B. Describe the purpose of this data mining report by doing the following:

B1. Propose one question relevant to a real-world organizational situation that you will answer using one of the following clustering techniques: k-means, hierarchical

The goal of unsupervised machine learning is exploratory instead of predictive. The data is created to facilitate groupings that uncover patterns or segments within the data. Clustering relies on continuous data. I want to explore what groups of women patients are revealed based on age, income, household size, and total charges. This focuses on the demographic variables to create meaningful subgroups within the patient data. Such focus and meaning can lead to not only better resource allocation but also effective, targeted interventions.

B2. Define one goal of the data analysis. Ensure your goal is reasonable within the scope of the selected scenario and is represented in the available data.

The goal is to help the hospital understand how it can improve patient relationship management in a cost-effective way for all patients, regardless of class. By performing K-means clustering, it is expected that patterns will emerge among low-income and high-income earners, child-bearing women, and post-menopausal women. All have different needs and exploring how these groups are similar and different when it comes to managing patient costs can lead to reduced bias and enhanced care.

C. Explain the reasons for your chosen clustering technique from part B1 by doing the following:

C1. Explain how the clustering technique you chose analyzes the selected dataset. Include expected outcomes.

K-means clustering will analyze patterns among the patient dataset based on similarities before "clustering results in a new feature containing group assignments," Aimee Schwab-McCoy wrote

in the ZyBooks titled "D 603: Machine Learning." The machine learning model will perform steps:

- It will examine the women's patient data, defining clusters using "elbow and silhouette methods that help determine the optimal number of clusters" (Schwab-McCoy).
- It will calculate centroids, which aid in assigning instances to clusters.
- It will calculate the squared distances of the instances to the centroid, recomputing the clusters' means. In simpler terms, this examines how close or far patients are in terms of income, doctor visits, household, and charges.
- The process will repeat until the clusters are as tight as possible, completing stabilization among cluster assignments, while also minimizing variation.

The expected outcome is that subgroups with similar characteristics are revealed within the female patient data. In the simplest terms, low income versus high income, small households versus big households, etc. Clearly seeing the demographics can help the hospital understand its patients, which will inform financial planning, preventive care, and targeted outreach based on the needs of each demographic.

C2. Summarize one assumption of the clustering technique.

While I can encode my data and use it for K-means, that is not the standard. This is because K-means computes the mean of the points in the clusters before calculating the distance. Encoding categorical data can distort the clusters. And it is recommended to use k-prototypes or Hierarchical clustering. Therefore, one assumption that must be met is that data is continuous, and that all data is scaled with mean and standard deviations normalized.

C3. List the packages or libraries you have chosen for Python or R and justify how each item on the list supports the analysis.

I used the following Python packages to support specific parts of my analysis:

- **datetime**: I used this to add the date to my filename when running my class SaveIt, which I built a more efficient way to back up my file.
- **pandas**: I used pandas to load, clean, and explore my data. This helped me decide on my question, what features can I use to help me gain an understanding of the women patients related to class, household, and expenditure.

- **numpy**: I used numpy for numerical operations, such as averaging cross-validation scores and handling arrays.
- **matplotlib.pyplot**: I used to visualizations of the clusters, displaying overlap and separation as well as knee and elbow.
- **os, sys, re**: I used these to manage files and clean up column names.
- **scikit-learn (sklearn)**: The primary machine learning library used for this analysis:
    - **sklearn.cluster.KMeans:** This is the core model used for the k-means clustering algorithm, which allows me to group patients into clusters based on similar demographics and expenditures.
    - **sklearn.preprocessing.StandardScaler:** Because many of my features are measured on different scales, such as income in dollars versus age, I have to standardize features by subtracting the average, so the data is centered around zero and then scaling to unit variance by dividing by the spread. Doing so will result in values ending up between –1 and +1. This allows measurements to have equal weight in the K-Means model.
    - **sklearn.metrics.silhouette_score and silhouette_samples:** To evaluate the clusters' quality, I use the silhouette score, which measures how well each point fits within its cluster. "The silhouette method calculates the silhouette coefficient for each instance," Schwab-McCoy writes in the book D 603: Machine Learning. This results in the cluster with the highest mean score. Silhouette coefficients fall between -1 and +1. A high score means the point is much closer to its group than others, while a low score means that it is about the same distance to others as it is within its own. A negative score means the point is closer to other groups than it is to its own. Silhouette_samples also compares datapoints to other clusters and helps in selecting an appropriate number of clusters.
    - **sklearn.pipeline.make_pipeline: Using** make_pipeline links together steps in the preprocessing the data. I used it to link StandardScaler and K-Means, thereby creating a single streamlined workflow. This creates consistency during both training and prediction and reduces leakage.
    - **sklearn.decomposition.PCA:** Principal Component Analysis can transform large datasets while still maintaining “most of the information” (Habibimoghaddam). In order to visualize the clusters in a 2D space, I had to reduce my 6 features to 2 principal components using the PCA model. Using linear algebra, PCA transforms the data into eigenvectors and eigenvalues before selecting the top components with the highest eigenvalues, which best capture the data, according to the article*, Principal Component Analysis(PCA),* by GeeksforGeeks. I followed much of Habibimoghaddam’s code that he wrote in his article, *Customer Segmentation (K-Means Clustering & PCA),* adapting the code to meet my needs.
    - **kneed.KneeLocator:** According to the Kneed documentation on GitHub, this package identifies the knee and elbow point on the inertia curve. Commonly used to determine the optimal number of clusters in k-means, it provides a way to

balance the model's complexity with its cluster quality. I chose the curve="convex" parameter to identify the elbows.

D. Perform data preparation for the chosen dataset by doing the following:

D1. Describe one data preprocessing goal relevant to the clustering technique from part B1.

I standardized the data using StandardScaler() before performing the K-Means clustering analysis.

K-means relies on distance calculations to form clusters. Therefore, variables with different scales and units will impact the model, causing imbalance and increasing inaccuracy. For example, the variable TotalCharge can be in the thousands of dollars while the variable Children is at most seven. Without applying the StandardScaler(), the clustering would mostly reflect TotalCharge while ignoring the smaller-scale features such as Children.

I used the make_pipleline function to apply StandardScaler, transforming each variable to have a mean of 0 and a standard deviation of 1. This balances out my model, making certain that the distance calculations for my clusters are meaningfully reflected across age, income, doctor visits, and charges.

code

```
kmeans = KMeans(n_clusters=5, n_init=10, max_iter=100, random_state=123) scaler =
StandardScaler() pipeline = make_pipeline(scaler, kmeans)
pipe = pipeline.fit(Xw)
#pull out the scaled data and store in variable
Xw_scaled = pipeline.named_steps['standardscaler'].transform(Xw)
df_scaled = pd.DataFrame(Xw_scaled, columns=Xw.columns)
df_scaled.head()
```

D2. Identify the initial dataset variables you will use to perform the analysis for the clustering question from part B1, and label each as continuous or categorical.

I picked six variables to help me answer the question of, *"What groups of female patients are revealed based on age, income, doctor visits, household size, and total charges?,"* I picked six variables from the medical dataset:

- **Children:** A discrete number (datatype: int64). This shows the number of children in the household.

- **Age:** A continuous number measured in age (datatype: float). This shows the age of the patient.
- **Income**: A continuous number (datatype: float). This shows the annual income of the patient in dollars.
- **DocVisits:** A discrete number (datatype: int). This shows the number of doctor visits per year.
- **TotalCharge:** A continuous number (datatype: float). This shows the total charges billed in dollars.
- **AdditionalCharges**: A continuous number (datatype: float). This shows miscellaneous charges for procedures, medicines, and treatments added in dollars.

D3. Explain each of the steps used to prepare the data for the analysis. Identify the code segment for each step.

In order to prepare my data for clustering, I first loaded my dataset into my Jupyter Notebook using pandas. Then, I filtered my dataset to focus on female patients before storing the filtered data in the variable 'women'. I then selected the six columns I wanted for clustering, storing the six columns in the Xw variable before scaling the rows using the StandardScaler so that all the variables were placed on the same scale before conducting K-Means. Both the StandardScaler and K-Means were applied using the make_pipeline() function.

Below is the code:

```
df = pd.read_csv('data/medical_clean_2025-09-27_v1.csv')
women = df[df["Gender"] == "Female"]
Xw = women[["Children", "Age", "Income","DocVisits", "TotalCharge", "AdditionalCharges"]]
scaler = StandardScaler()
Xw_scaled = scaler.fit_transform(Xw)
kmeans = KMeans(n_clusters=5, n_init=10, random_state=123)
pipeline = make_pipeline(scaler, kmeans)
pipeline.fit(Xw)
```

E. Perform the data analysis and report on the results by doing the following:

E1. Determine the optimal number of clusters in the dataset, and describe the method used to determine this number.

I employed a loop to generate two lists. One contained my inertias while the other contained my silhouette scores.

```
#Identify the best k
inertias = []
avg_silhouette = []

#Fit a cluster model with k=2 . . . 9
for k in range (2, 20):
cluster = KMeans(n_clusters=k, random_state=123, n_init=10)
cluster.fit(Xw_scaled)
inertias.append(cluster.inertia_)
avg_silhouette.append(silhouette_score(Xw_scaled, cluster.labels_))

#print results
for k, inertia, silhouette in zip(range(2,20), inertias, avg_silhouette):
print(f"k={k}: inertia={inertia:.2f}, silhouette={silhouette:.3f}")
```

I used two methods to help me determine the best number of clusters.

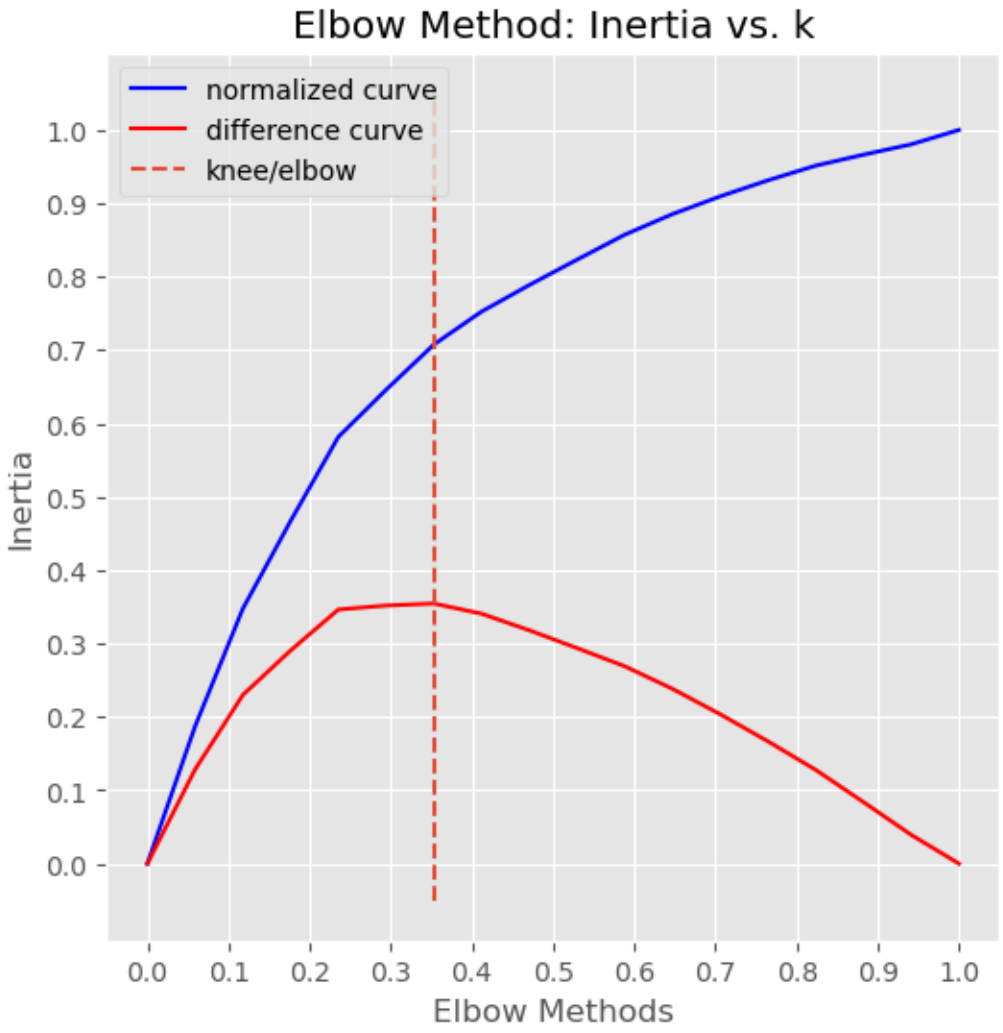
Each method has its pros and cons, offering a slightly different perspective.

Method 1: Elbow Method

One method I used was the elbow method. This method measured inertia, how close each point came to the center of its cluster. Lower inertia meant tighter clusters. Inertia decreased as the number of clusters *k* increased. When examining inertia, the goal is to identify the point at which the "elbow" slows. After looping through the clusters and storing them in the lists inertias and avg_silhouettes, I used the KneeLocator package to detect the bend automatically before visualizing in a line chart. My results showed that the elbow appeared around eight. K = 8 means that the model did not improve much beyond 8 clusters.

The chart produced the following lines:

- **This blue line** is the normalized inertia curve that shows how inertia drops as k increases.
- **This red line** is the difference curve that is used internally by KneeLocator to find the knee.
- **This dashed vertical line** is the point KneeLocator, which marks the elbow method at .35.



Code

```
KNEE = KneeLocator(range(2,20), inertias, S=1.0, curve='convex', direction='decreasing')
print("Knee:", round(KNEE.knee, 3))
print("Elbow:", round(KNEE.elbow, 3))

#charting the normalized data
plt.style.use('ggplot')
KNEE.plot_knee_normalized()
```

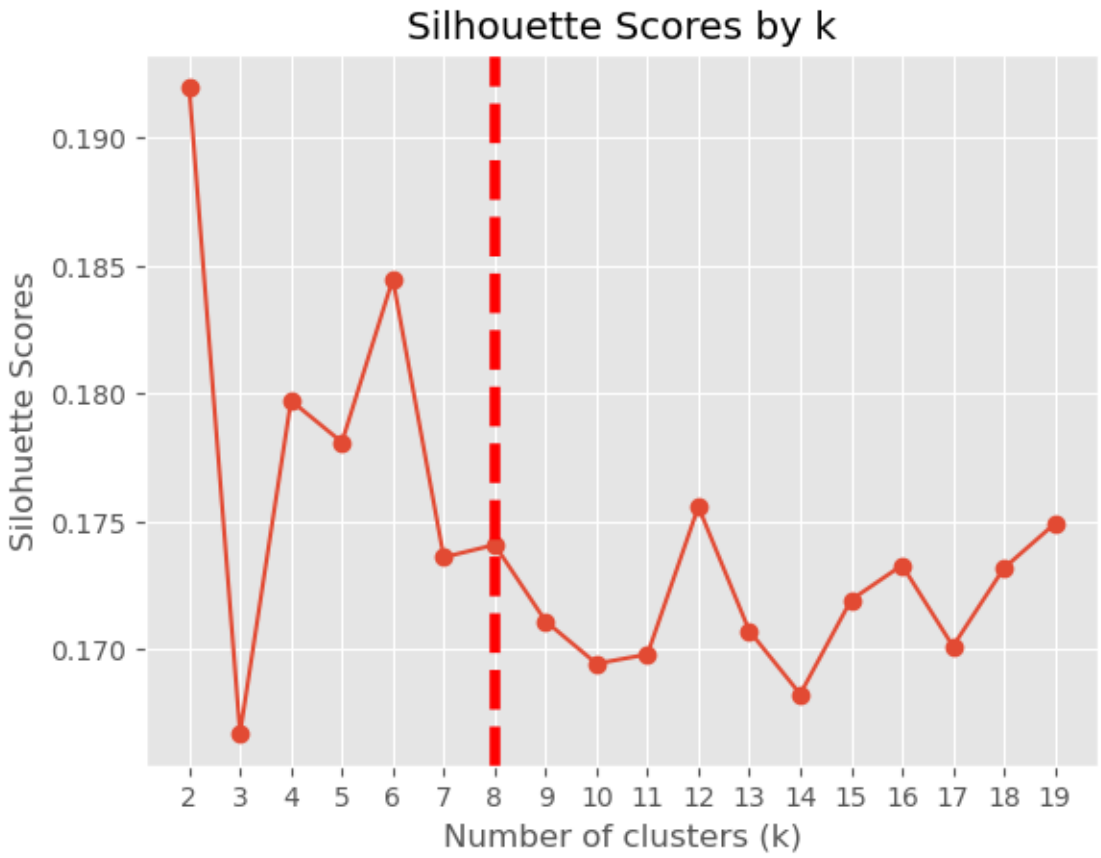
```
plt.ylabel("Inertia")
plt.xlabel("Elbow Methods")
plt.title("Elbow Method: Inertia vs. k")
```

Method 2: Silhouette Score

The second method that I used is the silhouette score. This method checked the quality of the cluster separation. Silhouette scores range from -1 to 1. Clusters that are c**loser to 1** are very distinct, clusters that are around 0 have overlap, while clusters that produce a negative silhouette score may have data in the wrong cluster. Using the same loop, I calculated silhouette scores for k = 2 to k = 19. The highest silhouette score was about 0.19 at k = 2, meaning the data separated the best at 2 clusters.

Code

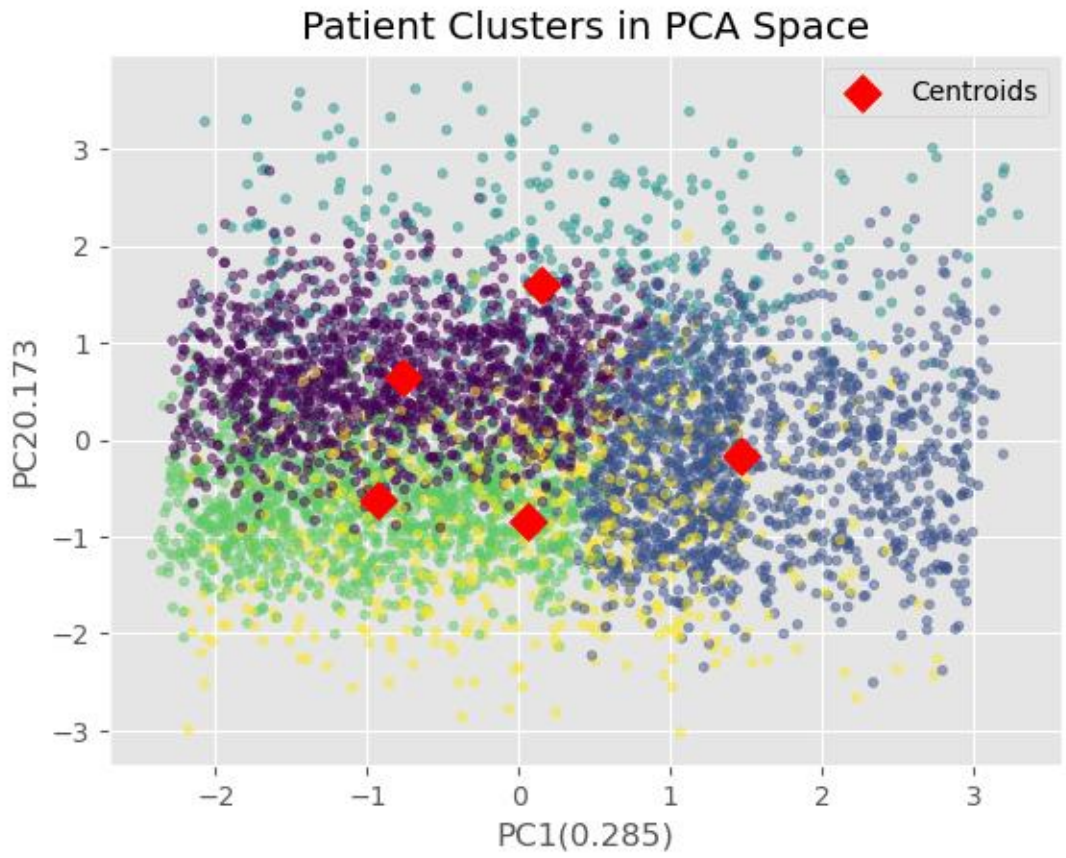
```
print("Knee:", round(KNEE.knee, 3))
print("Elbow:", round(KNEE.elbow, 3))

#charting the silohuette scores
plt.style.use('ggplot')
plt.plot(range(2,20), avg_silhouette, marker="o")
plt.xticks(range(2,20))
plt.xlabel("Number of clusters (k)")
plt.ylabel("Silohuette Scores")
plt.title("Silhouette Scores by k")
#axvline = ax vertical line
plt.axvline(x=KNEE.elbow, color="red", label="axvline - full height", linewidth=4, linestyle="--
")
plt.show()
```



In conclusion, the elbow method's chart suggests 8 clusters. This provides a more detailed subgrouping of patients. However, it risks weaker the group with a smaller silhouette score of .174 than 2 clusters which shows the maximum silhouette score of .19. I chose to go with eight to show stakeholders the various groupings, which show a more meaningful story to provide more enhanced decisions.

F. Summarize your data analysis by doing the following:

F1. Visualize the clusters and explain the quality of the clusters created. Include a screenshot of the cluster visualizations.



I need to reduce my six variables to two, so I used Principal Component Analysis (PCA). This transformed my variables into two dimensions that could be visualized using a scatterplot. After applying PCA(), I conducted the K-Means clustering, and then plotted the results. Each colorful grouping of dots represents a cluster. Within each cluster is a red diamond that is the centroid or the "average center point" of each cluster. The centroid represents one female patient within the cluster.

It is easy to see from the visualization that the clusters overlap significantly, which aligns with the results I obtained when calculating the silhouette scores, which ranged from 0.17 to 0.19. Remember that a silhouette score close to 1 means the clusters are very well-separated, while a score closer to 0 means the clusters overlap. This means that the cluster quality is only moderate. The model identified structure in the data; however, the differences between the groupings are vague at best, resulting in a low silhouette score. Yet, this result makes sense for the hospital's

data. Real patients do not fall into perfectly separated groups, leading to more complexity and nuance within the data.

F2. Discuss the results and implications of your clustering analysis.

Even though silhouette scores fell closer to 0, interesting patterns emerged in the patient groupings. For example, the typical woman patient in Cluster 3 had the following characteristics: Children: 1.70, Age: 77.97, Income: 40173.60, DocVisits: 5.23, TotalCharge: 5271.36, Additional: 25830.29. The average woman of 77.97 with two children and an average income of $40,000 and 5 doctor's visits has 24 percent of the sum of AdditionalCharges. Why is this? Further exploration can lead to answers. Such as maybe this grouping requires more attention to reduce the costs of treatments further down the line.

This shows that , even though the clusters overlap, useful insights can still be gleaned, allowing healthcare organizations to design programs that uniquely serve their diverse patient subgroups.

F3. Discuss one limitation of your data analysis.

Because K-Means Clustering cannot use categorical data, variables related to medical conditions and state locations were excluded. Thus, this dataset focuses on the few continuous variables, which greatly narrows the information gleaned from the patient population. This model can present questions that lead to more modeling using different techniques. Additionally, note that the silhouette score is closer to 0, indicating that the model produced only moderate results.

F4. Recommend a course of action for the real-world organizational situation from part B1 based on the results and implications discussed in part F2.

Based on the cluster profile obtained from the K-Means modeling, I have two main recommendations for a course of action.

Focus on Cluster 3: Older, high-charge group: This grouping has an average age of 78, with a moderate income; however, it has the highest AdditionalCharges of $25,830. The hospital should explore the why behind this, focusing on further research that leads to reducing expenditures among this group. The hospital can ask itself, should it address the high costs this group is producing by implementing chronic care management or training doctors to pay special attention?

Create specialized programs for Cluster 1: This grouping can be described as the middle-aged, high-income earners group. The age is approximately 51, with earnings falling at $104,000. This group has moderate DocVisits with AdditionalCharges of $11,545.96. This group

could benefit from premium preventive care and optional wellness programs to keep the DocVisits from increasing an to provide the hospital with additional revenue streams that can be allocated to other programs.

Works Cited

"Principal Component Analysis (PCA)." *GeeksforGeeks*, https://www.geeksforgeeks.org/data-analysis/principal-component-analysis-pca/. Accessed 29 Sept. 2025.

Kneed Documentation. Read the Docs, https://kneed.readthedocs.io/en/stable/. Accessed 29 Sept. 2025.

Habibimoghaddam, Fatemeh. "Customer Segmentation (K-Means Clustering & PCA)." *Kaggle,* https://www.kaggle.com/code/fhabibimoghaddam/customer-segmentation-k-means-clustering-pca. Accessed 28 Sept. 2025.

Schwab-McCoy, Aimee, et al. *D 603: Machine Learning*. Zyante Inc., 2024. zyBooks, https://learn.zybooks.com/zybook/WGUD603MachineLearningv1/chapter/11/section/1?modal_name=about-zybook. Accessed 28 Sept. 2025.