Predicting Customer Churn In Telecommunications Company

A person in a suit holding a net

Description automatically generated

M CHALANA AGNIHOTRI

27-05-2024

SPEAKX ONLINE ASSIGNMENT

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| 1 | Introduction | Pg.no 3 |
| 2 | Data Collection and Preprocessing | Pg.no 3 |
| 3 | Exploratory Data Analysis | Pg.no 4 |
| 4 | Feature Engineering | Pg.no 4 |
| 5 | Churn Prediction Model | Pg.no 4,5 |
| 6 | Cross Validation and Hyperparameter Tuning | Pg.no 5 |
| 7 | Final Model Comparision | Pg.no 6 |
| 8 | Conclusion | Pg.no 6 |
| 9 | Challenges Faced | Pg.no 6 |

# Introduction

In this project, we aim to predict customer churn in a telecommunications company using various machine learning models. Customer churn refers to the loss of clients or customers, and predicting churn helps the company take proactive measures to retain customers. This report details the data collection, preprocessing steps, exploratory data analysis (EDA), feature engineering, model building, evaluation, and the final conclusions drawn from the analysis.

Data Collection and Preprocessing

For Data Collection, imported Telco Customer Churn dataset from <https://www.kaggle.com/datasets/blastchar/telco-customer-churn> , a CSV file . It contained various attributes related to customers, their service usage, and whether or not they have churned.

**Data Preprocessing:**

Initial Data Overview

1. Shape of the data: Checked the shape of the dataset to understand the number of rows and columns.
2. Basic Information: The dataset's basic information, including data types and non-null counts, was examined.
3. Summary Statistics: Then generated summary statistics for all the columns to get an overview of the data distribution.

Handling Missing Values

Checked for missing values using df.isnull().sum() and df.isna().sum(). There were no missing values in the dataset.

Handling Duplicates

Checked for duplicate rows and found that there were no duplicates in the dataset.

Identifying Categorical and Numerical Columns

Categorical Columns: Columns with object data types were identified as categorical.

Numerical Columns: Columns with numeric data types were identified as numerical.

Encoding Categorical Variables

We encoded categorical variables using Label Encoding to convert them into a format suitable for machine learning algorithms.

# Exploratory Data Analysis

1. Distribution of Churn

We plotted the distribution of churn using a count plot to understand the proportion of churned vs. non-churned customers.

1. Correlation Matrix

A heatmap of the correlation matrix was plotted to identify the relationships between numerical features.

1. Pair Plot

We created a pair plot of selected features (tenure, MonthlyCharges, TotalCharges, and Churn) to visualize the relationships and distributions.

1. Distribution of Tenure

A histogram was plotted to show the distribution of the tenure feature.

1. Boxplot for MonthlyCharges by Churn

A boxplot was created to compare the distribution of MonthlyCharges for churned and non-churned customers.

1. Distribution of Tenure Group

We created a new feature, tenure\_group, based on the tenure duration, and plotted its distribution.

# Feature Engineering

We created a new feature called tenure\_group by segmenting the tenure feature into groups representing different ranges of tenure. This was encoded using Label Encoding.

# Churn Prediction Model

First, the dataset was split into features (X) and the target variable (y). Further, the data was divided into training and testing sets using an 80-20 split. Then used various machine learning models to predict churn and evaluated their performance using accuracy, precision, recall, and F1 score.

1. Logistic Regression

Initial Logistic Regression model showed an accuracy of 79.5%, precision of 57.1%, recall of 63.3%, and F1 score of 60.0%.

1. Decision Tree Classifier

The Decision Tree model had an accuracy of 73.8%, precision of 53.7%, recall of 51.4%, and F1 score of 52.5%.

1. Random Forest Classifier

The Random Forest model achieved an accuracy of 77.9%, precision of 48.7%, recall of 61.5%, and F1 score of 54.3%.

1. XGBoost Classifier

The XGBoost model resulted in an accuracy of 76.7%, precision of 49.7%, recall of 58.0%, and F1 score of 53.5%.

Confusion matrices were plotted for each model to visualize the true positives, true negatives, false positives, and false negatives.

# Cross-Validation and Hyperparameter Tuning

To improve the model performance, we performed cross-validation and hyperparameter tuning for each classifier.

* Logistic Regression

After tuning, the optimal Logistic Regression model showed a slight improvement with an accuracy of 79.6%, precision of 55.3%, recall of 64.0%, and F1 score of 59.3%.

* Decision Tree Classifier

The optimized Decision Tree model showed an accuracy of 79.9%, precision of 49.2%, recall of 67.5%, and F1 score of 56.9%.

* Random Forest Classifier

The optimized Random Forest model achieved an accuracy of 79.7%, precision of 47.1%, recall of 67.8%, and F1 score of 55.6%.

* XGBoost Classifier

The optimized XGBoost model resulted in an accuracy of 77.9%, precision of 49.2%, recall of 61.1%, and F1 score of 54.5%.

Final Model Comparison

We compared all the models, including the baseline and optimized versions, to determine the best-performing model. The following table summarizes the results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASSIFIER | TEST ACCURACY | PRECISION | RECALL | F1 SCORE |
| Logistic Regression | 0.795 | 0.571 | 0.633 | 0.600 |
| Decision Tree | 0.738 | 0.537 | 0.514 | 0.525 |
| Random Forest | 0.779 | 0.487 | 0.615 | 0.543 |
| XGBoost | 0.767 | 0.497 | 0.580 | 0.535 |
| Optimal Logistic Regression | 0.796 | 0.553 | 0.640 | 0.593 |
| Optimal Decision Tree | 0.799 | 0.492 | 0.675 | 0.569 |
| Optimal Random Forest | 0.797 | 0.471 | 0.678 | 0.556 |
| Optimal XGBoost | 0.779 | 0.492 | 0.611 | 0.545 |

# Conclusion

From the baseline models, Logistic Regression showed the highest test accuracy and F1 score. After cross-validation and hyperparameter tuning, the Decision Tree model achieved the highest test accuracy of 79.9%. Overall, cross-validation and hyperparameter tuning significantly reduced the chances of overfitting and improved model performance.

# Challenges Faced

1. Handling class imbalance: The dataset had an imbalance between churned and non-churned customers, which affected model performance.
2. Feature encoding: Converting categorical variables into numerical formats without losing information was crucial.
3. Hyperparameter tuning: Finding the optimal hyperparameters for each model was time-consuming but necessary for improving model accuracy.