CUSTOMER BEHAVIOUR SIMULATOR FOR TRAINING RL AGENTS: REAL WORLD APPLICATION

Department of Mathematics
IIT MADRAS

Name of the Project Guides
Prof. Dr. S. Sundar
Dr. Sri Vallabha Deevi(Tiger Analytics)

Submitted by:

Sandip Sing (MA21M024)

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

DEPARTMENT OF MATHEMATICS

NAME : Sandip Sing

Roll No : MA21M024

PROGRAMME : M.Tech.

DATE OF JOINING : 28.07.2021

Guide : Prof. Dr. S. Sundar

Co-guide : Dr. Sri Vallabha Deevi

Certificate

This is to certify that the report **"CUSTOMER BEHAVIOUR SIMULATOR FOR TRAINING RL AGENTS: REAL WORLD APPLICATION"**, submitted by **Sandip Sing (MA21M024)**, in partial fulfilment of the requirement for the award of the degree of Master of Technology in Industrial Mathematics and Scientific Computing, Indian Institute of Technology Madras, is the record of the work done by him during the academic year 2022-2023 in the Department of Mathematics, IIT Madras, India, under my supervision.

Prof. Dr. S. Sundar
Department of Mathematics
IIT Madras

Dr. Sri Vallabha Deevi
Tiger Analytics
Chennai, India

Acknowledgement

I would like to thank Prof. Dr. S. Sundar and Dr. Sri Vallabha Deevi for their constant help and support, without which the completion of my project for this academic year would be impossible. They have always been there to guide me throughout the course and have patiently solved all the queries associated with the project. Lastly, I would like to thank the Department of Mathematics, IIT Madras and Tiger Analytics for providing me the opportunity to work in this exciting field.

Sandip Sing MA21M024
Department of Mathematics
IIT Madras

Abstract

The customer behaviour simulator model effectively trains Reinforcement Learning (RL) agents to predict and influence customer behaviour. In this study, a simulator model was developed that considers various customer attributes, such as age, income and purchase history, to create a realistic and dynamic environment for the agents. The Q-learning algorithm was used to train the agents and learn more rewarding actions, leading to better decision-making. The effectiveness of the approach was evaluated by analyzing the validation error and comparing the predicted and actual reward values. RL predicted reward was significantly higher than the historical reward, indicating the success of the model's performance. The model has great potential for optimizing marketing strategies, improving customer retention and increasing revenue. In future work, more sophisticated reinforcement learning algorithms can be explored, more variables can be incorporated into the model and the application can be extended to other domains and industries. Overall, the customer behaviour simulator model for training reinforcement learning agents has shown great promise for enhancing customer behaviour prediction and decision-making.

Contents

Chapter 1

Introduction

1.1 Consumer behaviour in marketing

The study of how individuals make decisions regarding the purchase, usage and disposal of goods and services is known as consumer behaviour [1]. It is an essential aspect of marketing, as understanding consumer behaviour helps marketers identify the needs and wants of their target market and develop effective marketing strategies to meet those needs.

Figure 1.1 illustrates how consumer behaviour is influenced by various factors, including cultural, social, personal and psychological factors. For example, cultural factors such as religion, language and social norms can influence consumer behaviour by shaping consumers' values, beliefs and attitudes towards specific products or services. Social factors such as family, friends and reference groups can influence consumer behaviour by providing social support, information and influence.

Personal factors such as age, income, lifestyle and personality can also influence consumer behaviour. Consumer behaviour varies depending on individual factors such as age, where younger consumers tend to be more influenced by trends and fashion when making purchasing decisions. In comparison, older consumers tend to prioritize convenience and comfort. Psychological factors like motivation, perception, learning and attitudes can also affect consumer behaviour. For example, consumers may be motivated to buy a product if it meets a particular need, such as hunger, thirst or a desire for social status.

Marketers must understand consumer behaviour well to develop effective marketing strategies. By understanding the needs and wants of their target market, marketers can develop products and services that meet those needs and wants, create compelling advertising campaigns that appeal to their target market and set acceptable prices. They can also use consumer behaviour research to identify new market opportunities, monitor consumer behaviour changes and adjust their marketing strategies. Ultimately, the better a marketer understands consumer behaviour, the more successful they are likely to be in developing and marketing products and services that meet the needs and wants of their target market.

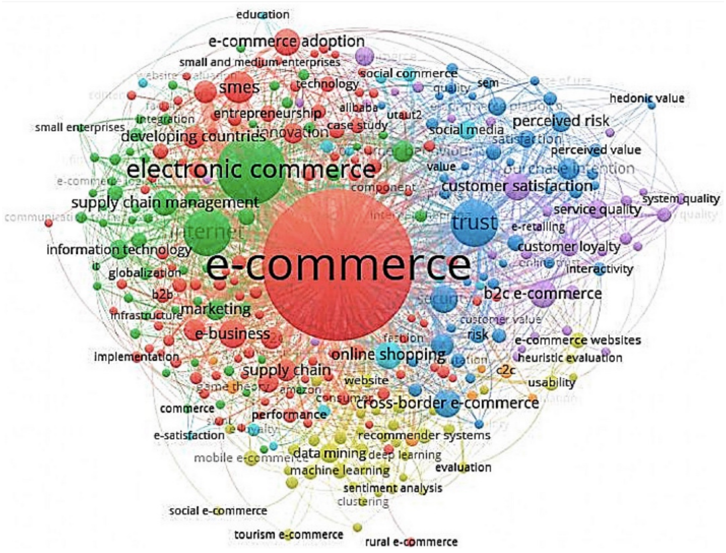

Figure 1.1: Consumer behaviour [2]

1.2 Importance of understanding the behaviour of consumers

Consumer behaviour is crucial for industries as it provides insights into their target customers' behaviour and decision-making processes. Understanding consumer behaviour helps industries develop effective marketing strategies influencing consumer behaviour and driving sales. Here are some reasons why consumer behaviour is essential in industries:

Identify customer needs: By conducting consumer behaviour research, industries can acquire valuable insights into their intended audience's needs, preferences and desires. By understanding customer needs, industries can develop products and services that meet those needs and create value for customers.

Develop effective marketing strategies: Understanding consumer behaviour helps industries to develop effective marketing strategies that can influence consumer behaviour and drive sales. By understanding consumer behaviour, industries can tailor their marketing messages and promotional efforts to specific customer segments, increasing the effectiveness of their marketing campaigns.

Competitive advantage: Understanding consumer behaviour can provide a competitive advantage to industries. By creating a distinct value proposition that caters to the requirements of their target audience, industries can set themselves apart from their competitors.

Innovation: Consumer behaviour research can also provide insights into emerging trends and changing consumer preferences. By identifying these trends early, industries can develop innovative products and services to address changing customer needs and preferences.

Increase sales: Ultimately, understanding consumer behaviour can help industries increase sales and revenue. Developing effective marketing strategies, creating customer-centric products, and providing a unique value proposition is essential for industries to attract and retain customers, increase sales, and improve profitability.

1.3 Types of customer behaviour

In figure 1.2, many types of customer behaviour are depicted, but the following are some of the most commonly occurring types of customer behaviour :

Purchase behaviour: This refers to the actions a customer takes when making a purchase, such as the type of product they buy, how frequently they make purchases and how much they are willing to spend.

Brand loyalty: This is the degree to which a customer is committed to a particular brand and continues to purchase from that brand over time.

Information-seeking behaviour: This refers to a customer's actions when researching a product or service before purchasing, such as reading reviews, comparing prices and looking for product specifications.

Social influence: This refers to other people's impact on a customer's purchasing decisions, including the opinions of friends, family and online influencers.

Customer satisfaction: The level of satisfaction a customer experiences with a product or service can influence their future purchasing decisions and affect their likelihood of recommending the product or service to others.

Complaint behaviour: This refers to a customer's actions when they are dissatisfied with a product or service, including complaining to the company, leaving negative reviews and telling others about their experience.

Figure 1.2: Types of consumer behaviour [3]

1.4 Present work

Several simulator models were developed using Machine Learning techniques on the "Dunnhumby-The Complete Journey" dataset [4] obtained from Kaggle. These models included predicting the next time a customer will visit, determining what a customer is likely to purchase next, identifying the discounts needed to retain a customer, and forecasting the revenue generated from a customer after offering discounts.

To simulate the effect of an unknown coupon discount percentage (action), a model was developed to predict the customer's next state and reward value. Tensorflow was used to train Reinforcement Learning (RL) models using this customer simulator environment. The RL models were trained to learn the optimal policy based on the historical dataset. From this optimal policy, RL determined the best action for unknown data in the test dataset.

Additionally, the customer simulator was used to predict the reward value, which was compared to the actual reward value of the test data to evaluate the RL model's performance. The RL model's effectiveness in determining the optimal action was assessed by measuring the improvement of the predicted reward values against the actual reward values of the test dataset.

Chapter 2

Evaluation metrics for predictive models

2.1 Introduction

Evaluation metrics are essential to determine the effectiveness and accuracy of predictive models. In machine learning, predictive models are trained to predict future outcomes based on past data. These predictions can be used for various purposes, such as forecasting sales or predicting customer behaviour. Also, it is essential to assess how well the predictive models are performing. This is where evaluation metrics come into play. Evaluation metrics provide a quantitative measure of the model's accuracy, allowing us to compare different models and choose the best one for the task.

Various evaluation metrics are used in machine learning, each with its strengths and weaknesses. The most commonly used metrics include accuracy, precision, recall, and F_1 score. This chapter will discuss these evaluation metrics in detail and explain how they are calculated and interpreted. Evaluating a predictive model often involves analyzing the difference between the predicted and actual values. The actual value refers to the true or observed value of the target variable. In contrast, the predictive value refers to the value predicted by the model for a given set of input features.

2.2 Coefficient of determination(R^2)

R^2, also known as the coefficient of determination, is a commonly used metric to evaluate the goodness of fit of regression models. It is often used in regression problems, where the goal is to predict a continuous numerical value. The mathematical expression for the coefficient of determination [5] is:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

SS_{res} is the sum of the squared differences between the predicted and actual values. SS_{tot} is the sum of the squared differences between the actual values and the mean of the dependent variable.

R-squared value is a statistical measure representing the proportion of the variance in the dependent variable explained by the independent variable(s). It ranges between 0 and 1, where a value of 0 indicates that the model explains none of the variance in the dependent variable and a value of 1 indicates that the model explains all of the variances in the dependent variable.

2.3 Mean squared error (MSE)

MSE [6] stands for Mean squared error, which is a common metric used to evaluate the performance of a machine learning model. It is calculated as the average squared differences between predicted and actual values. The formula for MSE is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where n is the number of data points, y_i is the actual value of the i-th data point and $\hat{y_i}$ is the predicted value of the i-th data point. MSE is useful because it penalizes large errors more heavily than small errors due to the squaring operation. It is often used in regression problems, where the goal is to predict a continuous numerical value. A lower MSE value indicates better model performance, as the predictions are closer to the actual values.

2.4 Mean absolute error (MAE)

Mean absolute error (MAE) [7] is a statistical metric that measures the average absolute difference between the predicted and actual values in a dataset. It is calculated by taking the mean of the absolute differences between each predicted value and its corresponding actual value. The formula for MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}|$$

Where n is the number of data points, y_i is the actual value of the i-th data point and $\hat{y_i}$ is the predicted value of the i-th data point. The MAE is a commonly used measure of the accuracy of a predictive model and provides a useful estimate of the average magnitude of the errors in the predictions. A lower MAE value indicates a better fit between the predicted and actual values.

2.5 Mean absolute percentage error (MAPE)

Mean absolute percentage error (MAPE) [8] is a metric used in machine learning to evaluate the accuracy of a regression model. It measures the percentage difference between the actual and predicted values of the target variable. The formula for MAPE is:

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n} \frac{|y_i - \hat{y_i}|}{y_i}\right) \times 100\%$$

Where n is the number of data points, y_i is the actual value of the i-th data point and $\hat{y_i}$ is the predicted value of the i-th data point. MAPE is a relative measure of accuracy, meaning that it is scale-invariant and can be used to compare the accuracy of different models regardless of the scale of the target variable. A lower MAPE indicates better accuracy, and a MAPE of 0 indicates perfect accuracy. However, MAPE has a limitation in that it becomes infinite when the actual value is zero.

2.6 Precision

Precision [9] is a term used in statistics and machine learning to measure the accuracy of a classification model. Precision is the ratio of true positives to the model's total number of positive predictions. In other words, precision measures the proportion of correct positive predictions made by the model. For example, a spam filtering model predicts whether an email is spam. In this case, True Positives refer to the number of emails correctly predicted as spam, and False Positives refer to the number of non-spam emails incorrectly predicted as spam.

Precision can be calculated using the following formula:

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

2.7 Recall

The recall metric measures the proportion of actual positive cases in a dataset correctly identified as positive by a classification model expressed as the ratio of true positives to the total number of actual positives in the data set. Recall is an important metric to consider when evaluating a classification model, especially in cases where false negatives can have serious consequences. For example, a false negative could result in a serious untreated condition in medical diagnosis, which can harm the patient. In such cases, a model with high recall is preferred. Recall can be calculated using the following formula:

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negative}}$$

2.8 F_1 score

The F_1 score [10] measures a classification model's accuracy that considers both its precision and recall. It is the harmonic mean of precision and recall, where a perfect score is 1.0, and the worst score is 0.0. The F_1 score is instrumental when the dataset is imbalanced or false positives and negatives have different consequences. It balances precision and recall and is a more reliable measure of a model's overall performance. The F_1 score can be calculated using the following formula:

$$F_1 score = 2 \times \frac{precision \times recall}{precision + recall}$$

Chapter 3

Predicting customer's next interaction

3.1 Introduction

Anticipating when a customer will visit next is crucial for businesses to optimize operations, allocate resources efficiently, and enhance customer satisfaction. Accurately predicting when a customer is expected to visit next enables businesses to prepare for the demand by scheduling staff, ensuring adequate inventory, and targeting marketing campaigns for specific customers. Furthermore, understanding when a customer is likely to visit next allows businesses to customize their services and offerings to meet the customer's needs and preferences, which can foster customer loyalty and drive repeat business.

While the churn rate is a valuable metric for subscription-based products or those with regular interactions like netflix subscriptions, it may not be appropriate for non-regular transactional products since it is challenging to determine which customers have churned and which are dormant. For instance, identifying which customers have churned and their reasons can help businesses prioritize the necessary fixes to retain them.

3.2 Dataset

The model has been built with the help of the Dunnhumby-The Complete Journey data to predict whether a customer will churn. The data covers two years of purchase transactions made by 2500 households and includes demographic information of households and data on campaigns and coupon redemptions. These tables will be merged during modelling to create the final dataset.

- Campaign Descriptions (campaigndesc.csv)
- Campaigns (campaigntable.csv)
- Coupons (coupon.csv)
- Coupon Redemptions (couponredempt.csv)
- Transactions (transactiondata.csv)
- Demographics (hhdemographic.csv)

3.3 Exploratory data analysis

The campaign description data is a reference table that lists the start and end dates of 30 different campaigns, along with their corresponding categories (Type A, B, or C). Table 3.1 describes the sample dataset of campaign description data.

Description	**Campaign**	**Start day**	**End day**
Type B	24	659	719
Type C	15	547	708
Type B	25	659	691
Type C	20	615	685
Type B	23	646	684

Table 3.1: campaigndesc.csv

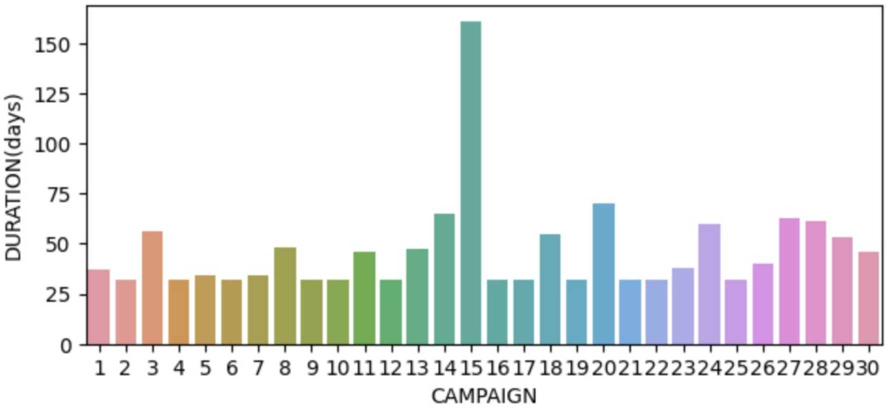

Figure 3.1: Duration of each campaign during the two years

Figure 3.1 show the duration of the 30 campaigns in the two year period. Campaign No 15 had the longest duration of 160 days, which was significantly longer than the other campaigns that ranged from 30 to 70 days. The majority of campaigns lasted for 20 to 30 days. The most common campaigns are depicted in Figure 3.2, and campaigns 18, 13 and 8 are found to be the most frequently occurring campaigns among all campaigns based on the data presented.

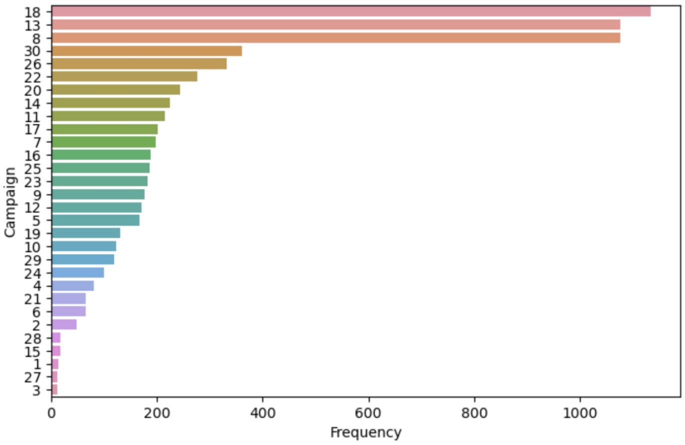

Figure 3.2: Frequency of each campaign

Coupon redemption is a household key-ordered data table that indicates which households redeemed coupon numbers, the redemption date, and the corresponding campaign number. Based on the information in these tables, it was found that only 434 households out of a total of 2500 redeemed coupons during the specified period. The coupons table presents a detailed list of coupons distributed to customers as part of a campaign and the associated products that can be redeemed using each coupon. On the other hand, transactional data contains each household's purchase history, including the product ID, sales value, store ID and other transactional features.

Descriptive analysis of the transactional data:

- The average household purchase amount within the two years is 3223.0
- The average total number of products purchased by a household within two years is 104274.2
- The average number of unique products a household purchases within two years is 560.6
- The average number of store visits made by a household within two years is 90.2
- Figure 3.3 shows the top 15 stores ranked by total sales amount in USD out of 582 stores. The analysis reveals that stores with IDs 367 and 406 generated the highest sales, with each exceeding 200,000 USD.

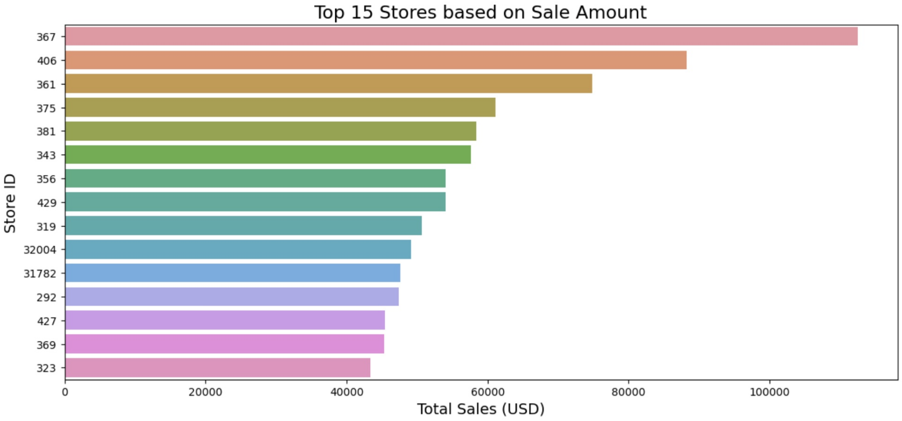

Figure 3.3: Top 15 stores based on sales Amount

- According to figure 3.4, the top customer with ID 1023 made the most purchases among 2500 households, with a total spend of almost 40,000 USD.

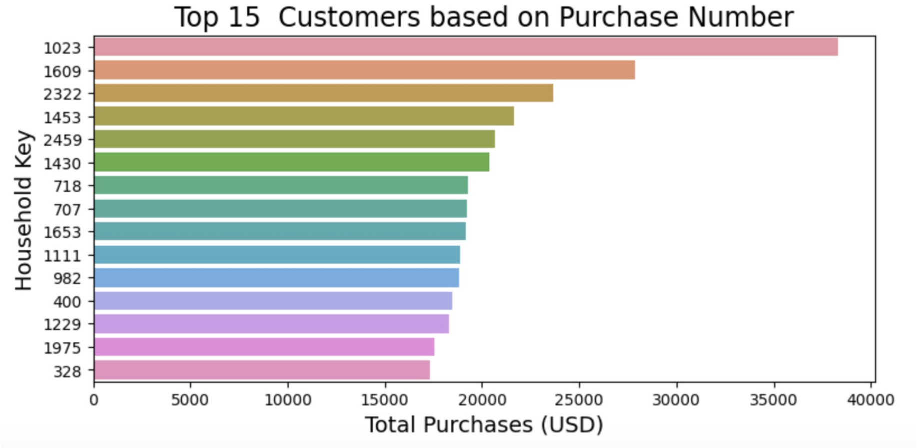

Figure 3.4: Top 15 customers based on purchase number

The demographics data provide information about household demographics, including age groups, marital status, and household size. Since all variables in this dataset are categorical, the categorical pie function will be used to visualize the distributions of the categories. The results presented in figure 3.6 reveal some interesting findings, including the fact that

- Majority of customers fall within the 35-54 age range
- Married couples make up nearly three times as many as singles
- Around 50% of the population has an annual salary ranging from 35K to 74K

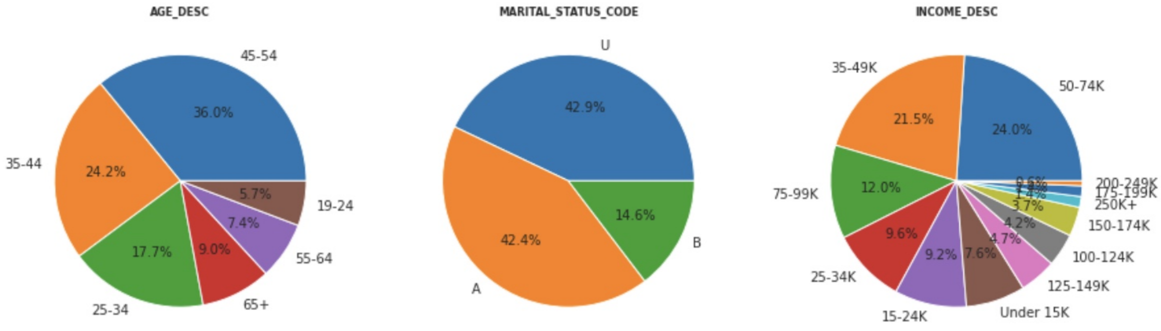

Figure 3.5: Pie chart of household demographics

3.4 Feature engineering

A customer will be classified as having churned if they have not purchased from the store for two weeks or longer. However the dataset does not provide any information on the churn variable. A target variable is created by adding a feature determining whether customers have churned.

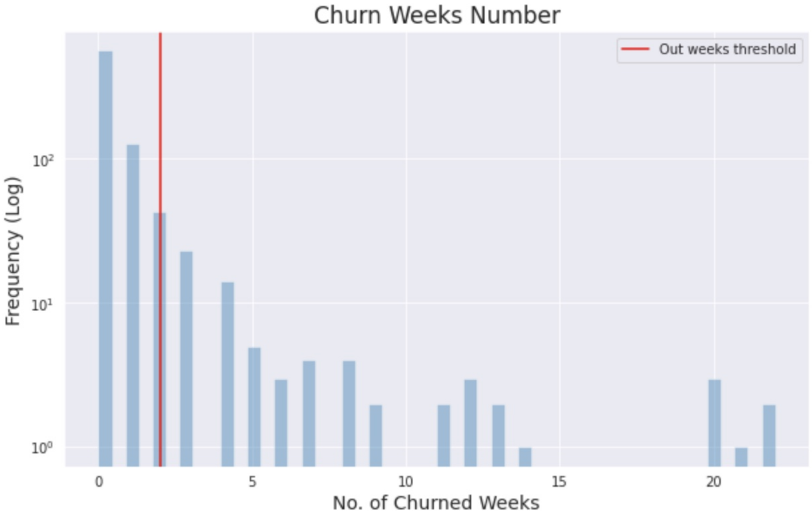

Figure 3.6: Number of weeks since last purchase

Households split with the defined churn:

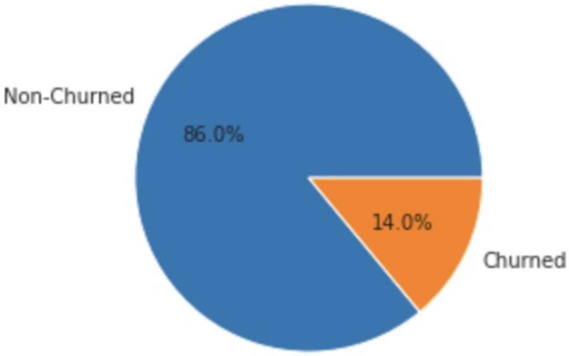

Figure 3.7: Churn variable distribution

The following features can be derived from the "Campaign Table", "Transaction Data" and "Coupon Redemption" tables to develop a churn prediction model:

- List of campaigns received by each household
- Total number of received campaigns per household
- List of campaigns that resulted in coupon redemption
- The number of coupon redemptions made by each household
- Most Frequent Campaign Type received by each household
- The total purchase amount made by a household within two years

Insights from data:

Based on Figure 3.8, it can be observed that the age group of 55-64 has a lower churn rate compared to other age groups, and there is no clear trend of increasing or decreasing churn rates with age. This indicates that age alone may not be a robust predictor of churn. Factors like campaign effectiveness, purchase behaviour, and demographic information may need to be considered in the churn prediction model.

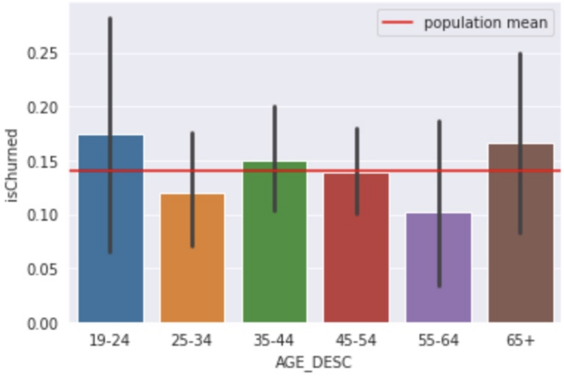

Figure 3.8: Age vs Churn Rate

According to Figure 3.9, married couples (Group A) exhibit a higher churn rate as compared to singles (Group B), while households with unknown marital status (Group U) contribute to an overall increase in the population's churn rate.

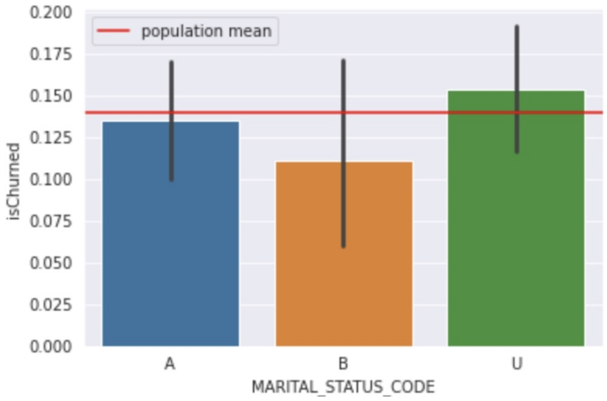

Figure 3.9: Marital Status vs Churn Rate

Figure 3.10 shows no clear trend of increasing or decreasing churn rates with income groups, but households with an income of 175K and above have a zero churn rate.

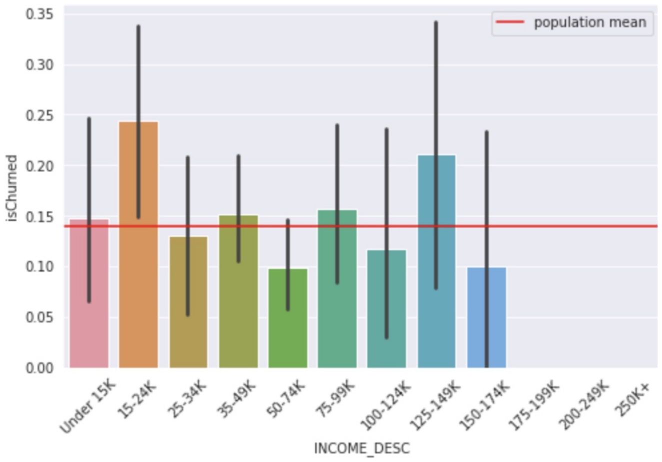

Figure 3.10: Income vs Churn Rate

3.5 Building a classification model

The categorical variables were transformed using a one-hot encoding algorithm [11] to prepare the data for the classifier. A train-test split of the dataset was performed, with

75% of the data used for training and 25% used for evaluation. Machine learning classifier models were then build using the training set, and their accuracy was evaluated using the test data. Table 3.2 below shows the results from various classifier models.

Model	Accuracy
Logistic Regression	0.80
Decision tree	0.82
Random Forest	0.84
Support vector machine	0.83

Table 3.2: Accuracy of the classification model

The dataset's imbalanced nature was considered when building the classification model for this study. The XGBoost [12] modelling technique was employed to address this issue, which is well-suited for handling imbalanced data and can help improve the model's accuracy and robustness. Upon applying this approach, the resulting model demonstrated high performance with an F_1 Score of 0.91 and an accuracy of 0.84. These metrics indicate that the model effectively predicts new data classification and can be a valuable tool for decision-making. Overall, the successful application of the XGBoost modelling technique to this study highlights its potential to address the challenges posed by imbalanced datasets and improve the accuracy and robustness of classification models. Table 3.4 shows the classification report on the test dataset.

	precision	recall	f1-score	support
false	0.86	0.97	0.91	172
true	0.25	0.07	0.11	29
accuracy			0.84	201
macro avg	0.56	0.52	0.51	201
weighted avg	0.77	0.84	0.79	201

Table 3.3: Classification report on the test dataset

Chapter 4

Recommendations for customer purchases

4.1 Introduction

Recommendations [13] play an essential role in any industry, especially in today's world, where consumers can access vast information and options. Here are some reasons why recommending products is essential in the industry.

Helps consumers make informed decisions: Recommending products can help consumers decide which product best suits their needs. By providing relevant information and insights about the product, consumers can make well-informed decisions that save them time and money.

Increases customer satisfaction: Recommending the right product to customers can increase their satisfaction with the product and the company. Customers who are satisfied with their experience are more likely to make future purchases and recommend the company to others.

Boosts sales and revenue: Recommending products to customers can increase sales and revenue for the company. When customers are provided with a recommendation, they are more likely to make a purchase, leading to increased sales and revenue for the company.

Builds trust and loyalty: Providing recommendations can help build trust and loyalty between the customer and the company. When customers feel that the company has their best interests at heart, they are more likely to trust the company and become loyal customers.

Overall, recommending products is important in the industry as it helps consumers make informed decisions, increases customer satisfaction, boosts sales and revenue, and builds trust and loyalty between the customer and the company.

4.2 Dataset and feature engineering

Used the "Transaction data" for recommending a product to a customer. This dataset contains all purchase history.

Hous-ehold key	Basket ID	Day	Pro-duct ID	Qu-an-tity	Sales value	Store ID	Ret-ail disc	Tra-ns time	W-eek no	Cou-pon disc
1	27601281298	51	825123	1	3.99	436	0.00	1456	8	0.0
2	27601281345	67	831447	2	2.99	437	0.00	1230	6	0.25
3	27601281567	81	840361	2	1.09	434	-0.30	1432	5	0.35
4	27601281345	61	845307	3	3.71	456	-0.62	1341	5	0.15
5	27601281299	34	852014	1	2.79	489	-1.20	1467	6	1

Table 4.1: Transaction data

Based on the data in table 4.1, various features can be extracted to recommend products to a customer.

- Gathers all the products a customer purchased in their previous transactions. This will give us an idea of the customer's buying habits and preferences.
- List the frequency of each item purchased by the customer. This will help us identify the items the customer buys frequently and those they don't.

- Top-selling product based on the number of times all customers have purchased it in the transaction data.
- Created a data frame that contains the items purchased by the customer in their previous transactions, the items purchased in the current transaction, and the number of times each item has been purchased. This will help us identify the items the customer will likely purchase based on their past buying behaviour.

4.3 Procedure to recommend products

First, the product IDs purchased by each customer are listed, and the frequency of each product ID is counted. The least frequently purchased products, corresponding to the 20% tail, are then removed from the dataset. Then, pairs of products are created from the (d-1)th day and the dth day for each customer in all transactions, and the occurrence of each pair is counted. These pairs are then aggregated across all customers. The less frequently occurring pairs, corresponding to the 20% tail pairs, are then removed from the association data. Pairs of 5 items are then listed for each product purchased on the previous transaction day. Finally, based on the occurrence of these pairs, the top 5 recommended items are suggested on the current transaction day.

4.4 Result

To ensure the accuracy of the recommendations, the top 5 items are validated against the products in the current transaction. This helps avoid recommending products the customer has already purchased in their current transaction. After validating the top 5 items against the products in the current transaction, a histogram of the average item overlaps can be made. This histogram shows how many of the top 5 recommended items overlap with the items the customer purchased in their current transaction. Analyzing this histogram can provide insights into the effectiveness of the recommendation system. A high overlap between the recommended and purchased items in the current transaction suggests that the recommendation system accurately captures the customer's preferences and needs.

Overall, this procedure provides a way to analyze transaction data and make personalized product recommendations to customers without using a traditional recommendation system. Using associations between frequently purchased products, it is possible to suggest relevant products to customers and improve their overall shopping experience. After validating the top 5 items against the products in the current transaction, a histogram of the average item overlaps and a scatter plot between the number of transactions set by each customer and the average item overlap are shown in figures 4.1 and 4.2, respectively.

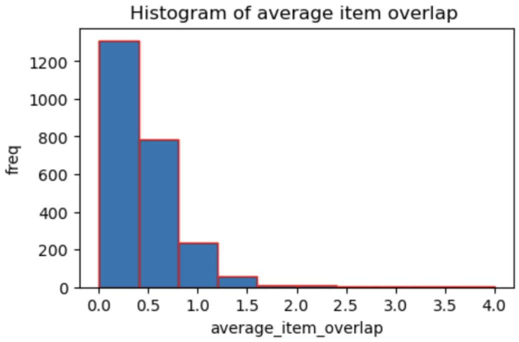

Figure 4.1: Histogram of the average number of item overlap

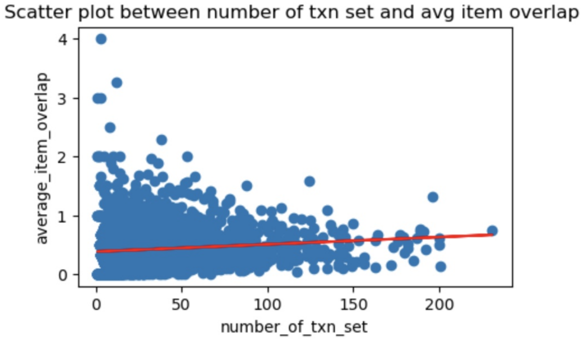

Figure 4.2: Scatter plot between the number of transactions and average item overlap

Chapter 5

Predicting coupon redemption in customer's next transaction

5.1 Introduction

The likelihood of customers redeeming coupons in their next transactions is a crucial question for marketers and businesses. This information can help companies avoid giving offers to customers who are not likely to redeem and instead focus on customers who are more likely to redeem. The analysis of coupon redemption in the next transaction is a challenging task that requires the integration of various data sources and analytical techniques. Many factors influence a customer's decision to redeem a coupon in their next purchase, such as the discount amount, the product category, and the customer's past purchasing behaviour. One of the main challenges in predicting coupon redemption is the presence of unobserved factors that influence customer behaviour, such as personal preferences and external events. Accurate prediction of coupon redemption can have significant implications for businesses, including increased customer retention and loyalty, improved targeting of marketing campaigns, and improved revenue generation. The results of the simulation will aid businesses in identifying the most effective marketing strategies and optimizing their efforts to boost customer retention and sales.

5.2 Feature engineering

The classification model was built using the ”Transaction Data” from the ”Dunnhumby dataset”. Various features were extracted from this dataset to train the model. These features included transaction day, previous transaction day, days since the last transaction, the average number of days between transactions, future transaction day, number of coupon redemptions, coupon redemption percentage in the previous transaction, transaction days to coming transaction day, number of products, the average number of products, transaction value, coupon discounts percentage (CDP) and average transaction value.

Household key	**Txn day**	**Is coupon redem**	**Prev txn day**	**days since last txn**	**Avg no of days between txn**	**Future txn day**	**txn day to future txn day**	**No of product**	**Avg no of product**	**Txn value**	**CDP**	**Avg txn value**	**Sensitivity**
1	51	0	0	0	0	67	16	34	34	78	0	78	insensitive
1	67	0	51	16	8	88	21	14	24	41	0	59	insensitive
1	88	0	67	21	12	94	6	13	20	26	0	48	sensitive
1	94	1	88	6	10	101	7	32	23	63	1	52	insensitive
1	101	0	94	7	10	108	7	20	22	53	0	52	insensitive

Table 5.1: Extracted features from transaction data

Furthermore, the target variable was determined to be sensitive or insensitive depending on whether the coupon was redeemed in the next transaction. This classification model was designed to predict whether a customer will redeem his coupon in the next transaction and thus determine his level of sensitivity.

To accomplish this task, a machine learning algorithm was employed. The model was trained using historical transaction data. The trained model predicts the sensitivity of future transactions. By accurately predicting customer behaviour, businesses can optimize their marketing strategies, offer personalized promotions, and improve customer satisfaction. The demographic variables were merged with the new data extracted from ”Transaction Data” to analyze the relationship between the demographic variables and the target variable. These demographic variables included age, marital status, income, and other relevant information about households. The analysis was designed to identify patterns or relations between these demographic and target variables.

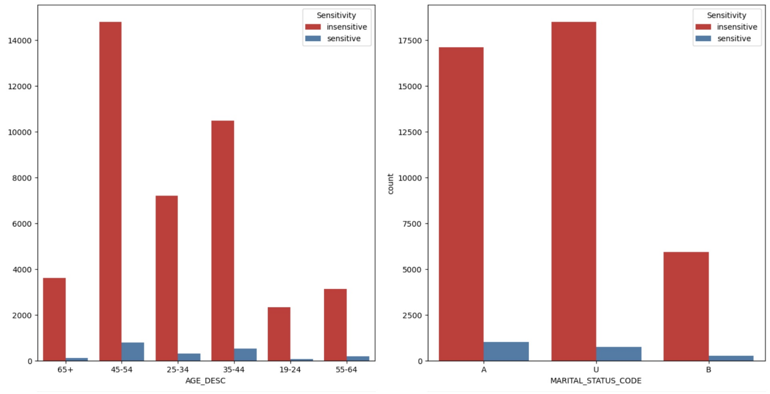

Figure 5.1: Relation between age and marital status with the target variable

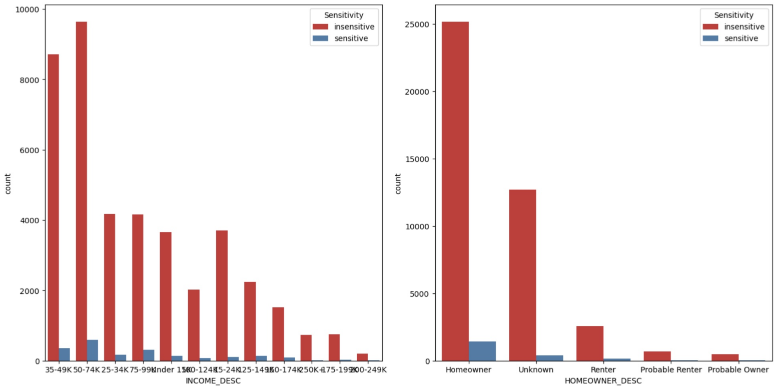

Figure 5.2: Relation between income and homeowner with the target variable

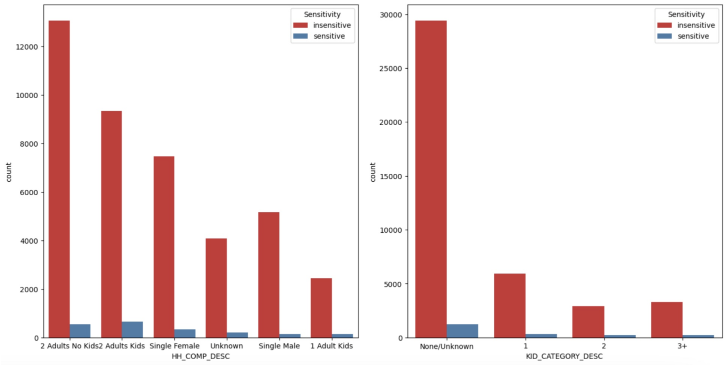

Figure 5.3: Relation between households description and kid category with target variable

5.3 Result

The categorical variables in the data set were converted to numerical variables using the get-dummies technique to prepare them for the model. A correlation matrix was generated and examined to determine the correlation between attributes to ensure that there are no variables with similar information. Correlation refers to the relationship between two variables and how they move together. Plotting the correlation matrix allows us to visualize the relationship strength between each pair of variables.

Based on the correlation matrix, it was observed that the average number of products and the average transaction value have a positive correlation. However, the average number of products was removed as a feature for modelling purposes. The independent variables used for the model were:

- Average number of days between transactions
- Number of coupon redemptions
- Coupon redemption percentage
- CDP (coupon discount percentage)
- Average transaction value.
- Age description

- Income description

	1	2	3	4	5	6	7	8
1	1	0.61	0.019	0.077	0.065	0.12	0.03	0.097
2	0.61	1	0.0092	0.023	0.11	0.0.067	0.0066	0.099
3	0.019	0.0092	1	0.14	0.15	0.13	0.46	0.22
4	0.077	0.023	0.14	1	0.39	0.83	0.15	0.5
5	0.065	0.11	0.15	0.39	1	0.3	0.062	0.47
6	0.12	0.067	0.13	0.83	0.3	1	0.18	0.6
7	0.03	0.0066	0.46	0.15	0.0642	0.18	1	0.11
8	0.097	0.099	0.22	0.5	0.47	0.6	0.11	1

Table 5.2: Correlation matrix

1 - days since the last transaction
2 - average number of days between transactions
3 - coupon redemption percentage
4 - number of products
5 - average number of products
6 - transaction value
7 - coupon discount percentage
8 - average transaction value

The dataset was split into training and testing data with a ratio of 80/20. For this classification model, several algorithms were considered, such as logistic regression [14], decision trees [15], random forest [16] and boosting techniques [17]. Overall, the successful application of this model suggests that it can be a valuable tool in accurately classifying new data and identifying patterns or trends in the data that may be useful in making informed decisions. Table 5.3 shows the results of building classifier models.

Model	**Accuracy**
Logistic Regression	75
Decision tree	77
Random Forest	78.3
Support vector machine	77
XGBoost	81.4

Table 5.3: Accuracy table of the classifier model

Chapter 6

Analysis of discounted customer transactions

6.1 Introduction

Tracking revenue from customers after offering discounts is essential in many industries because it allows businesses to measure the effectiveness of their discount strategies and assess their impact on overall revenue and profitability. Discounts often attract new customers, increase customer loyalty, and stimulate sales during slow periods. However, if discounts are not appropriately managed, they can eat into a business's profit margins and have a negative impact on the bottom line. By tracking the revenue of customers after discounts, businesses can better understand the impact of their discounting strategy on overall revenue and profitability. This information can be used to optimize discount strategies and ensure that discounts are offered to maximize revenue and profitability.

Furthermore, tracking revenue from customers after discounts can also help businesses identify any potential issues with their discounting strategy. For example, if a particular discount is not generating enough revenue or if customers are not taking advantage of a certain discount, it may be necessary to adjust the discounting strategy. This information can help businesses make data-driven decisions to optimize their discounting strategies and increase revenue.

In general, tracking customer revenue after discounts is a crucial aspect of managing a successful discount strategy and can provide valuable insights to improve overall revenue and profitability.

6.2 Dataset

The dataset mentioned in table 5.1 was used in this chapter to predict revenue after offering discounts. The target variable was fixed as the next transaction value. Based on the scatter plot analysis shown in figure 6.1, it can be observed that there is no significant correlation between the number of product features and the value of the next transaction. This suggests that the number of product features may not be an important predictor of the target variable.

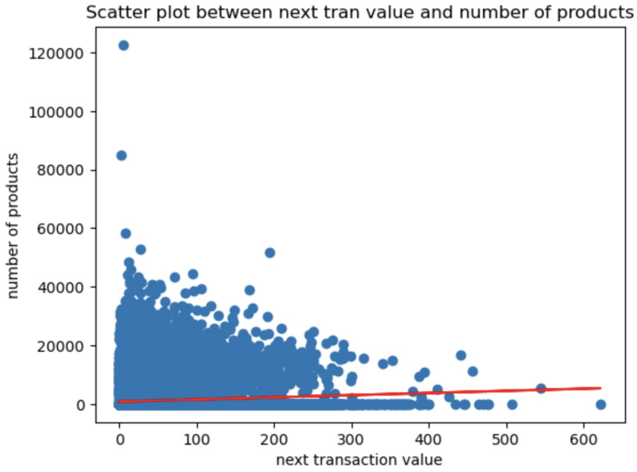

Figure 6.1: Scatter plot between next transaction value and number of products purchased in that transaction

6.3 Regression model building

The dataset was divided into two sets, one for training and the other for testing, to evaluate the performance of the model. A histogram was plotted for the target variable in both sets. The shape of the distribution of the target variable in figure 6.2 appeared to be logarithmic normal.

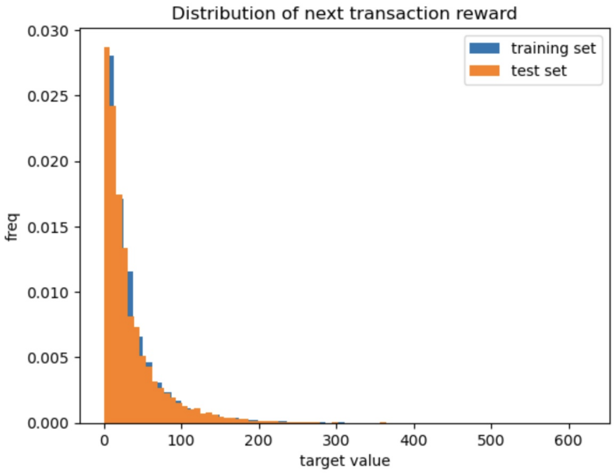

Figure 6.2: Distribution of next transaction reward

The target variable was transformed using a logarithmic function in order to achieve a more normal distribution, as shown in figure 6.3.

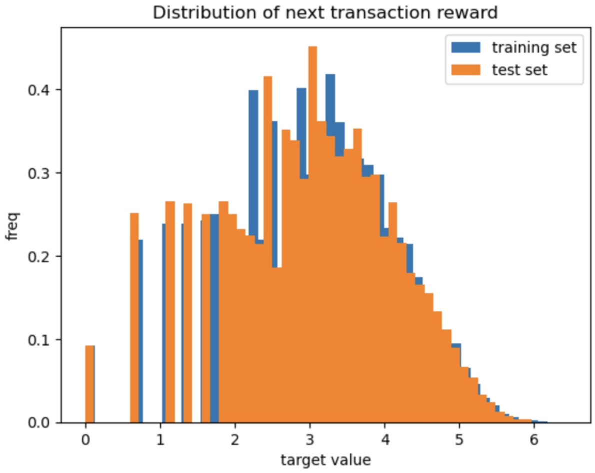

Figure 6.3: Distribution of next transaction reward

The accuracy of the multiple regression model [18] used to predict revenue was found to be unsatisfactory, as indicated by the low R^2 value of 0.24. To improve the performance of the model, backward and forward feature selection processes were applied to identify the most relevant features for the model. However, the R^2 value did not improve significantly after applying these techniques.

Different regression algorithms were experimented with to explore various modeling approaches. These algorithms included DecisionTreeRegressor [19], Random Forest Regressor [20], and Generalized Linear Model [21]. However, even after trying these models, the R^2 value remained low and did not show much improvement. Table 6.1 below shows the results from building regression models.

Model	R^2 **value**
Multiple Regression Model	0.24
DecisionTreeRegressor	0.27
Random Forest Regressor	0.27
generalized linear model	0.29

Table 6.1: R^2 value from building regression model

Since R^2 is very less for all regressor models, a statistical significance method was developed by creating a data frame that contains each household's total transaction value using the "Transaction Data". Also, calculate the coefficient of variation, which is a dimensionless measure of dispersion that can be used to compare the variability of datasets with different units or scales.

The formula for calculating the coefficient of variation is:

$$CV = \frac{\text{standard deviation}}{mean} \times 100\%$$

"standard deviation" measures the data spread around the mean, and the "mean" is the average of the data set. The result is expressed as a percentage.

The coefficient of variation is often used in scientific research and quality control to compare the interpretation of different datasets or to evaluate measurement reliability. A lower coefficient of variation indicates that the data points are closer to the mean, while a higher coefficient of variation indicates that the data points are more widely dispersed. A histogram of the standard deviation divided by the mean was plotted to investigate the distribution of the data.

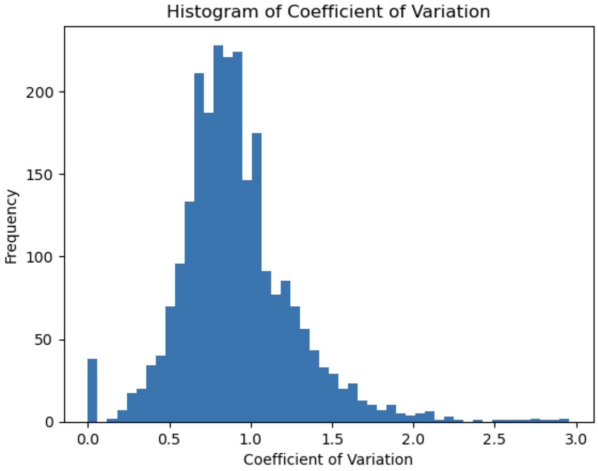

Figure 6.4: Histogram of Coefficient of Variation

Based on the histogram, it is evident that the coefficient of variation (CV) is greater than 1 for most of the historical data for a customer. In statistical analysis, a CV greater than 1 indicates that the data have a high degree of dispersion and are not well suited for predictive modeling. Therefore, exploring alternative modelling techniques better suited to handling such data may be necessary.

In the analysis of the revenue data, it was observed that more accurate and stable results could be obtained by converting the revenue data to a rolling 7-day average. By taking a rolling average of the revenue data, the resulting values reflect a more comprehensive representation of the underlying trends in the data rather than focusing on individual day-to-day fluctuations. To apply this approach, all relevant features were converted to rolling features using a rolling window of 7 days. This approach effectively smoothed out the noise in the data and helped identify more meaningful trends that could be used to predict future revenue levels. A multiple linear regression model was applied to rolling 7-day average revenue data to predict revenue for unknown data points. The obtained value of R^2 is 0.87 indicated a strong fit between the predicted and actual revenue values. Based on these results, the rolling 7-day revenue model provides a reliable and effective approach to predict revenue for future periods. Taking into account a longer-term rolling average, this approach can capture a more comprehensive representation of the underlying trends in the data, which can help produce more accurate and reliable revenue predictions.

Chapter 7

Customer behaviour simulator

7.1 Introduction

Reinforcement learning(RL) methods are being increasingly used in marketing, to give offers to customers [22]. While traditional Machine Learning methods are known to focus on the immediate outcome, Reinforcement Learning considers long-term rewards, to make decisions. This long term is a time period over which multiple interactions of the customer are seen. RL methods learn by simulating different scenarios to make an assessment of the best action in a given scenario. Historical data might have a few customer-offer combinations for a set of customers and offers due to selection bias. But RL requires all customer-offer combinations to be seen, to learn. Hence, customer simulators, i.e., machine learning models that predict customer responses to unseen offers, are required. Deep learning models for multi-output regression [23] are utilized, which refer to neural network models with multiple output nodes representing different output variables. This type of neural network model learns to predict multiple output variables simultaneously.

The architecture of a multi-output regression neural network typically consists of an input layer, one or more hidden layers, and an output layer. The output layer can have any number of nodes, depending on the number of output variables. Figure 7.1 depicts a neural network comprising an input layer, two hidden layers, and an output layer consisting of four nodes. The input layer receives the input data, and the hidden layers process it and learn to represent it more abstractly. Finally, the output layer with four nodes predicts the values of four output variables.

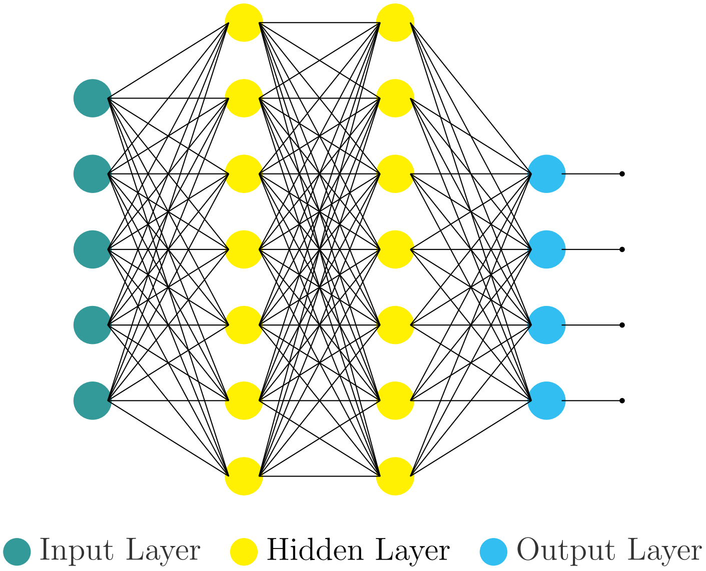

Figure 7.1: Multi output neural network model [24]

During training, the network learns to predict the values for all the output variables simultaneously. The loss function used for training is a multi-output version of the mean squared error (MSE) loss, which penalizes the difference between the predicted and actual values for all the output variables. The customer simulator environment was used to train reinforcement learning (RL) models [25] by wrapping using TensorFlow. The RL models learned the optimal policy based on the historical dataset, and from this optimal policy, the best action for unknown data in the test dataset was determined. Additionally, the customer simulator was used to predict the reward value and compare it to the actual reward value of the test data to evaluate the RL model's performance.

Reinforcement learning(RL)

Reinforcement learning(RL) is a machine learning approach that enables an agent to learn through trial and error in an interactive environment. The agent receives feedback from its actions and experiences to improve its decision-making capabilities. In supervised learning, feedback given to the agent is the correct action to perform a task. In reinforcement learning, feedback is in the form of rewards and punishments that signal positive and negative behaviour. Reinforcement learning and unsupervised learning differ in their goals, where unsupervised learning aims to identify similarities and differences between data points, and reinforcement learning aims to develop an action model that maximizes the cumulative reward for the agent. The action-reward feedback loop of a typical RL model is illustrated in figure 7.2 below.

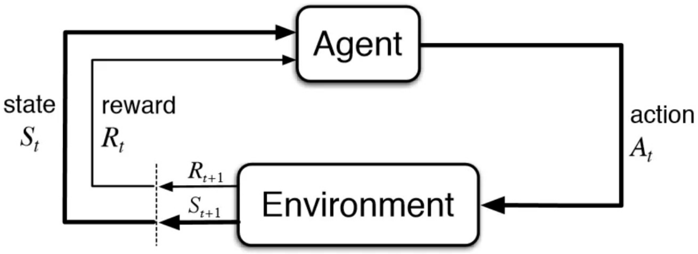

Figure 7.2: Action reward feedback loop [26]

Reinforcement learning (RL) involves several key terms that describe the basic elements of a problem. The agent operates in a physical world called the environment, and its current situation in the environment is referred to as its state. After agents take action in its state, the agent receives feedback from the environment in the form of a reward. The policy is a method for mapping the agent's state to actions, and the value refers to the future reward that the agent would receive by taking a particular action in a given state.

Example:

In the game PacMan, the agent's goal is to maximize its cumulative reward by taking actions in the environment. The environment is the game board or the grid world, the state is the current position of the agent and the location of the food and ghosts, the reward is positive for eating food and negative for getting caught by a ghost, and the policy is the method by which the agent decides which action to take in a particular state. The value is the expected future reward an agent would receive by taking an action in a particular state. By learning from its experience and adjusting its policy, the agent can improve its performance and win more games.

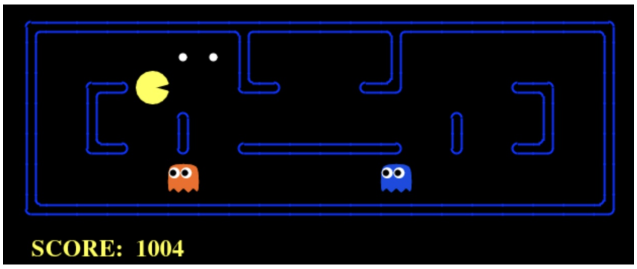

Figure 7.3: How RL plays PacMan game [27]

When building an optimal policy, the agent must balance exploring new states and exploiting current knowledge to maximize the overall reward. This is known as the exploration vs. exploitation trade-off. To balance both, the agent may need to make short-term sacrifices to gather enough information to make better decisions in the future.

Markov decision process(MDP):

Markov decision process (MDP) is a mathematical framework used to model reinforcement learning problems. If the environment is fully observable, it can be modeled as a Markov process. In an MDP, an agent interacts with the environment by taking actions, and the environment responds by generating a new state. This process is repeated at each time step, and the agent learns from the rewards received in each state to improve its decision-making policy.

MDP is used to describe the environment for the RL, and almost all the RL problem can be formalized using MDP. MDPs provide a flexible and powerful framework for modeling sequential decision-making problems and training RL agents to make optimal decisions in uncertain environments. MDP contains a tuple of four elements (S, A, P_a, R_a):

- A set of finite States S
- A set of finite Actions A
- Rewards received after transitioning from state S to state S', due to action a.
- Probability P_a.

Q-learning algorithm:

The Q learning algorithm [28] is a model-free iterative approach to solving the reinforcement learning problem. The objective of the Q-learning algorithm is to learn a policy that maximizes the expected cumulative reward over time. The algorithm achieves this by iteratively updating the Q-values, representing the expected cumulative reward for taking a particular action in a particular state. The Q-learning algorithm initializes the Q-values to a small value close to zero. Then, each iteration chooses an action in the current state based on the best Q-value, performs the action, and observes the new state. The algorithm measures the reward from the action and uses the Bellman equation to update the Q-values. The Bellman equation represents the relationship between the current Q value, the immediate reward, and the discounted future reward.

The Q-learning algorithm continues to update the Q values until convergence. At convergence, the Q table represent the optimal policy for the given environment, which maximizes the expected cumulative reward over time. The idea here is to update the Q value using Bellman equation as shown below:

$$Q_{new}(s, a) = Q(s, a) + \alpha[R(s, a) + \gamma \times maxQ'(s', a') - Q(s, a)]$$

where $Q(s, a)$ is the current Q value, α is the learning rate, $R(s, a)$ is the reward of taking action a in state s, γ is the discount rate and $maxQ'(s', a')$ is maximum possible Q value in the next state s', for all possible actions a' in that state.

Deep Q-network(DQN):

Deep Q-Network (DQN) is a reinforcement learning algorithm that uses a neural network to approximate the Q-function, which is used to determine the optimal action for a given state in a markov decision process. DQN uses a neural network to approximate the Q-value function, which maps a state-action pair to the expected reward. During training, the DQN is fed with experience tuples consisting of the current state, action, reward and next state. These tuples are stored in a replay buffer and are used to train the DQN using stochastic gradient descent.

One of the key innovations in the DQN algorithm is the use of a target network to stabilize the training process. The target network is a separate neural network used to estimate the Q-values for the next state during training. The target network is updated periodically, typically every few thousand steps, using the weights from the main DQN network. This helps to reduce the correlation between the target and predicted Q-values, which can lead to instability during training.

7.2 Multi output neural network model for customer simulation

Defined state and action:

The historical customer data in table 7.1 was used to create a multi-output neural network model with four features defining the current state and actions. Features that define the current state are the average number of products purchased in the last seven transactions, the average number of coupons redeemed in the last seven transactions, the average transaction value in the last seven transactions, and the coupon redemption percentage in the last seven transactions, while the action represents the coupon discount percentage(CDP).

number of product	**number of coupon redem**	**txn value**	**coupon redemption percentage**	**CDP**	**reward**
34.00	0.00	78.0	0.00	0	72.5
24.00	0.0	59.5	0.00	0	55.62
20.33	0.00	48.33	0.00	1	46.54
23.25	0.25	52.0	0.25	0	43.33
22.60	0.20	52.20	0.20	2	47.33

Table 7.1: Sample dataset

Given an unknown coupon discount percentage(CDP) as an action, the model will predict the next state and reward based on the current state. The next state will contain the updated values of the four features that constitute the state. The reward will be the average of three previous transaction values. The multi-output neural network model will take the current state and action as inputs and output the predicted next state and reward. The model will use historical customer data to learn patterns and relationships between the current state, action, next state and reward. It will use this information to make accurate predictions for unknown coupon discount percentages. One of the features, the average number of products purchased in the last seven transactions, had an outlier in the current state. To address this issue in the modelling process, the outlier was removed and the data was standardized by subtracting the mean value and dividing it by the standard deviation. Table 7.2 provides a description of the dataset after removing outliers.

	number of product	**number of coupon redem**	**txn value**	**coupon redemption percentage**	**CDP**
count	58955.00	58955.00	58955.00	58955.00	58955.00
mean	46.54	0.02	28.39	0.01	0.017
std	149.02	0.09	19.20	0.056	0.13
min	0	0	0	0	0
25%	7.14	0	14	0	0
50%	12.14	0	24	0	0
75%	19.71	0	38.29	0	0
max	1001.43	0.86	107	0.29	1

Table 7.2: Description of the dataset after removing outliers

Building multi-output neural network model:

The data was split into training, testing, and validation sets in a 75/15/15 ratio to accomplish this. The neural network architecture was designed with two hidden layers, one with eight nodes and the other with six, an input layer with five nodes, and an output layer with four nodes. The rectified linear unit (ReLU) activation function was utilized in the model as it is known for its simplicity and effectiveness in deep learning models. The model was trained using the training dataset, and its performance was evaluated on the validation set. Adam Optimizer was used to iteratively update the model weights during training. A learning rate of 0.01 was set to ensure that the model converges quickly without overshooting the optimal solution. The mean absolute error (MAE) was chosen as the loss function to measure the difference between the predicted and actual values. Table 7.3 below shows the summary of the neural network model that was used.

Model: ”sequential”		
Layer (type)	**Output Shape**	**Param #**
dense (Dense)	(None, 8)	48
dense_1 (Dense)	(None, 6)	54
dense_2 (Dense)	(None, 4)	28
Total params: 130 **Trainable params:** 130 **Non-trainable params:** 0		

Table 7.3: Neural network model summary

The mean absolute error (MAE) was calculated on the test data using predicted values, resulting in MAE values of $[0.14, 0.23, 0.06, 0.05]$ for the four output variables. The accuracy of the model was found to be 87%. The figure in 7.4 shows a decreasing trend in the validation error versus the training error as the number of epochs increases, indicating that the model has successfully learned to generalize to new data.

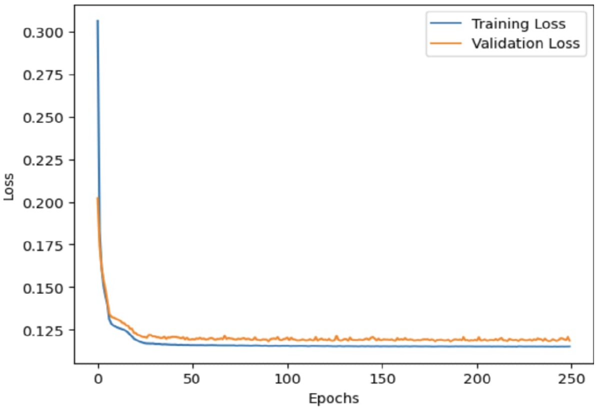

Figure 7.4: Graph of training loss and validation loss

Using a previously trained model, the next state of a customer can be predicted after taking an unknown action in the current state. Another neural network model with one hidden layer and linear activation function was created to calculate the reward value based on this predicted state. The input layer of this model contains four nodes, representing the predicted current state as input features, and the output layer has one node, representing the reward value. The value of R^2 of this model is 0.56, indicating a moderate level of accuracy in predicting the reward value based on the predicted current state of the customer.

The simulator model is ready to train reinforcement learning agents. A histogram plot was created to verify the original reward values versus the predicted reward values generated by the simulator model. Figure 7.5 showed that the two reward values were close, indicating that the simulator model accurately predicted the reward values. This verifies the reliability of the simulator model for training reinforcement learning agents.

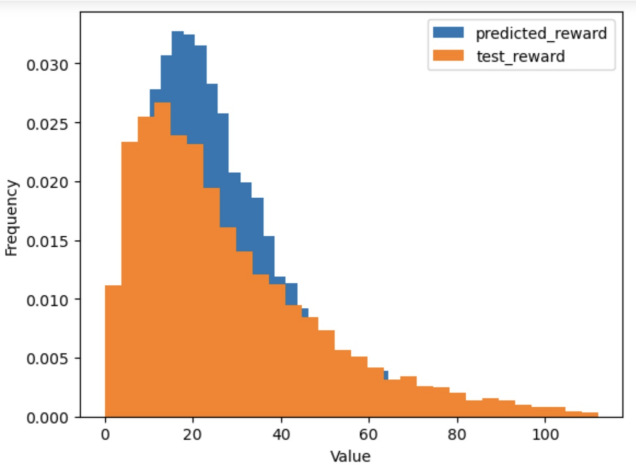

Figure 7.5: Original reward values versus the predicted reward values generated by the simulator model

7.3 Training reinforcement learning

Reinforcement learning (RL) aims to develop intelligent agents that can improve their decision-making skills by actively engaging with an environment. In a typical RL scenario, the agent receives environmental observations and takes the corresponding actions. These actions trigger environmental changes, resulting in rewards and new observations. The agent then learns to optimize its policy by selecting actions that maximize the total rewards accumulated over time, also referred to as the return. Using TensorFlow agents [29], created a custom Python environment for RL.

Python environment:

The Python environments used in reinforcement learning typically include a step(action) → next_time_step method that applies the agent's chosen action to the environment and provides information about the subsequent time step. This information is typically presented in the form of a named tuple called

- **Observation:** The observation refers to the portion of the environment state available to the agent to perceive and use to make decisions about its actions in subsequent steps.
- **Reward:** The rewards obtained by the agent across multiple steps are accumulated to calculate the agent's overall performance. The agent's objective is to learn to maximize the total sum of these rewards.
- **Step_type:** When interacting with an environment, the actions taken by an agent are typically part of a sequence of steps or episodes.
- **Discounts:** This refers to a floating-point value that indicates the weighting or importance given to the reward obtained in the next time step compared to the reward obtained in the current time step.

These are grouped into tuple TimeStep(step_type, reward, discount, observation). Environments have a reset() method that allows starting a new sequence or episode and provides the initial TimeStep. It is not mandatory to explicitly call the reset() method, as environments usually reset automatically at the end of an episode or when the step() method is called for the first time in a new episode.

Following the Python environment creation procedure, a Python environment was created to simulate customer behaviour. Subsequently, an environment wrapper was implemented to take the Python environment as input and return a TensorFlow version of that environment. The following parameters were set to train a deep Q learning model to learn customer behaviour.

- Num_iterations: The number of training iterations to perform, set to 10000.
- Initial_collect_steps: The number of steps to take to fill the replay buffer is initially set to 1000.

- Collect_steps_per_iteration: The number of steps per iteration to add to the replay buffer, set to 1.

- Replay_buffer_capacity: The maximum number of steps to store in the replay buffer, set to 10000.

- Fc_layer_params: A tuple containing the number of neurons in each fully connected neural network layer used in the Q-learning algorithm, set to (10,5).

- Batch_size: The number of samples to use in each training batch, set to 128.

- Learning_rate: The learning rate for the optimizer used in training the Q-learning algorithm, set to 10^-4.

- Log_interval: The interval to log the training progress is set to 200.

- Num_eval_episodes: The number of episodes to evaluate the model during training, set to 2.

- Eval_interval: The interval to evaluate the model during training, set to 1000.

During training, the RL agent stores its experiences in a replay buffer, and the loss function vs the number of episodes can be plotted to track the model's progress. The resulting figure 7.6 shows that the loss function decreases significantly after a certain number of episodes, indicating that the model is learning well from the new dataset.

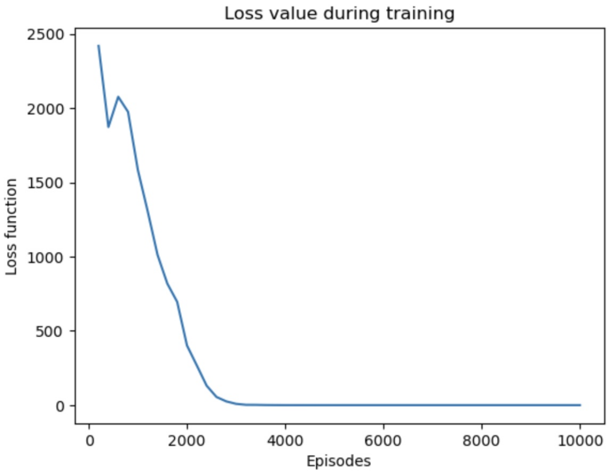

Figure 7.6: Loss function during training

After obtaining the optimal policy, the best action for the customer was determined based on the current state of the test data (unknown data). The previous neural network model was then used to find the reward value for the best action. A histogram, Figure 7.7, compares the original reward values with the predicted reward values of the reinforcement learning algorithm. The figure shows that the reward value increased with the RL prediction. Additionally, there were no high values that were significantly different from the original reward values. This suggests that the RL algorithm successfully predicted the reward values for the best actions.

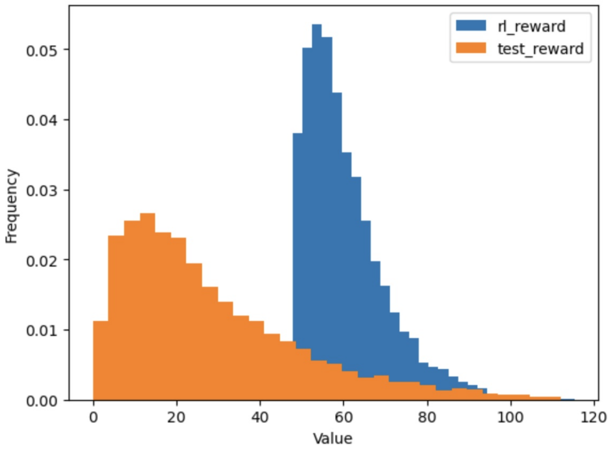

Figure 7.7: Histogram of test reward with RL predicted reward

Chapter 8

Conclusions

The customer behaviour simulator model can be helpful for companies optimising their marketing strategies. By using the model to simulate customer behaviour, companies can test different marketing strategies and see how they would impact customer behaviour before actually implementing them. For example, a company could simulate the effect of offering a discount coupon to a specific set of customers and see if this leads to increased sales or customer loyalty. In addition, the customer behaviour simulator model can also be used to improve customer retention. Furthermore, the model can help companies increase their revenue by predicting which customers are most likely to purchase or respond to specific marketing strategies.

The customer behaviour simulator model has demonstrated its effectiveness in training reinforcement learning agents. Analyzing the validation error and the predicted versus actual reward values confirmed that the simulator model was reliable and accurate. The policy trained by Q learning led to better agent decision-making. As a result of the training process, a reasonably high reward was achieved, indicating the model's success in predicting and influencing customer behaviour. This model has tremendous potential for real-world applications, such as optimizing marketing strategies, improving customer retention and increasing revenue.

Future work:

While the customer behaviour simulator model has shown promising results, there is always room for improvement and future work. One avenue for further research and development is to explore using more complex and sophisticated reinforcement learning algorithms such as SARSA, DDPG, TRPO and A2C. These algorithms can be helpful for larger and more complex environments by involving more sophisticated methods for exploration, exploitation, and function approximation.

Another direction for future work is to incorporate more features and variables into the model. The current simulator model considers only a few variables, such as the customer's age, income, and purchase history. However, there may be other factors, such as social media activity, online behaviour, and geographic location, that can significantly impact customer behaviour. Including these variables can create a more realistic and nuanced simulation environment, which can improve the accuracy and effectiveness of the reinforcement learning agent. Finally, applying the customer behaviour simulator model can be extended to other domains and industries.

Bibliography

[1] Wayne D Hoyer, Deborah J MacInnis, and Rik Pieters. *Consumer behavior*. Cengage learning, 2012.

[2] Consumer Behaviour Marketing Photos. https://www.alamy.com/stock-photo/consumer-behavior.html.

[3] Type of Consumer Behaviour in Marketing. https://www.shutterstock.com/image.

[4] Dunnhumby Dataset in Kaggle. https://www.kaggle.com/datasets/frtgnn/dunnhumby-the-complete-journey.

[5] Luke Plonsky and Hessameddin Ghanbar. Multiple regression in l2 research: A methodological synthesis and guide to interpreting r2 values. *The Modern Language Journal*, 102(4):713–731, 2018.

[6] Hans Marmolin. Subjective mse measures. *IEEE transactions on systems, man, and cybernetics*, 16(3):486–489, 1986.

[7] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[8] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7:e623, 2021.

[9] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[10] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27*, pages 345–359. Springer, 2005.

[11] Mwamba Kasongo Dahouda and Inwhee Joe. A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9:114381–114391, 2021.

[12] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[13] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1–49, 2012.

[14] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[15] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[16] Angshuman Paul, Dipti Prasad Mukherjee, Prasun Das, Abhinandan Gangopadhyay, Appa Rao Chintha, and Saurabh Kundu. Improved random forest for classification. *IEEE Transactions on Image Processing*, 27(8):4012–4024, 2018.

[17] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[18] Jonathan Mark and Michael A Goldberg. Multiple regression analysis and mass assessment: A review of the issues. *Appraisal Journal*, 56(1), 1988.

[19] Min Xu, Pakorn Watanachaturaporn, Pramod K Varshney, and Manoj K Arora. Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, 97(3):322–336, 2005.

[20] Tim F Cootes, Mircea C Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 278–291. Springer, 2012.

[21] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[22] Tech Companies Apply Reinforcement Learning To Marketing. https://www.topbots.com/reinforcement-learning-in-marketing/.

[23] Dajun Du, Kang Li, and Minrui Fei. A fast multi-output rbf neural network construction method. *Neurocomputing*, 73(10-12):2196–2202, 2010.

[24] Structure of a multi output neural network model. https://www.researchgate.net/figure/Structure-of-a-multi-input-multi-output-neural-network_fig2_316738074.

[25] Marco A Wiering and Martijn Van Otterlo. Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729, 2012.

[26] Reinforcement Learning reward action feedback loop. https://www.javatpoint.com/reinforcement-learning.

[27] Nicola Beume, Holger Danielsiek, Christian Eichhorn, Boris Naujoks, Mike Preuss, Klaus Stiller, and Simon Wessing. Measuring flow as concept for detecting game fun in the pac-man game. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3448–3455. IEEE, 2008.

[28] Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip Thomas. Evaluating the performance of reinforcement learning algorithms. In *International Conference on Machine Learning*, pages 4962–4973. PMLR, 2020.

[29] Danijar Hafner, James Davidson, and Vincent Vanhoucke. Tensorflow agents: Efficient batched reinforcement learning in tensorflow. *arXiv preprint arXiv:1709.02878*, 2017.