```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/customer_churn_dataset-testing-master.csv')
```

```
df.head()
```

	CustomerID	Age	Gender	Tenure	Usage Frequency	Support Calls	Payment Delay	Subscription Type	Contract Length	Total Spend
0	1	22	Female	25	14	4	27	Basic	Monthly	598
1	2	41	Female	28	28	7	13	Standard	Monthly	584
2	3	47	Male	27	10	2	29	Premium	Annual	757
3	4	35	Male	9	12	5	17	Premium	Quarterly	232
4	5	53	Female	58	24	9	2	Standard	Annual	533

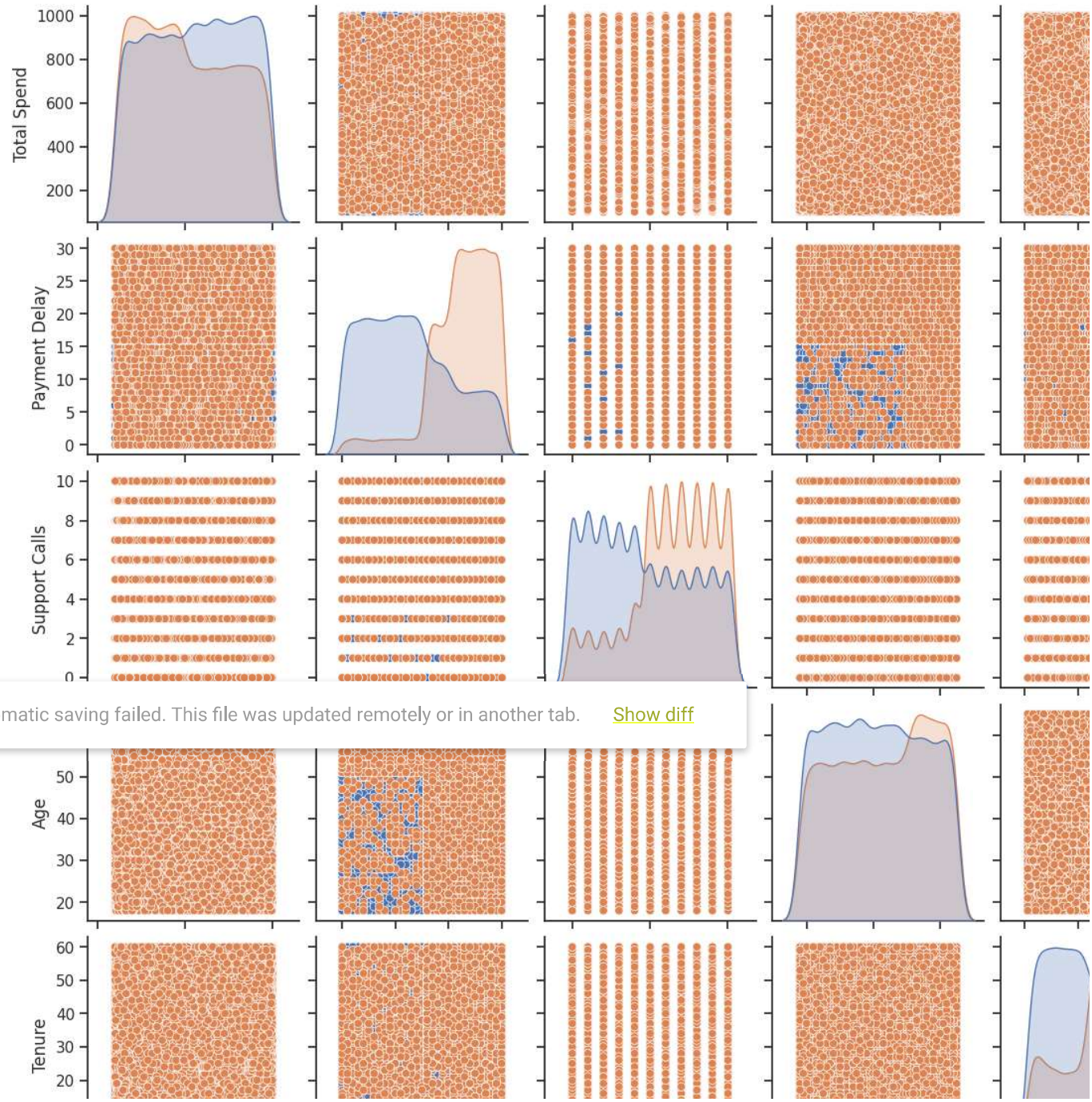

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64374 entries, 0 to 64373
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   CustomerID         64374 non-null  int64
 1   Age                64374 non-null  int64
 2   Gender             64374 non-null  object
 3   Tenure             64374 non-null  int64
 4   Usage Frequency    64374 non-null  int64
 5   Support Calls      64374 non-null  int64
 6   Payment Delay      64374 non-null  int64
 7   Subscription Type  64374 non-null  object
 8   Contract Length    64374 non-null  object
 9   Total Spend        64374 non-null  int64
 10  Last Interaction   64374 non-null  int64
 11  Churn              64374 non-null  int64
dtypes: int64(9), object(3)
memory usage: 5.9+ MB
```

```
attribute_selected = ['Total Spend', 'Payment Delay','Support Calls','Age', 'Tenure','Churn']
df_selected = df[attribute_selected].copy()

sns.set(style="ticks")
sns.pairplot(df_selected, hue="Churn")

plt.show()
```

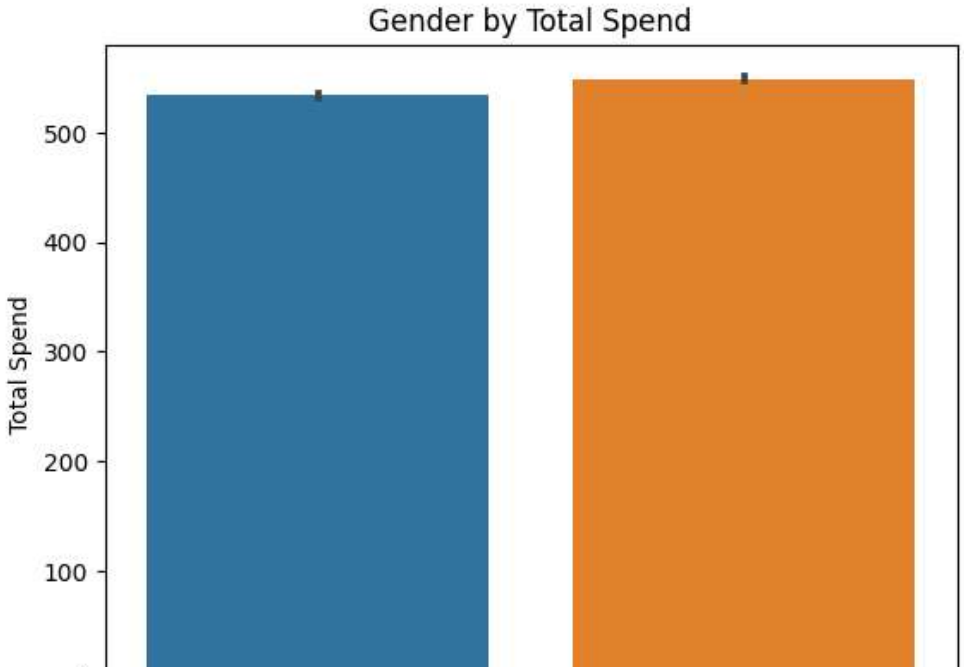


Double-click (or enter) to edit

```
sns.barplot(data=df, x='Gender', y='Total Spend')

# Add labels and title
plt.xlabel('Gender')
plt.ylabel('Total Spend')
plt.title('Gender by Total Spend')

# Show the plot
plt.show()
```



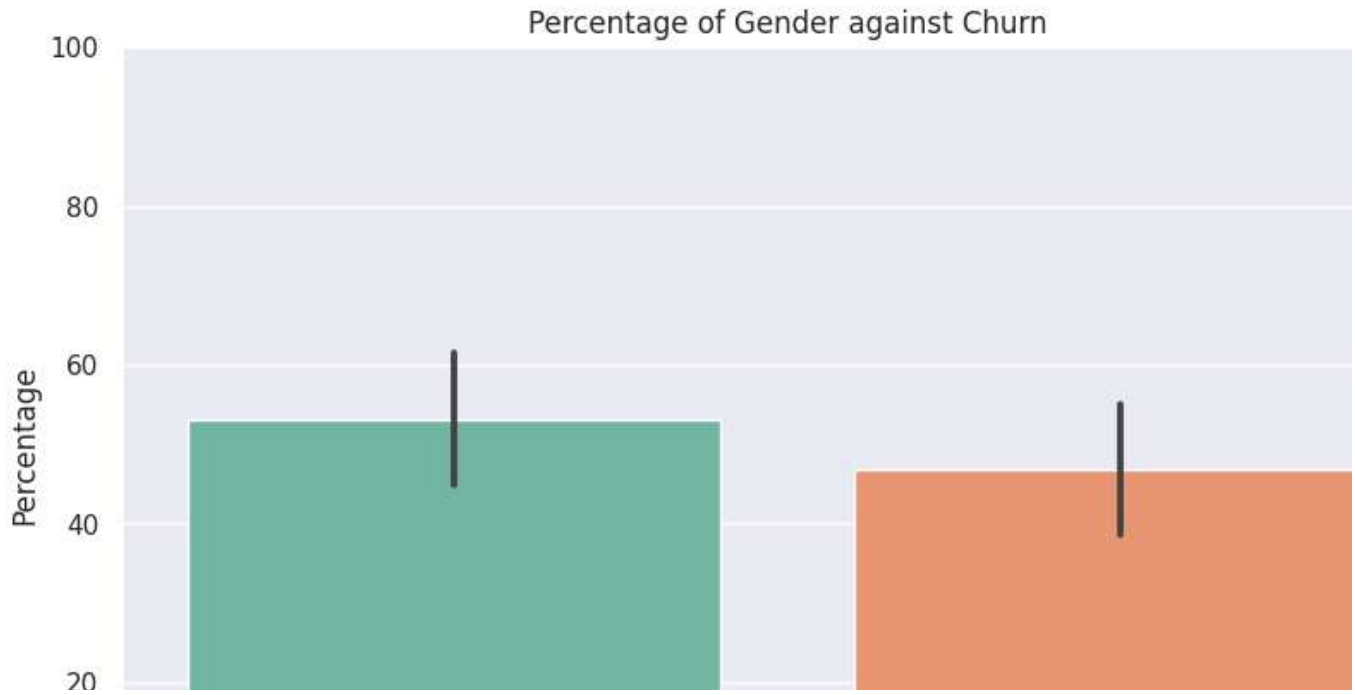
Cross tab between gender and customer churn

```
cross_t=pd.crosstab(df['Gender'], df['Churn'])
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Churn	0	1
Gender		
Female	44.950950	55.049050
Male	61.420339	38.579661

```
sns.set(style="darkgrid")
plt.figure(figsize=(10, 6))
by = sns.barplot(data=cross_tPercent, palette="Set2")
plt.title("Percentage of Gender against Churn")
plt.xlabel("Churn: 0 Yes 1 No")
plt.ylabel("Percentage")
plt.ylim(0, 100)  # Set y-axis limit to 0-100 for percentages
plt.show()
```




```
cross_t=pd.crosstab(df['Subscription Type'], df['Churn'])

cross_tPercent = cross_t.apply(lambda x: x / x.sum() * 100, axis=1)
cross_t
cross_tPercent
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

Subscription Type		
Basic	51.722530	48.277470
Premium	53.503571	46.496429
Standard	52.669519	47.330481

```
cross_t=pd.crosstab(df['Subscription Type'], df['Contract Length'])

cross_tPercent = cross_t.apply(lambda x: x / x.sum() * 100, axis=1)
cross_t
cross_tPercent
```

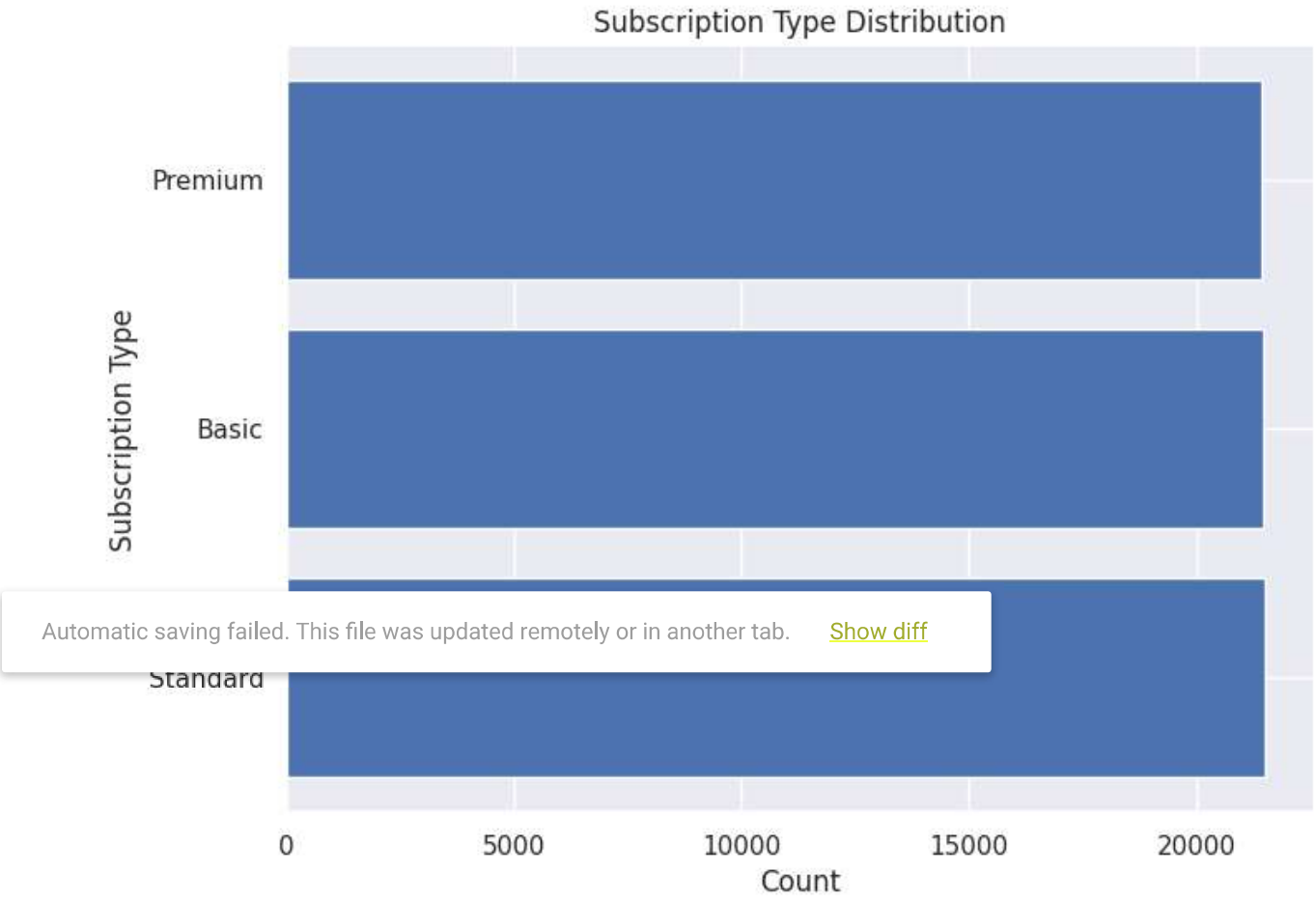
Contract Length	Annual	Monthly	Quarterly
Subscription Type			
Basic	33.322456	34.371358	32.306186
Premium	33.028337	34.587554	32.384109
Standard	33.424798	34.173565	32.401637

```
Subscription = df['Subscription Type'].value_counts()

# Create a bar plot
plt.figure(figsize=(8, 6))
plt.barh(Subscription.index, Subscription)
```

```
# Add labels and title
plt.xlabel('Count')
plt.ylabel('Subscription Type')
plt.title('Subscription Type Distribution')

# Show the plot
plt.show()
```



The analysis proceeds to k-means clustering

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64374 entries, 0 to 64373
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   CustomerID         64374 non-null  int64
 1   Age                64374 non-null  int64
 2   Gender             64374 non-null  object
 3   Tenure             64374 non-null  int64
 4   Usage Frequency    64374 non-null  int64
 5   Support Calls      64374 non-null  int64
 6   Payment Delay      64374 non-null  int64
 7   Subscription Type  64374 non-null  object
 8   Contract Length    64374 non-null  object
 9   Total Spend        64374 non-null  int64
 10  Last Interaction   64374 non-null  int64
 11  Churn              64374 non-null  int64
```

```
dtypes: int64(9), object(3)
memory usage: 5.9+ MB
```

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler


data = pd.read_csv('/content/customer_churn_dataset-testing-master.csv')



#selection of numerical attributes
numerical_features = data[['Age', 'Tenure', 'Usage Frequency', 'Support Calls', 'Payment Delay', 'Total Spenc
```

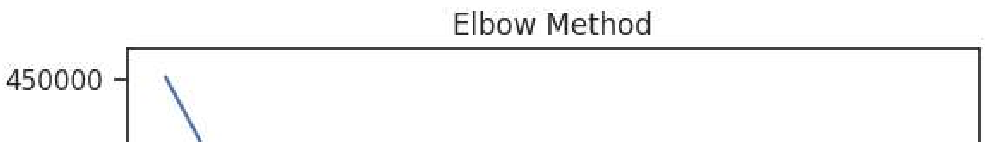
```
# Perform data scaling
scaler = StandardScaler()
scaled_data = scaler.fit_transform(numerical_features)

# Choose the number of clusters using the elbow method
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_data)
```



```
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:
  warnings.warn(
```



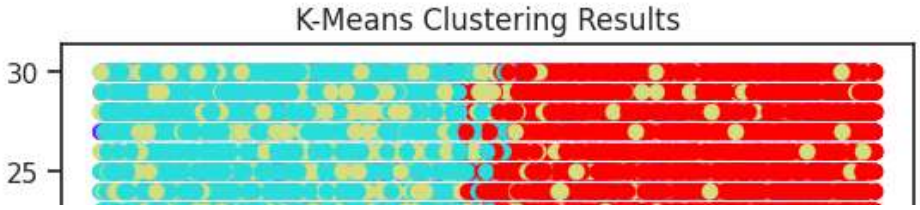

```
# rom the elbow plot the analysis chooses four clusters
k = 4
```

Automatic saving failed. This file was updated remotely or in another tab. Show diff

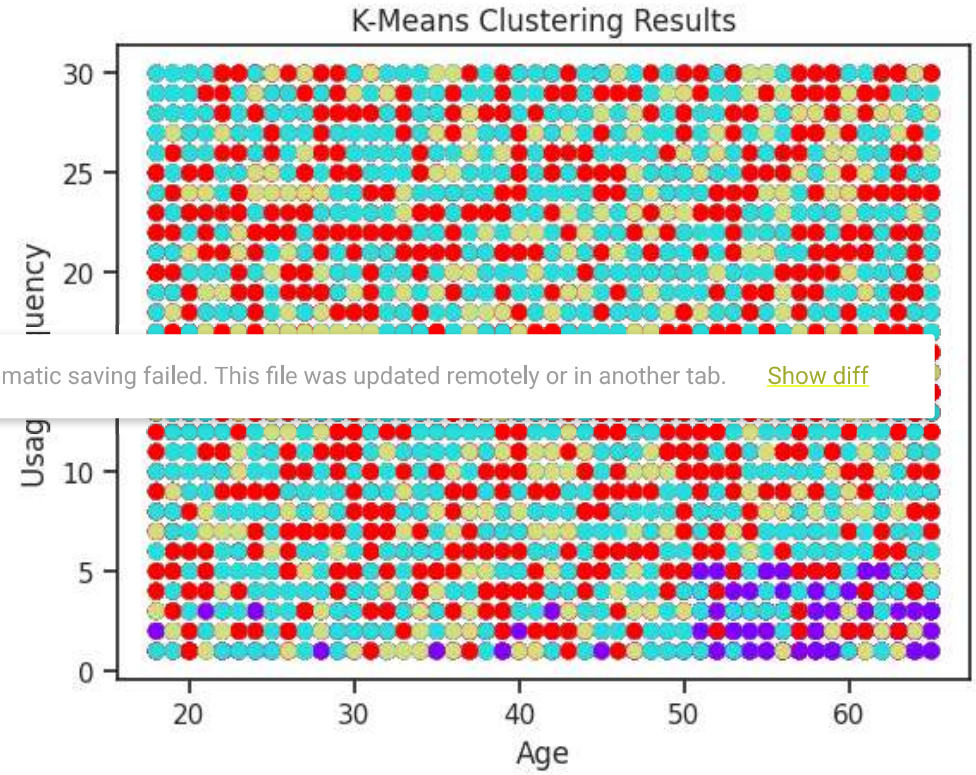
```
# Add cluster labels to the original data
data['cluster'] = kmeans.labels_

# Scatter plot of the cluster1
plt.scatter(data['Total Spend'], data['Usage Frequency'], c=data['cluster'], cmap='rainbow')
plt.xlabel('Total Spend')
plt.ylabel('Usage Frequency')
plt.title('K-Means Clustering Results')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default valu
  warnings.warn(
```




```
# Scatter plot of the cluster1
plt.scatter(data['Age'], data['Usage Frequency'], c=data['cluster'], cmap='rainbow')
plt.xlabel('Age')
plt.ylabel('Usage Frequency')
plt.title('K-Means Clustering Results')
plt.show()
```



✓ 8s completed at 12:06 PM

Automatic saving failed. This file was updated remotely or in another tab. Show diff