Analyzing Consumer Behavior in the Grocery Store Industry

Table of Contents

A New Shopping Experience: Foutz's Foods

EST. 2021

FOUTZ'SFOODS

Convenient

One stop-shop for groceries, household items, and other miscellaneous items

Affordable

Lots of discounts and coupons for members!

Family-Friendly

Appropriate for adults and kids of all ages; perfect for families!

Fun

Happy and welcoming environment for all shoppers

Research Questions

EST. 2021
FOUTZ'S FOODS

1

How can we identify which customers are parents using purchase behavior alone?

2

What are the major differences in shopping behavior between parents and non-parents?

3

What subgroups exist within the base of parent customers?

4

What combinations of items do grocery store shoppers tend to purchase together?

Datasets: Customer & Grocery

Customer Personality

Source: Kaggle
Unique values: 2240
Variables: 27 (customer ID, birth year, income, education, kids, $ spent by category)
Purpose: used this dataset to identify and cluster possible consumers for Foutz's Foods

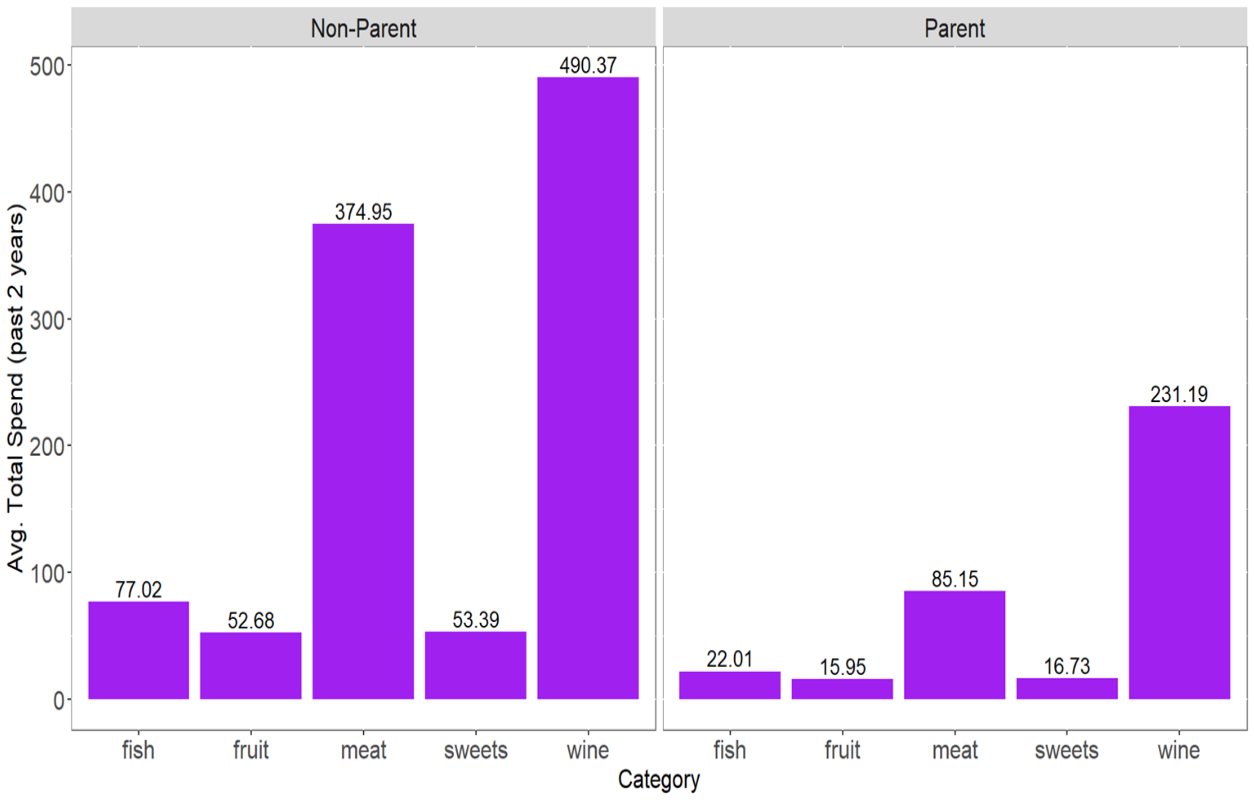
Grocery Products Purchased

Source: Kaggle
Unique Values: 9000+
Variables: 32 different products
Purpose: used to conduct association rule mining to see what consumers purchase together

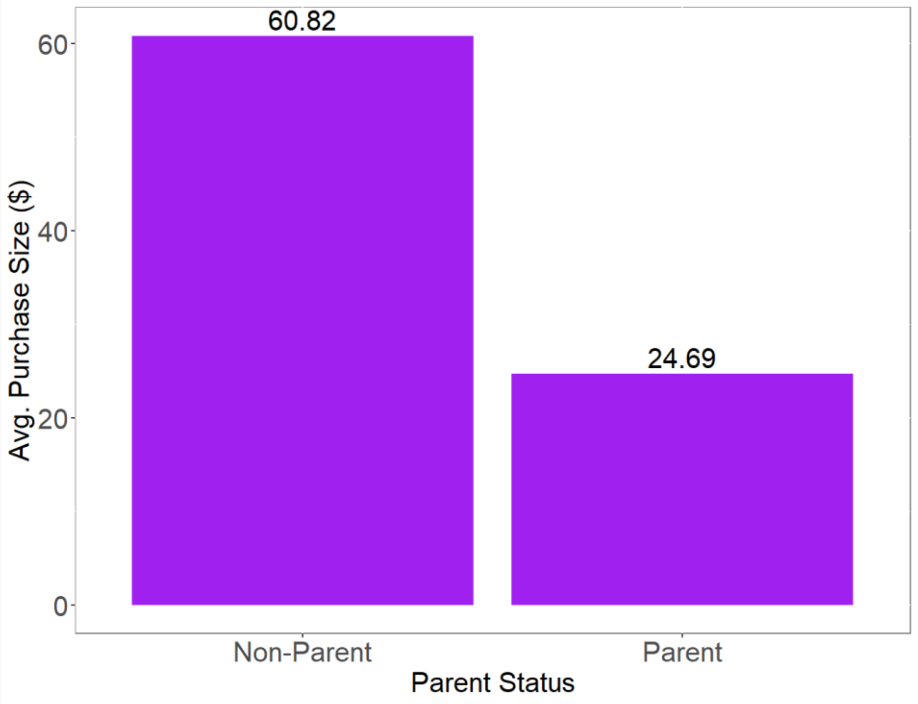
Parents are disproportionately low spenders

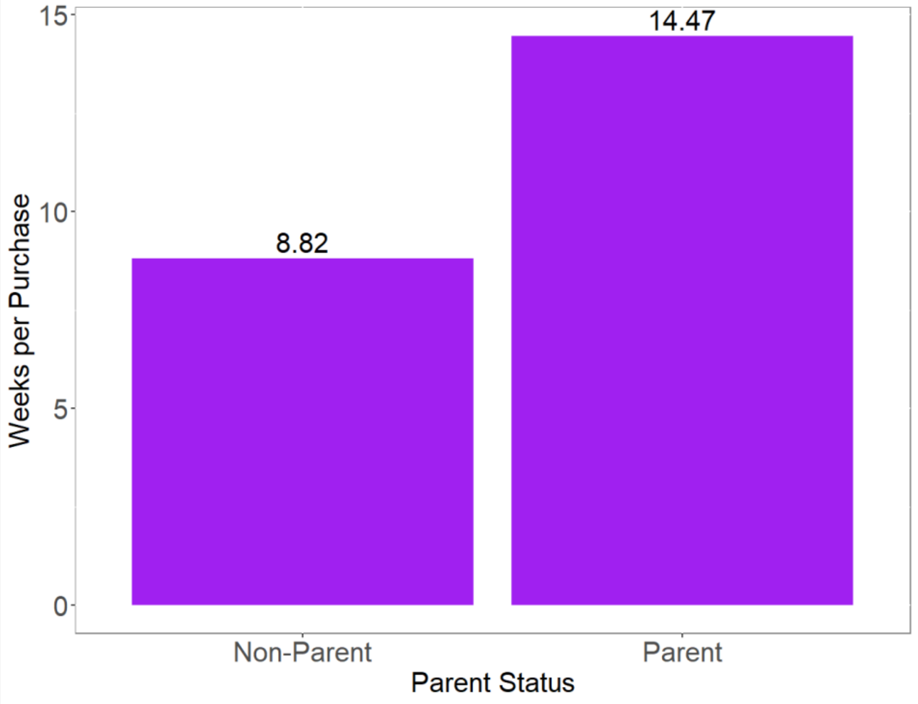
71.4%
of customer base

47.2%
of sales revenue



Parents shop less frequently and spend less per visit





Predictive Modeling

How can we identify which customers are parents?

FOUTZ'SFOODS

EST. 2021

Predictive Modeling

Purpose:

Be able to identify which customers are parents based on purchase behavior alone, and broadly understand what differentiates parents and non-parents.

Goal:

Utilize family-oriented marketing campaigns and loyalty programs to drive up sales from parents.

Data Pre-Processing for Predictive Modeling

```
# Identify parents
dataset["Children"]=dataset["Kidhome"]+dataset["Teenhome"]
dataset["Is_Parent"] = np.where(dataset.Children> 0, 1, 0)

# Calculate monthly spend for products
import datetime as dt
dataset["Dt_customer"] = pd.to_datetime(dataset['Dt_Customer'])
dataset["Month_customer"] = 12 * (2015 - dataset.Dt_customer.dt.year) + (1 - dataset.Dt_customer.dt.month)

dataset['MntWinesMonth'] = dataset['MntWines'] / dataset['Month_customer']
dataset['MntFruitsMonth'] = dataset['MntFruits'] / dataset['Month_customer']
dataset['MntMeatProductsMonth'] = dataset['MntMeatProducts'] / dataset['Month_customer']
dataset['MntFishProductsMonth'] = dataset['MntFishProducts'] / dataset['Month_customer']
dataset['MntSweetProductsMonth'] = dataset['MntSweetProducts'] / dataset['Month_customer']

# Convert total purchases to proportions
dataset["TotalPurchases"] = dataset["NumCatalogPurchases"] + dataset["NumStorePurchases"] + dataset["NumWebPurchases"]
dataset["CatalogProp"] = dataset["NumCatalogPurchases"] / dataset["TotalPurchases"]
dataset["DiscountProp"] = dataset["NumDealsPurchases"] / dataset["TotalPurchases"]
```

1 Identify Parents

2 Convert Units

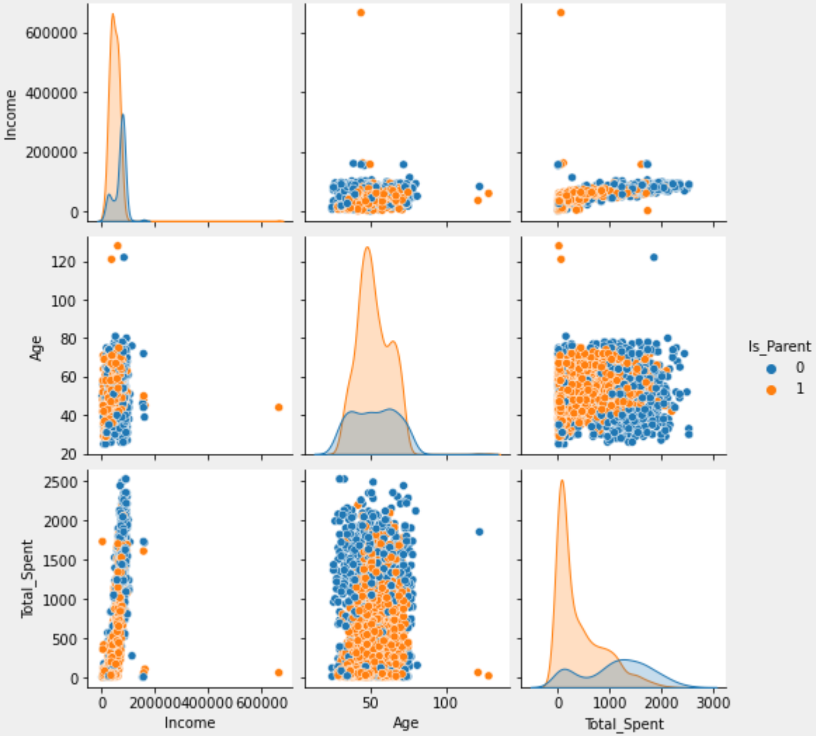
FOUTZ'S FOODS EST. 2021

More Pre-Processing

```
Plot1 = ["Income", "Age", "Total_Spent", "Is_Parent"]
sns.pairplot(dataset[Plot1], hue = 'Is_Parent')

# dropping outliers
dataset = dataset[(dataset['Age']<100)]
dataset = dataset[(dataset['Income']<600000)]
```

3 Analyze Potential Skew

4 Remove Outliers



Predictive Modeling Background

FOUTZ'S FOODS
EST. 2021

Omitted:

Demographic

Personally Identifiable Information

Purpose:

Ease of Prediction: No need for loyalty programs, personal information to analyze parental status

Key Terms to Remember:

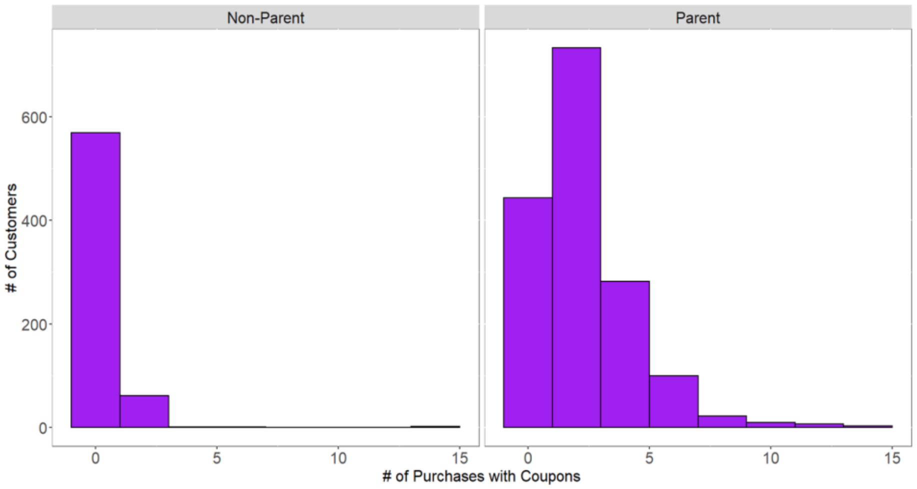
Recall: When the true class is positive, how often is the model's prediction correct?

Precision: When the predicted class is positive, how often is the model's prediction correct?

FOUTZ'S FOODS
EST. 2021

Shopping habits vary by parent status

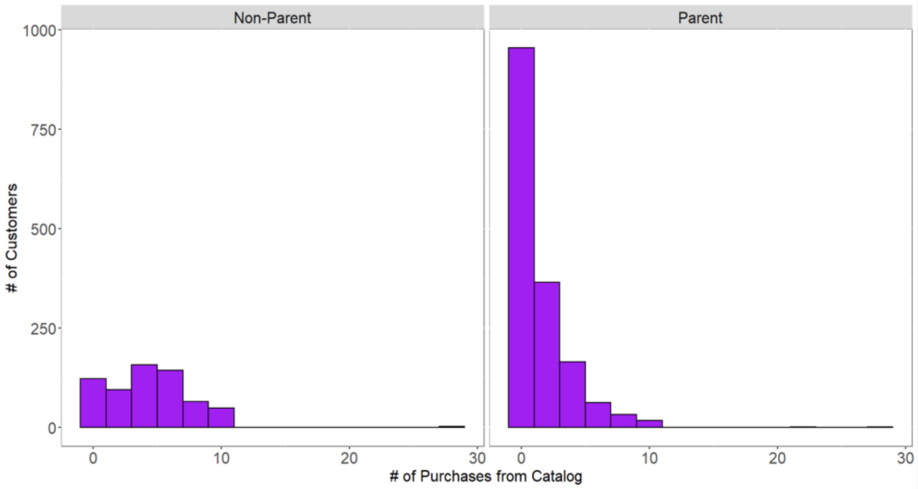
Parents tend to shop when they have coupons



Mean: 1.12 purchases -> 11.1% of purchases

Mean: 2.79 purchases -> 30.2% of purchases

Parents tend to not use catalogs for purchases



Mean: 4.79 purchases -> 26.2% of purchases

Mean: 1.83 purchases -> 12.6% of purchases

FOUTZ'S FOODS

EST. 2021

Code for Predictive Models

```
target1 = ['Is_Parent']
features1 = ['MntWinesMonth', 'MntFruitsMonth', 'MntMeatProductsMonth', 'MntFishProductsMonth', 'MntSweetProductsMonth',
         'CatalogProp', 'DiscountProp']
from sklearn.model_selection import train_test_split
X_train1, X_test1, y_train1, y_test1 = train_test_split(dataset[features1],dataset[target1], test_size= 0.2, random_state = 121)
```

```
# Support Vector Machines

from sklearn.svm import SVC
clf_svc = SVC(kernel = 'rbf', random_state=0)
clf_svc.fit(X_train1, y_train1)

y_pred1 = clf_svc.predict(X_test1)

# Build first model

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test1, y_pred1)
print(cm)
accuracy_score(y_test1, y_pred1)
```

```
[[ 68  64]
 [ 13 298]]

0.8261851015801355
```

TN	FP
FN	TP

Metric Calculations:

Accuracy = (68+298)/(443) = 82.6%
Precision = 298/(298+64) = 82.3%
Recall = 298/(298+13) = 95.8%

EST. 2021 FOUTZ'SFOODS

Model Comparison

4 Total Spend by Category + Prop of Discount Purchases + Prop of Catalog Purchases

Model Type	Variables	Accuracy	Precision	Recall
SVM	Category Purchase $, prop discount purchases, prop catalog purchases	88.71%	90.40%	93.89%
Neural Network	Category Purchase $, prop discount purchases, prop catalog purchases	88.04%	90.57%	92.60%
Decision Tree	Category Purchase $, prop discount purchases, prop catalog purchases	85.78%	93.66%	85.53%
kNN	Category Purchase $, prop discount purchases, prop catalog purchases	86.46%	92.26%	88.10%

5 Monthly Spend by Category + Prop of Discount Purchases + Prop of Catalog Purchases

Model Type	Variables	Accuracy	Precision	Recall
SVM	Category Purchase $ (monthly), prop discount purchases, prop catalog purchases	82.62%	82.32%	95.82%
Neural Network	Category Purchase $ (monthly), prop discount purchases, prop catalog purchases	82.84%	82.02%	96.78%
Decision Tree	Category Purchase $ (monthly), prop discount purchases, prop catalog purchases	83.52%	89.40%	86.82%
kNN	Category Purchase $ (monthly), prop discount purchases, prop catalog purchases	81.72%	84.64%	90.35%

Best model to use depends on objective.
When cost of false positive < opportunity cost of false negative, we should maximize recall.

FOUTZ'S FOODS

Decision Tree Variable Importance

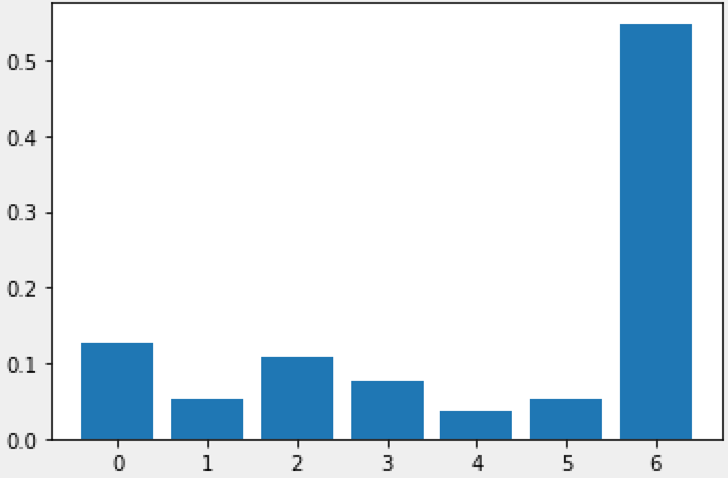
Code Input:

```
from matplotlib import pyplot
importance = clf_dt.feature_importances_
for i,v in enumerate(importance):
	print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

```
Feature: 0, Score: 0.12650
Feature: 1, Score: 0.05342
Feature: 2, Score: 0.10708
Feature: 3, Score: 0.07554
Feature: 4, Score: 0.03695
Feature: 5, Score: 0.05208
Feature: 6, Score: 0.54842
```

We can most easily identify parents by their proportion of deal purchases and wine spend.

Importance Output:



Feature 0: Wine Spend
Feature 1: Fruit Spend
Feature 2: Meat Spend
Feature 3: Fish Spend
Feature 4: Sweets Spend
Feature 5: Prop. of Catalog Purchases
Feature 6: Prop. of Deal Purchases

K-Means Clustering

What subgroups exist within
the parent consumer group?

K-Means Clustering

Purpose:

Identify different customer segments within the parent consumer group

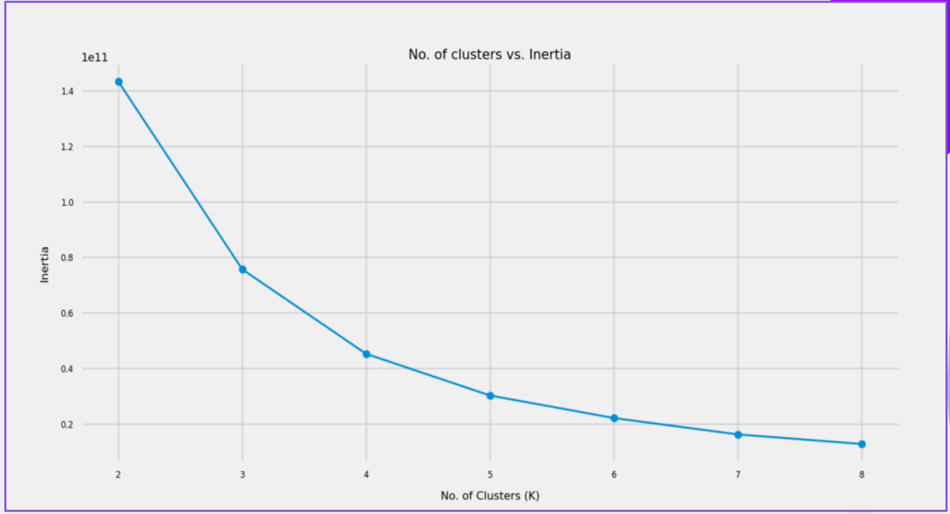
Goal:

Use the analysis from clustered data to create specific marketing decisions for each type of parent

K-Means Clustering w/ Elbow Method

EST. 2021 FOUTZ'S FOODS

```python
# CLUSTERING with Elbow method
X = customer.drop(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Kidhome', 'Teenhome', 'MntWines', 'MntFruits','MntMeatProducts',
                   'MntFishProducts', 'MntSweetProducts', 'MntGoldProds','Dt_Customer', 'Z_CostContact',
                   'Z_Revenue', 'Recency', 'NumDealsPurchases', 'NumWebPurchases','NumCatalogPurchases',
                   'NumStorePurchases', 'NumWebVisitsMonth', 'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5',
                   'AcceptedCmp1', 'AcceptedCmp2', 'Complain',  'Response', 'AgeGroup'], axis=1)

from sklearn.cluster import KMeans

options = range(2,9)
inertias = []

for n_clusters in options:
    model = KMeans(n_clusters, random_state=42).fit(X)
    inertias.append(model.inertia_)

plt.figure(figsize=(16,10))
plt.title("No. of clusters vs. Inertia")
plt.plot(options, inertias, '-o')
plt.xticks( fontsize=12)
plt.yticks( fontsize=12)
plt.xlabel('No. of Clusters (K)', fontsize= 16, labelpad=16)
plt.ylabel('Inertia', fontsize=16, labelpad=16);
plt.show() #insight: looking at the plot, the inertia value does not  decrease much after that; 5 could also be an option
model = KMeans(n_clusters=4, init='k-means++', random_state=42).fit(X)

preds = model.predict(X)

customer_kmeans = X.copy()
customer_kmeans['clusters'] = preds
```

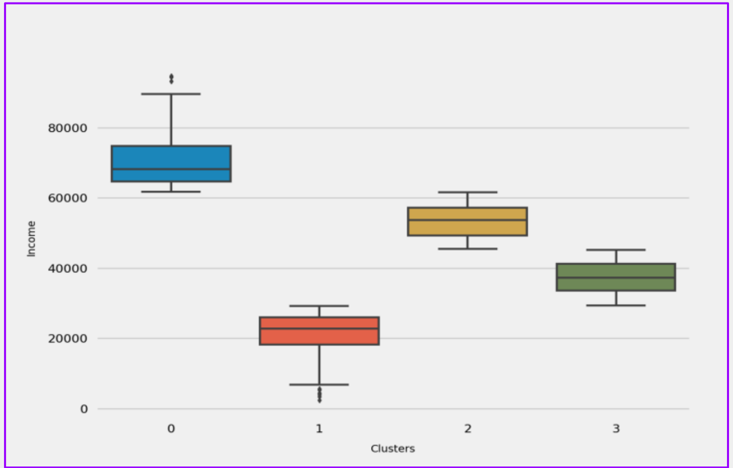


In Python, we used sklearn.cluster from the KMeans library to conduct the clustering. We created a for loop called n_clusters to create a model that would help us identify how many clusters to use.

Using the Elbow Method, we found the optimal number of clusters to use is four, because the inertia value slowly stops decreasing there.

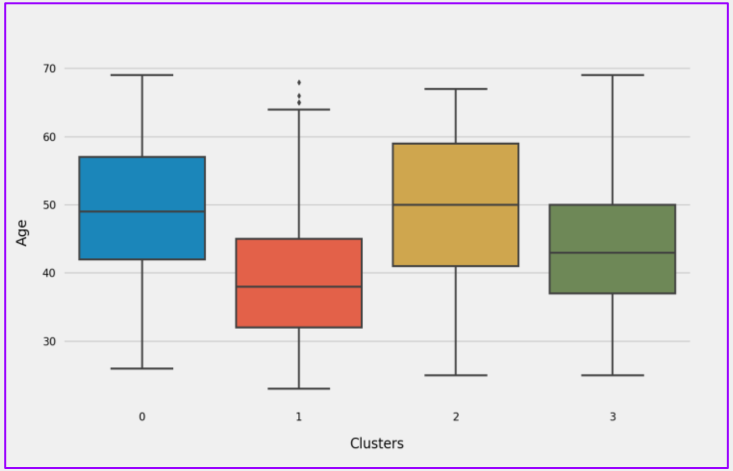
Parent Cluster Characteristics

Income




```python
## Plot 1: Looks at the 4 clusters by income
plt.figure(figsize=(18,10))

sns.boxplot(data=customer_kmeans, x='clusters', y = 'Income');
plt.xlabel('Clusters', fontsize=14, labelpad=14)
plt.ylabel('Income', fontsize=14, labelpad=14);
```

Age

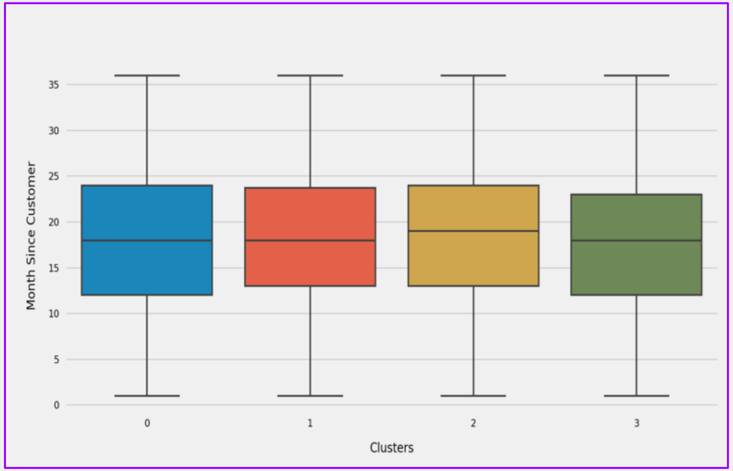



```python
## Plot 4: Looks at the 4 clusters by age
plt.figure(figsize=(20,10))

sns.boxplot(data=customer_kmeans, x='clusters', y = 'Age');
plt.xlabel('Clusters', fontsize=20, labelpad=20)
plt.ylabel('Age', fontsize=20, labelpad=20);
```

EST. 2021 FOUTZ'S FOODS

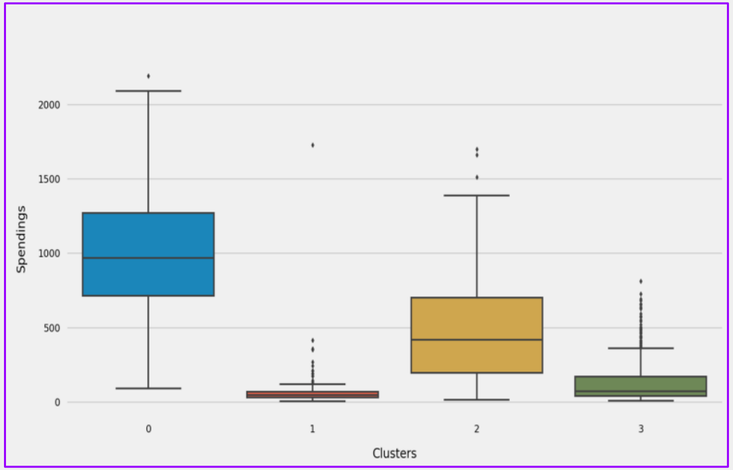
Parent Cluster Characteristics

Months as Customer




```
## Plot 3: Looks at the 4 clusters by month since customer
plt.figure(figsize=(20,10))

sns.boxplot(data=customer_kmeans, x='clusters', y = 'Month_Customer');
plt.xlabel('Clusters', fontsize=20, labelpad=20)
plt.ylabel('Month Since Customer', fontsize=20, labelpad=20);
```

Total Spending




```
## Plot 2: Looks at the 4 clusters by spending habits

plt.figure(figsize=(20,10))
sns.boxplot(data=customer_kmeans, x='clusters', y = 'TotalSpendings');
plt.xlabel('Clusters', fontsize=20, labelpad=20)
plt.ylabel('Spendings', fontsize=20, labelpad=20);
```

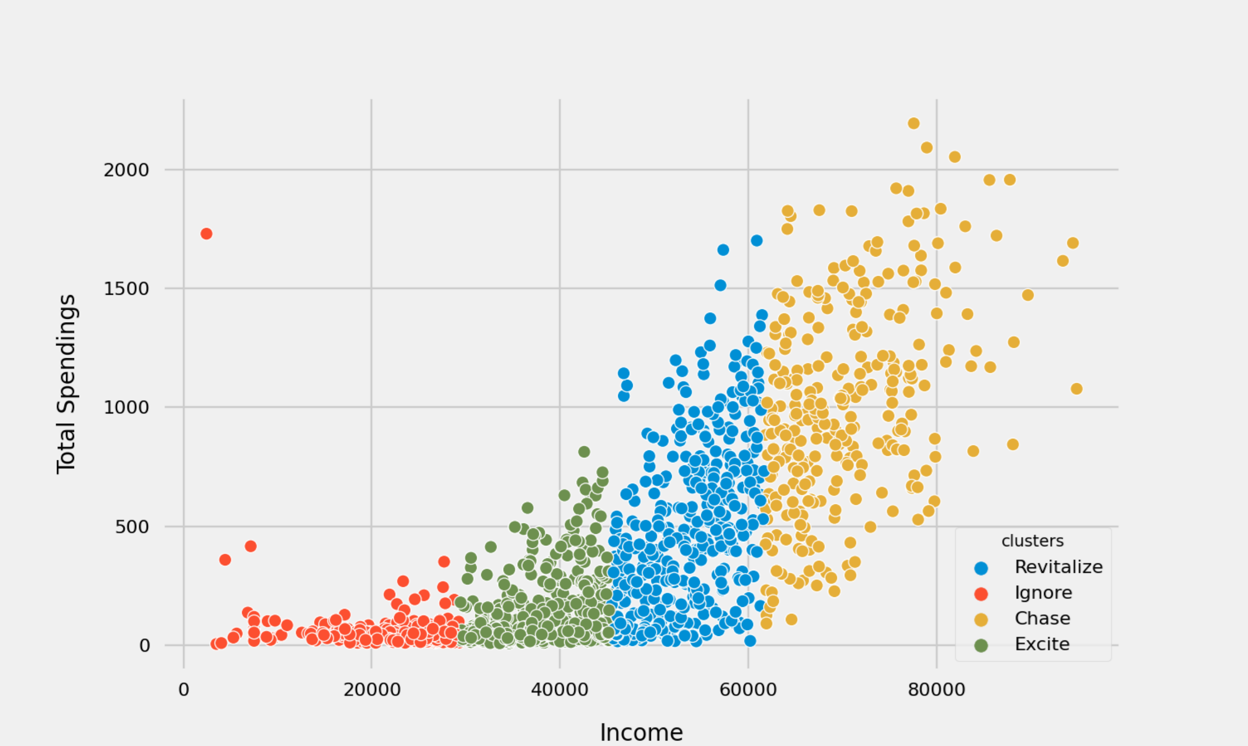
EST. 2021
FOUTZ'SFOODS

Cluster Identification

Cluster	Strategy Label	Characteristics
0	Chase	1 child, average to high income, high spending, middle-age
1	Ignore	1 child, low income, little to no spending, millennials
2	Revitalize	1-3 children, average income, average spending, middle-age
3	Excite	1-3 children, low to average income, low spending, between millennials and middle-age

Total Spending by Clusters

FOUTZ'S FOODS

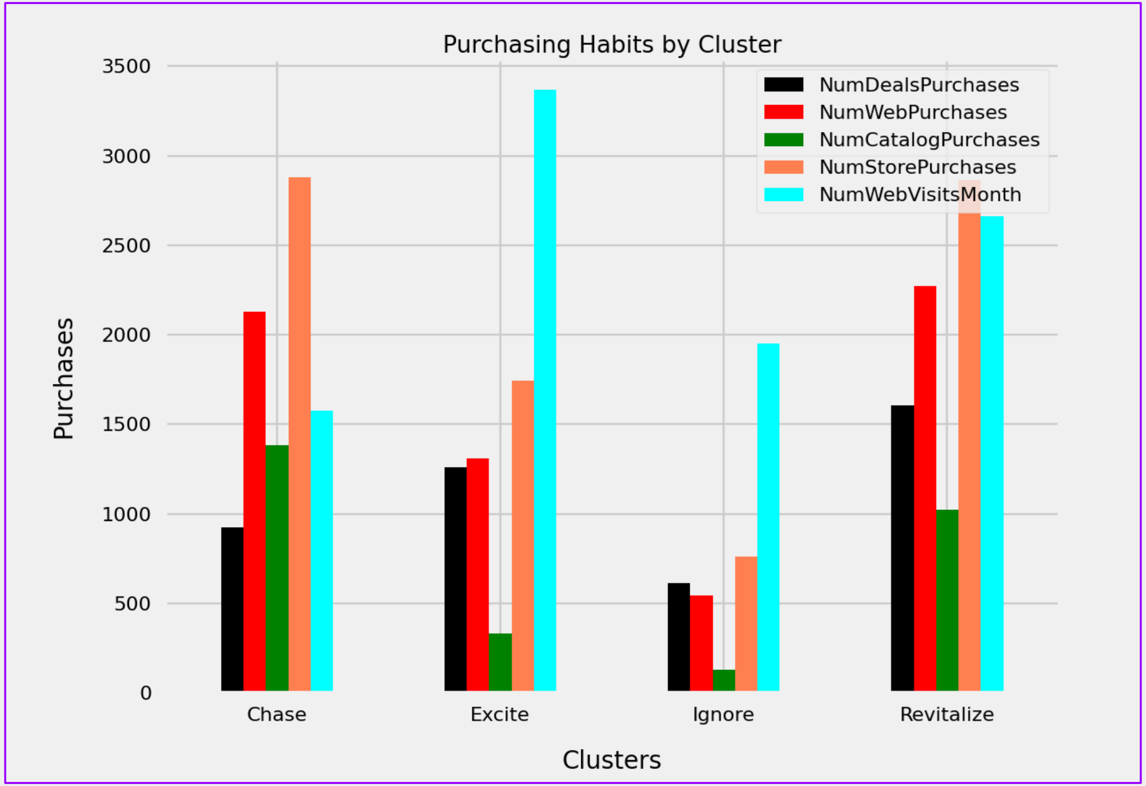


Insights:

Cluster 0 (Chase) spends the most as a collective group, hence, we will target these customers as they are a key revenue source

Cluster 3 (Ignore) spends the least and has overall lowest income, therefore, we will not target them as potential customers

Purchasing Habits by Cluster

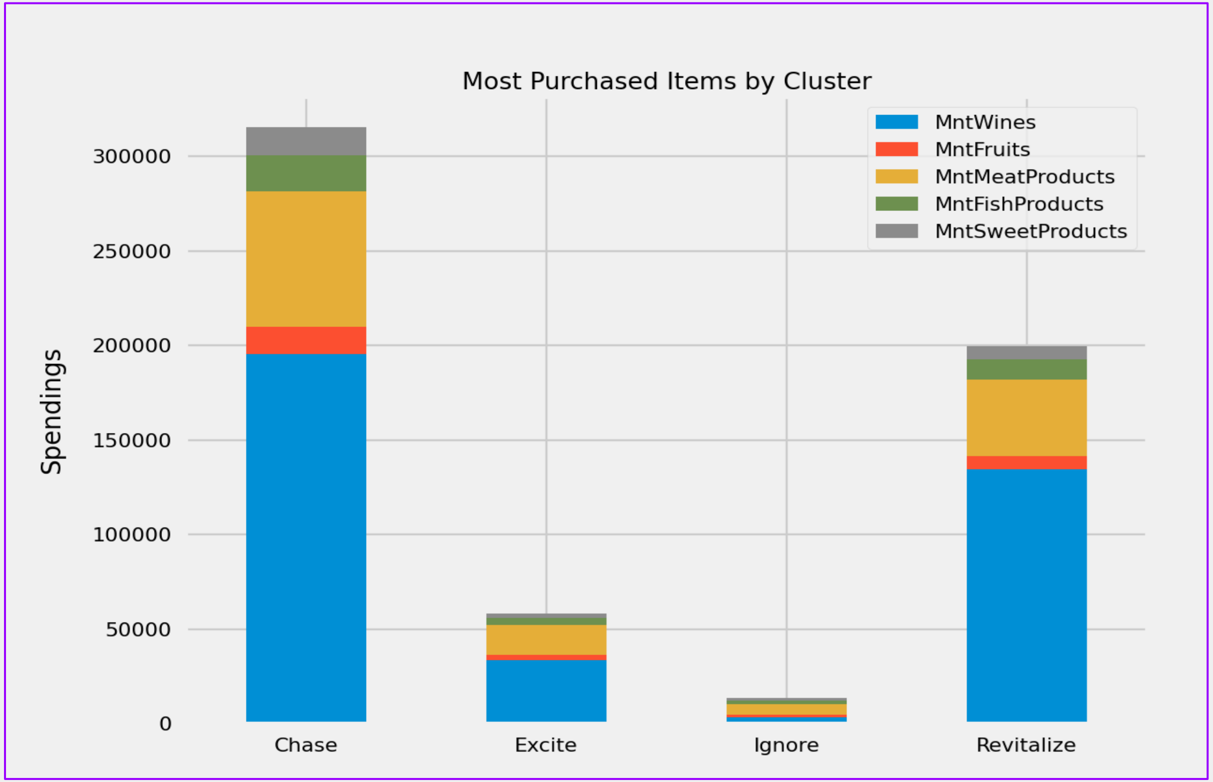


Insights:

Cluster 3 (Excite) tends to visit the website most frequently but does not purchase often

Clusters 0 and 2 (Chase and Revitalize) make the most in-store purchases

Most Purchased Items by Cluster

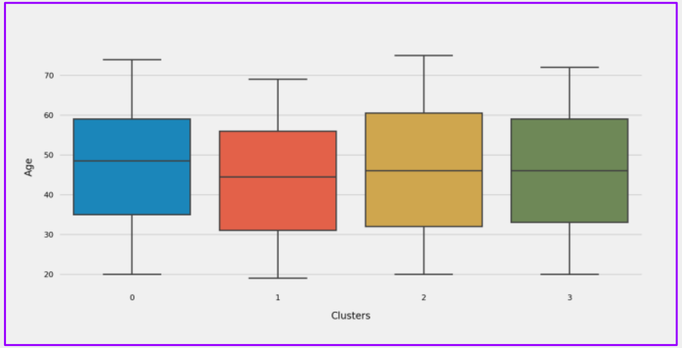


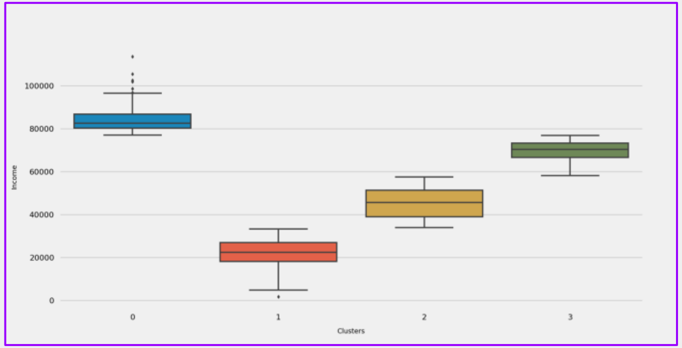
Insights:

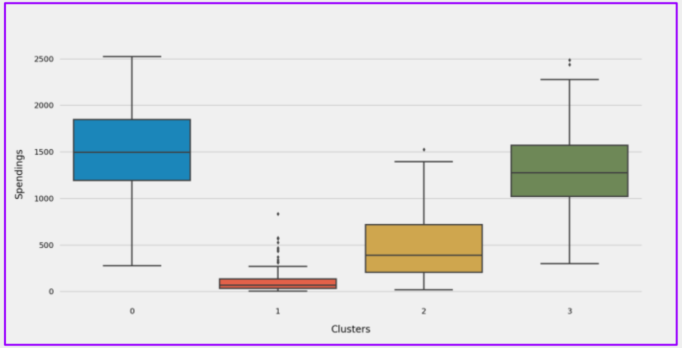
Clusters 0, 2 and 3 (Chase, Excite and Revitalize) have a high proportional wine spend to other product categories

Meat is the second highest spend item driving revenue

Non-Parent Clusters







Non-Parent Clusters and Identification

0: high income, high spending - CHASE

1: low income, little to no spending - IGNORE

2: average income, low spending - INSPIRE

3: average/high income, average spending - REVITALIZE

Association Rule Mining

What Products Do Consumers Often Buy Together?

Association Rule Mining

Purpose:

Identify types of products that are typically purchased together.

Goal:

Pair frequently bought items together, and personalize shopping experience by recommending new items based on purchase history

Association Rule Mining

Support Values: explains how popular an itemset is

P(A and B) -> % of total transactions with both A and B

Confidence: indicates how often the rule is found to be true

P(B|A) -> % of transactions containing A that also contain B

Lift: The ratio of the observed support to that expected if X and Y were independent

Conviction: the frequency that the rule predicts incorrectly

Leverage: leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y were statistically dependent

FOUTZ'S FOODS

EST. 2021

Association Rule Mining Examples

Support = # of transactions with both A and D ÷ All transactions

Confidence = # transactions with A and D ÷ transactions with A

<u>Support:</u>

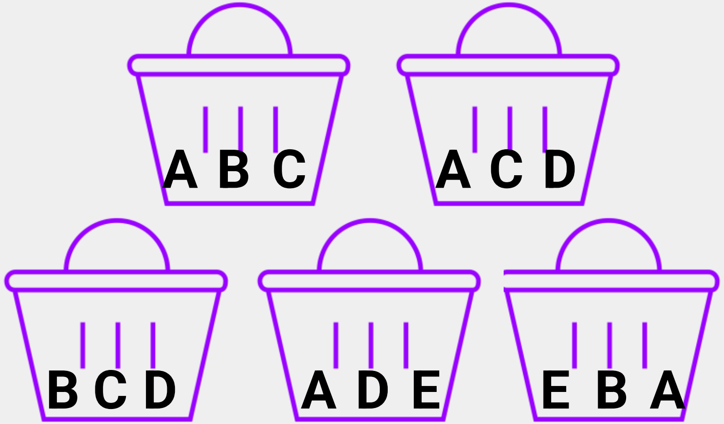
A → B: 2/5 (0.40)

A → C: 2/5 (0.40)

A → D: 2/5 (0.40)

A→ E: 2/5 (0.40)

B & C → D: 1/5 (0.20)



<u>Confidence</u>

A → B: 2/4 (0.50)

A → C: 2/4 (0.50)

A → D: 2/4 (0.50)

A→ E: 2/4 (0.50)

B & C → D : 1/2 (0.50)

EST. 2021
FOUTZ'S FOODS

Code for Association Rule Mining

Python code to establish Support Values:

```python
def ar_iterations(data, num_iter = 1, support_value = 0.1, iterationIndex = None):

    # Next Iterations
    def ar_calculation(iterationIndex = iterationIndex):
        # Calculation of support value
        value = []
        for i in range(0, len(iterationIndex)):
            result = data.T.loc[iterationIndex[i]].sum()
            result = len(result[result == data.T.loc[iterationIndex[i]].shape[0]]) / data.shape[0]
            value.append(result)
        # Bind results
        result = pd.DataFrame(value, columns = ["Support"])
        result["index"] = [tuple(i) for i in iterationIndex]
        result['length'] = result['index'].apply(lambda x:len(x))
        result = result.set_index("index").sort_values("Support", ascending = False)
        # Elimination by Support Value
        result = result[result.Support > support_value]
        return result
```

```python
# First Iteration
first = pd.DataFrame(df.T.sum(axis = 1) / df.shape[0], columns = ["Support"]).sort_values("Support", ascending = False)
first = first[first.Support > support_value]
first["length"] = 1

if num_iter == 1:
    res = first.copy()

# Second Iteration
elif num_iter == 2:

    second = list(itertools.combinations(first.index, 2))
    second = [list(i) for i in second]
    res = ar_calculation(second)

# All Iterations > 2
else:
    nth = list(itertools.combinations(set(list(itertools.chain(*iterationIndex))), num_iter))
    nth = [list(i) for i in nth]
    res = ar_calculation(nth)

return res
```

Python code to establish Confidence:

```python
freq_items = apriori(df, min_support = 0.1, use_colnames = True, verbose = 1)
freq_items.sort_values("support", ascending = False)
```

```python
from mlxtend.frequent_patterns import association_rules

rules_ap = association_rules(freq_items, metric="confidence", min_threshold=0.8)
rules_fp = association_rules(freq_items, metric="confidence", min_threshold=0.8)
```

```python
df_ar = association_rules(freq_items, metric = "confidence", min_threshold = 0.5)
df_ar
```

```python
df_ar[(df_ar.support > 0.15) & (df_ar.confidence > 0.5)].sort_values("confidence", ascending = False)
```

To conduct ARM, we used a machine-learning algorithm called Apriori to identify individual items in a consumer's cart and whether they are in a reoccurring set

EST. 2021 FOUTZ'SFOODS

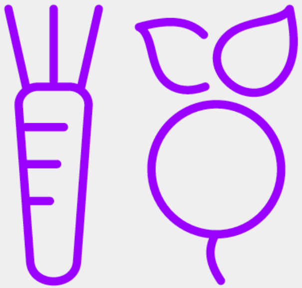
Association Rule Mining Results

After conducting our analysis, we found that most products are paired to whole milk and non-root vegetables

Antecedents	Consequents	Support	Confidence	Conviction	Lift
(root vegetables)	**(whole milk)**	**0.001017**	**0.714286**	**2.605694**	**2.795464**
(bottled beer, bottled water)	(whole milk)	0.001017	0.714286	2.605694	2.795464
(white bread, eggs)	(other vegetables)	0.001017	0.625000	2.150686	3.230097
(bottled water)	(other vegetables)	0.001017	0.588235	1.958661	3.040091
(soda)	**(yogurt)**	**0.001017**	**0.526316**	**1.816607**	**3.772825**

Key Takeaways from Results

Association Rule Mining allowed us to explore grocery store data to identify frequently paired items



Root vegetables and whole milk are the most frequent itemsets

Healthy items are often paired with non-healthy products

Frequent itemsets should feature promotional offerings

Tying It All Together

FOUTZ'S FOODS
EST. 2021

Strategic Insights Drawn

Need For Visits

Increased # of visits from parents will drive segment growth

Coupon Effectiveness

Strategically timed insights will appease parents

Need For Spend

Increased spend through targeted promotions as an essential motivator

Meet Lisa. She's been a customer for almost a year, but recently has stopped visiting us.

What should we do?

Step 1: Consult Predictive Model

Insights/Result: Due to a high level of purchases with coupons, high wine spend relative to other categories, and no purchases from catalogs, our model predicts that Lisa is a parent.

Action Taken: Invite Lisa to join Foutz's Parents Loyalty Program

Now, we have access to demographic information, and can further personalize the experience.

Step 2: Determine Lisa's Parent Cluster

Insights/Result: Lisa is 35 and a mother of 3. We suspect her income is average or lower based on her low spend and reliance on coupons.

Action Taken: Assign Lisa to the "Excite" cluster

Now, Lisa will receive promotional materials and coupons specific to this group.

Step 3: Regularly Engage to Create Personalized Experience

Insights/Result: Lisa regularly purchases whole milk and fruit. Following our ARM rules, we suspect Lisa would be likely to buy root vegetables.

Action Taken: Send Lisa "Foutz's Fun Family Foods" newsletter with easy family-friendly recipes. Provide a coupon to encourage Lisa to buy root vegetables for one of the recipes.

Now, we have made it convenient for Lisa to come visit us and get everything her family needs.

Example: Back to School Campaign

One of many ways to drive parents into Foutz's Foods

EST. 2021 FOUTZ'S FOODS

Re-Schooling Rachel

Business Case: Rachel is a 40-year old mom of 2 taking her kids back to school for the year, how do we get her to complete back to school shopping at Foutz's Foods?

Research Implementation: We know this mom of 2 has a high wine spend and likes to use coupons - let’s get to her in one of her favorite magazines with a coupon!

EST. 2021
FOUTZ'S FOODS

Combo Carrie

Business Case: Carrie is a 50 year old mom of two college boys that attend school nearby each other. On top of tuition expenses, Carrie has to worry about providing her children with school supplies, groceries, and other necessities. She wants to take advantage of store coupons for a bigger discount across a wide range of products. How do we make Foutz's Foods their one stop grocery shop?

Research Implementation: We know that through our clustering and ARM implementations, we can find methods to target Carrie not only on personal preferences but also for products they often purchase together. By using our ARM analysis, we can provide joint discounts for items frequently purchased together to get customers like Carrie into our store.

Front

Back

FOUTZ'SFOODS

Limitations and Next Steps

Limitations

- Our data pre-dates COVID, so it may not fully represent current customer behavior - likely undersells the importance of online presence
- Most consumers in the dataset are middle-aged, and consumer trends may change as more millennial and gen Z consumers become parents

Next Steps

- Implement a Test & Learn framework to determine optimal frequency with which to engage different clusters
- Utilize A/B testing to identify coupon features that lead to highest conversion rate and attributed sales
- Build out tiered loyalty program to reward parents for shopping more frequently and at greater volumes

EST. 2021

FOUTZ'S FOODS

Questions?

Appendix

FOUTZ'SFOODS

Don’t Care Dan

Business Case: Dan is a 35 year old father of one. He is often sent to run errands in his free time and isn’t very educated on where to go to fulfill these errands or where the best shopping locations are. How do we get him to walk into Foutz's Foods when he drives by?

Research Implementation: We know Dan needs to associate discounts with Foutz's Foods to make a judgement about going in. A banner hanging on the storefront will successfully notify him, and others, about the deals inside and prompt him to come to us over our nearby competitors.

Sources

https://www.kaggle.com/imakash3011/customer-personality-analysis

https://www.kaggle.com/ekrembayar/apriori-association-rules-grocery-store

https://www.kaggle.com/mariekaram/apriori-association-rule/notebook

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier.feature_importances_

https://scikit-learn.org/stable/modules/neural_networks_supervised.html

https://thecleverprogrammer.com/2021/02/08/customer-personality-analysis-with-python/

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://machinelearningmastery.com/calculate-feature-importance-with-python/