

Importing Libraries & Data Load

```
In [1]: ▶ 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

# Define the path to the Excel file
file_path = r"D:\DMA_Project\DMA_PROJECT_EXCEL_FILES\New_files\FINAL\Full Time Applications\Prediction_Da

# 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	19877-DEL	Male	35	No	Delhi	7	
1	58353-MAH	Female	45	Yes	Maharashtra	14	
2	25063-WES	Male	51	No	West Bengal	4	
3	59787-KAR	Male	79	No	Karnataka	3	
4	28544-TAM	Female	80	No	Tamil Nadu	3	

	Tenure_in_Months	Value_Deal	Phone_Service	Multiple_Lines	...	\
0	27	None	Yes	No	...	
1	13	None	Yes	Yes	...	
2	35	Deal 5	Yes	No	...	
3	21	Deal 4	Yes	No	...	
4	8	None	Yes	No	...	

	Payment_Method	Monthly_Charge	Total_Charges	Total_Refunds	\
0	Credit Card	65.6	593.30	0.00	
1	Credit Card	-4.0	542.40	38.33	
2	Bank Withdrawal	73.9	280.85	0.00	
3	Bank Withdrawal	98.0	1237.85	0.00	
4	Credit Card	83.9	267.40	0.00	

	Total_Extra_Data_Charges	Total_Long_Distance_Charges	Total_Revenue	\
0	0	381.51	974.81	
1	10	96.21	610.28	
2	0	134.60	415.45	
3	0	361.66	1599.51	
4	0	22.14	289.54	

	Customer_Status	Churn_Category	Churn_Reason
0	Stayed	Others	Others
1	Stayed	Others	Others
2	Churned	Competitor	Competitor had better devices
3	Churned	Dissatisfaction	Product dissatisfaction
4	Churned	Dissatisfaction	Network reliability

[5 rows x 32 columns]

Data Preprocessing

```
In [2]: ▶ # 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 [6]: ▶ y.head()
```

```
Out[6]: 0    0
        1    0
        2    1
        3    1
        4    1
        Name: Customer_Status, dtype: int64
```

Training Random Forest Model

```
In [9]: ▶ # 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[9]: RandomForestClassifier(random_state=42)

**In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.**

Evaluating the Model

```

In [10]: ► # 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))

# 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()

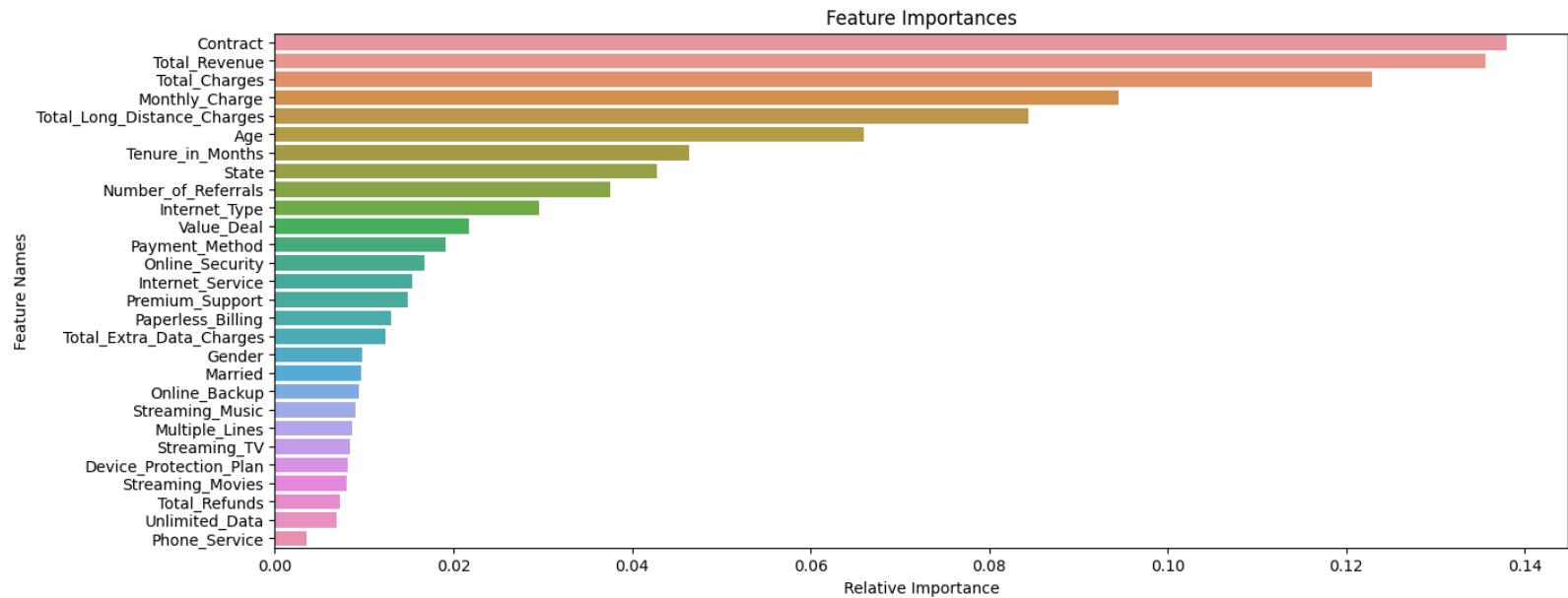
```

Confusion Matrix:

```
[[791  50]
 [125 236]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.94	0.90	841
1	0.83	0.65	0.73	361
accuracy			0.85	1202
macro avg	0.84	0.80	0.81	1202
weighted avg	0.85	0.85	0.85	1202



Using the model for making Predictions on New Data

```
In [11]: ▶ # Define the path to the Joiner Data Excel file
file_path = r"D:\DMA_Project\DMA_PROJECT_EXCEL_FILES\New_files\FINAL\Full Time Applications\Prediction_Da

# 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"D:\DMA_Project\DMA_PROJECT_EXCEL_FILES\New_files\FINAL\Full Time Applications\Pred
```


	Customer_ID	Gender	Age	Married	State	Number_of_Referrals	\
0	93520-GUJ	Female	67	No	Gujarat	13	
1	57256-BIH	Female	18	No	Bihar	9	
2	72357-MAD	Female	53	No	Madhya Pradesh	14	
3	66612-KAR	Female	58	Yes	Karnataka	11	
4	22119-WES	Male	31	Yes	West Bengal	5	

	Tenure_in_Months	Value_Deal	Phone_Service	Multiple_Lines	...	\
0	19	Deal 5	Yes	Yes	...	
1	7	None	Yes	No	...	
2	12	Deal 5	Yes	No	...	
3	18	None	Yes	No	...	
4	5	None	Yes	No	...	

	Payment_Method	Monthly_Charge	Total_Charges	Total_Refunds	\
0	Bank Withdrawal	72.10	72.1	0.0	
1	Credit Card	19.85	57.2	0.0	
2	Credit Card	44.30	44.3	0.0	
3	Credit Card	19.95	58.0	0.0	
4	Credit Card	20.05	33.7	0.0	

	Total_Extra_Data_Charges	Total_Long_Distance_Charges	Total_Revenue	\
0	0	7.77	79.87	
1	0	9.36	66.56	
2	0	42.95	87.25	
3	0	8.07	66.07	
4	0	3.62	37.32	

	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 []: ▶

In []: ▶