CUSTOMER PURCHASE PREDICTION AND PRODUCT RECOMMENDATIONS

BY

WONG JI HIN

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Customer Purchase Prediction and Product Recommendations

Academic Session: June 2024

I WONG JI HIN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:
70, Tanah Hitam,
31200 Chemor,
Perak

Kh'ng Xin Yi
Supervisor's name

Date: 11-Sep-2024

Date: 13/9/2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF _Information and Communication Technology__

UNIVERSITI TUNKU ABDUL RAHMAN

Date: _11-Sep-2024____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Wong Ji Hin***_________________ (ID No: __***21ACB00298***__) has completed this final year project/ dissertation/ thesis* entitled "___Customer Purchase Prediction and Product Recommendations _____" under the supervision of _Dr Kh'ng Xin Yi__ (Supervisor) from the Department of _Computer Science__, Faculty/Institute* of _Information and Communication Technology__ , and ____________________ (Co-Supervisor)* from the Department of _________________________, Faculty/Institute* of ___________________________.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Wong Ji Hin*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**CUSTOMER PURCHASE PREDICTION AND PRODUCT RECOMMENDATIONS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________

Name : ___Wong Ji Hin______________________

Date : __11-Sep-2024_______________________

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Kh’ng Xin Yi. Her unwavering support and guidance have helping me a lot through this project. and I am deeply grateful for entrusting me with such an extraordinary chance. A million thanks to you for your belief in my abilities and for inspiring me to push the boundaries of my knowledge.

Furthermore, I would like to say thanks to my moderator in charge, Ts Lim Seng Poh. Finally, I must say thanks to my parents and my entire family for their support and boundless love throughout this journey.

ABSTRACT

Understanding customer behaviour is important for businesses that seeking growth and sustainability in today's competitive market. This project has three main objectives where the first objective is to develop a customer purchase prediction model to estimate the probability of future purchases. The second objective is to conduct market basket analysis to gain insights into customer shopping patterns. The third objective is to integrate customer purchase prediction and market basket analysis to provide different product recommendations for different customer groups. This project tries to provide an integrated solution that can improve the customer satisfaction and produce revenue growth by using the strength of data analytics. The BG/NBD model is used to determine the optimal period that have highest prediction accuracy and it was determined that the six-month period is the optimal period with highest prediction accuracy. Then, the RFM segmentation categorizes customers into distinct groups such as where 50.74% of customers have been classified as "At risk," 46.98% as "Loyal customers", 1.62% as "Hibernating" and only 0.66% as "Champions". Besides, both Apriori or the FP-growth algorithms are compared and find out that Apriori is faster for small datasets while FP-Growth is more efficient with large datasets due to its lower memory consumption. Thus, the FP-Growth algorithm is applied for each customer groups to perform Market Basket Analysis to discover frequent itemsets and provide product recommendations to each customer groups. The top-recommended items are different for each customer group with "PARTY BUNTING" being the most popular for both "Champions" and "Hibernating" customers, "REGENCY CAKESTAND 3 TIER" for "Loyal customers", and "WHITE HANGING HEART T-LIGHT HOLDER" for "At risk" customers. Furthermore, it was discovered that "At risk" customers generated fewer frequent itemsets, indicating less diverse purchase behaviour. In conclusion, this project is able to provide a better understanding of complex customer purchase patterns. The integration of those model offers practical tools to improve the customer engagement and enhance sales performance across various customer groups.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Background Information

Nowadays, understanding and utilizing customer behaviour has never been more important for an industry to success in their business and increase the competitiveness of itself. Thus, this is where the technology with its continuous improvement and development and play important role in almost every field, not only business field. The industries are continuously looking for the new way that can help them to increase their competitiveness and one of the most transformative applications of technology in the business field is the implementation of customer purchase prediction model. With the help of the customer purchase prediction model, it allows the companies to plan their strategies lead to enhance the customer satisfaction and revenue growth. Thus, the importance of customer purchase prediction cannot be overstated.

There are several algorithms that can be applied to the customer purchase prediction model such as Apriori and FP-growth algorithms [1]. In this project, it primarily focuses on two important models which are customer purchase prediction model and the product recommendation model. For customer purchase prediction model, it will use the BG/NBD model to build this model. It is used to predict the customer they will continue to make future purchase.

On the other hand, the product recommendation model uses the Apriori or FP-growth algorithm to carry out the market basket analysis (MBA). Both algorithms are used to find out the frequent itemset and generate the association rules. Then, the product recommendation system utilizes these rules and recommend the item to the customer based on the specific item.

This project aims to provide a comprehensive solution that not only helps businesses to have a better understanding on their customers which will lead to improved customer satisfaction and sustained revenue growth by integrating both customer purchase prediction and product recommendation models.

1.2 Problem Statement

For most of the people, especially those working in business field such as the retailers, it is difficult to understand the complex purchase pattern of the products that include in the purchased item of the customer. First, understand the purchase pattern of the products is a challenging task. This is due to the fact that the dataset is required to conduct the necessary analysis in order to understand the purchase pattern and figure out the frequent itemset. The retailers can discover what items are frequently purchased by the customer and make sure the stock is enough to avoid the occurrence of overstocking or understocking which would result in financial losses. Other than that, the customer may need the recommendation of the product to get along with the product there may not considered by them. As a result, it will improve both the shopping and decision-making processes.

Besides, it is also difficult to predict whether a customer will continue to make future purchases or not. By performing the prediction, the company or the organization can avoid revenue loss and maintain the customer relationships which will bring benefits to the business such as the long-term financial success. It can help ensure a consistent revenue stream and can have a significant impact on a company's profitability. This project tries to overcome these issues with the help of a client purchase prediction model and deliver of the product recommendation model. It is appropriate for enterprises and organizations to make better decisions and build more successful strategies.

1.3 Motivation

The study's purpose is to create a method for people in relevant fields, particularly those within the business field to understand the buying pattern of the products to recommend product and predict the likelihood of customer to make further purchase. The end product of the project will be customer purchase prediction model and product recommendation model. The product recommendation model is designed to recognise the relationship between items and extract the frequent patterns of the itemsets from a large dataset using the Apriori and FP-growth algorithm. Hence, it can suggest the item for specific product based on these discovered patterns. Other than that, the customer purchase prediction model also can predict the customer who are likely to make future purchases. The fundamental goal of this study is to promote the practical utility and adoption of the combination of the customer purchase prediction models and product recommendation model. Additionally, this project aims to show how the model can be implemented into real-world scenarios so that it can help people in the decision-making and improve the business. Finally, it shows that client purchase prediction may be conducted not just for major corporations, but also for small grocery stores. In conclusion, this project is motivated by a desire to provide business professionals with an in-depth understanding of consumer behaviour, by providing practical models for product recommendation and purchase prediction, along with promoting the adaptability of these tools for organizations of all sizes.

1.4 Project Scope and Project Objectives

1.4.1 Project Scope

The scope of this project is to develop two models which are the customer purchase prediction model and the product recommendation model. The customer purchase prediction model utilizing BG/NBD model and it will enable the people to predict which customers are likely to make future purchases. Simultaneously, the product recommendation model will deliver the product recommendations to customer by using the algorithms known as Apriori and FP-growth. Both models are developed using Python programming language. Other than that, the project extends its scope to identify the patterns of frequently purchased items within the transaction dataset with the association rules so that the frequent patterns between items can be listed out with the use of product recommendation model. It figures out the associations and co-occurrences between items in customer transactions. With the frequent patterns that identified, the project provides the product recommendation that suggests products commonly purchased together to the customer based on the items that have been selected by them.

1.4.2 Project Objectives

The objectives of this project are:

1. **To develop a customer purchase prediction model to estimate the probability of future purchases.**

   The advanced techniques such as BG/NBD model has been applied in the customer purchase prediction model. This model will analyze the dataset and determine the likelihood of customers making future purchases. The model's output will provide significant insights into customer behavior and enable the businesses to customize marketing strategies and optimize inventory management.

2. **To conduct market basket analysis to gain insights into customer shopping patterns.**

   The frequent itemsets can be discovered and the association rules can be generated from transaction data by using the algorithms such as Apriori algorithm and FP-Growth algorithms. These results will provide information on how customers make purchasing decisions and showing which products are frequently purchased in combination. It is useful for businesses to optimize product placement, store layouts and marketing strategies.

3. **To integrate customer purchase prediction and market basket analysis to provide different product recommendations for different customer groups.**

   This project aims to integrate customer purchase prediction with the use of BG/NBD model and market basket analysis to provide product recommendations to each customer group. The proposed approach first testing the BG/NBD model with different time periods to determine the optimal period that have highest prediction accuracy. The data is then split using RFM analysis to categorize customers into different group such as "Champions" and "At risk". Then, the algorithm such as Apriori or FP-Growth algorithms is used to find out the frequent itemsets and provide product recommendations to each customer group. This will help the businesses to identify and recommend items to each customer group and improve the shopping experience of the customer.

1.5 Contribution

One of the contributions from this project is to improve the understanding of people about the complex purchase patterns across different industries. It provides a systematic framework for businesses to decode and analyse these patterns, enabling them to make data-driven decisions. This understanding is invaluable for optimizing product offerings, inventory management, and marketing strategies.

Another contribution of this project is the development of an innovative customer purchase prediction model with BG/NBD model. It not only identifies the existing patterns but also predicts which customers are likely to make future purchases. With this model, the retailers or the businesses are able to predict or find out the customer that have bigger chance to make purchases in the future such as next 5 days, next week or even next month. As a result, businesses may concentrate their advertising strategies on clients who are going to make future purchase. For example, a store may discover the prediction that a particular customer is likely to make a future purchase in next week. Then, the retailer can send the discount offer email to the customer in order to enhance the probability of that customer making another purchase and resulting in increased sales for the shop.

Other than that, the development of product recommendation model also is one of the contributions that can enhance the overall customer experience. Unlike typical product recommendations which focus on individual consumer behaviour, it recommends the product by identifying the frequent itemsets from all customer's transactions. It recommends the additional products that are commonly purchased along with the specific item with the use of algorithms such as Apriori algorithm and FP-growth algorithm. For example, when a consumer selects a product, a similar product that the customer may not have considered or that is frequently purchased together can be recommended. Thus, it is saving the time for the customer.

Another contribution of this project would be the identification of frequent itemsets or association rules that generated by the product recommendation model based on the dataset provided. The patterns of items that are commonly purchased together can be detected by analysing transaction data from all customers and this will provide useful insights to the businesses. Thus, allow them to optimize product placements and store layout. For example, the grocery store has identified the frequent itemsets and noticed that the butter and jam were purchased together when the customer purchased bread in the same transaction. With this understanding, the retailer can arrange the bread, jam and butter in the same location and increase the likelihood that the customer will pick up butter or jam along with bread.

1.6 Report Organization

This report is divided into six chapters and each chapter discussed the different aspects of the project. For Chapter 1 Introduction, it provides a comprehensive overview of the entire project. This chapter includes the background information, problem statement, motivation, project scope, project objectives, contributions and the overall structure of the entire report. For Chapter 2 Literature Review, the related backgrounds to this project are reviewed where it includes Market Basket Analysis, data mining algorithm such as Apriori and FP-Growth algorithms, BG/NBD model and RFM model. All the details of the related work are discussed here and serves as the foundation for the following chapters. In Chapter 3 Methodology, the entire model design or the flow of the model is discussed which included data processing, customer prediction, Market Basket Analysis and customer recommendations. This chapter provides a full explanation of the chosen approach and ensures that readers understand the approaches used. For Chapter 4 Implementation, it explained in detail on how to implement every single of part with code. Next, is Chapter 5 Model Evaluation and Discussion. In this chapter, it focuses on discussing the results and the evaluation of the model in this project. It also discussed about the comparison between Apriori and FP-Growth algorithms, and validation of the model. Finally, Chapter 6 Conclusion and Recommendations provides a summary of the project. It also provides several recommendations for future work to improve the entire project.

Chapter 2

Literature Review

2.1 Market Basket Analysis and Data Mining Algorithm

2.1.1 Overview of Market Basket Analysis

Market Basket Analysis (MBA) is a strong statistical approach that focuses on uncovering the correlations among products that are frequently purchased together by the client when the items are purchased. MBA's major purpose is to study the customer purchasing patterns and predict the items that customers purchase together [1]. Besides, MBA also play a role in providing businesses with valuable insights to improve their marketing strategies, optimize product placements, and lastly increase revenue. By carrying out the MBA, the marketing team will gain a better grasp of the customer purchasing pattern and which is going help them in the decision-making process and develop better pricing strategies.

Other than that, Market Basket Analysis is also referred to associations rule mining and it helps to find out the customer purchasing patterns inside massive data sets such as purchase history or transaction history by using the association rules. Instead of applying associations rule mining in fields of marketing and sales, it also widely used in various industries such as medicine, biological sciences and more [2,3]. It often used in the recommendation system. The association rule refers to an if/then statement that is used to discover relationships between independent variables from a data set, such as associations, correlations, and patterns [4]. For example, if item X appears, then item Y is also likely to appear together. It often includes two fundamental components in association rules which are support and confidence [5].

2.1.2 Apriori Algorithm

The Apriori algorithm is a fundamental approach in data mining and association rule discovery that was developed in 1994 by Agrawal and Srikant. It is used as a way to efficiently mine frequent itemsets from large datasets to and generate Boolean association rules [6]. Frequent item sets are groups of products that appear together in transactions more frequently than a preset criterion, such as minimum support. The association rules also can be known as the implication that suggest or predict the presence of items based on the presence of another items. The Apriori method uses three parameters: support, confidence, and lift, and the equations for the parameters are listed below [1].

$$\text{Support (A)} = \frac{\text{Count (A)}}{\text{N}} \tag{2.1}$$

$$\text{Support } (\text{A} \rightarrow \text{B}) = \frac{\{\text{A, B}\}}{\text{N}} \tag{2.2}$$

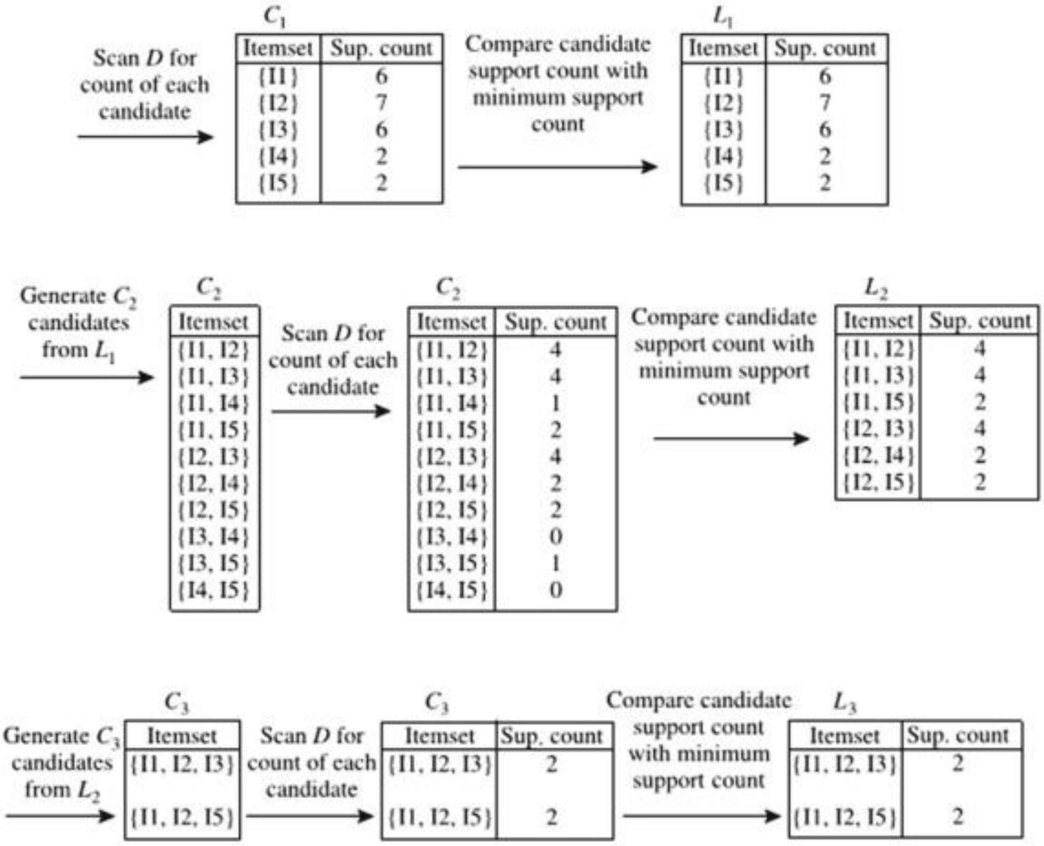
$$\text{Confidence } (\text{A} \rightarrow \text{B}) = \frac{\{\text{A, B}\}}{\text{Support } \{\text{A}\}} \tag{2.3}$$

$$\text{Lift } (\text{A} \rightarrow \text{B}) = \frac{\{\text{A, B}\}}{\text{Support } \{\text{B}\}} \tag{2.4}$$

The support of item A is defined by the equation (2.1), where the count (A) is the amount of item A appears within the transactions data and N is the total amount of transactions. The equation (2.1) is used to determine the frequency of transactions of item A appears in the dataset by dividing the total number of transactions. It is used to find out how often the specific item appeared in all transactions. Besides, the equation (2.2) is defined as the percentage that include both item A and item B in the transactions by dividing the total number of transactions. The *{A, B}* parameter is the number of transactions that comprise item A and item B. The itemsets with more than or equivalent to the minimum support threshold are considered frequent and association rules is generated according to frequent itemsets. Then, the confidence for rule A→ B is calculated using the equation (2.3) and confidence is an important parameter that measured the strength of an association between items in the dataset. The equation (2.3) is used to find out the number of times item B occurs in all transactions that contain item A and divide it by the number of transactions that contain item A. In other words, it is used to find out how often the item B will appear or being purchased when item A is purchased, and it considers the popularity of item A in this equation. Unlike equation (2.3), equation (2.4) considers the popularity of item B, the Lift (A → B) refers to the rise in the ratio of sale of item B when A is sold. The lift value can show how items are connected. A lift value larger than 1 indicates a significant relationship between item B and the co-occurrence of another item, whereas a lift value less than 1 indicates that item B is less probable to appear together. If the lift value is equal to 1, it means that both items are independent and did not affect each other [1].

The Apriori method has two primary steps where it first determines all of the common item sets in the dataset and then generates the associated rules. It finds the relationship between item sets which is K-itemsets by using the method of layer-by-layer search and form the rules [7]. By figuring out the relationship or the rules that formed, it can help in decision-making process. For example, the retailers can know which items are frequently purchased together and it allow them to optimize the product placement like put both frequently purchased item together and increase the sales.

The primary stage in how Apriori algorithm works is scan the database to get all the frequent itemsets by calculate the support value of each item and record the frequent itemsets that fulfill the minimum support threshold that has been set before the scanning start. In this step, the frequent 1-itemsets can be determined where the individual items satisfy or exceed the minimum support threshold. After that, the algorithm generates the frequent k-itemsets according to the frequent (k-1)-itemsets that generate in previous step. The candidate k-itemsets that did not meet the required level of support will be eliminated. The algorithm continues to scan the database and repeat the previous process in order to get all the frequent itemsets until there do not exist any frequent k-itemsets can be created. Lastly, the frequent (k-1)-itemset that meet the minimum support threshold is generated and the association rules as the outputs [6,7]. The outputs can provide better grasp of the link between items within the dataset and has the potential to apply it in the fields such as market basket analysis, recommendation systems and more. The example of the process of Apriori algorithm work is shown in figure 2.1.


Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_2 candidates from L_1
Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_3 candidates from L_2
Scan D for count of each candidate
Compare candidate support count with minimum support count

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

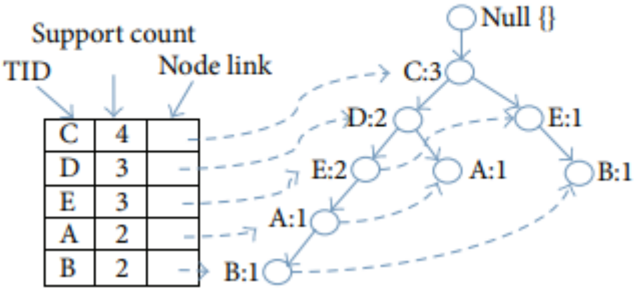
Figure 2.1: Example of the generation of frequent itemsets. Adapted from [6]

One of the positive aspects of the Apriori algorithm is that it is easier to grasp due to its low spatial complexity. As a result, the Apriori algorithm is straightforward to grasp. Aside from that, the parameters such as the minimum support and confident threshold can be modified to limit the number of times the database is scanned. It also reduces the size of itemsets and perform well. However, the Apriori algorithm has its drawbacks, such as low algorithm efficiency because each item in the frequent item set must scan the dataset once, which implies the dataset must be scanned as many times as the items in the frequent item set. Furthermore, a large number of frequent item sets may be generated during the process and requiring extra time and main memory [8].

2.1.3 Frequent Pattern Growth Algorithm

The Frequent Pattern Growth (FP-Growth) algorithm is a popular and efficient algorithm that used to figure out frequent itemsets without the need of generating the candidate itemsets and this algorithm is proposed by Han [6]. It is developed to overcome the limitations of Apriori algorithm such as large number of candidates itemsets generated and it became larger when the database was huge and the Apriori algorithm need to scan the database multiple times to check the support of each itemset and lead to high costs and require more time. The strategy of FP-Growth algorithm use is the divide-and-conquer strategy and it use the tree where it called frequent pattern tree to represent the database. The frequent pattern tree is created by the formation of nodes and each node represents an item within the itemset. The first node is the root node and it will remain null while the lower nodes are the itemsets and form the association rule [9].

The first step of how FP-Growth algorithm work is the algorithm will scan the database and get the support of each item and get the frequent itemsets. This step is just same as the first step of Apriori algorithm. Then, store the item within frequent itemsets that obtained in the list and sort the items in decreasing order of support count. After that, the algorithm creates the root node of the tree by labelled the root node with null and create the tree. Next, the algorithm required to scan the database for second time to build the frequent pattern tree by order each item in the first transaction according to the sequence of the list. The first transaction will generate the first branch of the tree by link each node together. It is following by the second transaction in the dataset, and it also will ordered in descending order of count. If any item in the second transaction is already present in the previous transaction or branch, then it will share the same node with the previous branch. In this way, the algorithm will make the count of the common node increased by 1 when they are linked together according to the transactions. The following step is to mine the frequent pattern tree that created from previous step. The algorithm creates a conditional pattern base where it consists the prefix path set that have lowest node or called suffix pattern and following by create a conditional FP-tree that used to generate the frequent patterns [10].



C	4	
D	3	
E	3	
A	2	
B	2	

Figure 2.2: Example of FP-tree. Adapted from [10]

One of the advantages of FP-Growth algorithm is FP-Growth algorithm only need to scan the database twice only compared to Apriori algorithm where Apriori algorithm need to scan the database when there is one candidate itemset is generated. As a result, the FP-Growth method has a shorter execution time and is more efficient since it does not produce candidate itemsets. Besides, it is time consuming to build the FP-tree if the dataset is huge [6, 11].

2.2 Purchase Prediction

2.2.1 BG/NBD model

The Beta Geometric / Negative Binomial Distribution model, often known as the BG-NBD model, acts as one of the probabilistic models used to compute customer lifetime value (CLV) and predict future behaviour based on each customer's purchasing history. The model is proposed to improve the existing models such as the Pareto/NBD model where both model is used to predict the customer’s future activity by considering the past buying behaviour and the main difference between both model is how the dropout process handled in both models. In BG/NBD model, it required three measures for each customer which are recency, frequency and time period [12].

Based on the customer's purchasing history, the BG/NBD model estimates the number of times the customer will make a buying decision or order in a particular period assuming the customer is still alive after a set time. There are several assumptions that have been stated and listed below [12]:

- The customer will remain alive for a specified period of time or will be permanently inactive throughout their lifetime.
- While active, the number of transactions performed by a client follow a Poisson process with transaction rate λ where *t_j* is the time of the *j*th purchase, λ (lambda) is the transaction rate for customer, *e* is the base of the natural logarithm.

$$f\left(t_j \middle| t_j - 1;\ \lambda\right) = \lambda e^{-\lambda\,(t_j - t_{j-1})}, t_j > \ t_{j-1}\ \geq 0 \tag{2.1}$$

- Heterogeneity in transaction rate λ across customers follows a gamma distribution where α (alpha) is the scale parameter, *r* (beta) is the shape parameter and *Γ(r)* is the gamma function and defined as (*r* - 1)!.

$$f(\lambda\,|r, \alpha) = \frac{\alpha^r \lambda^{r-1} e^{-\lambda\alpha}}{\Gamma(r)}\ , \lambda > 0 \tag{2.2}$$

- The client turns into inactive with probability p after any transaction. As a result, the moment at which a client "drops out" is dispersed over transactions using a geometric distribution with a probability mass function (pmf).

$$P = p\ (1-p)^{j-1}\ , j = 1, 2, 3\ ... \tag{2.3}$$

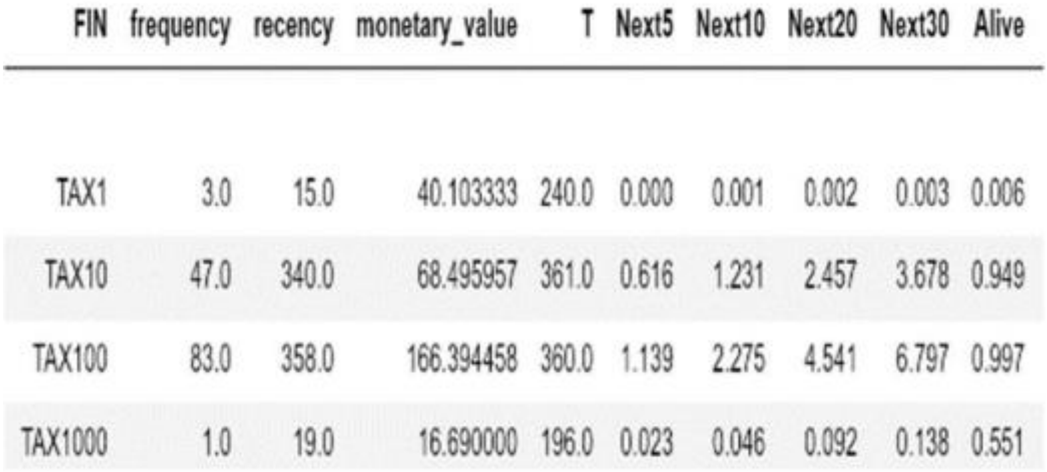
- Heterogeneity in *p* across customers follows a beta distribution where *B(a,b)* is the beta function and *a* and *b* are the shape parameter of beta distribution.

$$f(p\ |a, b) = \frac{p^{a-1}\ (1-p)^{b-1}}{B(a,b)}\ , 0\ \leq p\ \leq 1 \tag{2.4}$$

- The transaction rate λ and the dropout probability *p* differ amongst clients.

Thus, the prediction of the number of transactions that done by customer can be calculated by using these assumptions.

In paper [12], the author used this model to forecast the number of future transactions for financial institutions such as banks because it only requires three features to calculate the customer's lifetime, compared to other classification algorithms that require around 300 features, and the BG/NBD model takes less time make it is more suitable. In this paper, it predicted the amount of repurchases in the next 5, 10, 20, and 30 days based on parameters such as recency, which is the time difference between the customer's first and last purchase, frequency, which is the total number of purchases that the customer made in the specific time period, and time period in this case represents the customer's age. Below is the sample of the prediction of number of repeat purchase.



FIN	frequency	recency	monetary_value	T	Next5	Next10	Next20	Next30	Alive
TAX1	3.0	15.0	40.103333	240.0	0.000	0.001	0.002	0.003	0.006
TAX10	47.0	340.0	68.495957	361.0	0.616	1.231	2.457	3.678	0.949
TAX100	83.0	358.0	166.394458	360.0	1.139	2.275	4.541	6.797	0.997
TAX1000	1.0	19.0	16.690000	196.0	0.023	0.046	0.092	0.138	0.551

Figure 2.3: Example of the prediction of number of repeat purchase. Adapted from [12].

2.2.2 RFM model

The RFM model is a customer segmentation strategy that commonly used in marketing to group the customer into different categories and it can used to analyze the customer behavior. The RFM model evaluates the customer by considering three crucial variables which are recency, frequency and monetary [13]. The recency is defined as the number of days since the last purchase while frequency is stand for total number of purchases within the period and monetary refers to the total amount of money spent by a customer within the period [14].

Each factor is then scored based on customer transaction data and the score is typically on a scale of 1 to 5. The customers are often divided into five quantiles and the top quantile is given a score of 5, followed by 4, 3, 2 and 1 for both frequency and monetary [14]. The opposite approach is used for recency where only the customer who purchase in most recent is given a score of 5 and those who have not made a recent purchase will give lower score. This is because recency refers to the time since a customer made transactions and customer who made a recent purchase are considered more likely to engage again soon.

Once the RFM scores are calculated for each customer, they can be segmented into different group. For example, a customer with a score 555 can be categorized as the potential customer or ideal customer compared to the customer with a score 111 which is considered as the worst customer group [13].

Chapter 3

Methodology

3.1 Overall System Design

The figure 3.1.1 illustrates the overall process of this project which aims to predict customer purchases and generate product recommendations through a combination of customer behavior modeling and Market Basket Analysis (MBA). The process is divided into several distinct stages where each of it contributes to the final goal of this project. The key components of the overall project include data collection, data preprocessing, customer prediction using BG/NBD model, Market Basket Analysis, customer segmentation using RFM analysis and top 10 product recommendations for each customer group. In this section, , each step of the workflow will be explained in detail by emphasize how they interconnect and contribute to the final result. Thus, the methodology part is important in providing an in-depth explanation of how the data is processed, analyzed, and generated recommendations to the customer.

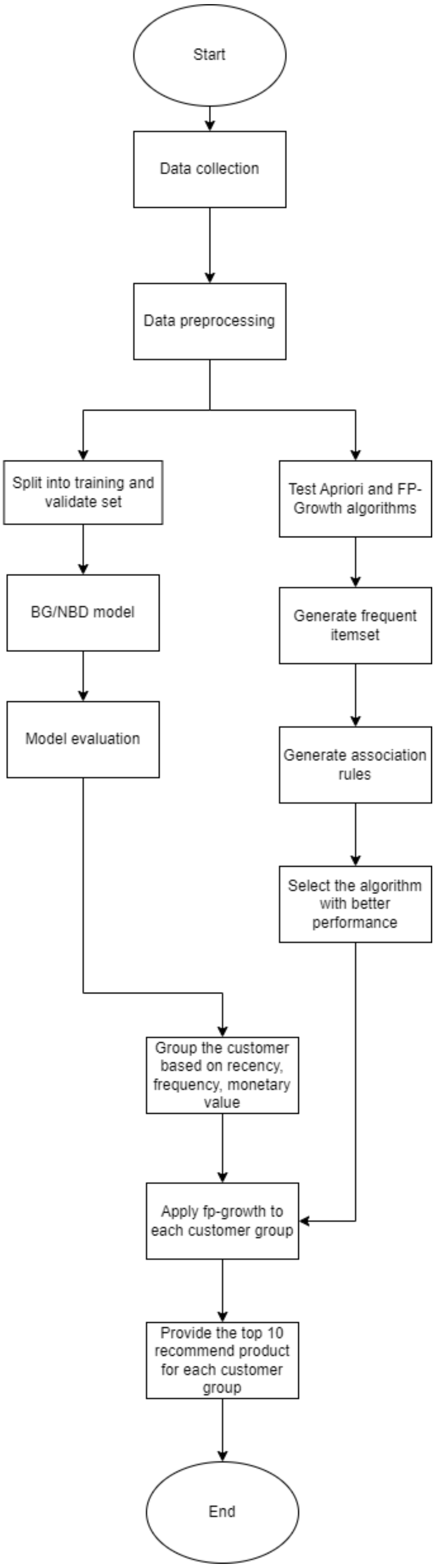
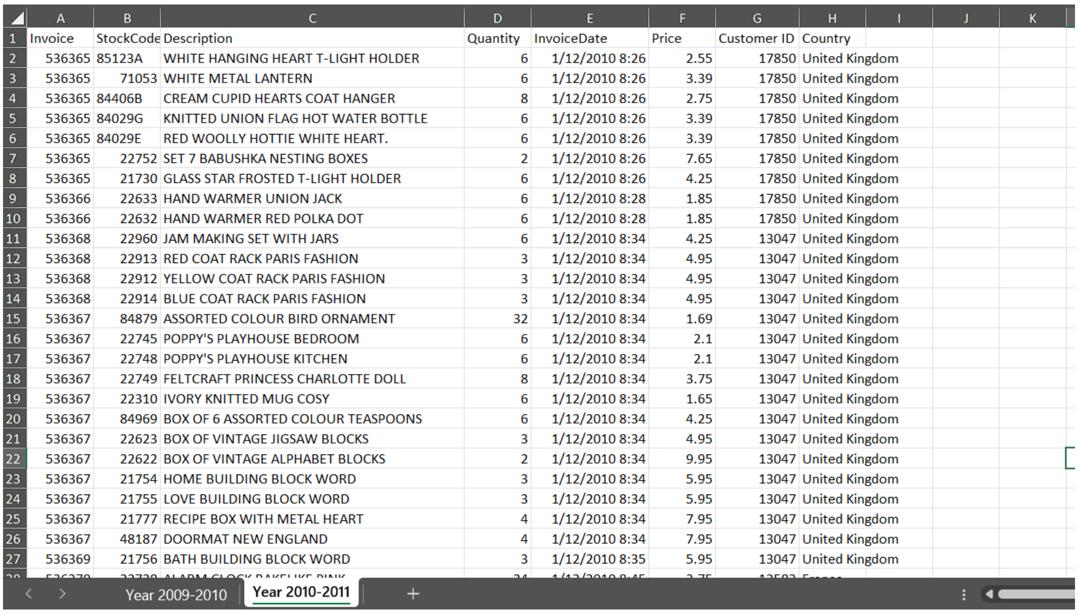

Start
Data collection
Data preprocessing
Split into training and validate set
Test Apriori and FP-Growth algorithms
BG/NBD model
Generate frequent itemset
Model evaluation
Generate association rules
Select the algorithm with better performance
Group the customer based on recency, frequency, monetary value
Apply fp-growth to each customer group
Provide the top 10 recommend product for each customer group
End

Figure 3.1.1: General flowchart of proposed system.

3.2 Data Processing

3.2.1 Dataset Used

The dataset used in this project is s Online Retail II dataset that obtained from UCI Machine Learning Repository. It is the dataset that contains all transactions for online retail from 9 December 2009 until 9 December 2011. Inside the dataset, it separated to two sheets which are “Year 2009-2010” and “Year 2010-2011”, the sheet “Year 2010-2011” is used in this project. It included a total of 541910 instances with 8 variables which are Invoice, StockCode, Description, Quantity, InvoiceDate, Price, Customer ID and Country.



	A	B	C	D	E	F	G	H
1	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
2	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	1/12/2010 8:26	2.55	17850	United Kingdom
3	536365	71053	WHITE METAL LANTERN	6	1/12/2010 8:26	3.39	17850	United Kingdom
4	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	1/12/2010 8:26	2.75	17850	United Kingdom
5	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	1/12/2010 8:26	3.39	17850	United Kingdom
6	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	1/12/2010 8:26	3.39	17850	United Kingdom
7	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	1/12/2010 8:26	7.65	17850	United Kingdom
8	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	1/12/2010 8:26	4.25	17850	United Kingdom
9	536366	22633	HAND WARMER UNION JACK	6	1/12/2010 8:28	1.85	17850	United Kingdom
10	536366	22632	HAND WARMER RED POLKA DOT	6	1/12/2010 8:28	1.85	17850	United Kingdom
11	536368	22960	JAM MAKING SET WITH JARS	6	1/12/2010 8:34	4.25	13047	United Kingdom
12	536368	22913	RED COAT RACK PARIS FASHION	3	1/12/2010 8:34	4.95	13047	United Kingdom
13	536368	22912	YELLOW COAT RACK PARIS FASHION	3	1/12/2010 8:34	4.95	13047	United Kingdom
14	536368	22914	BLUE COAT RACK PARIS FASHION	3	1/12/2010 8:34	4.95	13047	United Kingdom
15	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	1/12/2010 8:34	1.69	13047	United Kingdom
16	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	1/12/2010 8:34	2.1	13047	United Kingdom
17	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	1/12/2010 8:34	2.1	13047	United Kingdom
18	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	1/12/2010 8:34	3.75	13047	United Kingdom
19	536367	22310	IVORY KNITTED MUG COSY	6	1/12/2010 8:34	1.65	13047	United Kingdom
20	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	1/12/2010 8:34	4.25	13047	United Kingdom
21	536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	1/12/2010 8:34	4.95	13047	United Kingdom
22	536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	1/12/2010 8:34	9.95	13047	United Kingdom
23	536367	21754	HOME BUILDING BLOCK WORD	3	1/12/2010 8:34	5.95	13047	United Kingdom
24	536367	21755	LOVE BUILDING BLOCK WORD	3	1/12/2010 8:34	5.95	13047	United Kingdom
25	536367	21777	RECIPE BOX WITH METAL HEART	4	1/12/2010 8:34	7.95	13047	United Kingdom
26	536367	48187	DOORMAT NEW ENGLAND	4	1/12/2010 8:34	7.95	13047	United Kingdom
27	536369	21756	BATH BUILDING BLOCK WORD	3	1/12/2010 8:35	5.95	13047	United Kingdom

Figure 3.2.1.1: Sample of dataset used.

3.2.2 Data Preprocessing

Before the data is applied into the model, the data is required to undergo preprocessing to ensure the accuracy and relevance of the input data for further modeling and evaluation. In this dataset, there are some rows that contain missing values especially in the “Description” and “Customer ID” column. Thus, rows with any missing values are dropped to ensure that there are only clean and complete records are used in the following sections to prevent any inaccuracies of the result. Then, the “Invoice” column contains the transaction records, and each cancelled order is recognized when the

"Invoice" record is started with the letter "C". These cancellation orders need to filter out to make sure that only successful transactions were considered. Besides, some records contained non-positive values in the "Quantity" also need to remove as it indicates that the item is not valid. After that, a new column called "TotalPrice" is created that represents the total monetary value for each transaction and it is crucial for calculating the monetary value in RFM segmentation. Lastly, the extra spaces in the "Description" column also need to remove to avoid any issues that caused by inconsistent formatting in product description.

3.3 Customer Prediction

3.3.1 Training and Testing Data

After preprocessing, the data was separated into training and testing set. The training set was used to train the BG/NBD model to estimate the number of transactions for each customer. The testing set is used to access the model's accuracy by comparing the predicted results with actual customer transaction data.

3.3.2 BG/NBD Model

The BG/NBD (Beta-Geometric/Negative Binomial Distribution) model is chosen for the model training set because it is widely used to predict the probability of future purchases. The model works based on the equation below.

$$f(\lambda\,|r,\alpha) = \frac{\alpha^r \lambda^{r-1} e^{-\lambda\alpha}}{\Gamma(r)}\,, \lambda > 0 \qquad (3.2)$$

In this formula, the parameter λ is the transaction rate where it represents the expected number of purchases made by customer in a specific time period. Then, the parameter *r* (beta) is the shape parameter and α (alpha) is the scale parameter that will influence the shape and scale of the gamma distribution while the *Γ(r)* is the gamma function to ensure the normalization of the distribution. These parameters control the shape and scale of the gamma distribution which represents the customer's transaction rate over time. By fitting this model to the dataset, the likelihood of future purchases can be predicted.

3.3.3 Model Evaluation

After training the BG/NBD model, the performance of the BG/NBD model is evaluated by comparing the predicted number of purchases against the actual number of purchases. This comparison assesses how accurately the model can predict the future transactions based on the historical data of recency, frequency, and monetary value. The evaluation process involves calculating the error metrics such as RMSE to measure the different between predicted and actual values, hence ensuring the model's effectiveness in predicting customer behavior. The equation for the Root Mean Squared Error (RMSE) is shown below. The parameter N is representing the number of data while the parameter y_i represents the actual value and the parameter $\breve{y}_i$ represents the predicted value.

$$RMSE = \sqrt{\frac{\Sigma(y_i - \breve{y}_i)^2}{N}}, \lambda > 0 \tag{3.3}$$

Furthermore, the graph is used to visualize the performance of the model so that it is easier to interpret and understand.

3.4 Market Basket Analysis (MBA)

3.4.1 Apriori Algorithm

After preprocessing, the preprocessed data is used for association rule mining to generate frequent itemsets and determines the associations between items using association rules. The Apriori algorithm is applied to construct frequent itemsets based on the minimum support value. After generating the frequent itemsets, the association rules will be derived from them to identify the relationship between items. The equations for MBA which are support, confidence and lift is shown below [1].

$$\text{Support (A)} = \frac{\text{Count (A)}}{\text{N}} \tag{3.1}$$

$$\text{Confidence } (\text{A} \rightarrow \text{B}) = \frac{\{\text{A}, \text{B}\}}{\text{Support } \{\text{A}\}} \tag{3.2}$$

$$\text{Lift } (\text{A} \rightarrow \text{B}) = \frac{\{\text{A}, \text{B}\}}{\text{Support } \{\text{B}\}} \tag{3.3}$$

The equation (3.1) is used to measure how frequently an item appears in dataset. The higher support values indicates that the item appeared more often in the dataset. For example, if the support value is 0.5, it indicates that the item appeared in 50% of the transactions record. Then, the equation (3.2), confidence is used to measure the strength of an association between items in the dataset. It evaluates how often item B is purchased when item A is also purchased. Next, the equation (3.3), lift is used to measure the strength of the association by showing how likely items A and B are purchased together.

3.4.2 FP-Growth Algorithm

The FP-Growth algorithm is tested an alternative approach to Apriori due to its higher efficiency especially in handling larger datasets. Unlike Apriori, FP-Growth works by constructing a Frequent Pattern Tree (FP-tree) and do not need the candidate generation. This approach compresses the data and generates frequent itemsets more quickly by searching the FP-tree directly. Then, FP-Growth algorithm also applied to generate association rules from frequent itemsets.

3.4.3 Comparison between Apriori and FP-Growth Algorithms

Both Apriori and FP-Growth algorithms are evaluated based on the execution time and memory usage. After compared both algorithms, the algorithm with better performance will be chosen for further analysis and product recommendation generation.

3.5 Customer Segmentation and Recommendation

3.5.1 Customer Segmentation Using RFM Model

After the prediction for customer future purchases and evaluation of BG/NBD model is completed, the customers were segmented using the RFM model which are Recency, Frequency, and Monetary. The RFM model evaluates customer behavior and categorizes the customer based on their past purchases where it allows for targeted marketing and product recommendations. In RFM, the term recency refers to how recently a customer made a purchase, whereas frequency refers to how frequently the customer made a purchase within the period and monetary values is the total amount of a customer has spent within the period. With the RFM segmentation, the customers were divided into four different groups which are “Champions”, “Loyal customers”, “At risk” and “Hibernating”. Customers who are "Champions" are those who have purchased recently, frequently, and spend the most. The term "Loyal customers" refers to customers who purchase regularly and spend significant but not as frequently or recently as “Champions”. Customers who are "At risk" are those who make frequent and valuable purchases but have not made any purchases or not bought anything recently. Finally, "Hibernating" are those customers who have not purchased for a long time or have made only a few purchases and are at risk of being lost.

3.5.2 Customer Recommendation

Once the customers were segmented using RFM model, the FP-Growth algorithm was applied separately to each customer group to discover the frequent itemsets that each customer group tend to buy together. With the help of frequent itemsets, the purchasing patterns of each customer group can be discovered and allowed for precise product recommendations. Based on the frequent itemsets generated by the FP-Growth algorithm, the top 10 products recommendations were made for each customer group. The products were ranked by their support values to ensure that the most popular and relevant products were suggested to each customer group.

3.6 Tools to Use

3.6.1 Hardware

The hardware involved in this project is the laptop and it is used to code and develop the project.

Table 3.1 Specifications of laptop

Description	Specifications
Model	Asus TUF Gaming A15 FA506IH_FA506IH
Processor	AMD Ryzen 5 4600H
Operating System	Windows 11 Home
Graphic	NVIDIA GeForce GTX 1650
Memory	16GB DDR4 RAM
Storage	512 SSD

3.6.2 Software Setup

In this project, python programming language and Jupyter Notebook will be used throughout this study. Jupyter Notebook is the free and open web application which for data scientists to perform the data science tasks. It also can be used to create and share the documents containing live code and equations.

Table 3.2 Software involved

Software	Descriptions
Programming Language	Python programming language
Software	Jupyter Notebook (Anaconda)

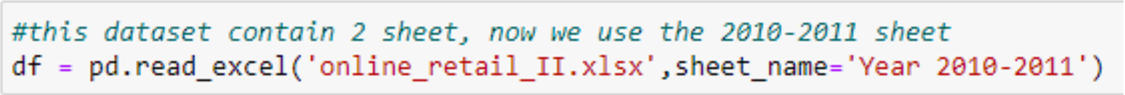
Chapter 4

Implementation

4.1 Data Processing

4.1.1 Dataset Used

The sheet “Year 2010-2011” in Online Retail II dataset is used and it includes total of 541910 instances. Each instance includes 8 variables which are “Invoice”, “StockCode”, “Description”, “Quantity”, “InvoiceDate”, “Price”, “Customer ID” and “Country”. The figure below shows the code for getting data from the “Year 2010-2011” sheet in Online Retail II dataset.

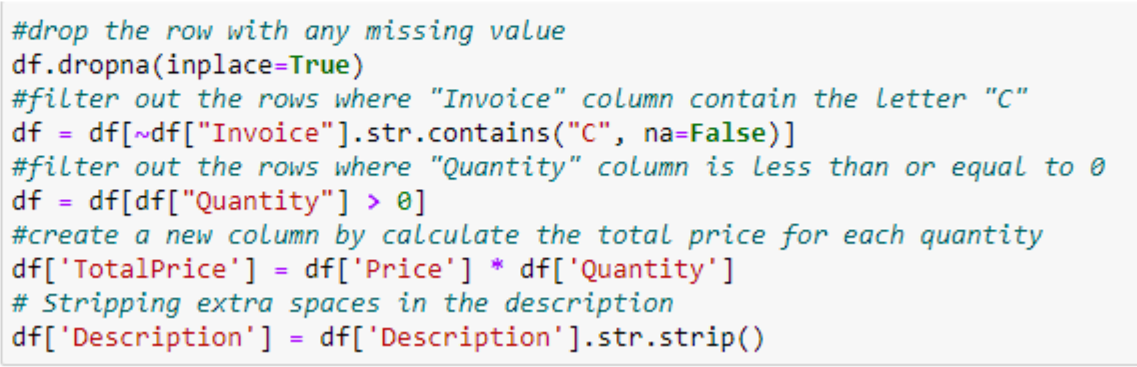


```
#this dataset contain 2 sheet, now we use the 2010-2011 sheet
df = pd.read_excel('online_retail_II.xlsx',sheet_name='Year 2010-2011')
```

Figure 4.1.1.1: Code for getting data from the “Year 2010-2011” sheet in Online Retail II dataset.

4.1.2 Data Preprocessing

Before performing any analysis, the data needs to be preprocessed to ensure it is clean and consistent. First, the row with any missing value among any column is removed. Then, the data is filtered out rows where the Invoice column that start with letter “C” as it indicates that it is a cancelled order. The rows where the Quantity column is less than or equal to 0 also removed as it is not valid for purchases. After that, a new column named TotalPrice is created that represents the total price for each item purchased. Lastly, the extra spaces in the “Description” column are removed to ensure consistency when it is used for comparison and avoid any duplicate records caused by inconsistent spacing. The figure below shows the code for data cleaning.



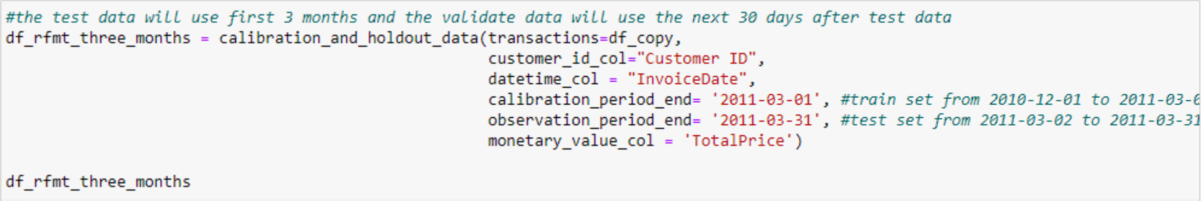
```
#drop the row with any missing value
df.dropna(inplace=True)
#filter out the rows where "Invoice" column contain the letter "C"
df = df[~df["Invoice"].str.contains("C", na=False)]
#filter out the rows where "Quantity" column is less than or equal to 0
df = df[df["Quantity"] > 0]
#create a new column by calculate the total price for each quantity
df['TotalPrice'] = df['Price'] * df['Quantity']
# Stripping extra spaces in the description
df['Description'] = df['Description'].str.strip()
```

Figure 4.1.2.1: Code for data cleaning.

4.2 Customer Prediction

4.2.1 Split into Training and Testing Set

After cleaning the data, it is divided into two subsets which are calibration set (training set) and holdout set (testing set) by applying the "calibration_and_holdout_data" function from lifetimes package. The calibration-holdout split is similar to train-test split that commonly used in machine learning. In this process, the calibration set servers as the data for model training while the holdout set is reserved for evaluating the model's performance. The split is based on the "calibration_period_end" parameter where it determines the end date for the calibration period, and it means that the calibration set includes the data up to this chosen end date. On the other hand, the holdout period includes data after the end of calibration period to another specified end date that determined by the "observation_period_end" parameter. In short, this function splits the data into calibration and holdout sets based on the provided time period and generates summary statistics for both periods. The summary statistics include the frequency (the number of repeat purchases), recency (the duration between customer's first purchase and the latest purchase), T (the duration between first purchase of customer and the end of observation period), and monetary value for both calibration and holdout periods. The summary statistics for both periods are then combined and get the duration of the holdout period. Figure below shows the code for splitting.

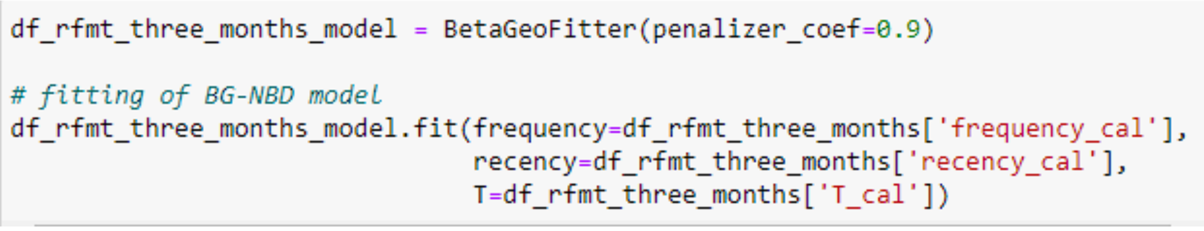


```
#the test data will use first 3 months and the validate data will use the next 30 days after test data
df_rfmt_three_months = calibration_and_holdout_data(transactions=df_copy,
                                                    customer_id_col="Customer ID",
                                                    datetime_col = "InvoiceDate",
                                                    calibration_period_end= '2011-03-01', #train set from 2010-12-01 to 2011-03-0
                                                    observation_period_end= '2011-03-31', #test set from 2011-03-02 to 2011-03-31
                                                    monetary_value_col = 'TotalPrice')

df_rfmt_three_months
```

Figure 4.2.1.1: Code for splitting the data into calibration set and holdout set.

4.2.2 BG/NBD Model

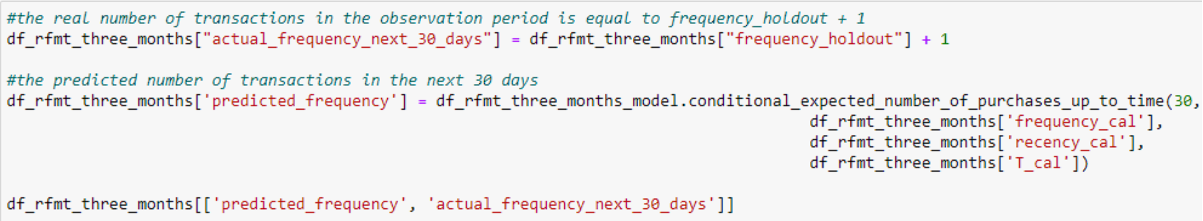
The model is built using the BetaGeoFitter function with the parameter regularization coefficient which is used to prevent overfitting. The model is then fitted to historical data on customer transactions which are frequency, recency and age from the data frame that created. The model will discover the underlying pattern of customer behaviour such as the number of repeat purchases the customer has made, the duration between customer's first purchase and the end of observation period, and duration between customer first purchase and their latest purchase.



```
df_rfmt_three_months_model = BetaGeoFitter(penalizer_coef=0.9)

# fitting of BG-NBD model
df_rfmt_three_months_model.fit(frequency=df_rfmt_three_months['frequency_cal'],
                               recency=df_rfmt_three_months['recency_cal'],
                               T=df_rfmt_three_months['T_cal'])
```

Figure 4.2.2.1: Code of creating the BG/NBD model.

Then, the prediction for number of purchases can be made using the "conditional_expected_number_of_purchases_up_to_time()" function from lifetimes package with the trained BG/NBD model. To predict the number of purchases for customer within a certain period of time, there are 4 necessary parameters are required which are time, historical frequency, recency and age of the customer. In short, this function will predict the number of purchases for each customer based on their past transactions.

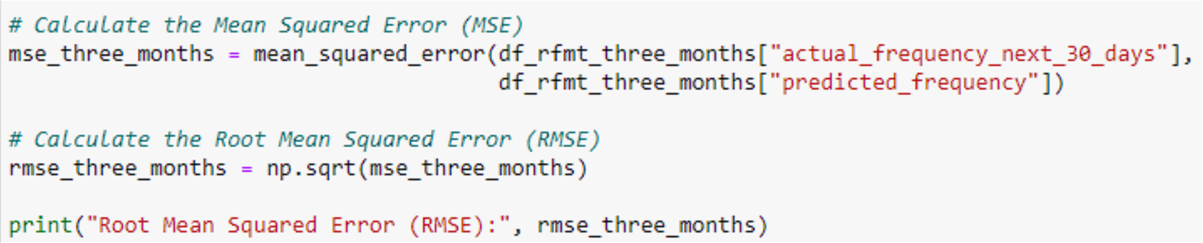


```
#the real number of transactions in the observation period is equal to frequency_holdout + 1
df_rfmt_three_months["actual_frequency_next_30_days"] = df_rfmt_three_months["frequency_holdout"] + 1

#the predicted number of transactions in the next 30 days
df_rfmt_three_months['predicted_frequency'] = df_rfmt_three_months_model.conditional_expected_number_of_purchases_up_to_time(30,
                                                                                    df_rfmt_three_months['frequency_cal'],
                                                                                    df_rfmt_three_months['recency_cal'],
                                                                                    df_rfmt_three_months['T_cal'])

df_rfmt_three_months[['predicted_frequency', 'actual_frequency_next_30_days']]
```

Figure 4.2.2.2: Code of predict the number of purchases.

4.2.3 Model Evaluation

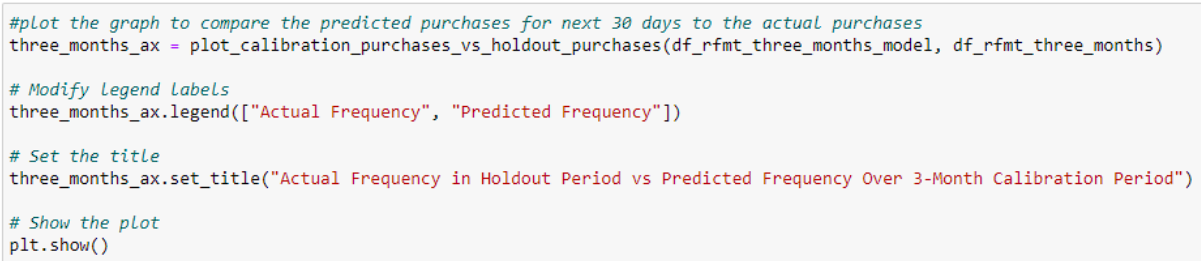
To access the performance of model, the Root Mean Squared Error (RMSE) is used to evaluate the accuracy of prediction. RMSE will provide a number that summarize the overall performance of the model by calculating the average difference between predicted values and actual value.



```
# Calculate the Mean Squared Error (MSE)
mse_three_months = mean_squared_error(df_rfmt_three_months["actual_frequency_next_30_days"],
                                      df_rfmt_three_months["predicted_frequency"])

# Calculate the Root Mean Squared Error (RMSE)
rmse_three_months = np.sqrt(mse_three_months)

print("Root Mean Squared Error (RMSE):", rmse_three_months)
```

Figure 4.2.3.1: Code for Root Mean Squared Error (RMSE) calculation.

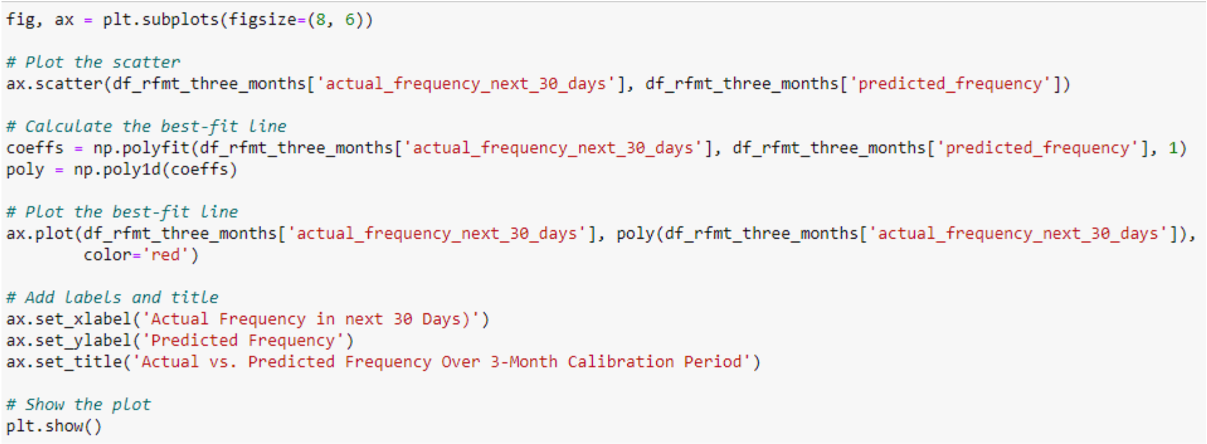
Other than that, the "plot_calibration_purchases_vs_holdout_purchases" function also used to generate a graph to compare the predicted frequency to actual frequency based on the model and the dataset that split. In the graph, it consists of two line which represent actual frequency and predicted frequency where it also illustrates the accuracy of prediction. This is helpful as it provides the visual comparison between actual frequency and predicted frequency.



```
#plot the graph to compare the predicted purchases for next 30 days to the actual purchases
three_months_ax = plot_calibration_purchases_vs_holdout_purchases(df_rfmt_three_months_model, df_rfmt_three_months)

# Modify legend labels
three_months_ax.legend(["Actual Frequency", "Predicted Frequency"])

# Set the title
three_months_ax.set_title("Actual Frequency in Holdout Period vs Predicted Frequency Over 3-Month Calibration Period")

# Show the plot
plt.show()
```

Figure 4.2.3.2: Code to plotting the graph using "plot_calibration_purchases_vs_holdout_purchases" function.

Besides, there are another graph that generated using scatter plot to visualize the relationship between the actual frequency and predicted frequency with a best-fit line. This scatter plot allows for a deeper understanding of the accuracy of prediction. In short, these visualizations provide a better understanding of the performance of the model.



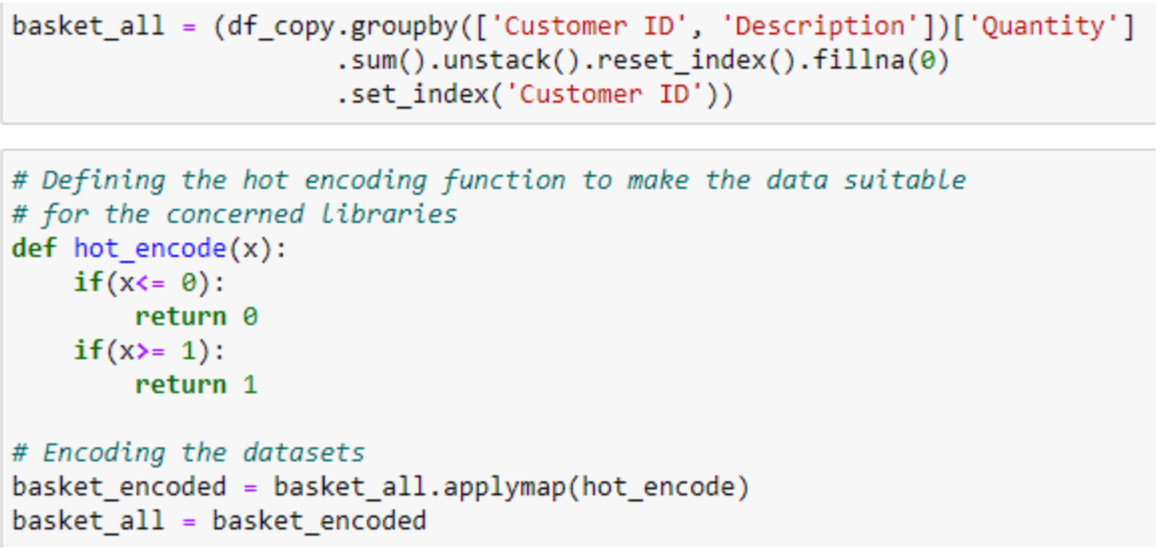
```
fig, ax = plt.subplots(figsize=(8, 6))

# Plot the scatter
ax.scatter(df_rfmt_three_months['actual_frequency_next_30_days'], df_rfmt_three_months['predicted_frequency'])

# Calculate the best-fit line
coeffs = np.polyfit(df_rfmt_three_months['actual_frequency_next_30_days'], df_rfmt_three_months['predicted_frequency'], 1)
poly = np.poly1d(coeffs)

# Plot the best-fit line
ax.plot(df_rfmt_three_months['actual_frequency_next_30_days'], poly(df_rfmt_three_months['actual_frequency_next_30_days']),
        color='red')

# Add labels and title
ax.set_xlabel('Actual Frequency in next 30 Days)')
ax.set_ylabel('Predicted Frequency')
ax.set_title('Actual vs. Predicted Frequency Over 3-Month Calibration Period')

# Show the plot
plt.show()
```

Figure 4.2.3.3: Code to plotting the "Actual vs. Predicted Frequency Over 3-Month Calibration Period" graph using scatter plot.

4.3 Market Basket Analysis (MBA)

4.3.1 Apriori Algorithm

Before executing the Apriori algorithm, the transaction data is aggregated by grouping the two columns “Customer ID” and “Description”, and it sums the “Quantity” column for each group. This is used to create a dataframe to count the number of times each product was purchased by each customer. Then, the hot encoding is applied to convert the purchase quantities into binary values. In this case, the value of 1 indicates that the customer purchased the product while the value of 0 indicates that there is no purchase. This binary encoding is important to apply on the data first because the Apriori algorithm is relying on binary input to identify frequent itemsets.



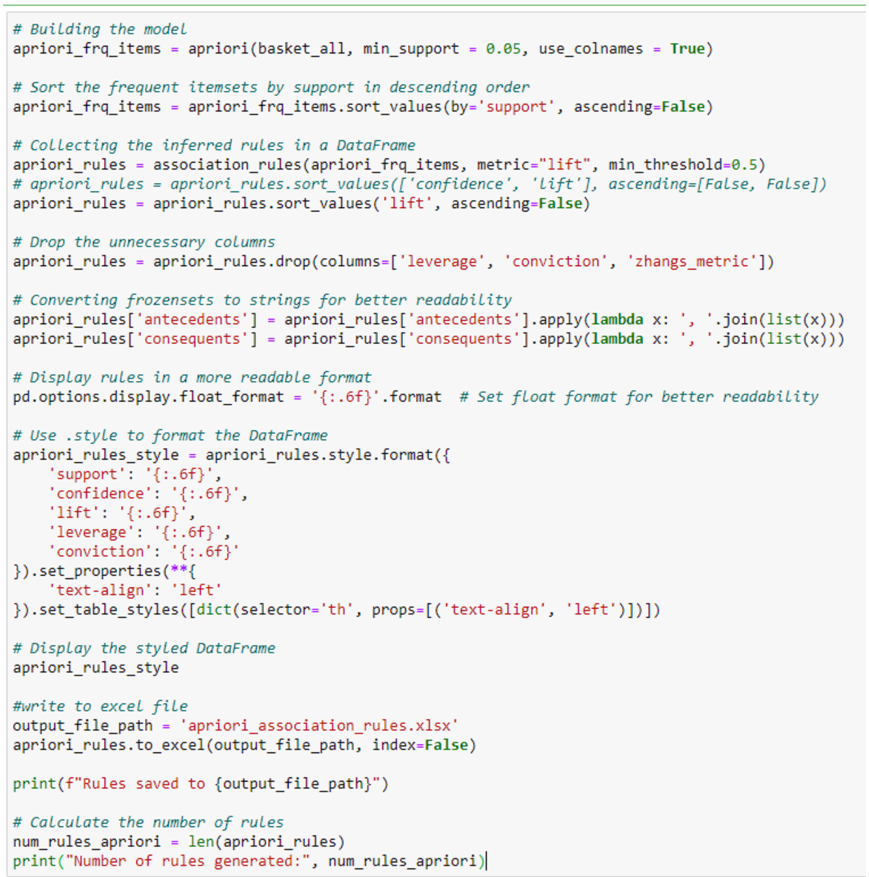
```
basket_all = (df_copy.groupby(['Customer ID', 'Description'])['Quantity']
                     .sum().unstack().reset_index().fillna(0)
                     .set_index('Customer ID'))
```

```
# Defining the hot encoding function to make the data suitable
# for the concerned libraries
def hot_encode(x):
    if(x<= 0):
        return 0
    if(x>= 1):
        return 1

# Encoding the datasets
basket_encoded = basket_all.applymap(hot_encode)
basket_all = basket_encoded
```

Figure 4.3.1.1: Code to prepare the data for Apriori and FP-Growth algorithms testing.

After preparing the data, the apriori() function from the mlxtend library is used to create the Apriori model with a minimum support threshold of 0.05. This means that only itemsets that appear in at least 5% of the transactions are considered frequent. The frequent itemsets are then sorted in descending order based on support values to show the most frequent itemsets.

Next, the association rules are derived from the frequent itemsets using the association_rules() function with the "lift" metric and a minimum threshold of 0.5. The rules are sorted in descending order based on lift values to identify the items that often purchased together. To improve the readability of the rules, the rules were formatted by converting the frozensets to strings and any float values are formatted to 6 decimal points. The association rules are then saved to Excel file for further analysis. Finally, the total number of rules generated by Apriori is calculated to check how many association rules are created.

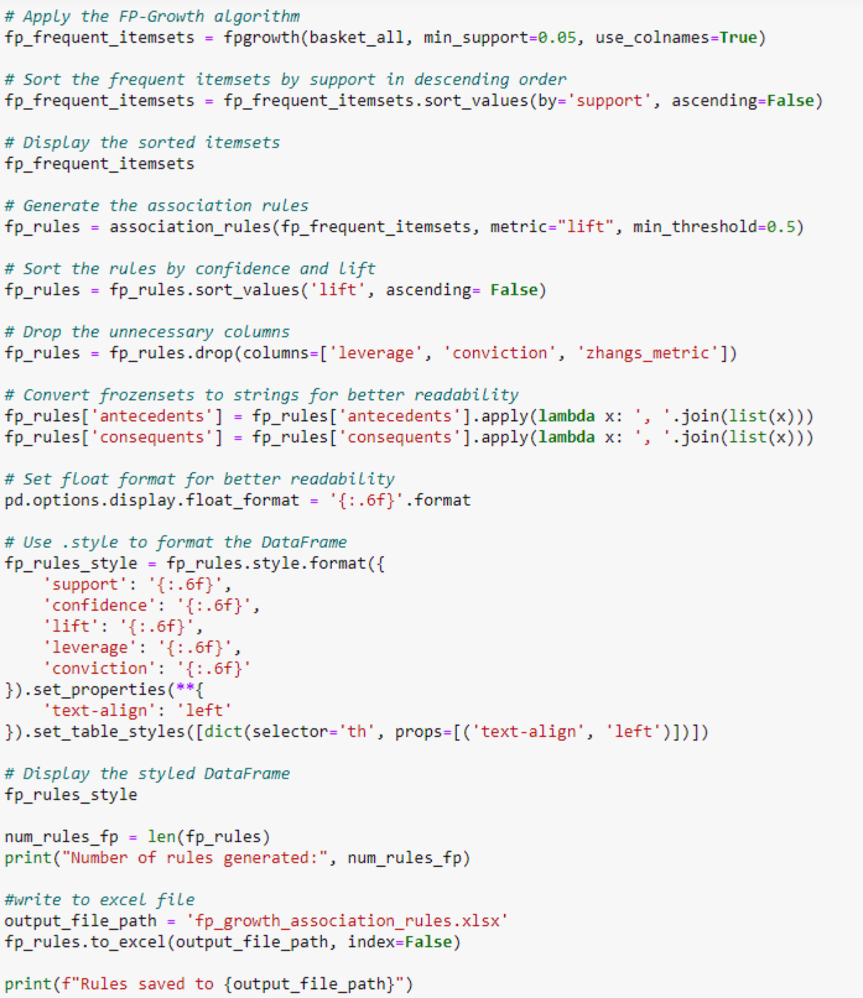


```
# Building the model
apriori_frq_items = apriori(basket_all, min_support = 0.05, use_colnames = True)

# Sort the frequent itemsets by support in descending order
apriori_frq_items = apriori_frq_items.sort_values(by='support', ascending=False)

# Collecting the inferred rules in a DataFrame
apriori_rules = association_rules(apriori_frq_items, metric="lift", min_threshold=0.5)
# apriori_rules = apriori_rules.sort_values(['confidence', 'lift'], ascending=[False, False])
apriori_rules = apriori_rules.sort_values('lift', ascending=False)

# Drop the unnecessary columns
apriori_rules = apriori_rules.drop(columns=['leverage', 'conviction', 'zhangs_metric'])

# Converting frozensets to strings for better readability
apriori_rules['antecedents'] = apriori_rules['antecedents'].apply(lambda x: ', '.join(list(x)))
apriori_rules['consequents'] = apriori_rules['consequents'].apply(lambda x: ', '.join(list(x)))

# Display rules in a more readable format
pd.options.display.float_format = '{:.6f}'.format  # Set float format for better readability

# Use .style to format the DataFrame
apriori_rules_style = apriori_rules.style.format({
    'support': '{:.6f}',
    'confidence': '{:.6f}',
    'lift': '{:.6f}',
    'leverage': '{:.6f}',
    'conviction': '{:.6f}'
}).set_properties(**{
    'text-align': 'left'
}).set_table_styles([dict(selector='th', props=[('text-align', 'left')])])

# Display the styled DataFrame
apriori_rules_style

#write to excel file
output_file_path = 'apriori_association_rules.xlsx'
apriori_rules.to_excel(output_file_path, index=False)

print(f"Rules saved to {output_file_path}")

# Calculate the number of rules
num_rules_apriori = len(apriori_rules)
print("Number of rules generated:", num_rules_apriori)
```

Figure 4.3.1.2: Code to generate association rules for Apriori algorithm.

4.3.2 FP-Growth Algorithm

Similar to the Apriori algorithm, the FP-Growth algorithm is used the same binary-encoded data to generate the frequent itemsets using the fp_growth() function from the mlxtend library to create the FP-Growth model with a minimum support threshold of 0.05. The frequent itemsets are then sorted in descending order based on support values to show the most frequent itemsets. Then, the association rules are generated from the frequent itemsets using the association_rules() function with the "lift" metric and a minimum threshold of 0.5. The rules are sorted in descending order based on lift values to identify the items that often purchased together. To improve the readability of the rules, the rules were formatted by converting the frozensets to strings and any float values are formatted to 6 decimal points. The association rules are then saved to Excel file for further analysis. Finally, the total number of rules generated by FP-Growth is calculated to check how many association rules are created.

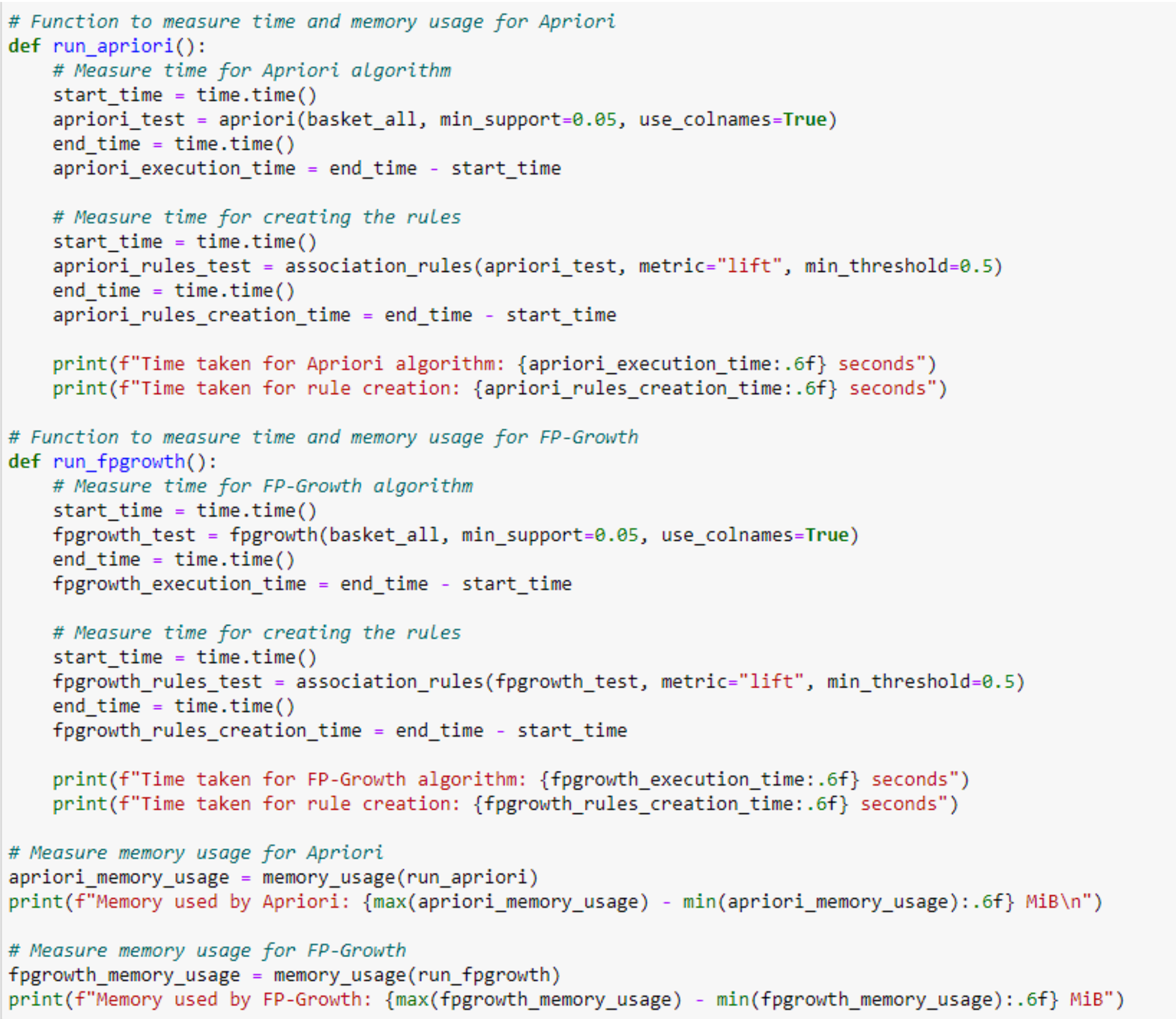


```
# Apply the FP-Growth algorithm
fp_frequent_itemsets = fpgrowth(basket_all, min_support=0.05, use_colnames=True)

# Sort the frequent itemsets by support in descending order
fp_frequent_itemsets = fp_frequent_itemsets.sort_values(by='support', ascending=False)

# Display the sorted itemsets
fp_frequent_itemsets

# Generate the association rules
fp_rules = association_rules(fp_frequent_itemsets, metric="lift", min_threshold=0.5)

# Sort the rules by confidence and lift
fp_rules = fp_rules.sort_values('lift', ascending= False)

# Drop the unnecessary columns
fp_rules = fp_rules.drop(columns=['leverage', 'conviction', 'zhangs_metric'])

# Convert frozensets to strings for better readability
fp_rules['antecedents'] = fp_rules['antecedents'].apply(lambda x: ', '.join(list(x)))
fp_rules['consequents'] = fp_rules['consequents'].apply(lambda x: ', '.join(list(x)))

# Set float format for better readability
pd.options.display.float_format = '{:.6f}'.format

# Use .style to format the DataFrame
fp_rules_style = fp_rules.style.format({
    'support': '{:.6f}',
    'confidence': '{:.6f}',
    'lift': '{:.6f}',
    'leverage': '{:.6f}',
    'conviction': '{:.6f}'
}).set_properties(**{
    'text-align': 'left'
}).set_table_styles([dict(selector='th', props=[('text-align', 'left')])])

# Display the styled DataFrame
fp_rules_style

num_rules_fp = len(fp_rules)
print("Number of rules generated:", num_rules_fp)

#write to excel file
output_file_path = 'fp_growth_association_rules.xlsx'
fp_rules.to_excel(output_file_path, index=False)

print(f"Rules saved to {output_file_path}")
```

Figure 4.3.2.1: Code to generate association rules for FP-Growth algorithm.

4.3.3 Comparison between Apriori and FP-Growth Algorithms

The performance of the Apriori and FP-Growth Algorithms are compared by evaluating the execution times and memory usage with the same data. To do this, the execution time is monitored for both algorithms to measure how long for both algorithms to generate the frequent itemsets. This is done using the time.time() function to record the start and end times. Then, the memory consumption for both algorithms is traked using the memory_usage() function to record the memory used when executing the Apriori and FP-Growth algorithms.



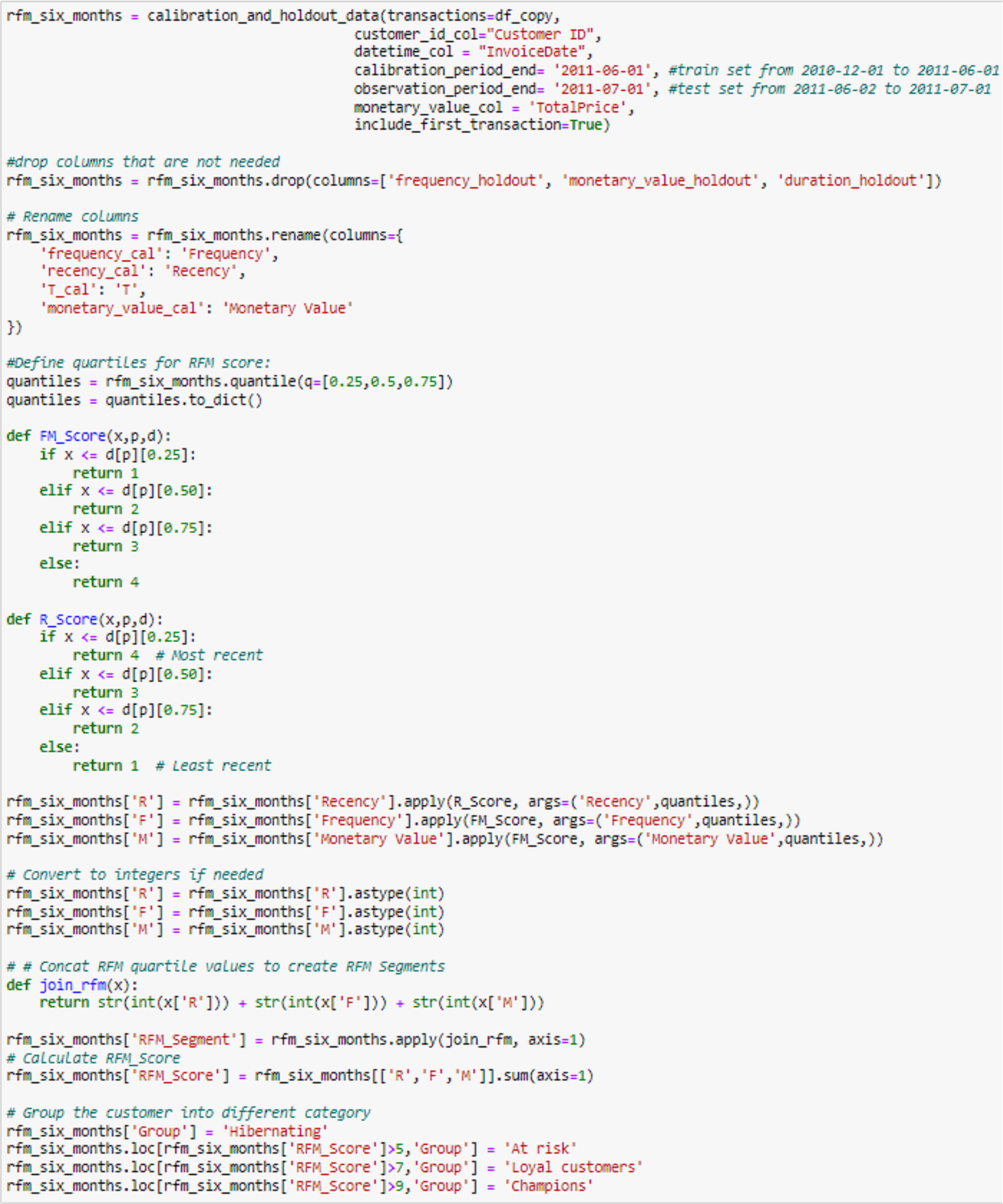
```python
# Function to measure time and memory usage for Apriori
def run_apriori():
    # Measure time for Apriori algorithm
    start_time = time.time()
    apriori_test = apriori(basket_all, min_support=0.05, use_colnames=True)
    end_time = time.time()
    apriori_execution_time = end_time - start_time

    # Measure time for creating the rules
    start_time = time.time()
    apriori_rules_test = association_rules(apriori_test, metric="lift", min_threshold=0.5)
    end_time = time.time()
    apriori_rules_creation_time = end_time - start_time

    print(f"Time taken for Apriori algorithm: {apriori_execution_time:.6f} seconds")
    print(f"Time taken for rule creation: {apriori_rules_creation_time:.6f} seconds")

# Function to measure time and memory usage for FP-Growth
def run_fpgrowth():
    # Measure time for FP-Growth algorithm
    start_time = time.time()
    fpgrowth_test = fpgrowth(basket_all, min_support=0.05, use_colnames=True)
    end_time = time.time()
    fpgrowth_execution_time = end_time - start_time

    # Measure time for creating the rules
    start_time = time.time()
    fpgrowth_rules_test = association_rules(fpgrowth_test, metric="lift", min_threshold=0.5)
    end_time = time.time()
    fpgrowth_rules_creation_time = end_time - start_time

    print(f"Time taken for FP-Growth algorithm: {fpgrowth_execution_time:.6f} seconds")
    print(f"Time taken for rule creation: {fpgrowth_rules_creation_time:.6f} seconds")

# Measure memory usage for Apriori
apriori_memory_usage = memory_usage(run_apriori)
print(f"Memory used by Apriori: {max(apriori_memory_usage) - min(apriori_memory_usage):.6f} MiB\n")

# Measure memory usage for FP-Growth
fpgrowth_memory_usage = memory_usage(run_fpgrowth)
print(f"Memory used by FP-Growth: {max(fpgrowth_memory_usage) - min(fpgrowth_memory_usage):.6f} MiB")
```

Figure 4.3.3.1: Code to evaluate the performance of Apriori and FP-Growth algorithms.

4.4 Customer Segmentation and Recommendation

4.4.1 Customer Segmentation Using RFM Model

The customer segmentation is performed by calculating RFM (Recency, Frequency, Monetary) values and group the customers into different categories such as “Hibernating”, “At risk”, “Loyal customers”, and “Champions”. Before performing the customer segmentation, first it needs to extract the customer transaction data for a specific period. To perform this action, the “calibration_and_holdout_data” function is used to retrieve the transactions from 2010-12-01 to 2011-06-01. After getting the data, the unnecessary columns such as “frequency_holdout”, “monetary_value_holdout”, and “duration_holdout” are dropped as these columns are not required for segmentation and the remaining columns are renamed for better readability. The clean data is then divided into four equal parts using quartiles based on the RFM metrics. Then, two functions are defined which are FM_Score and R_score to calculate the RFM scores by assigning scores to each customer’s Recency, Frequency, and Monetary values based on their position within predefined quartiles. For FM_Score function, this function calculates the score for both Frequency and Monetary values. The score is range from 1 to 4 and higher frequency or monetary values will get higher scores. For example, if a customer ’s frequency falls below the 25th percentile, they are given a score of 1. Then, for R_Score function, it calculates the score for Recency with a reverse logic. Since the recent purchases are considered more valuable, the customer with a lower recency value will receive a higher score. Therefore, the recency value in the lowest quartile is given a score of 4. Once the RFM values are calculated, these three scores are concatenated to form a RFM Segment Code. For example, a customer with scores R = 1, F = 2, and M = 3 will have the segment code “123”. Additionally, the RFM scores are summed to generate a total RFM score which is used to categorize the customer into one of the four groups. For example, a customer with scores R = 1, F = 2, and M = 3 would have a total RFM score of 6, thus the customer will be categorized in “At risk” group.

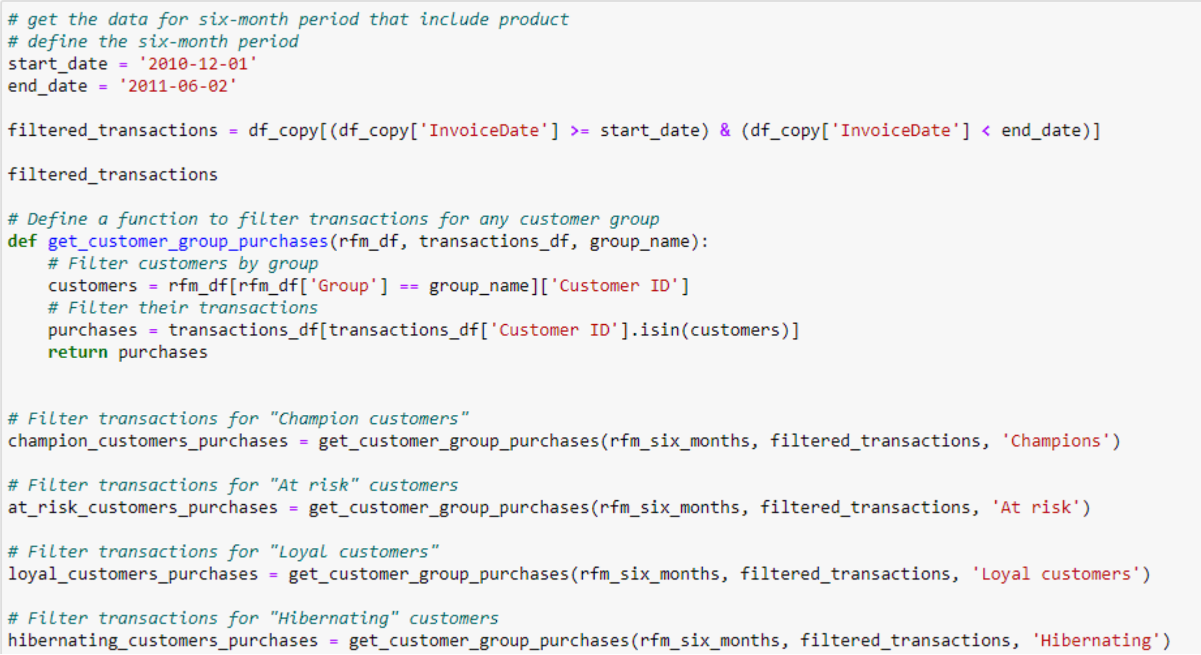


```
rfm_six_months = calibration_and_holdout_data(transactions=df_copy,
                                              customer_id_col="Customer ID",
                                              datetime_col = "InvoiceDate",
                                              calibration_period_end= '2011-06-01', #train set from 2010-12-01 to 2011-06-01
                                              observation_period_end= '2011-07-01', #test set from 2011-06-02 to 2011-07-01
                                              monetary_value_col = 'TotalPrice',
                                              include_first_transaction=True)

#drop columns that are not needed
rfm_six_months = rfm_six_months.drop(columns=['frequency_holdout', 'monetary_value_holdout', 'duration_holdout'])

# Rename columns
rfm_six_months = rfm_six_months.rename(columns={
    'frequency_cal': 'Frequency',
    'recency_cal': 'Recency',
    'T_cal': 'T',
    'monetary_value_cal': 'Monetary Value'
})

#Define quartiles for RFM score:
quantiles = rfm_six_months.quantile(q=[0.25,0.5,0.75])
quantiles = quantiles.to_dict()

def FM_Score(x,p,d):
    if x <= d[p][0.25]:
        return 1
    elif x <= d[p][0.50]:
        return 2
    elif x <= d[p][0.75]:
        return 3
    else:
        return 4

def R_Score(x,p,d):
    if x <= d[p][0.25]:
        return 4  # Most recent
    elif x <= d[p][0.50]:
        return 3
    elif x <= d[p][0.75]:
        return 2
    else:
        return 1  # Least recent

rfm_six_months['R'] = rfm_six_months['Recency'].apply(R_Score, args=('Recency',quantiles,))
rfm_six_months['F'] = rfm_six_months['Frequency'].apply(FM_Score, args=('Frequency',quantiles,))
rfm_six_months['M'] = rfm_six_months['Monetary Value'].apply(FM_Score, args=('Monetary Value',quantiles,))

# Convert to integers if needed
rfm_six_months['R'] = rfm_six_months['R'].astype(int)
rfm_six_months['F'] = rfm_six_months['F'].astype(int)
rfm_six_months['M'] = rfm_six_months['M'].astype(int)

# # Concat RFM quartile values to create RFM Segments
def join_rfm(x):
    return str(int(x['R'])) + str(int(x['F'])) + str(int(x['M']))

rfm_six_months['RFM_Segment'] = rfm_six_months.apply(join_rfm, axis=1)
# Calculate RFM_Score
rfm_six_months['RFM_Score'] = rfm_six_months[['R','F','M']].sum(axis=1)

# Group the customer into different category
rfm_six_months['Group'] = 'Hibernating'
rfm_six_months.loc[rfm_six_months['RFM_Score']>5,'Group'] = 'At risk'
rfm_six_months.loc[rfm_six_months['RFM_Score']>7,'Group'] = 'Loyal customers'
rfm_six_months.loc[rfm_six_months['RFM_Score']>9,'Group'] = 'Champions'
```

Figure 4.4.1.1: Code for customer segmentation.

4.4.2 Customer Recommendation

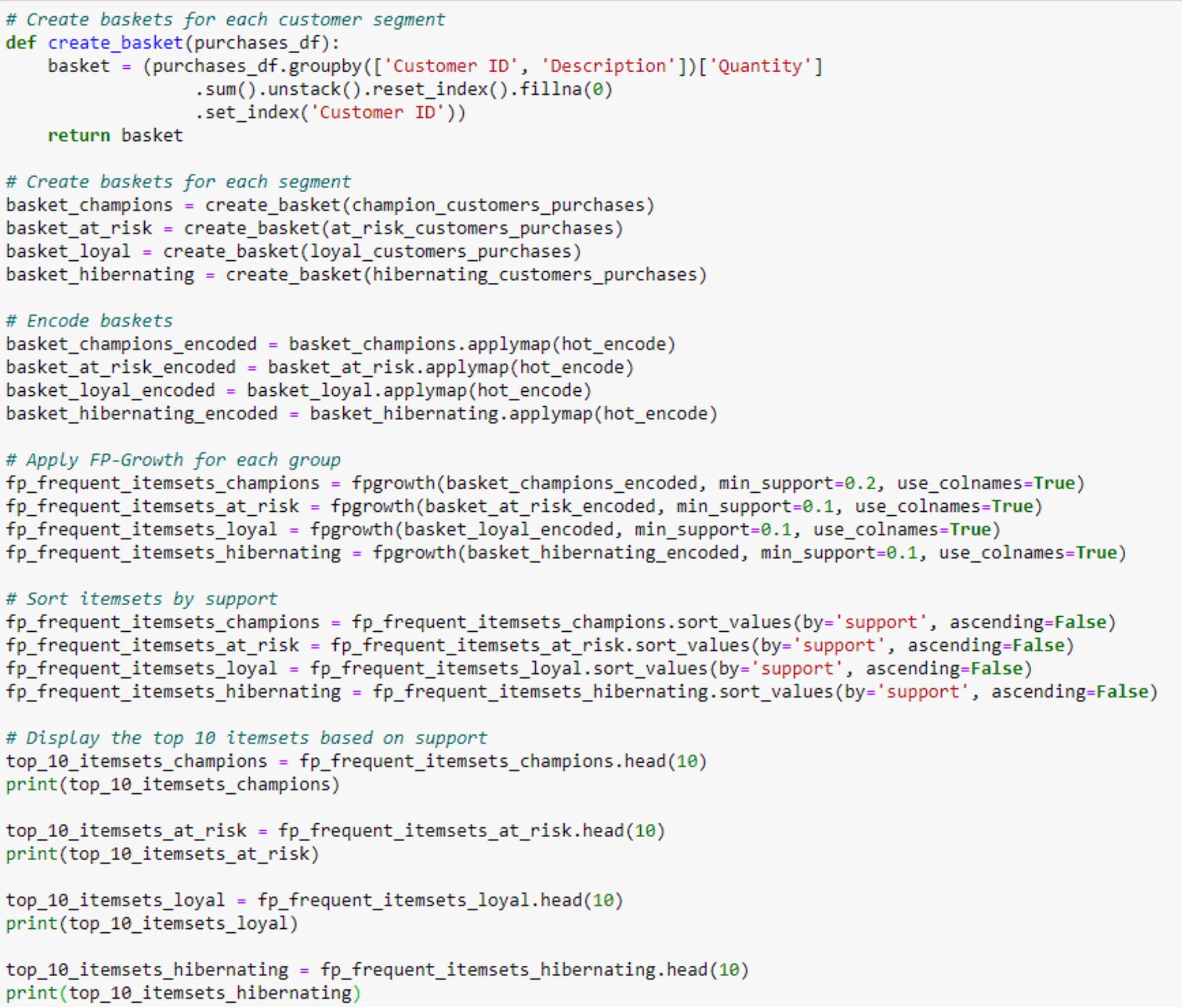
In this section, the FP-Growth algorithm is applied to each customer groups to generate product recommendations, FP-Growth is chosen because it efficiently identifies frequent itemsets in large datasets without the need to generate candidate sets. Before applying the FP-Growth algorithm, it is necessary to filter the transactions data that have the same period with the data used for RFM segmentation to ensures that the product recommendations are based on the same period as the RFM analysis. Then, the “get_customer_group_purchases” function is defined to retrieve the transactions for customers based on their group. The function is then applied to each customer group.



```
# get the data for six-month period that include product
# define the six-month period
start_date = '2010-12-01'
end_date = '2011-06-02'

filtered_transactions = df_copy[(df_copy['InvoiceDate'] >= start_date) & (df_copy['InvoiceDate'] < end_date)]

filtered_transactions

# Define a function to filter transactions for any customer group
def get_customer_group_purchases(rfm_df, transactions_df, group_name):
    # Filter customers by group
    customers = rfm_df[rfm_df['Group'] == group_name]['Customer ID']
    # Filter their transactions
    purchases = transactions_df[transactions_df['Customer ID'].isin(customers)]
    return purchases


# Filter transactions for "Champion customers"
champion_customers_purchases = get_customer_group_purchases(rfm_six_months, filtered_transactions, 'Champions')

# Filter transactions for "At risk" customers
at_risk_customers_purchases = get_customer_group_purchases(rfm_six_months, filtered_transactions, 'At risk')

# Filter transactions for "Loyal customers"
loyal_customers_purchases = get_customer_group_purchases(rfm_six_months, filtered_transactions, 'Loyal customers')

# Filter transactions for "Hibernating" customers
hibernating_customers_purchases = get_customer_group_purchases(rfm_six_months, filtered_transactions, 'Hibernating')
```

Figure 4.4.2.1: Code for filtering transaction data and isolated the purchases for each customer group.

Next, the "create_basket" function was used to aggregate each customer group's purchases where each basket reprensents the total quantities of products purchased by customers in each group. The basket data is then encoded using hot encoding to convert the quantities into binary value which is required before continuing with the FP-Growth algorithm. After the baskets are encoded, the FP-Growth algorithm is applied to each customer group’s encoded basket to identify frequent itemsets. The itemsets are sorted based on support values to show the most frequent purchases items. Lastly, the top 10 frequent itemsets were extracted and shown to create product recommendations for each group.



```
# Create baskets for each customer segment
def create_basket(purchases_df):
    basket = (purchases_df.groupby(['Customer ID', 'Description'])['Quantity']
              .sum().unstack().reset_index().fillna(0)
              .set_index('Customer ID'))
    return basket

# Create baskets for each segment
basket_champions = create_basket(champion_customers_purchases)
basket_at_risk = create_basket(at_risk_customers_purchases)
basket_loyal = create_basket(loyal_customers_purchases)
basket_hibernating = create_basket(hibernating_customers_purchases)

# Encode baskets
basket_champions_encoded = basket_champions.applymap(hot_encode)
basket_at_risk_encoded = basket_at_risk.applymap(hot_encode)
basket_loyal_encoded = basket_loyal.applymap(hot_encode)
basket_hibernating_encoded = basket_hibernating.applymap(hot_encode)

# Apply FP-Growth for each group
fp_frequent_itemsets_champions = fpgrowth(basket_champions_encoded, min_support=0.2, use_colnames=True)
fp_frequent_itemsets_at_risk = fpgrowth(basket_at_risk_encoded, min_support=0.1, use_colnames=True)
fp_frequent_itemsets_loyal = fpgrowth(basket_loyal_encoded, min_support=0.1, use_colnames=True)
fp_frequent_itemsets_hibernating = fpgrowth(basket_hibernating_encoded, min_support=0.1, use_colnames=True)

# Sort itemsets by support
fp_frequent_itemsets_champions = fp_frequent_itemsets_champions.sort_values(by='support', ascending=False)
fp_frequent_itemsets_at_risk = fp_frequent_itemsets_at_risk.sort_values(by='support', ascending=False)
fp_frequent_itemsets_loyal = fp_frequent_itemsets_loyal.sort_values(by='support', ascending=False)
fp_frequent_itemsets_hibernating = fp_frequent_itemsets_hibernating.sort_values(by='support', ascending=False)

# Display the top 10 itemsets based on support
top_10_itemsets_champions = fp_frequent_itemsets_champions.head(10)
print(top_10_itemsets_champions)

top_10_itemsets_at_risk = fp_frequent_itemsets_at_risk.head(10)
print(top_10_itemsets_at_risk)

top_10_itemsets_loyal = fp_frequent_itemsets_loyal.head(10)
print(top_10_itemsets_loyal)

top_10_itemsets_hibernating = fp_frequent_itemsets_hibernating.head(10)
print(top_10_itemsets_hibernating)
```

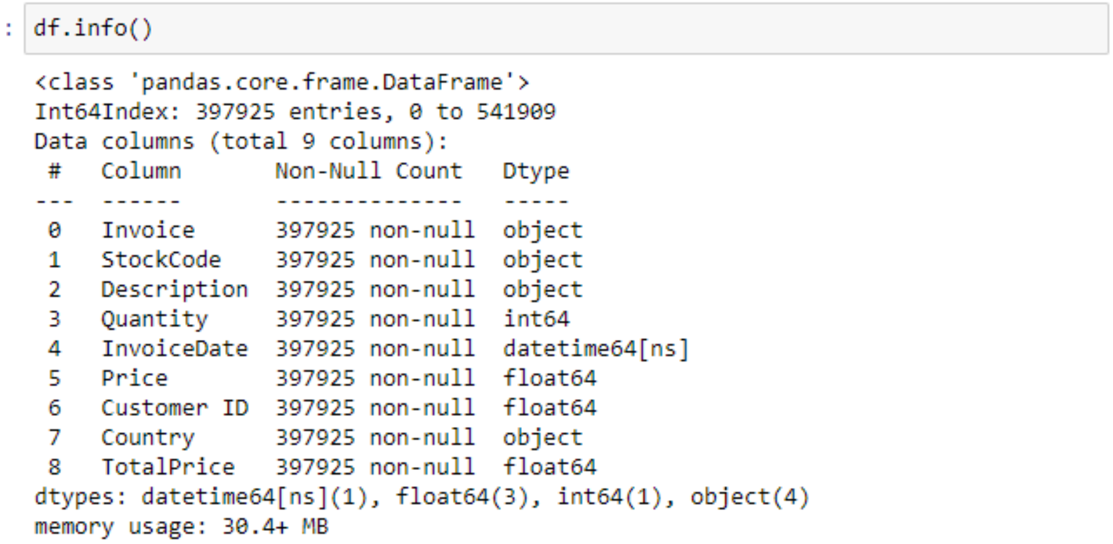
Figure 4.4.2.2: Code for applying FP-Growth to Each Customer Group.

Chapter 5

Model Evaluation and Discussion

5.1 Processed Data

After data preprocessing, the data have been cleaned up by removing the missing value and any invalid data. As a result, the rows with any missing value in the “Description” and “Customer ID” columns were removed. Furthermore, the transactions that started with the letter “C” where represents the cancelled order in the “Invoice” column were removed, along with the rows where the “Quantity” column are less than or equal to zero to ensure that only valid purchases remained. Then, a new column named “TotalPrice” is created to calculate the total price for each purchased item. Finally, any extra spaces in the 'Description' column were stripped to maintain data consistency. Thus, after processed the data, the cleaned data now contains 397925 records for further analysis.



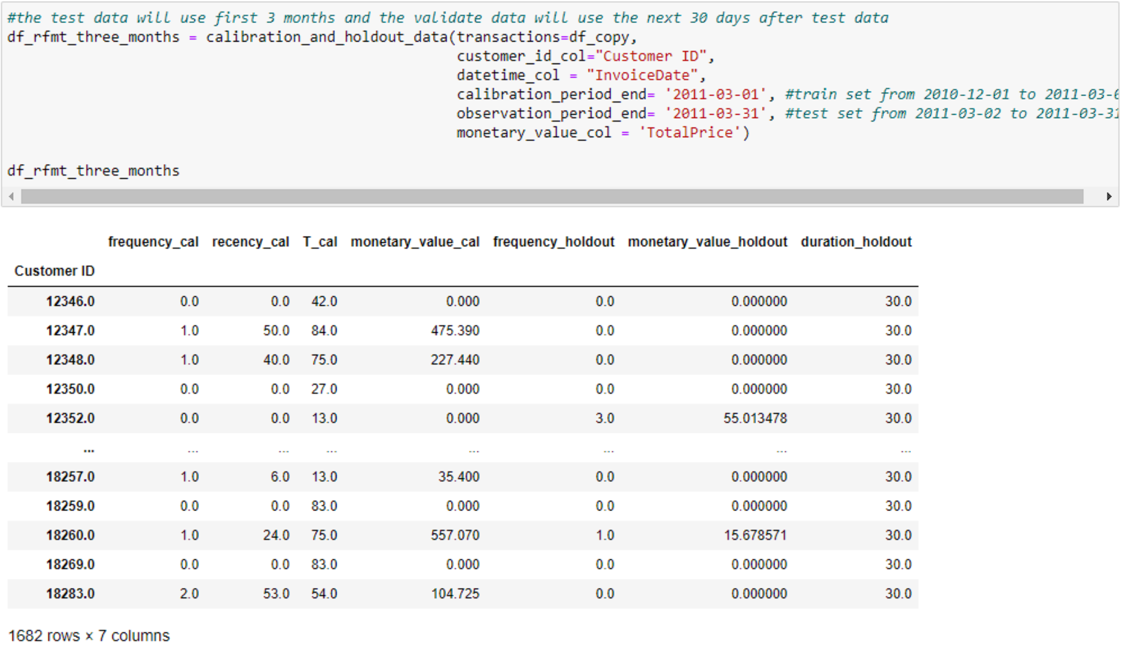
```
: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 397925 entries, 0 to 541909
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      397925 non-null  object
 1   StockCode    397925 non-null  object
 2   Description  397925 non-null  object
 3   Quantity     397925 non-null  int64
 4   InvoiceDate  397925 non-null  datetime64[ns]
 5   Price        397925 non-null  float64
 6   Customer ID  397925 non-null  float64
 7   Country      397925 non-null  object
 8   TotalPrice   397925 non-null  float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 30.4+ MB
```

Figure 5.1.1: Result after performing data cleaning.

5.2 Customer Prediction Model Performance Analysis

5.2.1 Data Splitting for BG/NBD Model

The cleaned data is split into calibration and holdout set where the calibration period is set to first 3 months and holdout period is set to the next 30 days after calibration period with the "calibration_and_holdout_data" function. To use this function, several parameters must be defined. The "transactions" parameter refers to the data frame that contain the transaction data and it is denoted as "df_copy" in this example. The "customer_id_col" parameter specifies the column in the data frame that contain customer identification number. Similarly, the "datetime_col" parameter identifies the column that contains transaction timestamps. Besides, setting the "calibration_period_end" parameter defines the end date of the calibration period, which is necessary for model training while the "observation_period_end" parameter specifies the end date of observation period. From figure 5.2.1.1, the calibration period is set from 1st December 2010 until 1st March 2011 while the holdout period is set from 2nd March 2011 until 31 March 2011. Lastly, the "monetary_value_col" parameter is optional where it used to show the total amount that paid by customer. Figure below shows the output of split data.



```
#the test data will use first 3 months and the validate data will use the next 30 days after test data
df_rfmt_three_months = calibration_and_holdout_data(transactions=df_copy,
                                                    customer_id_col="Customer ID",
                                                    datetime_col = "InvoiceDate",
                                                    calibration_period_end= '2011-03-01', #train set from 2010-12-01 to 2011-03-
                                                    observation_period_end= '2011-03-31', #test set from 2011-03-02 to 2011-03-3
                                                    monetary_value_col = 'TotalPrice')

df_rfmt_three_months
```

	frequency_cal	recency_cal	T_cal	monetary_value_cal	frequency_holdout	monetary_value_holdout	duration_holdout
Customer ID							
12346.0	0.0	0.0	42.0	0.000	0.0	0.000000	30.0
12347.0	1.0	50.0	84.0	475.390	0.0	0.000000	30.0
12348.0	1.0	40.0	75.0	227.440	0.0	0.000000	30.0
12350.0	0.0	0.0	27.0	0.000	0.0	0.000000	30.0
12352.0	0.0	0.0	13.0	0.000	3.0	55.013478	30.0
...	...	...	...	...	...	...	...
18257.0	1.0	6.0	13.0	35.400	0.0	0.000000	30.0
18259.0	0.0	0.0	83.0	0.000	0.0	0.000000	30.0
18260.0	1.0	24.0	75.0	557.070	1.0	15.678571	30.0
18269.0	0.0	0.0	83.0	0.000	0.0	0.000000	30.0
18283.0	2.0	53.0	54.0	104.725	0.0	0.000000	30.0

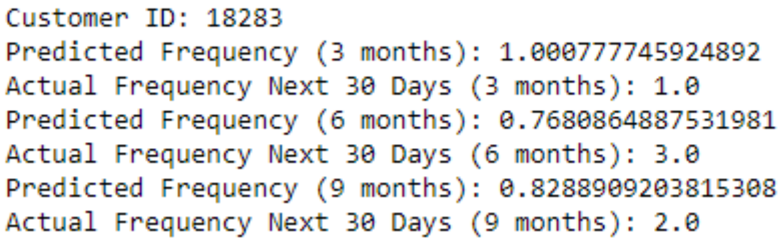
1682 rows × 7 columns

Figure 5.2.1.1: Output of split data based on calibration period and observation period.

The data was split into calibration and holdout periods and returned a combined data frame after processing. The combined data frame includes the frequency (the number of repeat purchases), recency (the duration between customer's first purchase and the latest purchase), T (the duration between first purchase of customer and the end of observation period), and monetary value for both calibration and holdout periods. Consider customer ID "18260" as an example. The customer made total of 2 purchases during the calibration period but the first purchase was excluded, thus there was only one repeat purchase. The time span between their first and last purchases during this period was 24 days, while the time between their first purchase and the end of the observation period was 75 days. Throughout the calibration period, the average value of his transactions is 557.07. Besides, the customer made a total of 2 purchases and resulting in 1 repeat purchase during the holdout period and the average transaction value during this holdout period was 15.68. Lastly, the "duration_holdout" columns show the duration of the holdout period which is 30 days based on the end date that had been set.

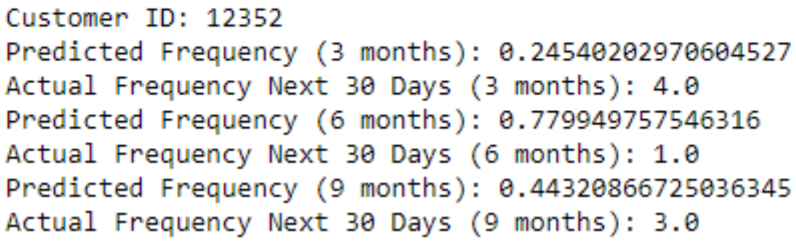
5.2.2 BG/NBD Model Evaluation and Results

The BG/NBD model is trained using calibration data from three different periods to evaluate the prediction accuracy for the number of purchases for each customer in the next 30 days across different calibration periods which are 3 months, 6 months and 9 months During the 3-month calibration period, the model was trained on data from 1 December 2010 until 1 March 2011. Then, the training data for the 6-month calibration period was from 1 December 2010 until 1 June 2011. Finally, the 9-month calibration period included training data from 1 December 2010 to 1 September 2011. The model is fitted with frequency, recency and T parameters that obtained from each calibration period.



```
Customer ID: 18283
Predicted Frequency (3 months): 1.000777745924892
Actual Frequency Next 30 Days (3 months): 1.0
Predicted Frequency (6 months): 0.7680864887531981
Actual Frequency Next 30 Days (6 months): 3.0
Predicted Frequency (9 months): 0.8288909203815308
Actual Frequency Next 30 Days (9 months): 2.0
```

Figure 5.2.2.1: Actual data and predicted data over 3, 6, 9 months for Customer ID 18283.



```
Customer ID: 12352
Predicted Frequency (3 months): 0.24540202970604527
Actual Frequency Next 30 Days (3 months): 4.0
Predicted Frequency (6 months): 0.779949757546316
Actual Frequency Next 30 Days (6 months): 1.0
Predicted Frequency (9 months): 0.44320866725036345
Actual Frequency Next 30 Days (9 months): 3.0
```

Figure 5.2.2.2: Actual data and predicted data over 3, 6, 9 months for Customer ID 12352.

Figure 5.2.2.1 and Figure 5.2.2.2 show the predicted frequency which is the number of purchases that predicted by the model for a customer during the holdout period and the actual frequency which is the number of purchases that observed during the holdout period for two customers across 3, 6, and 9 months of calibration data. For example, the Figure 5.2.2.1 shows that for Customer ID 18283, the prediction of BG/NBD for the 3-month calibration period is closely match the actual frequency. However, the model predicted fewer purchases than the actual frequencies of 3 and 2 for the 6-month and 9-month periods. Then, for Customer ID 12352 in Figure 5.2.2.2, the model underestimated the frequency for 3-month and 9-month periods but nearly matched the actual frequency in the 6-months period.

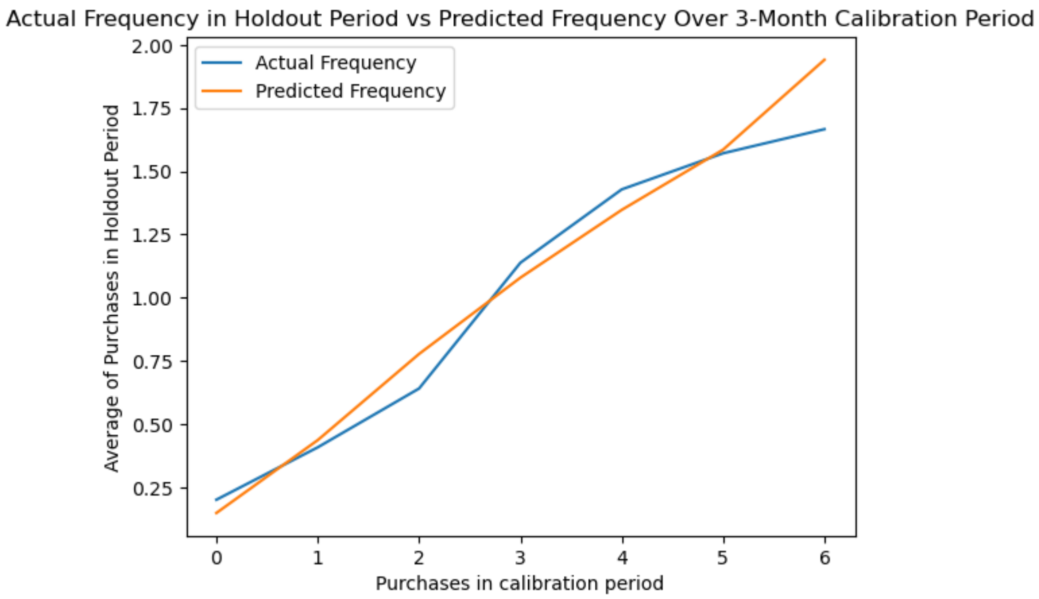


Figure 5.2.2.3: Actual Frequency in Holdout Period vs Predicted Frequency Over 3-Month Calibration Period.

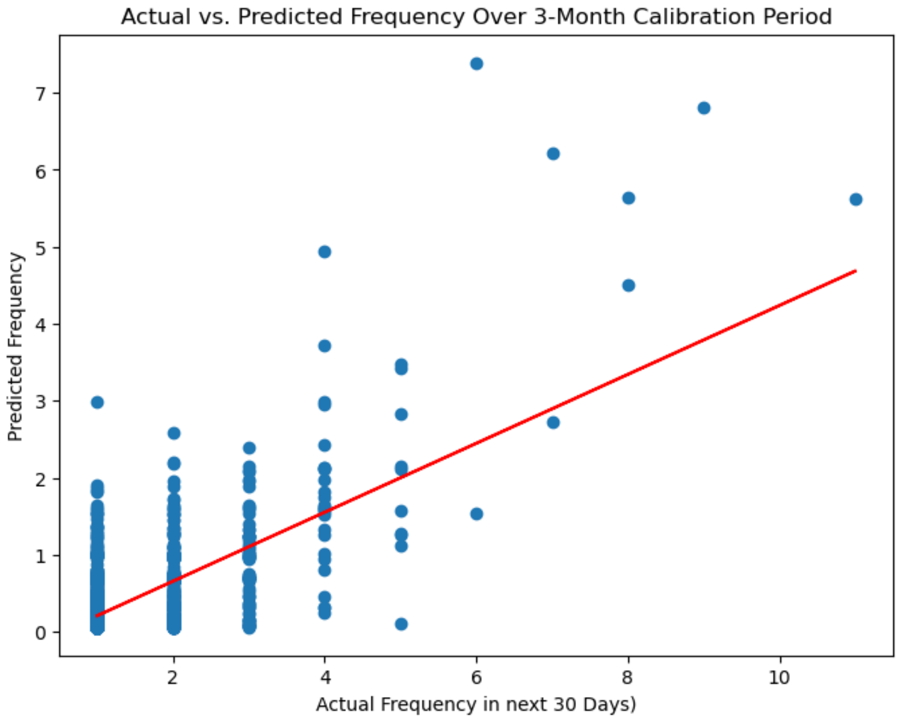


Figure 5.2.2.4: Actual vs. Predicted Frequency Over 3-Month Calibration Period.

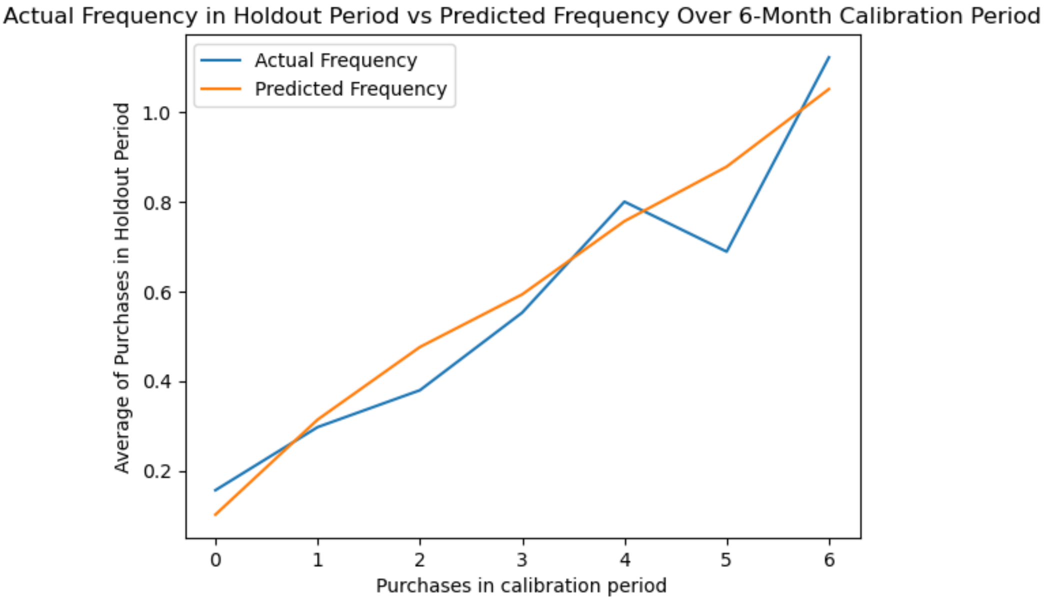


Figure 5.2.2.5: Actual Frequency in Holdout Period vs Predicted Frequency Over 6-Month Calibration Period.

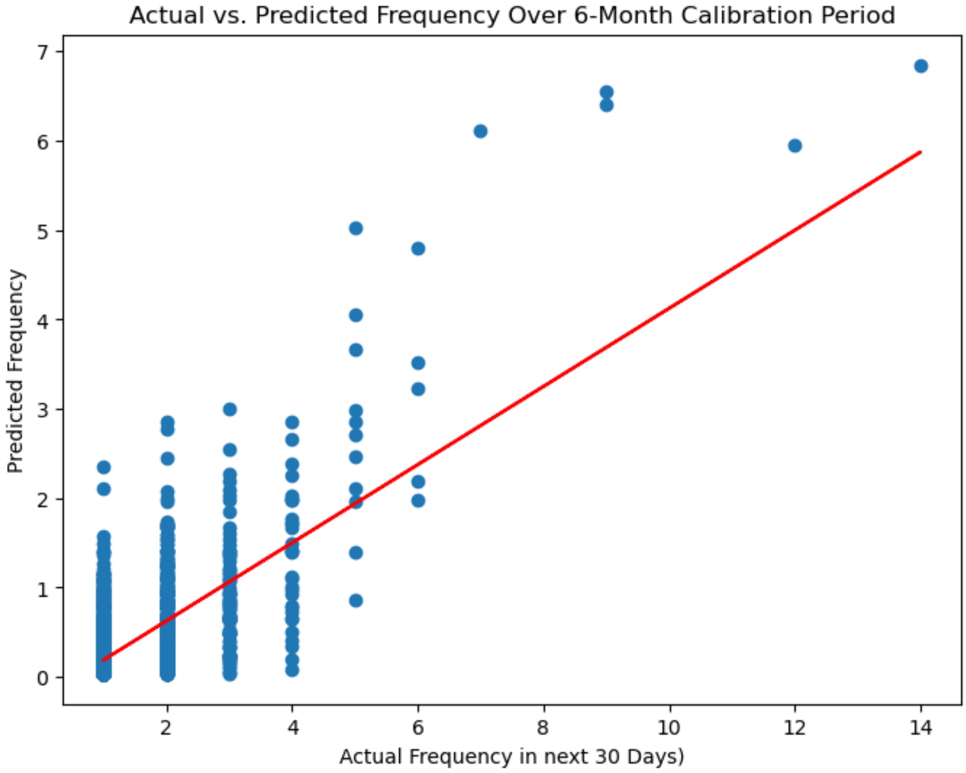


Figure 5.2.2.6: Actual vs. Predicted Frequency Over 6-Month Calibration Period.

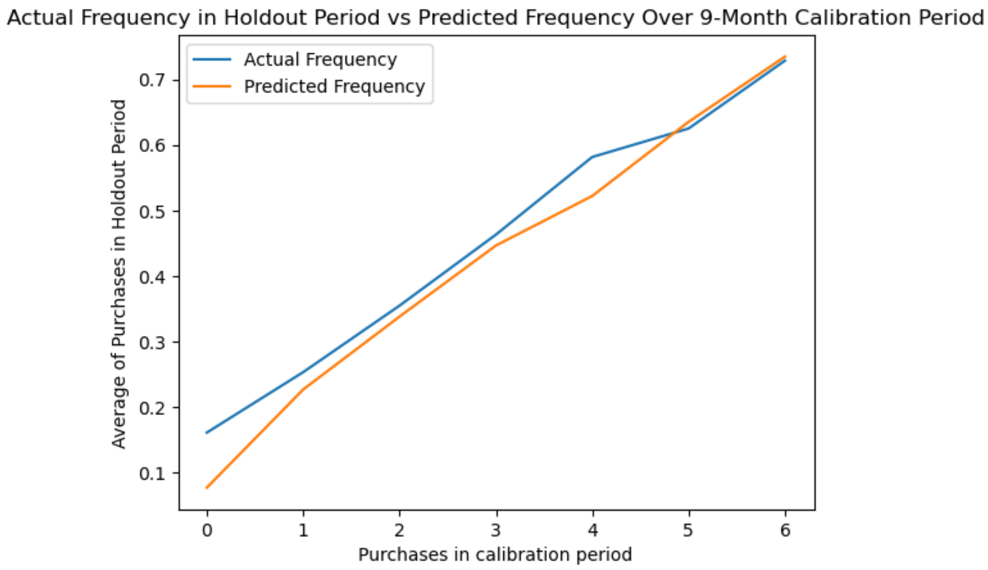


Figure 5.2.2.7: Actual Frequency in Holdout Period vs Predicted Frequency Over 9-Month Calibration Period.

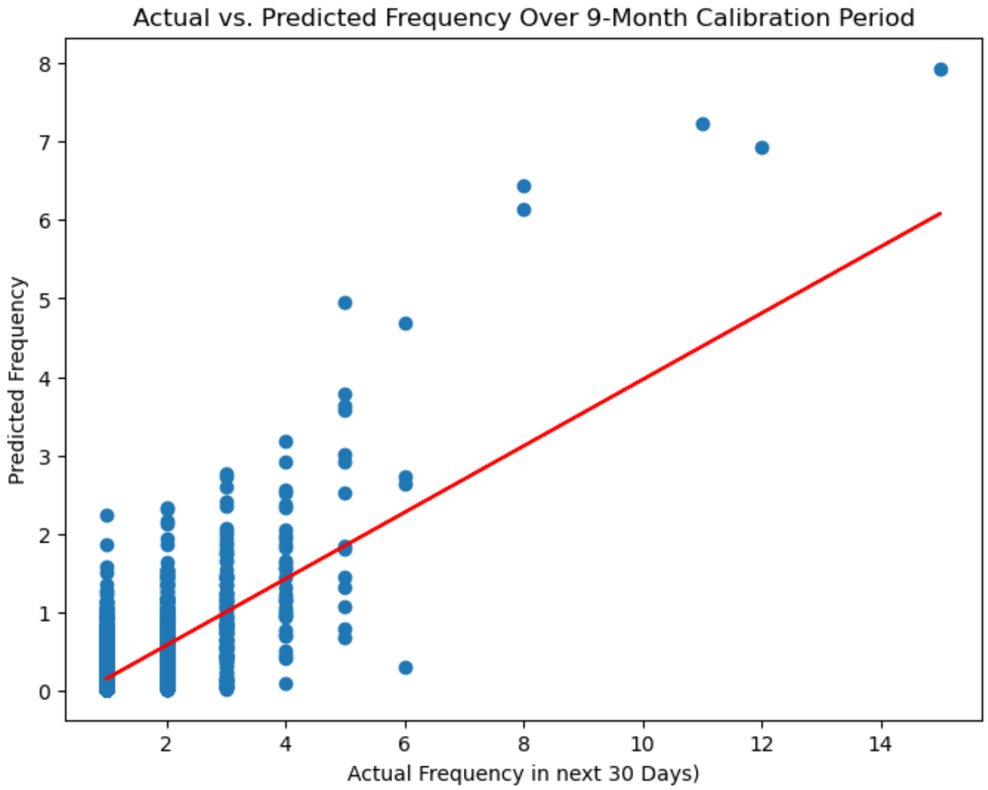


Figure 5.2.2.8: Actual vs. Predicted Frequency Over 9-Month Calibration Period.

The visualization of the model performance also provided. There are two charts were generated for each model to illustrate the relationship between predicted and actual frequencies. The first chart for each model which are Figure 5.2.2.3, Figure 5.2.2.5 and Figure 5.2.2.7 are generated through the "plot_calibration_purchases_vs_holdout_purchases" function to provide an overview of the model's accuracy by comparing the actual frequency in the holdout period to the predicted frequency. Besides, the second chart for each model which are Figure 5.2.2.4, Figure 5.2.2.6 and Figure 5.2.2.8 provides a more detailed analysis with scatter plot. The scatter plot shows the distribution of actual frequency versus predicted frequency for each customer with a best-fit line to illustrate the linear relationship between two axes. From the figures above, the graph shows that the prediction model performs well during the 6-month calibration period. The dot is closely aligned with the line where it is the best-fit line through the data points. However, in the graphs for the 3-months and 9-mobths calibration period, the dots are more spread out and do not match closely with the best fit line and indicates that the model is less accurate compared to the model with 6-month calibration period.

Table 5.1: RMSE scores of models with different calibration periods.

Calibration Period	RMSE Value
3 months	1.2042790356684872
6 months	1.1631185794207155
9 months	1.1965352498009296

After evaluating the model, it was found that the model trained over a 6-month calibration period has the best performance among three difference calibration periods. This model has the lowest RMSE value of 1.16 and demonstrates that the model trained over a 6-month calibration period has higher accuracy in predicting future frequency compared to the models trained over 3-month and 9-month calibration periods.

5.3 Algorithm Selection for Market Basket Analysis

5.3.1 Comparison between Apriori and FP-Growth Algorithms

To select the algorithm with better performance, the comparison between the Apriori and FP-Growth algorithms is carried out in terms of execution time, rule generation time, and memory consumption. The same dataset is applied to both algorithms to make sure that the comparison is fair.

Table 5.2: Performance Comparison of Apriori and FP-Growth Algorithms

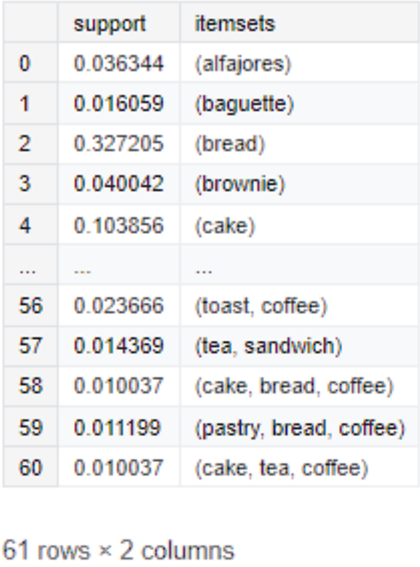
Performance Measure	Apriori	FP-Growth
Execution Time	1.187246 seconds	1.571455 seconds
Rule Creation Time	0.002999 seconds	0.003000 seconds
Memory Usage	2460.980469 MB	0.218750 MB

Table 5.2 shows the performance of both algorithms after executing the Apriori and FP-Growth algorithms. For the execution time, the Apriori algorithm generated the frequent itemset in around 1.19 seconds while FP-Growth took 1.57 seconds. Although FP-Growth is well-known for its better efficiency, the dataset used in this comparison might not be large enough to fully utilize the advantage of FP-Growth. Hence, the time difference between two algorithms might not be significant if the dataset is small or medium in size. After generating the frequent itemsets, both algorithms proceeded to the rule creation phase. Apriori completed this step in 0.002999 seconds while FP-Growth took slightly longer which is 0.003 seconds which can considered almost same. For the memory usage, Apriori used 2460.98 MB of memory while FP-Growth only used 0.22 MB. The memory usage for Apriori is higher because it needs to generate and scan multiple candidate itemsets which is resource-intensive especially for larger datasets. On the other hand, FP-Growth used a more memory-efficient method which is created a FP-Tree that reduces the need for multiple scans of the data.

Thus, since both algorithms performed similarly in execution time and rule creation time but huge difference in memory usage indicates that Apriori’s larger memory consumption could make it unsuitable for large datasets while FP-Growth would perform better for large datasets. Hence, FP-Growth is selected to apply in the following sections.

5.3.2 Algorithm Validation for Market Basket Analysis

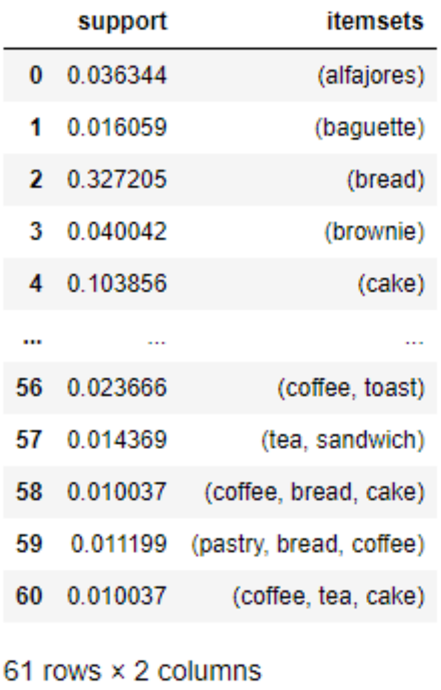
To further validate the performance of the Apriori algorithm, an additional dataset from Kaggle was used. The dataset used in this validation is “The Bread Basket” dataset that obtained from Kaggle [15]. In this dataset, it consists of 20507 transaction records with 4 column which are “Transaction”, “Item”, “date_time”, period_day” and “weekday_weekend”. Before applying the Apriori algorithm, the preprocessing step need to perform to ensure that the dataset is clean. Then, same with the original dataset, the transaction data was transformed into a suitable format for Apriori algorithm. Next, the Apriori algorithm is applied with the use of apriori() function and using the same parameter where the minimum support is equal to 0.01 to generate the frequent itemsets. Besides, the association rules also generated from the frequent itemsets.



	support	itemsets
0	0.036344	(alfajores)
1	0.016059	(baguette)
2	0.327205	(bread)
3	0.040042	(brownie)
4	0.103856	(cake)
...	...	...
56	0.023666	(toast, coffee)
57	0.014369	(tea, sandwich)
58	0.010037	(cake, bread, coffee)
59	0.011199	(pastry, bread, coffee)
60	0.010037	(cake, tea, coffee)

61 rows × 2 columns

Figure 5.3.2.1: The frequent itemsets generated from the Kaggle by the original author.



	support	itemsets
0	0.036344	(alfajores)
1	0.016059	(baguette)
2	0.327205	(bread)
3	0.040042	(brownie)
4	0.103856	(cake)
...	...	...
56	0.023666	(coffee, toast)
57	0.014369	(tea, sandwich)
58	0.010037	(coffee, bread, cake)
59	0.011199	(pastry, bread, coffee)
60	0.010037	(coffee, tea, cake)

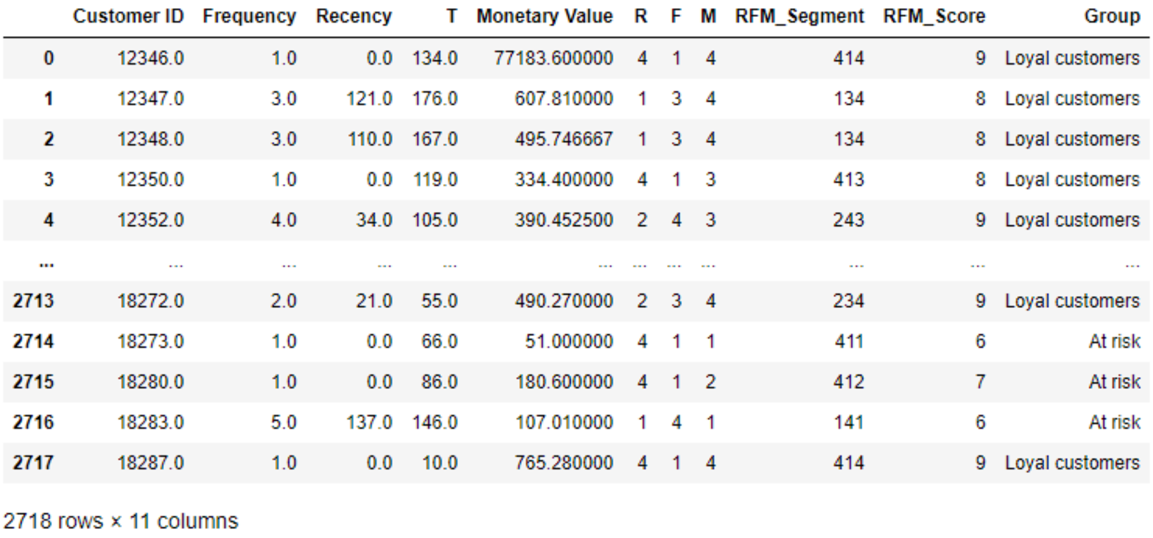
61 rows × 2 columns

Figure 5.3.2.2: The frequent itemsets generated by own Apriori model.

From Figure 5.3.2.1 and Figure 5.3.2.2, it can be observed that both the frequent itemsets produced by own implementation are identical to the frequent itemsets that generated by the original author.

5.4 Customer Segmentation

For the customer segmentation, the RFM analysis is applied to the dataset to segment the customers into different groups based on their purchasing behavior. RFM stands for Recency, Frequency, and Monetary value, and it is widely used in grouping the customers based on their transaction history. Figure 5.4.1 shows the result after performed RFM analysis.



	Customer ID	Frequency	Recency	T	Monetary Value	R	F	M	RFM_Segment	RFM_Score	Group
0	12346.0	1.0	0.0	134.0	77183.600000	4	1	4	414	9	Loyal customers
1	12347.0	3.0	121.0	176.0	607.810000	1	3	4	134	8	Loyal customers
2	12348.0	3.0	110.0	167.0	495.746667	1	3	4	134	8	Loyal customers
3	12350.0	1.0	0.0	119.0	334.400000	4	1	3	413	8	Loyal customers
4	12352.0	4.0	34.0	105.0	390.452500	2	4	3	243	9	Loyal customers
...	...	...	...	...	...	...	...	...	...	...	...
2713	18272.0	2.0	21.0	55.0	490.270000	2	3	4	234	9	Loyal customers
2714	18273.0	1.0	0.0	66.0	51.000000	4	1	1	411	6	At risk
2715	18280.0	1.0	0.0	86.0	180.600000	4	1	2	412	7	At risk
2716	18283.0	5.0	137.0	146.0	107.010000	1	4	1	141	6	At risk
2717	18287.0	1.0	0.0	10.0	765.280000	4	1	4	414	9	Loyal customers

2718 rows × 11 columns

Figure 5.4.1: Result after performing RFM analysis.

From Figure 5.4.1, it was found that each customer is assigned to different group based on the RFM score. For example, the Customer ID 12352 has made 4 purchases within the six-month period and the customer last made a purchase. Then, the customer 12352 has spent total amount of 390.45 which is higher and indicating their high value. Based on the Frequency, Recency and Monetary Value, the R, F, M score has been assigned and sum into the RFM score which is 9 where 9 is considered as “Loyal customers”.

5.5 Recommendations for Each Customer Group

The frequent itemsets has been generated using FP-Growth for each customer group and it is considered as the product recommendations for each customer group since it shows the most popular products for each customer group. Tables below show the recommended products for each customer group.

Table 5.3: Top 10 recommended products for “Champions” group

No.	Support	Itemsets
1	0.56	PARTY BUNTING
2	0.50	JAM MAKING SET WITH JARS
3	0.44	SET OF 6 SPICE TINS PANTRY DESIGN
4	0.39	DOORMAT NEW ENGLAND
5	0.39	WHITE HANGING HEART T-LIGHT HOLDER
6	0.39	JUMBO BAG RED RETROSPOT
7	0.39	DOORMAT HEARTS
8	0.39	REGENCY CAKESTAND 3 TIER
9	0.33	VINTAGE UNION JACK BUNTING
10	0.33	RECIPE BOX PANTRY YELLOW DESIGN

Table 5.4: Top 10 recommended products for “Loyal customers” group

No.	Support	Itemsets
1	0.27	REGENCY CAKESTAND 3 TIER
2	0.22	WHITE HANGING HEART T-LIGHT HOLDER
3	0.20	PACK OF 72 RETROSPOT CAKE CASES
4	0.19	PARTY BUNTING
5	0.19	SET OF 3 CAKE TINS PANTRY DESIGN
6	0.18	JAM MAKING SET WITH JARS
7	0.17	JUMBO BAG RED RETROSPOT
8	0.17	ASSORTED COLOUR BIRD ORNAMENT
9	0.16	JAM MAKING SET PRINTED
10	0.16	LUNCH BAG RED RETROSPOT

Table 5.5: Top 10 recommended products for “At risk” group

No.	Support	Itemsets
1	0.20	WHITE HANGING HEART T-LIGHT HOLDER
2	0.13	REGENCY CAKESTAND 3 TIER
3	0.12	PARTY BUNTING
4	0.11	REX CASH+CARRY JUMBO SHOPPER
5	0.11	NATURAL SLATE HEART CHALKBOARD
6	0.11	ASSORTED COLOUR BIRD ORNAMENT

Table 5.6: Top 10 recommended products for “Hibernating” group

No.	Support	Itemsets
1	0.25	PARTY BUNTING
2	0.20	REX CASH+CARRY JUMBO SHOPPER
3	0.20	HEART OF WICKER LARGE
4	0.18	WHITE HANGING HEART T-LIGHT HOLDER
5	0.14	3 HEARTS HANGING DECORATION RUSTIC
6	0.14	HEART OF WICKER LARGE, WHITE HANGING HEART T-LIGHT HOLDER
7	0.14	COLOUR GLASS T-LIGHT HOLDER HANGING
8	0.14	REGENCY CAKESTAND 3 TIER
9	0.14	HEART OF WICKER SMALL
10	0.11	POPPY'S PLAYHOUSE BEDROOM

From the tables above, the most popular products across different customer groups can be observed. For the “Champion” group, it has higher support values and more variety recommendations. For example, the items like “PARTY BUNTING” and “JAM MAKING SET WITH JARS” are popular with support values of 0.56 and 0.50 respectively indicates that both items appeared in 56% and 50% respectively of the transactions that made by “Champion” customers.

For the “Loyal customers” group, it has slightly lower support values compared to the “Champion” group, but it still includes a variety of recommended products. For instance, “REGENCY CAKESTAND 3 TIER” and “WHITE HANGING HEART T-

LIGHT HOLDER" have support values of 0.27 and 0.22 indicates that they appeared in 27% and 22% respectively of "Loyal" customer transactions. From table 5.4, there are some products such as "PARTY BUNTING", "JAM MAKING SET WITH JARS", "JUMBO BAG RED RETROSPOT" and "JAM MAKING SET WITH JARS" have existed in both tables. This indicates that even though the "Loyal customers" group may not purchase frequently as "Champions", they still purchased the same products where it demonstrated that they have similar interest.

From table 5.5, it was noticed that the "At risk" customer group only have 6 recommended products which means that the customers purchase fewer distinct products. The most popular item in this group is "WHITE HANGING HEART T-LIGHT HOLDER" with a support value of 0.20, meaning it appears in 20% of transactions. This indicates that "At risk" customers have a lower engagement and thus the range of recommended products is lesser compared to more active groups like "Champion" or "Loyal customers."

From table 5.6, the "Hibernating" group has support values similar to the "At Risk" group. In this group, the top item is "PARTY BUNTING" with a support value of 0.25 means that it appeared in 25% of transactions indicates that while these customers are not active as other group, they still show their interest in specific products.

In conclusion, the difference of the recommended products between the customer groups highlights the importance of study the behaviours of each customer group and find out the similarity among these groups. For example, the products such as "PARTY BUNTING", "WHITE HANGING HEART T-LIGHT HOLDER" and "REGENCY CAKESTAND 3 TIER" appeared in all customer group indicates that these are products that need to be considered first.

5.6 Project Challenges

There are several challenges or limitations encountered during the project. One of the challenges is to determine the suitable value for regularization coefficient which is the regularization parameter that used to prevent overfitting. The relationship between model complexity and generalization performance needs to be considered as choosing too low regularization coefficient may lead to overfitting while choosing too high regularization coefficient may result in underfitting.

The other issue was understanding the functions that provided by lifetimes package. There are lots of functions in the lifetimes package such as prediction, visualization, summarization and so on. Those functions need to explore by reviewing the documentation of lifetimes package and the documentation may not include the detailed explanations of how each function works or the theory behind it. Thus, it is time consuming to explore the function and understand it.

Then, the dataset used in this project initially contained 541910 rows. However, after performing data cleaning, the dataset was reduced to 397925 records. Furthermore, the size of the data is reduced when only focusing on the six-month period and only the transactions data of 2718 customers are included in this analysis. Although the reduction is necessary to ensure the data quality, it also limited the scope of the analysis and affected the final results of the product recommendations.

Chapter 6
Conclusion and Recommendations

6.1 Conclusion

It was a big challenge for businesses in understanding the customer behavior to increase competitiveness in today's market especially in predicting the likelihood of future purchases for each customer and provide the product recommendations to the customer. By applying the BG/NBD model, RFM segmentation and FP-Growth algorithm, it is able to group customers into different customer groups and recommend products based on their purchasing behavior.

Through experimentation, it was found that using 6-month period for the BG?NBD model provided the most accurate predictions of customer purchase frequency with the lowest RMSE compared to other periods which are three and nine months. This indicates that choosing an appropriate time period is also important to get more accurate predictions.

For comparison between Apriori and FP-Growth algorithms, the Apriori algorithm is proved that it is faster in execution compared to FP-Growth algorithms when using the data of this project. However, the Apriori algorithm require more memory due to the candidate generation process making it only suitable for small dataset. Thus, the FP-Growth algorithm is more preferred when the size of the dataset is increased.

Through RFM segmentation, the customers are categorized into different groups which are "Champions", "Loyal customers", "At risk" and "Hibernating". The majority of customers were classified into the "At risk" category with 50.74% and followed closely by "Loyal customers" which is 46.98%. Moreover, small minority were classified as "Champions" with only 0.66% and "Hibernating" with 1.62%. Then, product recommendations were generated for each customer group using FP-Growth algorithm. Due to the limited number of Champions, it required a minimum support value of at least 0.2 to create the frequent itemsets while other groups can used minimum support values of at least 0.1 to generate the frequent itemsets. Despite having the largest proportion of customers, the "At risk" category generated only 6 frequent itemsets indicates that their purchased products are less diverse. This may conclude that “At risk” customers may have limited interests on other products and difficult to provide product recommendations to them.

6.2 Recommendations

To further improve the entire project, the future work can focus on applying the personalized product recommendations. Instead of recommending products based on the customer group, another technique can be applied such as collaborative filtering or content-based filtering. These techniques analyze the behavior of the customer and provide better personalized product recommendations. For example, the collaborative filtering can used to predict the products that a customer might interested based on the purchasing records of other customers who have similar purchasing habits. This approach allows the businesses to customize their offerings to the customers especially for the customers that categorized in “Champions’ and “Loyal customers” who may receive the customized offers and discounts.

Then, it can integrate the real-time transaction data into the recommendation model to provide more flexible and accurate recommendations. This is because the real-time data consists of the latest data and allows for quick adaptation to the latest trends so that the recommendation can adjust based on the latest trends. Furthermore, the model can provide the latest and customized recommendations by constantly updated the data.

REFERENCES

[1] D. Mensouri and A. Azmani, “A New Marketing Recommendation System Using a Hybrid Approach to Generate Smart Offers,” *Applied Computer Systems*, vol. 27, no. 2, pp. 149–158, 2022. https://doi.org/10.2478/acss-2022-0016

[2] C. Gakii and R. Rimiru, “Identification of cancer related genes using feature selection and Association Rule Mining,” *Informatics in Medicine Unlocked*, vol. 24, p. 100595, 2021. https://doi.org/10.1016/j.imu.2021.100595

[3] A. L. Pearce *et al.*, “Using association rules mining to characterize loss of Control Eating in childhood,” *Appetite*, vol. 163, p. 105236, 2021. https://doi.org/10.1016/j.appet.2021.105236

[4] T. A. Kumbhare and S. V. Chobe, “An Overview of Association Rule Mining Algorithms”, 2014. [Online]. Available: https://www.semanticscholar.org/paper/An-Overview-of-Association-Rule-Mining-Algorithms-Kumbhare-Chobe/d4058d9f3f66c53ddea776c974fbd740afd994b4

[5] A. A. Aldino, E. D. Pratiwi, Setiawansyah, S. Sintaro, and A. Dwi Putra, “Comparison of Market Basket Analysis To Determine Consumer Purchasing Patterns Using Fp-Growth And Apriori Algorithm,” *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, pp. 29-34, 2021. doi: 10.1109/icomitee53461.2021.9650317

[6] C.-H. Chee, J. Jaafar, I. A. Aziz, M. H. Hasan, and W. Yeoh, “Algorithms for frequent itemset mining: A literature review,” *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2603–2621, 2018. https://doi.org/10.1007/s10462-018-9629-z

[7] X. Zhang and J. Zhang, “Analysis and research on library user behavior based on Apriori algorithm,” *Measurement: Sensors*, vol. 27, p. 100802, 2023. https://doi.org/10.1016/j.measen.2023.100802

[8] Y. Guo, M. Wang, and X. Li, “Application of an improved Apriori algorithm in a mobile e-commerce recommendation system,” *Industrial Management &amp; Data Systems*, vol. 117, no. 2, pp. 287–303, 2017. https://doi.org/10.1108/IMDS-03-2016-0094

[9] K. Garg and D. Kumar, “Comparing the Performance of Frequent Pattern Mining Algorithms,” *International Journal of Computer Applications*, vol. 69, no. 25, pp. 29–32, 2013. http://dx.doi.org/10.5120/12129-8502

[10] Y. Zeng, S. Yin, J. Liu, and M. Zhang, “Research of Improved FP-growth Algorithm in Association Rules Mining,” *Scientific Programming*, vol. 2015, pp. 1–6, 2015. https://doi.org/10.1155/2015/910281

[11] S. Nasreen, M. A. Azam, K. Shehzad, U. Naeem, and M. A. Ghazanfar, "Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey," *Procedia Computer Science*, vol. 37, pp. 109–116, 2014. https://doi.org/10.1016/j.procs.2014.08.019

[12] A. Mammadzada, E. Alasgarov, and A. Mammadov, "Application of BG / NBD and Gamma-Gamma Models to Predict Customer Lifetime Value for Financial Institution," *2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–6, 2021. doi: 10.1109/aict52784.2021.9620535

[13] A. Joy Christy, A. Umamakeswari, L. Priyatharsini, and A. Neyaa, "RFM ranking – An effective approach to customer segmentation," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, pp. 1251-1257, 2021. https://doi.org/10.1016/j.jksuci.2018.09.004

[14] C. Rungruang, P. Riyapan, A. Intarasit, K. Chuarkham, and J. Muangprathub, "RFM model customer segmentation based on hierarchical approach using FCA," *Expert Systems with Applications*, vol. 237, p. 121449, Mar. 2024. https://doi.org/10.1016/j.eswa.2023.121449

[15] V. Bagga, "Apriori algorithm," Kaggle, Accessed: Sep. 5, 2024. [Online] Available: https://www.kaggle.com/code/nandinibagga/apriori-algorithm

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 1**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Revise the previous work

2. WORK TO BE DONE

- Copy and paste chapter 1 and 2 to fyp2 template
- Do the apriori algorithm

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Progress is good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 3**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Copy and paste chapter 1 and 2 to fyp2 template
- Do the apriori algorithm

2. WORK TO BE DONE

- Do the FP-Growth algorithm

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Progress is good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 5**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Do the FP-Growth algorithm
- Do the comparison between two algorithm and select the algorithm with better performance

2. WORK TO BE DONE

- Do the RFM analysis
- Find another dataset for validation

3. PROBLEMS ENCOUNTERED

- Difficult to find the paper that consist dataset

4. SELF EVALUATION OF THE PROGRESS

- Progress is good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 7**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Do the RFM analysis
- Find another dataset for validation

2. WORK TO BE DONE

- Do the customer segmentation for each group

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Need to manage time for this project and other subjects since midterm is near.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 9**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Do the customer segmentation for each group
- Redo the flowchart

2. WORK TO BE DONE

- Apply FP-Growth to each customer group
- Modify the code
- Complete the report

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Progress is good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 11**
Student Name & ID: Wong Ji Hin 21ACB00298	
Supervisor: Dr Kh'ng Xin Yi	
Project Title: Customer Purchase Prediction and Product Recommendations	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finish up the code

2. WORK TO BE DONE

- Finish the rest of the report

3. PROBLEMS ENCOUNTERED

- N/A

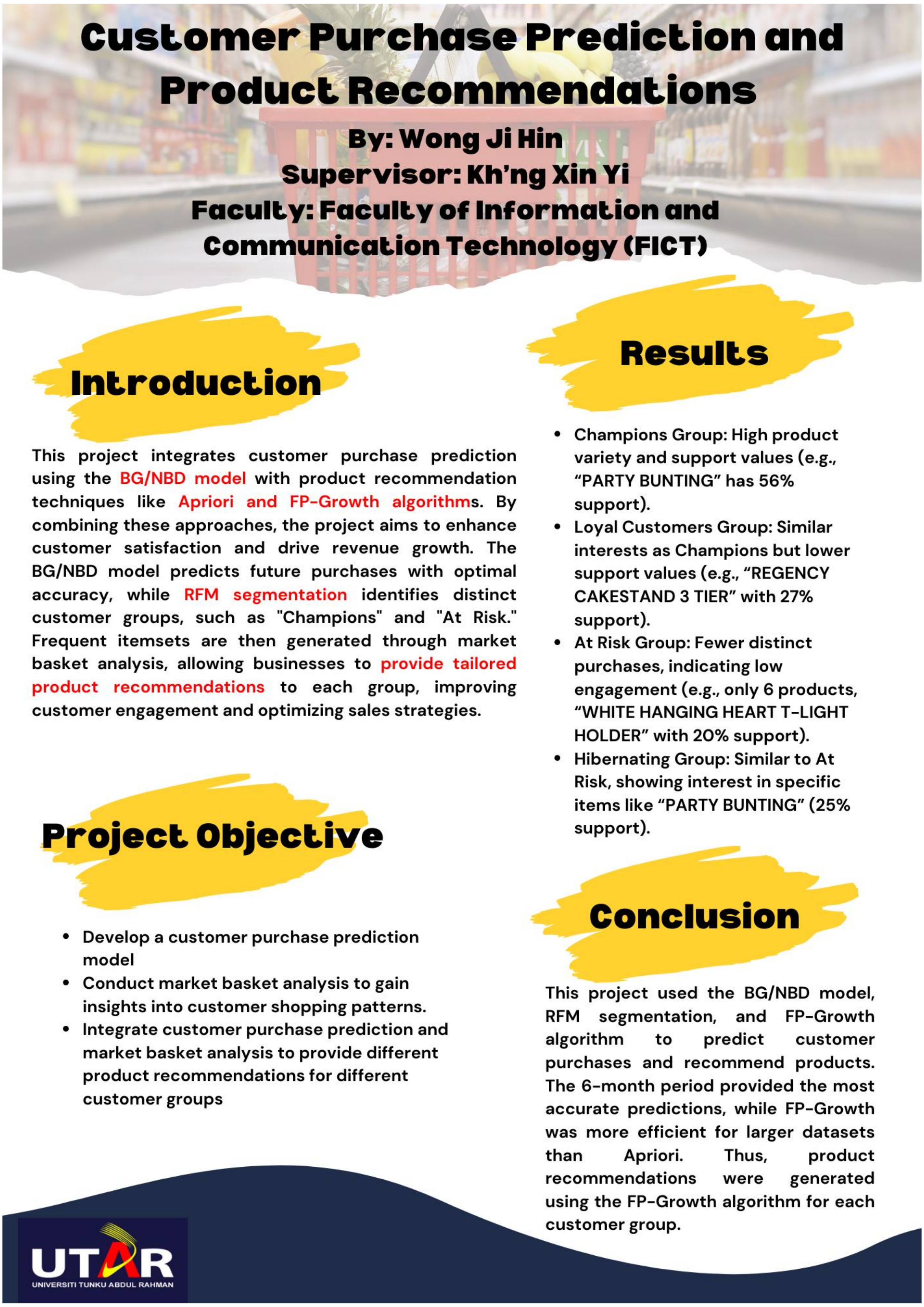
4. SELF EVALUATION OF THE PROGRESS

- Progress is good.

Supervisor's signature

Student's signature

POSTER



Customer Purchase Prediction and Product Recommendations

By: Wong Ji Hin
Supervisor: Kh'ng Xin Yi
Faculty: Faculty of Information and Communication Technology (FICT)

Introduction

This project integrates customer purchase prediction using the BG/NBD model with product recommendation techniques like Apriori and FP-Growth algorithms. By combining these approaches, the project aims to enhance customer satisfaction and drive revenue growth. The BG/NBD model predicts future purchases with optimal accuracy, while RFM segmentation identifies distinct customer groups, such as "Champions" and "At Risk." Frequent itemsets are then generated through market basket analysis, allowing businesses to provide tailored product recommendations to each group, improving customer engagement and optimizing sales strategies.

Project Objective

- Develop a customer purchase prediction model
- Conduct market basket analysis to gain insights into customer shopping patterns.
- Integrate customer purchase prediction and market basket analysis to provide different product recommendations for different customer groups

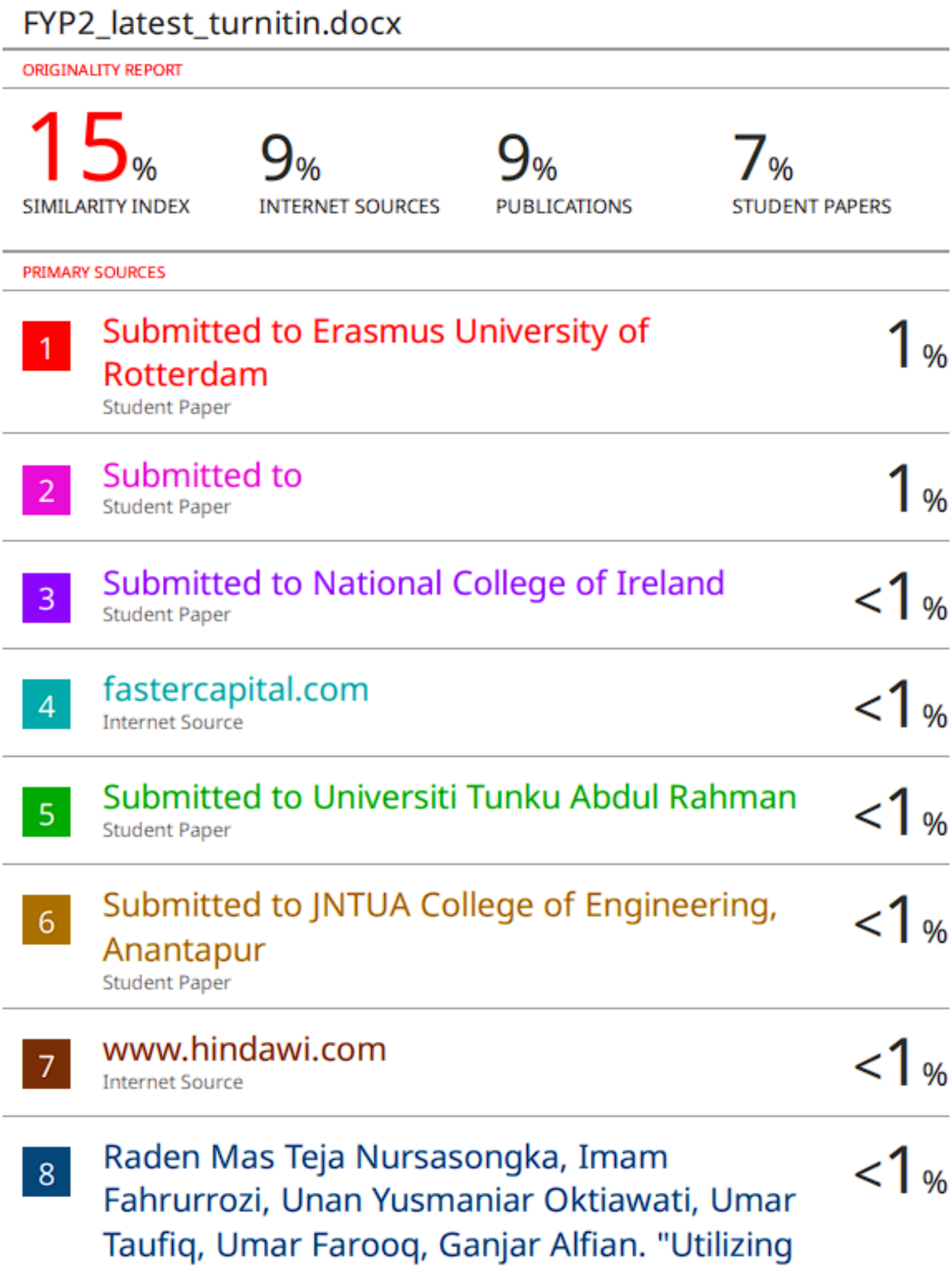
Results

- Champions Group: High product variety and support values (e.g., "PARTY BUNTING" has 56% support).
- Loyal Customers Group: Similar interests as Champions but lower support values (e.g., "REGENCY CAKESTAND 3 TIER" with 27% support).
- At Risk Group: Fewer distinct purchases, indicating low engagement (e.g., only 6 products, "WHITE HANGING HEART T-LIGHT HOLDER" with 20% support).
- Hibernating Group: Similar to At Risk, showing interest in specific items like "PARTY BUNTING" (25% support).

Conclusion

This project used the BG/NBD model, RFM segmentation, and FP-Growth algorithm to predict customer purchases and recommend products. The 6-month period provided the most accurate predictions, while FP-Growth was more efficient for larger datasets than Apriori. Thus, product recommendations were generated using the FP-Growth algorithm for each customer group.

PLAGIARISM CHECK RESULT



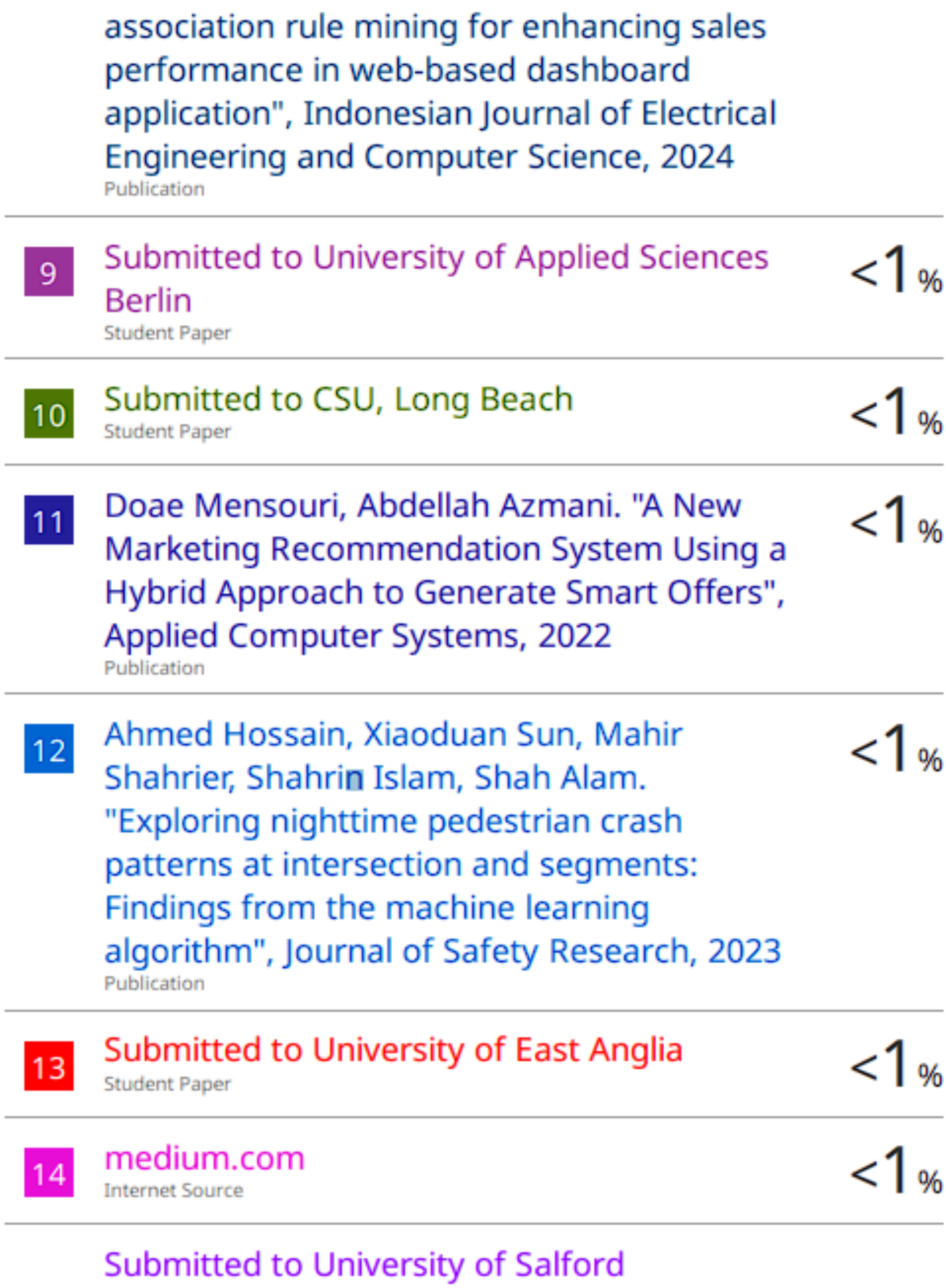
FYP2_latest_turnitin.docx

ORIGINALITY REPORT

15%	9%	9%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Erasmus University of Rotterdam Student Paper	1%
2	Submitted to Student Paper	1%
3	Submitted to National College of Ireland Student Paper	<1%
4	fastercapital.com Internet Source	<1%
5	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
6	Submitted to JNTUA College of Engineering, Anantapur Student Paper	<1%
7	www.hindawi.com Internet Source	<1%
8	Raden Mas Teja Nursasongka, Imam Fahrurrozi, Unan Yusmaniar Oktiawati, Umar Taufiq, Umar Farooq, Ganjar Alfian. "Utilizing	<1%



	association rule mining for enhancing sales performance in web-based dashboard application", Indonesian Journal of Electrical Engineering and Computer Science, 2024 Publication	
9	Submitted to University of Applied Sciences Berlin Student Paper	<1%
10	Submitted to CSU, Long Beach Student Paper	<1%
11	Doae Mensouri, Abdellah Azmani. "A New Marketing Recommendation System Using a Hybrid Approach to Generate Smart Offers", Applied Computer Systems, 2022 Publication	<1%
12	Ahmed Hossain, Xiaoduan Sun, Mahir Shahrier, Shahrin Islam, Shah Alam. "Exploring nighttime pedestrian crash patterns at intersection and segments: Findings from the machine learning algorithm", Journal of Safety Research, 2023 Publication	<1%
13	Submitted to University of East Anglia Student Paper	<1%
14	medium.com Internet Source	<1%
	Submitted to University of Salford	

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Wong Ji Hin
ID Number(s)	21ACB00298
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Customer Purchase Prediction and Product Recommendations

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 15 **%** **Similarity by source** Internet Sources: 9 % Publications: 9 % Student Papers: 7 %	Overall similarity index < 20%
Number of individual sources listed of more than 3% similarity: -	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor

Name: Kh'ng Xin Yi

Date: 13/9/2024

____
Signature of Co-Supervisor

Name: ______________________

Date: ______________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	21ACB00298
Student Name	Wong Ji Hin
Supervisor Name	Dr Kh'ng Xin Yi

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 11-Sep-2024