Marketing Campaign (Customer Segmentation and Profiling)

Introduction

In this project, I aimed to segment customers based on specific features using clustering techniques. The features considered for clustering were:

- Marital Status
- Education Status
- Number of Children
- Recency (how recently the customer visited the site)

Objectives

- **Identify Customer Segments**: Group customers into distinct segments based on demographic and behavioral features.
- **Enhance Marketing Strategies**: Provide insights that can be used to tailor marketing campaigns to specific customer segments.
- **Improve Customer Engagement**: Understand the characteristics of different customer groups to foster better engagement and retention.

Goals

- **Optimal Clustering**: Determine the optimal number of clusters that best represent the data.
- **High-Quality Clusters**: Ensure the clusters are well-defined and distinct from each other.
- **Actionable Insights**: Derive meaningful and actionable insights from the clustering results to inform marketing decisions.

Methodology

K-means Clustering K-means clustering is an unsupervised machine learning algorithm widely used for its simplicity and efficiency in partitioning datasets into K distinct, non-overlapping subsets or clusters. Each data point belongs to the cluster with the nearest mean, serving as a prototype of the cluster. The algorithm iterates through the following steps:

- Initialization: Randomly select K initial centroids.
- Assignment: Assign each data point to the nearest centroid based on the Euclidean distance.
- Update: Recalculate the centroids as the mean of all data points assigned to each cluster.
- Iteration: Repeat the assignment and update steps until convergence (i.e., the centroids no longer change significantly). This iterative process ensures that intra-cluster variance is minimized, leading to compact and well-separated clusters.

Elbow Method The Elbow Method is a heuristic used to determine the optimal number of clusters in K-means clustering. It involves plotting the sum of squared distances (inertia) between data points and their corresponding cluster centroids against the number of clusters (K). The goal is to identify the "elbow point," where the rate of decrease in inertia sharply slows, indicating that additional clusters beyond this point provide diminishing returns in explaining the variance in the data.

Silhouette Analysis

Silhouette analysis is a powerful technique for evaluating the quality of clusters. It measures how similar each data point is to its own cluster compared to other clusters, providing an indication of the cohesion and separation of the clusters. The silhouette score for a single sample is calculated as follows:

$s\ (i)=b(i)-a(i)/\max(a(i),b(i))$

s(i)=max(a(i),b(i))b(i)−a(i)

Where:

- $a(i)$a(i) is the mean distance between the sample and all other points in the same cluster.
- $b(i)$b(i) is the mean distance between the sample and all points in the next nearest cluster.

The silhouette score ranges from -1 to 1:

- Close to 1: Indicates that the sample is well matched to its own cluster and poorly matched to neighboring clusters.
- Close to 0: Indicates that the sample is on or very close to the decision boundary between two neighboring clusters.
- Negative score: Indicates that the sample might have been assigned to the wrong cluster.
- A higher average silhouette score indicates better-defined clusters.

In [ ]:
```
# importing dependencies
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from scipy.cluster.hierarchy import linkage
from scipy.cluster.hierarchy import dendrogram
from scipy.cluster.hierarchy import cut_tree
```

In [ ]:
```
df = pd.read_csv("C:\\Datasets\\General_Datasets\\marketing_campaign.csv",sep='\t')
```

In [ ]:
```
df.head(10)
```

Out[ ]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recer
0	5524	1957	Graduation	Single	58138.0	0	0	04-09-2012	
1	2174	1954	Graduation	Single	46344.0	1	1	08-03-2014	
2	4141	1965	Graduation	Together	71613.0	0	0	21-08-2013	
3	6182	1984	Graduation	Together	26646.0	1	0	10-02-2014	
4	5324	1981	PhD	Married	58293.0	1	0	19-01-2014	
5	7446	1967	Master	Together	62513.0	0	1	09-09-2013	
6	965	1971	Graduation	Divorced	55635.0	0	1	13-11-2012	
7	6177	1985	PhD	Married	33454.0	1	0	08-05-2013	
8	4855	1974	PhD	Together	30351.0	1	0	06-06-2013	
9	5899	1950	PhD	Together	5648.0	1	1	13-03-2014	

10 rows × 29 columns

In [ ]:
```
df.isnull().sum()
```

In [ ]:
```
df['Income'] = df['Income'].fillna(df['Income'].mean()) # REPLACING NULL VALUES IN
```

In [ ]:
```
# Creating a new column that represents the customers current age,a new column for
df['Age'] = 2023 - df['Year_Birth']
df['Total_Children'] = df['Kidhome'] +df['Teenhome']
```

In [ ]:
```
df['Age']
```

Out[ ]:
```
0       66
1       69
2       58
3       39
4       42
        ..
2235    56
2236    77
2237    42
2238    67
2239    69
Name: Age, Length: 2240, dtype: int64
```

In [ ]:
```
# Get a general statistical analysis of our data
df.describe()
```

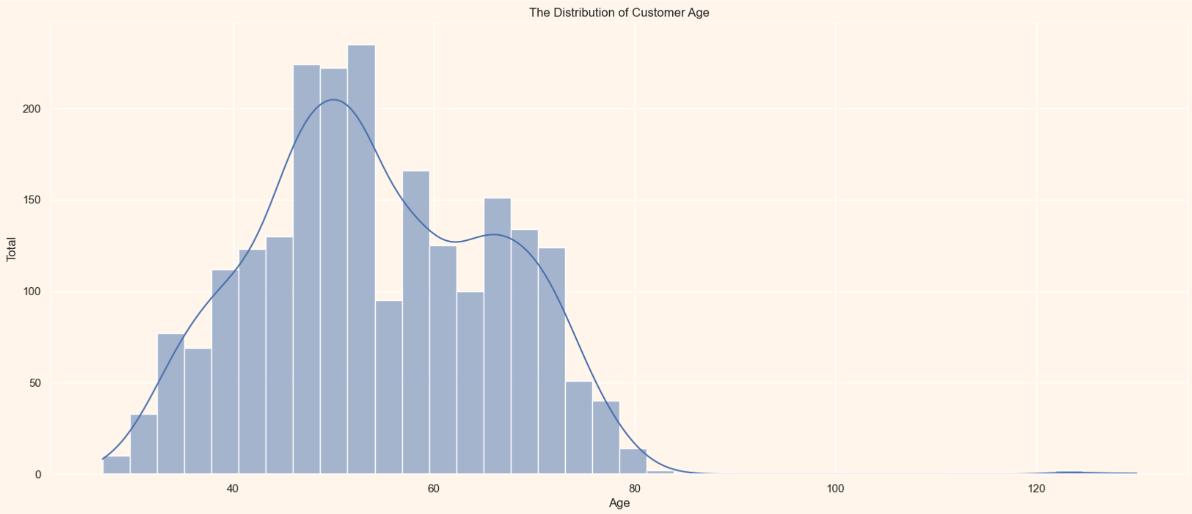
DATA VISUALIZATIONS

In this section I will visualization to have a feel of the data , Some of the questions I have asked of my data include

- What is the distribution of Age?
- Can we group our customers based on their Age ,Education Status or Number of Children?
- How is the distribution of Income?

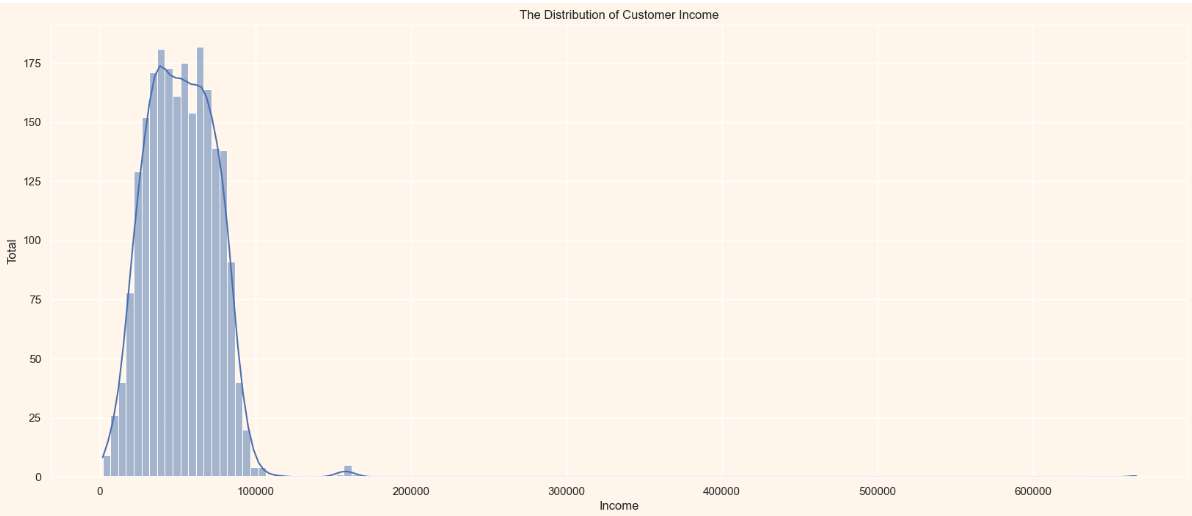
In [ ]:

```
# Plot the distribution of age
plt.figure(figsize=(20,8))
sns.histplot(x='Age',data=df,kde=True)
plt.ylabel('Total')
plt.title("The Distribution of Customer Age")
plt.show() # A majority of our customers are aged between 45-55
```



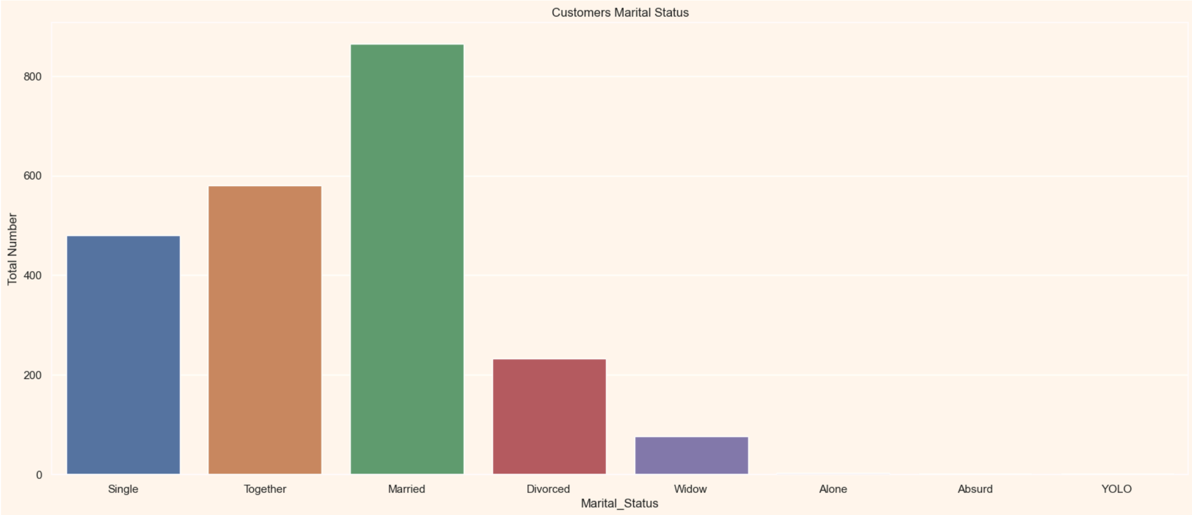
In [ ]:

```
plt.figure(figsize=(20,8))
sns.histplot(x='Income',data=df,kde=True)
plt.ylabel('Total')
plt.title("The Distribution of Customer Income")
plt.show() # A majority of our customers earn 9000
```

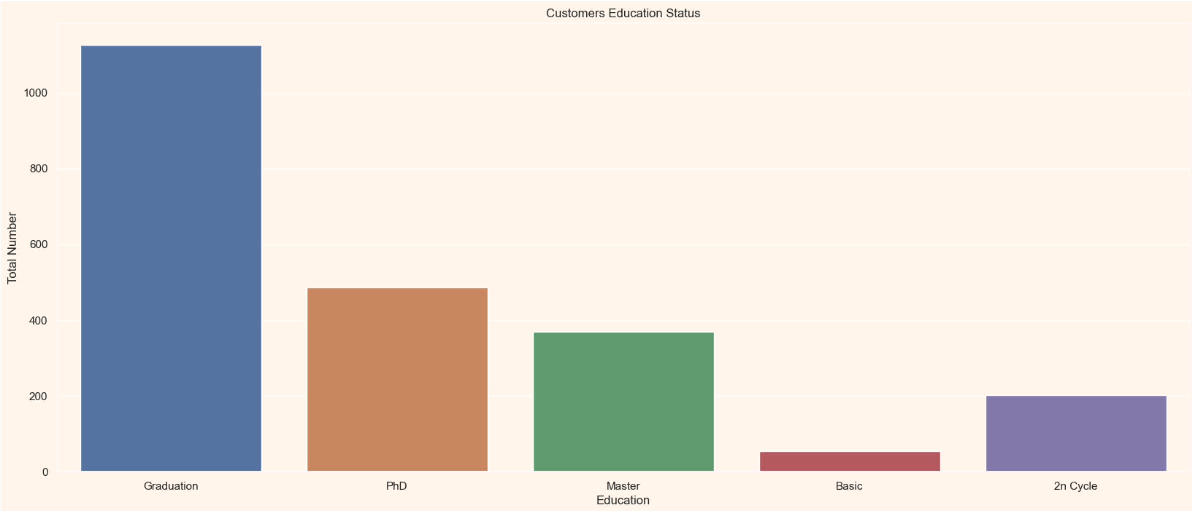


In [ ]:

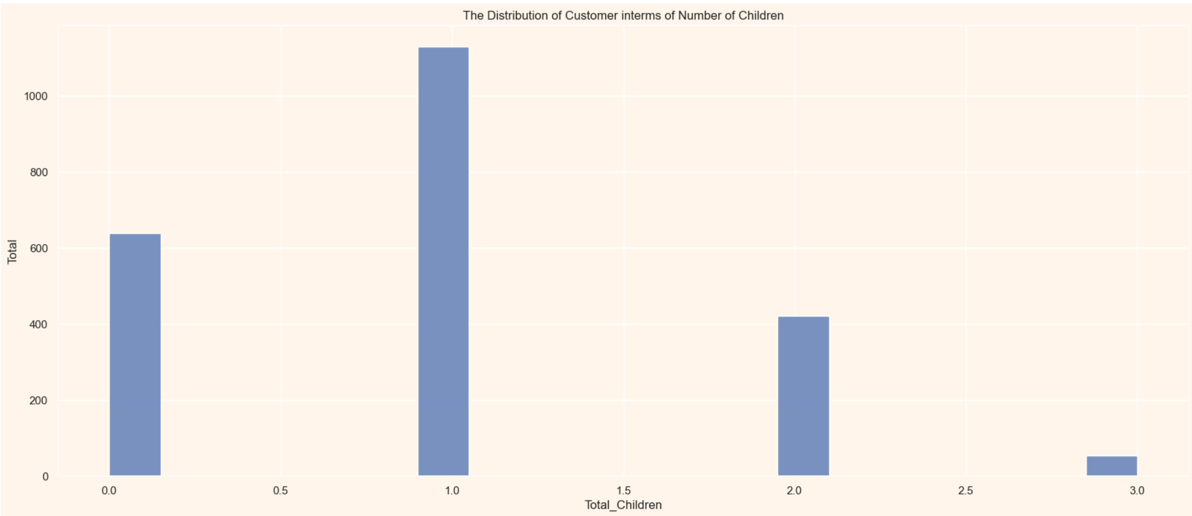
```
plt.figure(figsize=(20,8))
sns.countplot(x='Marital_Status',data=df)
plt.ylabel("Total Number")
plt.title("Customers Marital Status")
plt.show() # Majority of the CUstomers are married
```



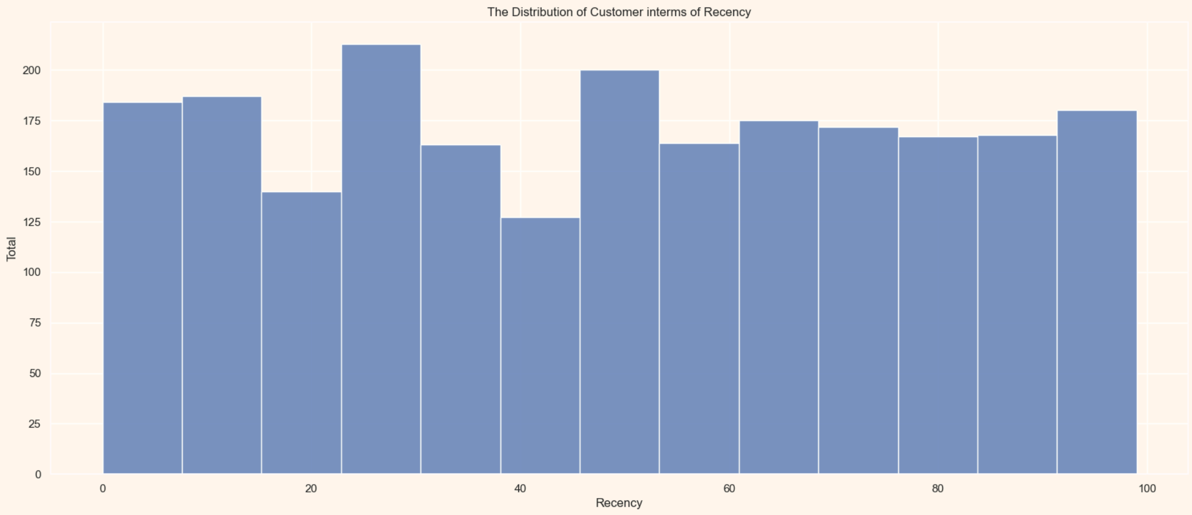
In [ ]:
```
plt.figure(figsize=(20,8))
sns.countplot(x='Education',data=df)
plt.ylabel("Total Number")
plt.title("Customers Education Status")
plt.show() # Majority of the CUstomers are Graduating
```



In [ ]:
```
# Number of Children
plt.figure(figsize=(20,8))
sns.histplot(x='Total_Children',data=df,kde=False)
plt.ylabel('Total')
plt.title("The Distribution of Customer interms of Number of Children")
plt.show() # A majority of our customers  have one child
```

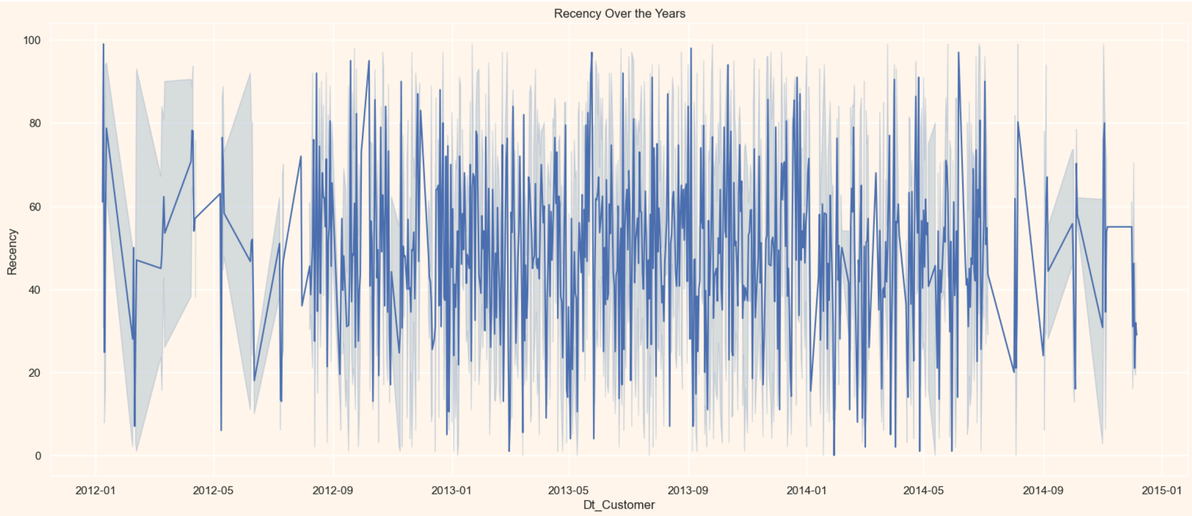


In [ ]:
```
plt.figure(figsize=(20,8))
sns.histplot(x='Recency',data=df,kde=False)
plt.ylabel('Total')
plt.title("The Distribution of Customer interms of Recency")
plt.show()
```



In [ ]:
```
import datetime
df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'],infer_datetime_format=True)
```

In [ ]:
```
# Lets see the recency over the years
plt.figure(figsize=(20,8))
sns.lineplot(x='Dt_Customer',y='Recency',data=df)
plt.title("Recency Over the Years")
plt.show()
```



DATA PREPROCESSING

In [ ]:
```
categorical_columns = ['Education','Marital_Status']
lbl_encoder = {}
for column in categorical_columns:
    lbl_encoder = LabelEncoder()
    df[column] = lbl_encoder.fit_transform(df[column])
```

In [ ]:
```
df.head()
```

Out[ ]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recen
0	5524	1957	2	4	58138.0	0	0	2012-04-09	
1	2174	1954	2	4	46344.0	1	1	2014-08-03	
2	4141	1965	2	5	71613.0	0	0	2013-08-21	
3	6182	1984	2	5	26646.0	1	0	2014-10-02	
4	5324	1981	4	3	58293.0	1	0	2014-01-19	

5 rows × 31 columns

In [ ]:
```
X = df.drop(['Dt_Customer'],axis=1)
X
```

In [ ]:
```
# select our columns to cluster
cluster_df =df[ ['Age','Income','Total_Children','Education','Marital_Status','Rece
cluster_df.columns =['Age','Income','Total_Children','Education','Marital_Status','
scaled_df = StandardScaler().fit_transform(cluster_df)
scaled_df = pd.DataFrame(scaled_df)
```

In [ ]:
```
scaled_df
```

Out[ ]:

	0	1	2	3	4	5
0	0.985345	0.235327	-1.264505	-0.350141	0.251004	0.307039
1	1.235733	-0.235826	1.396361	-0.350141	0.251004	-0.383664
2	0.317643	0.773633	-1.264505	-0.350141	1.180340	-0.798086
3	-1.268149	-1.022732	0.065928	-0.350141	1.180340	-0.798086
4	-1.017761	0.241519	0.065928	1.428354	-0.678332	1.550305
...	...	...	...	...	...	...
2235	0.150717	0.358568	0.065928	-0.350141	-0.678332	-0.107383
2236	1.903435	0.470064	2.726794	1.428354	1.180340	0.237969
2237	-1.017761	0.189106	-1.264505	-0.350141	-1.607669	1.446700
2238	1.068807	0.679035	0.065928	0.539106	1.180340	-1.419719
2239	1.235733	0.024838	1.396361	1.428354	-0.678332	-0.314594

2240 rows × 6 columns

BUILDING A CLUSTERING MODEL

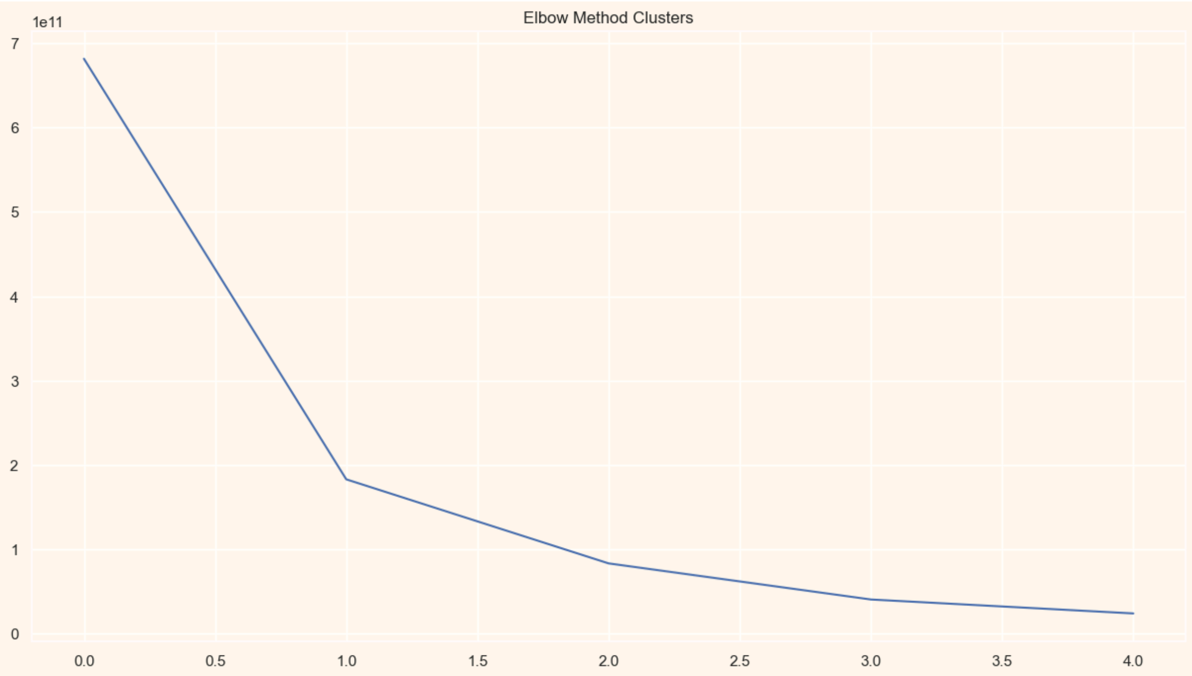
In [ ]:
```
kmeans = KMeans(n_clusters= 4,max_iter=50)
kmeans.fit(cluster_df)
kmeans.labels_
```

Out[ ]:
```
array([3, 3, 0, ..., 3, 0, 3])
```

In [ ]:
```
# Using Elbow Method to get the number of clusters
clust = []
```

```
range_n_clusters = [2,4,6,8,10]
for num_clusters in range_n_clusters:
        kmeans = KMeans(n_clusters=num_clusters,max_iter=50)
        kmeans.fit(cluster_df)
        clust.append(kmeans.inertia_)

plt.figure(figsize=(15,8))
plt.title("Elbow Method Clusters")
plt.plot(clust)
plt.show()
```



Silhoute Analysis

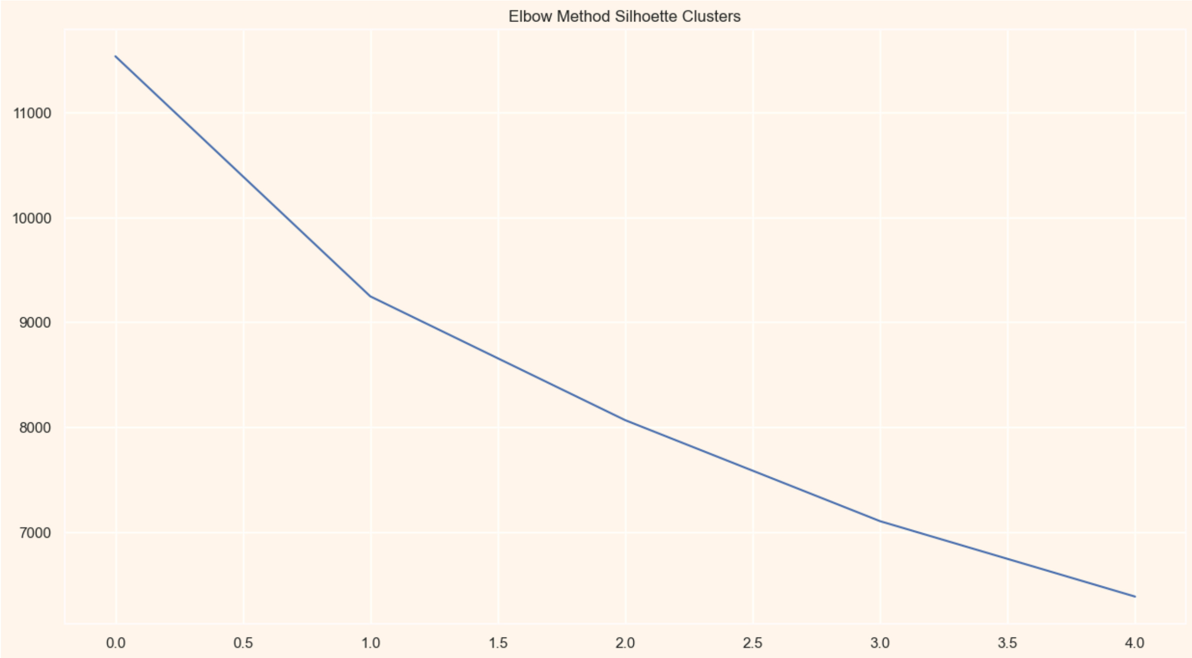
In silhoute analysis the value of the silhouete score ranges from -1 to 1 where a score point of 1 indicates that a data point is very close to the other data point

In [ ]:

```
sill_h = []
range_n_clusters = [2,4,6,8,10]
for num_clusters in range_n_clusters:
        kmeans = KMeans(n_clusters=num_clusters,max_iter=50)
        kmeans.fit(scaled_df)
        cluster_labels = kmeans.labels_
        silhouette_avg = silhouette_score(scaled_df,cluster_labels)
        sill_h.append(kmeans.inertia_)
        print(f"For Cluster{num_clusters} the silhouette_score is {silhouette_avg}'

plt.figure(figsize=(15,8))
plt.title("Elbow Method Silhoette Clusters")
plt.plot(sill_h)
plt.show()
```

```
For Cluster2 the silhouette_score is 0.1434423819682224
For Cluster4 the silhouette_score is 0.14712044831213947
For Cluster6 the silhouette_score is 0.146434657459397
For Cluster8 the silhouette_score is 0.14551758041215115
For Cluster10 the silhouette_score is 0.15441306036563665
```



In [ ]:
```
# FINAL MODEL WITH K = 3
kmeans = KMeans(n_clusters=4,max_iter=50)
kmeans.fit(cluster_df)
kmeans.labels_
```

Out[ ]:
```
array([0, 0, 2, ..., 0, 2, 0])
```

CLUSTER ANALYSIS

In [ ]:
```
cluster_df['Cluster_Id'] = kmeans.labels_
```

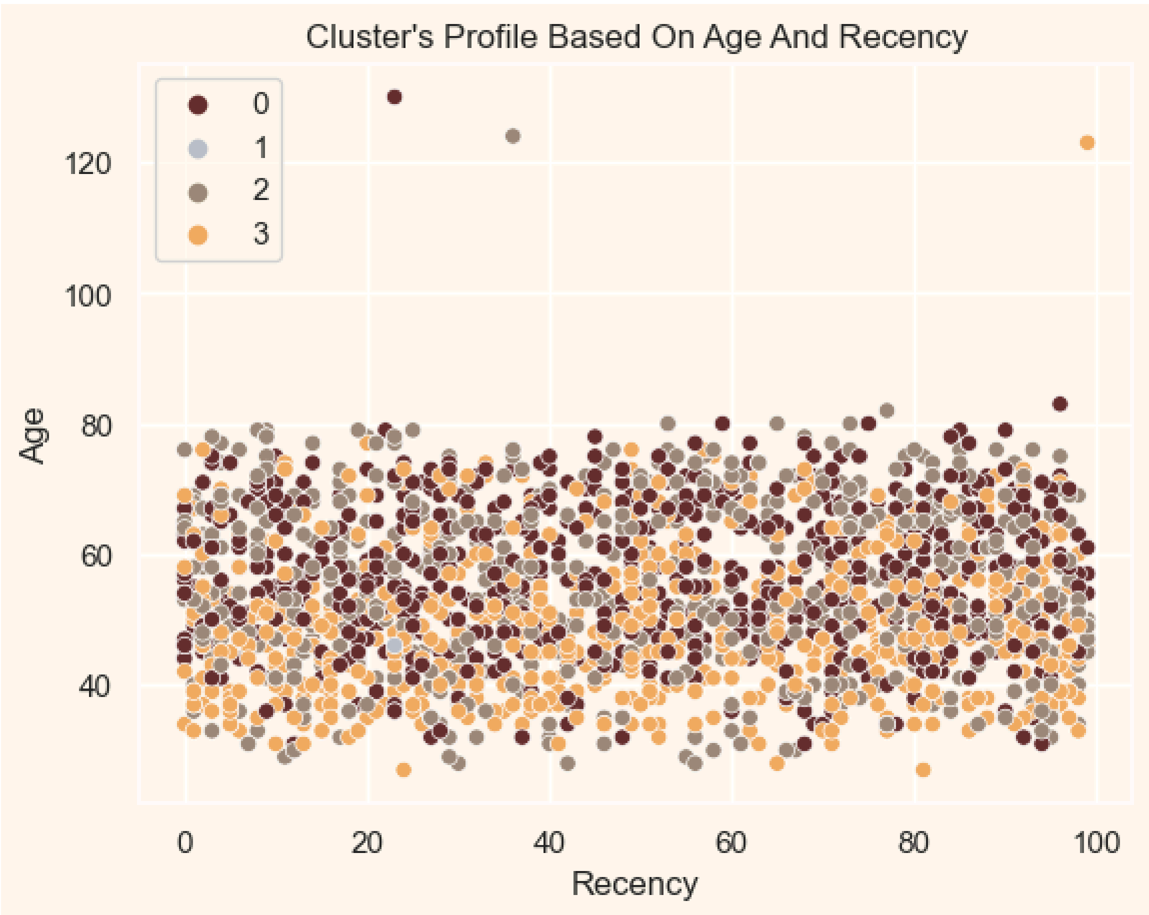
In [ ]:
```
cluster_df.head()
```

Out[ ]:

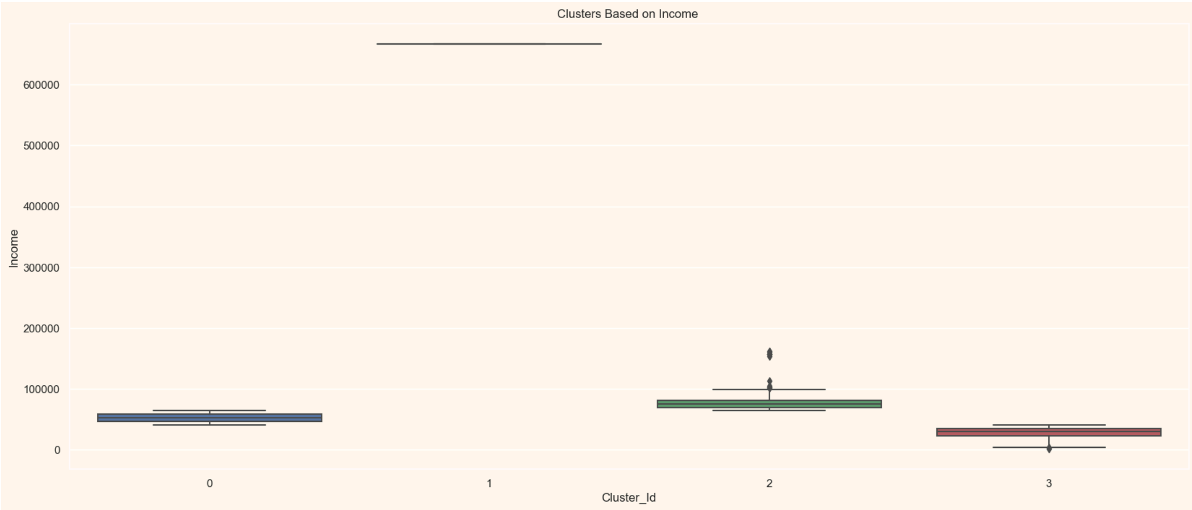
	Age	Income	Total_Children	Education	Marital_Status	Recency	Cluster_Id
0	66	58138.0	0	2	4	58	0
1	69	46344.0	2	2	4	38	0
2	58	71613.0	0	2	5	26	2
3	39	26646.0	1	2	5	26	3
4	42	58293.0	1	4	3	94	0

In [ ]:
```
#Plotting countplot of clusters
pal = ["#682F2F","#B9C0C9", "#9F8A78","#F3AB60"]
pl = sns.countplot(x=cluster_df["Cluster_Id"], palette= pal)
plt.ylabel("Total Number")
pl.set_title("Distribution Of The Clusters")
plt.show()
```

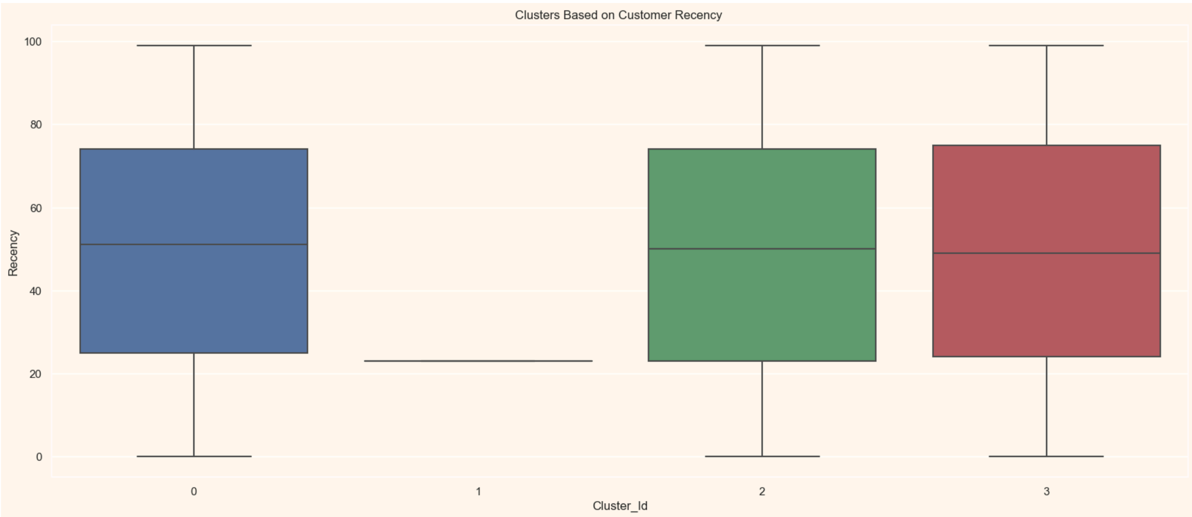
In [ ]:
```
pl = sns.scatterplot(data = cluster_df,x=cluster_df["Recency"], y=cluster_df["Age"]
pl.set_title("Cluster's Profile Based On Age And Recency")
plt.legend()
plt.show()
```



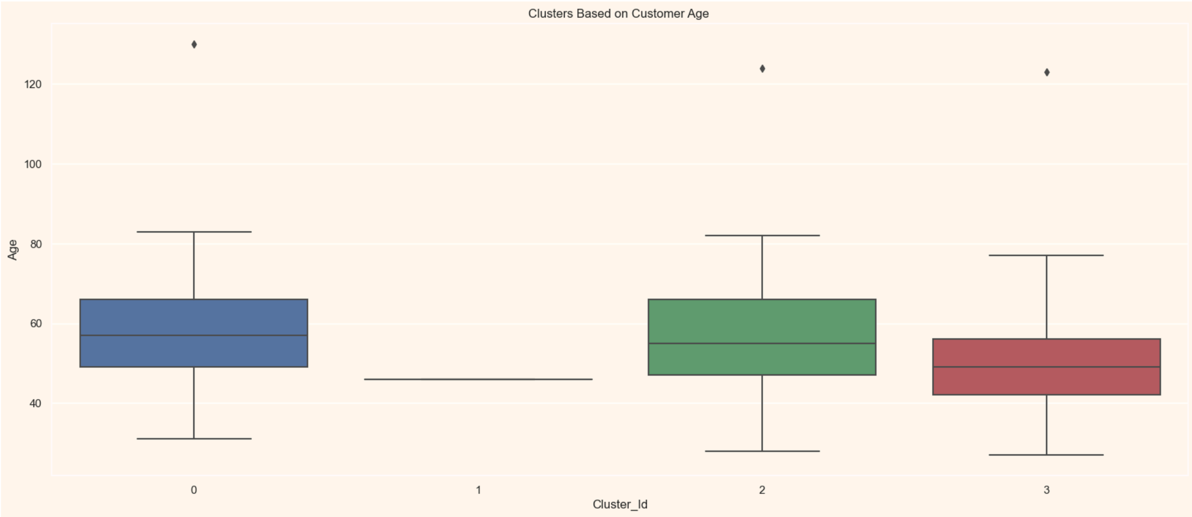
In [ ]:
```
# BoxPlot to show cluster based on income
plt.figure(figsize=(20,8))
sns.boxplot(x='Cluster_Id',y='Income',data=cluster_df)
plt.title('Clusters Based on Income ')
plt.show()
```



In [ ]:
```
plt.figure(figsize=(20,8))
sns.boxplot(x='Cluster_Id',y='Recency',data=cluster_df)
plt.title('Clusters Based on Customer Recency ')
plt.show()
```



In [ ]:
```
plt.figure(figsize=(20,8))
sns.boxplot(x='Cluster_Id',y='Age',data=cluster_df)
plt.title('Clusters Based on Customer Age ')
plt.show()
```



Findings

- The highest silhouette score: 0.15613
- The lowest silhouette score: 0.14342910
- The total number of clusters: 4
- Cluster basis:Income,Recency,Customer Age

Detailed Findings:

- Income: Customers were grouped based on their income levels, providing insights into high, medium, and low-income segments.
- Recency: The clustering revealed patterns in how recently customers interacted with the site, identifying frequent, occasional, and lapsed visitors.
- Customer Age: Age-based clusters highlighted different generational segments, allowing for targeted marketing strategies.

Conclusion

This project effectively utilized K-means clustering to segment customers based on key demographic and behavioral features. The combination of the Elbow Method and silhouette analysis ensured that the chosen number of clusters was both optimal and meaningful. The insights gained from the clustering can significantly enhance targeted marketing efforts, allowing for more personalized and effective customer engagement strategies.

```
In [ ]:
```