

```
In [6]: 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, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import joblib
```

```
In [8]: file_path=r"C:\Users\91801\OneDrive\Prediction_file.xlsx"
```

```
In [10]: # Define the sheet name to read data from
sheet_name = 'vw_ChurnData'
# Read the data from the specified sheet into a pandas DataFrame
data = pd.read_excel(file_path, sheet_name=sheet_name)
# Display the first few rows of the fetched data
print(data.head())
```

	Customer_ID	Gender	Age	Married	State	Number_of_Referrals	\
0	11098-MAD	Female	30	Yes	Madhya Pradesh	0	
1	11114-PUN	Male	51	No	Punjab	5	
2	11167-WES	Female	43	Yes	West Bengal	3	
3	11179-MAH	Male	35	No	Maharashtra	10	
4	11180-TAM	Male	75	Yes	Tamil Nadu	12	

	Tenure_in_Months	Value_Deal	Phone_Service	Multiple_Lines	...	\
0	31	Deal 1	Yes	No	...	
1	9	Deal 5	Yes	No	...	
2	28	Deal 1	Yes	Yes	...	
3	12	NaN	Yes	No	...	
4	27	Deal 2	Yes	No	...	

	Payment_Method	Monthly_Charge	Total_Charges	Total_Refunds	\
0	Bank Withdrawal	95.099998	6683.399902	0.00	
1	Bank Withdrawal	49.150002	169.050003	0.00	
2	Bank Withdrawal	116.050003	8297.500000	42.57	
3	Credit Card	84.400002	5969.299805	0.00	
4	Credit Card	72.599998	4084.350098	0.00	

	Total_Extra_Data_Charges	Total_Long_Distance_Charges	Total_Revenue	\
0	0	631.719971	7315.120117	
1	10	122.370003	301.420013	
2	110	1872.979980	10237.910156	
3	0	219.389999	6188.689941	
4	140	332.079987	4556.430176	

	Customer_Status	Churn_Category	Churn_Reason
0	Stayed	Others	Others
1	Churned	Competitor	Competitor had better devices
2	Stayed	Others	Others
3	Stayed	Others	Others
4	Stayed	Others	Others

[5 rows x 32 columns]

```
In [11]: # Drop columns that won't be used for prediction
data = data.drop(['Customer_ID', 'Churn_Category', 'Churn_Reason'], axis=1)

# List of columns to be label encoded
columns_to_encode = [
    'Gender', 'Married', 'State', 'Value_Deal', 'Phone_Service', 'Multiple_Lines',
    'Internet_Service', 'Internet_Type', 'Online_Security', 'Online_Backup',
    'Device_Protection_Plan', 'Premium_Support', 'Streaming_TV', 'Streaming_Movies',
    'Streaming_Music', 'Unlimited_Data', 'Contract', 'Paperless_Billing',
    'Payment_Method'
]

# Encode categorical variables except the target variable
label_encoders = {}
for column in columns_to_encode:
    label_encoders[column] = LabelEncoder()
    data[column] = label_encoders[column].fit_transform(data[column])

# Manually encode the target variable 'Customer_Status'
data['Customer_Status'] = data['Customer_Status'].map({'Stayed': 0, 'Churned': 1})
# Split data into features and target
X = data.drop('Customer_Status', axis=1)
y = data['Customer_Status']
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [12]: # Initialize the Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the model
rf_model.fit(X_train, y_train)
```

```
Out[12]: RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
In [13]: #Evaluate Model
# Make predictions
y_pred = rf_model.predict(X_test)

# Evaluate the model
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

Confusion Matrix:

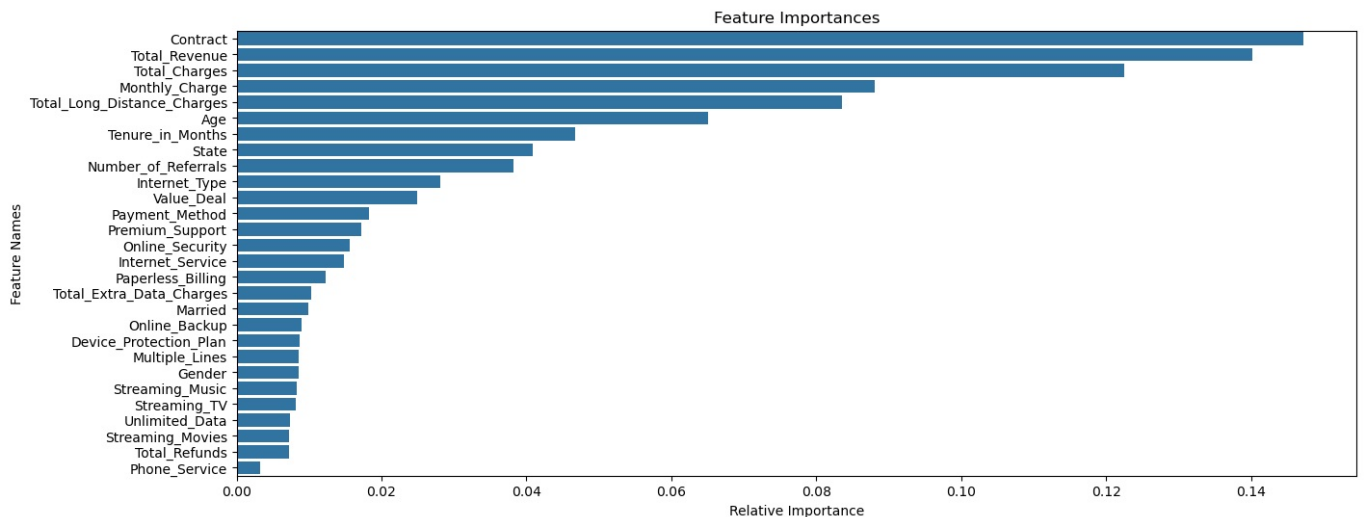
```
[[783  64]
 [126 229]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.92	0.89	847
1	0.78	0.65	0.71	355
accuracy			0.84	1202
macro avg	0.82	0.78	0.80	1202
weighted avg	0.84	0.84	0.84	1202

```
In [16]: # Feature Selection using Feature Importance
importances = rf_model.feature_importances_
indices = np.argsort(importances)[::-1]

# Plot the feature importances
plt.figure(figsize=(15, 6))
sns.barplot(x=importances[indices], y=X.columns[indices])
plt.title('Feature Importances')
plt.xlabel('Relative Importance')
plt.ylabel('Feature Names')
plt.show()
```



```
In [19]: # Define the path to the Joiner Data Excel file
file_path = r"C:\Users\91801\OneDrive\Prediction_file.xlsx"
# Define the sheet name to read data from
sheet_name = 'vw_JoinData'

# Read the data from the specified sheet into a pandas DataFrame
new_data = pd.read_excel(file_path, sheet_name=sheet_name)

# Display the first few rows of the fetched data
print(new_data.head())

# Retain the original DataFrame to preserve unencoded columns
original_data = new_data.copy()

# Retain the Customer_ID column
```

```
customer_ids = new_data['Customer_ID']

# Drop columns that won't be used for prediction in the encoded DataFrame
new_data = new_data.drop(['Customer_ID', 'Customer_Status', 'Churn_Category', 'Churn_Reason'], axis=1)

# Encode categorical variables using the saved label encoders
for column in new_data.select_dtypes(include=['object']).columns:
    new_data[column] = label_encoders[column].transform(new_data[column])

# Make predictions
new_predictions = rf_model.predict(new_data)

# Add predictions to the original DataFrame
original_data['Customer_Status_Predicted'] = new_predictions

# Filter the DataFrame to include only records predicted as "Churned"
original_data = original_data[original_data['Customer_Status_Predicted'] == 1]

# Save the results
original_data.to_csv(r"C:\Users\91801\Documents\Predictions.csv", index=False)
```

	Customer_ID	Gender	Age	Married	State	Number_of_Referrals	\
0	11751-TAM	Female	18	No	Tamil Nadu	5	
1	12056-WES	Male	27	No	West Bengal	2	
2	12136-RAJ	Female	25	Yes	Rajasthan	2	
3	12257-ASS	Female	39	No	Assam	9	
4	12340-DEL	Female	51	Yes	Delhi	0	

	Tenure_in_Months	Value_Deal	Phone_Service	Multiple_Lines	...	\
0	7	Deal 5	No	No	...	
1	20	NaN	Yes	No	...	
2	35	NaN	Yes	No	...	
3	1	NaN	Yes	No	...	
4	10	NaN	Yes	No	...	

	Payment_Method	Monthly_Charge	Total_Charges	Total_Refunds	\
0	Mailed Check	24.299999	38.450001	0.0	
1	Bank Withdrawal	90.400002	268.450012	0.0	
2	Bank Withdrawal	19.900000	19.900000	0.0	
3	Credit Card	19.549999	19.549999	0.0	
4	Credit Card	62.799999	62.799999	0.0	

	Total_Extra_Data_Charges	Total_Long_Distance_Charges	Total_Revenue	\
0	0	0.000000	38.450001	
1	0	94.440002	362.890015	
2	0	11.830000	31.730000	
3	0	10.200000	29.750000	
4	0	42.189999	104.989998	

	Customer_Status	Churn_Category	Churn_Reason
0	Joined	Others	Others
1	Joined	Others	Others
2	Joined	Others	Others
3	Joined	Others	Others
4	Joined	Others	Others

[5 rows x 32 columns]

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js