SCHOOL OF INFORMATICS & IT

Machine Learning for Decision Making

Estimating the expected expenditure of a customer for Q4

Student Name (Matric Number):	Javen Lai Le Yu (2202934B)
Tutorial Group:	P03
Tutor:	Zaw Htet Wai
Submission Date:	2/12/2023

Declaration of Originality

I am the originator of this work and I have appropriately acknowledgedall other original sources used as my references for this work.

I understand that Plagiarism is the act of taking and using the whole or any part of another person's work, including work generated by AI, andpresenting it as my own.

I understand that Plagiarism is an academic offence

and if I am found to have committed or abetted the offence of plagiarism in relation to this submitted work, disciplinary action will beenforced.

Declaration on the use of Generative AI tools for assignments

Describe how you have used Generative AI tools such as ChatGPT or Dall.E-2 in your assignment. Show snapshots of the conversations with the AI tool (i.e., the prompts you used and the response you get from the AI tool).

PDCA + Coding:
https://chat.openai.com/share/4207c097-4921-43ce-9b3a-a3e6523cbf3a
https://chat.openai.com/share/141fa7be-eeeb-4895-8b66-36aae1ff3a9d
https://chat.openai.com/share/8be6cac6-6c72-49af-85fa-1aa825ffaa0e
REPORT:
https://chat.openai.com/share/17ba933c-e697-4718-a226-57c7a0e3eee1

I used ChatGPT to generate code under 'Do' for some PDCAs. I would write my intended outcome and what code I want, followed by a snippet of a sample code from Part 1 (to create new features):

Anonymous
# Count the number of cross-selling occurrences in the dataset
cross_sell_count_dataset = dataset.groupby(['customer_id', 'datetime'])['product_id'].nunique().reset_index(name='cross-selling_count')

# Merge the cross-selling count back to the df_cross DataFrame
df_cross = pd.merge(df_cross, cross_sell_count_dataset, on=['customer_id'], how='left')

Fill NaN values with 0 for customers who didn't engage in cross-selling in the dataset
df_cross['cross-selling_count'].fillna(0, inplace=True)

# drop datetime
df_cross.drop(columns=['datetime'], inplace=True) // fix this to calculate upselling count for EACH CUSTOMER. EACH CUSTOMER IS GROUPBY CUSTOMER ID. WHEN ACCOUNTING FOR UPSELLING FOR EACH CUSTOMER, ONLY COUNT THE ROWS WHERE UPSELLING IS MORE THAN 1

ChatGPT
It seems like you want to calculate the total upselling count for each customer and then filter the rows where the upselling count is more than 1. Here's how you can do it:

python Copy code
```python
# Count the number of cross-selling occurrences in the dataset
cross_sell_count_dataset = dataset.groupby(['customer_id', 'datetime'])

# Count the total upselling count for each customer
total_upselling_per_customer = cross_sell_count_dataset.groupby('custom

# Merge the total upselling count back to the df_cross DataFrame
df_cross = pd.merge(df_cross, total_upselling_per_customer, on='custome

# Fill NaN values with 0 for customers who didn't engage in cross-selli
```

I also used ChatGPT for debugging code and troubleshooting.:

Anonymous

Input In [160]

def check_performance(name, df, target=TARGET_COLUMN, exclude, params='', SEED=SEED):

^

SyntaxError: non-default argument follows default argument

ChatGPT

The error you're encountering is due to the placement of the `exclude` parameter after the default parameter `target=TARGET_COLUMN`. In Python, non-default arguments cannot follow default arguments.

To fix this, you should place the non-default argument `exclude` before any default arguments. Here's the corrected function definition:

python Copy code

```python
def check_performance(name, df, exclude, target=TARGET_COLUMN, params='
    # Your function code here
```

This way, the `exclude` parameter is defined before any default parameters. Update your function definition accordingly, and the error should be resolved.

I also asked it to rephrase some explanation to be clearer, fluent, and concise for easier readability and understanding:

Anonymous

9. How many different products bought per category + Total quantity of products bought for Q1-Q2.

Plan:

There is high collinearity between expenses and total items and this is an issue that makes the model uninterpretable as I cannot identify the true drivers for customer engagement.

Thus, I want to explore if simply providing the count of how many different products bought per category and a total count of how many products bought for Q1-Q2 is provides better explainability, because it is more meaningful to analyze the engagement of products in each category without inflating it with quantity (how many times a customer buy), and this PDCA cycle will add depth of detail while making it simple to understand what the features represent. // rephrase plan to make it clear and more convincing to justify that this pdca cycle helps business usable the model. keep it concise. then, give code to implement the stipulated changes

ChatGPT

Revised Plan:

The current model faces interpretability challenges due to collinearity between expenses and total items. To enhance interpretability and focus on meaningful engagement metrics, I propose incorporating counts of unique products bought per category and the total quantity of products purchased during Q1-Q2. This approach allows for a more granular analysis of product engagement within each category without being influenced by purchase frequency. By implementing this PDCA cycle, we aim to provide a clearer understanding of the features, making the model more business-usable.

Report writing:

You

Usage for Decision-Making:
Revenue Forecasting: Summing predicted spending for all customers allows businesses to forecast Q4 revenue, informing investment decisions without risking financial stability.
Campaign Evaluation: Business can experiment strategies in Q1-Q2 and compare predicted Q4 spending with previous years to assess the success of campaigns.
Product Ventures: Validate the feasibility of investing in new products or ventures by understanding the expected Q4 revenue.

Usage for Business Planning:
Product Evaluation: Understand which product categories attracts customers to spend more, and leverage this knowledge to implement ????
Discount & Cross-selling Strategies: Study the interaction of discounts and bundling products on various categories to find the most effective discount strategies that improve profitability.
Inventory Management: Optimize inventory based on predicted spending, ensuring sufficient stock of high-demand products without excess.
Targeted Marketing: Identify customers who are likely losing engage in Retail Hypermarket, so that necessary prevention steps can be executed before we lose customers.
Loyalty Programs: Implement data-driven loyalty programs where customers have to shop for certain number of days to be eligible for a loyalty program with benefits that entice customers.
Financial Planning: Incorporate predicted Q4 expenditures into financial planning for accurate resource allocation.
// make these more actionable where business can actually know how to use model to derive these

PDCA5 (ACCEPTED):
I segmented customers, functioning on the belief that the first letter of customer_id represents the customer type. It's discovered that customer_type B are the weakest spenders for all quarters. I tried out various encoding methods like one-hot, target, and ordinal to see which produces best performance. I chose ordinal encoding to segregate customer B from the rest as its clearly understandable and produces best model performance.
// make PDCA 3,4,5 AS CONCISE AS 1 AND 2

I use chatGPT to generate ideas on how model can be used after deployment. I also ask it to give me ideas on how to phrase the ideas in concise, clear, and succinct manner.

How do you indicate the reference?

The content generated by AI tools are not retrievable except by the user who generated them, so they are considered non-recoverable sources. Although non-recoverable data or quotations in APA Style papers are usually cited as personal communications, with ChatGPT-generated text there is no person communicating. Quoting text from ChatGPT chat is therefore more like sharing the output of an algorithm, with a reference list entry and the corresponding in-text citation.

According to the official APA Style site, ChatGPT references should be cited as:

E.g. OpenAI. (2023). *ChatGPT* (Sep 25 version) [Large language model].

https://chat.openai.com/chat

Content Page

Introduction

The Random Forest Regressor is trained on Retail Hypermarket's 2022 transactional data (Q1, Q2, Q4) to predict a customer's expected Q4 expenditure based on Q1-Q2 behavior.

Preview of Model

```
df_final.sample(random_state=SEED)
```

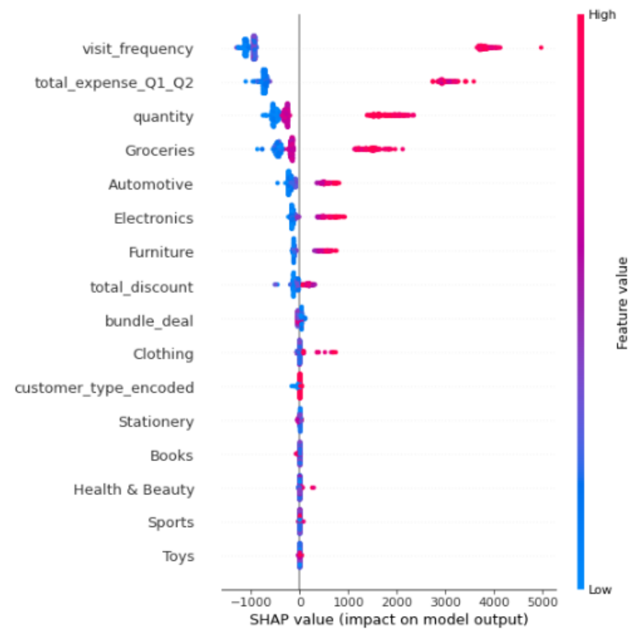
	total_expense_2022Q4	total_expense_Q1_Q2	Automotive	Books	Clothing	Electronics	Furniture	Groceries	Health & Beauty	Spo
762	6478.45	8776.46	3	2	2	2	2	23	5	

```
predict_engagement(X_train)
```

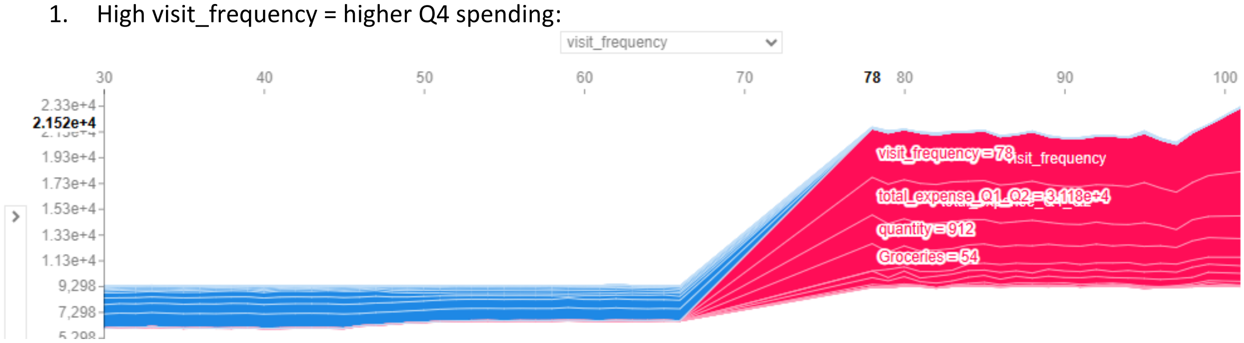
```
Enter total_expense_Q1_Q2: 8776.46
Enter Automotive: 3
Enter Books: 2
Enter Clothing: 2
Enter Electronics: 2
Enter Furniture: 2
Enter Groceries: 23
Enter Health & Beauty: 5
Enter Sports: 0
Enter Stationery: 3
Enter Toys: 2
Enter quantity: 183
Enter bundle_deal: 7
Enter total_discount: 736.7
Enter visit_frequency: 37
Enter customer_type_encoded: 0
Predicted Value: 6015.081750000003
```

Model Interpretation

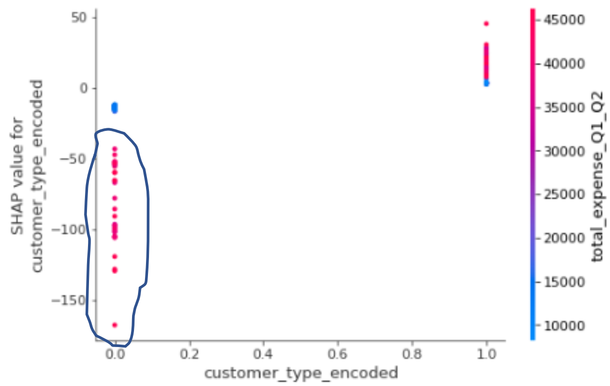
```
# summary plot
shap.summary_plot(shap_values_dtr[:,:], features=X_test, feature_names=X_test.columns)
```



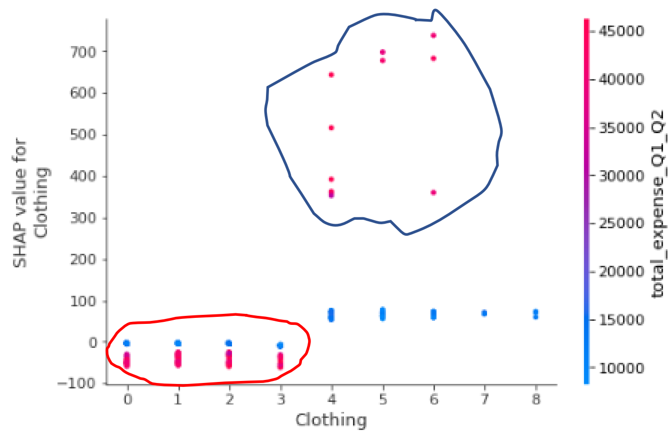
1. High visit_frequency = higher Q4 spending:



2. Engagement in wide variety of groceries, automotive, electronics, furniture, or clothing leads to higher spending.
3. High-spending customer_type B spent less during Q4, **suggesting potential loss of interest in shopping at Retail Hypermarket**:

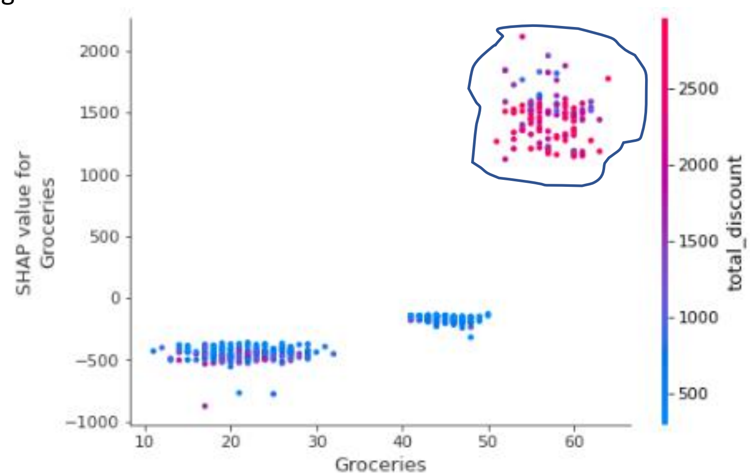


4. There are **2 types of clothing customers**:
   - Branded consumers (focus on 1-3 brands)
   - Thrift consumers (fast fashion)

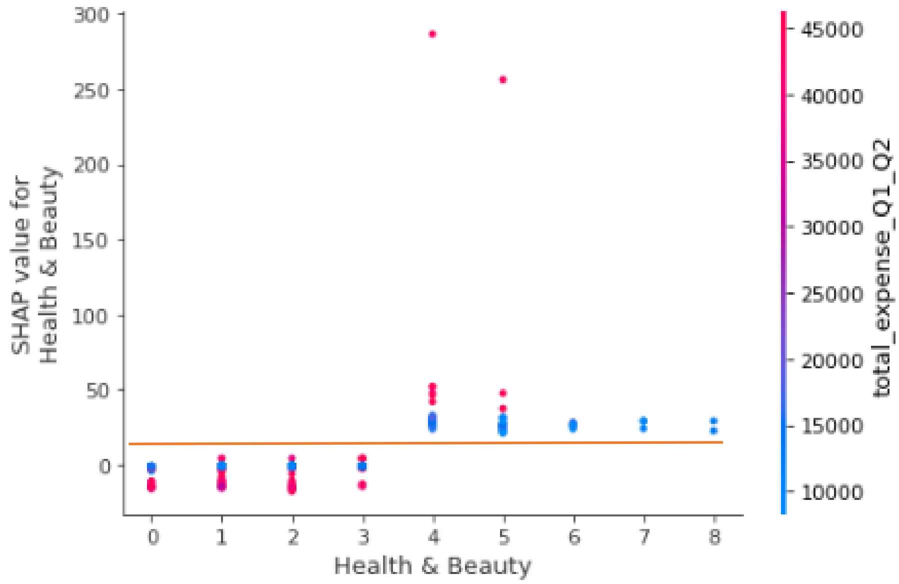


Targeting Thift consumers is more profitable than focusing on certain brands/designs.

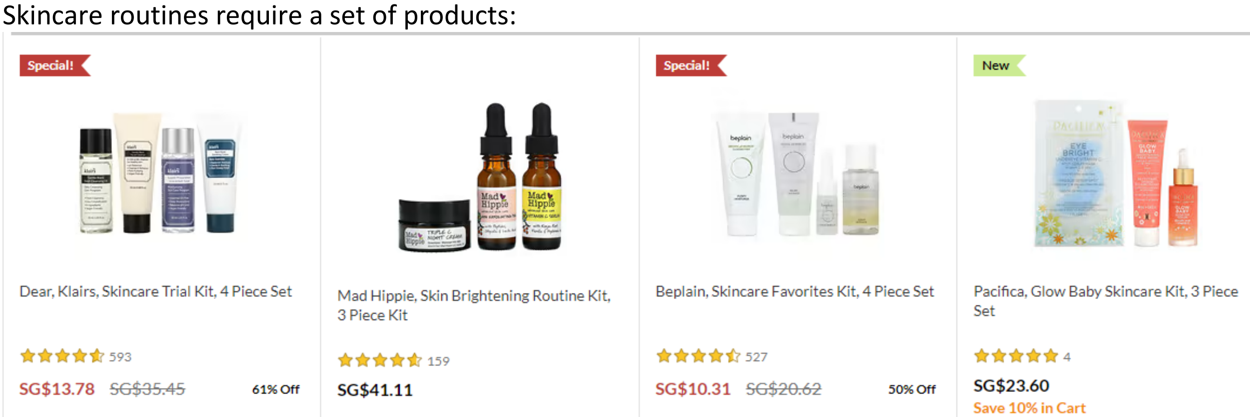
5. **Discounts could be effective** in improving sales of groceries because customers engaged in groceries received a lot more discounts:



6. Customers buying at least 4 **Health/Beauty products usually spend more**, revealing a cross-selling opportunity where products could be bundled to encourage customers to buy them together:

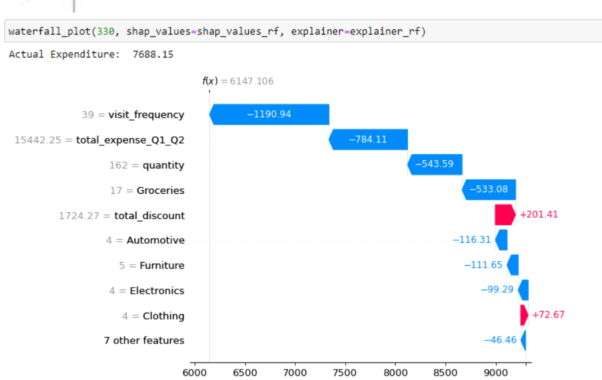


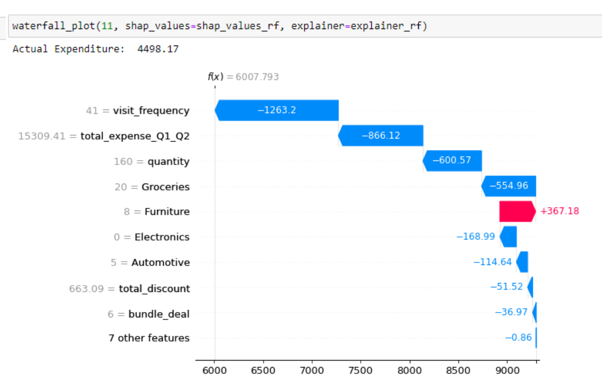
Skincare routines require a set of products:



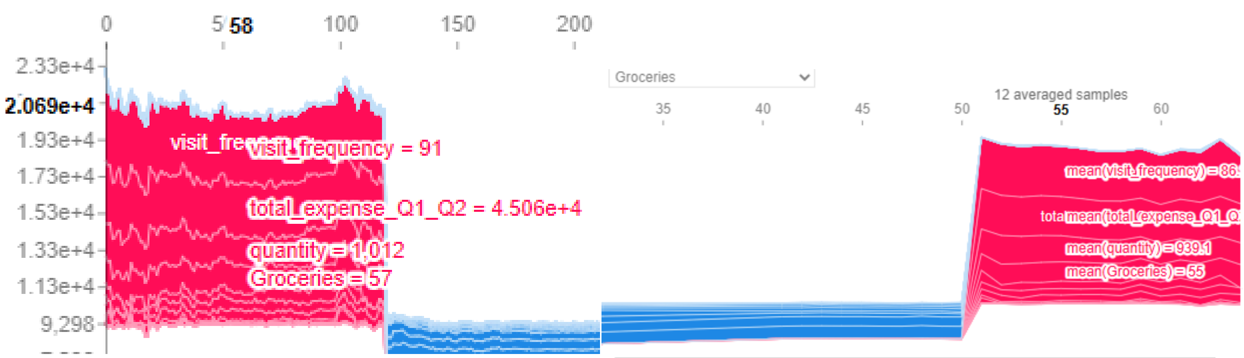
Customer Study

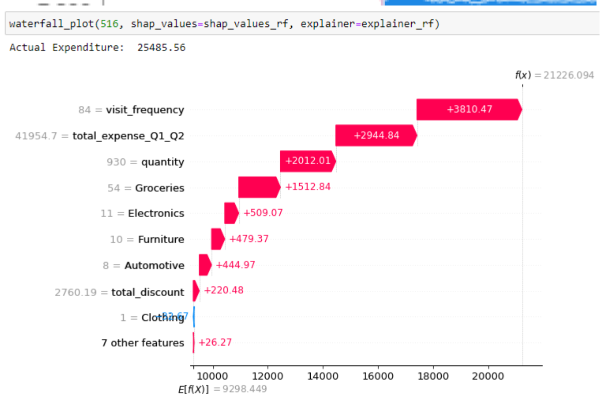
Low-spender:

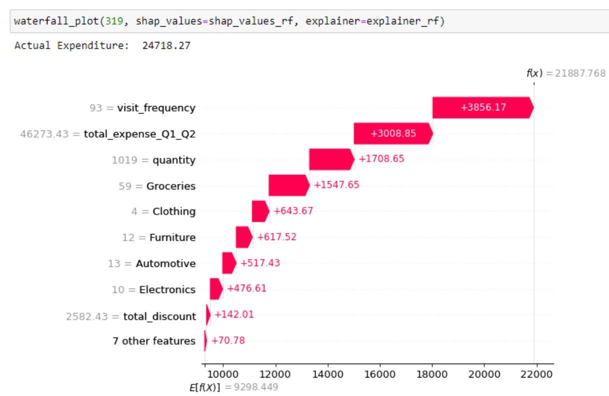




High-spender:







Insights:

- Customers who visited below 78 days are expected to spend less.
- Higher engagement in groceries, electronics, furniture, automotive, and clothing leads to increased spending.

Recommendations

1. **Build customer loyalty:**
   Loyalty programs that encourage customers to shop at least 156 days a year for exclusive benefits (gifts/discounts).

2. **More variety of groceries and fashion/clothing:**
   Improve engagement as customers have more choices, leading to increased spending.

3. **Focus marketing on automotive, electronics, and furniture:**
   Customers who are engaged in lucrative categories spend more. Enticing customers to buy more of such products using compelling promotions could improve revenue.

4. **Bundle relevant beauty products:**
   Ensure customers won't miss out on a product from their skincare routine because they don't have to manually find each product if products are bundled.

5. **Discount on slow-moving groceries:**
   Discounts are effective in boosting sales for groceries, hence could be used to get rid of poor-performing groceries that are hard-to-sell.

6. **Rekindle engagement of Customer Type B:**
   Conduct surveys to understand changing preferences and implement campaigns that attract them.

Model Development

	Model	MAPE Test	R2 Test	MAPE Train	R2 Train	Parameters	Dataset
0	Benchmark DTR	0.403236	0.63825	0.0	1.0	None	final_df
1	Pruned Benchmark DTR	0.304871	0.820964	0.274062	0.841938	{'criterion': 'squared_error', 'max_depth': 5,...	[total_items_2022Q1, total_expense_2022Q1, tot...
2	PDCA 1 (dropping Q3 + combine across quarters)	0.30398	0.81998	0.27044	0.839315	{'criterion': 'squared_error', 'max_depth': 5,...	[total_items_Q1_Q2, total_expense_Q1_Q2]
3	PDCA 2 (delving into categories + total quantity)	0.295522	0.821345	0.270774	0.839907	{'criterion': 'squared_error', 'max_depth': No...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
4	PDCA 3 (sales features)	0.286756	0.815136	0.260141	0.838803	{'criterion': 'absolute_error', 'max_depth': 5...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
5	PDCA 4 (visit frequency)	0.296855	0.821699	0.269587	0.841343	{'criterion': 'squared_error', 'max_depth': 5,...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
6	PDCA 5 (one-hot customer)	0.301601	0.818476	0.265442	0.845375	{'criterion': 'squared_error', 'max_depth': 5,...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
7	PDCA 5 (target customer)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
8	PDCA 5 (segregating customer B)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
9	PDCA 5 (ordinal customer)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
10	DTR	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
11	PDCA 5 (segregate B) drop expenses	0.290995	0.814602	0.259741	0.839431	{'criterion': 'absolute_error', 'max_depth': N...	[Automotive, Books, Clothing, Electronics, Fur...
12	Random Forest	0.284191	0.830237	0.258043	0.842556	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
13	Random Forest dropping unimportant	0.284429	0.829988	0.257968	0.842897	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...

```
results.sort_values('MAPE Test')
```

	Model	MAPE Test	R2 Test	MAPE Train	R2 Train	Parameters	Dataset
12	Random Forest	0.284191	0.830237	0.258043	0.842556	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
13	Random Forest dropping unimportant	0.284429	0.829988	0.257968	0.842897	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
7	PDCA 5 (target customer)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
8	PDCA 5 (segregating customer B)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
9	PDCA 5 (ordinal customer)	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
10	DTR	0.28504	0.82207	0.260308	0.837304	{'criterion': 'absolute_error', 'max_depth': N...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
4	PDCA 3 (sales features)	0.286756	0.815136	0.260141	0.838803	{'criterion': 'absolute_error', 'max_depth': 5...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
11	PDCA 5 (segregate B) drop expenses	0.290995	0.814602	0.259741	0.839431	{'criterion': 'absolute_error', 'max_depth': N...	[Automotive, Books, Clothing, Electronics, Fur...
3	PDCA 2 (delving into categories + total quantity)	0.295522	0.821345	0.270774	0.839907	{'criterion': 'squared_error', 'max_depth': No...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
5	PDCA 4 (visit frequency)	0.296855	0.821699	0.269587	0.841343	{'criterion': 'squared_error', 'max_depth': 5,...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
6	PDCA 5 (one-hot customer)	0.301601	0.818476	0.265442	0.845375	{'criterion': 'squared_error', 'max_depth': 5,...	[total_expense_Q1_Q2, Automotive, Books, Cloth...
2	PDCA 1 (dropping Q3 + combine across quarters)	0.30398	0.81998	0.27044	0.839315	{'criterion': 'squared_error', 'max_depth': 5,...	[total_items_Q1_Q2, total_expense_Q1_Q2]
1	Pruned Benchmark DTR	0.304871	0.820964	0.274062	0.841938	{'criterion': 'squared_error', 'max_depth': 5,...	[total_items_2022Q1, total_expense_2022Q1, tot...
0	Benchmark DTR	0.403236	0.63825	0.0	1.0	None	final_df

PDCA1:

1. Rounded-off expenses to 2d.p. to match conventional understanding of prices.

2. Dropped Q3 features as total_items and total_expense across Q1-Q3 have high collinearity. This makes the model usable directly after Q2, giving the business the duration of Q3 to approve and implement strategies formulated using the model.

	total_items_2022Q1	total_expense_2022Q1	total_items_2022Q2	total_expense_2022Q2
total_items_2022Q1	1.000000	0.922950	0.869373	0.851799
total_expense_2022Q1	0.922950	1.000000	0.841242	0.847217
total_items_2022Q2	0.869373	0.841242	1.000000	0.922085
total_expense_2022Q2	0.851799	0.847217	0.922085	1.000000

3. Lastly, total_items and total_expense for Q1-Q2 are combined to provide overview interpretation of how total expenses or items for Q1-Q2 influence predictions.

PDCA2:

1. To understand how engagement for various categories influences potential spending, I will count how many products a customer purchased for each category.

2. I will replace total_items with quantity to represent amount of items the customer bought.

PDCA3:

To understand how common sales strategies influence customer spending.

1. Implemented discounts by deducting the unit_price the customer bought a product for from its highest Q1-Q2 pricing.
2. Introduced Bundles/Cross-Selling that counts occurrences where customer buys multiple products a day.

PDCA4:

To enable data-driven loyalty programs, I added visit_frequency as a metric to assess customer loyalty based on the number of days they shopped.

PDCA5:

Segmented customers by encoding 0 for customer_type B and 1 for others based on the first letter of customer_id, because type B is weakest spender across all quarters.

DTR:

Cannot contain total_expense_Q1_Q2 due to negative importance, which indicates negative impact on model performance.

	Feature	Importance
13	visit_frequency	0.980911
2	Clothing	0.003802
10	quantity	0.003548
11	bundle_deal	0.003421
12	total_discount	0.003253
6	Health & Beauty	0.002684
14	customer_type_encoded	0.002381
0	Automotive	0.000000
1	Books	0.000000
3	Electronics	0.000000
4	Furniture	0.000000
5	Groceries	0.000000
7	Sports	0.000000
8	Stationery	0.000000
9	Toys	0.000000

```
Training performance
Mean Absolute Percentage Error on Train Set: 0.26
R-squared on Train Set: 0.84

Testing performance
Mean Absolute Percentage Error on Test Set: 0.29
R-squared on Test Set: 0.81
```

```
find_perm_importance(final_dtr, X_test, y_test)
```

Permutation Importances:

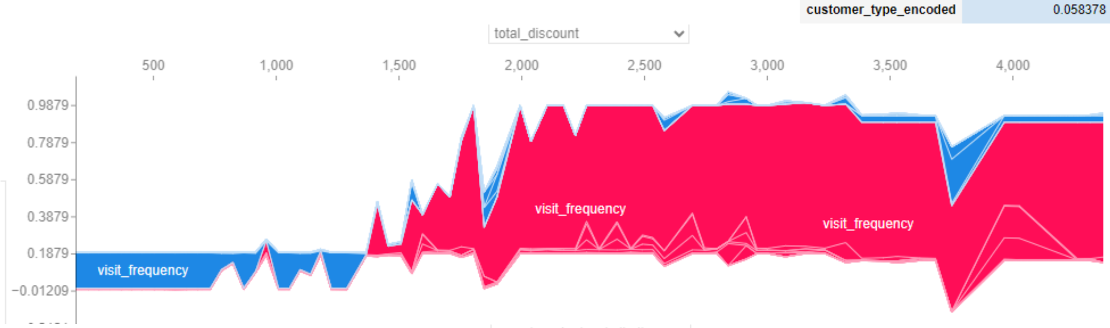
	Feature	Importance Mean	Importance Std
14	visit_frequency	1.855596	0.093445
13	total_discount	0.065048	0.012712
12	bundle_deal	0.002102	0.003462
3	Clothing	0.001872	0.003640
1	Automotive	0.000000	0.000000
2	Books	0.000000	0.000000
4	Electronics	0.000000	0.000000
5	Furniture	0.000000	0.000000
6	Groceries	0.000000	0.000000
7	Health & Beauty	0.000000	0.000000
8	Sports	0.000000	0.000000
9	Stationery	0.000000	0.000000
10	Toys	0.000000	0.000000
11	quantity	0.000000	0.000000
15	customer_type_encoded	0.000000	0.000000
0	total_expense_Q1_Q2	-0.002872	0.001428

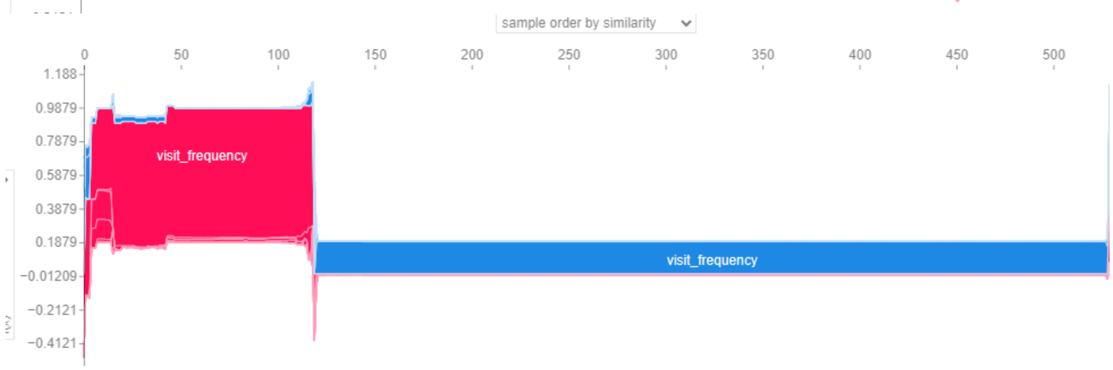
Additionally, total_expense has multicollinearity with many features, suggesting information overlaps with other features which could affect interpretability.

	total_expense_2022Q4	total_expense_Q1_Q2
total_expense_2022Q4	1.000000	0.878001
total_expense_Q1_Q2	0.878001	1.000000
Automotive	0.784258	0.821017
Books	-0.150067	-0.143884
Clothing	-0.110979	-0.122509
Electronics	0.783372	0.882625
Furniture	0.802892	0.913507
Groceries	0.636484	0.681254
Health & Beauty	-0.141342	-0.108339
Sports	-0.101777	-0.086302
Stationery	-0.136358	-0.150466
Toys	-0.138260	-0.137596
quantity	0.626305	0.677014
bundle_deal	0.600675	0.674586
total_discount	0.702608	0.775214
visit_frequency	0.839764	0.916800
customer_type_encoded	0.058378	0.043693

DTC:

Rejected due to poor splits and heavy reliance on "visit_frequency"; cannot learn other features' impact.





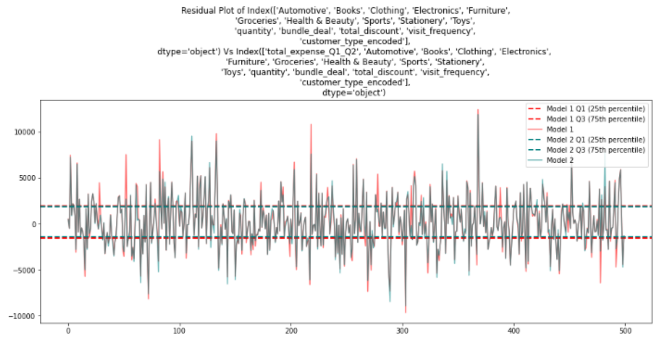
RF (CHOSEN):

As RF is robust against multicollinearity, **total_expense can be used**.

```
                 Feature  Importance
14       visit_frequency    0.266243
11              quantity    0.197445
0      total_expense_Q1_Q2  0.178659
6              Groceries    0.144987
5              Furniture    0.063255
4            Electronics    0.058868
1             Automotive    0.056139
13        total_discount    0.028270
12           bundle_deal    0.001934
3               Clothing    0.001888
7        Health & Beauty    0.000667
15 customer_type_encoded    0.000491
9             Stationery    0.000442
8                 Sports    0.000339
2                  Books    0.000272
10                  Toys    0.000100
Training performance
Mean Absolute Percentage Error on Train Set: 0.26
R-squared on Train Set: 0.84

Testing performance
Mean Absolute Percentage Error on Test Set: 0.28
R-squared on Test Set: 0.83
```

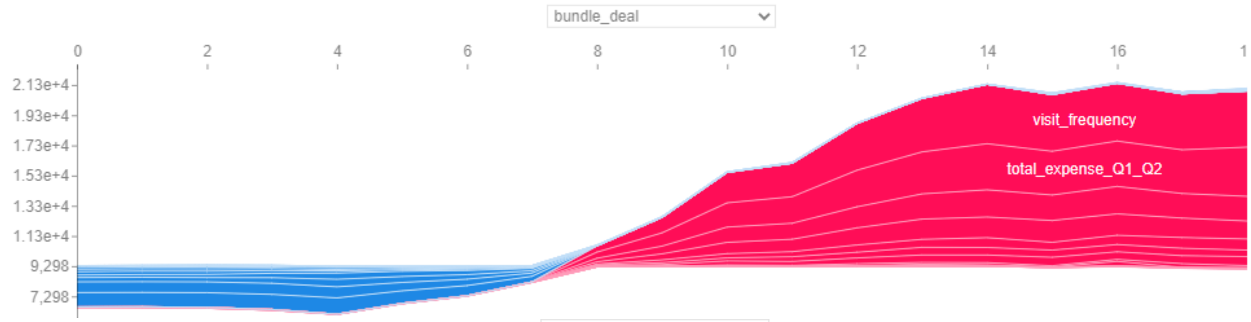
```
model_comparer(final_dtr, rf_regressor, df_final_drop, df_final)
```

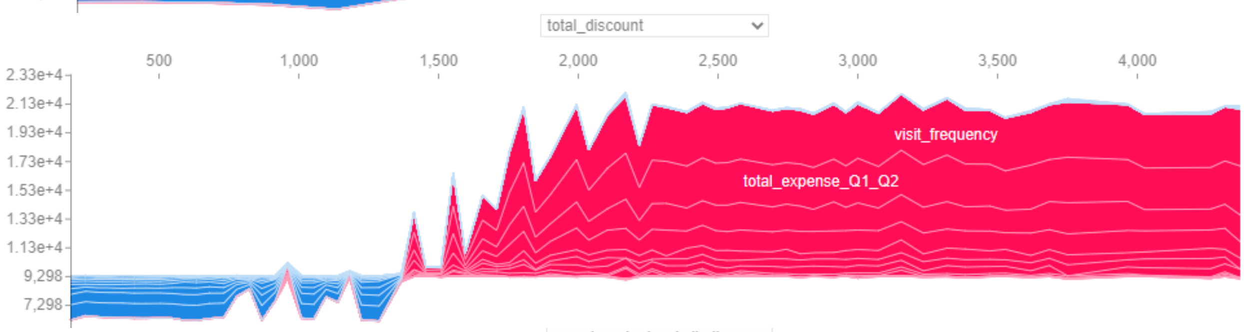


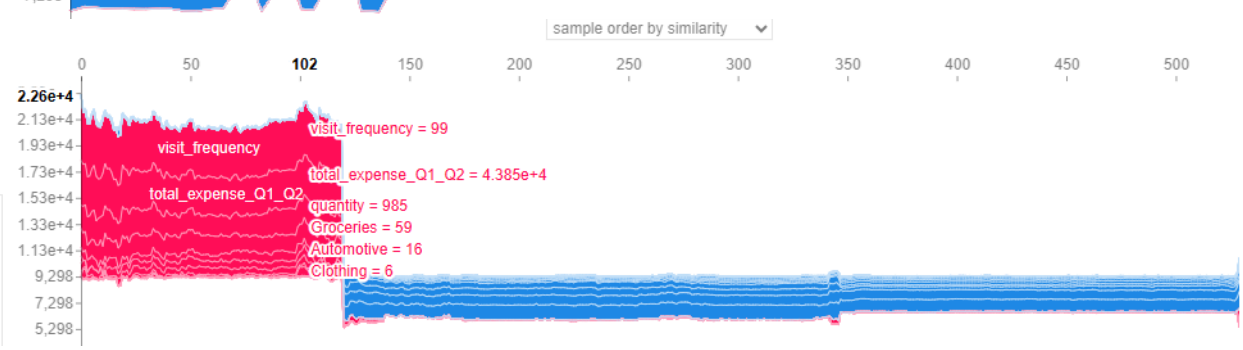

```
"Model 2 RandomForestRegressor(bootstrap=False, criterion='absolute_error',\n                    ma
x_features='sqrt', max_leaf_nodes=10, n_jobs=-1,\n                      random_state=2202934) with
['total_expense_Q1_Q2', 'Automotive', 'Books', 'Clothing', 'Electronics', 'Furniture', 'Groceries',
'Health & Beauty', 'Sports', 'Stationery', 'Toys', 'quantity', 'bundle_deal', 'total_discount', 'visi
t_frequency', 'customer_type_encoded'] has the smallest IQR range."
```

Random Forest has better accuracy and explainability,and provides insights into how customer spending in Q1-Q2 affects Q4 expected spending. Thus, RF is better than DTR.

Based on Residual Plot, its clear that RF has better performance due to noticably smaller residuals.







1. Best-performing Regressor.
2. Clean splits in features, clearer interpretation.
3. Multiple features assessed to derive prediction, allowing comprehensive insights.

Model Deployment

Deployment Plan:

The model can be hosted on scalable cloud services like AWS or Azure to ensure business can expand instances if needed. After the last day of Q2, the business' transactional dataset will be loaded into an ETL pipeline to transform the data for model usage, to predict each customer's expected Q4 spending.

Decision-Making:

1. Summing predicted spending of customers to **forecast the business' Q4 revenue** enables **informed investment decisions and resource allocation** without jeopardizing financial stability.

2. Experiment with strategies in Q1-Q2 and compare the predicted Q4 spending with previous years to **assess the effectiveness of campaigns**. This aids in **refining future strategies** for optimal results.

3. **Validate the feasibility of new products/ventures** by gaining insights into expected Q4 revenue. E.g. Check if diversifying clothing products leads to higher revenue.

Business Planning:

1. Identify product categories that attract higher spending, allowing businesses to **focus resources on optimizing and expanding successful products**.

2. Analyze the interaction of discounts and bundled products across various categories to **uncover the most profitable strategies** for each category; Implement **data-driven pricing and bundling strategies**.

3. Optimize inventory to **ensure sufficient stock for products without excess**.
   - e.g. electronics customer is expected to spend $XXXX on Q4 based on Q1-Q2 behavior, so the business should prepare around $XXXX worth of electronics for this customer to avoid stockouts or overstocking.

4. **Identify customers showing signs of disengagement** with Retail Hypermarket.
   - early prevention against customer loss by re-engaging customers using campaigns that target their needs.

5. Implement **data-driven loyalty programs** that gets customers shop for XX number of days to qualify. Include enticing benefits that drive more/repeated business.
   - Bundle deals for loyalty program customers.

Implications:

1. Random Forest is computationally expensive due to its robustness in evaluating the effects of each factor. Hence, more computational power and time (for model to run) is required when RF is deployed.

2. Intricacies of customer engagement may change over time and differ from the pattern the RF uses to derive predictions. It's recommended to re-train the model annually using the previous year's Q1-Q4 data.

3. Unanticipated factors like economic downturn, health crisis, etc. are not factored by model. Business should account for external factors as they could cause discrepancies in actual Q4 spending.

Word Count: 999 (excluding cover page, declaration, and content page)

End of Report

References

Skincare routine:
https://sg.iherb.com/c/gifts?swkw=skin&msclkid=67851ff0c6ca18cadc587a98ef96abc0&utm_source=bing&utm_medium=cpc&utm_campaign=Category%20%3E%20Bath%20%26%20Beauty%20%3E%20Singapore%20%3E%20Acquisition%20(English)&utm_term=skin%20care%20gift%20set&utm_content=Category%20-%20Bath%20%26%20Beauty%20-%20Gift%20Sets&gclid=67851ff0c6ca18cadc587a98ef96abc0&gclsrc=3p.ds

Random Forest with Multicollinearity:
Are Random Forests affected by multi-collinearity between features? | ResearchGate