Report: Customer Data Segmentation/Clustering

The objective of this analysis was to group the the data from two datasets—customers.csv, and transactions.csv—focusing on each customers spending behaviour and from the days of history with the e-commerce platform. The findings are supported by visualizations and other clustering metrices.

The steps are given below

Basic data cleaning

Basic data cleaning includes data and necessary library loading, searching for null values and duplicates and dealing, changing data types of data

We created a new column named 'how_long_customer' showing the how long(days) that person is a customer

After dropping unnecessary columns , we used 'TotalValue' which shows customers spending and 'how_long_customer' for clustering.

Clustering : K-Means Clustering

K-Means is a popular clustering algorithm in machine learning.

To find the desired number of clusters, we use 'Elbow method'. Here we analyse by plotting graph between sum of square errors(SSE) of different K-value. And we select the K-value with SSE is low and stable

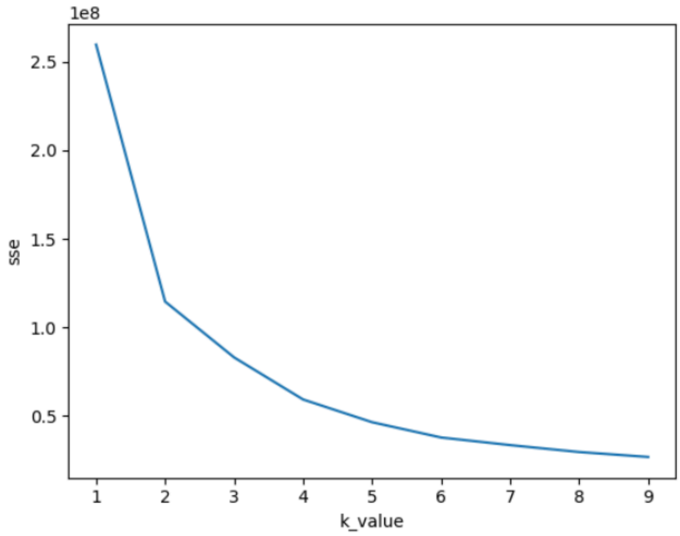
Then import libraries for performing K-Means clustering

Fit the data and predict the cluster

finding k- value

[21]:
```
#finding the ideal number of clusters by finding sum of square errors sse
sse=[]
k_range=range(1,10)
for i in k_range:
    km=KMeans(n_clusters=i)
    km.fit(df1)
    sse.append(km.inertia_)
```

[23]:
```
# plotting sum of square errors with corresponding iterated k value
plt.plot(k_range,sse)
plt.xlabel("k_value")
plt.ylabel("sse")
plt.show()
```

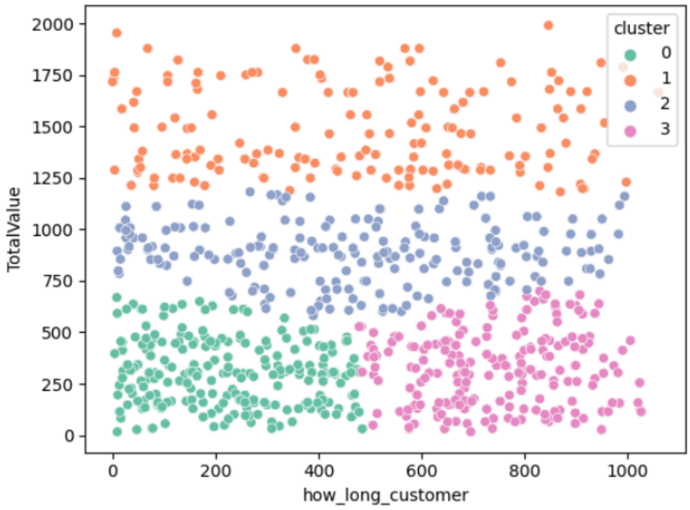


take cluster k value as 4

Plotting and Conclusion

[30]:
```
sns.scatterplot(data = df1,x=df1["how_long_customer"], y=df1["TotalValue"],hue=df1["cluster"],palette="Set2")
```

[30]: `<Axes: xlabel='how_long_customer', ylabel='TotalValue'>`



Here 4 clusters are visible

New customers with low spending is cluster 0

High spending customers are cluster 1

Old customers with low spending is cluster 3

Customers with medium spending is cluster 2

Majority is cluster 0-new customer low spending

Least is high spending customers

Least spending customers are more in number than most spending customers

Evaluation

```
[33]: # Calculate clustering metrics
sil = silhouette_score(df1, km.labels_)
db = davies_bouldin_score(df1, km.labels_)
ch = calinski_harabasz_score(df1, km.labels_)
ari = adjusted_rand_score(df1.cluster, km.labels_)
mi = mutual_info_score(df1.cluster, km.labels_)

# Print the metric scores
print("Silhouette Score:", round(sil,2))
print("Davies-Bouldin Index:", round(db,2))
print("Calinski-Harabasz Index:", round(ch,2))
print("Adjusted Rand Index:", round(ari,2))
print("Mutual Information (MI):", round(mi,2))
```

```
Silhouette Score: 0.4
Davies-Bouldin Index: 0.89
Calinski-Harabasz Index: 900.46
Adjusted Rand Index: 1.0
Mutual Information (MI): 1.38
```

- Silhouette Score(0.4) : This score reveals how similar data points are inside their clusters when compared to data points from other clusters. A result of 0.4 indicates that there is some separation between the clusters, but there is still space for improvement. Closer to 1 values suggest better-defined clusters.
- Davies-Bouldin Index(0.89) : This index calculates the average similarity between each cluster and its closest neighbours. A lower score is preferable.
- The score Index (900.46): calculates the ratio of between-cluster variation to within-cluster variance. Higher values suggest more distinct groups.
- The Adjusted Rand Index (1): compares the resemblance of genuine class labels to predicted cluster labels.
- Mutual Information (MI) (1.38): This metric measures the agreement between the true class labels and the predicted cluster labels. It signifies that the clustering solution captures a significant portion of the underlying structure in the data, aligning well with the actual class labels.

  prepared by

  BIJI T J

Thank you