

 **Julez89 / dsc-phase-3-project-v2-3** Public

forked from [learn-co-curriculum/dsc-phase-3-project-v2-3](#)


☆ 0 stars


 33 forks


☆ Star


 Watch


<> Code


 Pull requests


 Actions


 Projects

 Wiki

 Security


 Insights


 Settings


 main ▾


...

This branch is **7 commits ahead** of learn-co-curriculum:main.



 Contribute ▾

 Sync fork ▾

 Julez89 ... 


1 minute ago 

[View code](#)

 README.md 

# Phase 3 Project Description

## CUSTOMER CHURN



## Project Overview

For this project, I tried different classification methods to predict customer churn for a telephone company. My final model uses a gradient boosting algorithm after addressing the class imbalance by a combination of over and undersampling.

## Business Problem & Stakeholder

<https://github.com/Julez89/dsc-phase-3-project-v2-3/tree/main>

1/6

A US-based telephone company wants to reduce their customer churn rate. Currently, 15% of their customers leave which is resulting in higher marketing costs to win new customers. At the moment, they randomly offer discounts or other perks to some customers without knowing if they were planning to leave or not. They wish to identify customers ahead of time to change their mind and keep them as customers. On the other hand, they do not want to spend money on customers who did not even plan to leave. Questions that we want to answers are:

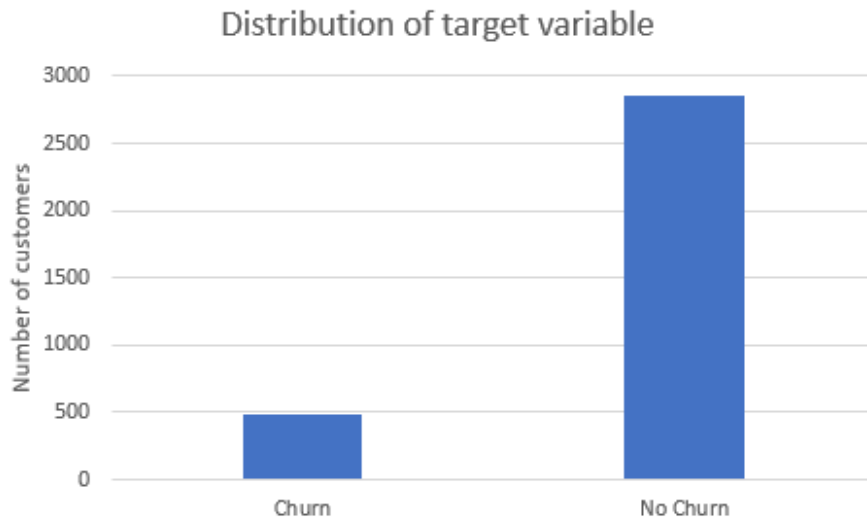
What are the most important features of a customer that can predict customer churn? Will I be able to correctly identify customers who are planning to leave? Can I minimize offering customers discounts who do not want to leave?

## The Data

This project uses a customer churn data set. The data did not include any missing data but we removed certain information such as the telephone number or the area in the United States. Each row represents a customer from the phone company and information about their call behavior and contract. In our model, we will look at the following information:

- International Plan (yes/no)
- Voice Mail Plan (yes/no)
- Calls to Customer Service
- Length of contract
- Total day minutes
- Total day calls
- Total evening minutes
- Total evening calls
- Total night minutes
- Total night calls
- Total international minutes
- Total international calls
- Number of voice mail messages

Regarding my target variable (churn), we are dealing with class imbalance. 85% of the data set contain information about staying customers and only 15% churn.



## Approach

This task is a classification problem because my target variable is a categorical variable. It is either yes or no for a customer that churns. I used a machine learning approach to solve the classification task. Machine learning algorithms are designed to automatically identify patterns and relationships within data, allowing them to detect complex relationships that may not be immediately apparent through manual analysis. Also, machine learning algorithms can improve over time as they are trained on more data, providing an ongoing opportunity to refine and improve the accuracy of the classification model. Finally, they are able to handle large amounts of data efficiently and quickly, making them an ideal choice for tasks involving large datasets. I am using an iterative approach to try to find the best model for my predictions. In summary, I took the following steps:

- Loading and combining data
- Check for data completeness and overall structure
- Inspect all variables visually to determine relevance for future model
- Prepare data for modelling (one hot encoding, scaling, train - test split, resampling)
- Start with a vanilla, simple logistic regression model
- Try a more complex model with random forest classification and tune it
- Try an ensemble method using gradient descent algorithm.
- Evaluate which model works best and determine the most important features to be able to give practical recommendations to my customer.

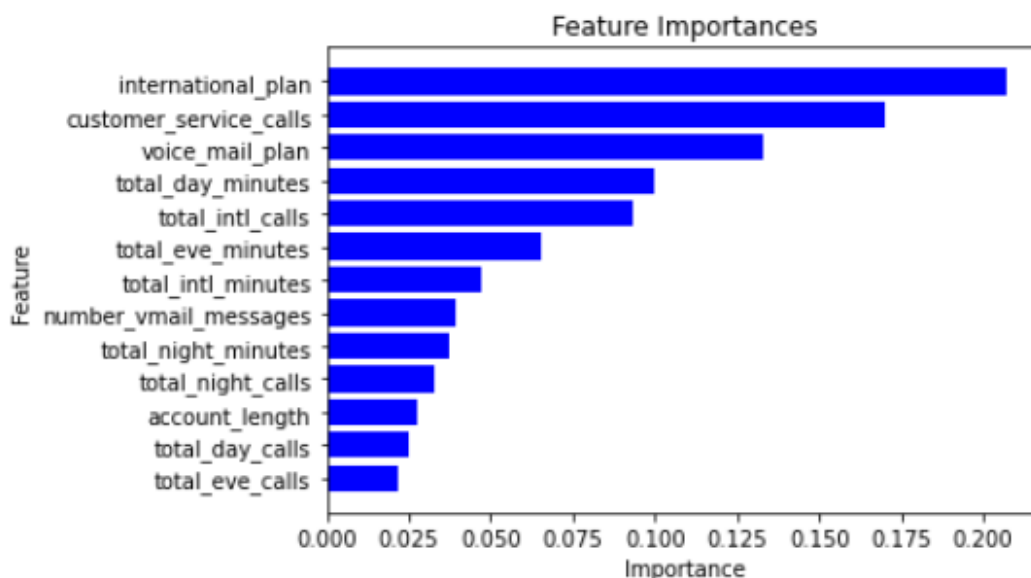
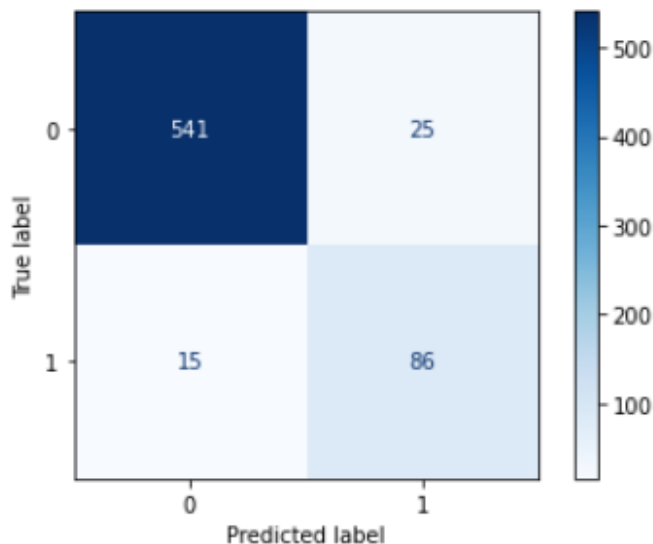
## Modeling

For my customer, my main priority is to be able to identify customers who are likely to leave. Both letting a customer go and spending money on customers who are not actually planning to leave result in costs for my stakeholder that could be reduced. However, losing a customer is much more costly than overspending a bit on staying customers. Therefore, my model should reduce false negatives (missing the identification of a leaving customer) and I will focus on a high recall score while accepting lower precision score. My preparation steps included the standardization of data and oversampling using SMOTE. To follow an iterative approach, I am starting with a very simple, vanilla logistic regression model which did not perform too well (Recall = 0.79). The accuracy on training and testing data was similar (0.77 and 0.79) which shows that it was neither under nor overfitting. As next model, I am trying a Random Forest model with the standard hyperparameters. This model already improved (Recall=0.81). Also the accuracy improved a lot even though the accuracy of 1 for training is a bit suspicious. The accuracy of test data is 0.94 so it's only slightly overfitting and overall performing quite well. The adjustment of hyperparameters for my random forest model did not improve the model. Next, I am trying an ensemble method with bagging and XGB Classifier. The main difference between BaggingClassifier and XGBClassifier is in their approach to ensemble learning. BaggingClassifier creates multiple samples of the training data and trains base classifiers on each sample, while XGBClassifier uses gradient boosting to iteratively improve the prediction accuracy. The bagging classifier showed worse metrics than the random forest model. The XGBClassifier worked fairly well. It had a high precision (0.91) but a bit lower recall (0.78). Since I want get a higher recall, I tried different strategies to come to a better core. First, I tried different hyperparameters, then I tried using a lower threshold (0.3 instead of default 0.5). Finally, I tried a different sampling method than Smote that worked better. It was a combination of oversampling and undersampling. I applied my slightly tuned XGBClassifier model without adjusting the threshold.

Model	Accuracy Train	Accuracy Test	Precision	Recall	F1
Log Reg	0,77	0,79	0,4	0,79	0,53
Random Forest vanilla	1	0,94	0,83	0,81	0,82
Random Forest tuned	1	0,94	0,82	0,8	0,81
Bagging	0,99	0,92	0,77	0,76	0,77
XGB vanilla	1	0,955	0,91	0,78	0,84
XGB vanilla threshold	1	0,955	0,84	0,83	0,84
XGB tuned threshold	0,99	0,955	0,83	0,83	0,83
xgb tuned sampling	1	0,956	0,77	0,85	0,81

## Evaluation

My final model is a XGB Classifier with slightly tuned hyperparameters but default threshold. (learning\_rate=0.1, max\_depth=5, n\_estimators=300) after addressing class imbalance with oversampling and undersampling instead of SMOTE. The combination of oversampling of the minority and undersampling of majority has lead to better results than the approach with SMOTE. The XGB Classifier model has performed well in predicting customer churn. The precision score for class 1 (churn) is 0.77, which means that when the model predicts a customer will churn, it is correct 77% of the time. The recall score for class 1 is 0.85, which means that the model correctly identifies 85% of all customers who actually churn. The F1-score for class 1 is 0.81, which is a harmonic mean of precision and recall and indicates overall performance of the model for class 1. Additionally, the macro-average F1-score is 0.91, which means that the model performs well in both classes. The accuracy of the model on the test set is 0.9565, which means that the model correctly predicts 95.65% of all customers' churn status. The high accuracy on the training set (1.0) and a comparable accuracy on the test set (0.9565) suggests that the model has not overfit the training data.



Based on the feature importance chart, the three most important features for predicting customer churn are `international_plan`, `voice_mail_plan`, and `customer_service_calls`. This suggests that customers who have an international plan are more likely to churn, as are those with a voice mail plan. In addition, customers who have made a larger number of customer service calls are also more likely to churn.

## Summary & Recommendation

---

With the model, I have identified several factors that are important in determining the churn of the customer. While the model is not perfect, it will mostly catch those customers who are likely to leave. With this information, the telephone company can take steps to reduce churn and retain customers. For example, they could investigate whether their international plans are meeting the needs of their customers or if there are alternative plans that may be more suitable. The company could also evaluate the importance of offering voice mail plans to customers, and consider the impact on churn rates if such plans are not offered. Finally, they may want to review their customer service processes and practices to identify areas that could be improved to reduce the number of customer service calls and, in turn, decrease the likelihood of churn. We would have to monitor the performance of the model when we introduce more data to see if it's overfitting with more data or not.

## This repository

---

My technical code is stored in this [jupyter notebook](#)

My presentation can be found [here](#)

My github repository is [here](#)

---

### Releases

No releases published

[Create a new release](#)

---

### Packages

No packages published

[Publish your first package](#)

---

### Languages

● Jupyter Notebook 100.0%