Rancel Hernandez

D212 - Data Mining II

01-05-2025

TASK 1: CLUSTERING TECHNIQUES

A1: PROPOSAL OF QUESTION

Do the clusters identified by k-means clustering reveal distinct patterns when visualized through t-SNE, and are these patterns related to readmission status or any categorical variables in the dataset?

A2: DEFINED GOAL

A goal of this analysis is to determine whether the continuous variables can be used to create clusters that align with patient readmission status. If the cluster labels correspond with readmission rates, the features can be further examined to understand their influence on readmission. Additionally, the analysis may reveal other trends between the features and categorical variables in the dataset. This question can be answered by the available data, and may help to guide initiatives to lower readmission rates and avoid readmission penalties from the Centers for Medicare and Medicaid Services (CMS).

B1: EXPLANATION OF THE CLUSTERING TECHNIQUE

The K-means clustering technique analyzes the data by initializing a set number of random centroids in the feature space made up of the selected continuous variables, and these centroids represent the center points of the clusters. It then assigns each data point to the nearest centroid based on the Euclidean distance to form clusters. After this assignment, the centroids are recalculated as the average of all the features for the data points within the cluster with the goal to minimize the inertia, the sum of squared distances between each data point and its assigned cluster centroid [1]. During this recalculation points can switch between clusters, and this process iterates until the assignments no longer change or the number of defined iterations is reached.

In the end, the data points should be assigned labels based on the cluster they belong to. Consequently, data points within the same cluster should be similar to one another, while exhibiting significant variation compared to those in other clusters. The expected outcome is that the clusters should reveal patterns that correspond with the distribution of relevant categorical variables in the dataset, particularly readmission status, since the selected continuous variables are likely correlated with these categorical features.

B2: SUMMARY OF THE TECHNIQUE ASSUMPTION

The k-means clustering technique assumes that the clusters in the data are of a similar size, meaning they contain approximately the same number of data points. This assumption is due to the algorithm's objective of minimizing the sum of squared distances between points and their centroids because the squared Euclidean distance metric treats all dimensions equally, leading to a preference for clusters with comparable sizes and variances. Consequently, the k-means technique is biased towards forming clusters of a uniform size [2]. This limitation can result in the algorithm failing to capture the true structure of the data, especially when the actual clusters vary in size. For example, data points from a larger cluster might be incorrectly assigned to a neighboring smaller cluster to achieve uniform clusters. Therefore, for k-means to effectively capture the data's structure, it is assumed that the clusters have similar sizes.

B3: PACKAGES OR LIBRARIES LIST

The following python packages were used to support the analysis:

- Pandas - This package was used to load the data from the CSV file into a DataFrame. It was essential for exploring the relationships between continuous and categorical variables and optimized the process of cleaning and transforming the data for analysis.
- Matplotlib - The Pyplot module from Matplotlib was used to plot scatter plots to visualize trends in the data, and plot the distribution of the features through histograms.
- Seaborn - This package was used to efficiently visualize line and scatter plots, with the hue parameter simplifying the process of coloring data points by the desired columns.
- Scikit-Learn - This package was used to visualize and assess the quality of the clusters. The t-distributed Stochastic Neighbor Embedding (t-SNE) function from Scikit-Learn's manifold library was used to reduce the high-dimensional dataset into a 2D feature space for interpretable visualizations. Additionally, the silhouette score was used as a metric to assess the quality of the clusters from the metrics library and will be explained in further depth in later sections.

  The StandardScaler function from Scikit-Learn's preprocessing library was used to standardize the data to have a mean of 0 and a standard deviation of 1, as required for the k-means algorithm, which also helped in identifying potential outliers that could skew the centroids of the clusters [3]. Since the data was standardized, any data point more than three standard deviations away from the mean was considered an outlier [4]. The standardized values were then plotted to visualize the distribution and identify any abnormal outliers. Lastly, the KMeans function from Scikit-Learn's cluster library was used to initialize the k-means model for the analysis. This package was essential for executing the core elements of the analysis.

- Yellowbrick - The Yellowbrick package was used to efficiently visualize the silhouette plot using the SilhouetteVisualizer function. While this could have been accomplished with Matplotlib, Yellowbrick simplified the process into just a few lines of code.
- Sys - The Sys package was used to install the Yellowbrick package and to downgrade the scikit-learn version to 0.24 for compatibility with Yellowbrick.

C1: DATA PREPROCESSING

One data preprocessing goal for the k-means clustering technique is to scale the data to ensure that all features contribute equally to the distance calculations. Without scaling, features with larger ranges can disproportionately influence the positions of the centroids, leading to biased clusters that do not accurately represent the underlying structure of the data. Standardizing the features ensures they have equal influence on the centroids by setting them to have a mean of 0 and a standard deviation of 1. While normalization could be used to scale the features, standard scaling was selected to preserve the relationships between variables, group similar data points, and reduce sensitivity to outliers [5]. The StandardScaler function from Scikit-Learn was used to initialize a scaler object, and then the features were fit and transformed into a NumPy array for use in the analysis.

C2:DATA SET VARIABLES

The following table presents the initial set of variables used in the k-means clustering analysis, along with their respective types.

Initial Variables

Name	Type
Income	Continuous
Intial_days	Continuous
TotalCharge	Continuous
Age	Continuous
Additional_charges	Continuous
Population	Continuous
VitD_levels	Continuous

C3: STEPS FOR ANALYSIS

The following steps are are part of the data preprocessing plan:

1. Standardize Features (cell 5) - The continuous variables used in the k-means clustering were standardized to ensure they have equal importance in the clustering process. The scaled values were then stored in a NumPy array named 'scaled_data.'
2. Identify Outliers (cell 6) - Abnormal outliers can skew the centroids of the clusters, resulting in clusters not representative of the data structure. Since standardization scaled the data to have a mean of 0 and a standard deviation of 1, any data point three standard deviations away from the mean is considered an outlier. After plotting the standardized values, three features showed outliers: 'Income', 'Population', and 'VitD_levels.'

   The outliers in the 'Income' column ranged from $126,063.69 to $207,095.02. While these values were statistical outliers, they reflect realistic middle-class American incomes. They appeared as outliers because the dataset mostly contained patients from lower-income households. Similarly, the outliers in the 'Population' column aligned with typical population densities of U.S. cities, so they were retained. The 'VitD_levels' outliers initially raised concerns due to some values as low as 9.81 ng/ml. However, research indicated that some individuals can have vitamin D levels within this range, so these data points were also considered valid [6]. Thus, there were no outliers identified that needed to be handled.
3. Duplicates (cell 7) - The dataset was checked for duplicate patient entries because these values could skew the centroids by influencing the average calculation when the centroids are updated. This would introduce bias, pulling the centroids away from their actual position. To identify duplicates, the count of unique patient IDs was compared to

the total number of records. Since there should only be one unique ID per patient, any discrepancy would indicate duplicates. The analysis revealed 10,000 unique patient IDs and 10,000 total entries, confirming there were no duplicates to affect the centroids.

4. Missing values (cell 8) - The KMeans algorithm from Scikit-Learn will throw an error if there are any missing values in the DataFrame. Therefore, the total number of missing values for each column was counted using .isna().sum(), and no missing values were detected.

D1: OUTPUT AND INTERMEDIATE CALCULATIONS

The optimal number of clusters in the dataset was two, and this was determined by using an elbow plot, which shows inertia values across a range of cluster numbers. The inertia value represents the average distance between data points and their assigned centroids. As the number of centroids increases, the inertia naturally decreases because adding more centroids reduces the distance between data points and their nearest centroid by increasing the number of centroids in the feature space. The optimal number of clusters is the point where inertia begins to decrease gradually because additional centroids will not distinctly separate the data points from this point forward. After one cluster, the inertia drops significantly from 70,000 to 51,830 with two clusters, and this is a decrease of around 20,000. Beyond this, each additional cluster reduces the inertia by only less than 5,000. Therefore, the elbow occurs at two clusters, where the rate of decrease in the inertia plateaus. The elbow plot is shown in *Figure 1*.

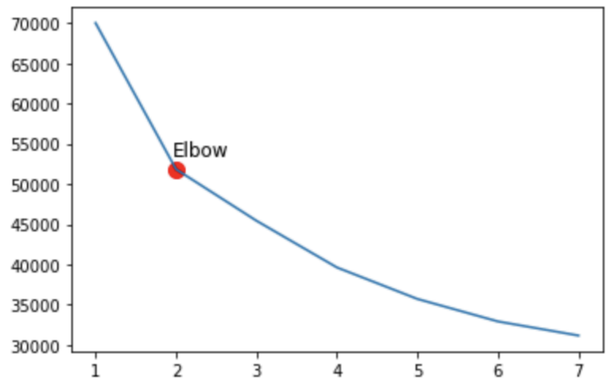


Figure 1: The elbow plot used to determine the optimal number of clusters.

D2: CODE EXECUTION

The code used to determine the optimal number of clusters is located in cells 9 and 10 of the Python file, which will be submitted separately.

E1: QUALITY OF THE CLUSTERING TECHNIQUE

The quality of the clusters can be assessed by examining the inertia values, the distinctiveness of the cluster labels, the de-standardized centroids, and the silhouette scores. The clusters should be of a good quality because the number of clusters was selected at the point at which the rate of decrease in the inertia plateaued, indicating that any additional cluster would not further segment the data points. The optimal number of clusters was determined to be two clusters. Therefore, when plotting the high dimensional data in a 2D feature space, two distinct clusters should be visible.

The t-SNE function from Scikit-Learn was used to reduce the high-dimensional data to a more interpretable form. The learning rate parameter was tuned to optimize the plot. The learning rate controls how much the points are adjusted during each iteration of the optimization process [7]. A high learning rate can cause instability, while a low learning rate may result in the algorithm getting stuck. Therefore, a range of learning rates from 10 to 150, with steps of 20, were utilized to test a varying range. A learning rate of 10 was determined to be the best value because it resulted in the most spread-out data points while maintaining the structure of the data, ensuring that the relationships between data points were preserved [8].

The distinctiveness of the cluster labels was evaluated by coloring the data points in the t-SNE plot and checking if they were contained within their respective clusters. Before coloring the points, two distinct clusters were visible, separated by a thin space. The cluster labels are almost perfectly aligned with these clusters, with only a few outliers randomly scattered within the space. However, this is negligible considering the total of 10,000 data points. This further supports the conclusion that two clusters are optimal and that these clusters are of a good quality. The t-SNE plot is shown in *Figure 2*.

The de-standardized centroids can help determine the quality of the clusters by comparing the difference in the averages of the features for both clusters. The features 'Initial_days' and 'TotalCharge' had significantly different values between clusters. Patients in cluster one approximately had an average 'Initial_days' of 59.68 and 'TotalCharge' of $7,377.08, while patients in cluster two approximately had an average 'Initial_days' of 9.23 and 'TotalCharge' of $3,247.26.

However, there were several features that showed minimal variation between clusters, suggesting they may be introducing noise: 'Income', 'Age', 'Additional_charges', 'Population', and 'VitD_levels.' The significant differences in 'Initial_days' and 'TotalCharge' likely dominated the centroid calculations, potentially reducing the influence of other features. However, this does not indicate poor clustering because the strong differentiation in these key features suggests these clusters are of good quality. Typically, meaningful clusters show strong separation along important dimensions because the significant differentiation between features drives the centroids apart, leading to distinct clusters.

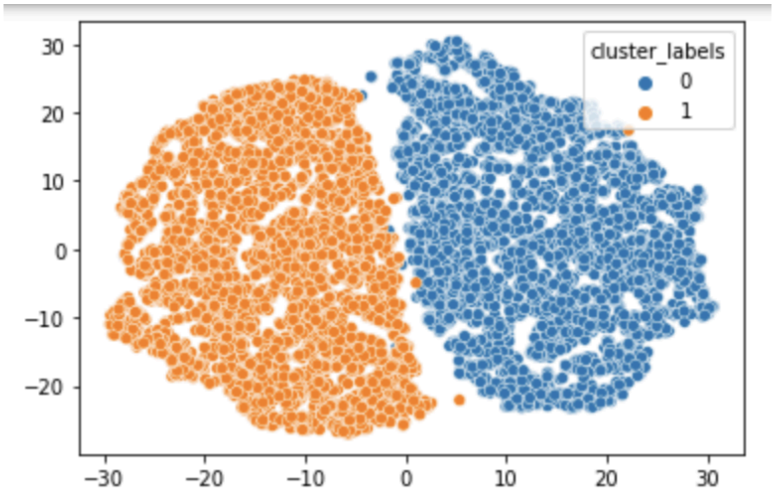


Figure 2: Depicts the t-SNE plot of the initial features.

To further assess the quality of the clusters, the silhouette score was calculated for each k-means model with a varying range of clusters. This metric quantifies the separation between clusters and the closeness within clusters for data points, with scores ranging from -1 to 1 [9]. Scores closer to 1 indicate clusters that are well separated and of good quality, whereas scores near -1 suggest poor clustering. Additionally, the average silhouette score for all data points estimates the overall clustering quality.

The silhouette score peaked at two clusters, reaching approximately 0.2652, and remained consistent for other clustering options at around 0.21. This suggests that two clusters are the optimal number to best segment the data and capture its underlying structure. However, the score of 0.2652 is low and indicates poor clustering quality. This conclusion is further supported by the silhouette plot for the two clusters, which showed that none of the data points achieved scores near 1. Instead, the highest silhouette score observed for individual data points was around 0.5, as shown in *Figure 3*.

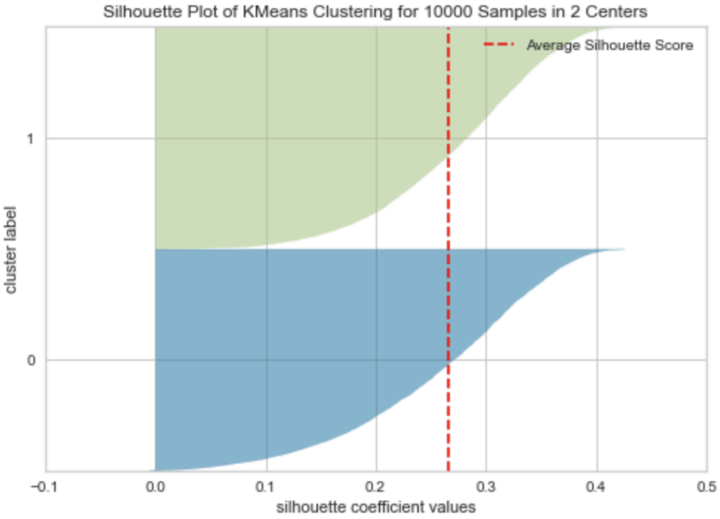
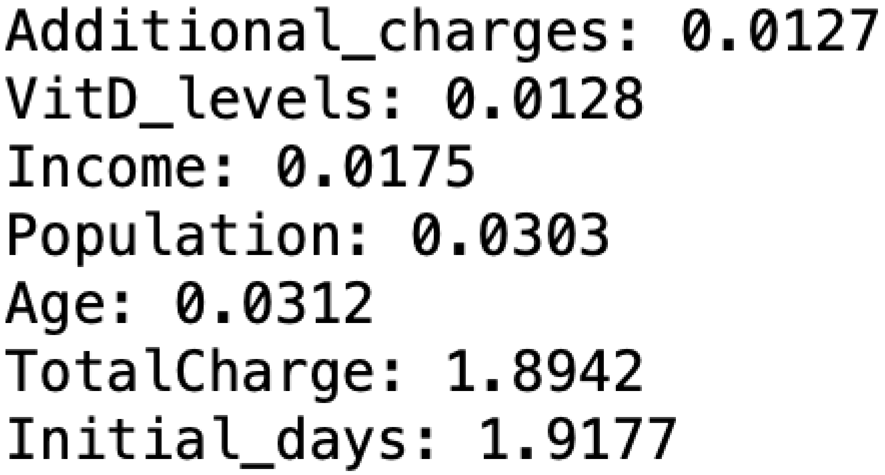


Figure 3: Depicts the silhouette plot of the k-means model with two clusters.

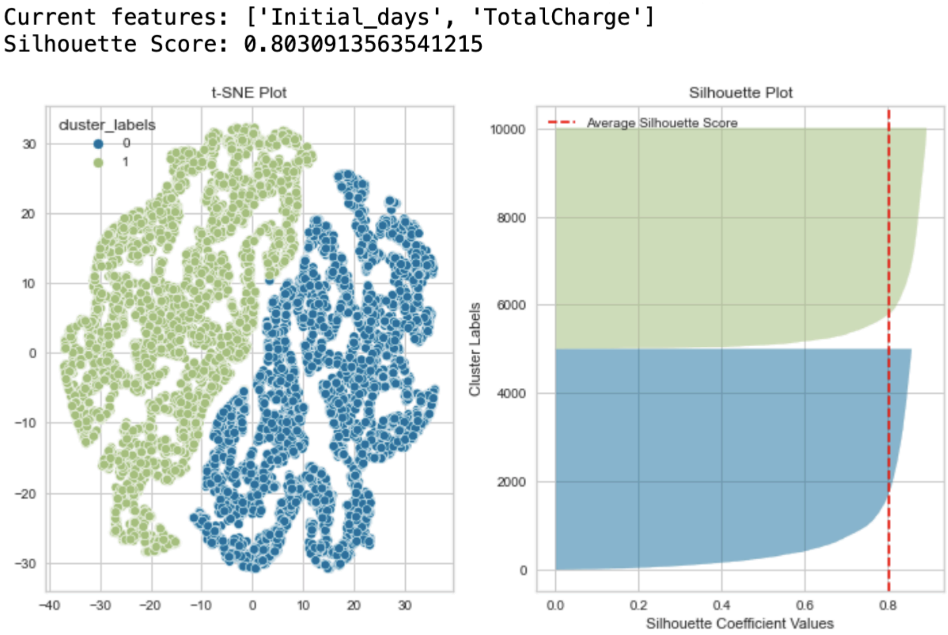
The low silhouette score is likely due to the presence of irrelevant features in the clustering process. Since the silhouette score quantifies the separation between clusters, irrelevant features can add noise that obscures meaningful separations in the feature space, and this could reduce the overall score [10]. To test this hypothesis, the separation of features between clusters was calculated as the absolute difference in feature means for each cluster, with the results shown in *Figure 4*. The features were scaled prior to the calculation, so the absolute difference between the feature means is a reliable indicator of separation between clusters. This approach does assume that features with minimal separation contribute less to distinguishing clusters because they are similar and show little variation. However, this may not always be true because features with minimal separation could still play a significant role in the clustering process. Nevertheless, this method was a safe and straightforward way to test the hypothesis.

The features were then ranked from least to greatest in terms of separation, and one feature was removed at a time. The silhouette score was recalculated, and the silhouette and t-SNE plots were re-created after each removal. This process revealed that removing features with minimal separation significantly improved the silhouette score, and it peaked at approximately 0.8 when only 'TotalCharge' and 'Initial_days' remained, as shown in *Figure 5*. These findings align with the clustering analysis that identified these features as highly influential in the clustering process.



```
Additional_charges: 0.0127
VitD_levels: 0.0128
Income: 0.0175
Population: 0.0303
Age: 0.0312
TotalCharge: 1.8942
Initial_days: 1.9177
```

Figure 4: Depicts the order of features by their calculated separation as the absolute difference of their scaled values between the two clusters.



```
Current features: ['Initial_days', 'TotalCharge']
Silhouette Score: 0.8030913563541215
```

Figure 5: Depicts the silhouette plot with the reduced set of features and increased silhouette score.

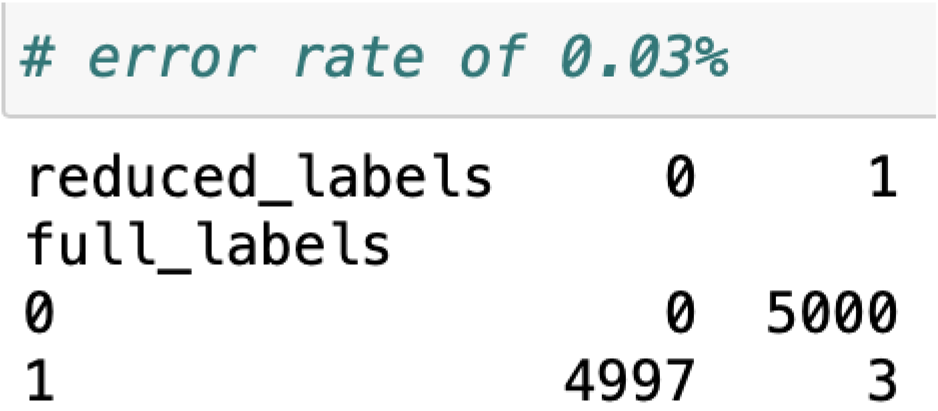
It is important to note that this test was not intended as a feature selection method to reduce the initial set of features. Instead, it was performed to assess the quality of the clusters and test the hypothesis regarding irrelevant features. Additionally, one goal of the analysis is to use all the continuous features in the k-means clustering to visualize their relationships with the categorical variables in the t-SNE plot and uncover underlying trends.

To further support the quality of the clusters created by the k-means model trained on the full set of features, the cluster labels were compared between two models: the one trained on the full feature set and the one trained on the reduced feature set that achieved the highest silhouette score. If the cluster labels matched or were sufficiently similar, the high silhouette score of the reduced model would be an indication of good quality of clusters in the full model. This assumption is based on the idea that irrelevant features did not significantly affect the clustering process if the cluster labels were consistent across the two models.

The comparison revealed that only three data points were misclassified between the models, resulting in an error rate of approximately 0.03%, as shown in *Figure 6*. This result reinforces that the high silhouette score of the reduced model reflects the good quality clusters in the full model, providing confidence that these clusters meaningfully segment the data despite the presence of irrelevant features.

Additionally, the t-SNE method handles irrelevant features by focusing on those that best separate the data, preserving both local and global structures [11]. This was demonstrated in the t-SNE plot, which showed clear clusters for both the full and reduced feature sets. Furthermore, the ‘Additional_Charges’ variable contributed the least to cluster separation and did not significantly influence the clustering process. However, when coloring the t-SNE plot by the "highblood" variable, distinct clusters appeared perpendicular to those formed by the k-means method. This was likely due to its correlation with ‘Additional_Charges,’ which was further explored in the report. This suggests that irrelevant features in the clustering process can still provide valuable insights for the analysis.

In conclusion, the k-means model trained on the full set of features produces clusters of a good quality, despite the low silhouette score. Irrelevant features contributed noise but did not significantly affect the clustering outcome. Additionally, the t-SNE method is designed to handle irrelevant features, and the analysis suggests that even irrelevant features may offer valuable insights for the analysis.



```
# error rate of 0.03%
```

```
reduced_labels     0     1
full_labels
0                  0  5000
1               4997     3
```

Figure 6: Depicts the cross-tabulation table of the cluster labels for the reduced and full model.

E2: RESULTS AND IMPLICATIONS

The clustering analysis revealed that the cluster labels aligned with the distribution of readmitted patients and highlighted an interesting pattern related to patients with high blood pressure. However, only the 'HighBlood' and ‘ReAdmis’ features showed a correlation with the clusters, while other categorical variables resulted in randomly scattered data points with no discernible pattern, an example of which can be seen for ‘Complication_risk’ in *Figure 5*. This suggests that the selected continuous variables do not correlate with the other categorical features, or there is too much noise from irrelevant features affecting the clustering process.

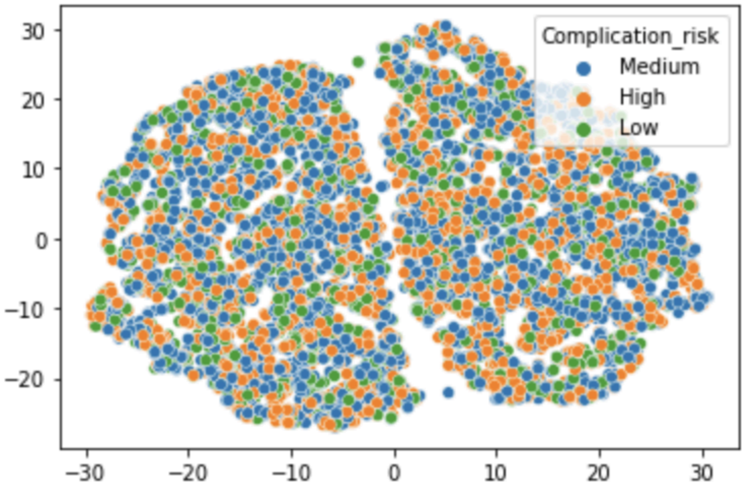


Figure 5: Depicts the t-SNE plot colored by 'Complication_risk.'

According to the cross-tabulation table, all 5,000 patients in cluster one were not readmitted, while 3,669 patients in cluster zero were readmitted. This suggests that the features used in the k-means clustering are correlated with readmission status because the clusters are largely divided by readmission status. However, 1,331 patients in cluster zero were not readmitted, with these points scattered randomly throughout the cluster as shown in *Figure 6*. The cross-tabulation table can be seen in *Figure 7*.

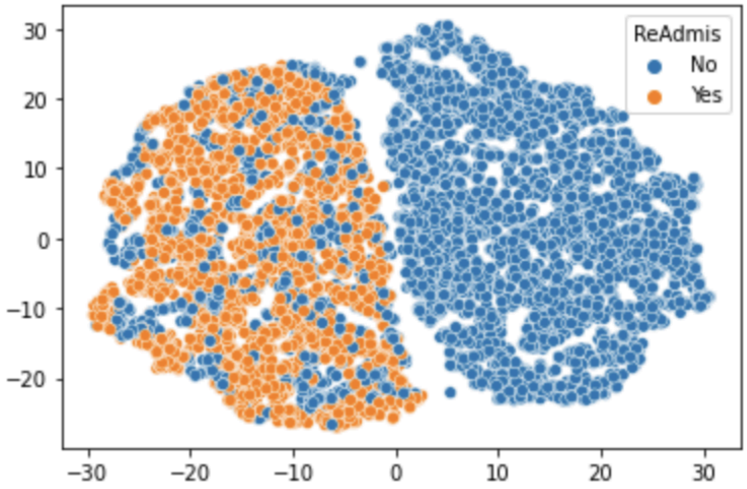
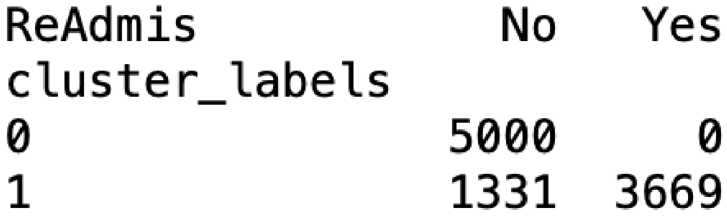


Figure 6: Depicts the t-SNE plot colored by readmission status.



```
ReAdmis           No   Yes
cluster_labels
0               5000     0
1               1331  3669
```

Figure 7: Depicts the cross-tabulation table.

To explore this overlap, the average of each feature was compared between cluster labels and readmission status. Only the features 'Initial_days' and 'TotalCharge' showed similar values:

Name	Initial_days	TotalCharge
cluster zero	9.23	$3,247.26
cluster one	59.68	$7,377.08
non-readmitted	17.41	$3,911.77
readmitted	63.86	$7,728.62

These features were further analyzed using two scatter plots shown in *Figure 8*. The scatter plot on the left, colored by cluster labels, reveals two distinct clusters. Patients who stayed below approximately 34 days, which is the average for 'Initial_days', clustered around 9.23 days, while those who stayed longer clustered around 59.68 days. The notable gap in patient stays around the average suggests that k-means clustering naturally split the patients at this point.

The scatter plot on the right, colored by readmission status, shows that 1,331 non-readmitted patients in cluster one are positioned closer to cluster zero. Therefore, this overlap between clusters can be explained by the correlation between initial hospitalization durations and readmission status. Using the minimum duration for readmitted patients of around 48 days as a threshold reveals that patients who stayed under this threshold had a 0% chance of being readmitted, while approximately 84% of patients who stayed over the threshold were readmitted. Although the clustering technique captures the correlation between readmission status and initial hospitalization durations to some extent, the cluster labels do not fully align with the distribution of readmission status.

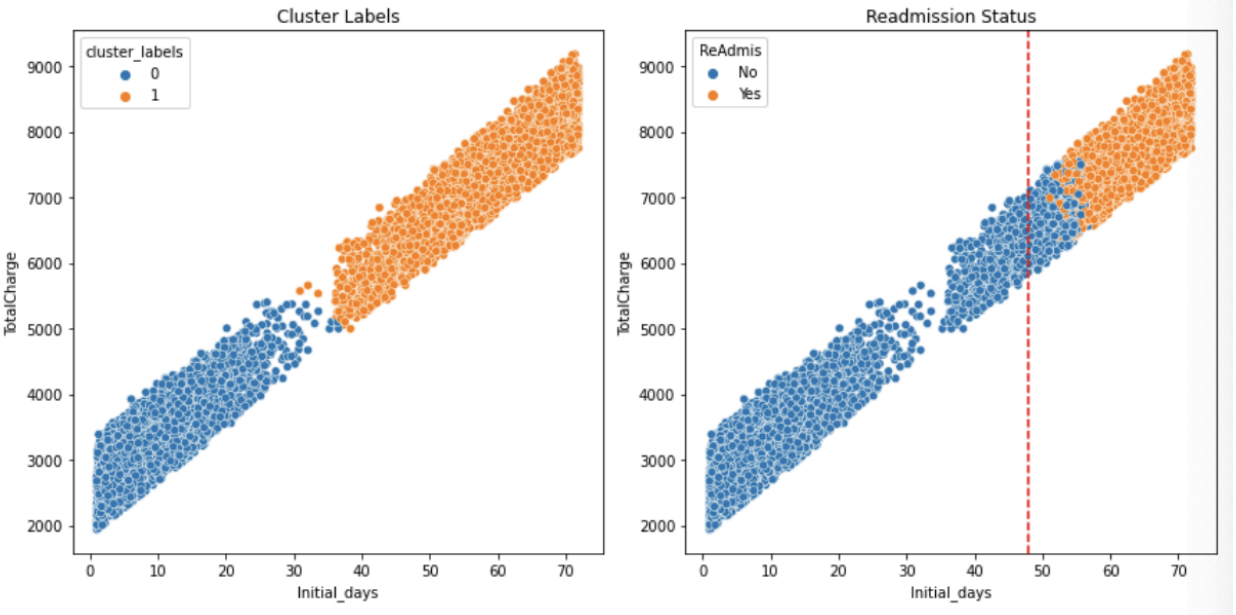


Figure 8: Depicts two scatter plots for 'Initial_days' vs. 'TotalCharge' colored by cluster label and readmission status.

An interesting trend emerged when analyzing the relationship between cluster labels and the 'HighBlood' feature. While the k-means algorithm split clusters along a natural vertical gap in the t-SNE plot, coloring the data points by 'HighBlood' status revealed horizontal clusters. These horizontal clusters formed within the k-means clusters, with no clear gap between them. This trend can be seen in *Figure 9*.

This horizontal separation may stem from features correlated with 'HighBlood' influencing the clustering process. Therefore, the averages of the features were compared between 'HighBlood' status, and which showed that only 'Additional_charges' displayed significant variation. When plotting 'Additional_charges' against 'Age', the visualization revealed that patients with high blood pressure were charged higher additional charges compared to those without high blood pressure. This relationship can be seen in *Figure 10*. This correlation between ‘HighBlood’ status and ‘Additional_charges’ may have influenced the data point positions in the t-SNE plot, resulting in horizontal clusters.

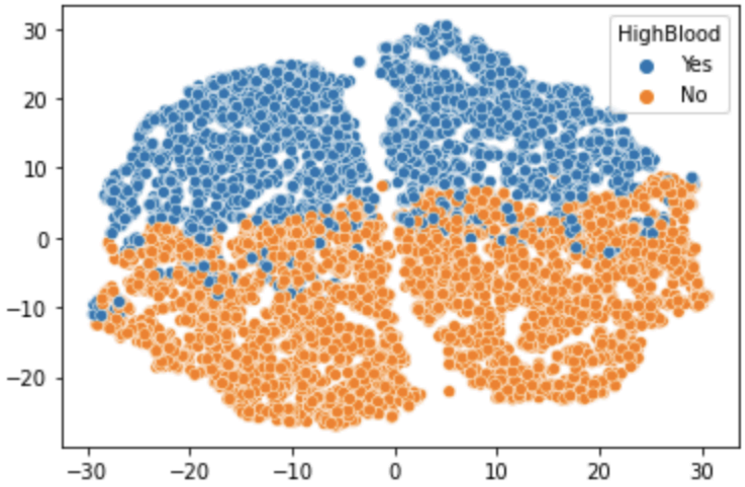


Figure 9: depicts the t-SNE plot colored by ‘HighBlood.’

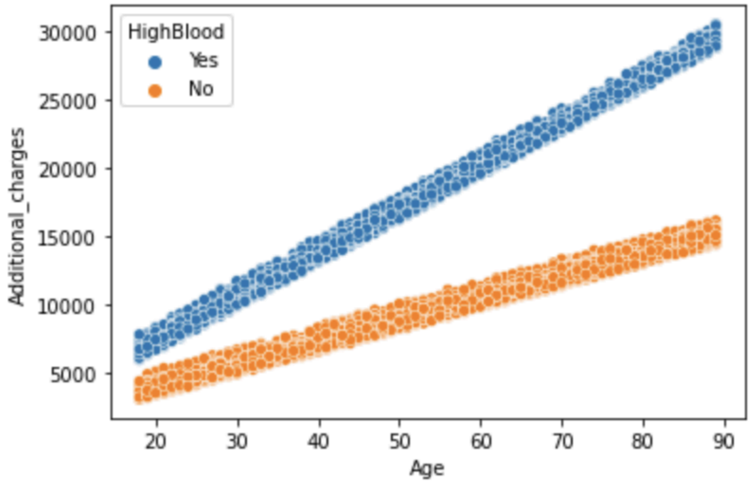


Figure 10: depicts the scatter plot of ‘Age’ vs. ‘Additional_charges.’

E3: LIMITATION

A limitation of the analysis is the imbalance in the dataset regarding readmission status. Since the goal is to identify correlations between the clusters and readmission status, the imbalanced dataset may introduce bias into the clustering process. With around 6,500 non-readmitted patients and 3,500 readmitted patients, there is a bias toward non-readmitted patients. This is particularly problematic because k-means clustering tends to favor clusters of similar sizes, which could result in an overlap between clusters due to the unequal distribution of readmission statuses. In other words, the bias towards uniform cluster sizes meant that k-means was never going to create clusters that were distinctly split by readmission status.

A potential solution would be to randomly remove 2,000 non-readmitted patients, balancing the dataset. However, non-readmitted patients had a wider range for ‘Initial_days’ between approximately 1 and 59 days, while readmitted patients stayed only between 48 and 59 days. The near 40 day difference in ranges might significantly alter the data's structure if many entries were removed, especially since the majority of patients did not stay within the natural gap found in the range of non-readmitted patients of around 34 days. Therefore, the imbalance in the dataset presents a limitation when applying the k-means clustering technique to explore correlations between cluster labels and readmission status.

E4: COURSE OF ACTION

While the analysis failed to create clusters distinctly split by readmission status due to k-means clustering's bias toward uniform cluster sizes and the dataset's readmission status imbalance, it still revealed valuable insights. A recommended next step is to conduct a hierarchical clustering analysis, which doesn't favor uniform clusters and could better capture the structure of the data. Additionally, hierarchical clustering can incorporate categorical variables, potentially uncovering trends that k-means clustering missed by only using continuous variables.

Despite these limitations, the analysis revealed a crucial relationship between initial hospitalization durations and readmission status. Patients staying under around 48 days had a 0% readmission rate, while those staying longer had an approximately 84% chance of readmission. This finding suggests that healthcare providers should prioritize reducing initial hospitalization durations by allocating additional resources to patients approaching this threshold. This intervention strategy would help reduce readmission rates to avoid potential high readmission rate penalties from the Centers for Medicare and Medicaid Services. Ultimately, while the clustering technique had its limitations, the analysis successfully identified trends that can guide interventions to lower readmission rates, avoid fines, and improve patient outcomes.

G: SOURCES FOR THIRD-PARTY CODE

Matplotlib, "matplotlib.pyplot.subplots," [Online]. Available: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.subplots.html. [Accessed: Jan. 17, 2025].

Scikit-learn, "Clustering," *scikit-learn: Machine Learning in Python*, [Online]. Available: https://scikit-learn.org/1.5/modules/clustering.html, [Accessed: Jan. 5, 2025].

Scikit-learn, "sklearn.metrics.silhouette_samples," Version 1.5, [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.silhouette_samples.html#sklearn.metrics.silhouette_samples. [Accessed: Jan. 17, 2025].

Scikit-learn, "sklearn.metrics.silhouette_score," Version 1.5, [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.silhouette_score.html. [Accessed: Jan. 17, 2025].

Scikit-Yellowbrick, "SilhouetteVisualizer," [Online]. Available: https://www.scikit-yb.org/en/latest/api/cluster/silhouette.html. [Accessed: Jan. 17, 2025].

H: SOURCES

Reference List

[1] S. Soni, "The Art and Science of K-Means Clustering: A Practical Guide," *Medium*, Dec. 12, 2019. [Online]. Available: https://medium.com/@sachinsoni600517/the-art-and-science-of-k-means-clustering-a-practical-guide-e71b11638867. [Accessed: Jan. 6, 2025].

[2] J. Ahmad, "The Anatomy of K-Means," *Towards Data Science*, Oct. 2, 2019. [Online]. Available: https://towardsdatascience.com/the-anatomy-of-k-means-c22340543397. [Accessed: Jan. 6, 2025].

[3] P. Sharma, "Comprehensive Guide to K-Means Clustering," *Analytics Vidhya*, Aug. 16, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/. [Accessed: Jan. 7, 2025].

[4] R. E. Lewis, "Determining Outliers Using Standard Deviation," *Study.com*. [Online]. Available: https://study.com/skill/learn/determining-outliers-using-standard-deviation-explanation.html. [Accessed: Jan. 8, 2025].

[5] A. Bhandari, "Feature Scaling for Machine Learning: Understanding the Difference Between Normalization and Standardization," *Analytics Vidhya*. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/. [Accessed: Jan. 9, 2025].

[6] S, Bhowmik, "Vitamin D Test: What You Need to Know," *Health.com*. [Online]. Available: https://www.health.com/vitamin-d-test-8712026. [Accessed: Jan. 9, 2025].

[7] Sachinsoni, "Mastering t-SNE (t-distributed stochastic neighbor embedding)," Medium, Feb. 11, 2024. [Online]. Available: https://medium.com/@sachinsoni600517/mastering-t-sne-t-distributed-stochastic-neighbor-embedding-0e365ee898ea. [Accessed: Jan. 9, 2025].

[8] Scikit-learn Developers, "sklearn.manifold.TSNE," *Scikit-learn: Machine Learning in Python*, 2025. [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.manifold.TSNE.html. [Accessed: Jan. 9, 2025].

[9] Educative, "What is Silhouette Score?," Educative, n.d. [Online]. Available: https://www.educative.io/answers/what-is-silhouette-score. [Accessed: Jan. 20, 2025].

[10] M. McCrory and S. A. Thomas, "Cluster Metric Sensitivity to Irrelevant Features," arXiv, [Online]. Available: https://arxiv.org/abs/2402.12008. [Accessed: Jan. 20, 2025].

[11] "How do PCA and t-SNE handle high-dimensional data with many irrelevant features?" Infermatic. [Online]. Available: https://infermatic.ai/ask/?question=How+do+PCA+and+t-SNE+handle+high-dimensional+data+with+many+irrelevant+features%3F. [Accessed: Jan. 2025].