```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Load data
data = pd.read_csv() # Replace with your path
print(data.head())

# 2. Preprocess
# Drop customer ID or any non-informative columns
if 'customerID' in data.columns:
    data.drop('customerID', axis=1, inplace=True)

# Convert categorical columns to dummy variables
data = pd.get_dummies(data, drop_first=True)

# 3. Train-test split
X = data.drop('Churn_Yes', axis=1)  # Assuming 'Churn' was turned into dummy
y = data['Churn_Yes']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 4. Scale features
```

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# 5. Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# 6. Evaluate
y_pred = model.predict(X_test_scaled)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

# 7. Feature importance to uncover patterns
feature_importances = pd.DataFrame({
    'feature': X.columns,
    'importance': model.feature_importances_
}).sort_values(by='importance', ascending=False)

# Plot top 10 features
sns.barplot(x='importance', y='feature', data=feature_importances.head(10))
plt.title('Top 10 Important Features for Churn Prediction')
plt.show()
```