

# Cloud Final Project

Team Members: Jaya Pavani Pathakota, Lakshmi Praneetha Singamreddy, Shruthi Asolkar

**Azure WebApp Url:** <https://cloudgroup44-fgdkbfc6cghzbpa6.centralus-01.azurewebsites.net>

**GitHub Link:** <https://github.com/jayapavanip/Data-Science-Project-on-Retail-Experience-in-Kroger>

Screenshots for Project Requirements:

## 1. **Write-Up on ML Models:**

- **Linear Regression**

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. It assumes a linear relationship and minimizes the error between predicted and actual values using the least squares method. While it's simple and interpretable, it struggles with non-linear relationships and complex datasets.

- **Random Forest**

Random Forest is an ensemble learning method based on decision trees. It creates multiple trees during training and combines their predictions (averaging for regression or majority voting for classification). It handles non-linear relationships, reduces overfitting, and performs well on complex data. However, it can be computationally intensive and less interpretable.

- **Gradient Boosting**

Gradient Boosting is another ensemble technique that builds models sequentially, with each new model correcting the errors of the previous ones. It uses weak learners (e.g., shallow decision trees) and combines them for a strong predictive model. Gradient Boosting excels in accuracy and handles non-linear relationships well but requires fine-tuning to avoid overfitting.

- **Predictive Modelling for CLV**

Gradient Boosting is the most suitable technique for predicting Customer Lifetime Value (CLV). CLV involves non-linear relationships and complex interactions between customer behaviour, spending patterns, and demographics. Gradient Boosting's ability to capture such complexities makes it ideal for accurately forecasting long-term revenue potential, enabling businesses to prioritize high-value customers effectively.

How can we predict long-term revenue potential to prioritize high-value customers?

To predict long-term revenue potential, we can identify high-value customers by implementing a Customer Lifetime Value (CLV) prediction algorithm using Gradient Boosting, as demonstrated below:

```
[9]: # Feature selection
X = df[['Spend', 'Units', 'Income_range', 'Hshd_size', 'Children']]
y = df['CLV'] # Target: Customer Lifetime Value

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the Gradient Boosting model
model_clv = GradientBoostingRegressor()
model_clv.fit(X_train, y_train)

# Evaluate the model
y_pred = model_clv.predict(X_test)
print("CLV Prediction R2 Score:", r2_score(y_test, y_pred))
print("CLV Prediction MSE:", mean_squared_error(y_test, y_pred))

# Save the model
joblib.dump(model_clv, "gradient_boosting_clv.pkl")

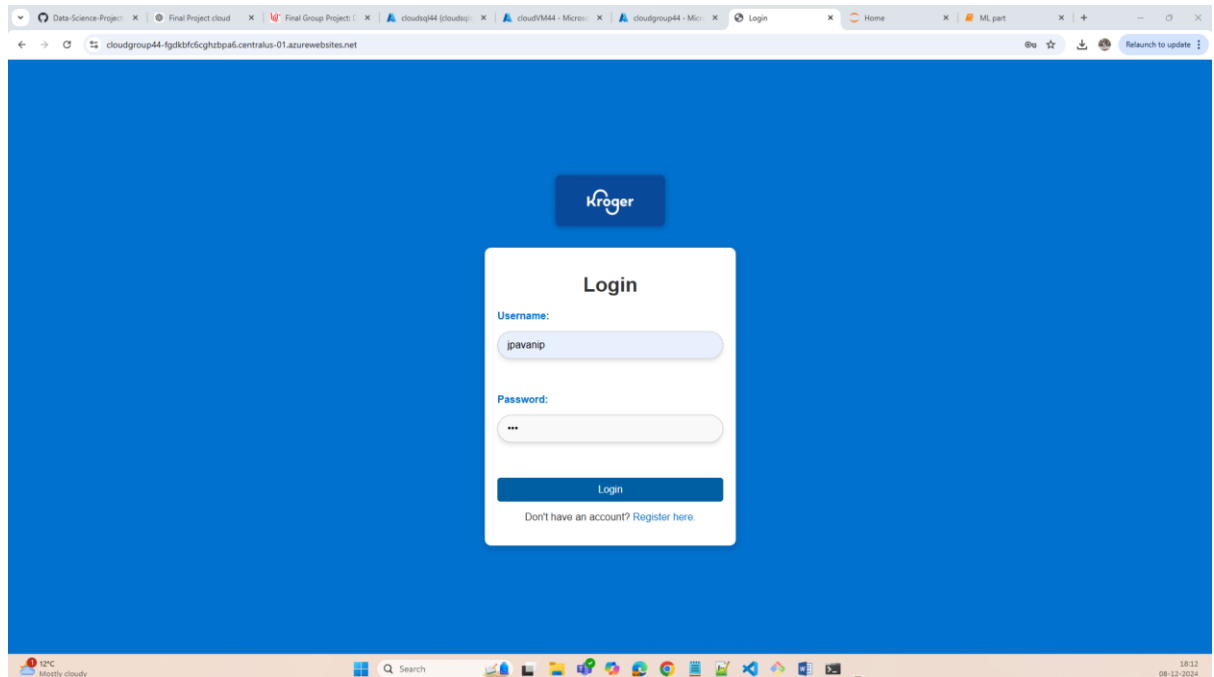
CLV Prediction R2 Score: 0.25551513270988446
CLV Prediction MSE: 0.5131383751720974

[9]: ['gradient_boosting_clv.pkl']
```

The screenshot shows a web browser window with a single tab titled 'Retail ML Predictions'. The address bar shows the URL 'cloudgroup44-fgdkbfcgghztpa6.centralus-01.azurewebsites.net/clv\_predict'. The page has a blue background and a white central form titled 'Customer Lifetime Value Predictions'. The form contains several input fields with labels and values: 'Spend:' with value '10000', 'Units:' with value '3', 'Income Range:' with value '24000', 'Household Size:' with value '2', and 'Children:' with value '0'. Below these fields is a blue 'Predict' button. Under the button, the prediction result is displayed: 'Prediction: ("CLV Prediction":[4.609309840831628])'. At the bottom of the form is a blue 'Go Back' button. The browser's taskbar at the bottom shows the system clock as 18:10 on 09-12-2024, and the weather as '12°C Mostly cloudy'.

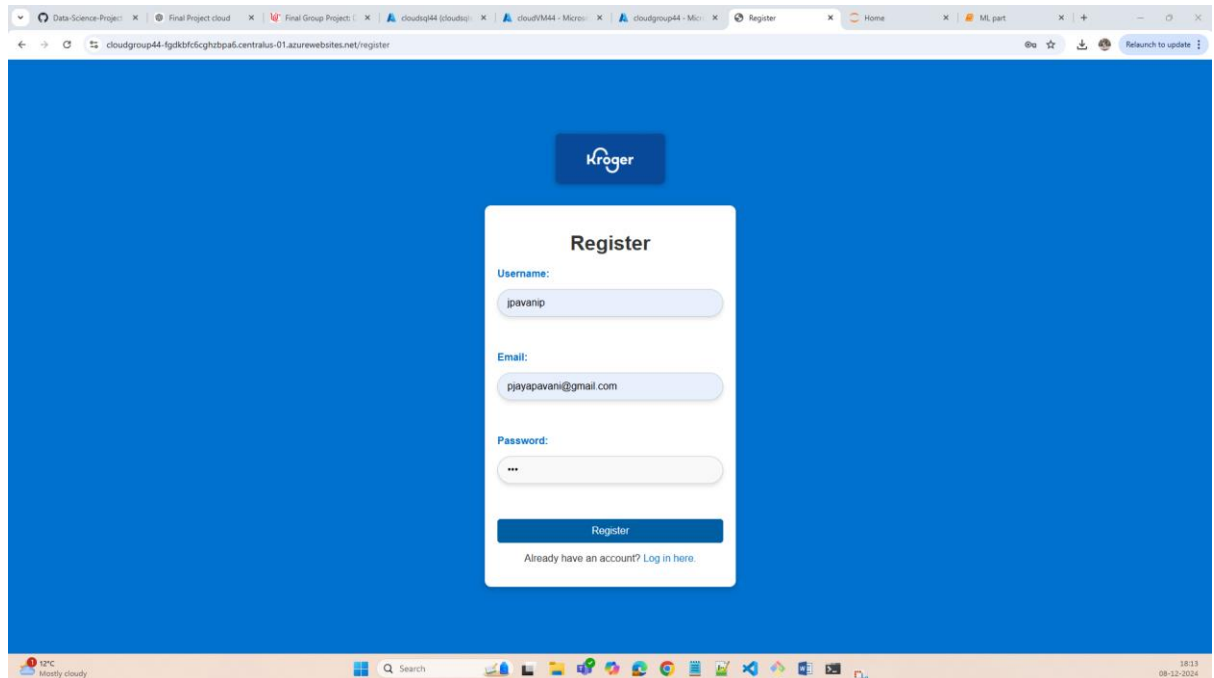
## 2. Web Server Setup:

### Login Page:



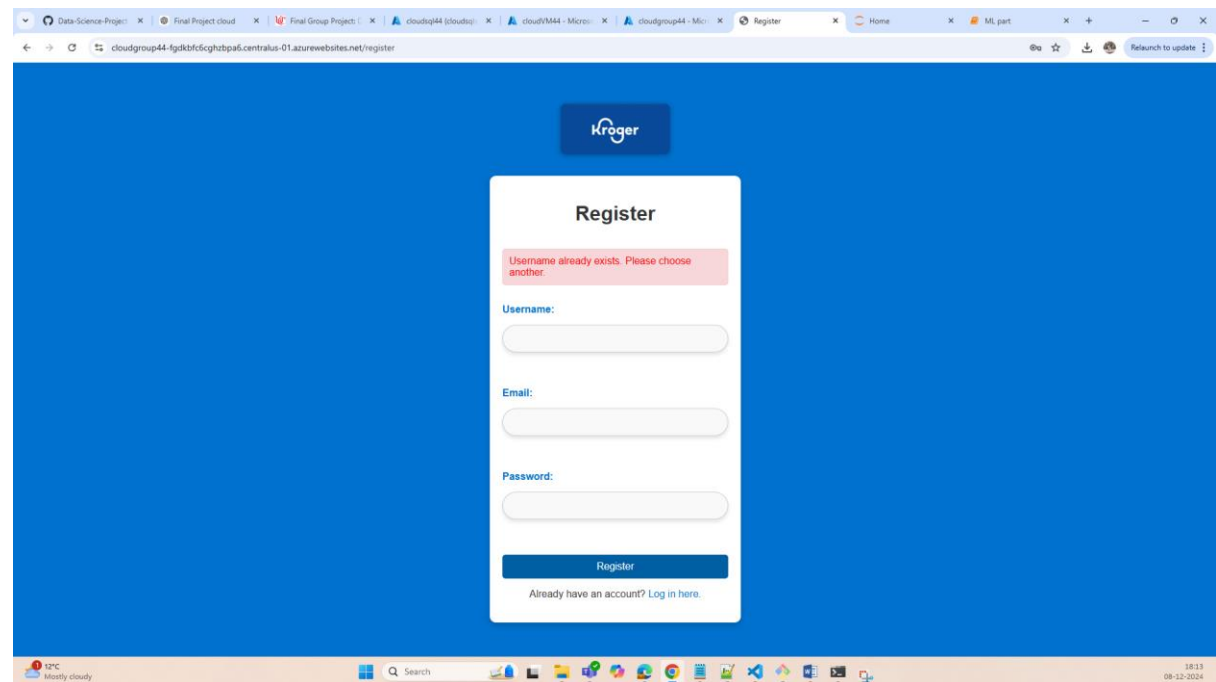
The screenshot shows a web browser window with the URL `cloudgroup44-fgdkbfcghzbp6.centralus-01.azurewebsites.net`. The page has a solid blue background. At the top center is the Kroger logo. Below it is a white login form titled "Login". The form contains two input fields: "Username:" with the value "jpavanip" and "Password:" with masked characters "...". Below the password field is a blue "Login" button. At the bottom of the form is a link: "Don't have an account? [Register here.](#)". The browser's taskbar at the bottom shows the system clock as 18:12 on 08-12-2024.

### Registration page:



The screenshot shows a web browser window with the URL `cloudgroup44-fgdkbfcghzbp6.centralus-01.azurewebsites.net/register`. The page has a solid blue background. At the top center is the Kroger logo. Below it is a white registration form titled "Register". The form contains three input fields: "Username:" with the value "jpavanip", "Email:" with the value "jpavapavani@gmail.com", and "Password:" with masked characters "...". Below the password field is a blue "Register" button. At the bottom of the form is a link: "Already have an account? [Log in here.](#)". The browser's taskbar at the bottom shows the system clock as 18:13 on 08-12-2024.

Added validations for register and login pages too, sample example:

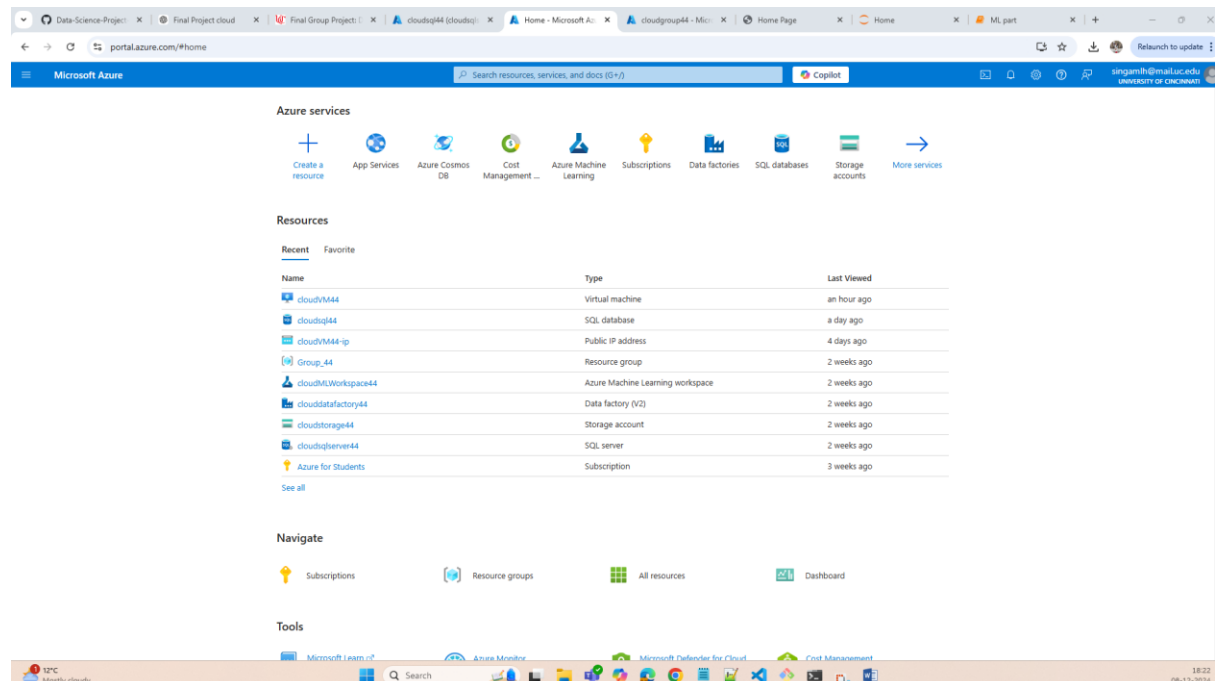


### 3. Datastore and Data Loading:

Procedure:

1. Created a Blob Storage and uploaded the 3 CSV files. And then created an Azure SQL Database, followed by copying data through pipeline from Storage to SQL Database using Azure Data Factory, below are the screenshots:

Azure Home:



Azure Blob Storage:

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home > Recent > cloudstorage44 | Containers >

cloudproject

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: cloudproject

Search blobs by prefix (case-sensitive)

show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
400_households.csv	11/20/2024, 4:01:51 PM	Hot (inferred)		Block blob	259.18 KB	Available
400_products.csv	11/20/2024, 4:01:52 PM	Hot (inferred)		Block blob	6.29 MB	Available
400_transactions.csv	11/20/2024, 4:02:01 PM	Hot (inferred)		Block blob	122.22 MB	Available
cleaned_households.csv	11/21/2024, 3:56:26 PM	Hot (inferred)		Block blob	155.18 KB	Available
cleaned_products.csv	11/20/2024, 4:57:37 PM	Hot (inferred)		Block blob	4.73 MB	Available
cleaned_transactions.csv	11/20/2024, 4:57:39 PM	Hot (inferred)		Block blob	45.4 MB	Available



Azure SQL Database:

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home > Recent > cloudsql44 (cloudsqlserver44/cloudsql44)

cloudsql44 (cloudsqlserver44/cloudsql44) | Query editor (preview)

Search

Login New Query Open query Feedback Getting started

Overview

Activity log

Tags

Diagnose and solve problems

Query editor (preview)

Mirror database in fabric (preview)

Settings

Data management

Integrations

Power Platform

Security

Intelligent performance

Monitoring

Automation

Help

cloudsql44 (sqladmin)

Showing limited object explorer here. For full capability please click here to open Azure Data Studio.

tables

dbo.households

dbo.products

dbo.transactions

dbo.transformedData

Views

Stored Procedures

Query 1 Query 2

Run Cancel query Save query Export data as Show only Editor Open Copilot

1

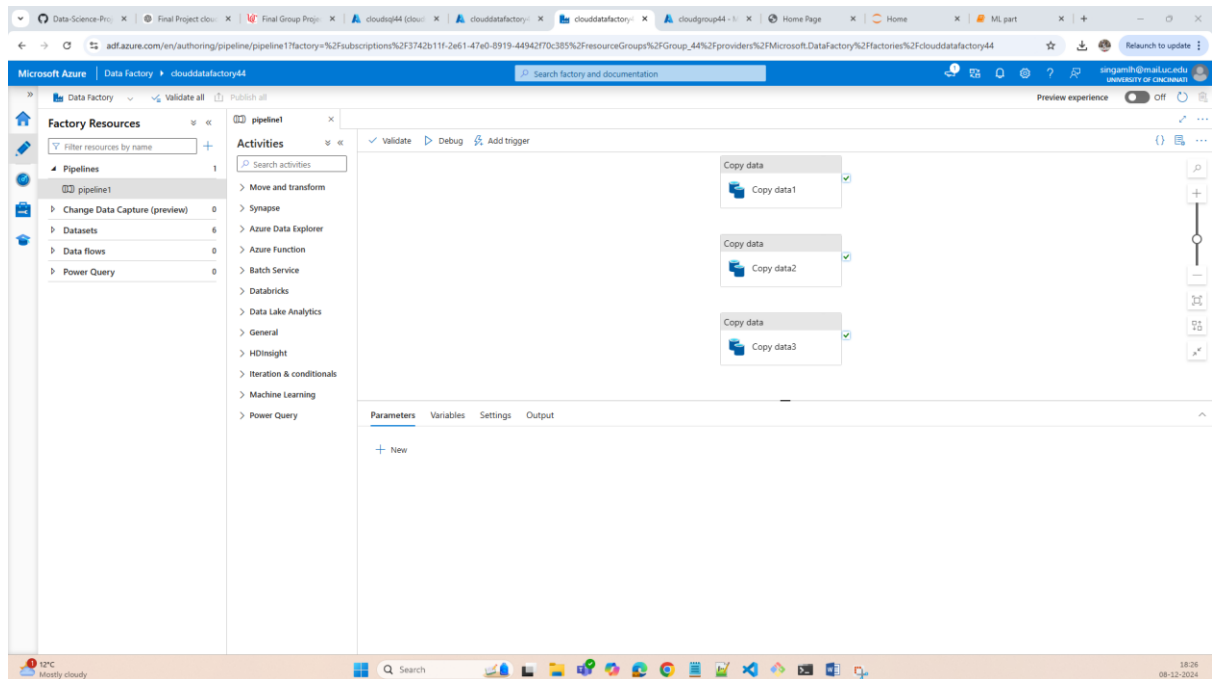
Results Messages

Search to filter items...

Ready



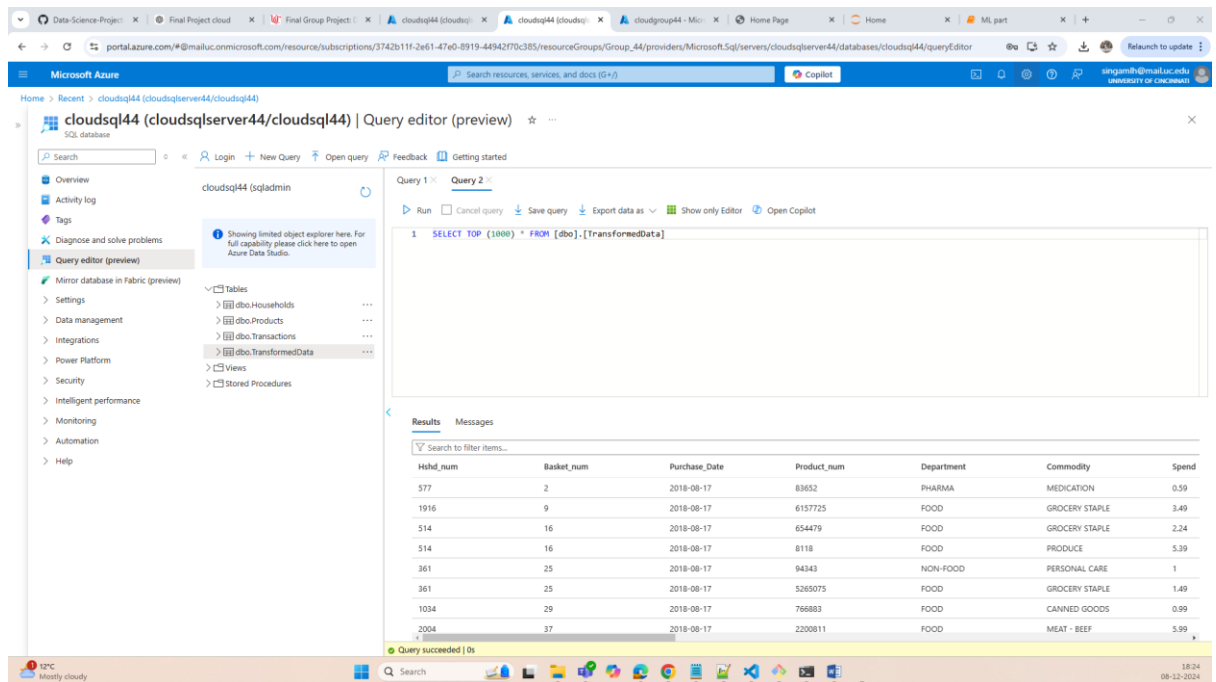
## Azure Data Factory



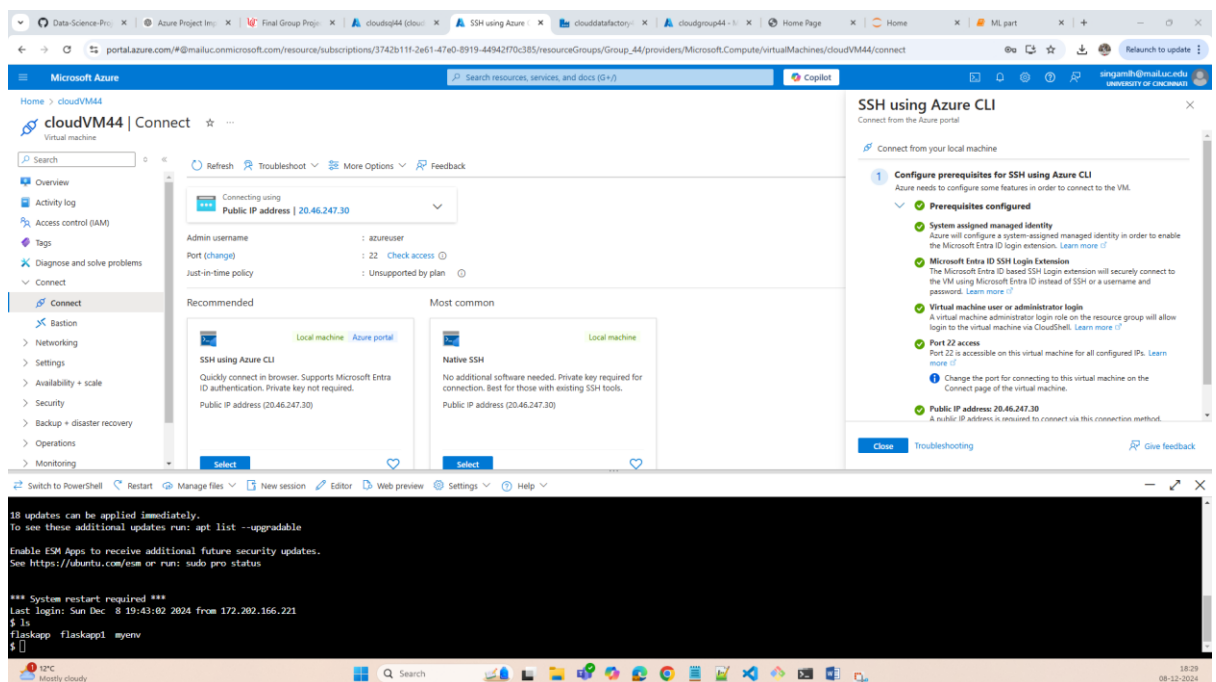
2. Created a TransformedData Table in Azure SQL by joining the 3 tables [Transactions, Products, Households].

Join Query:

```
CREATE TABLE TransformedData AS
SELECT
    t.Hshd_num,
    t.Basket_num,
    t.Purchase_Date,
    t.Product_num,
    p.Department,
    p.Commodity,
    t.Spend,
    t.Units,
    t.Store_region,
    t.Week_num,
    t.Year,
    h.Loyalty_flag,
    h.Age_range,
    h.Marital_status,
    h.Income_range,
    h.Homeowner_desc,
    h.Hshd_composition,
    h.Hshd_size,
    h.Children
FROM Transactions t
JOIN Products p ON t.Product_num = p.Product_num
JOIN Households h ON t.Hshd_num = h.Hshd_num;
```



- Created an Azure VM and developed the flaskapp application, below is the folder structure:



## Folder Structure:

```
$ cd flaskapp1
$ ls
app.py flaskapp.zip model requirements.txt static templates uploads
$ ls -ltr
total 116
-rw-rw-r-- 1 singamlh@mail.uc.edu singamlh@mail.uc.edu 91 Dec 8 18:31 requirements.txt
drwxrwxr-x 2 singamlh@mail.uc.edu singamlh@mail.uc.edu 4096 Dec 8 22:02 model
-rw-rw-r-- 1 singamlh@mail.uc.edu singamlh@mail.uc.edu 12379 Dec 8 22:02 app.py
drwxrwxr-x 2 singamlh@mail.uc.edu singamlh@mail.uc.edu 4096 Dec 8 22:02 static
drwxrwxr-x 2 singamlh@mail.uc.edu singamlh@mail.uc.edu 4096 Dec 8 22:02 uploads
drwxrwxr-x 2 singamlh@mail.uc.edu singamlh@mail.uc.edu 4096 Dec 8 22:02 templates
-rw-rw-r-- 1 singamlh@mail.uc.edu singamlh@mail.uc.edu 81091 Dec 8 22:12 flaskapp.zip
```

4. Created and pushed the flaskapp code to Azure WebApp using Local git and hosted the application in webapp.

## Azure WebApp and Deployment Logs:

The screenshot displays the Azure Portal interface for the 'cloudgroup44' Web App. The left sidebar shows the 'App Services' section with 'cloudgroup44' selected. The main pane shows the 'Deployment Center' with a table of deployment logs.

Time	Commit ID	Commit Author	Status	Message
Sunday, December 8, 2024 (2)				
5:05:07 PM -05:00	ff5aeb	Singameddy	Success (Active)	Final code for kroger application
1:42:35 PM -05:00	ba1ce19	Singameddy	Success	Updated requirements



Azure Webapp Home Page and sample data pull for Household number = 10 and sorted by Hshd\_num, Basket\_num, Date, Product\_num, Department, Commodity:

The screenshot shows the home page of an Azure Webapp. At the top, there's a navigation bar with the 'Kroger' logo. Below it, there's a search bar for a specific household, with '10' entered. To the right of the search bar is a button labeled 'Search'. Further right, there's a section for 'Upload Retail Data' with a button labeled 'Click here to upload data'. Below the search bar, there's a section for 'Predictions for New Customers' with three buttons: 'CLV Prediction', 'Basket Analysis', and 'Churn Prediction'. To the right of this section is a button labeled 'Explore Retail Insights' with a sub-link 'Go to Visual Analysis Dashboard'. The main content area features a table titled 'Sample Data for Household #10'. The table has columns: Hshd\_num, Basket\_num, Purchase\_Date, Product\_num, Department, Commodity, Spend, Units, Store\_region, Week\_num, Year, and Loyalty. The data is sorted by Hshd\_num, Basket\_num, and Purchase\_Date. The table shows 10 rows of data for Household #10. At the bottom of the page, there's a status bar showing the temperature as 12°C and the date as 08-12-2024.

Hshd_num	Basket_num	Purchase_Date	Product_num	Department	Commodity	Spend	Units	Store_region	Week_num	Year	Loyalty
10	281	2018-06-19	163380	FOOD	GROCERY STAPLE	2.29	1	EAST	33	2018	Y
10	281	2018-06-19	248793	PHARMA	MEDICATION	7.99	1	EAST	33	2018	Y
10	281	2018-06-19	985784	FOOD	BAKERY	2.49	1	EAST	33	2018	Y
10	281	2018-06-19	1189945	FOOD	ALCOHOL	12.99	1	EAST	33	2018	Y
10	281	2018-06-19	2213539	FOOD	ALCOHOL	4.49	1	EAST	33	2018	Y
10	281	2018-06-19	5150409	FOOD	DAIRY	2.19	1	EAST	33	2018	Y
10	281	2018-06-19	5290835	FOOD	DELI	5.73	1	EAST	33	2018	Y
10	281	2018-06-19	5290841	FOOD	BULK PRODUCTS	4.05	1	EAST	33	2018	Y

4. Interactive Web Page:

Data Pull for Specific Household 29:

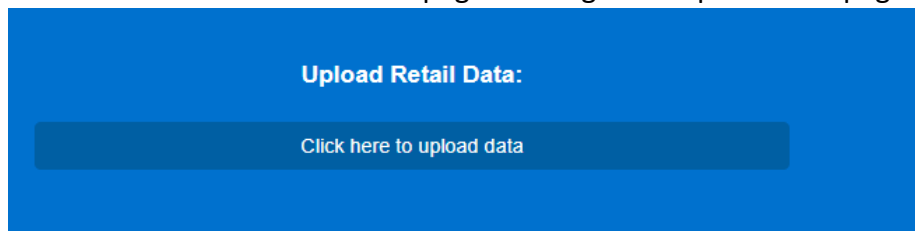
The screenshot shows a search form on a blue background. It has a label 'Search for a specific Household:' above a text input field. The input field contains the number '29'. Below the input field is a button labeled 'Search'.

The screenshot shows the search results page. At the top, there's a navigation bar with the 'Kroger' logo. Below it, there's a section titled 'Search Results'. The section contains a table with columns: Hshd\_num, Basket\_num, Purchase\_Date, Product\_num, Department, Commodity, Spend, Units, Store\_region, Week\_num, Year, and Loyalty. The table shows 7 rows of data for Household 29. Below the table is a button labeled 'Go Back'. At the bottom of the page, there's a status bar showing the temperature as 12°C and the date as 08-12-2024.

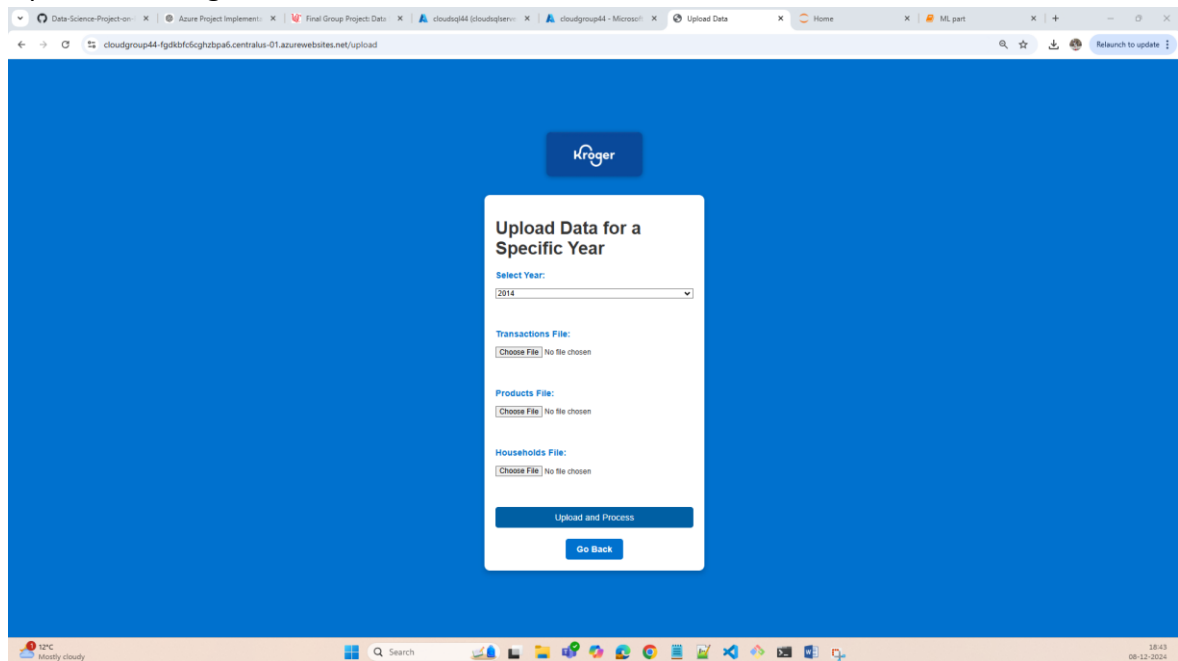
Hshd_num	Basket_num	Purchase_Date	Product_num	Department	Commodity	Spend	Units	Store_region	Week_num	Year	Loyalty
29	25150	2019-03-10	93771	NON-FOOD	HOUSEHOLD	1.00	1	SOUTH	10	2019	Y
29	25150	2019-03-10	101612	FOOD	PRODUCE	1.79	1	SOUTH	10	2019	Y
29	52171	2019-10-19	190985	FOOD	GROCERY STAPLE	3.49	1	CENTRAL	41	2019	Y
29	52171	2019-10-19	1167952	FOOD	GROCERY STAPLE	1.99	1	CENTRAL	41	2019	Y
29	52171	2019-10-19	6438605	FOOD	GROCERY STAPLE	4.69	1	CENTRAL	41	2019	Y
29	52171	2019-10-19	6438686	FOOD	GROCERY STAPLE	6.99	1	CENTRAL	41	2019	Y

## 5. Data Loading WebApp:

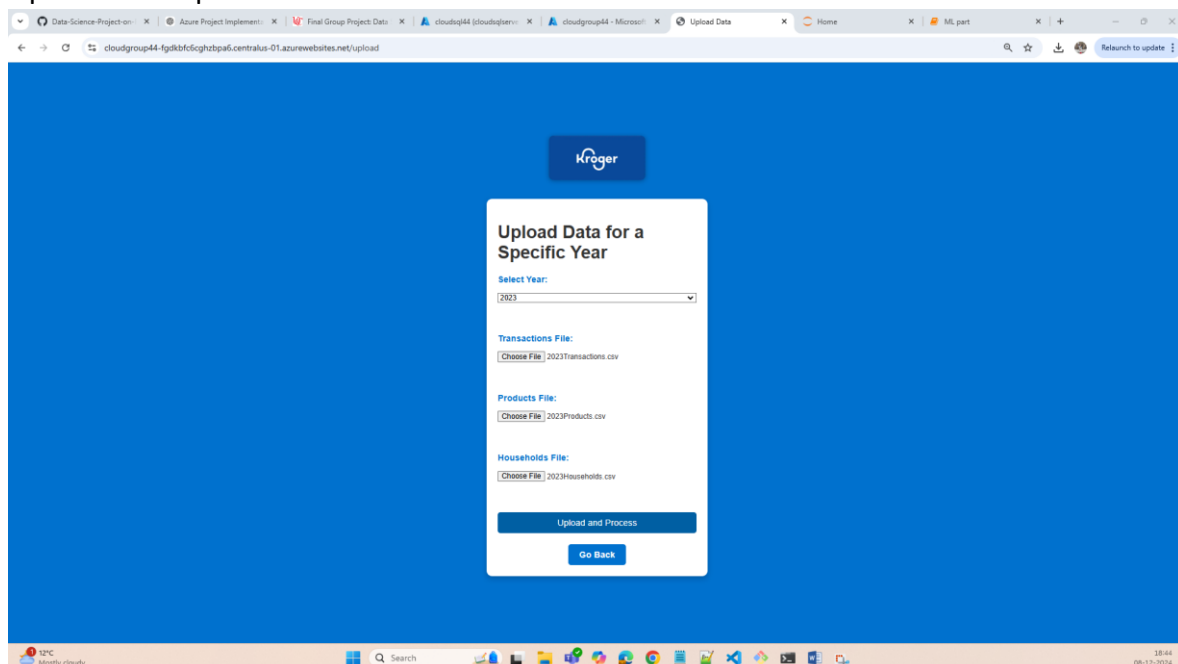
Below is the button from home page to navigate to upload data page:



## Upload data Page:



## Uploaded sample data for Year 2023 for Household number 2:



Searching Uploaded data in Search bar:

Search for a specific Household:

cloudgroup44-fgiktrfcgzhbpa6.centralus-01.azurewebsites.net/search

Kroger

Search Results

Hashd_num	Basket_num	Purchase_Date	Product_num	Department	Commodity	Spend	Units	Store_region	Week_num	Year	Loyalt
2	1126	2018-08-26	1	FOOD	BAKERY	2.67	4	EAST	34	2018	Y
2	2547	2018-09-06	1	FOOD	BAKERY	2.67	4	EAST	35	2018	Y
2	4140	2018-09-20	1	FOOD	BAKERY	4.00	6	EAST	37	2018	Y
2	4392	2018-09-22	1	FOOD	BAKERY	2.00	3	EAST	37	2018	Y
2	5853	2018-10-03	1	FOOD	BAKERY	2.67	4	EAST	39	2018	Y
2	8232	2018-10-22	1	FOOD	BAKERY	4.00	6	EAST	42	2018	Y
2	9472	2018-11-01	1	FOOD	BAKERY	2.00	3	EAST	43	2018	Y
2	10296	2018-11-08	1	FOOD	BAKERY	2.67	4	EAST	44	2018	Y
2	11097	2018-11-11	1	FOOD	BAKERY	2.00	3	EAST	45	2018	Y

Go Back

## 6. Web Page with Dashboard:

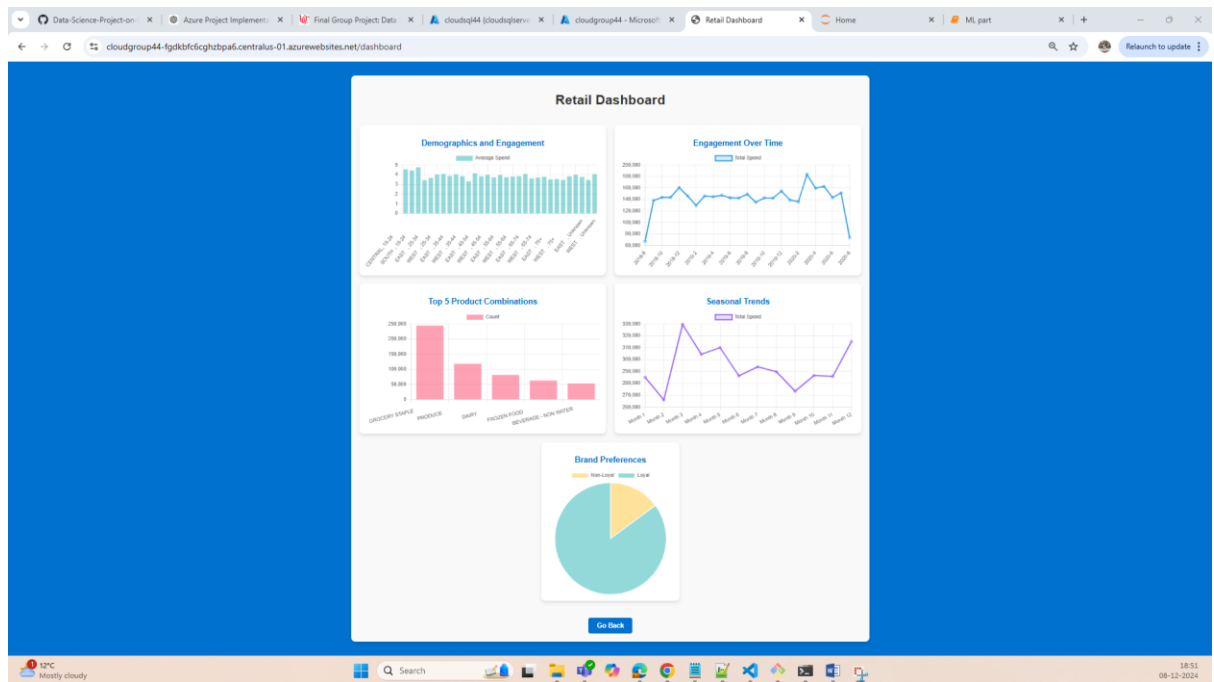
Retail Dashboard which answers possible Retail Questions like:

1. Demographics and Engagement:
2. Engagement Over Time:
3. Basket Analysis:
4. Seasonal Trends:
5. Brand Preferences:

Link from home page to navigate to Retail Dashboard:

Explore Retail Insights:

Go to Visual Analysis Dashboard



**Note: Since the flaskapp has to run 5 database queries with large data in database, it will take few seconds to load the visualization plots as above.**

## 7. ML Model Application:

Procedure followed:

1. Cleaned the given 3 CSV files and saved the cleaned data.
2. Developed ML models for CLV Prediction, Basket Analysis and Churn Prediction and deployed the models in Flask application using .pkl files and predicted results for new customers.

main.ipynb file to pre-processing the data:

```

jupyter main Last checkpoint: 17 days ago
File Edit View Run Kernel Settings Help
+ + + + + Code
Python 3 (ipykernel)

Transactions Missing Values:
BRAND_NAME 0
HOUSE_NUM 0
PURCHASE 0
PRODUCT_NAME 0
SPEED 0
UNITS 0
STORE_A 0
HOUSE_NUM 0
YEAR 0
dtype: object
Products Missing Values:
PRODUCT_NAME 0
DEPARTMENT 0
COMPOSITY 0
BRAND_TY 0
NATURAL_ORGASIC_FLAG 0
dtype: object

[0]:
households.rename(columns=lambda x: x.strip(), inplace=True) # Removes extra spaces
print(households.columns)

In[0]: ['HOUSE_NUM', 'L', 'AGE_RANGE', 'MARITAL', 'INCOME_RANGE', 'HOPEOMER',
        'HOUSE_NUM', 'HOUSE_SIZE', 'CHILDREN']
dtype: object

[0]:
# Handling missing data for Households
# Convert CHILDREN and HOUSE_SIZE to numeric
households['CHILDREN'] = pd.to_numeric(households['CHILDREN'], errors='coerce')
households['HOUSE_SIZE'] = pd.to_numeric(households['HOUSE_SIZE'], errors='coerce')

# Calculate the mean excluding zero values for CHILDREN
children_mean = households.loc[households['CHILDREN'] > 0, 'CHILDREN'].mean()

# Calculate the mean excluding zero values for HOUSE_SIZE
hs_size_mean = households.loc[households['HOUSE_SIZE'] > 0, 'HOUSE_SIZE'].mean()

children_mean = round(children_mean, 2)
hs_size_mean = round(hs_size_mean, 2)

# Replace missing values in CHILDREN and HOUSE_SIZE with their respective means
households['CHILDREN'].fillna(children_mean, inplace=True)
households['HOUSE_SIZE'].fillna(hs_size_mean, inplace=True)

print(f"Mean for CHILDREN (excluding zeros): {children_mean}")
print(f"Mean for HOUSE_SIZE (excluding zeros): {hs_size_mean}")

# Replace missing values in categorical columns with 'Unknown'
households.fillna({
    'AGE_RANGE': 'Unknown',
    'MARITAL': 'Unknown',
    'INCOME_RANGE': 'Unknown',
})

```

ML part.ipynb file to create and save ML models:

```
# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the Gradient Boosting model
model_clv = GradientBoostingRegressor()
model_clv.fit(X_train, y_train)

# Evaluate the model
y_pred = model_clv.predict(X_test)
print("CLV Prediction R2 Score:", r2_score(y_test, y_pred))
print("CLV Prediction MSE:", mean_squared_error(y_test, y_pred))

# Save the model
joblib.dump(model_clv, "gradient_boosting_clv.pkl")

CLV Prediction R2 Score: 0.255353270988446
CLV Prediction MSE: 0.5131383751728904

[9]: ['gradient_boosting_clv.pkl']

[10]: # Step 1: Commodity Frequency
commodity_frequency = df['Commodity'].value_counts()

# Step 2: Identify Top Commodities by Frequency
top_commodities_by_frequency = commodity_frequency.nlargest(5).index.tolist()

# Step 3: Analyze Revenue Contribution
commodity_revenue = df.groupby('Commodity')['Spend'].sum().sort_values(ascending=False)

# Step 4: Identify Top Commodities by Revenue
top_commodities_by_revenue = commodity_revenue.nlargest(5).index.tolist()

# Step 5: Combine Results
# Use union of both top frequency and revenue commodities to get a robust selection
selected_commodities = list(set(top_commodities_by_frequency + top_commodities_by_revenue))

# Display Results
print("Selected Commodities for Basket Analysis:")
print(f"Top Commodities by Frequency: {top_commodities_by_frequency}")
print(f"Top Commodities by Revenue: {top_commodities_by_revenue}")
print(f"Final Selected Commodities: {selected_commodities}")

Selected Commodities for Basket Analysis:
Top Commodities by Frequency: ['GROCERY STAPLE', 'PRODUCE', 'DAIRY', 'FROZEN FOOD', 'BEVERAGE - NON WATER']
Top Commodities by Revenue: ['GROCERY STAPLE', 'PRODUCE', 'FROZEN FOOD', 'DAIRY', 'HOUSEHOLD']
Final Selected Commodities: ['DAIRY', 'GROCERY STAPLE', 'BEVERAGE - NON WATER', 'PRODUCE', 'FROZEN FOOD', 'HOUSEHOLD']

[11]: from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
```

Link from Home page to navigate to respective prediction pages:

### Predictions for New Customers:

CLV Prediction

Basket Analysis

Churn Prediction

Basket Analysis prediction:

### Basket Analysis Prediction

Spend:

5000

Units:

50

Income Range:

1200

Household Size:

2

Children:

0

Store Region:

CENTRAL

Predict

Basket Analysis Prediction: 0

Go Back

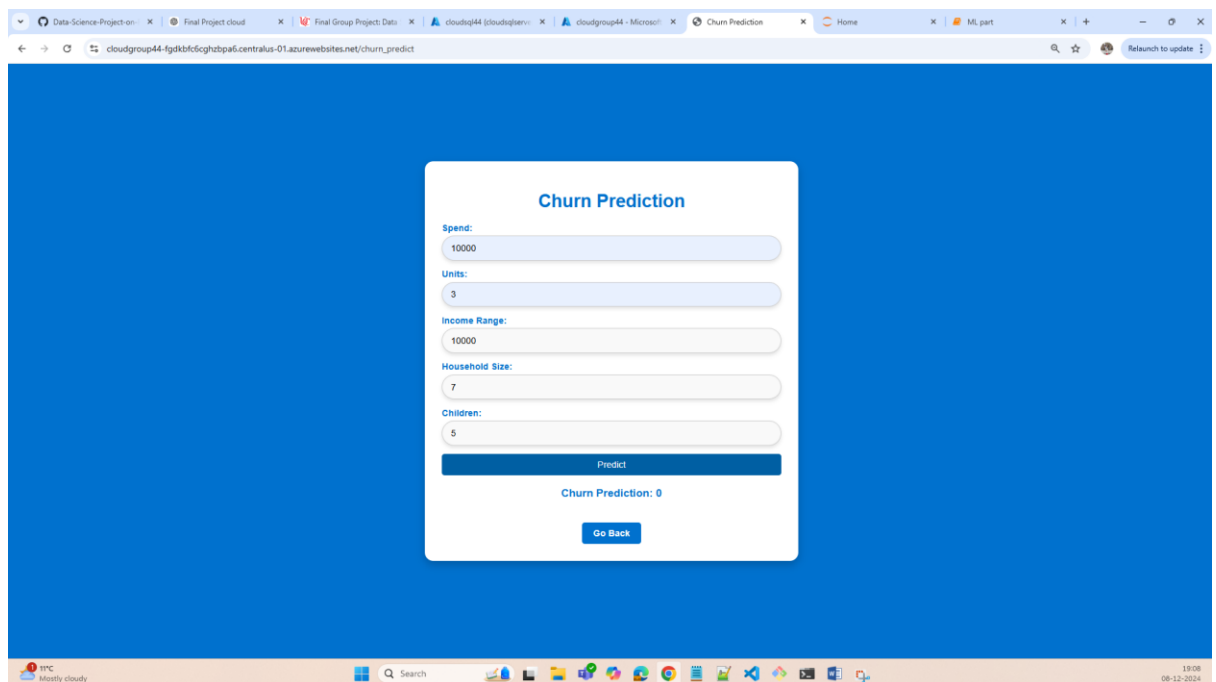
**Retail Question:** What are the commonly purchased product combinations, and how can they drive cross-selling opportunities?

To perform Basket Analysis, we utilized the **Random Forest** model due to its ability to handle high-dimensional data and capture complex relationships between features. Random Forest operates by constructing multiple decision trees and aggregating their results for classification or regression tasks, making it suitable for analysing customer purchasing behaviours.

Using this model, we identified key product combinations frequently purchased together, such as *FROZEN FOOD* and *DAIRY* or *GROCERY STAPLE* and *BEVERAGE - NON-WATER*. These insights can be leveraged to design targeted cross-selling strategies, such as offering discounts on complementary products or bundling items to encourage customers to increase their basket size.

By focusing on these combinations, retailers can drive revenue growth while enhancing customer satisfaction by meeting their needs effectively.

## 8. Churn Prediction:

A screenshot of a web browser displaying a 'Churn Prediction' web application. The browser's address bar shows the URL 'cloudgroup44-fgdkbfscghztpa6.centralus-01.azurewebsites.net/churn\_predict'. The application has a blue background and a white central form. The form is titled 'Churn Prediction' and contains five input fields: 'Spend:' with a value of 10000, 'Units:' with a value of 3, 'Income Range:' with a value of 10000, 'Household Size:' with a value of 7, and 'Children:' with a value of 5. Below these fields is a blue 'Predict' button. Under the button, the text 'Churn Prediction: 0' is displayed. At the bottom of the form is a blue 'Go Back' button. The browser's taskbar at the bottom shows various icons and the system clock indicating 19:08 on 08-12-2024.

**Retail Question:** Which customers are at risk of disengaging, and how can retention strategies address this?

To identify customers at risk of churn, we employed Logistic Regression, a robust and interpretable model for binary classification tasks. The model analysed key features such as spend, purchase frequency, household size, and loyalty flag to predict the likelihood of disengagement.

The analysis revealed patterns indicating high churn risk, such as reduced spending or infrequent purchases. Using correlation analysis, we observed that customers

with lower loyalty scores or smaller basket sizes were more likely to disengage. Graphical representations of churn probabilities further supported these insights.

To address churn, targeted retention strategies can be implemented, such as personalized promotions, loyalty programs, and proactive communication to re-engage at-risk customers. These strategies can improve customer satisfaction, reduce churn rates, and enhance long-term revenue potential.