Part 1

A1.
Research question: How can we segment customer base to identify distinct groups based on their purchasing behavior, utilizing k-means clustering with continuous variables such as income, email (marketing emails sent to customer), and tenure?

A2.
The goal of the data analysis is to use k-means clustering to identify distinct customer segments, understand similar purchasing behaviors and preferences, and develop targeted marketing strategies to enhance customer satisfaction and drive business performance. This analysis can, in turn lead to higher sales, revenue growth, and long-term success.

Part 2

B1.
K-means clustering efficiently analyzes the dataset because it partitions the data into distinct clusters based on similarities in the continuous variables of income, email engagement, and tenure. After standardizing (scaling) the data, k-means begins by randomly selecting k initial cluster centroids, where k represents the number of clusters. The term k can be determined through techniques like the elbow method. Next, each data point (customer) is assigned to the nearest cluster centroid based on a distance metric, usually the Euclidean distance. For this analysis, the distance is calculated using the values of income, email engagement, and tenure. After assigning all data points to clusters, the centroid of each cluster is recalculated as the mean of all data points assigned to that cluster. This step repeats until the centroids no longer change significantly or the predefined number of iterations is reached. The expected outcome of k-means clustering on this dataset is the identification of distinct customer segments with similar purchasing behaviors and preferences. Each cluster will represent a group of customers who exhibit similar patterns in terms of income, email engagement, and tenure. The clusters should be meaningful and interpretable, allowing for targeted marketing strategies tailored to the unique needs and characteristics of each segment. In summary, k-means clustering provides a systematic approach to analyze the dataset by separating customers into meaningful customer segments, enabling the development of targeted marketing strategies to enhance customer satisfaction and drive business performance.

B2.
One assumption of the k-means clustering technique is that clusters have similar sizes. This means that the algorithm assumes that the clusters in the dataset have approximately equal number of data points. If the clusters have significantly different sizes, k-means might not perform robustly, and could produce unsatisfactory clustering results. Therefore, it is essential to assess the fit of k-means to the dataset and consider alternative clustering techniques if this assumption is violated.

B3.

Packages and libraries used in Python:

1. Pandas: Pandas was used to import and manipulate the data for basic exploratory analysis.
2. Missingno: Missingno was used to visualize missing data in my dataset. This was crucial to quickly ensure there was no missing data.

Jasmine Cooper
D212
Task 1: Clustering Techniques
April 21, 2024
Western Governor's University

3. Matplotlib: Matplotlib was used to create various types of plots and visualizations such as bar charts, histograms, scatter plots, and line plots for analysis. The bar chart was used to evaluate missingness, while the line plot allowed me to determine the optimal number of clusters using the elbow plot method. Furthermore, the scatter plots where used to analyze the cluster outcomes and provide valuable insights and recommendations to the telecommunication companies.
4. Scikit-learn: Scikit-learn was used to import the Standard Scaler class from its' preprocessing module. Standard Scaler is used to standardize features by transforming the features of a dataset so that they have a mean of 0 and a standard deviation of 1. Standardization is a critical preprocessing step in the k-means clustering algorithm to ensure features are on the same scale.
5. SciPy: SciPy was used to import and conduct the k-means clustering algorithm to cluster data points into groups based on similarity. This was useful to explore patterns within the data.
6. Seaborn: Similarly, to Matplotlib, Seaborn was used to help create various types of plots and visualizations for analysis. It helped to provide a high-level interface for informative statistical graphics. Seaborn was able to simplify the creation of complex visualizations like the pair plots used for cluster analysis to enhance the interpretability of the results.

In text citations:

("Transforming features for better clusterings", n.d.)
("Basics of k-means clustering", n.d.)

Part 3

C1.
In k-means clustering, the algorithm assigns data points to clusters based on their distances from cluster centroids. Features with larger variances contribute more to these distance calculations because their values span a wider range. Consequently, these features can have a stronger influence on the formation of the clusters. When features have significantly different variances, those with higher variances can dominate the clustering process, leading to clusters that are biased towards those features. This can result in poor clustering results where clusters are formed based primarily on high variance features rather than capturing the overall structure of the data.

To mitigate the influence of variance on the k-means clustering algorithm, standardizing, or scaling the features is required so that the features have similar variances. Standard scaling aims to transform the features of a dataset so that they have a mean of 0 and a standard deviation of 1. This ensures that all features contribute more equally to the distance calculations, leading to more balanced clusters that better represent the underlying structure of the data. The resulting standardized features can be very informative.

C2.

Initial data set variables

Variable	Variable Type
Income	Continuous

Email	Continuous
Tenure	Continuous

C3.

Steps used to prepare the data for analysis:

1. Checking for Missing Values: Although provided a clean dataset, before beginning any analysis, it is still crucial to ensure that the dataset is clean and free from missing values. To prevent bias or inaccuracies in the analysis, I utilized visualization to confirm the dataset contained no missing values to establish a reliable foundation for subsequent analysis.
2. Creating a Separate Dataset with Variables of Interest: Once the dataset had been verified, I proceeded to select the variables that were relevant for my analysis. This step involved creating a new dataset that contained only the variables of interest: Income, Tenure, and Email. By focusing on a subset of variables, I was able to simplify the analysis and improve interpretability.
3. Standardizing the Data: Standardization (or scaling the features) is a preprocessing technique used to rescale the features of the dataset to have a mean of 0 and a standard deviation of 1. This step ensured that all variables were on the same scale, which is important when using k-means clustering. The process involved: fitting the standardization scaler to the data and then transforming the data using the calculated mean and standard deviation.

Overall, by completing these steps, I ensured the data is clean, relevant to my analysis, and appropriately scaled to acquire accurate and insightful data analysis.

Code for checking missing values:

```
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
churn_clean_data = pd.read_csv ('/Users/jasminemoniquecooper/Downloads/churn_clean_pres.csv')
pd.set_option('display.max_columns', None)
churn_clean_data.head(10)

#ensure no missing data
column_order = churn_clean_data.isnull().sum().sort_values().index
msno.bar(churn_clean_data[column_order])
plt.show()
```

Code for creating a separate data frame for variables of interest:

```
#reducing the dataset to only include neccessary variables

selected_columns = ['Income', 'Email', 'Tenure']

churn_clean_kmeans = churn_clean_data[selected_columns]
```

```
print(churn_clean_kmeans)
```

Code for standardizing the data:

```
#standardize the features

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_churn_data = scaler.fit_transform(churn_clean_kmeans)
print(scaled_churn_data)
```

C4. Please see the cleaned and scaled data set attached.

In text citations:
(“Transforming features for better clusterings”, n.d.)

Part 4

D1.

To determine the optimal number of clusters in the dataset, I focused on finding a balance between the number of clusters and inertia, aiming for clusters that are both tight (low inertia) and not too numerous. Here are the steps I followed to determine the optimal number of clusters:

1. Initialization: I initialized an empty list to store inertia values and defined a range of potential cluster numbers to evaluate.
2. Looping through Cluster Numbers: Within a loop, I iterated over each number of clusters in the specified range. For each iteration, I applied the K-Means clustering algorithm to the dataset with the current number of clusters. The algorithm then calculated the cluster centroids and the inertia value, representing the sum of squared distances between each data point and its assigned centroid.
3. Storing Distortion Values: After each iteration, the inertia value was appended to the list of inertia values.
4. Identifying the Elbow Point: Upon inspecting the plot, I identified the elbow point, where the inertia begins to decrease more slowly. This point signifies a balance between the number of clusters and inertia, indicating the optimal number of clusters.

Considering the balance between the number of clusters and inertia, there is a noticeable decrease in inertia from 4 to 5 clusters, and a further decrease from 5 to 6 clusters. However, the decrease in inertia is gradual between 5 to 6 clusters compared to the decrease between 4 and 5 clusters. While 6 clusters might slightly reduce inertia, the marginal gain in tightness might not justify the additional complexity introduced by an extra cluster. Given this observation, 5 clusters appear to be the optimal number of clusters.

D2.

Code used to perform k-means clustering analysis technique:

```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Set the random seed
np.random.seed(21)

# Convert scaled_churn_data to DataFrame with index
scaled_churn_data_df = pd.DataFrame(scaled_churn_data, index=range(len(scaled_churn_data)))

# Fit KMeans clustering algorithm
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(scaled_churn_data_df)

# Get cluster centers
cluster_centers = kmeans.cluster_centers_

# Generate cluster labels
cluster_labels = kmeans.predict(scaled_churn_data_df)

# Calculate inertia
inertia = kmeans.inertia_
print("Inertia:", inertia)

# Calculate silhouette score
silhouette_avg = silhouette_score(scaled_churn_data_df, cluster_labels)
print("Silhouette Score:", silhouette_avg)

# Convert cluster_labels array to DataFrame
cluster_labels_df = pd.DataFrame({'cluster_labels': cluster_labels})

# Concatenate cluster_labels DataFrame with scaled_churn_data_df
clustered_data = pd.concat([scaled_churn_data_df, cluster_labels_df], axis=1)

# Plot clusters with pairplot
pairplot = sns.pairplot(clustered_data, hue='cluster_labels')
```

```
# Customize titles and axis labels
pairplot.fig.suptitle('Pairwise Scatterplots with Cluster Labels', y=1.02)

# Customize individual plot titles
titles = ['Income vs Income', 'Income vs Email', 'Income vs Tenure',
        'Email vs Income', 'Email vs Email', 'Email vs Tenure',
        'Tenure vs Income', 'Tenure vs Email', 'Tenure vs Tenure']

for i, ax in enumerate(pairplot.axes.flat):
   ax.set_title(titles[i])
   ax.set_xlabel('X-Axis Label')
   ax.set_ylabel('Y-Axis Label')

# Adjust layout and spacing
plt.tight_layout(pad=2.0)
```

In text citations:

(“Basics of k-means clustering”, n.d.)
(“How many clusters?”, n.d.)
(“Limitations of k-means clustering”, n.d.)
(“Evaluating a clustering”, n.d.)
(“Silhouette analysis: observation level performance”, n.d.)

Part 5

E1.

Analysis of Cluster Quality

Cluster Quality:

Integrating both inertia and silhouette score evaluations allow for a more comprehensive assessment of cluster quality. While inertia provides insights into the compactness of clusters, silhouette score offers information on the separation between clusters.

Inertia, representing the within-cluster sum of squares, provides insights into the compactness of clusters. A lower inertia value suggests tighter and more compact clusters, indicating a better separation between clusters and more homogeneous groups of data points. In this analysis, the inertia value obtained was 10,469, indicating that the clusters may not be as tightly packed as desired. While the clusters have been identified, there appears to be some overlap or dispersion of data points within clusters, potentially reducing the overall quality of clustering.

The silhouette score offers an overall assessment of clustering quality by measuring the degree of separation between clusters. A silhouette score closer to 1 indicates well defined clusters, with data points well separated from neighboring clusters. Conversely, a score closer to 0 suggests that data points are on the border between clusters, and a score closer to -1 suggests that clusters are poorly separated. In this analysis, the silhouette score obtained was 0.316. While this value indicates moderate clustering quality, it also suggests that there is room for improvement. The clusters are reasonably well

defined with some degree of separation between them. However, there are some instances where the data points lie on the border between clusters and some instances where the data points are poorly matched, potentially impacting the overall quality of clustering.

By considering both metrics, it becomes possible to identify areas where clustering performance can be improved. Since inertia is high and silhouette score is low, it indicates that clusters are spread out and poorly separated.

Cluster Observations:

1. Income vs. Email:
    - Cluster 0: Represents individuals with low income and low email engagement.
    - Cluster 1: Consists of individuals with high income but relatively low engagement.
    - Cluster 2: Displays a diverse range of incomes, spanning from low to high, accompanied by medium to high email engagement.
    - Cluster 3: Similar Cluster 1, encompasses individuals with low income and low email engagement.
    - Cluster 4: Comprises of individuals with high income and relatively low email engagement.
2. Income vs. Tenure:
    - Cluster 0 and 4: Combine into a cluster characterized by high income and low tenure.
    - Cluster 1 and 3: Merge into a single cluster characterized by low income and low tenure.
    - Cluster 2: Exhibits a spectrum of incomes, ranging from low to high, and customer tenure from medium to high.
3. Email vs Tenure:
    - Cluster 0: Consists of individuals with high email engagement, but low tenure.
    - Cluster 1: Encompasses individuals with low email engagement, but high tenure.
    - Cluster 2: Demonstrates a varied distribution with low to high email engagement and tenure.
    - Cluster 3: Compromises of individuals with both low email engagement and tenure.
    - Cluster 4: Encompasses individuals with high email engagement and high tenure.
4. Email vs Income (Inverse of Income vs Email):
    - Mirrors the Income vs Email plot due to reversed axes.
5. Tenure vs Income (Inverse of Income vs Tenure):
    - Mirrors the Income vs Tenure plot due to reversed axes.
6. Tenure vs Email (Inverse of Email vs Tenure):
    - Mirrors the Email vs Tenure plot due to reversed axes.

E2.

The analysis of cluster characteristics across different features such as income, email engagement, and customer tenure provide valuable insights into the underlying patterns within the dataset. Interpretation of clusters reveals distinct groups of individuals with varying income levels, email engagement behaviors, and customer tenure durations.

The analysis revealed several distinct clusters across different features providing insights into the characteristics and behaviors of various customer segments:

1. Income vs Email:
    - Clusters reveal distinct patterns of income and email engagement. Cluster 2 stands out

    as having varied income levels but relatively higher email engagement. This could indicate a segment of the population with diverse economic backgrounds but similar receptiveness to email communications.

2. Income vs Tenure:
    - The merging of Clusters 0 and 4 suggests a group of high-income individuals with relatively short customer tenure, which might indicate a potential churn risk among high earners. Similarly, the consolidation of Cluster 1 and 3 highlights a segment of low-income individuals with short customer tenure.
3. Email vs Tenure:
    - Clusters portray different combinations of email engagement and customer tenure. Notably, Clusters 0 and 2 exhibit high email engagement but short customer tenure, which could indicate a need for targeted efforts to increase retention among active email users.

The higher inertia value, 10,469, that was obtained implies that the clusters are more spread out, with data points scattered over a larger area. This is indicative of overlapping and less clearly defined clusters. On the other hand, the silhouette score of 0.316 suggests moderate clustering quality, indicating reasonably well-defined clusters with some degree of separation. These metrics help to explain the visual observations of the underlying dataset, as to why some clusters are easily interpretable while others, are more difficult to interpret.

The insights from the clustering analysis also have implications for decision making. Tailoring marketing campaigns to specific customer segments based on income levels, email engagement behaviors, and tenure durations can enhance targeting precision and campaign effectiveness. Targeted retention efforts can be designed to address the unique challenges and opportunities presented by each customer segment, such as improving engagement among high income customers with low tenure or increasing loyalty among low-income customers with high tenure.

In conclusion, the clustering analysis provided insights into the structure of the dataset and identified customer segments based on income, email engagement, and tenure. While there are opportunities of improvement in terms of cluster compactness and separation, the insights gained from the analysis can inform strategic decision making and drive business growth. By leveraging these insights, telecommunication companies can enhance customer satisfaction, optimize resource allocation, and achieve sustainable competitive advantage in the market.

E3.

One limitation of my data analysis is the reliance on the k-means clustering method, which may not always be the most appropriate choice for all datasets. While k-means clustering can be effective, its'

assumptions regarding cluster size, may not always hold true in real-world data. Furthermore, in comparison to hierarchical clustering, k-means clustering can result in non-intuitive clusters. Also, the interpretation of clusters is subjective and may vary depending on the context and domain knowledge of the analyst. Furthermore, the effectiveness of clustering algorithms can be influenced by the choice of input variables. Adding more numerical input variables, for this clustering method could have yielded different results, and changed the perception of the results and implications. To mitigate these limitations, future analyses could explore alternative clustering algorithms, and incorporate additional variables to improve accuracy and/or interpretability of the clustering outcomes. Additionally, collaborating with domain experts can provide valuable insights into the nuances of the data and help ensure that the clustering results are meaningful and actionable.

E4.

Based on the results and implications, it's recommended to develop a segmentation strategy tailored to the specific needs and behaviors of distinct customer segments. Using insights from the clustering analysis, telecommunication companies should foster collaboration between marketing, sales, and customer service teams. This collaboration aims to develop targeted marketing efforts and customer retention initiatives that resonate with the characteristics of each cluster. For example, for clusters exhibiting high email engagement but short customer tenure, implement retention-focused email campaigns to increase customer loyalty and reduce churn rates. Additionally, implementing targeted customer retention initiatives for high-income individuals with short customer tenure who may be at risk of churn, could effectively boost overall business and revenue growth. These initiatives could include personalized incentives, loyalty programs, and proactive outreach efforts to enhance the overall customer experience and increase retention rates.

After implementing recommendations, continuous monitoring key performance indicators like customer retention, email engagement metrics, and revenue generation can evaluate the effectiveness of the segmentation strategy and marketing initiatives. Using the data gathered from monitoring should help to refine and optimize the segmentation approach over time, ensuring alignment with evolving customer preferences and market dynamics. By implementing this recommendation and fostering collaboration and feedback exchange, the organization can leverage insights gained from the clustering analysis. This will enable the development of more targeted and effective marketing strategies, enhance customer retention efforts, and improve overall business performance and competitiveness in the market.

In text citations:
("Limitations of k-means clustering", n.d.)
("Evaluating a clustering", n.d.)
("Silhouette analysis: observation level performance", n.d.)

Part 6

D. Citations for code:

DataCamp. (n.d.). Transforming features for better clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/clustering-for-dataset-exploration?ex=8

DataCamp. (n.d.). Basics of k-means clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-python/k-means-clustering-3?ex=1

DataCamp. (n.d.). How many clusters? [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-python/k-means-clustering-3?ex=1

DataCamp. (n.d.). Evaluating a clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/clustering-for-dataset-exploration?ex=5

DataCamp. (n.d.). Silhouette analysis: observation level performance [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-r/k-means-clustering?ex=8

E. Citations for content:

DataCamp. (n.d.). Transforming features for better clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/clustering-for-dataset-exploration?ex=8

DataCamp. (n.d.). Basics of k-means clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-python/k-means-clustering-3?ex=1

DataCamp. (n.d.). Limitations of k-means clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-python/k-means-clustering-3?ex=7

DataCamp. (n.d.). Evaluating a clustering [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/clustering-for-dataset-exploration?ex=5

DataCamp. (n.d.). Silhouette analysis: observation level performance [Video file]. Retrieved from https://campus.datacamp.com/courses/cluster-analysis-in-r/k-means-clustering?ex=8