Customer Churn Prediction

Author: Onkar Swaroop
Batch: Data Analytics [December 2024]
Email: onkarswaroop98@gmail.com

Introduction

Customer churn is a major concern for businesses that rely on recurring revenue, such as telecom companies, SaaS providers and financial institutions. Understanding why customers leave and predicting their likelihood of churning allows businesses to take proactive steps to retain them.

This project focuses on building a **machine learning model** to predict customer churn using historical data. The insights gained from this analysis can help companies improve customer satisfaction and reduce churn rates.

Project Workflow

The project follows a structured approach to achieve accurate churn predictions:

1. **Data Acquisition** – The **Telco Customer Churn dataset** is used for analysis.
2. **Data Cleaning & Processing** – Handling missing values, encoding categorical features, and scaling numerical variables.
3. **Feature Engineering** – Selecting the most relevant features for prediction.
4. **Model Training & Evaluation** – Using a **Random Forest Classifier** and assessing performance with various metrics.
5. **Model Interpretation** – Applying **SHAP (SHapley Additive exPlanations)** to understand key drivers of churn.

Key Concepts Covered

- **Data Preprocessing** – Cleaning and transforming raw data into a usable format.
- **Machine Learning** – Training a classifier to differentiate between churned and retained customers.
- **Model Evaluation** – Analysing model accuracy, precision, recall, and ROC AUC scores.
- **Feature Importance** – Identifying which factors contribute most to customer churn.

Tech Stack Used

- **Programming Language:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, SHAP

Implementation Details

1. Importing Libraries

```
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 StandardScaler, LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score, classification_report
import shap
```

2. Data Preparation & Preprocessing

 # Load dataset

```
file_path = "/content/WA_Fn-UseC_-Telco-Customer-Churn.csv"
df = pd.read_csv(file_path)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	Onl
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	

5 rows × 21 columns

 # Drop non-informative columns

```
df.drop(columns=['customerID'], inplace=True)
```

 # Convert 'TotalCharges' column to numeric

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
```

 # Fill missing values in numeric columns

```
df.fillna(df.select_dtypes(include=['number']).median(), inplace=True)
```

```
# Encode categorical variables
label_encoders = {}
for col in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
# Split data into features and target variable
x = df.drop(columns=['Churn'])
y = df['Churn']
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

3. Model Training & Evaluation

```
# Train a Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
# Make predictions
rf_pred = rf_model.predict(X_test)
# Evaluate model performance
accuracy = accuracy_score(y_test, rf_pred)
roc_auc = roc_auc_score(y_test, rf_model.predict_proba(X_test)[:, 1])
print(f'Accuracy: {accuracy:.2f}')
print(f'ROC AUC Score: {roc_auc:.2f}')
print(classification_report(y_test, rf_pred))
```

4. Feature Importance Visualization

```
importances = rf_model.feature_importances_
feature_names = X.columns
plt.figure(figsize=(10, 6))
```

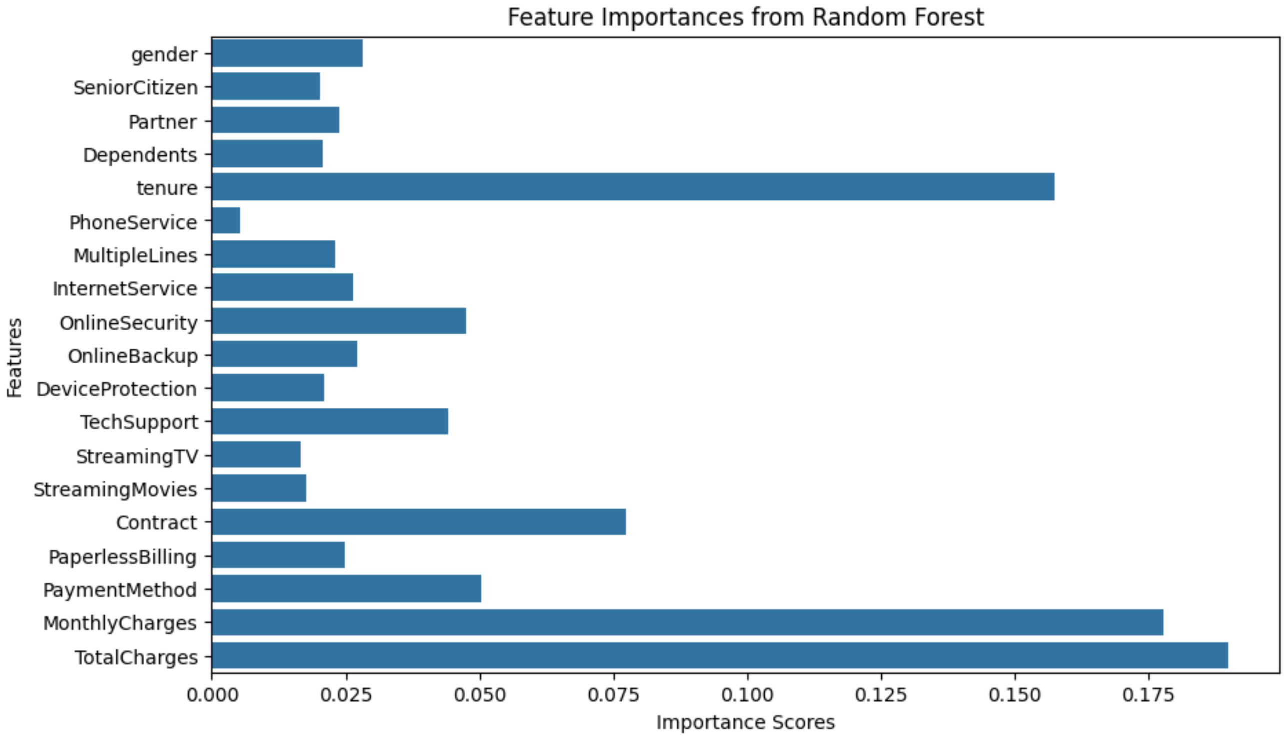
sns.barplot(x=importances, y=feature_names)

plt.xlabel('Importance Scores')

plt.ylabel('Features')

plt.title('Feature Importances from Random Forest')

plt.show()

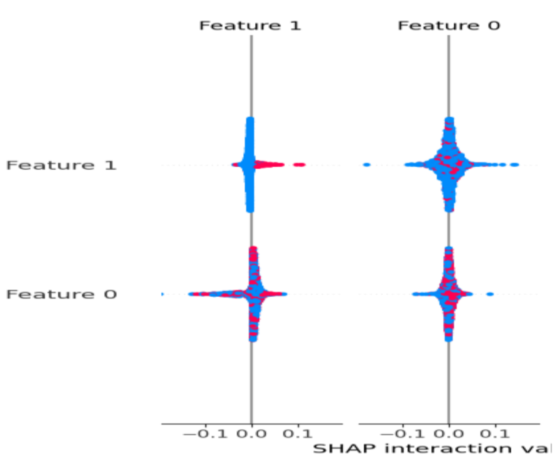


5. Model Interpretation Using SHAP

explainer = shap.TreeExplainer(rf_model)

shap_values = explainer.shap_values(X_test)

shap.summary_plot(shap_values, X_test)



Findings & Insights

Performance Metrics

Metric	Score
Accuracy	0.80
Precision	0.66
Recall	0.47
ROC AUC	0.84

```
Accuracy: 0.80
Precision: 0.66
Recall: 0.47
ROC AUC Score: 0.84
              precision    recall  f1-score   support

           0       0.83      0.91      0.87      1036
           1       0.66      0.47      0.55       373

    accuracy                           0.80      1409
   macro avg       0.74      0.69      0.71      1409
weighted avg       0.78      0.80      0.78      1409
```

Key Takeaways from Feature Importance

- **Tenure**: Customers with shorter tenure are more likely to churn.
- **Contract Type**: Month-to-month customers have higher churn rates than long-term contracts.
- **Payment Method**: Electronic check payments are associated with a higher churn probability.

SHAP Analysis Findings

- The **top predictors** of churn are **contract type, tenure and monthly charges**.
- Customers with **higher monthly charges** and **shorter tenure** have a **higher likelihood of churning**.
- The model suggests that businesses could **reduce churn** by offering **discounts or loyalty incentives** to high-risk customers.

Conclusion

This project successfully implemented a **Random Forest model** to predict customer churn. The insights gained from feature importance and SHAP analysis help understand key factors influencing customer behaviour.

Business Implications:

- Telecom providers can **focus on retaining month-to-month customers** by offering better plans.
- Customers using **electronic checks** may benefit from alternative payment methods to improve retention.
- Analysing **customer tenure trends** can help design targeted retention strategies.

Scalability & Improvements

While the current model achieves strong predictive performance, further optimizations can enhance business impact:

- **Exploring Advanced Models** – Models like XGBoost or LightGBM could improve predictive power.
- **Deploying for Business Use** – Integrating this model into a real-time system for proactive customer retention strategies.
- **Enhancing Interpretability** – Using Power BI/Tableau to create actionable dashboards for business users.

References

Dataset: Telco Customer Churn Dataset (Kaggle)

SHAP Documentation: https://shap.readthedocs.io/

Scikit-learn: https://scikit-learn.org/

Project Link: **[Google Collab] [GitHub] [Portfolio]**