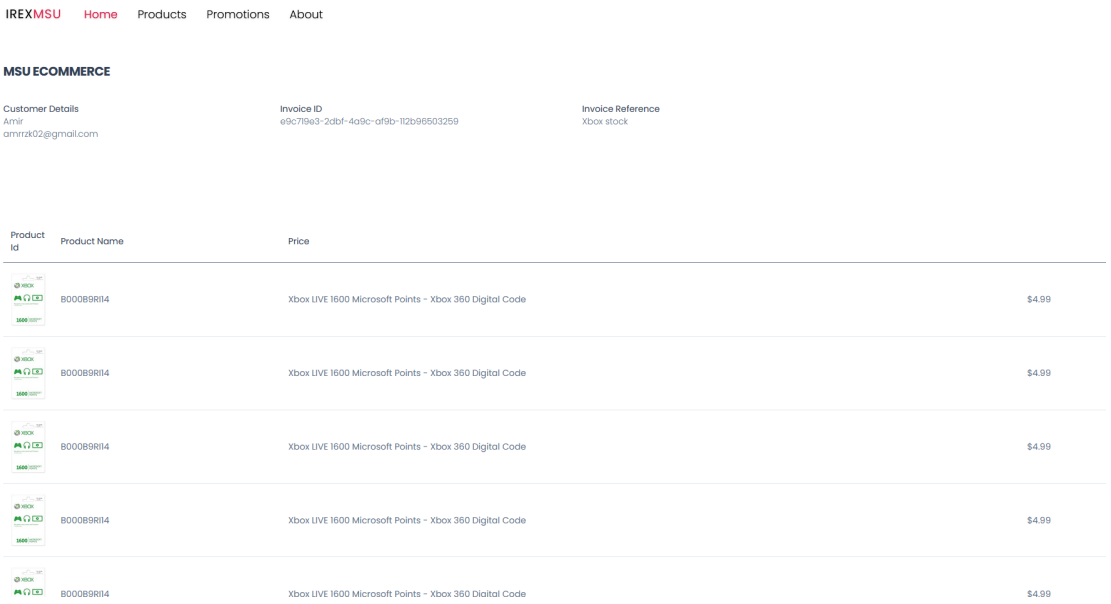
The product information page contains information about the product and allows the user to enter the products into the cart.

Figure 4.4: Shopping Cart Popup



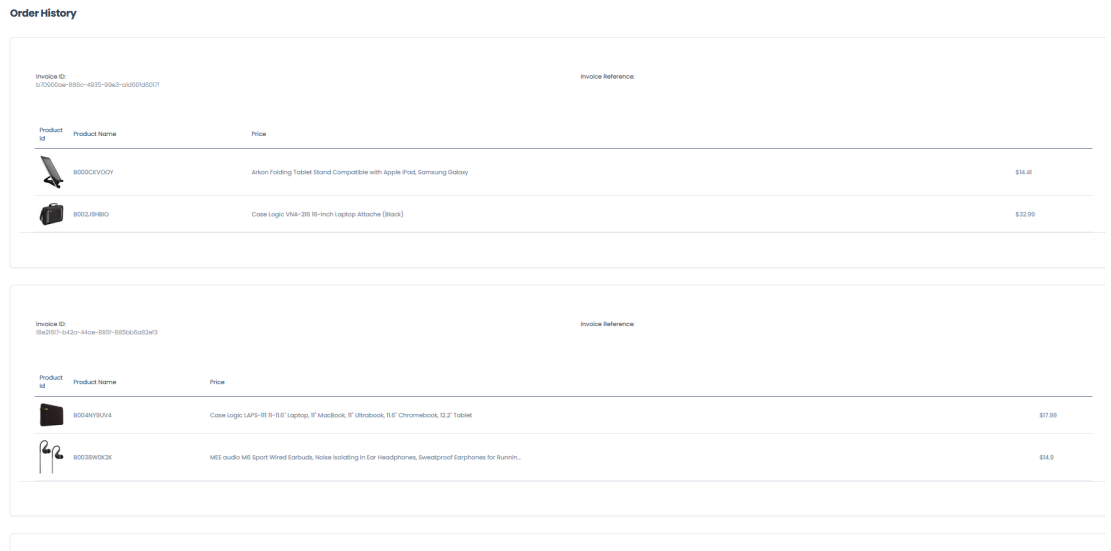
After clicking on a product and entering it into the cart, the user can then click on the cart button on the navigation bar. In turn, this popup will show up and users can checkout their products and proceed to the invoice page. Users are also allowed to remove their products from the cart.

Figure 4.5: Invoice Details Page



After checking out from the cart, users are then headed to this page. Because this is only a mock website to display the application of recommendation systems in online shopping, no payment is done and users are headed straight to the receipts page.

Figure 4.6: Invoice History Page

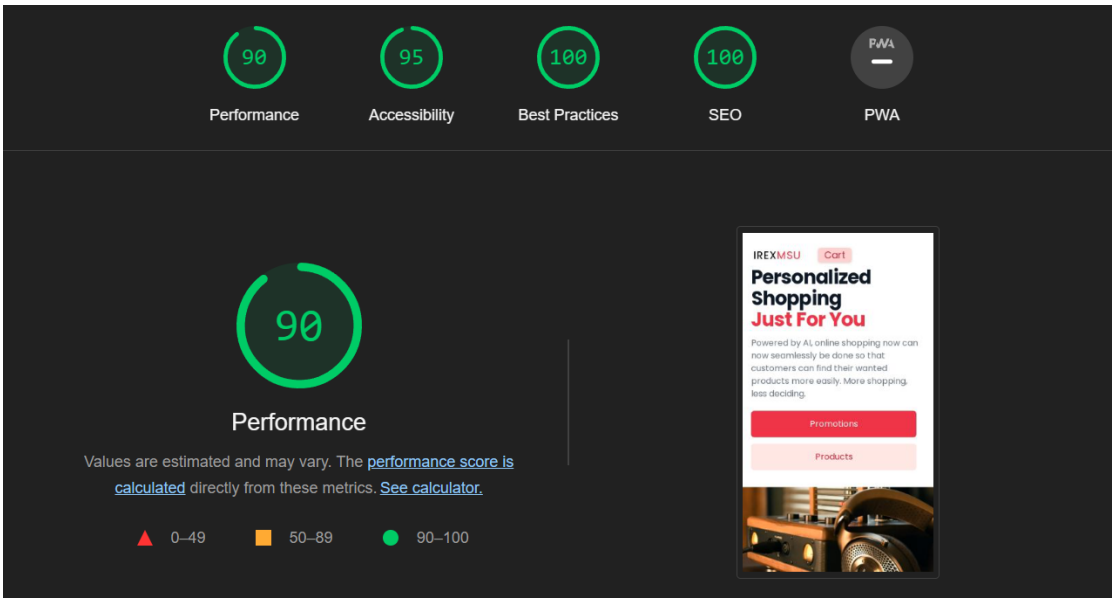


Users can also check their previous purchases by going to the invoice history page. This will list all previous purchases and their respective details.

4.2 Website analytics

Because of the optimised architecture, the pages load very fast and the ones that are not loaded fast were because it was not cached beforehand, and after the first load it will be cached on the CDN which would increase the speed again. Due to this, the website analytics are very positive. Below are analytics from Google's open-source analyzer for websites, Lighthouse.

Figure 4.7: Lighthouse Analytics Results

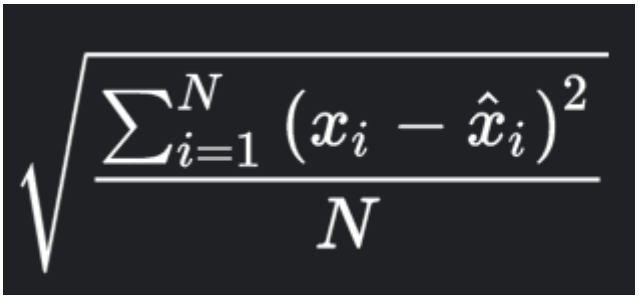


4.3 Recommendation System Evaluation

Evaluation of the recommender system will be using the root mean squared error method. Through calculation, the root mean squared error of 0.347, this means that it only errors by 0.347 per rating. But with a more punishing algorithm, the error rate becomes 1.07. This error is calculated through a lot of parsing and data cleaning before it is eligible to be calculated, but we will only cover the code analysis that is relevant to the actual prediction.

Root Mean Squared Error, also known as RMSE, is popular for calculating losses. It is a variation of the algorithm Mean Squared Error which punishes more deviation, but at the same time is easier to interpret. The mathematical notation is as below:

Figure 4.8: Root Mean Squared Error Formula



The algorithm inside python on the other hand is written as below:

Figure 4.9: Root Mean Squared Error Code

```
def rmse(predictions, ground_truth):
     predictions = np.array(predictions)
     ground_truth = np.array(ground_truth)
     return np.sqrt(np.mean((predictions - ground_truth)
** 2))
```

Because the main recommendation system is calculated through item-item, there are not any straight-forward paths to evaluate it as most evaluations need ground truths that are user-items. Thus, needing the correlation to be translated and parsed into a

user-item matrix. First, all of the ratings and reviews were stored in a dataframe that stored the coordinates of all the non-zero elements. Then the positions are then stored in properly parsed dictionaries for easier access. The same is done to the correlation matrix.

The general algorithm for the item-item correlation to prediction translation is as below:

Figure 4.10: Lenient Prediction Code

```
prediction += corr_matrix[corr_item][item] *
(actual_rating[reviewerID][item] - user_average)
```

This original algorithm is very forgiving and resulted in low RMSE which makes it an illusion that the system gives accurate predictions. This is because the general deviation is cushioned by deducting the user average which will skew the direction of the rating. Because of that, we will be using an algorithm that punishes the inaccuracy more. The following algorithm will put more weightage on products with higher correlations and will skew it more exponentially.

Figure 4.11: Harsh Prediction Code

```
prediction = [predict_rating(pred_corr[i],
[actual_rating[reviewer][corr_item] for corr_item in
corr_items[i]]) for i in range(len(items))]
```

This essentially creates a list of predictions that the algorithm predicts what the user will give to an item. A function is then called which will predict each item

respectively based on the correlations. On each list comprehension loop a function named predict rating is called that counts every non-negative pair.

Figure 4.12: Predict Rating Function for Harsh Prediction Code

```
for i in range(len(correlations)):
     if correlations[i] > 0:
     predicted_rating += correlations[i] * ratings[i]
     total_correlation += correlations[i]
return predicted_rating / total_correlation
```

This function essentially will find all the correlations with positive values and will predict with more weightage on deviation. The RMSE of both functions are as below:

Figure 4.13: Result of both evaluation

```
Recommendation System Accuracy Through RMSE Lenient:
0.3471288284211668


Recommendation System Accuracy Through RMSE Harsh:
1.0799780023949475
```

4.4 Technical Limitations

Recommendation systems are in general, hard to evaluate. To add to that, an added layer of abstraction is present because of the lack of a direct way to evaluate an

Item-Item correlation matrix due to its nature. Because of this, the recommendation system is going to be very hard to improve and the metrics that are set as success metrics are very vague.

Due to lack of expertise, only product rating is taken into account and only electronic products over 500 sales and reviews were used. The dataset gave some information such as review messages, but without the knowledge of sentimental analysis in machine learning, the feature was not taken into account.

4.5 Conclusion

As we discussed in this section, recommendation systems are hard to evaluate and need to be based on assumptions that in the future users will always agree with their tastes in their past. On top of that, taste and preference is a very subtle thing that can change based on external variables and personal interaction. Despite that, it is a staple to good websites as it increases the click rate by each recommendation and has proven to be very useful in many applications.

Here, we first covered the result of the general website architecture. Then we covered the evaluation method, which is root mean squared error. Then we covered the conversion from an Item-Item correlation matrix to a User-Item prediction matrix using two different algorithms. After that, we discussed the limitations that this discussion section faced.

CHAPTER 5

SUMMARY AND CONCLUSION

Using data mining concepts and techniques supplemented by machine learning approaches, this project will develop an intelligent shopping system where users can perform online shopping optimally. An example of a feature that would be present in this intelligent shopping system for web services would be optimization of recommendation systems which would lead to better customer conversion rate. Users can receive promotions on the website and through emails, smart tracking systems can track the users and give them extra offers if they appear to stop browsing.

This intelligent shopping system for web services will optimise services that are lacking on normal online shopping areas and fix the flaws that are present in current systems. It will make use and enhance upon previously developed algorithms in hopes of contributing to smart online shopping systems and data mining as a whole.

REFERENCES

Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. Applied Computing and Informatics, 12(1), 90-108.

Ahuja, R., Solanki, A., & Nayyar, A. (2019, January). Movie recommender system using K-Means clustering and K-Nearest Neighbor. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 263-268). IEEE.

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. Journal of Engineering and Applied Sciences, 12(16), 4102-4107.

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52(1), 1-37.

Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation. (2022). Beautiful Soup 4 Documentation. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Bennett, S., & Farmer, R. (2010). *Object-Oriented Systems Analysis and Design Using UML* (4th Revised ed.). McGraw-Hill Education.

Bohnenberger, T., Jacobs, O., Jameson, A., & Aslan, I. (2005, May).

Decision-theoretic planning meets user requirements: Enhancements and studies of an intelligent shopping guide. In *International Conference on Pervasive Computing* (pp. 279-296). Springer, Berlin, Heidelberg.

C. Danial. (2022). E-commerce as a percentage of total retail sales worldwide from 2015 to 2025.

Chang, M.K., Cheung, W., Lai, V.S., (2005), Literature derived reference models for the adoption of online shopping. Inf. Manag. 42(4) (pp. 543–559).

Chen, B., Tai, P. C., Harrison, R., & Pan, Y. (2005, August). Novel hybrid hierarchical-K-means clustering method (HK-means) for microarray analysis. In 2005 IEEE Computational Systems Bioinformatics Conference-Workshops (CSBW'05) (pp. 105-108). IEEE.

Cumby, C., Fano, A., Ghani, R., & Krema, M. (2005, January). Building intelligent shopping assistants using individual consumer models. In *Proceedings of the 10th international conference on Intelligent user interfaces* (pp. 323-325).

Duwairi, R., & Abu-Rahmeh, M. (2015). A novel approach for initializing the spherical K-means clustering algorithm. Simulation Modelling Practice and Theory, 54, 49-63

Efficient-Apriori — Efficient-Apriori 2.0.1 documentation. (2022). Efficient Apriori Documentation. https://efficient-apriori.readthedocs.io/en/latest/

Frawley, W. J., Piatetsky-Shapiro, G., & Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *AI magazine*, *13*(3), 57-57.

Fu, H., Manogaran, G., Wu, K., Cao, M., Jiang, S., & Yang, A. (2020). Intelligent decision-making of online shopping behaviour based on the internet of things. *International Journal of Information Management*, *50*, 515-525.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harvey Norman. (2022). Harvey Norman. https://www.harveynorman.com.my

Household, Electrical, Hardware Products & More | MR.DIY. (2022). Household, Electrical, Hardware Products & More | MR.DIY. https://www.mrdiy.com.my/default/

Islam, M. M., Lam, A., Fukuda, H., Kobayashi, Y., & Kuno, Y. (2019). An intelligent shopping support robot: understanding shopping behaviour from 2D skeleton data using GRU network. Robomech Journal, 6(1), 1-10.

Katz, M. (1997). Technology forecast: 1997. *Menlo Park, CA: Price Waterhouse Technology Centre*.

Kumar, T. S. (2020). Data mining based marketing decision support system using hybrid machine learning algorithm. Journal of Artificial Intelligence, 2(03), 185-193.

Keita, B. (2020, September 28). *What are Scrum Ceremonies?* Invensis Learning Blog. https://www.invensislearning.com/blog/what-are-scrum-ceremonies/

Leading Online Shopping Platform In Southeast Asia & Taiwan. (2022). Shopee. https://shopee.com/index.html

Liew, A. (2007). Understanding data, information, knowledge and their inter-relationships. Journal of knowledge management practice, 8(2), 1-16.

Lotus's | Shop Conveniently & Get Rewarded with Lotus's. (2022). Malaysia. https://www.lotuss.com.my/en

Mathew, S., & Joseph, J. (2014). Decision Fatigue. *LISTENING TO CONSUMERS OF EMERGING MARKETS*, 175.

Mehak, S., Zafar, R., Aslam, S., & Bhatti, S. M. (2019, January). Exploiting filtering approach with web scrapping for smart online shopping: Penny wise: A wise tool for online shopping. In 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-5). IEEE.

Miniwatt Marketing Group. (2022). Internet World Growth Statistics.

Mittal, R. (2021). Using Automated Predictive Analytics in an Online Shopping Ecosystem. In *Intelligent Computing and Applications* (pp. 235-244). Springer, Singapore.

Moon, N. N., Talha, I. M., & Salehin, I. (2021). An advanced intelligence system in customer online shopping behavior and satisfaction analysis. Current Research in Behavioral Sciences, 2, 100051.

Mughal, M. J. H. (2018). Data mining: Web data mining techniques, tools and algorithms: An overview. Information Retrieval, 9(6).

Naseri, R. N. N. (2021). What is a population in online shopping research? A perspective from Malaysia. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(4), 654-658.

Nature2U: Online Grocery Shopping Hacks You Need to Know. (2022, May 26). Nature2U. https://nature2u.com.my

Neelamegam, S., & Ramaraj, E. (2013). Classification algorithm in data mining: An overview. International Journal of P2P Network Trends and Technology (IJPTT), 4(8), 369-374.

Pignatiello, G. A., Martin, R. J., & Hickman Jr, R. L. (2020). Decision fatigue: A conceptual analysis. *Journal of health psychology*, *25*(1), 123-135.

Saleem, H., Muhammad, K. B., Nizamani, A. H., Saleem, S., & Aslam, A. M. (2019). Data Science and ML Approach to Improve E-Commerce Sales Performance on Social Web. *International Journal of Computer Science and Network Security (IJCSNS)*, *19*.

Saleem, H., Uddin, M. K. S., Habib-ur-Rehman, S., Saleem, S., & Aslam, A. M.

(2019). Strategic data driven approach to improve conversion rates and sales performance of e-commerce websites. *International Journal of Scientific & Engineering Research (IJSER)*.

Sixteen Clothing Free Website Template | Free CSS Templates | Free CSS. (2022). Sixteen Clothing | Free CSS Template. https://www.free-css.com/free-css-templates/page267/sixteen-clothing

smtplib — SMTP protocol client — Python 3.10.4 documentation. (2022). SMTP Documentation. https://docs.python.org/3/library/smtplib.html

Sollisch J. (2016) The cure for decision fatigue. Wall Street Journal. Available at: https://www.wsj.com/articles/the-cure-for-decisionfatigue-1465596928

Stanford (2018). Amazon review data. Stanford Data Repository.

https://nijianmo.github.io/amazon/index.html

Statista. (2022, March 28). *Number of monthly web visits on Shopee Malaysia Q1 2018-Q2 2021*. https://www.statista.com/statistics/1012761/malaysia-number-monthly-web-visits-shopee-quarter/#:%7E:text=In%20the%20second%20quarter%20of,in%20Southeast%20Asia%20and%20Taiwan.

Tierney J (2011) Do you suffer from decision fatigue? New York Times, p. 111.

Toth, A., Tan, L., Di Fabbrizio, G., & Datta, A. (2017, January). Predicting shopping behavior with mixture of RNNs. In eCOM@ SIGIR.

UCI Machine Learning Repository: Amazon Access Samples Data Set. (2022). UCI Machine Learning Repository: Amazon Access Samples. https://archive.ics.uci.edu/ml/datasets/Amazon+Access+Samples

Weiss, S. M., & Indurkhya, N. (1998). *Predictive data mining: a practical guide*. Morgan Kaufmann.

Wikipedia contributors. (2022, May 8). *Long short-term memory*. Wikipedia. https://en.wikipedia.org/wiki/Long_short-term_memory

Zaha, J. M., Barros, A., Dumas, M., & Hofstede, A. T. (2006, October). Let's dance: A language for service behaviour modelling. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 145-162). Springer, Berlin, Heidelberg.

Zahra, S., Ghazanfar, M. A., Khalid, A., Azam, M. A., Naeem, U., & Prugel-Bennett, A. (2015). Novel centroid selection approaches for KMeans-clustering based recommender systems. Information sciences, 320, 156-189.

Zeng, M., Cao, H., Chen, M., & Li, Y. (2019). User behaviour modelling, recommendations, and purchase prediction during shopping festivals. Electronic Markets, 29(2), 263-274.