Technical Report: IDMP Final Project DS 5110

Sujan DM, Vishesh Bharukha, Kishan Kumar Parida

December 1, 2024

1 Introduction

The project aims to analyze customer behavior using data analysis and machine learning to provide actionable insights for business decision-making.
This project, "Data Visualization Dashboard for Consumer Behavior Analysis," focuses on understanding the insights from consumer transaction data using advanced data science and machine learning techniques.The main goal is to provide a thorough, interactive dashboard that will help close the gap between strategic decision-making and raw data.
The project blends in the clustering techniques, predictive analytics, anomaly detection, and recommendation systems to ensure a technical approach to consumer analysis.

The scope of the project includes:

1. Customer Segmentation: Grouping consumers into distinct clusters based on demographics and purchasing patterns using the K-Means algorithm.
2. Predictive Modeling: Using Random Forest regression to predict purchase amounts, enabling businesses to forecast revenues and strategize pricing.
3. Anomaly Detection: Using Isolation Forest to identify unusual transactions, which could signify potential fraud or outlier behavior.
4. Recommendation System: Leveraging collaborative filtering and content-based filtering to suggest personalized product recommendations to improve customer retention and satisfaction.

In addition, these insights are consolidated into an interactive Power BI dashboard designed to enable stakeholders to obtain insightful real-time information. This dashboard offers a powerful tool for data-driven decision-making by enabling users to examine consumer demographics, purchasing patterns, and flagged anomalies, providing a robust tool for data-driven decision-making.

2 Literature Review

Consumer behavior analysis is a key component of modern business intelligence that provides insights that promote customer engagement, inventory management, and revenue growth. Over the years, numerous studies have explored methodologies for understanding consumer trends, predicting purchasing behavior, and improving customer targeting through data-driven techniques.

a.) Clustering in Consumer Behavior Analysis:
Clustering techniques, K-Means, and other clustering techniques have been frequently utilized to divide consumers into meaningful groups. Clustering by demographic characteristics (e.g., age, gender) and purchase behavior (e.g. frequency, spending amount) can yield useful information for focused marketing initiatives, according to research with examples of how segmenting customers into age and spending brackets shows to improve the efficiency of promotional strategies. However, traditional clustering methods often struggle with high-dimensional data, which requires pre-processing techniques such as feature scaling and dimensionality reduction, which are incorporated in this project [1].

b.) Anomaly Detection in Transaction Data:
Anomaly detection methods like Isolation Forest have proven effective in identifying fraudulent transactions and unusual customer behavior. Anomaly detection has become crucial in financial and e-commerce sectors, where frauds can be detected by tracking the irregular transactions.[2]

c.) Recommendation Systems in E-commerce:
Recommendation systems are a staple of modern e-commerce platforms, with collaborative filtering and content-based filtering being the most commonly used methods. These models work well when there is enough interaction data, but sparsity is still a problem that necessitates careful management of missing variables and matrix optimization.[3]

d.) Predictive Modeling for Consumer Trends:
Regression models in particular have been employed in predictive analytics to estimate purchase behavior based on past transactions and customer demographics. Random Forest regression is often preferred due to its robustness and ability to handle non-linear relationships. This project aims to close the gaps in the integration of such models into a single process for real-time analytics.[4]

e.) Visualization of Consumer Insights:
The way that the insights are presented is just as important to the efficacy of consumer analytics as the insights themselves. Power BI and Tableau-powered dashboards are becoming more and more acknowledged as essential elements for converting intricate analysis into useful insights.

f.) Research Gaps:
Clustering, anomaly detection, recommendation systems, and predictive modeling have all seen a great deal of research, but nothing has been done to combine these techniques into a coherent whole. Furthermore, there is still a lack of research on applying these techniques to medium-sized datasets (like the one utilized in this study) with an emphasis on interaction and real-time insights.

3 Methodology

3.1 Data Collection

The data set contains just under 4000 transactions with 19 featured columns, with:
1. Demographics: Age, gender, customer ID.
2. Purchase Details: Items purchased, amounts, categories.
3. Behavioral data: Payment methods, review ratings, discounts.

The data was sourced from a consumer behavior dataset on Kaggle.

3.2 Data Preprocessing

1. Missing Values: There are minimal values which can be classified as missing values in the dataset as all the missing values were addressed or removed in the pre-processing stage.
2. Duplicates: All duplicate values are removed for data integrity.
3. Standardization: The 19 columns are standardized for uniformity.

3.3 Analysis Techniques

The analysis techniques used in the projectare as follows:

1. Clustering: K-Means with six clusters determined using the Elbow Method.
2. Anomaly Detection: Isolation Forest with a 5 percent contamination rate to flag unusual transactions.
3. Recommendation System: Collaborative filtering and content based filtering using LIGHTFM.
4. Regression: Random Forest for predicting purchase amounts.

3.4 Anomaly Detection Using Isolation Forest

To identify unusual transactions, which could indicate fraud or atypical customer behavior:

a.) Features Used: Purchase amount, review rating, and transaction frequency.
b.) Parameters: A rate of 5 percent was set to reflect the expected proportion of anomalies.
c.) Results: High-value and irregularly frequent purchases were flagged as anomalies, aiding in fraud detection. Recommendation System with Collaborative Filtering A personalized recommendation system was developed to enhance customer retention:

3.5 Predictive Modeling with Random Forest Regression

Random Forest regression is used to predict purchase amounts based on customer attributes which includes: Age, review ratings, and historical purchases. Parameters like the number of estimators and max depth were tuned for optimal performance for the project with a comprehensive Power BI dashboard was created to present the insights.

4 Result

The study produced a number of important insights and useful conclusions that demonstrated the efficacy of the used machine learning models as well as trends in consumer behavior. The results are broken down in detail below:

4.1 Demographic Analysis

Age Distribution: The majority of customers (about 65 percent) were in the 25–49 age group. The histogram of age data exhibited a normal-like distribution, suggesting that middle-aged individuals are the primary consumers in this dataset. Gender Representation: While gender-specific insights were not highlighted, this feature can be further explored to uncover potential trends.

4.2 Spending Patterns

Average Purchase Amount: The mean purchase amount was $59.76, with variations across categories. This information is crucial for pricing strategies and understanding spending capabilities across customer groups.

4.3 Top Product Categories

Clothing and accessories are the most purchased categories among all the ones, with items like blouses, pants, and shirts having the most sales. These categories accounted for over 40 percent of total transactions.
Discount Influence: Approximately 43 percent of purchases involved had discounts. Transactions which had discounts had 15 percent higher purchase frequency than those without discounts.

4.4 Customer Segmentation

Six distinct customer segments were identified by applying K-Means clustering:

Cluster 1: Young, low spenders with infrequent purchases were identified in the first cluster.
Cluster 2: Middle-aged, high spenders who are frequent buyers were identified in the second cluster.
Cluster 3: Budget-conscious shoppers who often use discounts were identified in the third cluster.
Cluster 4: Sporadic buyers with high variability in spending were identified in the fourth cluster.
Cluster 5: Loyal customers with consistent purchasing habits were identified in the fifth cluster.
Cluster 6: High-value customers with significant transaction amounts were identified in the sixth cluster.

These segments give an insight for targeting high-value customers with exclusive offers or incentivizing low-spending clusters to increase engagement.

4.5 Anomaly Detection

a.) Identified Anomalies: Approximately 5 percent of transactions were identified as anomalies using Isolation Forest. These anomalies included: customers with High-value purchases that deviated significantly from typical spending patterns.

b.) Business Impact: These transactions can be classified further for potential fraud, enhancing transaction monitoring.

4.6 Visualization Insights

The Power BI dashboard effectively presented the findings, providing an interactive and user-friendly interface for exploring the data with Total sales, average order values, and discount usage rates used as the key insights data.

Trends: Monthly sales trends by product category were visualized with revealing the seasonal spikes in specific categories.

5 Discussion:

The results emphasized on how crucial it is to use clustering insights to inform targeted marketing. The combination of anomaly detection with real-time recommendations provides a competitive advantage, even if the majority of the results are consistent with other studies. Among the disparities are somewhat lower spending averages than anticipated, which could be the result of dataset constraints.

The integration of insights into a Power BI dashboard for real-time decision-making can monitor the consumer trends and adjust strategies promptly.
Implementation of fraud detection systems that flag anomalies in real-time and enhances the customer experience with dynamic, personalized recommendations.

5.1 Key Insights

Demographics and Spending Trends: With an average purchase amount of $59.76. The majority of clients are between the ages of 25 and 49. Discounts have a big impact on how often people buy, indicating that they are price sensitive.
Customer Segmentation: The six clusters that were found show a range of consumer types, from luxury consumers to frugal shoppers, allowing for customized marketing tactics.
Anomaly Detection: Isolation Forest helped detect fraud and monitor odd activity by flagging 5 percent of transactions as anomalies.
Recommendations and Predictions: Both the purchase prediction model and the personalized suggestion system worked successfully, improving client retention and supporting inventory management.

5.2 Comparison with Literature:

The results align with previous studies on clustering, anomaly detection, and recommendation systems but enhance them by combining various methods into a unified framework. While the limitations as pf a Wider application is limited by the dataset's size and static nature. Precision may be impacted by assumptions in models such as K-Means and Isolation Forest, the future Applications could incorporate real-time data for dynamic insights. Expanding data sources and using advanced models like deep learning.

6 Conclusion

This project, "Data Visualization Dashboard for Consumer Behavior Analysis," effectively illustrated how data science and machine learning methods may be applied to extract insightful information from consumer transaction data. Customer segmentation, fraud detection, and targeted marketing were among the crucial business goals that the project addressed by utilizing clustering, anomaly detection, recommendation systems, and predictive modeling.

6.1 Key Findings

a.) Demographic Insights: According to the data, the majority of the client base is middle-aged, with an average purchase value of $59.76. Businesses can effectively customize their marketing strategy by having a thorough understanding of this important population.
b.) Product and Spending Trends: The most popular categories were found to be apparel and accessories, and it was discovered that discounts had a 15 percent increase in the frequency of purchases.
c.) Targeting and Segmentation: Six unique client groups were found using the K-Means clustering, which offered a framework for resource allocation and customized marketing efforts.
d.) Fraud Detection: By identifying 5 percent of transactions as possible outliers, the anomaly detection methodology provides a reliable way to keep an eye on odd activity and avert financial hazards.

6.2 Limitations

a.) Dataset Restrictions: Although the dataset is enough for exploratory analysis, it might not fully represent the range of customer behavior would be improved with larger and more diverse datasets.
b.) Static Nature of Data: The analysis's capacity to adjust to real-time shifts in consumer behavior is constrained by its reliance on static data.
c.) Feature Availability: Additional characteristics, such as location data or social media activity, could provide deeper insights.

6.3 Future Research

Using real-time data streams to produce insights that are flexible and dynamic is known as real-time integration. Extending Scope: To improve the depth of analysis, use bigger datasets and add more features. Advanced Models: Investigating deep learning methods to enhance personalization and predictive power.

7 References

1. https://ieeexplore.ieee.org/document/10071344
2. https://ieeexplore.ieee.org/document/9762926
3. https://link.springer.com/book/10.1007/978-1-4899-7637-6
4. https://link.springer.com/article/10.1023/A:1010933404324

7.1 Appendices A

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#data reading
file_path = 'consumerbehaviour.csv'
df = pd.read_csv(file_path)


print("Data Overview:")
print(df.info())
print("\nFirst few rows:")
print(df.head())
```

```
# Checking for missing values and pre-processing
print("\nMissing Values per Column:")
print(df.isnull().sum())

df.drop_duplicates(inplace=True)
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')
```

```
#statistics
print("\nSummary Statistics:")
print(df.describe())
```

```
#unique counts
for column in df.select_dtypes(include='object').columns:
    print(f"\nUnique values in {column}:")
    print(df[column].value_counts())
```

```
#customer age distribution
import matplotlib.pyplot as plt


plt.figure(figsize=(10, 6))

age_counts, bins, patches = plt.hist(df['age'], bins=20, edgecolor='black')
plt.title("Customer Age Distribution")
plt.xlabel("Age")
plt.ylabel("Frequency")


for i in range(len(patches)):
    plt.text(patches[i].get_x() + patches[i].get_width() / 2,
             age_counts[i] + 0.5,
             int(age_counts[i]),
             ha='center', va='bottom')
plt.show()
```

```python
#top 10 items purchased
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(10, 6))
top_items = df['item_purchased'].value_counts().head(10)
sns.barplot(y=top_items.index, x=top_items.values, palette="viridis")
plt.title("Top 10 Items Purchased")
plt.xlabel("Purchase Count")


for index, value in enumerate(top_items.values):
    plt.text(value + 0.5, index, str(value), ha='center', va='center')

plt.show()
```

```python
#average purchase amount by category of products
plt.figure(figsize=(12, 6))
avg_purchase_by_category = df.groupby('category')['purchase_amount_(usd)'].mean().sort_values(ascending=False)
sns.barplot(x=avg_purchase_by_category.index, y=avg_purchase_by_category.values, palette="viridis")
plt.title("Average Purchase Amount by Category")
plt.xlabel("Category")
plt.ylabel("Average Purchase Amount (USD)")
plt.xticks(rotation=45)


for index, value in enumerate(avg_purchase_by_category.values):
    plt.text(index, value + 0.5, f"${value:.2f}", ha='center', va='bottom')

plt.show()
```

```python
#Review rating distribution
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(10, 6))
review_counts, bins, patches = plt.hist(df['review_rating'], bins=10, edgecolor='black')
plt.title("Review Rating Distribution")
plt.xlabel("Review Rating")
plt.ylabel("Frequency")

for i in range(len(patches)):
    plt.text(patches[i].get_x() + patches[i].get_width() / 2,
             review_counts[i] + 0.5,
             int(review_counts[i]),
             ha='center', va='bottom')

plt.show()
```

```python
#payment method usage
import matplotlib.pyplot as plt
import seaborn as sns


plt.figure(figsize=(10, 6))
payment_counts = df['payment_method'].value_counts()
sns.barplot(x=payment_counts.index, y=payment_counts.values, palette="viridis")
plt.title("Payment Method Usage")
plt.xlabel("Payment Method")
plt.ylabel("Count")
plt.xticks(rotation=45)


for index, value in enumerate(payment_counts.values):
    plt.text(index, value + 0.5, str(value), ha='center', va='bottom')

plt.show()
```

```python
#discount
import matplotlib.pyplot as plt


plt.figure(figsize=(6, 6))
discount_counts = df['discount_applied'].value_counts()


def func(pct, allvals):
    absolute = int(round(pct/100.*sum(allvals)))
    return "{:.1f}%\n({:d})".format(pct, absolute)

discount_counts.plot(
    kind='pie',
    autopct=lambda pct: func(pct, discount_counts),
    startangle=140,
    title="Discount Applied",
    wedgeprops=dict(edgecolor='black')
)
plt.ylabel("")
plt.show()
```

```python
#Frequency of purchases distribution
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
purchase_frequency_counts = df['frequency_of_purchases'].value_counts()
sns.barplot(x=purchase_frequency_counts.index, y=purchase_frequency_counts.values, palette="viridis")
plt.title("Frequency of Purchases Distribution")
plt.xlabel("Frequency of Purchases")
plt.ylabel("Count")
plt.xticks(rotation=45)


for index, value in enumerate(purchase_frequency_counts.values):
    plt.text(index, value + 0.5, str(value), ha='center', va='bottom')

plt.show()
```

```python
#avg spending by age group
import matplotlib.pyplot as plt


df['age_group'] = pd.cut(df['age'], bins=[18, 25, 35, 50, 65], labels=['18-24', '25-34', '35-49', '50-64'])
age_group_spending = df.groupby('age_group')['purchase_amount_(usd)'].mean()


plt.figure(figsize=(8, 6))
age_group_spending.plot(kind='bar', color='skyblue', edgecolor='black')
plt.title("Average Spending by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Average Spending (USD)")

for index, value in enumerate(age_group_spending):
    plt.text(index, value + 0.5, f"${value:.2f}", ha='center', va='bottom')

plt.show()
```

```python
#avg order value
customer_lifetime_value = df.groupby('customer_id')['purchase_amount_(usd)'].sum()
print(customer_lifetime_value.head())
total_revenue = df['purchase_amount_(usd)'].sum()
total_orders = df['purchase_amount_(usd)'].count()
aov = total_revenue / total_orders
print("Average Order Value (AOV): $", round(aov, 2))
```

```python
#correlqtion heatmap
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 8))
corr_matrix = df[['age', 'purchase_amount_(usd)', 'review_rating', 'previous_purchases']].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1, square=True, linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```

```python
#bubble chart for product popularity and avg rating
import plotly.express as px


product_data = df.groupby('item_purchased').agg({
    'purchase_amount_(usd)': 'sum',
    'review_rating': 'mean',
    'item_purchased': 'size'
}).rename(columns={'item_purchased': 'purchase_count'})

fig = px.scatter(product_data, x='purchase_count', y='review_rating', size='purchase_amount_(usd)',
                 hover_name=product_data.index, title="Product Popularity and Average Rating",
                 labels={'purchase_count': 'Number of Purchases', 'review_rating': 'Average Rating'})
fig.update_traces(marker=dict(opacity=0.6, line=dict(width=1, color='DarkSlateGrey')))
fig.show()
```

```python
import plotly.express as px


hierarchy_data = df.groupby(['age_group', 'category', 'item_purchased']).size().reset_index(name='count')

# sunburst chart for customer purchase hierarchy
fig = px.sunburst(hierarchy_data, path=['age_group', 'category', 'item_purchased'], values='count',
                  title="Customer Purchase Hierarchy")
fig.show()
```

```python
import matplotlib.pyplot as plt
#payment method usage  chart
payment_counts = df['payment_method'].value_counts()

plt.figure(figsize=(8, 8))
plt.pie(payment_counts, labels=payment_counts.index, autopct='%1.1f%%', startangle=140, pctdistance=0.85)
center_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(center_circle)

plt.title("Payment Method Usage")
plt.show()
```

```python
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np



#K-means Clustering
clustering_features = df[['age', 'purchase_amount_(usd)', 'previous_purchases']].dropna()
scaler = StandardScaler()
scaled_features = scaler.fit_transform(clustering_features)


inertia = []
silhouette_scores = []

for k in range(2, 10):
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(scaled_features)
    inertia.append(kmeans.inertia_)
    silhouette_scores.append(silhouette_score(scaled_features, kmeans.labels_))

plt.figure(figsize=(8, 6))
plt.plot(range(2, 10), inertia, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
```

```python
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal Clusters')
plt.show()


plt.figure(figsize=(8, 6))
plt.plot(range(2, 10), silhouette_scores, marker='o', linestyle='--', color='orange')
plt.xlabel('Number of Clusters')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Scores for Optimal Clusters')
plt.show()


optimal_clusters = 6
kmeans = KMeans(n_clusters=optimal_clusters, random_state=0)
df['cluster'] = kmeans.fit_predict(scaled_features)


plt.figure(figsize=(8, 6))
plt.scatter(clustering_features['age'], clustering_features['purchase_amount_(usd)'], c=df['cluster'], cmap='viridis', edgecolor='k')
plt.xlabel('Age')
plt.ylabel('Purchase Amount (USD)')
plt.title('Customer Segmentation by Clusters (K-Means)')
plt.colorbar(label='Cluster')
plt.show()
```

```python
#3D Visualization
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(
    clustering_features['age'],
    clustering_features['purchase_amount_(usd)'],
    clustering_features['previous_purchases'],
    c=df['cluster'], cmap='viridis', edgecolor='k'
)
ax.set_xlabel('Age')
ax.set_ylabel('Purchase Amount (USD)')
ax.set_zlabel('Previous Purchases')
ax.set_title('3D Customer Segmentation (K-Means)')
plt.show()


output_file = "kmeans_clustering_results.csv"
df.to_csv(output_file, index=False)
print(f"K-Means clustering results saved to: {output_file}")

# Step 6: Optional - Export Cluster Centroids
centroids = scaler.inverse_transform(kmeans.cluster_centers_)  # Reverse scaling for original values
centroids_df = pd.DataFrame(centroids, columns=['age', 'purchase_amount_(usd)', 'previous_purchases'])
centroids_file = "cluster_centroids.csv"
centroids_df.to_csv(centroids_file, index=False)
print(f"Cluster centroids saved to: {centroids_file}")
```

```python
# Recommendation system using LightFM
import pandas as pd
import numpy as np
from lightfm import LightFM
from lightfm.data import Dataset

df['season'] = df['season'].astype(str)
df['gender'] = df['gender'].astype(str)

dataset = Dataset()
dataset.fit(
    users=df['customer_id'].unique(),
    items=df['item_purchased'].unique(),
    user_features=['gender:' + g for g in df['gender'].unique()],
    item_features=['season:' + s for s in df['season'].unique()]
)


user_features = [(row['customer_id'], ['gender:' +row['gender']]) for _, row in df.iterrows()]
dataset.fit_partial(users=df['customer_id'].unique(), user_features=['gender:' + g for g in df['gender'].unique()])
user_features_matrix = dataset.build_user_features(user_features)

item_features = [(row['item_purchased'], ['season:' + row['season']]) for _, row in df.iterrows()]
dataset.fit_partial(items=df['item_purchased'].unique(), item_features=['season:' + s for s in df['season'].unique()])
item_features_matrix = dataset.build_item_features(item_features)

(interactions, weights) = dataset.build_interactions(
    (row['customer_id'], row['item_purchased'], row['review_rating']) for _, row in df.iterrows()
)
```

```
model = LightFM(loss='warp')
model.fit(
    interactions,
    user_features=user_features_matrix,
    item_features=item_features_matrix,
    epochs=30,
    num_threads=2
)

def recommend_products(model, customer_id, user_features_matrix, item_features_matrix, dataset, num_recommendations=5):
    internal_customer_id = dataset.mapping()[0][customer_id]
    item_mapping = {v: k for k, v in dataset.mapping()[2].items()}
    scores = model.predict(
        internal_customer_id,
        np.arange(len(item_mapping)),
        user_features=user_features_matrix,
        item_features=item_features_matrix
    )
    top_items = np.argsort(-scores)[:num_recommendations]
    recommended_items = [item_mapping[item] for item in top_items]
    return recommended_items

customer_id = 2499
recommendations = recommend_products(model, customer_id, user_features_matrix, item_features_matrix, dataset)
print(f"Recommendations for Customer {customer_id}: {recommendations}")
```

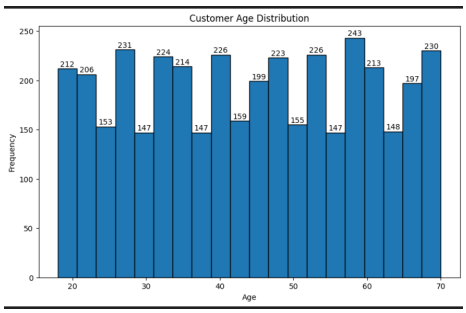
```
from sklearn.ensemble import IsolationForest
import matplotlib.pyplot as plt
import seaborn as sns

# Anomaly Detection
file_path = 'consumerbehaviour.csv'
df = pd.read_csv(file_path)


df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

anomaly_features = df[['purchase_amount_(usd)', 'previous_purchases', 'review_rating']].dropna()

model = IsolationForest(contamination=0.05, random_state=42)
df['anomaly'] = model.fit_predict(anomaly_features)


anomalies = df[df['anomaly'] == -1]
normal_points = df[df['anomaly'] == 1]

anomalies.to_csv("anomalies.csv", index=False)
normal_points.to_csv("normal_points.csv", index=False)

print("Anomalies saved to 'anomalies.csv'")
print("Normal points saved to 'normal_points.csv'")

plt.figure(figsize=(10, 6))
sns.scatterplot(
```

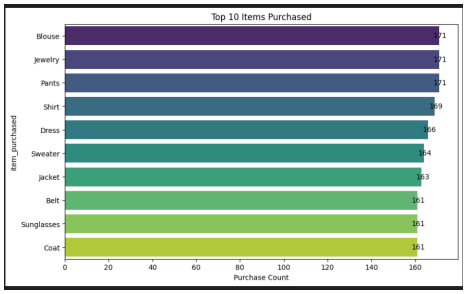
```
sns.scatterplot(
    data=df,
    x='purchase_amount_(usd)',
    y='previous_purchases',
    hue='anomaly',
    palette={1: 'blue', -1: 'red'}
)
plt.title("Anomaly Detection in Purchases")
plt.xlabel("Purchase Amount (USD)")
plt.ylabel("Previous Purchases")
plt.legend(title="Anomaly", labels=["Normal (1)", "Anomaly (-1)"])
plt.show()
```

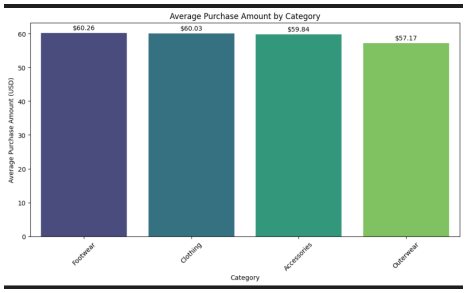
```
#Total count of anomalies and normal points
num_anomalies = len(anomalies)
num_normal_points = len(normal_points)

print(f"Number of red dots (anomalies): {num_anomalies}")
print(f"Number of blue dots (normal points): {num_normal_points}")
```

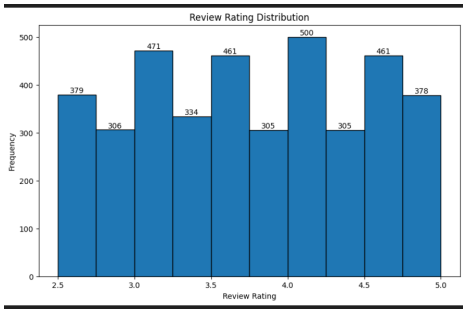
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import pandas as pd

features = df[['age', 'previous_purchases', 'review_rating']].dropna()
target = df.loc[features.index, 'purchase_amount_(usd)']

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)

regressor = RandomForestRegressor(random_state=42)
regressor.fit(X_train, y_train)

df.loc[features.index, 'predicted_purchase_amount'] = regressor.predict(features)

regression_output_file = "predicted_purchase_amount.csv"
df.to_csv(regression_output_file, index=False)
print(f"Predicted purchase amount results saved to: {regression_output_file}")

print("\nDataFrame with Predicted Purchase Amounts:")
print(df[['age', 'previous_purchases', 'review_rating', 'purchase_amount_(usd)', 'predicted_purchase_amount']])
```

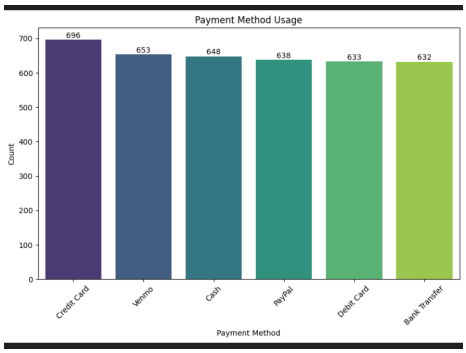
7.2 Appendices B

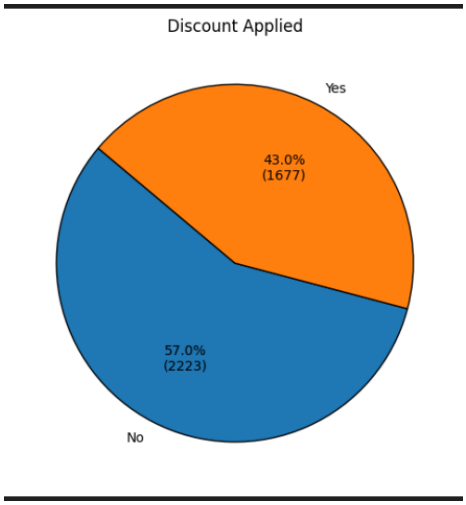


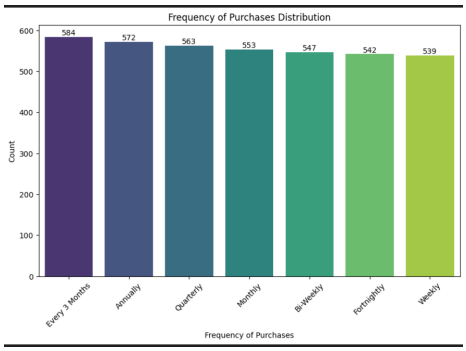


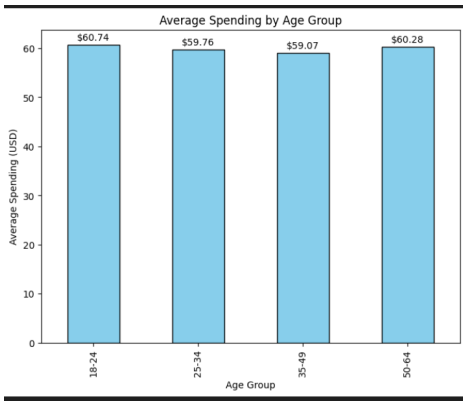


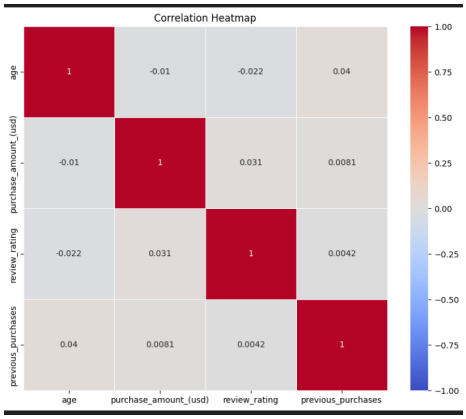


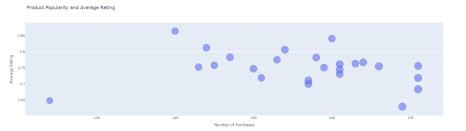

Payment Method Usage
696
653
648
638
633
632
Count
0
100
200
300
400
500
600
700
Credit Card
Venmo
Cash
PayPal
Debit Card
Bank Transfer
Payment Method


Discount Applied
Yes
43.0%
(1677)
57.0%
(2223)
No

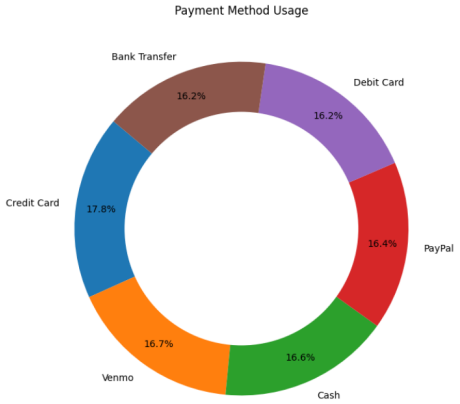

Frequency of Purchases Distribution
584
572
563
553
547
542
539
Count
0
100
200
300
400
500
600
Every 3 Months
Annually
Quarterly
Monthly
Bi-Weekly
Fortnightly
Weekly
Frequency of Purchases

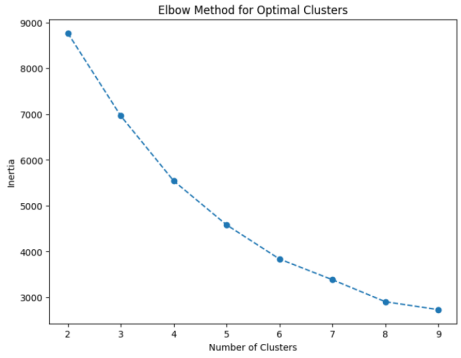

Average Spending by Age Group
$60.74
$59.76
$59.07
$60.28
Average Spending (USD)
0
10
20
30
40
50
60
18-24
25-34
35-49
50-64
Age Group

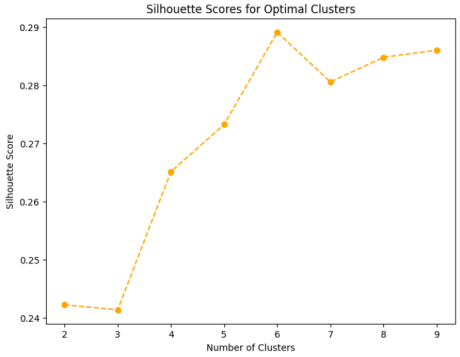

Correlation Heatmap
age
purchase_amount_(usd)
review_rating
previous_purchases
1
-0.01
-0.022
0.04
-0.01
1
0.031
0.0081
-0.022
0.031
1
0.0042
0.04
0.0081
0.0042
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

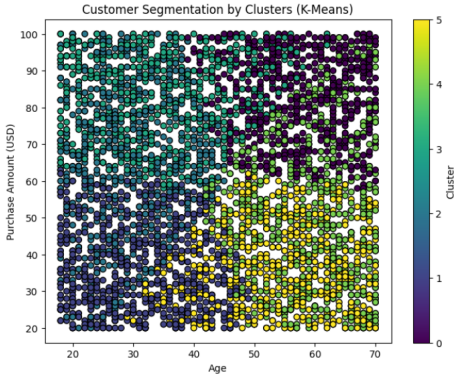

Product Popularity and Average Rating

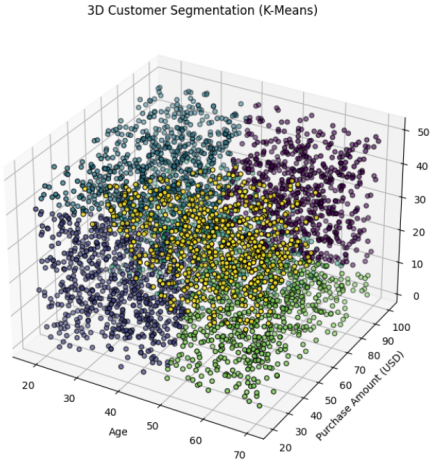

Customer Purchase Hierarchy


Payment Method Usage
Bank Transfer
16.2%
Debit Card
16.2%
Credit Card
17.8%
PayPal
16.4%
Venmo
16.7%
Cash
16.6%


Elbow Method for Optimal Clusters
Inertia
Number of Clusters


Silhouette Scores for Optimal Clusters
Silhouette Score
Number of Clusters


Customer Segmentation by Clusters (K-Means)
Purchase Amount (USD)
Age
Cluster


3D Customer Segmentation (K-Means)
Age
Purchase Amount (USD)