Customer Churn Analysis – Business Report

Executive Summary

This report presents the results of a customer churn prediction analysis aimed at identifying customers who are likely to discontinue services. The model helps the business understand overall churn risk and key drivers of customer attrition, enabling data-driven retention strategies.

Key Business Metrics:

Average Churn Risk: 26.20%
→ Approximately 1 in 4 customers are at risk of churn.

Model Performance Summary:

▪The confusion matrix shows strong predictive capability in distinguishing churned vs non-churned customers.

▪Feature importance analysis highlights contract type, monthly charges, and internet service as key churn drivers.

Business Recommendations:

▪Introduce incentives to convert month-to-month customers to long-term contracts.

▪Target high-risk payment methods with loyalty offers.

▪Design service-specific retention strategies for Fiber Optic users.

▪Use churn probability as a KPI to track retention improvements over time.

In [2]:

```python
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.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_r
```

In [3]:

```python
# --- STEP 1: LOAD & CLEAN DATA ---

df = pd.read_csv('churn_data.csv')

df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df.dropna(inplace=True)

df = df.drop('customerID', axis=1)

label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

print("Data Cleaned Successfully!")
```

```
Data Cleaned Successfully!
```

In [4]:

```python
# --- STEP 2: SPLIT DATA ---

X = df.drop('Churn', axis=1)
y = df['Churn']

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

In [5]:

```python
# --- STEP 3: TRAIN MODEL (XGBoost) ---

print("Training XGBoost Model...")
model = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
model.fit(X_train, y_train)
```

```
Training XGBoost Model...
```

```
c:\Users\Acer\AppData\Local\Programs\Python\Python313\Lib\site-packages\xgboos
t\training.py:199: UserWarning: [19:37:31] WARNING: C:\actions-runner\_work\xgb
oost\xgboost\src\learner.cc:790:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
```

Out[5]:

XGBClassifier

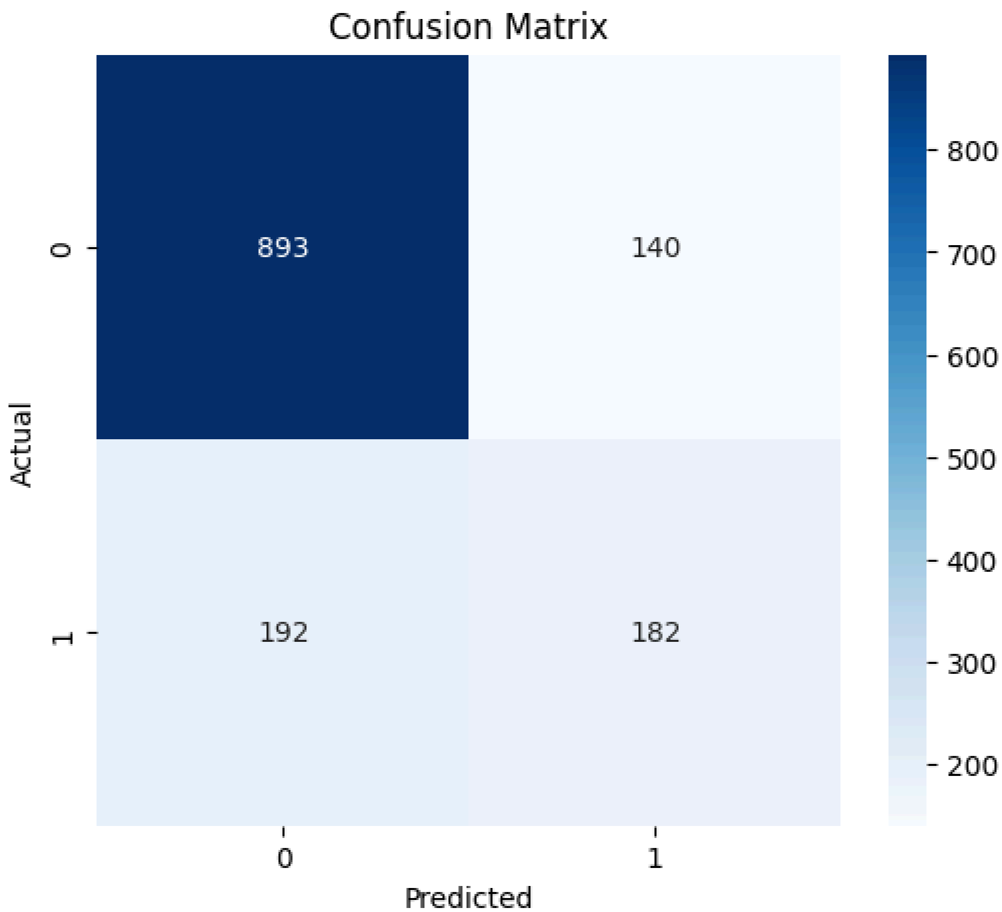
Parameters

	objective	'binary:logistic'
	base_score	None
	booster	None
	callbacks	None
	colsample_bylevel	None
	colsample_bynode	None
	colsample_bytree	None
	device	None
	early_stopping_rounds	None
	enable_categorical	False
	eval_metric	'logloss'
	feature_types	None
	feature_weights	None
	gamma	None
	grow_policy	None
	importance_type	None
	interaction_constraints	None
	learning_rate	None
	max_bin	None
	max_cat_threshold	None
	max_cat_to_onehot	None
	max_delta_step	None
	max_depth	None
	max_leaves	None
	min_child_weight	None
	missing	nan
	monotone_constraints	None
	multi_strategy	None

In [7]:
```
# --- STEP 4: EVALUATE MODEL ---

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"\nModel Accuracy: {accuracy:.2%}")
```

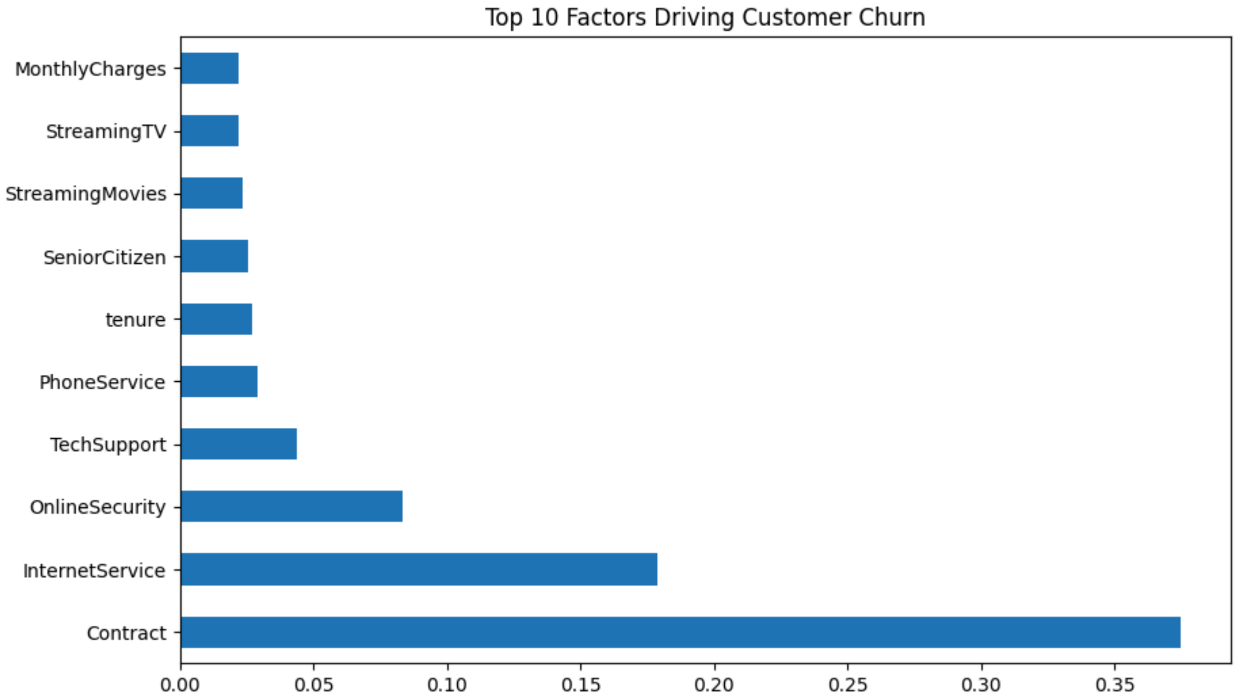
```
Model Accuracy: 76.40%
```

In [8]:
```
# Visualization 1: Confusion Matrix

plt.figure(figsize=(6,5))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues
plt.title('Confusion Matrix')
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.savefig('confusion_matrix.png')
plt.show()
```



In [9]:
```
# Visualization 2: Feature Importance

plt.figure(figsize=(10,6))
feature_importances = pd.Series(model.feature_importances_, index=X.columns)
feature_importances.nlargest(10).plot(kind='barh')
plt.title('Top 10 Factors Driving Customer Churn')
plt.savefig('feature_importance.png')
```

```python
plt.show()
```



In [10]:

```python
# --- STEP 5: SAVE RESULTS FOR DASHBOARD ---

results = X_test.copy()
results['Actual Churn'] = y_test
results['Predicted Churn'] = y_pred
results['Churn Probability'] = model.predict_proba(X_test)[:, 1]

results.to_csv('churn_predictions.csv', index=False)
print("Success! Results saved to 'churn_predictions.csv'")
```

```
Success! Results saved to 'churn_predictions.csv'
```

---------- Power BI Desktop ----------

Insights from the Dashboard / Analysis

1 Contract Type vs Churn:

Customers on month-to-month contracts exhibit significantly higher churn compared to those on one-year or two-year contracts.

Business Insight: Encouraging long-term contracts can reduce churn.

2 Payment Method vs Churn:

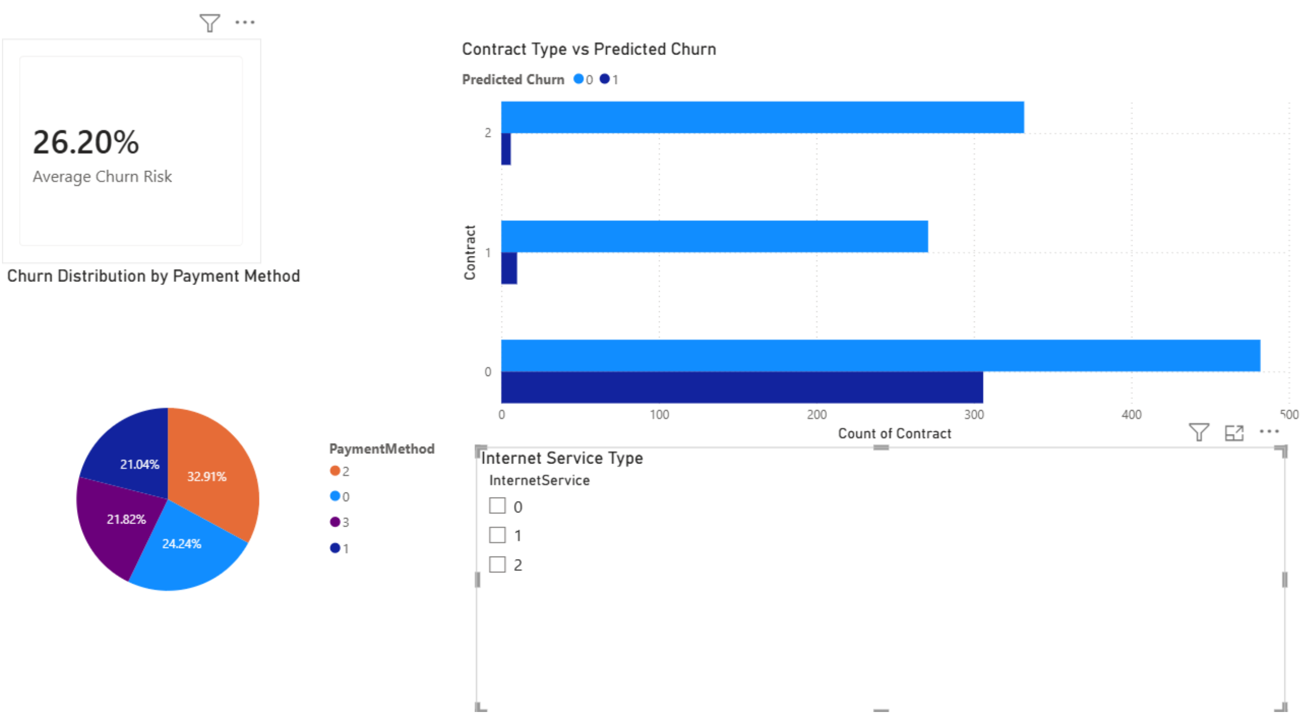
Customers using electronic check show a higher proportion of churned customers compared to other payment methods.

Business Insight: Promoting stable digital or auto-payment methods may improve retention.

3 Internet Service Impact:

Churn behavior varies across Fiber Optic and DSL customers.

Business Insight: Service-specific retention offers can be designed based on churn risk.



Conclusion

The churn prediction model provides actionable insights that can help reduce customer attrition. By focusing on high-risk segments identified in the dashboard and report,the business can proactively improve customer retention and long-term revenue.