

Kichimbi / DSFPT04AP3\_Project

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Set

0 stars

0 forks


1 watching

Activity

Public repository

main

BranchesTags

 Kichimbi A readme final change code update ...

now 11

[View code](#)

README.md

# Overview

This data science project focuses on building a binary classifier to predict customer churn for Syria Telecom, a telecommunications company. The primary objective is to reduce revenue loss by identifying customers who are likely to discontinue their services. The project will explore patterns within the available data to make proactive business decisions.

## Business Understanding

- Stakeholders: The primary stakeholders include Syria Telecom's management, customer service teams, and marketing departments. They will be directly affected by the outcomes of this project.
- Business Problem: The project aims to solve the problem of customer churn by identifying potential churners in advance, allowing the company to take targeted retention actions.
- Scope: This project will focus on building a predictive model for customer churn. It is within the scope of the project to gather, preprocess, and analyze relevant data to construct this model.

## Data Understanding

- Data Sources: *The data is obtained from Syria Telecom's and accessed through Kegel.*
- Data Access: *The data is publicly available for use by interested parties and will therefore require no authorizations.*
- Target Variable: *The target variable is "churn," which is binary (1 for churn, 0 for non-churn).*
- Predictors: *The predictors include customer account length, international plan status, voice mail plan status, number of voice mail messages, usage statistics for day, evening, and night, international usage statistics, and the number of customer service calls made.*
- Data Types: *The data includes numeric and categorical features.*

- Data Distribution: *The distribution of the data for each variable will be explored during data analysis.*
- Data Volume: *The dataset contains 21 features and 3333 records, sufficient for building a predictive model.*
- Data Quality: *Data quality will be verified during the preprocessing phase, and measures will be taken to address any potential issues.*

## Stakeholder audience choice [↗](#)

---

The primary stakeholders in this project are the management, customer service teams, and marketing departments within Syria Telecom. Their collaboration and input are crucial for the project's success.

## Dataset choice [↗](#)

---

The dataset used for this project is sourced from Syria Telecom through Keggale. It contains historical customer data with relevant features for the churn prediction.

## Modeling [↗](#)

---

The modeling phase of this project will involve the development of three different models, each with varying complexity and sophistication. These models will help us effectively predict customer churn for Syria Telecom.

### Simple Baseline Model (Logistic Regression): [↗](#)

---

In order to establish a baseline for our predictive modeling, we will begin with a simple yet interpretable model. Logistic regression is a logical choice for this. It will help us understand the basic relationships between predictors and customer churn. We will assess the model's performance using standard evaluation metrics such as accuracy, precision, recall, and the ROC-AUC score.

### More-Complex Model (Random Forest): [↗](#)

---

To capture more complex patterns and interactions within the data, we will implement a random forest classifier. Random forests are an ensemble method that can handle both categorical and numeric features efficiently. This model will be evaluated using the same metrics as the baseline model to compare performance.

### Tuned Hyperparameter Model (Random Forest with Tuned Hyperparameters): [↗](#)

---

The third model will be a refined version of the random forest model. We will use hyperparameter tuning techniques to optimize its performance. This will involve adjusting parameters such as the number of trees, maximum depth, and minimum samples required to split nodes.

Cross-validation and grid search will be used to identify the best hyperparameters. We will compare the performance of the tuned model with the baseline and initial random forest models to assess the impact of hyperparameter tuning.

Throughout the modeling phase, model interpretability will be considered. Interpretable models are valuable for understanding which features have the most influence on predicting customer churn. This information can be used by stakeholders to make informed decisions on customer retention strategies.

The final model selected for deployment will be the one that demonstrates the best performance in terms of accurately predicting customer churn. Its predictive power, along with its interpretability, will assist Syria Telecom in identifying at-risk customers and implementing proactive measures to reduce churn and improve customer retention.

### **Model Building and Evaluation Steps:**

#### **Train-Test Split:**

- Begin by splitting the dataset into training and testing sets. This will enable us to train and evaluate our models on different data subsets to assess their performance.

#### **Baseline Model:**

- Build a simple, interpretable baseline model, such as logistic regression, using the training data. This will serve as a reference for model performance.
- Evaluate the baseline model's performance using standard metrics like accuracy, precision, recall, and ROC-AUC.

#### **Custom Cross-Validation Function:**

- Develop a custom cross-validation function that will allow us to assess model performance more robustly. This function should take care of splitting the data, training, and evaluation within a cross-validation loop.

#### **StratifiedKFold for Splitting:**

- Utilize StratifiedKFold, a type of cross-validation strategy, to provide information for creating separate training and validation splits within the training dataset. StratifiedKFold ensures that the class distribution remains balanced in each fold.

#### **Custom Cross-Validation with StratifiedKFold:**

- Implement the custom cross-validation function with StratifiedKFold to train and evaluate the baseline model and later models within a cross-validation framework.
- Compare the model's performance using custom cross-validation with the baseline log loss to assess any improvements.

#### **Choosing and Evaluating a Final Model:**

- Evaluate the performance of different models, including the baseline model, more complex models (e.g., random forest), and tuned models.
- Select the model that exhibits the best performance based on evaluation metrics.

#### **Fitting the Final Model:**

- Once the final model is chosen, fit it on the full training dataset. This will enable the model to learn from all the available training data.

#### **Evaluating on Test Data:**

- Assess the model's performance on the test dataset. This step provides a realistic evaluation of how the model will perform in real-world scenarios.

Throughout these steps, attention was given to evaluation metrics, model interpretability, and the ability to generalize to unseen data. The final model selected would demonstrate improved predictive power and contribute to Syria Telecom's efforts in reducing customer churn and improving customer retention.

## Evaluation [↗](#)

The dataset used for this project is sourced from Syria Telecom's internal database, which contains historical customer data with relevant features for the churn prediction task.

## Conclusion [↗](#)

This project aims to provide Syria Telecom with a predictive tool to identify customers at risk of churning. By addressing this issue proactively, the company can implement strategies to retain customers and reduce financial losses. Successful implementation of the predictive model will empower Syria Telecom to make data-driven decisions, ultimately improving customer retention and business profitability.

### Releases

No releases published  
[Create a new release](#)

### Packages

No packages published  
[Publish your first package](#)

### Languages

- Jupyter Notebook 100.0%