In [12]:
```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("customer_segmentation_data.csv")
df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53503 entries, 0 to 53502
Data columns (total 20 columns):
 #   Column                          Non-Null Count  Dtype 
---  ------                          --------------  ----- 
 0   Customer ID                     53503 non-null  int64 
 1   Age                             53503 non-null  int64 
 2   Gender                          53503 non-null  object
 3   Marital Status                  53503 non-null  object
 4   Education Level                 53503 non-null  object
 5   Geographic Information          53503 non-null  object
 6   Occupation                      53503 non-null  object
 7   Income Level                    53503 non-null  int64 
 8   Behavioral Data                 53503 non-null  object
 9   Purchase History                53503 non-null  object
 10  Interactions with Customer Service  53503 non-null  object
 11  Insurance Products Owned        53503 non-null  object
 12  Coverage Amount                 53503 non-null  int64 
 13  Premium Amount                  53503 non-null  int64 
 14  Policy Type                     53503 non-null  object
 15  Customer Preferences            53503 non-null  object
 16  Preferred Communication Channel  53503 non-null  object
 17  Preferred Contact Time          53503 non-null  object
 18  Preferred Language              53503 non-null  object
 19  Segmentation Group              53503 non-null  object
dtypes: int64(5), object(15)
memory usage: 8.2+ MB
```

In [4]:
```
df.isna().sum()

# Check for duplicate rows
df.duplicated().sum()
```

Out[4]:0

In [6]:
```
X = df.drop(['Customer ID', 'Segmentation Group'], axis=1)

# ----- Encode categorical columns -----
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

for col in X.columns:
    if X[col].dtype == 'object':
        X[col] = le.fit_transform(X[col])

# ----- View processed data frame -----
X.head()
```

Out[6]:

	Age	Gender	Marital Status	Education Level	Geographic Information	Occupation	Income Level	Behavioral Data	Purchase History	Interactions with Customer Service	Insurance Products Owned	Coverage Amount	Premium Amount	Policy Type	C Pref
0	23	0	1	0	22	3	70541	4	270	4	1	366603	2749	2	
1	26	1	4	2	10	5	54168	4	942	0	0	780236	1966	2	
2	29	0	3	0	27	3	73899	4	387	1	2	773926	4413	2	
3	20	1	0	1	28	3	63381	4	582	0	1	787815	4342	1	
4	25	0	2	1	34	5	38794	0	630	0	3	366506	1276	1	

In [7]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=4, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)

df['Cluster'].value_counts()
df.head()
```

D:\anaconda\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)

Out[7]:

	Customer ID	Age	Gender	Marital Status	Education Level	Geographic Information	Occupation	Income Level	Behavioral Data	Purchase History	...	Insurance Products Owned	Coverage Amount	Premium Amount	Policy Type
0	84966	23	Female	Married	Associate Degree	Mizoram	Entrepreneur	70541	policy5	04-10-2018	...	policy2	366603	2749	Group
1	95568	26	Male	Widowed	Doctorate	Goa	Manager	54168	policy5	11-06-2018	...	policy1	780236	1966	Group
2	10544	29	Female	Single	Associate Degree	Rajasthan	Entrepreneur	73899	policy5	06-05-2021	...	policy3	773926	4413	Group
3	77033	20	Male	Divorced	Bachelor's Degree	Sikkim	Entrepreneur	63381	policy5	09-02-2018	...	policy2	787815	4342	Family
4	88160	25	Female	Separated	Bachelor's Degree	West Bengal	Manager	38794	policy1	09-10-2018	...	policy4	366506	1276	Family

5 rows × 21 columns

In [8]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Scale / standardize features so all columns are on the same scale
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Fit KMeans clustering model
kmeans = KMeans(n_clusters=4, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)

# Check how many customers are in each cluster
print(df['Cluster'].value_counts())
df.head()
```

D:\anaconda\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)

```
Cluster
3    15427
2    13511
1    12493
0    12072
Name: count, dtype: int64
```

Out[8]:

	Customer ID	Age	Gender	Marital Status	Education Level	Geographic Information	Occupation	Income Level	Behavioral Data	Purchase History	...	Insurance Products Owned	Coverage Amount	Premium Amount	Policy Type
0	84966	23	Female	Married	Associate Degree	Mizoram	Entrepreneur	70541	policy5	04-10-2018	...	policy2	366603	2749	Group
1	95568	26	Male	Widowed	Doctorate	Goa	Manager	54168	policy5	11-06-2018	...	policy1	780236	1966	Group
2	10544	29	Female	Single	Associate Degree	Rajasthan	Entrepreneur	73899	policy5	06-05-2021	...	policy3	773926	4413	Group
3	77033	20	Male	Divorced	Bachelor's Degree	Sikkim	Entrepreneur	63381	policy5	09-02-2018	...	policy2	787815	4342	Family
4	88160	25	Female	Separated	Bachelor's Degree	West Bengal	Manager	38794	policy1	09-10-2018	...	policy4	366506	1276	Family

5 rows × 21 columns

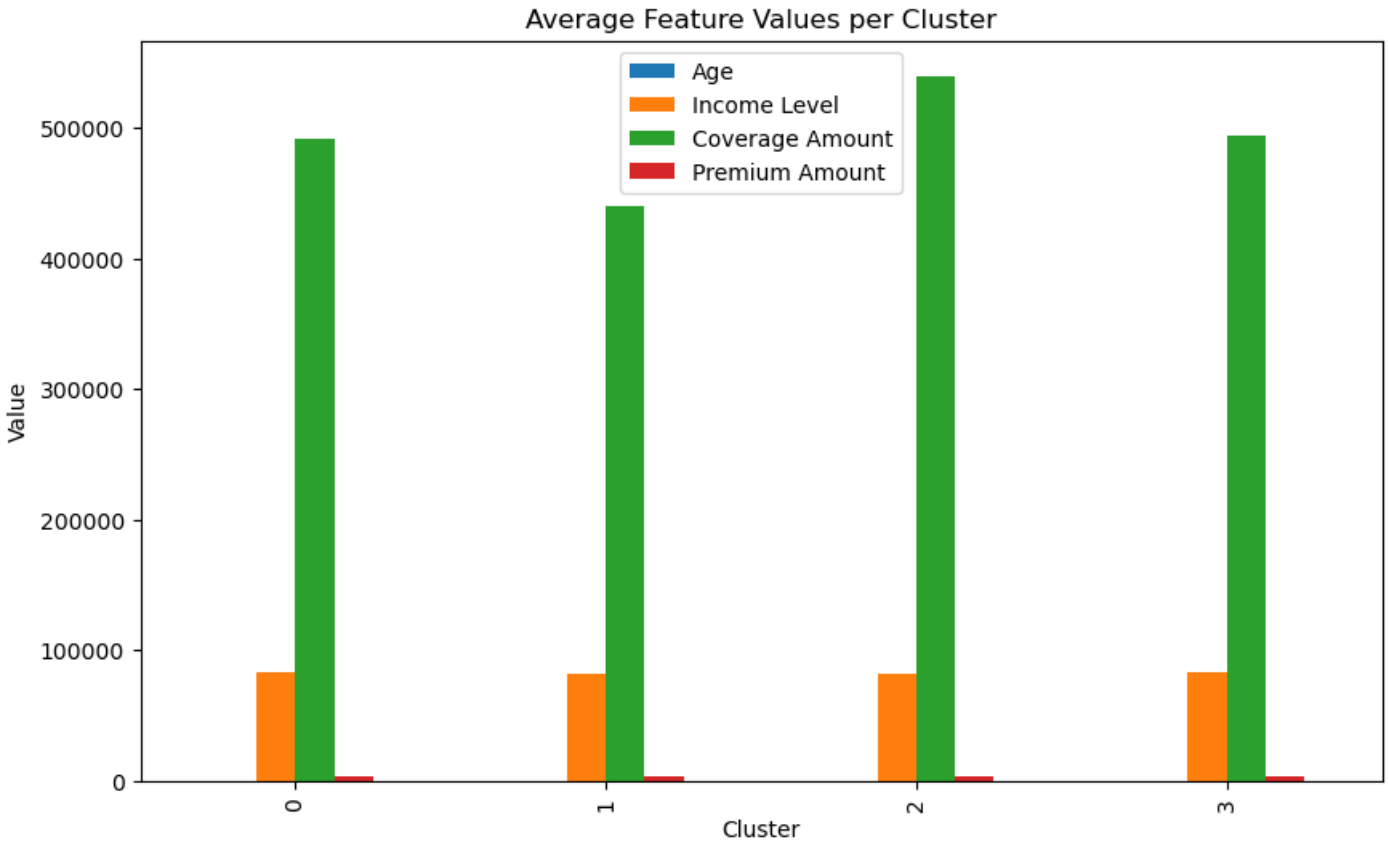
In [10]:
```
# List of numeric columns only
numeric_cols = ['Age', 'Income Level', 'Coverage Amount', 'Premium Amount']

cluster_summary = df.groupby('Cluster')[numeric_cols].mean()
cluster_summary
```

Out[10]:

Cluster	Age	Income Level	Coverage Amount	Premium Amount
0	43.700464	82896.975812	491907.670726	2990.450050
1	44.495317	82448.991275	440269.043704	3070.237573
2	44.650359	82348.480941	539765.529346	3057.186145
3	43.752512	83293.951514	494145.806508	2982.713360

In [13]:
```
cluster_summary.plot(kind='bar', figsize=(10,6))
```

```
plt.title('Average Feature Values per Cluster')
plt.ylabel('Value')
plt.show()
```



In [18]:
```
print(df.columns)
```

```
Index(['Customer ID', 'Age', 'Gender', 'Marital Status', 'Education Level',
       'Geographic Information', 'Occupation', 'Income Level',
       'Behavioral Data', 'Purchase History',
       'Interactions with Customer Service', 'Insurance Products Owned',
       'Coverage Amount', 'Premium Amount', 'Policy Type',
       'Customer Preferences', 'Preferred Communication Channel',
       'Preferred Contact Time', 'Preferred Language', 'Segmentation Group'],
      dtype='object')
```

In [22]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

kmeans = KMeans(n_clusters=4, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
```

```
D:\anaconda\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the valu
e of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

In [23]:
```
def label_segments(row):
    if row['Cluster'] == 0:
        return 'High Value'
    elif row['Cluster'] == 1:
        return 'Low Value'
    elif row['Cluster'] == 2:
        return 'Medium Value'
    else:
        return 'New/Young Customers'

df_labeled = df.copy()
df_labeled['Customer_Segment'] = df_labeled.apply(label_segments, axis=1)
df_labeled.head()
```

Out[23]:

	Customer ID	Age	Gender	Marital Status	Education Level	Geographic Information	Occupation	Income Level	Behavioral Data	Purchase History	...	Coverage Amount	Premium Amount	Policy Type	Custom Preferenc
0	84966	23	Female	Married	Associate Degree	Mizoram	Entrepreneur	70541	policy5	04-10-2018	...	366603	2749	Group	Em
1	95568	26	Male	Widowed	Doctorate	Goa	Manager	54168	policy5	11-06-2018	...	780236	1966	Group	M
2	10544	29	Female	Single	Associate Degree	Rajasthan	Entrepreneur	73899	policy5	06-05-2021	...	773926	4413	Group	Em
3	77033	20	Male	Divorced	Bachelor's Degree	Sikkim	Entrepreneur	63381	policy5	09-02-2018	...	787815	4342	Family	T
4	88160	25	Female	Separated	Bachelor's Degree	West Bengal	Manager	38794	policy1	09-10-2018	...	366506	1276	Family	Em

5 rows × 22 columns

In [ ]: