Customer Retention Strategies

This project focuses on identifying factors that influence customer churn, proposing strategies to reduce churn, calculating Customer Lifetime Value (CLV), and identifying high-value customers at risk of leaving. The data-driven insights will help the business improve customer retention effectively.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

Step 1: Load the Dataset

We begin by loading the customer data and understanding its structure.

In [3]:
```
import pandas as pd
df = pd.read_csv("customer_churn_sample.csv")
df.head()
```

Out[3]:

	customer_id	gender	senior_citizen	partner	dependents	tenure	monthly_charges	total_charges	churn
0	CUST001	Male	0	Yes	No	1	29.85	29.85	No
1	CUST002	Female	1	No	Yes	5	56.95	284.75	Yes
2	CUST003	Male	0	Yes	No	3	53.85	161.85	No
3	CUST004	Female	1	No	Yes	10	42.30	425.30	Yes
4	CUST005	Male	0	Yes	No	12	70.70	848.40	No

Step 2: Data Cleaning

We handle missing values and convert categorical columns into numerical format to prepare for modeling.

In [4]:
```
df.isnull().sum()  # Check for missing values
df = df.dropna()   # Remove rows with missing values

# Convert categorical columns to dummy variables
df = pd.get_dummies(df, drop_first=True)
```

Step 3: Exploratory Data Analysis

We analyze which features most strongly influence customer churn.

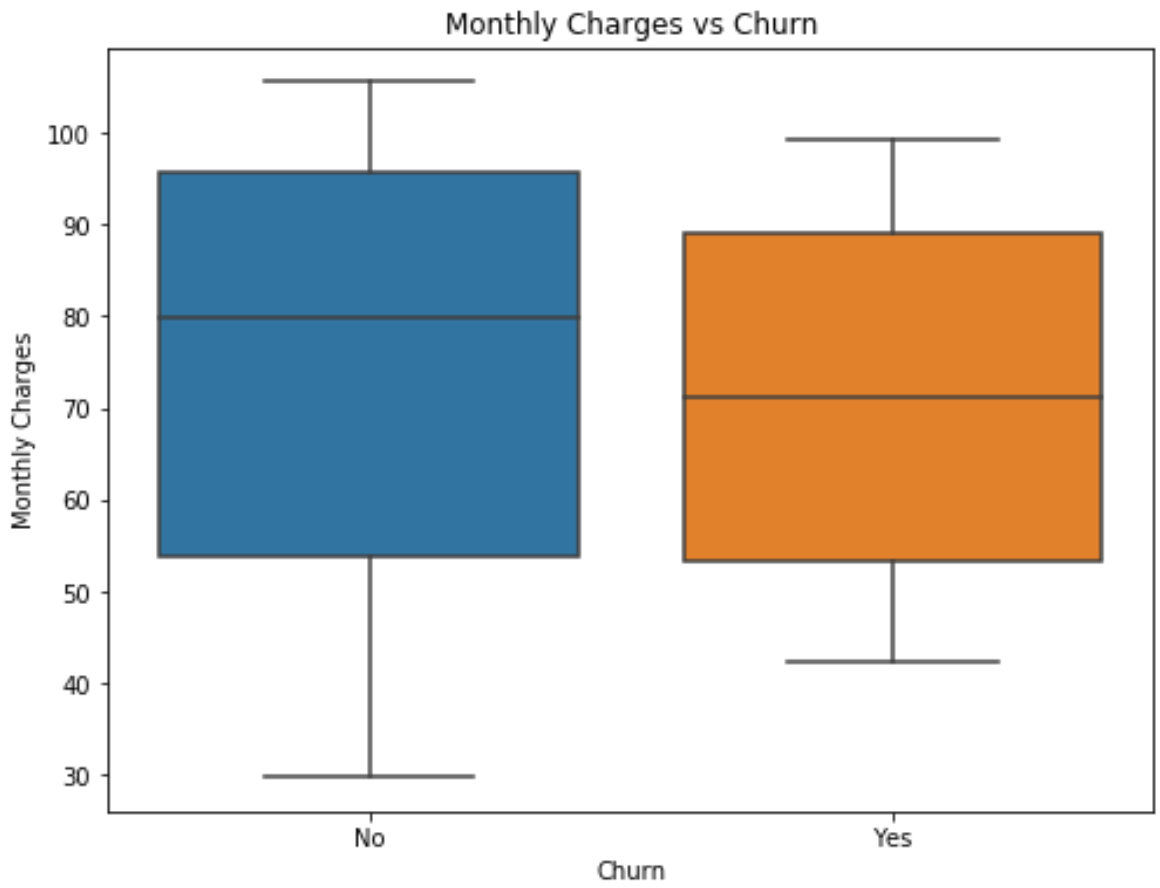
In [5]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("customer_churn_sample.csv")

# Show column names and sample values to verify
print("All columns:", df.columns)
print("Churn values:", df['churn'].unique())
print("Monthly charges sample:", df['monthly_charges'].head())

# Create the boxplot (only if the columns exist)
if 'churn' in df.columns and 'monthly_charges' in df.columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='churn', y='monthly_charges', data=df)
    plt.title("Monthly Charges vs Churn")
    plt.xlabel("Churn")
    plt.ylabel("Monthly Charges")
    plt.show()
else:
    print("Column names are incorrect. Please check your dataset.")
```

```
All columns: Index(['customer_id', 'gender', 'senior_citizen', 'partner', 'dependents',
       'tenure', 'monthly_charges', 'total_charges', 'churn'],
      dtype='object')
Churn values: ['No' 'Yes']
Monthly charges sample: 0    29.85
1    56.95
2    53.85
3    42.30
4    70.70
Name: monthly_charges, dtype: float64
```



Step 4: Predicting Churn

We train a machine learning model (Random Forest) to predict whether a customer will churn or not.

In [7]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Load dataset
df = pd.read_csv("customer_churn_sample.csv")

# Drop customer_id (it's just an identifier)
df = df.drop('customer_id', axis=1)

# Convert 'churn' column to binary: Yes → 1, No → 0
df['churn'] = df['churn'].map({'Yes': 1, 'No': 0})

# Convert categorical columns to numeric using one-hot encoding
df = pd.get_dummies(df, drop_first=True)

# Define features (X) and target (y)
X = df.drop('churn', axis=1)
y = df['churn']

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Random Forest model
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00         1

    accuracy                           1.00         4
   macro avg       1.00      1.00      1.00         4
weighted avg       1.00      1.00      1.00         4
```

Step 5: Calculating Customer Lifetime Value (CLV)

We calculate CLV using average monthly charges and tenure.

In [8]:
```
df['CLV'] = df['monthly_charges'] * df['tenure']
df[['monthly_charges', 'tenure', 'CLV']].head()
```

Out[8]:

	monthly_charges	tenure	CLV
0	29.85	1	29.85
1	56.95	5	284.75
2	53.85	3	161.55
3	42.30	10	423.00
4	70.70	12	848.40

Step 6: Identify High-Value Customers at Risk

We detect customers who are valuable (high CLV) and are predicted to churn, so we can target them with retention strategies.

In [9]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from IPython.display import display

# Step 1: Load dataset again to restore customer_id
df = pd.read_csv("customer_churn_sample.csv")

# Step 2: Save customer_id for later use
customer_ids = df['customer_id']

# Step 3: Drop customer_id temporarily (not useful for training)
df = df.drop('customer_id', axis=1)

# Step 4: Convert churn column to binary (Yes = 1, No = 0)
df['churn'] = df['churn'].map({'Yes': 1, 'No': 0})

# Step 5: One-hot encode all categorical columns
df = pd.get_dummies(df, drop_first=True)

# Step 6: Add customer_id column back for display
df['customer_id'] = customer_ids

# Step 7: Calculate Customer Lifetime Value (CLV)
df['CLV'] = df['monthly_charges'] * df['tenure']

# Step 8: Prepare data for prediction (drop churn, customer_id, CLV)
X = df.drop(['churn', 'customer_id', 'CLV'], axis=1)
y = df['churn']

# Step 9: Train a Random Forest model
model = RandomForestClassifier()
model.fit(X, y)

# Step 10: Predict churn
df['predicted_churn'] = model.predict(X)

# Step 11: Identify high-value customers who are at risk of churning
high_risk = df[(df['CLV'] > df['CLV'].quantile(0.75)) & (df['predicted_churn'] == 1)]

# Step 12: Display the results (this line shows the output)
display(high_risk[['customer_id', 'CLV', 'tenure', 'monthly_charges']])
```

	customer_id	CLV	tenure	monthly_charges
8	CUST009	5136.0	60	85.6
18	CUST019	5136.0	60	85.6

Step 7: Data-Driven Retention Strategies

Based on the analysis, we propose the following strategies:

1. Offer discounts or personalized plans to customers with high monthly charges and low tenure.
2. Implement loyalty rewards for long-term customers with high CLV.
3. Focus on improving onboarding and customer support for new users.
4. Proactively reach out to high-CLV customers predicted to churn.

Step 8: Conclusion

This project successfully used data analytics to identify churn factors, calculate CLV, and highlight valuable customers at risk. With these insights, businesses can take proactive actions to improve customer retention and reduce churn rate.