**Research Question**

Research Question*: What is the relationship between patient Income and length of their initial hospital visit (Initial\_days)?* We can answer this question using *k*-means clustering.

Ultimately, our main goal is to understand the factors that lead to patient readmissions. We’ve seen in prior analyses that Initial\_days has a clear correlation with patient readmissions. Therefore, our goal of this analysis is to understand the impact, if any, of patient income on these factors.

**Technique Justification and Analysis**

I used the k-means technique to analyze this data, so that we can cluster Income and Initial\_days into patient subgroups, which will allow our readmission prediction to be more accurate and take Income into account. Below are the cluster labels of the first 100 expected outcomes in our dataset:

[2, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 2, 0, 0,

2, 0, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2, 0, 0,

0, 2, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2, 2, 0,

0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 2, 0, 2, 2, 0,

2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2]

K-means clustering is an algorithm that attempts to separate, or “cluster”, data points into *k* distinct, non-overlapping groups, where each data point belongs to only one cluster (Dabbura, 2018). New data points can also be assigned to a cluster. The objective of k-means is to segment data into subgroupings, and can be used in this scenario to make predictions about patients. K-means clustering makes some important assumptions that will cause the technique to fail if violated. Specifically, it is assumed in K-means clustering that all clusters are “spherical” and of similar sizes. If we have data that is distributed in a way that produces non-spherical clusters, the k-means algorithm will not properly identify the clusters, since it identifies the centroids first and then selects K data points for the centroids. Similarly, if we have clusters of varying sizes, “the optimization approach used by k-means—effectively minimizing the distance between all the points in each cluster—can lead it astray” (Tait, 2017).

***Python packages used to support analysis:***

**import pandas as pd:** *Reads in the data and allows us to manipulate it.*

**from sklearn.preprocessing import StandardScaler:** *Used to scale the data.*

**from sklearn.cluster import KMeans:** *Create and fits k-means model to data; predicts clusters.*

**import matplotlib.pyplot as plt:** *Used to create the scatterplots of clusters/centroids, as well as inertia plot.*

The code used to perform this analysis is attached separately in the “Task1Code.pdf” file.

**Data Preparation**

I will use two variables to perform the analysis: *Income*, defined as annual income of the patient (or primary insurance holder) as reported at time of admission; and *Initial\_days*, which we can define as the number of days the patient stayed in the hospital during the initial visit. Both of these variables are continuous variables, as required by the k-means method, since the k-means technique is not applicable to categorical data (Kumar, 2021).

My data preparation goals begin with confirming the data is complete and accurate (i.e. clean), and if not, cleaning it as necessary. To do this, I’m going to read in the data in a Jupyter notebook using pandas (in the Python language). This dataset appeared to be clean and ready for analysis upon initial viewing. Since I’ll only be working with two variables (Income and Initial\_days), I want to put them into a separate array (in this case, that array was called *X*). I also want to view a scatterplot of the two variables to see the distribution of the data, and to tell whether a certain number of clusters is noticeable (it appeared there were two clusters upon initially viewing this scatterplot).

Prior to processing this data, I know I will need to make some manipulations – particularly, I will need to scale the data, since the ranges of our variables will largely vary. We want to standardize this data so that all features are on similar scales. To do this, we will use the StandardScaler and Pipeline packages in sklearn.

Cleaned dataset is attached separately.

**Part V: Data Summary and Implications**

The accuracy of this clustering technique is somewhat difficult to quantify. However, one way we can measure is using *inertia*, which is the distance from each point to the centroid of its cluster, and it evaluates how well the k-means clustering worked. An ideal k-means model is one “with low inertia AND a low number of clusters (K). However, this is a tradeoff because as K increases, inertia decreases” (*Clustering: K-means*). Our original model had a low number of clusters (two), but very high inertia (about 10803). I optimized the model using the scatterplot below, which shows number of clusters, *k*, versus inertia. The “elbow”, or value of *k* where the inertia decreases less quickly gives us an indicator of the ideal value of *k*. Here, it appears to be four clusters, where the inertia value equals about 4360.

Chart, line chart

Description automatically generated

Our final scatterplot of data is below. Each color represents one of the four clusters, and the red dot is the center of each cluster.

Scatter chart

Description automatically generated

I also created a cross-tabulation to compare cluster labels with the ReAdmis variable of our initial dataset, which gives us an insight into our goal of predicting hospital readmissions. The table below tell us which readmission status is more and less common for each label.

ReAdmis No Yes

labels

0 3652 0

1 970 2718

2 1346 0

3 363 951

Expected outcomes of first 100 cluster labels:

[2, 0, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 2, 2, 0, 0,

2, 0, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2, 0, 0,

0, 2, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 2, 2, 2, 0,

0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 2, 0, 2, 2, 0,

2, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2]

And the last 100 cluster labels:

[1, 1, 1, 1, 1, 1, 3, 3, 1, 3, 3, 1, 1, 1, 1, 1, 1, 3, 1, 3, 1, 1,

1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

3, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1,

3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1,

1, 1, 3, 1, 1, 1, 1, 1, 1, 3, 1, 3]

The results of our analysis essentially tell us that there are four clusters of patients when in terms of their income and length of initial hospital visit. Using the scatterplot above, the blue dots represent patients with lower incomes and higher Initial\_days, the yellow represent patients with higher incomes and higher Initial\_days, the purple represent patients with lower income and lower Initial\_days, and the green represent patients with higher income and lower Initial\_days. When we use this model to place patients into one of these clusters, we can better predict their behavior, such as whether they will be readmitted.

The limitations of this data analysis are significant. I have a considerable amount of skepticism about the accuracy of this data, and the accuracy of this method is only as accurate as the data itself. Furthermore, there is no way to validate the data to confirm. It is also important to understand that just because these variables have behaved a certain way in the past does not mean they will behave the same way in the future. An additional substantial limitation of our analysis is that income does not actually provide much additional information or predictive qualities about patient readmissions. Therefore, my recommendation to this organization would be to use this clustering method, but not to predict readmissions. It could still be useful, for example, if we wanted to offer discounts or reduced payment plans to certain groups of patients based on their income and length of initial visit. We should also further explore k-means techniques to create clusters of different variables. This clustering technique has shown to hold useful implications and we should use it moving forward.

**References**

*Clustering: K-means*. Codecademy. (n.d.). Retrieved December 6, 2021, from https://www.codecademy.com/learn/machine-learning/modules/dspath-clustering/cheatsheet.

Dabbura, I. (2018, September 17). *K-means clustering: Algorithm, applications, evaluation methods, and drawbacks*. Medium. Retrieved December 6, 2021, from https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

Kumar, S. (2021, May 7). *Clustering algorithm for data with mixed categorical and numerical features*. Medium. Retrieved December 6, 2021, from https://towardsdatascience.com/clustering-algorithm-for-data-with-mixed-categorical-and-numerical-features-d4e3a48066a0.

Tait, A. (2017, January 31). *Assumptions can ruin your K-means clusters*. Learning Tree Blog Assumptions Can Ruin Your KMeans Clusters Comments. Retrieved December 6, 2021, from https://blog.learningtree.com/assumptions-ruin-k-means-clusters/.