Customer Purchase Frequency Analysis using Machine Learning

Section 1: Dataset

	CustomerID	PurchaseDate	Amount
0	C001	2023-01-05	500
1	C002	2023-01-06	300
2	C001	2023-01-15	700
3	C003	2023-01-20	450
4	C002	2023-01-25	600
5	C001	2023-02-02	800

Table 1: Sample Customer Dataset

Section 2: Analysis

```
# Purchase count per customer
df['CustomerID'].value_counts()
```

```
            count
CustomerID
C001            3
C002            2
C003            1

dtype: int64
```

Table 2a: Customer Purchase Frequency

```
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red')
plt.xlabel("Day")
plt.ylabel("Amount")
plt.title("Purchase Amount Prediction")
plt.show()
```

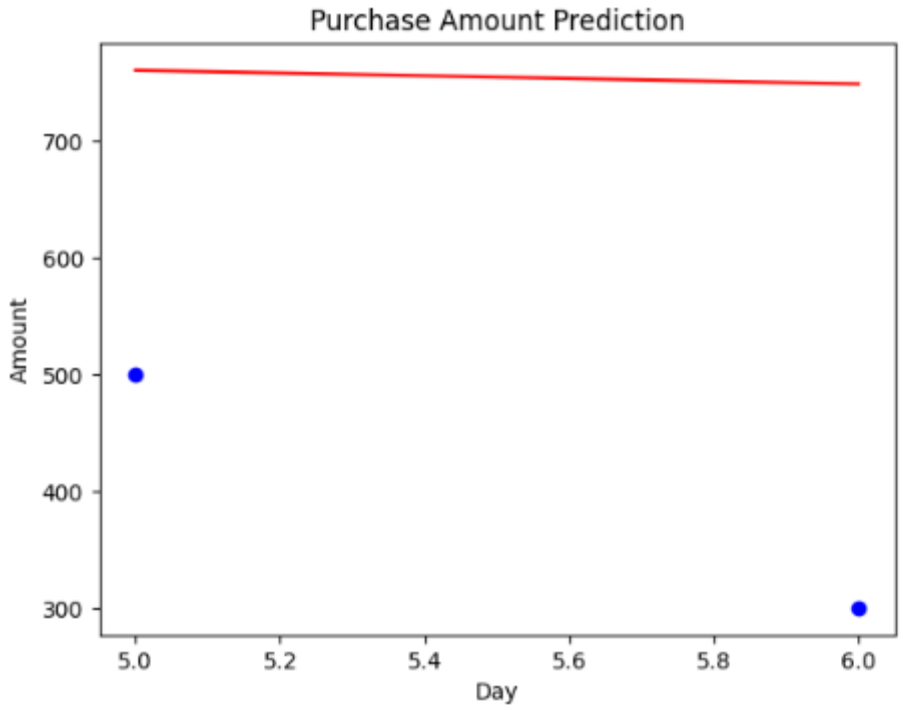

Table 2b: Total Amount per Customer

Section 3: Visualization

```
# Bar graph for purchase frequency
df['CustomerID'].value_counts().plot(kind='bar')
plt.xlabel("Customer ID")
plt.ylabel("Number of Purchases")
plt.title("Customer Purchase Frequency")
plt.show()
```

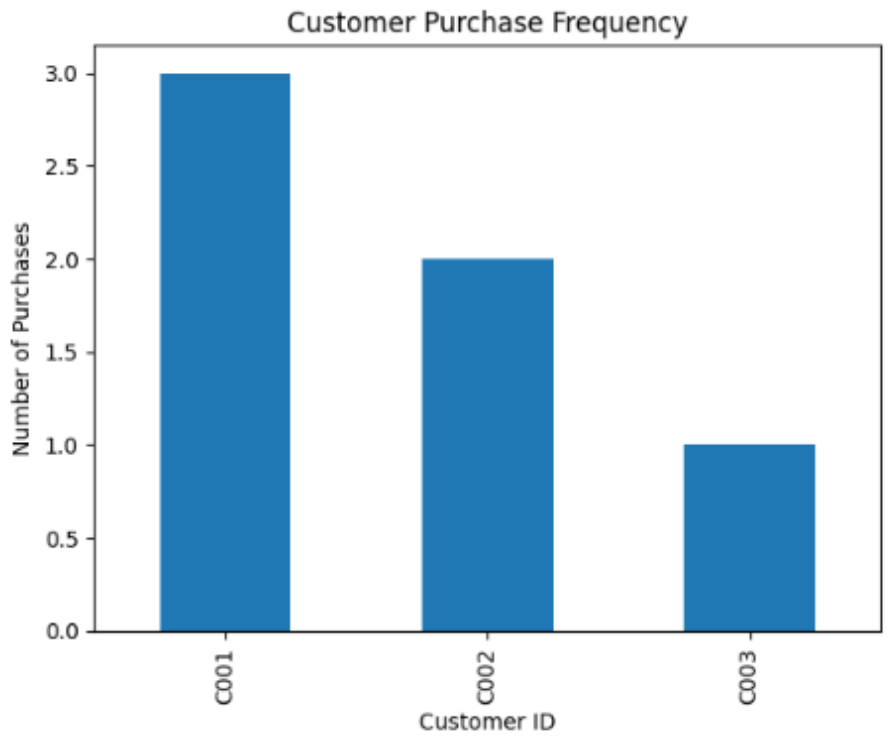

Figure 1: Customer Purchase Frequency Bar Chart

Section 4: Machine Learning Prediction

```python
# Convert PurchaseDate to Day number for ML
df['Day'] = df['PurchaseDate'].dt.day

# Features and target
X = df[['Day']]
y = df['Amount']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Check error
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 134460.46619646126
```

Figure 2: Purchase Amount Prediction using Linear Regression

Section 5: Conclusion

This project analyzes customer purchase frequency using a sample dataset.

Machine Learning (Linear Regression) is used to predict purchase amounts.

The results show that simple ML models can be applied for basic prediction tasks.

This project is suitable for beginners to understand data analysis and ML concepts.