Part I: Research Question

A1. Proposal of Question

Businesses are always trying to understand their customer's characteristics so that they can make informed decisions on business goals. Therefore, an important question to ask is, can we sift through the given dataset using Kmeans Clustering Analysis to better understand our customers' characteristics and understand the subgroups that exist within those data so that we can improve our business model?

A2. Defined Goals

As we know, it costs ten times more to acquire new customers than retain old ones and the industry's annual churn rate hovers around twenty-five percent. Though the goal of this analysis is to not predict Churn, we need to be able to understand our customers so that we can make an informed decision on our business model. Therefore, the goal of this analysis is to understand the features that make up our customers and understand the subgroups within those features and see if we can find any valuable insights within them.

Part II: Technique Justification

B1. Explanation of Clustering Technique

Clustering is one of the most common exploratory data analysis techniques used to get an intuition about the structure of the data (Dabbura, 2022). For our analysis, we chose to use the Kmeans clustering algorithm for its simplicity. Kmeans clustering algorithm is a popular centroid-based clustering algorithm, that divides the data points of the entire population into k clusters each having an almost equal number of data points (Anwla, 2021). The idea is to find k-centroid points and identify every point that belongs to either of the k-sets having the minimum Euclidean distance (Anwla, 2021).

The Kmeans Algorithm Analysis will analyze this dataset by partitioning the dataset into "K" pre-defined distinct groups where each data point belongs to only one group (Dabbura, 2022). It tries to make the intra-cluster data points as similar as possible while keeping the cluster as different as possible (Nagar, 2021). The expected outcome is that we will have less variation within the cluster, and more

similar data within the same cluster. In another word, it will allow us to see patterns in the data and understand the subgroups within the features.

B2. Summary of Technique Assumptions

One assumption of Kmeans clustering is that clusters have the same variance (Nagar, 2021). A quick summary of this would be that all points that belong to a cluster, are grouped based on their Euclidean distance and that their variance is equal (Nagar, 2021). By definition of the Euclidean distance, all variables used for clustering must be continuous (*K-Means Cluster Analysis*, 2022).

B3. Packages or Library List

For data frame manipulation and processing, we import Pandas, Numpy, and Scipy. These allow us to work with the dataset and do the necessary cleaning, data preprocessing, and handling for the analysis. For our data visualization, we use Matplotlib and Seaborn to do exploratory analysis to find outliers and see how our data is dispersed. It also allows us to create a Scree plot for the Elbow method of selecting our optimal value of K for our Kmeans clustering analysis. Finally, we use Sklearn to perform the Kmeans analysis.

Part III: Data Preparation

C1. Data Preprocessing

All data were first cleaned by dropping unnecessary columns, filling in null values, and removing outliers. Since Kmeans clustering works with continuous variables and to calculate the Euclidean distance, we need to ensure that all our variables are continuous. Therefore, we drop all categorical and binary columns. We then scaled, standardized, and fit the data using sklearn to prepare it for our analysis.

C2. Dataset Variables

All variables were standardized for future analysis. For this specific analysis, we used Age and Income for our customer demographics. We then selected MonhtlyCharge, Bandwidth_GB_Year, and Sum_services to understand how our customers use our services.

C3. Steps for Analysis

The steps to prepare the data for Kmeans Clustering Analysis are:

1. Import the necessary libraries into Jupyter Notebook.
2. Load and clean the dataset usings Pandas, Numpy, and Scipy. In this step, we fill in null values, drop data with too many missing values, and recode categorical columns into continuous columns for future analysis.
3. Perform exploratory data analysis such as plot histogram, boxplot, Z_score, and scatterplots to identify potential outliers.
4. Export the cleaned dataset.
5. Data preprocessing using sklearn to create dummy variables out of categorical variables. We then standardize and fit the data for this analysis.

Part IV: Analysis

D1. Output and Intermediate Calculations

In order to perform our Kmean clustering analysis, we first performed a Scree Plot to identify the elbow method needed in the selection of our "K" value for our number of clusters. We then performed a Kmean clustering analysis by declaring a variable for the Kmeans analysis in Python. We then fit it with our variable selection. Our variables were Age and Income, MonthlyCharge on Bandwidth_GB_Year, and MonthlyCharge on Sum_services. After that, we plot the clusters and then plotted additional clusters for comparison. A screenshot of the intermediate calculation was provided below.

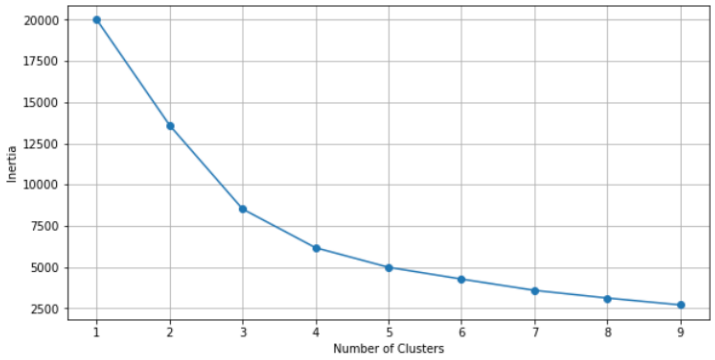
In [81]:
```
#Function for Performing Elbow Method
def optimise_k_means(data, max_k):
    means = []
    inertias = []

    for k in range(1, max_k):
        kmeans = KMeans(n_clusters=k)
        kmeans.fit(data)

        means.append(k)
        inertias.append(kmeans.inertia_)

    #Generate the elbow plot
    fig = plt.subplots(figsize = (10,5))
    plt.plot(means,inertias, 'o-')
    plt.xlabel('Number of Clusters')
    plt.ylabel('Inertia')
    plt.grid(True)
    plt.show()
```

In [82]:
```
df_KM_1 = df_final.copy()
optimise_k_means(df_KM_1[['Age_K','Income_K']], 10)
```

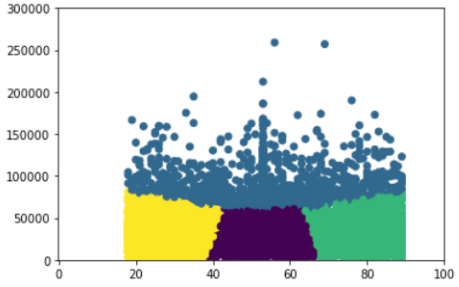


In [84]:
```
km = KMeans(n_clusters=4)
```

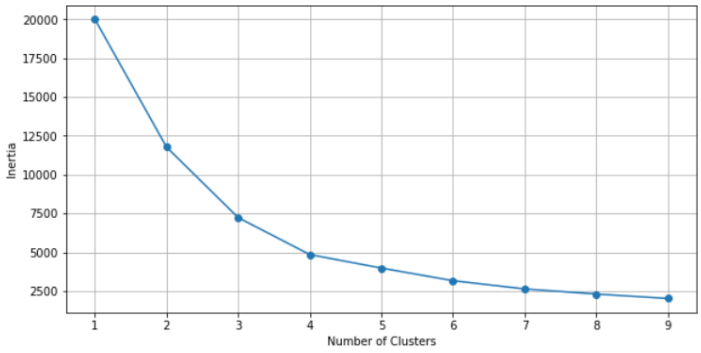
In [85]:
```
km_fit = km.fit(df_KM_1[['Age_K','Income_K']])
```

In [86]:
```
df_KM_1['kmeans_4'] = km_fit.labels_
```

In [89]:
```
plt.scatter(x=df_KM_1['Age'], y=df_KM_1['Income'], c=df_KM_1['kmeans_4'])
plt.xlim(-0.1, 100)
plt.ylim(0, 300000)
plt.show()
```



In [93]:
```
df_KM_2 = df_final.copy()
optimise_k_means(df_KM_2[['MonthlyCharge_K','Band_GB_Year_K']], 10)
```

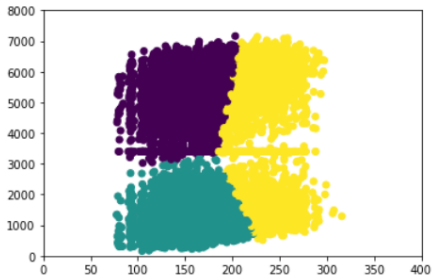


In [94]:
```
km_2 = KMeans(n_clusters=3)
km_fit_2 = km_2.fit(df_KM_2[['MonthlyCharge_K','Band_GB_Year_K']])
df_KM_2['kmeans_3'] = km_fit_2.labels_
df_KM_2
```

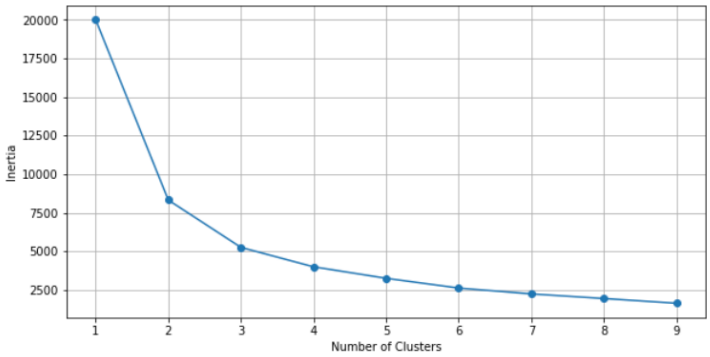
1	1.0	27.0	21704.770000	12.014541	1	242.948015	800.982766	3	4
2	4.0	50.0	39936.762226	10.245616	1	159.440398	2054.706961	4	4
3	1.0	48.0	18925.230000	15.206193	0	120.249493	2164.579412	4	4
4	0.0	83.0	40074.190000	8.960316	1	150.761216	271.493436	4	4
...	...	...	...	...	...	...	...	...	...
9995	3.0	53.0	55723.740000	9.265392	0	159.828800	6511.253000	3	2
9996	4.0	48.0	39936.762226	8.115849	0	208.856400	5695.952000	4	5
9997	1.0	53.0	39936.762226	4.837696	0	168.220900	4159.306000	4	4
9998	1.0	39.0	16667.580000	12.076460	0	252.628600	6468.457000	4	4
9999	1.0	28.0	39936.762226	12.641760	0	218.371000	5857.586000	2	2

10000 rows × 49 columns

In [95]:
```
plt.scatter(x=df_KM_2['MonthlyCharge'], y=df_KM_2['Bandwidth_GB_Year'], c=df_KM_2['kmeans_3'])
plt.xlim(-0.1,400)
plt.ylim(0, 8000)
plt.show()
```



In [97]:
```
df_KM_3 = df_final.copy()
optimise_k_means(df_KM_3[['MonthlyCharge_K','Sum_service_K']], 10)
```



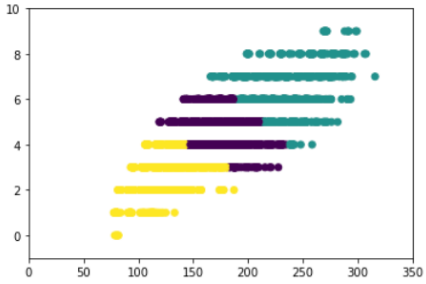
In [98]:
```
km_3 = KMeans(n_clusters=3)
km_fit_3 = km_3.fit(df_KM_3[['MonthlyCharge_K','Sum_service_K']])
df_KM_3['kmeans_3'] = km_fit_3.labels_
df_KM_3
```

Out[98]:

	Children	Age	Income	Outage_sec_perweek	Yearly_equip_failure	MonthlyCharge	Bandwidth_GB_Year	Timely_Response	Timely_Fixes	Timely_Rep
0	1.0	68.0	28561.990000	6.972566	1	171.449762	904.536110	5	5	
1	1.0	27.0	21704.770000	12.014541	1	242.948015	800.982766	3	4	
2	4.0	50.0	39936.762226	10.245616	1	159.440398	2054.706961	4	4	
3	1.0	48.0	18925.230000	15.206193	0	120.249493	2164.579412	4	4	
4	0.0	83.0	40074.190000	8.960316	1	150.761216	271.493436	4	4	
...	...	...	...	...	...	...	...	...	...	
9995	3.0	53.0	55723.740000	9.265392	0	159.828800	6511.253000	3	2	
9996	4.0	48.0	39936.762226	8.115849	0	208.856400	5695.952000	4	5	
9997	1.0	53.0	39936.762226	4.837696	0	168.220900	4159.306000	4	4	
9998	1.0	39.0	16667.580000	12.076460	0	252.628600	6468.457000	4	4	
9999	1.0	28.0	39936.762226	12.641760	0	218.371000	5857.586000	2	2	

10000 rows × 49 columns

In [99]:
```
plt.scatter(x=df_KM_3['MonthlyCharge'], y=df_KM_3['Sum_services'], c=df_KM_3['kmeans_3'])
plt.xlim(-0.1,350)
plt.ylim(-1, 10)
plt.show()
```



Part V: Data Summary and Analysis

E1. Accuracy of Clustering Technique

Since Kmeans clustering is an unsupervised clustering algorithm, there is no single whole answer for evaluating the accuracy of the analysis. What we can do is ensure that we are performing the analysis optimally by plotting a Scree plot and using the Elbow method to ensure that we are using the best "K" value for the number of clusters (Khandelwal, 2021). We performed that on all 3 of our analyses and ensured our "K" value selection is where the Scree plot bends using the Elbow method.

E2. Results and Implications

In reviewing our initial clustering visualizations for customer demographics, age and income, we did not see any outstanding differences in the four clusters that were selected based on our Elbow Method. It looked like income was spread normally throughout all age groups and that there was a group that made more income across all age groups which is expected. We even performed an additional analysis of different clusters and identified no major differences between the different numbers of clusters. The implications of this analysis could mean that we are already serving all age groups across all income levels.

On a customer's monthly charge and their bandwidth usage, we could not identify any significant differences between the groups. Groups that paid more did not use more bandwidth but instead resided on both spectrums of bandwidth usage. There were two groups in the visualization that had lower monthly charges but there was no indication of whether monthly charges had any effect on bandwidth usage. Additional clustering did not achieve any significant differences. This analysis implies that there is no defined group when it comes to who we can charge more monthly, based on their bandwidth usage.

Finally, an additional clustering visualization was performed to look at customers' monthly charges and their number of services. In it, we do see that each cluster performed as expected in that

the more services a customer signs up for, the more they pay monthly. We then performed additional clustering and that too didn't indicate any significant difference. This analysis implies that the more services we can get them to sign up for, the more we can charge them monthly which is expected.

E3. Limitations

One of the limitations of this analysis is that the dataset had missing information and because we had to impute those missing information with the mean from the rest of the dataset, it may have possibly skewed the dataset and introduced errors into our analysis. Data should always be collected as thoroughly as possible and because we do not have additional access to how the data was collected, we could make an informed enough decision on how missing data should have been handled. Also, this dataset may have been generated which could have caused our analysis to not perform optimally, leading to us being unable to solve real business issues.

E4. Course of Action

Since we could not identify any significant differences in our Kmeans clustering analysis, reiterations of additional variables using this clustering analysis may be needed. Additionally, we can try other clustering techniques such as hierarchical clustering, and see if that may help us find any significant differences in our dataset. Finally, to find a solution to mitigate customer churn, we must be persistent to ensure that our business operates optimally and is performing to the best of our customer's needs.

Part VI. Demonstration

F. Panopto Recording

G. Sources for Third-Party Code

K-Means Clustering Algorithm with Python Tutorial. (2021, November 17). [Video]. YouTube. https://www.youtube.com/watch?v=iNlZ3IU5Ffw

K-Means Clustering in Python - Machine Learning From Scratch 12 - Python Tutorial. (2019, December 4). [Video]. YouTube. https://www.youtube.com/watch?v=vtuH4VRq1AU

Machine Learning Tutorial Python - 13: K Means Clustering Algorithm. (2019, February 4). [Video]. YouTube. https://www.youtube.com/watch?v=EItlUEPCIzM

H. Sources

Anwla, P. K. (2021, December 1). *K-Means*. TowardsMachineLearning. https://towardsmachinelearning.org/k-means/

Bandgar, S. (2022, January 6). *K-MEANS CLUSTERING USING ELBOW METHOD - MLearning.ai*. Medium. https://medium.com/mlearning-ai/k-means-clustering-using-elbow-method-208b23c78150

Khandelwal, R. (2021, December 22). *Evaluating goodness of clustering for unsupervised learning case*. Medium. https://towardsdatascience.com/evaluating-goodness-of-clustering-for-unsupervised-learning-case-ccebcfd1d4f1

K-Means Cluster Analysis. (2022). Columbia Public Health. https://www.publichealth.columbia.edu/research/population-health-methods/k-means-cluster-analysis#:%7E:text=Euclidean%20distances%20can%20be%20extended%20to%20n-dimensions%20with,determine%20clustering%20using%20k-means%20must%20be%20continuous.%20Procedure

Nagar, A. (2021, December 13). *K-means Clustering — Everything you need to know - Analytics Vidhya*. Medium. https://medium.com/analytics-vidhya/k-means-clustering-everything-you-need-to-know-175dd01766d5#f6a0