**CUSTOMER SEGMENTATION USING DATA SCIENCE**

**Phase 4:** Development Part 2

We are employed here to execute several forms of feature engineering such as Income per Age, Principal Component Analysis (PCA), and Model Training and Model Evaluation. We also use machine learning algorithms such as K-Means Clustering, Hierarchical Clustering, Gaussian Mixture Model (GMM), and others.

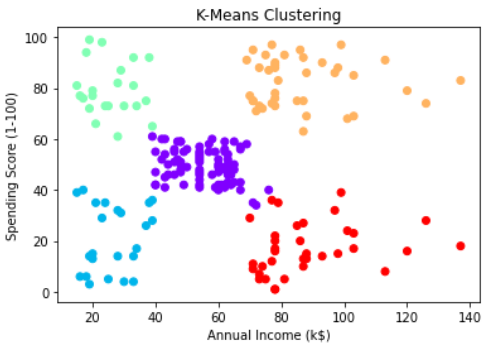
**Income per Age:**

We calculate "Income per Age" by dividing "Annual Income (k$)" by "Age" and save the result in a new column called "Income per Age." The round method can be executed to round the numbers to a given number of decimal places. Finally, the modified dataset is displayed with the new "Income per Age" feature.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **Genre** | **Age** | **...** | **Spending Score (1-100)** | **Cluster** | **Income per Age** |
| 1 | Male | 19 | ... | 39 | 3 | 0.789474 |
| 2 | Male | 21 | ... | 81 | 2 | 0.714286 |
| 3 | Female | 20 | ... | 6 | 3 | 0.800000 |
| 4 | Female | 23 | ... | 77 | 2 | 0.695652 |
| 5 | Female | 31 | ... | 40 | 3 | 0.548387 |

**K-Means Clustering:**

We load the dataset with pd.read\_csv, being sure to specify the correct location to the dataset file. Here, we select the most significant clustering features, namely "Annual Income" and "Spending Score," and place them in the X variable. We indicate the amount of clusters (k) that must be established. The K-Means model is then initialized with KMeans(n\_clusters=k). Using kmeans.fit(X), we fit the model to the data. Using kmeans.labels\_, you may obtain the cluster allocations for each data point. To keep track of which data points belong to which cluster, add the cluster labels to your original dataset. Ultimately, we use a scatter plot with distinct colors for each cluster to visualize the clusters.



**K-Means clustering using the silhouette score:**

Once again, we obtain the dataset here. To evaluate different numbers of clusters, iteration over different values of K is conducted. We integrated the K-Means model for each K, derived cluster assignments, and calculated the silhouette score. Eventually, for each K value, we print the silhouette scores. choosing the K with the highest silhouette score, as it suggests superior cluster quality.

For K = 2, Silhouette Score = 0.2968969162503008

For K = 3, Silhouette Score = 0.46761358158775435

For K = 4, Silhouette Score = 0.4931963109249047

For K = 5, Silhouette Score = 0.553931997444648

For K = 6, Silhouette Score = 0.5379675585622219

For K = 7, Silhouette Score = 0.5264283703685728

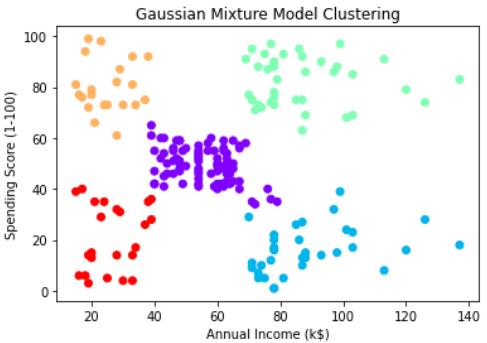
For K = 8, Silhouette Score = 0.4558493609925033

For K = 9, Silhouette Score = 0.45977797620553973

For K = 10, Silhouette Score = 0.45056557470336733

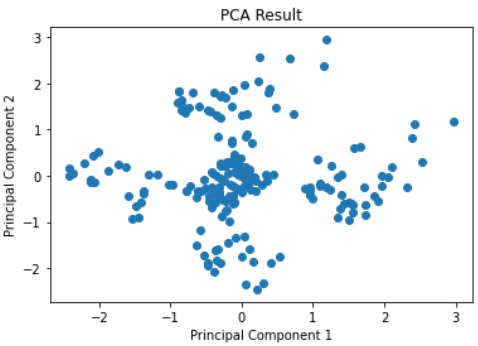
**Gaussian Mixture Models:**

As previously stated, we select "Annual Income" and "Spending Score" as significant clustering factors and keep them in the X variable. You can provide the number of components (clusters) to build using the n\_components variable. To initialise the Gaussian Mixture Model, we are now using GaussianMixture(n\_components=n\_components). After blending the model to the data with gmm.fit(X), we obtain the cluster affiliations for each statistic. Finally, we append the cluster labels to the original dataset to maintain track of which data points belong to which cluster. This investigation concludes with a visual representation of the findings.



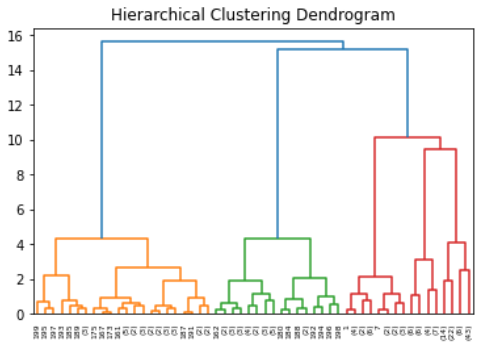
**Principal Component Analysis (PCA):**

We use StandardScaler to standardize the data so that it has a zero mean and unit variance. Before using PCA, it is critical to standardize. Using PCA(n\_components=n\_components), we initialize PCA with the amount of components that are desirable to retain. Then, using pca.fit\_transform(X\_scaled), fit PCA to the standardized data to generate the reduced-dimension data in X\_pca. Finally, visualize what occurred.



**Hiererchical Clustering:**

To support hierarchical clustering, initialise the Agglomerative Clustering model with n\_clusters=None, select the linking mechanism (in this example, 'ward' linkage), and set the distance\_threshold to regulate the granularity of the clusters. We fit the model to the data using agg\_clustering.fit(X) by adjusting the distance\_threshold as needed. Finally, the dendrogram is shown using the dendrogram function from scipy.cluster.hierarchy.

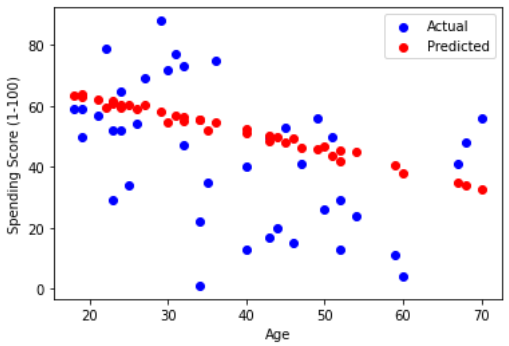


**Data Interpretation:**

We construct predictive models using machine learning frameworks such as scikit-learn (Python) by identifying crucial elements influencing effects. In data interpretation for regression models, we employ Mean Squared Error (MSE) and R-squared (R2) because they provide deeper insight into how well the model fits the data and how much variation is there in the target variable.

Mean Squared Error: 483.55682175408344

R-squared: 0.01963177813218009



**Income Quartiles:**

Here to divide customers into income quartiles based on their "Income per Age" to segment customers according to their income levels. We use the quartile method to accomplish this approach to provide a more granular view of income distribution within the dataset.

**Quartiles divide the data into four equal parts:**

Q1 (25th percentile)

Q2 (50th percentile or median)

Q3 (75th percentile)

Q4 (100th percentile).

