**Week 4**

**Summary**

**Milestones achieved**

* We performed 5-fold cross-validation on our KNN model. This indicates that our model performs well and consistently across the different folds, with minimal variation in accuracy between the splits. High accuracy and low standard deviation suggest good generalization and stability of the model.

**Clustering**

1. Data Preprocessing

* Data Split: I split the dataset into training and testing sets using train\_test\_split, with an 80-20 division for training and testing. To ensure reproducibility, I set random\_state=42.
* One-Hot Encoding: Since the dataset contained categorical variables, I used one-hot encoding to convert these into numerical format, which is required for most machine learning algorithms. I also ensured that the test set’s columns aligned with the training set by reindexing it.
* Handling Class Imbalance with SMOTE: The target variable (Loyalty) was imbalanced, so I applied SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic samples for the minority class. This ensured the model would not be biased toward the majority class, improving overall performance.
* Feature Scaling: After resampling, I used StandardScaler to scale the features, ensuring that all features were on the same scale, which is particularly important for clustering algorithms like K-Means.

2. Clustering

* K-Means Clustering: I applied K-Means clustering with 4 clusters to assess how well the data could be separated into distinct groups.
* Silhouette Score: After performing the clustering, I calculated the Silhouette Score, which measures the quality of the clusters. A score close to 1 indicates well-defined clusters, while a score near -1 suggests poor clustering.
* PCA for Visualization: To visualize the clusters, I used PCA (Principal Component Analysis) to reduce the data to two components. This allowed me to visually inspect how the clusters were formed, providing insights into their separation.

3. Exploring Multiple Cluster Counts

* I experimented with different numbers of clusters, ranging from 2 to 5, and calculated the silhouette score for each case. This helped me determine the optimal number of clusters by observing how changing the cluster count impacted performance.

4. Sampling and Additional Testing

* To speed up processing, I sampled a smaller fraction (10%) of the resampled dataset and tested clustering with a smaller range of clusters (2 to 5). This allowed me to efficiently explore the cluster performance while reducing computation time.

**Conclusion**

* Concluded evaluation metrics, If our goal is to balance false positives and false negatives (e.g., ensuring correct satisfaction predictions while minimizing errors), **F1 Score** is the most suitable metric. If false positives or false negatives are more important (e.g., predicting satisfied customers more accurately), choose **precision** or **recall** based on your priority. For an overall sense of how well the model distinguishes between satisfied and dissatisfied customers, **ROC AUC** is a solid metric to use.
* Concluded that **Significant combinations** are those where the p-value is less than a predefined threshold (0.05), indicating a statistically significant relationship between the feature and the target variable.
* Low silhouette score (close to 0) suggests poor clustering, where points may be assigned to the wrong clusters.
* The best score achieved is 0.1206 for 2 clusters, but it is still quite low.
* Visualizing the clusters in 2D using PCA helps provide insights into how the clusters are formed and whether there is a clear distinction between groups.
* By experimenting with different numbers of clusters, we aim to find the optimal number of clusters, though the results suggest it is a challenging dataset for clustering.

**Next Steps**

* Investigate K-Means further: Since K-Means assumes spherical clusters and might not perform well on datasets with more complex structures, review the spread of data using dimensionality reduction (like PCA or t-SNE) to see if the dataset can be meaningfully separated with more or fewer clusters.
* EM (Expectation-Maximization) / GMM (Gaussian Mixture Models)
* Hierarchical Clustering:
* Compare clustering performance: Apply these alternative algorithms to our dataset and compare their performance to K-Means using the silhouette score, visualizations, and interpretability.
* Vary the Number of Clusters Manually
* Test a wider range of clusters (e.g., 6-10) to see if the silhouette scores improve.
* Vary the cluster number for each algorithm (GMM, hierarchical) and observe how the number of clusters affects the silhouette score and visualization.
* Use Algorithms to Determine the Optimal Number of Clusters
* Elbow Method (plot)
* Silhouette Method (plot)
* Cluster Validity Indexes
* Explore Davies-Bouldin Index and Dunn Index, which measure how compact and well-separated clusters are.
* Compute the optimal number of clusters using the Elbow method, silhouette method, or other cluster validity measures.
* Use PCA and t-SNE to visualize your clustering results in 2D. This can provide insights into how well-separated the clusters are and whether a certain algorithm performs better in identifying distinct groups

**References**

<https://colab.research.google.com/drive/1JagU9RWa_dCVsposeZTCt5VyoVuXfisb?usp=sharing>