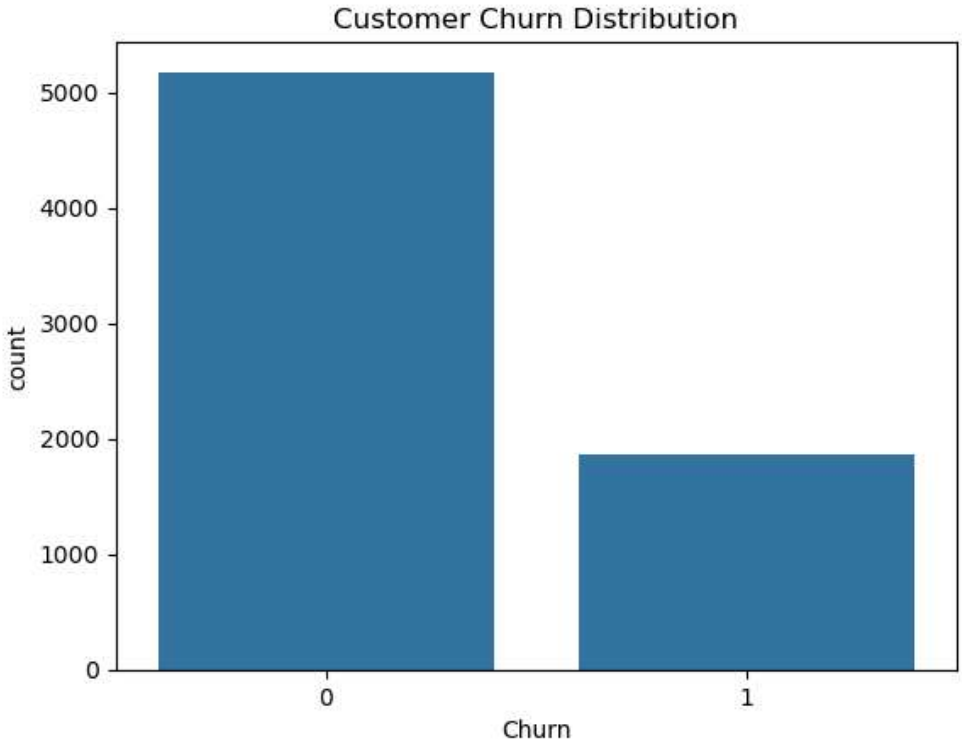
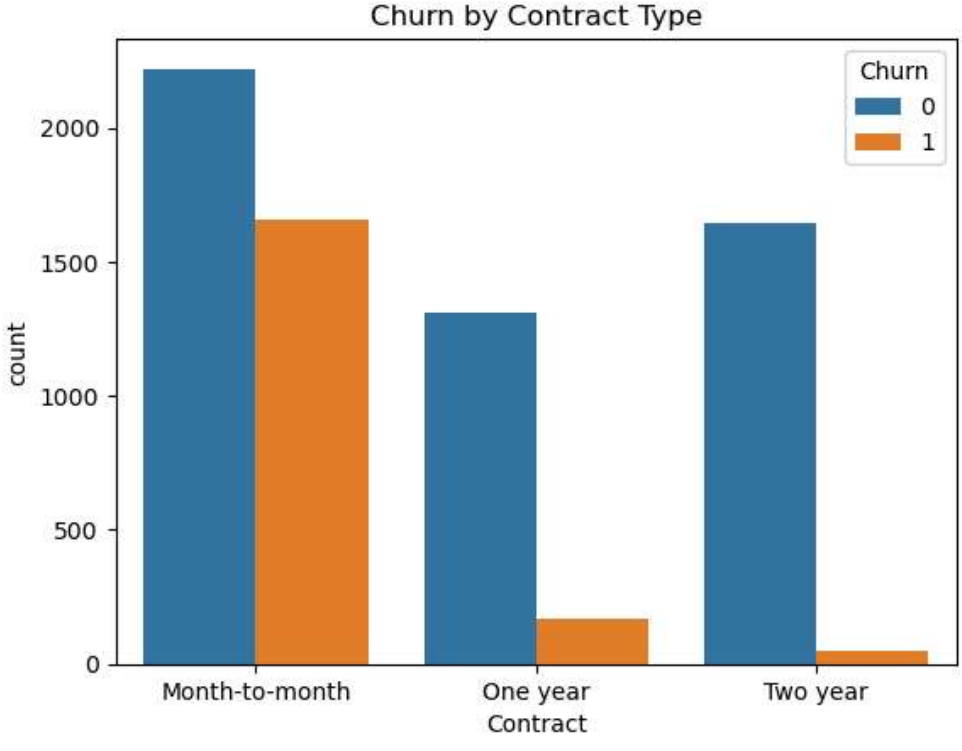
In [38]:
```
import pandas as pd

df = pd.read_csv(r"C:\Users\humer\Downloads\Telco-Customer-Churn.c


print(df.shape)
print(df.info())
print(df.head())
```

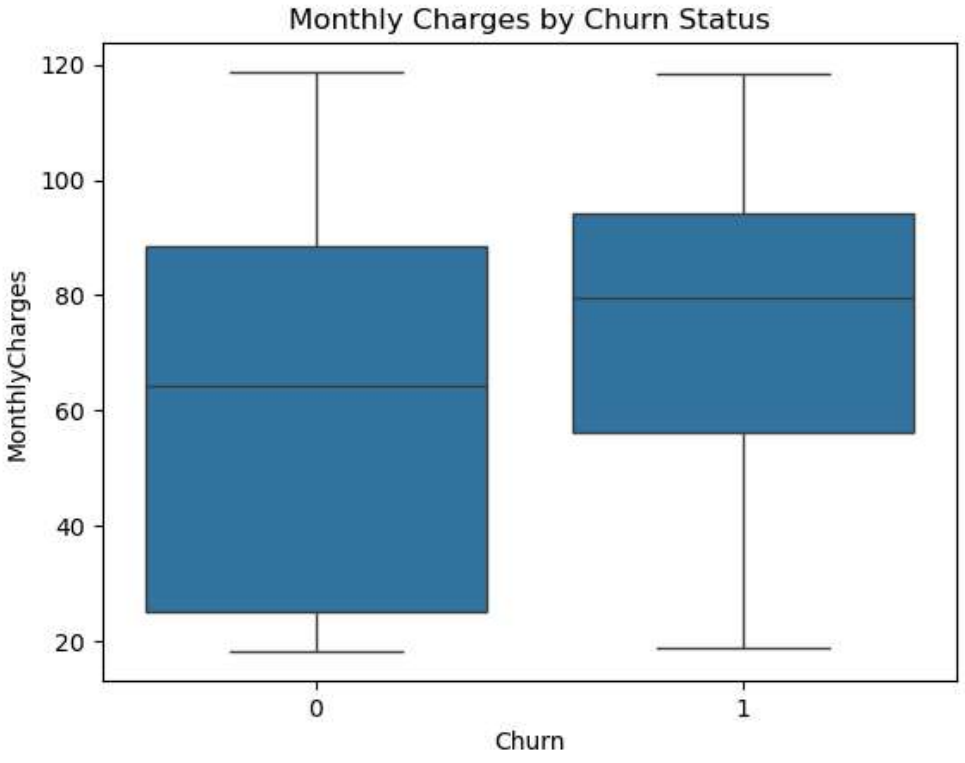
```
(7043, 21)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
None
   customerID  gender  SeniorCitizen Partner Dependents  tenure Pho
neService  \
0  7590-VHVEG  Female              0     Yes         No       1
No
1  5575-GNVDE    Male              0      No         No      34
Yes
2  3668-QPYBK    Male              0      No         No       2
Yes
3  7795-CFOCW    Male              0      No         No      45
No
4  9237-HQITU  Female              0      No         No       2
Yes

      MultipleLines InternetService OnlineSecurity  ... DeviceProte
ction  \
0  No phone service             DSL             No  ...
No
1                No             DSL            Yes  ...
Yes
2                No             DSL            Yes  ...
No
3  No phone service             DSL            Yes  ...
Yes
```

```
4                No     Fiber optic             No  ...
No

  TechSupport StreamingTV StreamingMovies        Contract Paperless
Billing  \
0          No          No              No  Month-to-month
Yes
1          No          No              No        One year
No
2          No          No              No  Month-to-month
Yes
3         Yes          No              No        One year
No
4          No          No              No  Month-to-month
Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

In [4]:
```python
import seaborn as sns
import matplotlib.pyplot as plt


sns.countplot(x="Churn", data=df)
plt.title("Customer Churn Distribution")
plt.show()


sns.countplot(x="Contract", hue="Churn", data=df)
plt.title("Churn by Contract Type")
plt.show()


sns.boxplot(x="Churn", y="MonthlyCharges", data=df)
plt.title("Monthly Charges by Churn Status")
plt.show()
```


Customer Churn Distribution
count
5000
4000
3000
2000
1000
0
0
1
Churn


Churn by Contract Type
Churn
0
1
count
2000
1500
1000
500
0
Month-to-month
One year
Two year
Contract



In [47]:
```
# Logistic regression
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(max_iter=1000)

# OR Decision Tree
# from sklearn.tree import DecisionTreeClassifier
# model = DecisionTreeClassifier()

model.fit(X_train, y_train)
```

Out[47]:
```
LogisticRegression
LogisticRegression(max_iter=1000)
```

In [49]:
```
from sklearn.metrics import confusion_matrix, classification_repor

y_pred = model.predict(X_test)

print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test,
```

```
Confusion Matrix:
 [[942  94]
 [152 221]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.91      0.88      1036
           1       0.70      0.59      0.64       373

    accuracy                           0.83      1409
   macro avg       0.78      0.75      0.76      1409
weighted avg       0.82      0.83      0.82      1409
```

In [50]:
```
# For Logistic Regression
import numpy as np

feature_importance = pd.Series(np.abs(model.coef_[0]), index=X.col
top_features = feature_importance.sort_values(ascending=False).hea

print("Top 5 Features Influencing Churn:")
print(top_features)
```

```
Top 5 Features Influencing Churn:
Contract_Two year              1.348139
TotalCharges_20.2              1.339040
TotalCharges_20.5              1.145939
TotalCharges_19.95             1.034152
InternetService_Fiber optic    1.030427
dtype: float64
```

In [33]:
```
df["Churn"] = df["Churn"].map({"Yes": 1, "No": 0})


categorical_cols = df.select_dtypes(include="object").columns
print("📋 Categorical columns to encode:", categorical_cols.tolist


df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_fir

print("✅ Encoding complete. New shape:", df_encoded.shape)
```

```
📋 Categorical columns to encode: ['gender', 'Partner', 'Dependent
s', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecu
rity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Streamin
gTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMe
thod']
✅ Encoding complete. New shape: (7032, 31)
```

In [ ]:
```

```