Predicting Customer Churn Using SEMMA Methodology: A Practical Approach

Jayanth Kalyanam
San Jose State University
jayanth.kalyanam@sjsu.edu

October 18, 2024

1 Introduction

Predicting customer churn is a critical business problem for companies, particularly those with subscription-based services. In this project, we use the Telco Customer Churn dataset from Kaggle to build predictive models using the SEMMA methodology: Sample, Explore, Modify, Model, and Assess. We implement various machine learning models to predict customer churn, including Logistic Regression, Random Forest, SVM, and XGBoost.

2 Methodology

2.1 1. Sample

We start by loading the Telco Customer Churn dataset and sampling 50% of the data to manage computational resources.

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Load the dataset
df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')

# Sample 50% of the dataset
sampled_df, _ = train_test_split(df, test_size=0.5, random_state=42)
```

2.2 2. Explore

Next, we perform exploratory data analysis (EDA) by visualizing the distribution of the target variable (Churn) and the correlation matrix.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Visualize the distribution of churn
sns.countplot(data=sampled_df, x='Churn')
plt.show()

# Correlation matrix
numeric_columns = sampled_df.select_dtypes(include=['int64', 'float64']).columns
plt.figure(figsize=(12,8))
sns.heatmap(sampled_df[numeric_columns].corr(), annot=True, cmap='coolwarm', fmt=
plt.title('Correlation Matrix of Numerical Features')
plt.show()
```

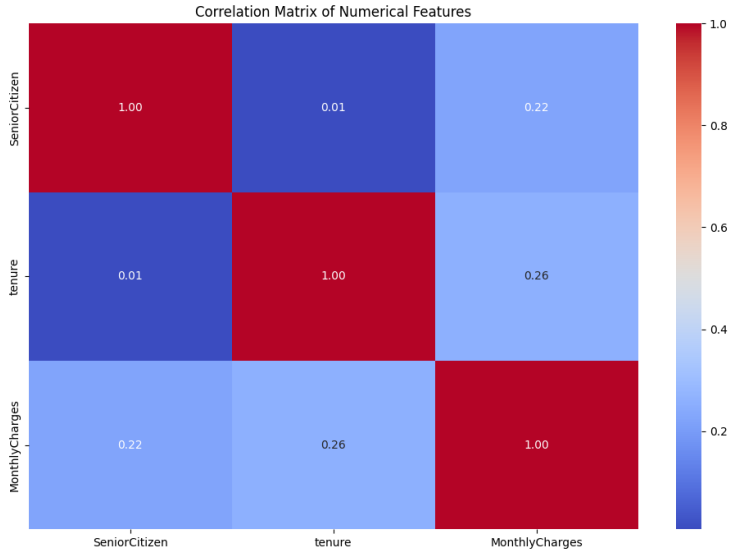

Figure 1: Correlation Matrix of Numerical Features

2.3 3. Modify

We clean the data by handling missing values and encoding categorical features using one-hot encoding. We also scale the numerical features.

```
from sklearn.preprocessing import StandardScaler

# Handle missing values and encode categorical features
sampled_df = pd.get_dummies(sampled_df, drop_first=True)

# Standardize numerical features
```

```
scaler = StandardScaler()
numeric_columns = sampled_df.select_dtypes(include=['float64', 'int64']).columns
sampled_df[numeric_columns] = scaler.fit_transform(sampled_df[numeric_columns])
```

2.4 4. Model

We train various machine learning models, including Logistic Regression, Random Forest, SVM, and XGBoost. Below is the code for Random Forest.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Split the dataset into training and test sets
X = sampled_df.drop('Churn_Yes', axis=1)
y = sampled_df['Churn_Yes']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random

# Train a Random Forest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions and evaluate the model
y_pred_rf = rf_model.predict(X_test)
accuracy_rf = accuracy_score(y_test, y_pred_rf)
print(f'Random Forest Accuracy: {accuracy_rf:.4f}')
```

2.5 5. Assess

We evaluate the Random Forest model using a confusion matrix.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

# Confusion matrix for Random Forest
cm_rf = confusion_matrix(y_test, y_pred_rf)
sns.heatmap(cm_rf, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix for Random Forest')
plt.show()
```

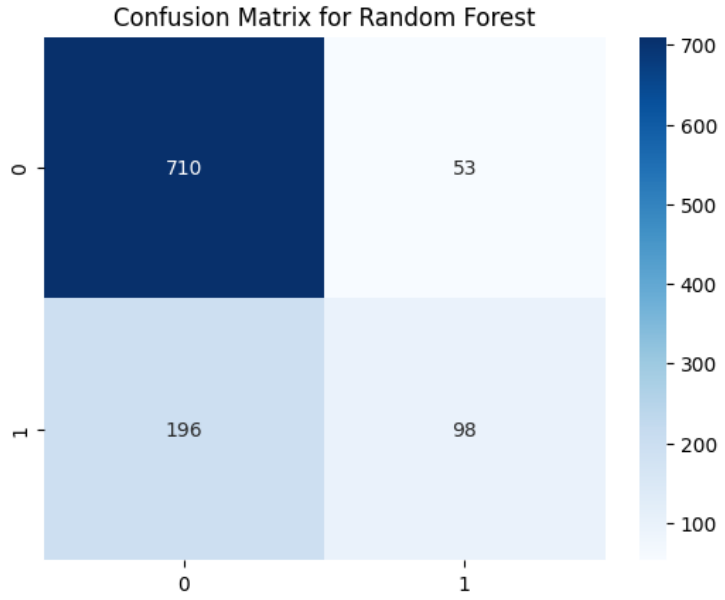

Figure 2: Confusion Matrix for Random Forest

3 Conclusion

Random Forest proved to be the best-performing model for predicting customer churn, offering an accuracy of 82.46% and stable cross-validation results. The SEMMA methodology allowed us to systematically approach this problem by carefully preparing, analyzing, and evaluating the dataset.

4 References

1. Berson, A., Smith, S., and Thearling, K. (2000). *Building Data Mining Applications for CRM*. McGraw-Hill.

2. Kaggle. (n.d.). Telco Customer Churn. Retrieved from https://www.kaggle.com/blastchar/telco-customer-churn