

```
In [62]: import pandas as pd
from datetime import timedelta
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
import joblib
```

```
In [63]: # Load the dataset
df = pd.read_csv("online_retail.csv", encoding='ISO-8859-1')
```

```
In [64]: # Step 1:
df.dropna(subset=['CustomerID'], inplace=True)
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df = df[df['Quantity'] > 0]
df['TotalAmount'] = df['Quantity'] * df['UnitPrice']
```

```
In [65]: # Step 2:
snapshot_date = df['InvoiceDate'].max() + timedelta(days=1)

clv_data = df.groupby('CustomerID').agg({
    'InvoiceDate': [lambda x: (snapshot_date - x.max()).days, 'count'],
    'TotalAmount': ['sum', 'mean']
})

clv_data.columns = ['Recency', 'Frequency', 'MonetaryValue', 'AvgOrderValue']
clv_data.reset_index(inplace=True)
```

```
In [66]: # Step 3:
X = clv_data[['Recency', 'Frequency', 'AvgOrderValue']]
y = clv_data['MonetaryValue']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat

model = XGBRegressor()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Model Training is Complete!")
```

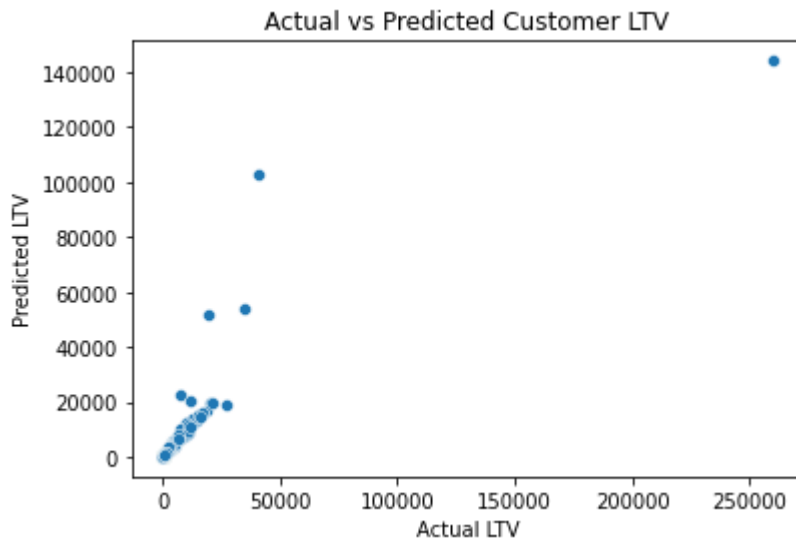
Model Training is Complete!

```
In [67]: # Step 4:
preds = model.predict(X_test)
mae = mean_absolute_error(y_test, preds)
rmse = np.sqrt(mean_squared_error(y_test, preds))
print(f"MAE: {mae:.2f}, RMSE: {rmse:.2f}")
```

MAE: 403.53, RMSE: 4681.49

```
In [68]: # Step 5:
sns.scatterplot(x=y_test, y=preds)
plt.xlabel("Actual LTV")
```

```
plt.ylabel("Predicted LTV")  
plt.title("Actual vs Predicted Customer LTV")  
plt.show()
```



In [69]:

```
# Step 6:  
joblib.dump(model, "model.pkl")  
clv_data['PredictedLTV'] = model.predict(X)  
clv_data.to_csv("predicted_ltv.csv", index=False)
```