Exploring Customer Loyalty Prediction through Behavioral Segmentation and Classification Models

Luisa Alejandra Sierra Guerra (30261956)

University Of Calgary

Faculty Of Science

Master’s in data science And Analytics

Course: DATA 605

Spring 2025

Contents

Introduction

In today’s competitive market, business success depends not only on operational efficiency and strategic positioning but also on the ability to understand and respond to consumer demand. Identifying the right customer segments and fostering loyalty are key to long-term growth and profitability.

Customer segmentation refers to the process of dividing a broader customer base into smaller, more homogeneous groups based on shared characteristics such as demographics, behavior, or preferences. Customer loyalty, in turn, refers to the consistent repurchase of a brand’s products or services over time and is often linked to higher customer lifetime value and reduced churn. Understanding both concepts is essential for designing targeted and effective marketing strategies.

The main objective of this project is to “APPLY MACHINE LEARNING TECHNIQUES TO **CLASSIFY** LOYAL AND NON-LOYAL CUSTOMERS BASED ON INDIVIDUAL PROFILES AND OBSERVED BEHAVIORS”. The analysis will leverage a dataset comprising variables such as age, gender, income, purchase frequency, membership duration, and spending behavior. The study will involve two main approaches: *clustering* methods to identify distinct customer segments, and *classification models*. Specifically *logistic regression* and *decision trees* to predict customer loyalty. The performance of these models will be compared to determine the most effective approach for loyalty prediction.

The goal is to determine the most accurate model for predicting customer loyalty while generating actionable insights to support strategic decision-making and improve customer retention.

Analysis Questions

Main Objective: Classify loyal and non-loyal customers based on individual demographic profiles and observed behavioral patterns using machine learning techniques.

EDA Objectives and Questions

1. **Obj**: Identify behavioral patterns based on customer membership duration.

   1.1 Q: What trends in purchase frequency are associated with the number of membership years?

   This analysis aims to understand how loyalty may evolve over time and helps define thresholds for classifying loyal vs. non-loyal customers

Statistical Data Objectives and Questions

2. **Obj**: Assess the relationship between preferred shopping category and spending score.

   2.1 Q: Is there a significant relationship between a customer's preferred category and their spending score?

   Understanding the link between preferred shopping categories and spending scores is crucial because different categories may have varying price ranges and purchase frequencies, which can influence customer loyalty indicators. For example, customers who prefer high-value categories like Electronics might naturally have higher spending scores than those who prefer everyday categories like Groceries. Identifying these patterns helps in detecting potential multicollinearity issues that could bias predictive models.

Machine Learning Objectives and Questions

3. **Obj**: Analyze how PCA can reduce the number of numerical features while preserving 90% of the dataset's variance.

   3.1 Q: Can PCA effectively reduce the feature space while retaining at least 90% of the variance?

4. **Obj**: Evaluate whether customer clusters contribute to improved classification performance.

   4.1 Q: Do clusters derived from customer profiles provide meaningful segments that enhance the predictive power of the classification model?

   Clustering can reveal underlying behavioral segments that are useful as new features for classification.

5. **Obj**: Compare the predictive performance of logistic regression and decision tree models.

   5.1 Q: Which classification technique (logistic regression or decision tree) achieves higher accuracy in predicting customer loyalty?

   This comparison will identify the most effective approach for operationalizing loyalty prediction.

Data Sourcing and Justification

The dataset used for this project was obtained from Kaggle website: https://www.kaggle.com/datasets/fahmidachowdhury/customer-segmentation-data-for-marketing-analysis?resource=download. This dataset is Licensed under CC0: Public Domain.

It contains 9 columns and 1,000 observations. However, only 8 features were used in the analysis, excluding the id column as it serves solely as a unique identifier and does not provide relevant analytical value. The remaining variables are described in Table 1 below.

Table 1.

Description of Variables in the Dataset

Variable Name	Description
id	Unique identifier for each customer (Discrete)
age	Age of the customer (Continuous)
gender	Gender of the customer (Categorical: Female, Male, Other)
income	Annual income of the customer in dollars (Continuous)
spending_score	Internal score representing the customer's spending behavior (Continuous)
membership_years	Number of years the customer has been a member (Discrete)
purchase_frequency	Average number of purchases in the last year (Continuous)
preferred_category	Most frequently purchased product category (Categorical)
last_purchase_amount	US Dollar amount spent in the customer's last purchase (Continuous)

This dataset was selected because it directly supports the main objective of the project: to classify loyal and non-loyal customers using machine learning techniques based on demographic profiles and observed behavioral patterns. It includes a balanced set of features, demographic (e.g., age, gender), behavioral (purchase_frequency, membership_years), and transactional (income, spending_score, last_purchase_amount), This allows robust classification, clustering, and dimensionality reduction analysis.

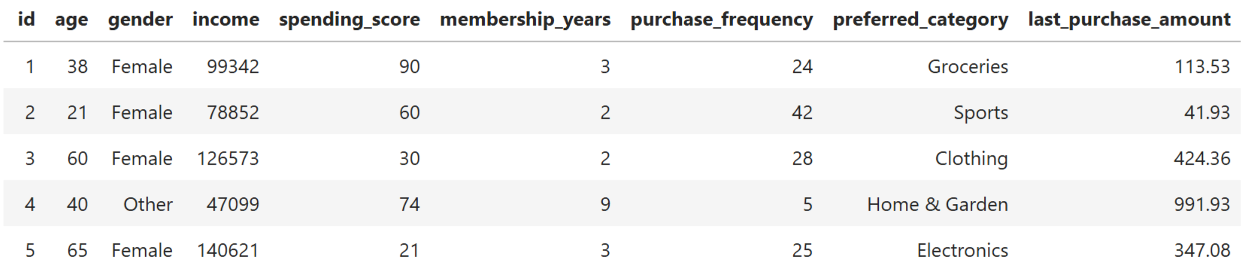
Furthermore, the categorical variable preferred_category offers opportunities to explore group-level patterns, while variables like purchase_frequency and membership_years are particularly relevant for defining customer loyalty. These insights are essential for identifying market segments and accurately predicting customer loyalty, thereby enabling companies to enhance their targeting strategies and improve customer retention.

Data Cleaning

The dataset was imported into Python as a DataFrame using the Pandas library. An overview of its structure is presented in Figure 1, which displays the column configuration.

Figure 1.

Overview of the dataset structure.



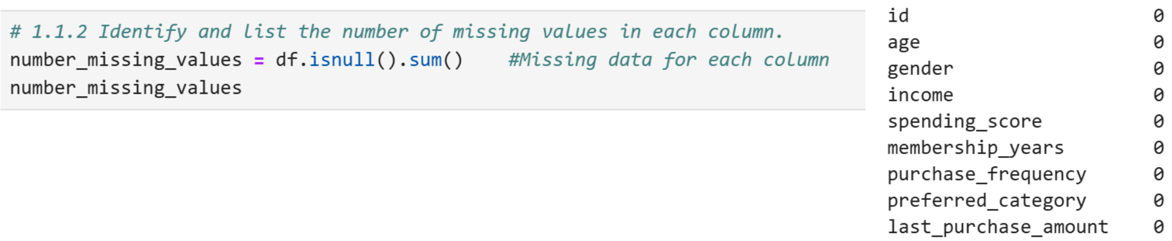
id	age	gender	income	spending_score	membership_years	purchase_frequency	preferred_category	last_purchase_amount
1	38	Female	99342	90	3	24	Groceries	113.53
2	21	Female	78852	60	2	42	Sports	41.93
3	60	Female	126573	30	2	28	Clothing	424.36
4	40	Other	47099	74	9	5	Home & Garden	991.93
5	65	Female	140621	21	3	25	Electronics	347.08

Missing values

An inspection was conducted to assess the presence of missing values across all columns. As shown in Figure 2, the analysis confirmed that there were no missing values in the dataset.

Figure 2.

Python code used to detect missing values and the resulting output.



```python
# 1.1.2 Identify and list the number of missing values in each column.
number_missing_values = df.isnull().sum()    #Missing data for each column
number_missing_values
```

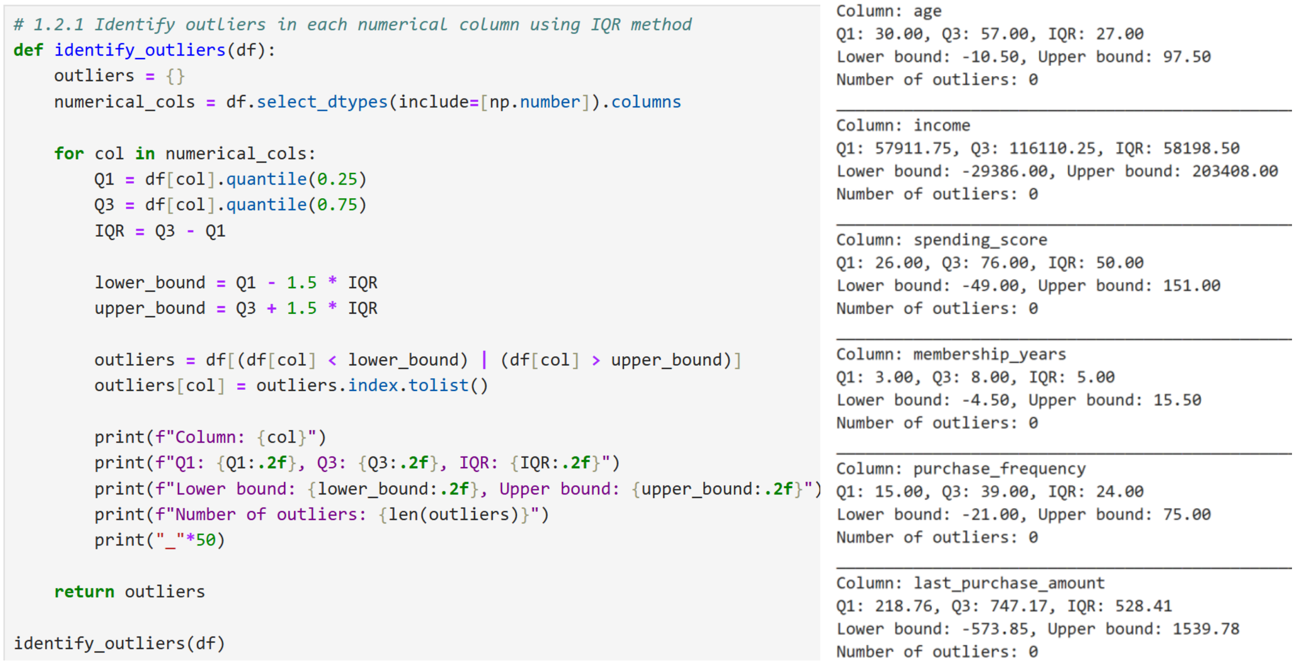
```
id                      0
age                     0
gender                  0
income                  0
spending_score          0
membership_years        0
purchase_frequency      0
preferred_category      0
last_purchase_amount    0
```

Outlier Detection

The Interquartile Range (IQR) method was applied to identify potential outliers in the numerical columns. This technique helps flag values that fall significantly outside the typical range, which could distort the analysis. However, the results indicated that no outliers were present in any of the numerical features.

Figure 3.

Code used for outlier detection and summary of results.



```
# 1.2.1 Identify outliers in each numerical column using IQR method
def identify_outliers(df):
    outliers = {}
    numerical_cols = df.select_dtypes(include=[np.number]).columns

    for col in numerical_cols:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1

        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]
        outliers[col] = outliers.index.tolist()

        print(f"Column: {col}")
        print(f"Q1: {Q1:.2f}, Q3: {Q3:.2f}, IQR: {IQR:.2f}")
        print(f"Lower bound: {lower_bound:.2f}, Upper bound: {upper_bound:.2f}")
        print(f"Number of outliers: {len(outliers)}")
        print("_"*50)

    return outliers

identify_outliers(df)
```

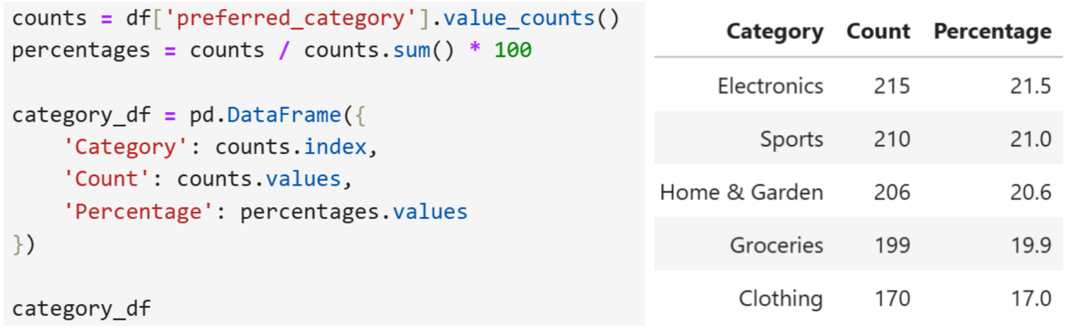
```
Column: age
Q1: 30.00, Q3: 57.00, IQR: 27.00
Lower bound: -10.50, Upper bound: 97.50
Number of outliers: 0
__________________________________________________
Column: income
Q1: 57911.75, Q3: 116110.25, IQR: 58198.50
Lower bound: -29386.00, Upper bound: 203408.00
Number of outliers: 0
__________________________________________________
Column: spending_score
Q1: 26.00, Q3: 76.00, IQR: 50.00
Lower bound: -49.00, Upper bound: 151.00
Number of outliers: 0
__________________________________________________
Column: membership_years
Q1: 3.00, Q3: 8.00, IQR: 5.00
Lower bound: -4.50, Upper bound: 15.50
Number of outliers: 0
__________________________________________________
Column: purchase_frequency
Q1: 15.00, Q3: 39.00, IQR: 24.00
Lower bound: -21.00, Upper bound: 75.00
Number of outliers: 0
__________________________________________________
Column: last_purchase_amount
Q1: 218.76, Q3: 747.17, IQR: 528.41
Lower bound: -573.85, Upper bound: 1539.78
Number of outliers: 0
```

Categorical Grouping

The number of unique categories in the ‘preferred_category’ column was examined to evaluate the diversity and potential need for dimensionality reduction techniques. The results, indicated that the dataset contains a manageable number of categories. Moreover, the distribution among them was relatively balanced, with differences of less than 5% between the most and least common categories. Therefore, no grouping or reduction was deemed necessary.

Figure 4.

Code and output for assessing the feasibility of category grouping.



```
counts = df['preferred_category'].value_counts()
percentages = counts / counts.sum() * 100

category_df = pd.DataFrame({
    'Category': counts.index,
    'Count': counts.values,
    'Percentage': percentages.values
})

category_df
```

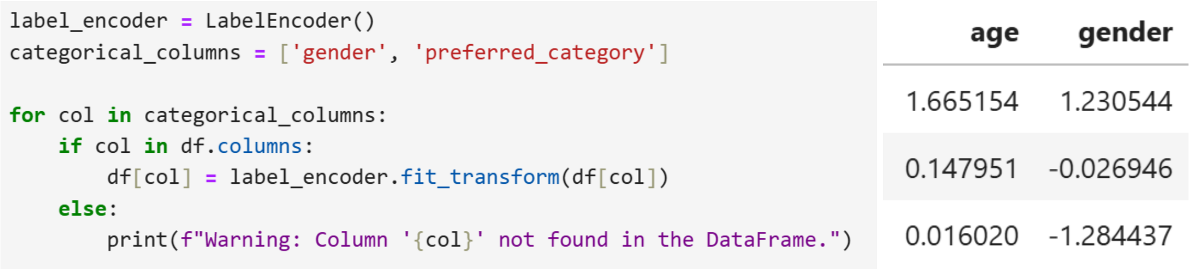
Category	Count	Percentage
Electronics	215	21.5
Sports	210	21.0
Home & Garden	206	20.6
Groceries	199	19.9
Clothing	170	17.0

Label Encoding for Categorical Variables

The code taught in class was used to perform label encoding for the categorical variables. These variables were selected after confirming that their data types matched the appropriate format (excluding the target variable y).

Figure 5.

Code for label encoding of categorical variables and output preview.



```
label_encoder = LabelEncoder()
categorical_columns = ['gender', 'preferred_category']

for col in categorical_columns:
    if col in df.columns:
        df[col] = label_encoder.fit_transform(df[col])
    else:
        print(f"Warning: Column '{col}' not found in the DataFrame.")
```

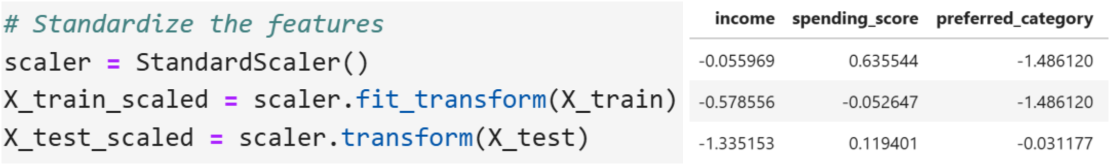
age	gender
1.665154	1.230544
0.147951	-0.026946
0.016020	-1.284437

Feature Scaling

For the PCA analysis, feature scaling was applied using the method introduced in class. The following figure shows the code implementation and the results after executing the scaling process.

Figure 6.

Code and summary of feature scaling.



```
# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

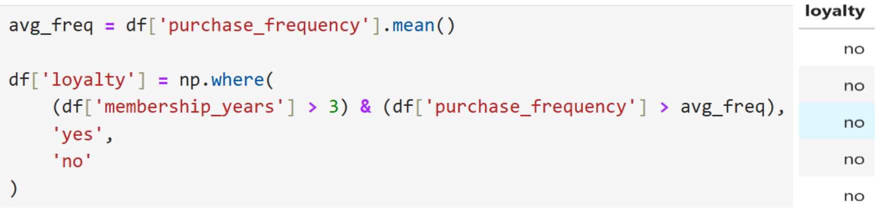
income	spending_score	preferred_category
-0.055969	0.635544	-1.486120
-0.578556	-0.052647	-1.486120
-1.335153	0.119401	-0.031177

Dummy Variable (y)

The creation of the target variable y was based on two input variables: purchase frequency and membership years. A rule was established based on EDA findings, which showed that after three years of membership, users exhibited different purchasing behavior, typically exceeding the average purchase frequency.

Figure 7.

Code for binary response variable creation and result.



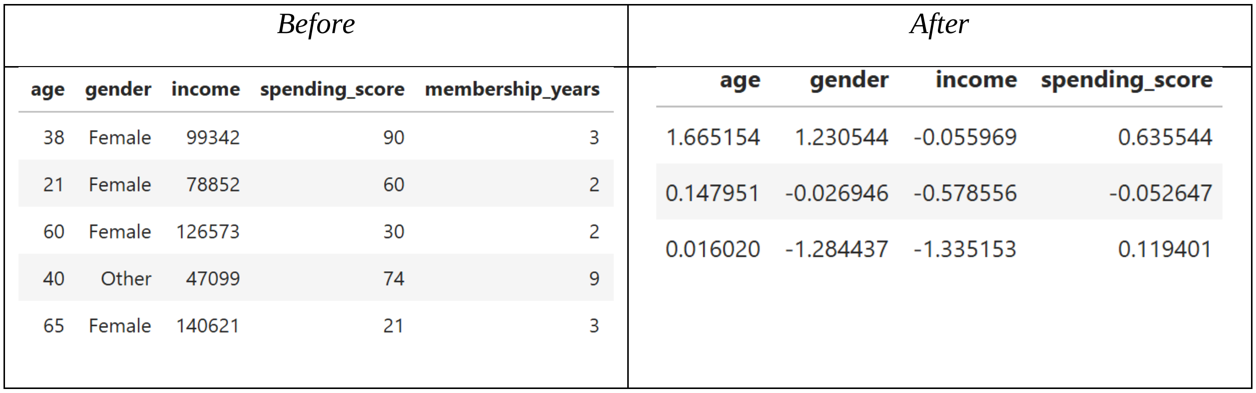
```
avg_freq = df['purchase_frequency'].mean()

df['loyalty'] = np.where(
    (df['membership_years'] > 3) & (df['purchase_frequency'] > avg_freq),
    'yes',
    'no'
)
```

loyalty
no
no
no
no
no

After completing the cleaning and transformation processes necessary for machine learning, a visual comparison is presented to show the dataset before and after preprocessing.

Figure 8.

Dataset before and after cleaning and preparation for machine learning.



Before

age	gender	income	spending_score	membership_years
38	Female	99342	90	3
21	Female	78852	60	2
60	Female	126573	30	2
40	Other	47099	74	9
65	Female	140621	21	3

After

age	gender	income	spending_score
1.665154	1.230544	-0.055969	0.635544
0.147951	-0.026946	-0.578556	-0.052647
0.016020	-1.284437	-1.335153	0.119401

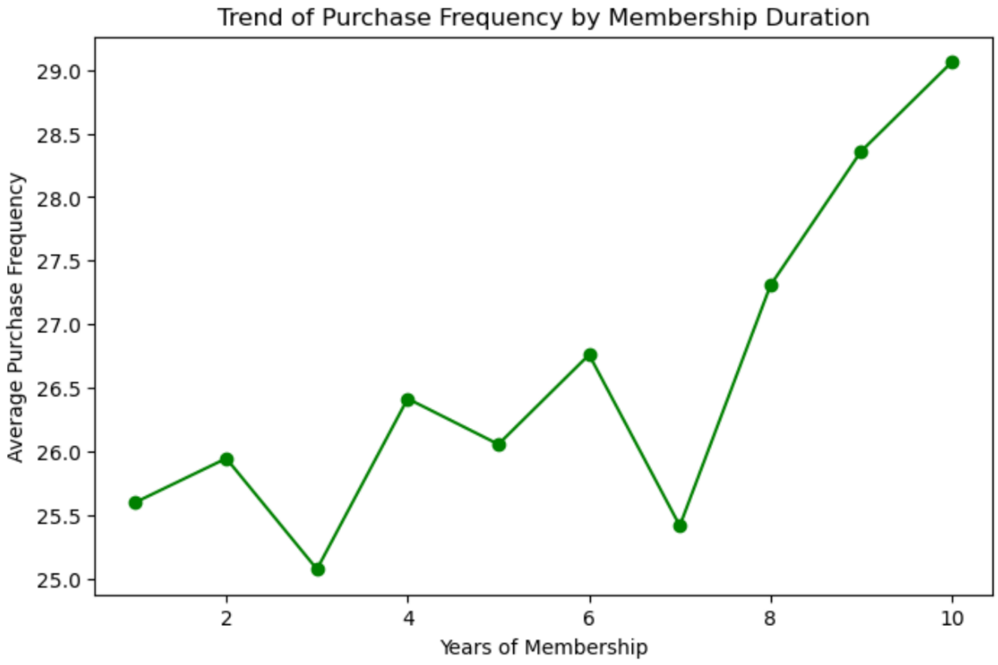
Visualizations, Analytical Methods & Implementation

Exploratory Data Analysis

1. What trends in purchase frequency are associated with the number of membership years?

To address this question, a line chart was selected, as it is the most effective method for visualizing trends and changes over time—in this case, the evolution of average purchase frequency across different membership durations. The Y-axis was adjusted to start at 25 to enhance the visual comparison between membership years.

Figure 9.
Trend in Average Purchase Frequency by Years of Membership



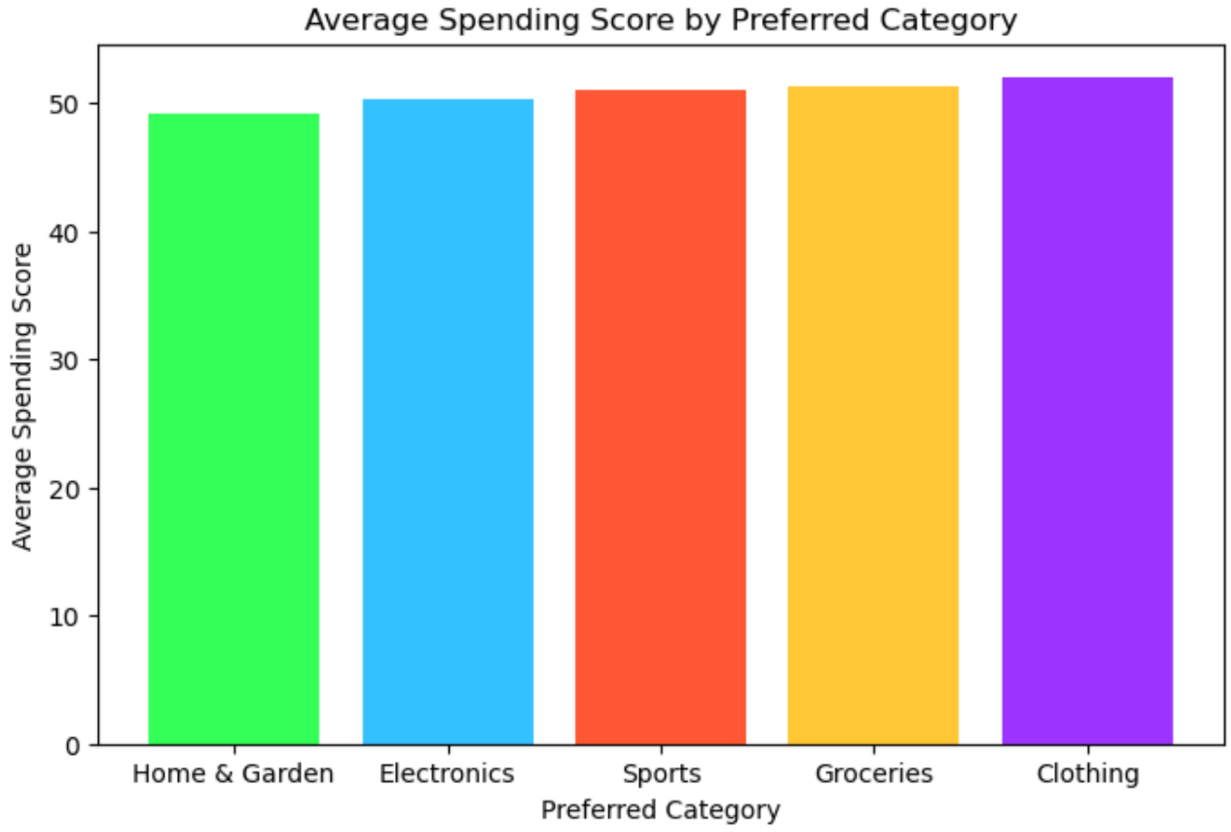
The results reveal a notable increase in purchase frequency among customers with over eight years of membership. Prior to this, an initial upward trend is observed from year 1 to year 2. However, there is a marked decline in years 3, 5, and 7. This pattern may indicate a cyclical engagement behavior or retention challenge, suggesting the need for deeper investigation into customer lifecycle dynamics and potential points of disengagement.

Statistical Data Analysis

2. Is there a significant relationship between a customer's preferred category and their spending score?

Two visualization techniques were employed to explore this question: a bar plot to compare the average spending score across customer-preferred categories, and a distribution matrix to identify any concentration patterns between spending levels and category preferences.

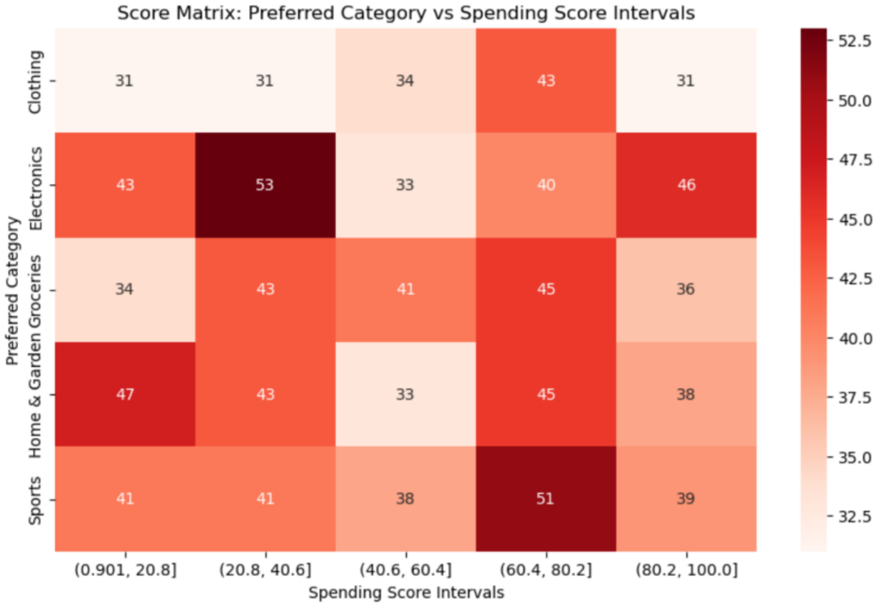
Figure 10.
Average Spending Score by Preferred Product Category



The bar plot shows that the Home & Garden category has the lowest average spending score, while Clothing has the highest. However, the differences between categories are relatively small, with most average scores ranging between 49 and 52, indicating a limited variation in customer spending by category.

Figure 11.

Spending Score Distribution Across Preferred Categories



The distribution matrix reveals stronger relationships between specific spending score intervals and certain preferred categories. For instance, Electronics is more frequently associated with scores in the 20–40 range, while Sports shows stronger associations with scores between 60–80. Although these two categories exhibit the most distinctive patterns, high spending scores above 80 are also primarily linked to Electronics. Additionally, the 60–80 score range shows consistent relevance across multiple categories, suggesting it may represent a key segment of high-engagement customers.

Machine Learning Analysis

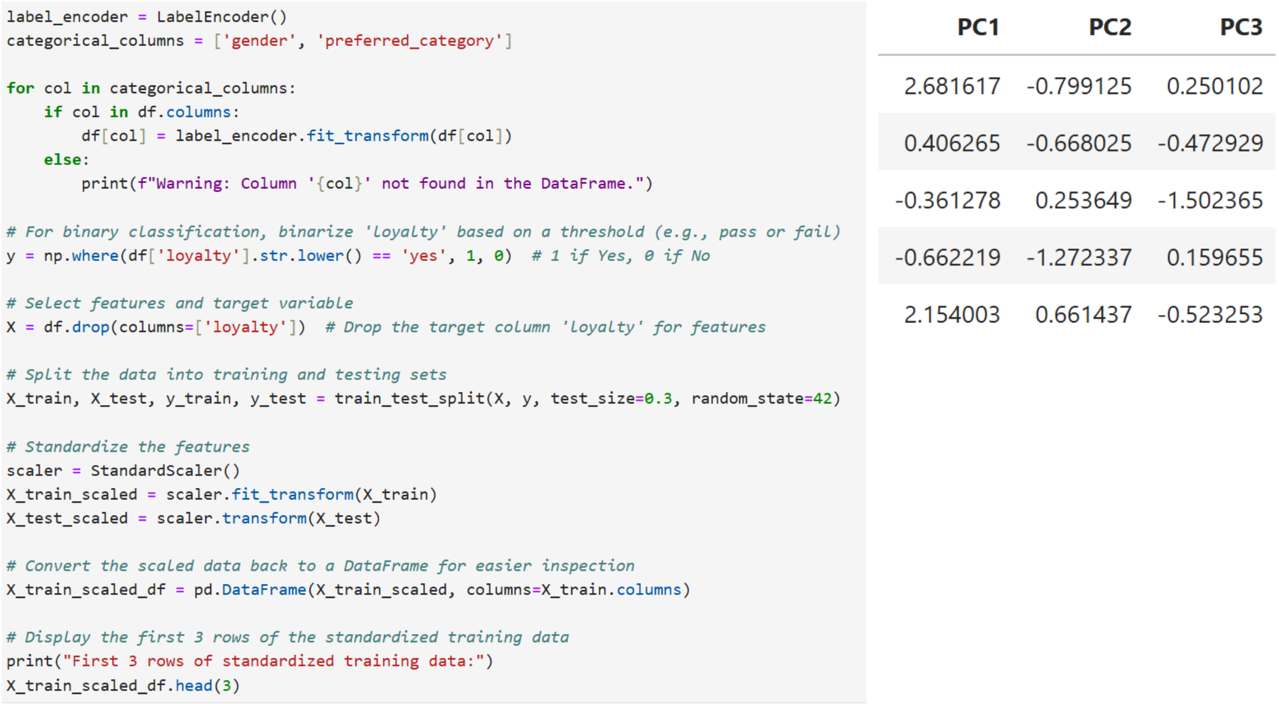
3. Can PCA effectively reduce the feature space while retaining at least 90% of the variance?

Principal Component Analysis (PCA) was employed for dimensionality reduction, condensing the original numerical features into a smaller set of orthogonal components that capture the majority of the variance in the data. This technique was selected both to apply concepts learned in class and to explore whether it would be possible to reduce the number of variables while preserving the quality and interpretability of the information.

PCA was initially conducted using two and three principal components. With only two components, the explained variance ratio did not exceed 15%, indicating a substantial loss of information. However, when three components were used, the cumulative explained variance increased significantly, offering a more informative representation of the original dataset.

Figure 12.

PCA implementation code and a preview of the resulting DataFrame with the principal components.



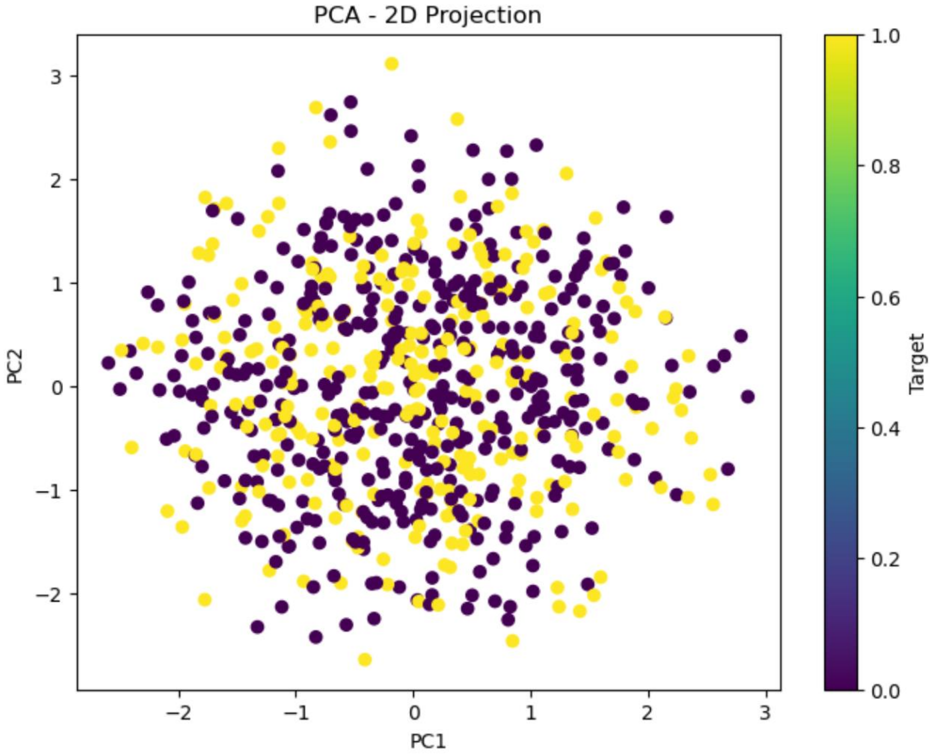
```
label_encoder = LabelEncoder()
categorical_columns = ['gender', 'preferred_category']

for col in categorical_columns:
    if col in df.columns:
        df[col] = label_encoder.fit_transform(df[col])
    else:
        print(f"Warning: Column '{col}' not found in the DataFrame.")

# For binary classification, binarize 'loyalty' based on a threshold (e.g., pass or fail)
y = np.where(df['loyalty'].str.lower() == 'yes', 1, 0)  # 1 if Yes, 0 if No

# Select features and target variable
X = df.drop(columns=['loyalty'])  # Drop the target column 'loyalty' for features

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Convert the scaled data back to a DataFrame for easier inspection
X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=X_train.columns)

# Display the first 3 rows of the standardized training data
print("First 3 rows of standardized training data:")
X_train_scaled_df.head(3)
```

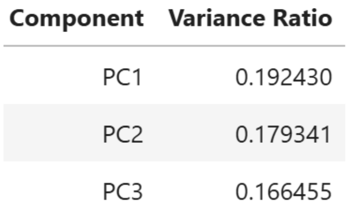
PC1	PC2	PC3
2.681617	-0.799125	0.250102
0.406265	-0.668025	-0.472929
-0.361278	0.253649	-1.502365
-0.662219	-1.272337	0.159655
2.154003	0.661437	-0.523253

The summary of the PCA transformation shows how the data is redistributed across the principal components. However, this alone does not allow us to determine whether PCA was beneficial for this specific dataset.

Figure 13.

2D PCA plot and variance ratio summary.





Component	Variance Ratio
PC1	0.192430
PC2	0.179341
PC3	0.166455

In Figure 13, only the first two principal components (PC1 and PC2) are visualized due to the two-dimensional nature of the plot. From this figure, we can observe that the separation between classes in the dummy variable is not clearly defined, as there is substantial overlap. This suggests limited class separability in the PCA-reduced space.

The variance ratio summary indicates that PC1 explains 19.24% of the variance, and the cumulative variance explained by the first three components is approximately 60%. This falls short of the 90% threshold initially set as the minimum acceptable level of retained variance.

As a result, PCA will not be used for this dataset, as the transformation does not meet the minimum requirements for preserving sufficient variance..

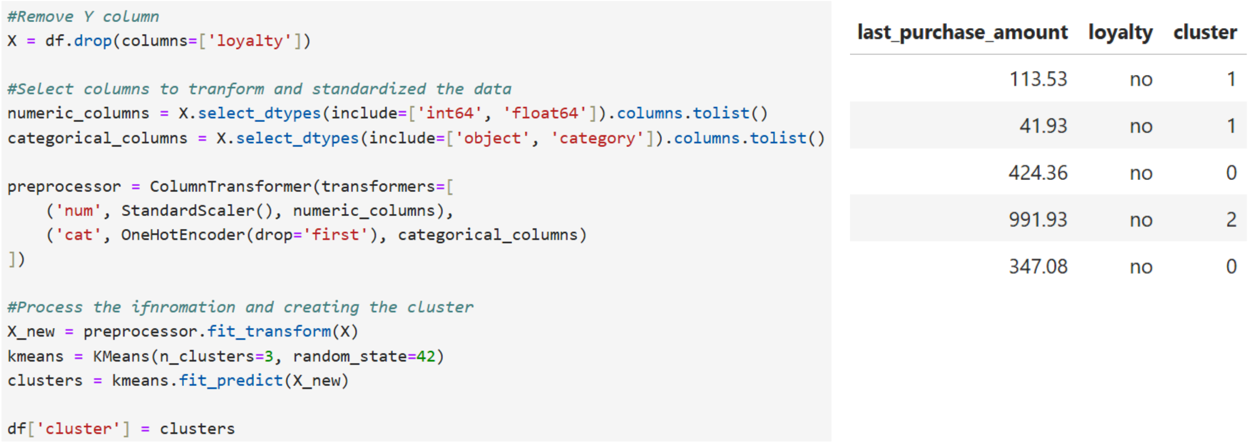
4. Do clusters derived from customer profiles provide meaningful segments that enhance the predictive power of the classification model?

To detect potential segments, we applied K-Means clustering—an unsupervised algorithm that segments data into distinct groups (clusters) by minimizing the distance between points and their assigned centroids. This approach was chosen for its simplicity and scalability in grouping similar observations.

Before performing clustering, the dataset underwent preprocessing: categorical variables were encoded using OneHotEncoder to convert them into a numerical format suitable for K-Means, and numerical features were standardized using StandardScaler to ensure equal weighting. The clustering process was evaluated using both the Silhouette Score, which measures how similar an object is to its own cluster compared to other clusters, and Inertia, which measures within-cluster sum-of-squares.

Figure 14.

Code and summary of the clustering procedure.

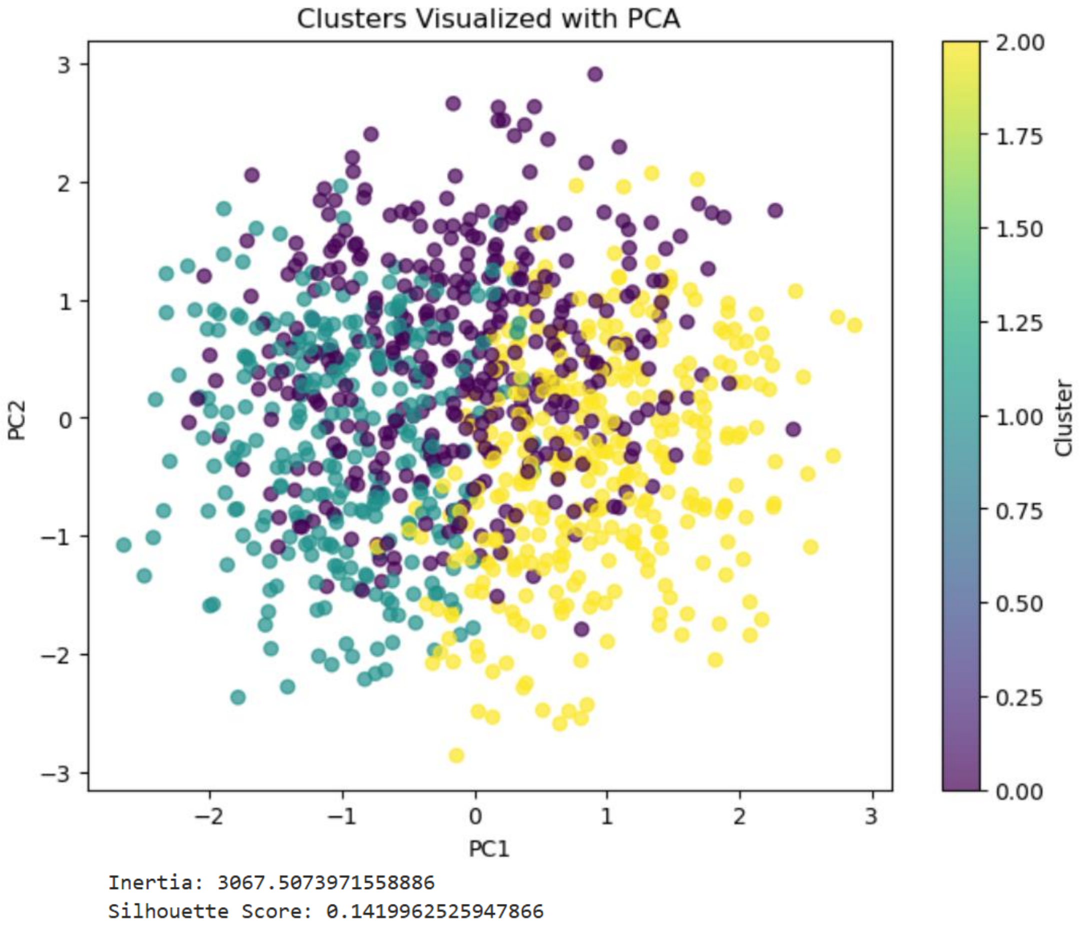


```
#Remove Y column
X = df.drop(columns=['loyalty'])

#Select columns to tranform and standardized the data
numeric_columns = X.select_dtypes(include=['int64', 'float64']).columns.tolist()
categorical_columns = X.select_dtypes(include=['object', 'category']).columns.tolist()

preprocessor = ColumnTransformer(transformers=[
    ('num', StandardScaler(), numeric_columns),
    ('cat', OneHotEncoder(drop='first'), categorical_columns)
])

#Process the ifnromation and creating the cluster
X_new = preprocessor.fit_transform(X)
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_new)

df['cluster'] = clusters
```

last_purchase_amount	loyalty	cluster
113.53	no	1
41.93	no	1
424.36	no	0
991.93	no	2
347.08	no	0

Figure 15.

Cluster plot using PCA, with evaluation metrics



```
Inertia: 3067.5073971558886
Silhouette Score: 0.1419962525947866
```

In the PCA-based plot, two of the three clusters are clearly distinguishable. However, the third cluster overlaps significantly with the other two, indicating weak separation. The inertia value was 3,067 and the Silhouette Score was 0.14, suggesting limited clustering quality and compactness. Despite this, the clustering result is retained, as it provides more structure than the previous PCA-only dimensionality reduction approach.

Table 2.

Cluster summary

Cluster	Age	Income	Spending_score
0	47.54717	79817.45	36.814465
1	40.35962	96223.63	29.504732
2	43.47671	89358.79	81.164384

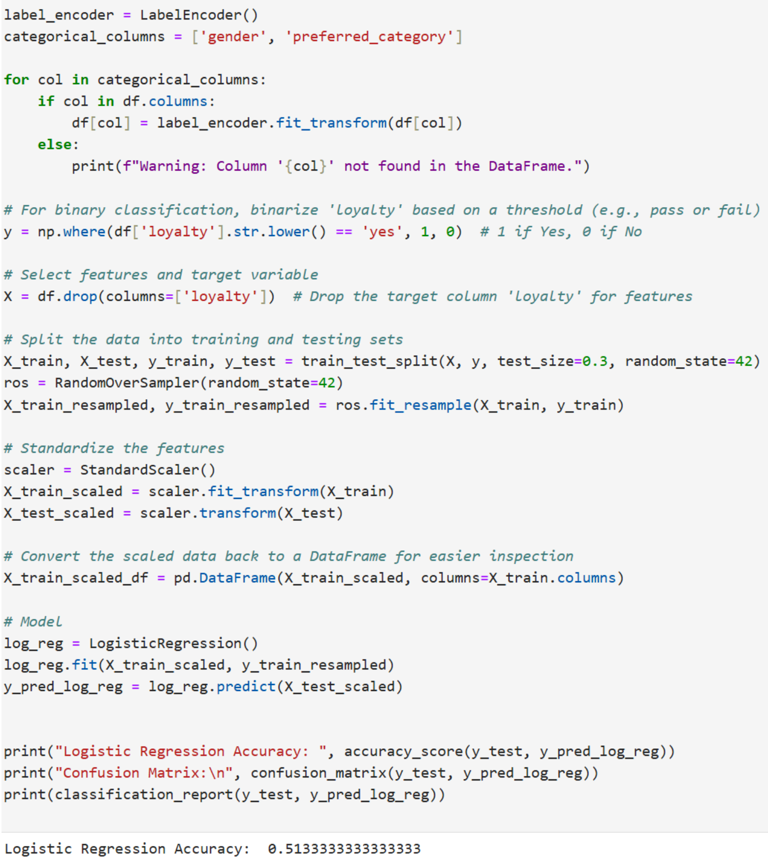
5. Which classification technique (logistic regression or decision tree) achieves higher accuracy in predicting customer loyalty?

The target variable in this analysis is whether a consumer is loyal or not loyal. Therefore, it is essential to apply classification models, such as Logistic Regression and Decision Tree Classifier. Logistic Regression is a statistical model that estimates the probability of a binary outcome based on one or more independent variables. Decision Tree Classifier, on the other hand, is a non-parametric model that operates by splitting the data into branches based on decision rules. While both models produce the same output (i.e., classification), their internal mechanisms differ significantly. The purpose of the following analysis is to compare both models to determine which performs better at classifying whether a customer is loyal.

For logistic regression, categorical variables were encoded and numerical variables were standardized. Below is a code snippet illustrating the implementation and the resulting accuracy:

Figure 16.

Code for Logistic Regression Model and Accuracy Output



```
label_encoder = LabelEncoder()
categorical_columns = ['gender', 'preferred_category']

for col in categorical_columns:
    if col in df.columns:
        df[col] = label_encoder.fit_transform(df[col])
    else:
        print(f"Warning: Column '{col}' not found in the DataFrame.")

# For binary classification, binarize 'loyalty' based on a threshold (e.g., pass or fail)
y = np.where(df['loyalty'].str.lower() == 'yes', 1, 0)  # 1 if Yes, 0 if No

# Select features and target variable
X = df.drop(columns=['loyalty'])  # Drop the target column 'loyalty' for features

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
ros = RandomOverSampler(random_state=42)
X_train_resampled, y_train_resampled = ros.fit_resample(X_train, y_train)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Convert the scaled data back to a DataFrame for easier inspection
X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=X_train.columns)

# Model
log_reg = LogisticRegression()
log_reg.fit(X_train_scaled, y_train_resampled)
y_pred_log_reg = log_reg.predict(X_test_scaled)


print("Logistic Regression Accuracy: ", accuracy_score(y_test, y_pred_log_reg))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred_log_reg))
print(classification_report(y_test, y_pred_log_reg))
```

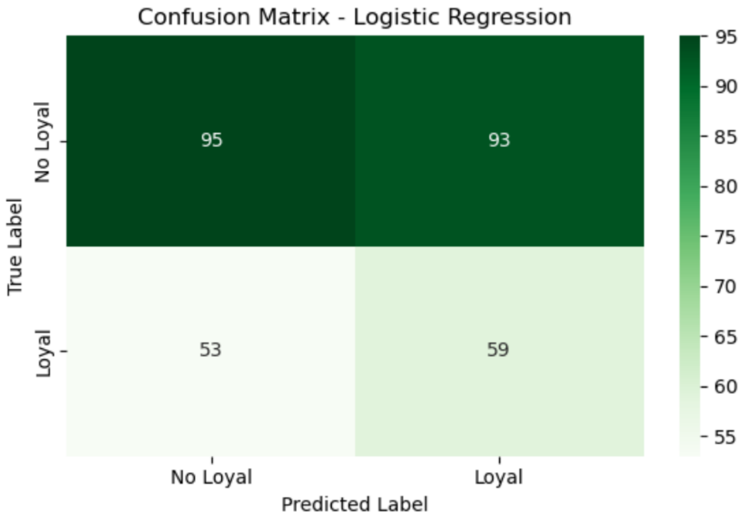
```
Logistic Regression Accuracy:  0.5133333333333333
```

The logistic regression model yielded an accuracy of 51.33%, which indicates that the model does not predict customer loyalty with sufficient precision. This means it only correctly classifies customer loyalty status approximately half the time, barely better than random guessing.

This is further confirmed by the confusion matrix:

Figure 17.

Confusion Matrix of Logistic Regression Model



In this matrix, we observe:

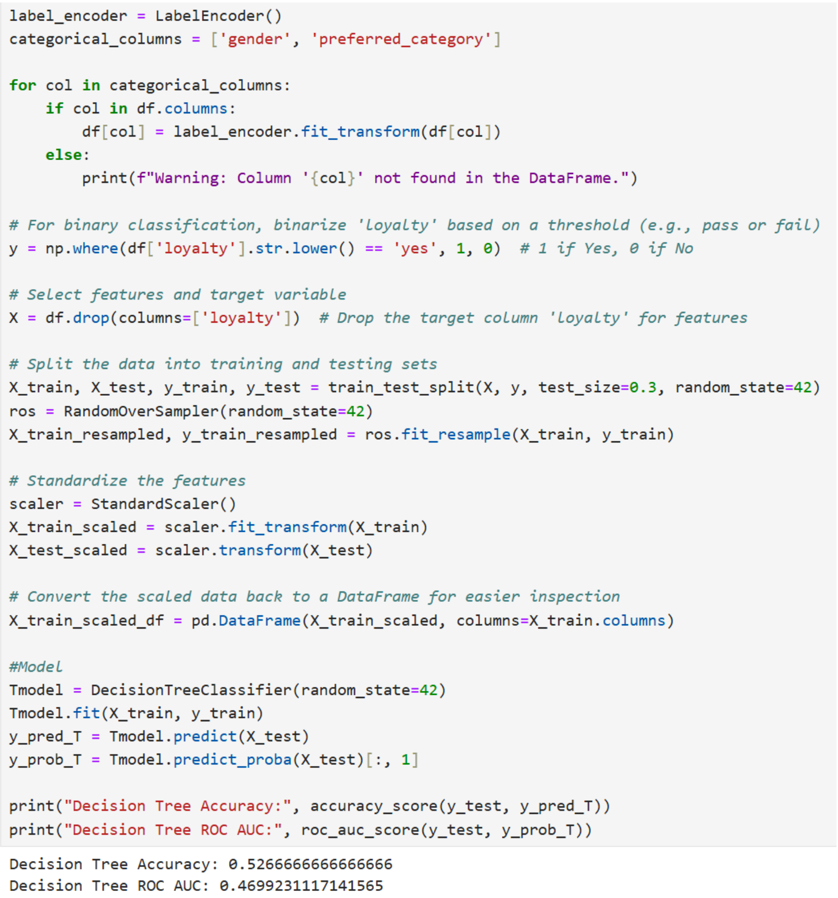
- The model correctly identified 95 non-loyal customers.
- It misclassified 53 loyal customers as non-loyal.
- It correctly predicted 59 loyal customers.
- It incorrectly classified 93 loyal customers as non-loyal.

These results illustrate poor model performance, with predictions close to random chance. The model struggles particularly with accurately identifying loyal customers.

A similar preprocessing pipeline was applied for the Decision Tree model. Below is the implementation code and its corresponding performance metrics:

Figure 18.

Code and Output of Decision Tree Classifier



```
label_encoder = LabelEncoder()
categorical_columns = ['gender', 'preferred_category']

for col in categorical_columns:
    if col in df.columns:
        df[col] = label_encoder.fit_transform(df[col])
    else:
        print(f"Warning: Column '{col}' not found in the DataFrame.")

# For binary classification, binarize 'loyalty' based on a threshold (e.g., pass or fail)
y = np.where(df['loyalty'].str.lower() == 'yes', 1, 0)  # 1 if Yes, 0 if No

# Select features and target variable
X = df.drop(columns=['loyalty'])  # Drop the target column 'loyalty' for features

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
ros = RandomOverSampler(random_state=42)
X_train_resampled, y_train_resampled = ros.fit_resample(X_train, y_train)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Convert the scaled data back to a DataFrame for easier inspection
X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=X_train.columns)

#Model
Tmodel = DecisionTreeClassifier(random_state=42)
Tmodel.fit(X_train, y_train)
y_pred_T = Tmodel.predict(X_test)
y_prob_T = Tmodel.predict_proba(X_test)[:, 1]

print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_T))
print("Decision Tree ROC AUC:", roc_auc_score(y_test, y_prob_T))
```

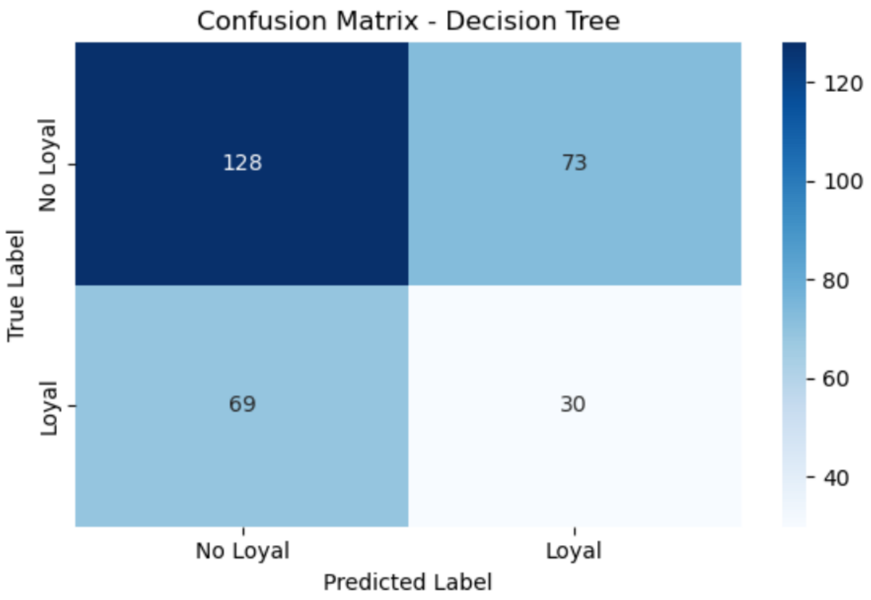
```
Decision Tree Accuracy: 0.5266666666666666
Decision Tree ROC AUC: 0.4699231117141565
```

The Decision Tree classifier achieved an accuracy of 52.66%, only marginally better than logistic regression. This small improvement suggests that the available variables are not sufficiently informative to predict customer loyalty effectively.

The confusion matrix below provides additional insights:

Figure 19.

Confusion Matrix of Decision Tree Classifier



Key observations:

- The model correctly predicted 128 non-loyal customers.
- It misclassified 69 loyal customers as non-loyal.
- It correctly identified only 30 loyal customers.
- It mistakenly classified 73 non-loyal customers as loyal.

This confirms that the Decision Tree model, while slightly better, also struggles to accurately predict loyalty due to limited predictive power of the features.

Both models performed poorly in classifying customer loyalty. Their low accuracy and imbalanced confusion matrices suggest that the current set of features lacks the necessary information to predict loyalty reliably. Future steps should include exploring additional variables (e.g., behavioral data, purchase history, engagement metrics)

Insights And Discussion

The study employed dimensionality reduction methods, including Principal Component Analysis (PCA) with both 2 and 3 principal components, as well as t-SNE (t-Distributed Stochastic Neighbor Embedding), to simplify the dataset while retaining meaningful patterns. However, neither PCA nor t-SNE achieved the desired cumulative explained variance of 90%, which was the predefined threshold for considering the reduction effective. PCA, while useful for visualizing high-dimensional data, only captured a limited portion of the variance, suggesting that the underlying structure of the data may not be strongly linear or that critical information was spread across many dimensions. Consequently, these techniques were excluded from the final modeling pipeline, and the original features were retained to avoid losing predictive power.

The clustering analysis identified three distinct customer segments, each with unique purchasing behaviors and financial profiles. The first segment comprised individuals with high purchasing power but surprisingly low spending scores, indicating a disconnect between their financial capacity and actual expenditure, a potential target for personalized engagement strategies. The second segment exhibited high spending scores paired with moderate income, suggesting financially savvy consumers who maximize value. The third and most lucrative segment combined high income with high spending, representing ideal candidates for premium offerings and loyalty programs. These findings enable marketing strategies, such as incentivizing the first segment to increase spending or deepening relationships with high-value customers in the third segment. The clear differentiation between clusters confirms the value of unsupervised learning in revealing hidden customer profiles that may not be apparent through traditional analysis.

Among the classification models tested, including logistic regression and decision trees, the decision tree algorithm delivered the best results, albeit with a modest accuracy of only 53%. While this performance was suboptimal, the decision tree's interpretability provided valuable insights into feature importance, revealing that behavioral and demographic variables were the primary drivers of predictions. The model's limitations likely stem from the absence of key predictive features, particularly emotional or sentiment-based data, which could better explain customer decision-making. Unlike more complex ensemble methods, the decision tree’s simplicity may have helped it avoid overfitting, but the low accuracy underscores the need for richer input variables. Future iterations of this model should incorporate psychographic or transactional sentiment data to improve predictive capability.

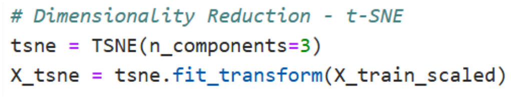
To improve classification performance, future research should consider incorporating emotional, attitudinal, and psychographic variables, such as brand sentiment, trust, perceived ethical alignment, and customer satisfaction metrics. These could significantly enhance the model’s ability to capture the complexity behind customer loyalty and produce more accurate predictions.

Reference

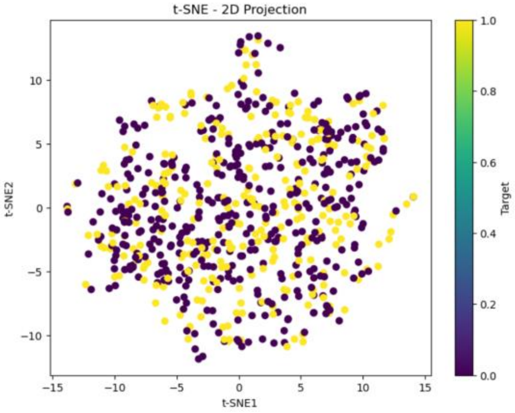
Chowdhury, F. (n.d.). *Customer segmentation data for marketing analysis* [Data set]. Kaggle. https://www.kaggle.com/datasets/fahmidachowdhury/customer-segmentation-data-for-marketing-analysis

Appendix

t-SNE



```
# Dimensionality Reduction - t-SNE
tsne = TSNE(n_components=3)
X_tsne = tsne.fit_transform(X_train_scaled)
```


t-SNE - 2D Projection
t-SNE1
t-SNE2
Target