Task 1: Clustering Techniques

Part 1: Research Question

1. Do churning customers have characteristics in common?

2. The goal of the data analysis is to use k-means to determine the clusters in the data. Then, these clusters will be analyzed to determine if they align with the churn feature.

Part 2: Technique Justification

1. K-means is an unsupervised method. The method runs with no predetermined target feature. K-means starts by selecting the number of centroids (k) for the data set. The data is fitted to the model by measuring its distance to the different centroids. The point is assigned to the nearest centroid. After all the data points are distributed, the average distance from the points to the centroids is calculated. The centroids are relocated to new locations based on these averages. This process is repeated until there is no more movement. (K-Means Clustering, n.d.) The expected outcome is the clusters align with the churn feature.

2. One assumption of k-means is that the clusters have the same variance. This assumption leads to a few consequences. All the clusters are the same size. The amount of space covered by the clusters is the same. The shape of the clusters is spherical. All points in the clusters are equidistance from the centroid. The clusters have a similar number of points assigned to them. (Winn, n.d.)

3. This analysis used seven packages or libraries. Pandas and NumPy were used to import the data to the Juypter notebook and create and manipulate the DataFrame. Visualizations were created with Matplotlib. Several packages were imported from Sklearn. KMeans was used to model the data. StandardScaler was used to set the variance of each feature to 1. Metrics was used to analyze how well the model performed.

Part 3: Data Preparation

1. One preprocessing goal is to remove the amount of influence that a feature has because of its variance. K-means is based on distance, and each feature has a different range. The features with a more extensive range like population (0 – 111850) have more influence over the clustering than a feature with a smaller range like children (0 – 10). All features are rescaled to have a mean of 0 and a variance of 1, creating a balance of influence.

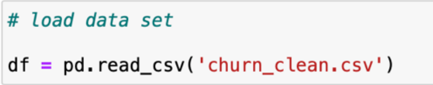
2. The features used for this analysis are as follows:

Continuous: Population, Children, Age, Income, Outage_sec_perweek, Email, Contracts,
Yearly_equip_failure, Tenure, MonthlyCharge, Bandwidth_BG_Year

K-means does not support using categorical features. K-means measures the distance a point is from a centroid. Categorical features are discrete and do not have a meaningful numeric value. (Kumar, 2021)

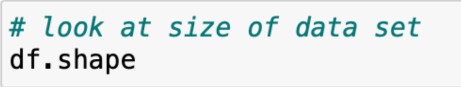
3. Steps used to clean and prepare data:

- Import dataset to juypter notebook

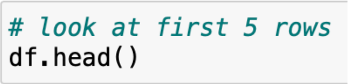


```
# load data set

df = pd.read_csv('churn_clean.csv')
```

- View the size and first five rows

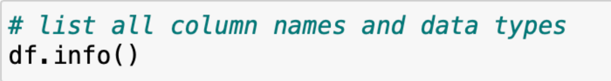


```
# look at size of data set
df.shape
```



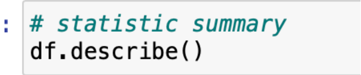
```
# look at first 5 rows
df.head()
```

- List the column names and data types



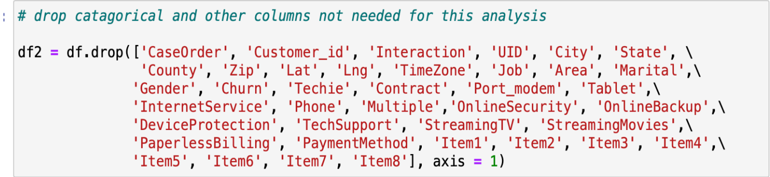
```
# list all column names and data types
df.info()
```

- View summary statistics



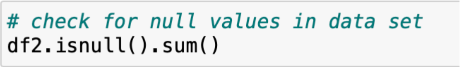
```
: # statistic summary
df.describe()
```

- Drop unneeded features



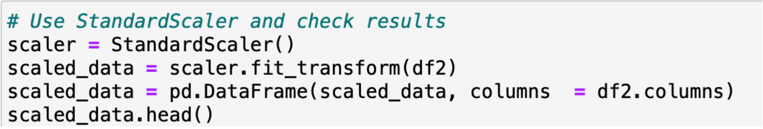
```
: # drop catagorical and other columns not needed for this analysis

df2 = df.drop(['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', \
               'County', 'Zip', 'Lat', 'Lng', 'TimeZone', 'Job', 'Area', 'Marital',\
              'Gender', 'Churn', 'Techie', 'Contract', 'Port_modem', 'Tablet',\
              'InternetService', 'Phone', 'Multiple','OnlineSecurity', 'OnlineBackup',\
              'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',\
              'PaperlessBilling', 'PaymentMethod', 'Item1', 'Item2', 'Item3', 'Item4',\
              'Item5', 'Item6', 'Item7', 'Item8'], axis = 1)
```

- Check for nulls



```
# check for null values in data set
df2.isnull().sum()
```

- Use StandardScaler on data

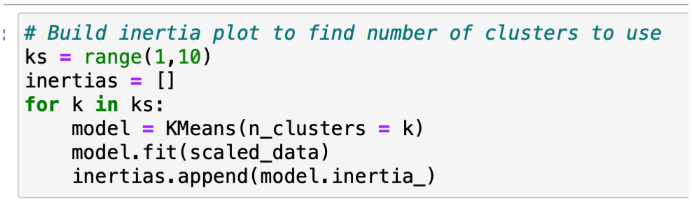


```
# Use StandardScaler and check results
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df2)
scaled_data = pd.DataFrame(scaled_data, columns  = df2.columns)
scaled_data.head()
```

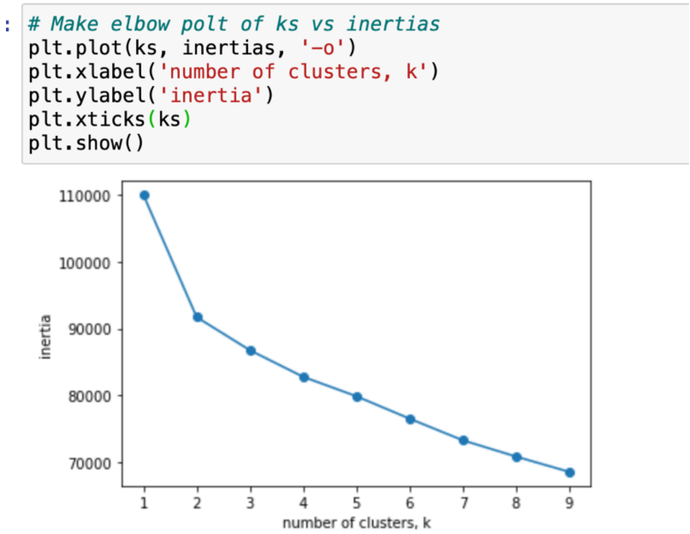
4. Copy of clean data set provided as Clustering_Prepared_Data.xlsx

Part 4: Analysis

1. The first step in using K-means for analysis is to choose the number of clusters or k to use. An inertia plot was built to find the optimum number of clusters. Inertia measures how spread out the clusters are. Lower inertia is the goal. An elbow plot was constructed to see the number of clusters to use more clearly. The range of k's tested was 1 to 10. Two clusters were chosen because the inertia begins to slow at 2 clusters.

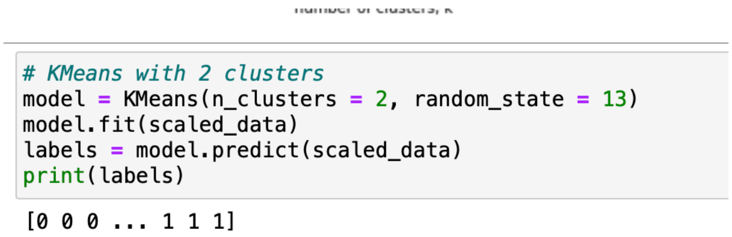


```
# Build inertia plot to find number of clusters to use
ks = range(1,10)
inertias = []
for k in ks:
    model = KMeans(n_clusters = k)
    model.fit(scaled_data)
    inertias.append(model.inertia_)
```



```
# Make elbow polt of ks vs inertias
plt.plot(ks, inertias, '-o')
plt.xlabel('number of clusters, k')
plt.ylabel('inertia')
plt.xticks(ks)
plt.show()
```

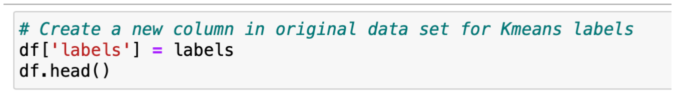
The data was then fitted to the model, creating labels for the clusters.



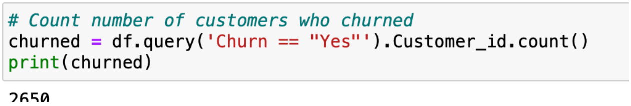
```
# KMeans with 2 clusters
model = KMeans(n_clusters = 2, random_state = 13)
model.fit(scaled_data)
labels = model.predict(scaled_data)
print(labels)
```

```
[0 0 0 ... 1 1 1]
```

How these labels relate to the original data set is unknown. Further analysis was required to see if the labels relate to churn. A new column was created in the original data set for the model labels. A query shows that 2650 customers churned. There are also 2366 who churned and are in cluster 1. The closeness of these numbers indicates a strong relationship between the churn and the labels. After some calculations, it is shown that the overall accuracy score is 70.81%. The churn prediction accuracy was 89.28%.

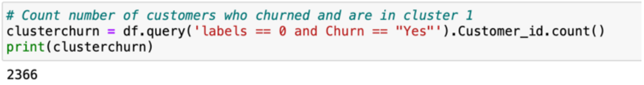


```
# Create a new column in original data set for Kmeans labels
df['labels'] = labels
df.head()
```



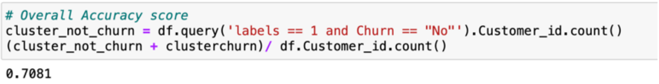
```
# Count number of customers who churned
churned = df.query('Churn == "Yes"').Customer_id.count()
print(churned)
```

```
2650
```



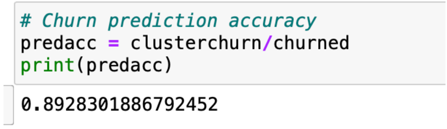
```
# Count number of customers who churned and are in cluster 1
clusterchurn = df.query('labels == 0 and Churn == "Yes"').Customer_id.count()
print(clusterchurn)
```

```
2366
```



```
# Overall Accuracy score
cluster_not_churn = df.query('labels == 1 and Churn == "No"').Customer_id.count()
(cluster_not_churn + clusterchurn)/ df.Customer_id.count()
```

```
0.7081
```



```
# Churn prediction accuracy
predacc = clusterchurn/churned
print(predacc)
```

```
0.8928301886792452
```

Part 5: Data Summary and Implications

1. The K-means analysis was able to detect 2 clusters. These clusters aligned closely to the churn feature. The overall accuracy is 70.81%. The focus of this analysis was on churning customers. The churn accuracy is 89.23%.

2. The clusters show that there are characteristics that churning customers have in common. Further analysis is required to determine what these commonalities are.

3. One limitation of K-means is that it does not include the use of categorical data. In this data set, twenty features were dropped because they were categorical. There is a lot of information lost.

4. It is recommended that, after commonalities are found, the marketing and retention teams use them to target the customers most likely to churn based on these features. For example, if tenure is an indicator of churn, offer incentives to sign a contract.

Works Cited

K-Means Clustering. (n.d.). Retrieved 2021, from Learn by Marketing: https://www.learnbymarketing.com/methods/k-means-clustering/

Kumar, S. (2021, May). *Clustering Algorithm for data with mixed Categorical and Numerical features*. Retrieved December 2021, from Towards Data Science: https://towardsdatascience.com/clustering-algorithm-for-data-with-mixed-categorical-and-numerical-features-d4e3a48066a0

Winn, J. (n.d.). *How To Read A Model*. Retrieved Dec 2021, from Model-Based Machine Learning: https://www.mbmlbook.com/ModelAnalysis_K-means_Clustering.html