**Harini’s Report :**

**Code 1:**

The code performs a series of operations on a dataset containing columns such as "booking\_date\_time," "distance\_travelled," "time\_taken," "commission\_base\_cost," "driver\_base\_cost," "total\_tax," "total\_trip\_cost," and "ratings."

Here's a summary of the code:

1. The necessary libraries are imported, including pandas, numpy, scikit-learn, matplotlib, and others.

2. The dataset is read from a CSV file using pandas and its column names are printed.

3. Basic information about the dataset is printed, including its dimensions (number of rows and columns) and the first few rows.

4. Relevant columns for segmentation are selected from the dataset.

5. Principal Component Analysis (PCA) is applied to the selected features for dimensionality reduction.

6. The data is preprocessed using MinMaxScaler to scale the features.

7. The optimal number of clusters is determined using the elbow method by calculating the within-cluster sum of squares for different numbers of clusters.

8. The elbow curve is plotted to visualize the optimal number of clusters.

9. K-means clustering is performed on the scaled features for different numbers of clusters, and the within-cluster sum of squares is plotted.

10. The histogram of cluster membership probabilities is plotted.

11. The cluster labels obtained from K-means clustering are added to the dataset.

12. Gaussian Mixture Models (GMM) are applied to the scaled features.

13. The cluster labels obtained from GMM are added to the dataset.

14. A dendrogram is plotted using hierarchical clustering with Ward's method.

15. Silhouette samples and scores are calculated for the K-means clustering.

16. The average silhouette score for each cluster is calculated and printed.

17. Gaussian mixture models are fitted with different numbers of components, and information criteria (AIC, BIC, ICL) are plotted.

18. The desired model with a specific number of components is selected, and cluster assignments from K-means and GMM are compared using a cross-tabulation table.

19. Gaussian mixture models with fixed cluster assignments are fitted and compared with K-means using a cross-tabulation table.

20. Log-likelihoods of the fitted Gaussian mixture models are calculated and printed.

21. Dendrograms are plotted using hierarchical clustering with Ward's method for the original features and scaled features.

22. Hierarchical clustering using Ward's method is performed, and cluster assignments are obtained.

23. A decision tree classifier is built for K-means clustering and plotted.

24. Another decision tree classifier is built for hierarchical clustering and plotted.

25. The clusters are visualized using the first two principal components.

26. The data is split into training and testing sets.

27. A decision tree classifier is trained on the training set for hierarchical clustering.

28. Predictions are made on the test set, and the accuracy of the classifier is calculated.

29. Optionally, the cluster assignments and silhouette values are saved to a new CSV file.

The outputs of the code are summarized as follows:

1. Cluster Assignments (K-means):

- Cluster 5: 1120 instances

- Cluster 7: 1031 instances

- Cluster 1: 691 instances

- Cluster 3: 628 instances

- Cluster 0: 491 instances

- Cluster 2: 440 instances

- Cluster 4: 423 instances

- Cluster 6: 126 instances

2. Cluster Assignments (Gaussian Mixture Models):

- Cluster 4: 1408 instances

- Cluster 0: 1230 instances

- Cluster 2: 807 instances

- Cluster 1: 781 instances

- Cluster 5: 543 instances

- Cluster 6: 150 instances

- Cluster 3: 16 instances

- Cluster 7: 15 instances

3. Average Silhouette Scores for each cluster (K-means):

- Cluster 3: 0.385

- Cluster 5: 0.357

- Cluster 2: 0.300

- Cluster 1: 0.265

- Cluster 7: 0.234

- Cluster 0: 0.234

- Cluster 4: 0.174

- Cluster 6: 0.165

4. Fitted Gaussian Mixture Models:

- Model 2: 2 components

- Model 3: 3 components

- Model 4: 4 components

- Model 5: 5 components

- Model 6: 6 components

- Model 7: 7 components

- Model 8: 8 components

5. Cross-tabulation of Cluster Assignments (K-means vs. GMM):

- Each value in the table represents the count of instances belonging to a particular cluster in K-means (rows) and GMM (columns).

6. Log-Likelihoods:

- Log-Likelihood (Mixture Model with Fixed Clusters): 20.300543660165935

- Log-Likelihood (Fitted Mixture Model): 20.157875600591762

7. Accuracy of the Decision Tree Classifier: 0.968

These outputs provide insights into the clustering results, silhouette scores, model selection, cross-tabulation comparisons, likelihoods, and classification accuracy achieved by the code.

**Code 2:**

The code performs segmentation analysis using various methods on a dataset containing the following columns: "booking\_date\_time", "distance\_travelled", "time\_taken", "commission\_base\_cost", "driver\_base\_cost", "total\_tax", "total\_trip\_cost", and "ratings".

Here's a summary of the code:

1. The code imports necessary libraries and modules for data processing, clustering, classification, and visualization.

2. The dataset is read from a CSV file into a pandas DataFrame.

3. Relevant columns for segmentation are selected from the DataFrame and stored in the variable "features".

4. The data in the "features" DataFrame is preprocessed by scaling the values using the MinMaxScaler.

5. The optimal number of clusters is determined using the elbow method by iterating through different numbers of clusters (1 to 10) and calculating the within-cluster sum of squares (WCSS) for each number of clusters.

6. Gaussian Mixture Model (GMM) clustering is performed using the optimal number of clusters obtained from the elbow method.

7. The cluster labels predicted by GMM are added as a new column, "cluster\_gmm", to the DataFrame.

8. The code prints the counts of instances in each cluster obtained from GMM.

9. The log-likelihood of the fitted GMM model is calculated and printed.

10. Hierarchical clustering using Ward's method is performed on the preprocessed data.

11. A decision tree classifier is built using the features and hierarchical cluster labels.

12. The decision tree is plotted and displayed using matplotlib.

13. Binary logistic regression is performed to predict whether an instance belongs to cluster 0 or not.

14. Multinomial logistic regression is performed to predict the cluster labels obtained from GMM.

15. The coefficients of the binary logistic regression model are printed, showing the relationship between features and the binary cluster label.

16. The coefficients of the multinomial logistic regression model are printed, showing the relationship between features and the GMM cluster labels.

17. Log-likelihood scores are calculated for both the binary and multinomial logistic regression models and printed.

18. Principal Component Analysis (PCA) is applied to the scaled features to reduce the dimensionality to 2.

19. The clusters obtained from GMM are visualized by plotting the instances in a scatter plot using the first two principal components.

20. The segmented dataset, including the cluster assignments and silhouette values, is saved to a new CSV file named "segmented\_dataset.csv".

This code performs clustering, classification, and visualization techniques to gain insights into the dataset and understand the underlying patterns and relationships within the data.

Here's a summary of the outputs from the above code:

1. The "cluster\_gmm" column in the DataFrame shows the counts of instances assigned to each cluster obtained from the Gaussian Mixture Model (GMM) clustering:

- Cluster 8: 1303 instances

- Cluster 0: 927 instances

- Cluster 2: 813 instances

- Cluster 7: 463 instances

- Cluster 5: 376 instances

- Cluster 9: 313 instances

- Cluster 1: 266 instances

- Cluster 3: 235 instances

- Cluster 4: 202 instances

- Cluster 6: 52 instances

2. The log-likelihood score of the fitted Gaussian Mixture Model is 23.576419559532688.

3. The coefficients of the binary logistic regression model are printed, indicating the relationship between the features and the binary cluster label (cluster 0 or not):

- distance\_travelled: 1.767

- time\_taken: -8.606

- commission\_base\_cost: -0.051

- driver\_base\_cost: 3.368

- total\_tax: 1.805

- total\_trip\_cost: 2.577

- ratings: -14.856

4. The coefficients of the multinomial logistic regression model are printed, showing the relationship between the features and the GMM cluster labels:

- distance\_travelled: [-1.160, 4.647, 0.357, -2.477, -1.162, -1.817, 19.198]

- time\_taken: [-0.248, -0.599, -0.390, -0.439, -0.443, -0.438, -21.219]

- commission\_base\_cost: [-5.237, -1.520, -2.855, -3.689, -3.459, -3.579, -10.262]

- driver\_base\_cost: [2.215, 3.999, 0.482, 3.383, 2.048, 2.846, 13.811]

- total\_tax: [4.113, 4.007, 0.020, 3.194, 1.703, 2.526, -8.971]

- total\_trip\_cost: [-4.213, -10.831, -2.568, -2.937, -2.906, -2.918, 15.954]

- ratings: [0.674, -2.876, 3.013, 3.187, 3.407, 3.024, -2.067]

5. The log-likelihood scores for the binary and multinomial logistic regression models are calculated and printed:

- Binary Logistic Regression Log-Likelihood: 0.955959595959596

- Multinomial Logistic Regression Log-Likelihood: 0.8909090909090909

These outputs provide information about the clusters obtained from GMM, the coefficients of the logistic regression models, and the log-likelihood scores, which can help in understanding the patterns and relationships within the dataset and making predictions based on the models.