# Model Selection for Customer Churn Prediction in the Telecommunications Industry

Adam Cooke, Anthony Gheen, Nitin Shirsat

University of North Carolina Chapel Hill

July 29, 2025

# Overview

# Introduction

# Background

- Telecommunications industry has high costs associated to gaining new customers/subscribers
- Construction, permits, regulatory agencies can make it difficult to expand the network and provide services to new clients
- Churn is defined as the termination of a customer's subscription to services
- The ability to identify customers at risk of Churn is valuable to any business operating on a subscription model

# Purpose

- The purpose of this report to perform a comprehensive comparison of common models used to predict customer churn in the telecommunications industry
- We will train and compare multiple combinations of the below list:
  - Sampling Technique
  - Scaling Technique
  - Model
- We aim to identify the combination that produces the best results and apply that to a new dataset to see if that combination is applicable beyond one Dataset
- After model selection, we will use Shapley Additive explanation (SHAP) Analysis to identify the impact each feature had in the model's prediction

# Process Outline

- We will train and compare possible combinations of various scaling and sampling techniques. The best combinations for each classification model type are chosen by best F1 scores.
- We will then perform hyper-parameter tuning using GridSearch on various hyper-parameters for each classification model type using the optimal scaling and sampling techniques found in the previous step. The best hyper-parameter combinations for each classification model type are chosen by the best F1 scores.
- We will then apply the the models consisting of the optimal scaling, and sampling techniques, and optimized hyper-parameters to the original dataset, a new dataset with additional engineered features, a different telecommunications dataset, and a banking customer churn dataset. This is to get a grasp on how well the models perform with engineered features, features from a similar industry, and features from a completely different industry.
- After each application we will gather performance metrics and feature importances and compare results.

# Methods

# Models

- We will compare the following classification models:
  - Decision Tree Classifier
  - Logistic Regression
  - Gaussian Naive Bayes Classifier
  - Random Forest Classifier
  - Support Vector Machine
  - XGBoost Classifier

# Sampling Methods

- No Sampling
  - Maintain proportion of churners to non churners present in original dataset
- Random Undersampling
  - Removes instances of majority class from main dataset until desired proportion of non-churners to churners is met
- Random Oversampling
  - Duplicates instances of the minority class until desired proportion of non-churners to churners is met
- Sythetic Minority Oversampling Technique (SMOTE)
  - Generates synthetic instances of minority class to increase the volume of churn instances in the final dataset

# Scaling Methods

- Standard Scaler:

$$x_{StandardScaler} = \frac{x_i - \mu}{\sigma} \tag{1}$$

- Min-Max Scaler:

$$x_{MinMaxScaler} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2}$$

- Robust Scaler:

$$x_{Robust} = \frac{x_i - x_{Median}}{Q_3 - Q_1} \tag{3}$$

- No Scaling

# SHAP

- Shapley Additive Explanation (SHAP) provides a method for identifying the impact of each feature on the model's prediction

$$f_i = \phi_0 + \sum_{features} \phi_{i,j} \tag{4}$$

- We will perform SHAP analysis for each dataset. In doing so, we will be utilizing the optimized XGBoost Model to gather feature importances for all features for each dataset.
- We will rank each feature based on mean absolute SHAP values, and plot the signed mean SHAP values to indicate direction and strength of impact.
- This analysis is crucial for companies to determine the strongest indicators of a customer churning.

# Validation Metrics

- We will be looking at Precision, Recall, F1 Score, and area under the receiver operator characteristic curve
- Precision:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

- Recall:

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

- F1 score will be primarily used when choosing the best model because it will be low if either precision or recall is low

$$F_1 score = 2 * \frac{precision * recall}{precision + recall} \tag{7}$$

# Process

- We will compare all combinations of scaling technique, sampling technique, and model by performing the following the steps:
  1. Split the original dataframe into training dataset and testing dataset using a 5-fold randomly shuffled cross validation
  2. Determine scaling technique to be used and scale continuous variables
  3. Determine sampling method to be used and resample from training dataset
  4. Fit model using the training set and score based on test dataset

# Data

# Data

- We will primarily be using 4 datasets for this report.
- The first dataset will be used to compare all sampling/scaling method and model combinations and final model selection.
- The other three datasets include a new copy of the original dataset with various engineered features, another dataset from a similar industry, and a dataset from a different industry. All being used to predict customer churn.

# Dataset 1

- This dataset contains data related to a fictional telecommunications company created by IBM
- 7032 rows total, with 1869 rows containing customers who churned ( 27%)
- Categorical features were one-hot encoded

# Dataset 2

- Second dataset is a copy of the original dataset with the addition of 12 new engineered features
- These newly engineered features are created from binning techniques, and feature derivation
- 7032 rows total, with 1869 rows containing customers who churned ( 27%)
- Categorical features were one-hot encoded

# Dataset 3

- Third dataset is a publicly available dataset also from the telecommunications industry
- 64374 rows total, 30493 customers who churned
- 11 columns total
- Categorical features were one-hot encoded
- We will train models with the optimal sampling/scaling techniques identified using the first dataset to test if those combinations are applicable beyond the original dataset

# Dataset 4

- Fourth dataset is also a dataset found online from the banking industry
- We want to further test the sampling/scaling/model combinations beyond the telecommunications industry
- This dataset contains 10,000 rows, with 2,037 customers who've churned

# Model Selection and Tuning

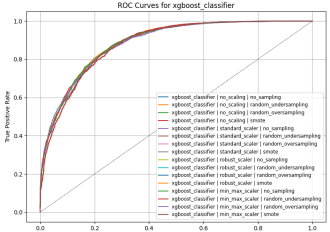
# Best Sampling and Scaling Combinations

- We tested a total of 96 combinations of model, scaling technique and sampling techniques
- The below table contains the combination for each model that led to the highest F1 score

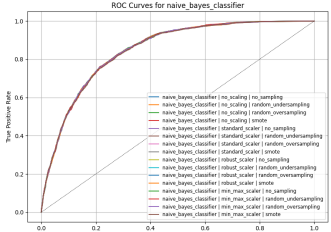
| **Classification Model** | **Scaling Technique** | **Sampling Technique** | **F1 Score** |
|---|---|---|---|
| XGBoost Classifier | No Scaling | Random Oversampling | 0.675115 |
| Random Forest Classifier | Standard Scaler | Random Undersampling | 0.673595 |
| Support Vector Classifier | Standard Scaler | Random Oversampling | 0.673592 |
| Logistic Regression | Standard Scaler | Random Oversampling | 0.668893 |
| Gaussian Naive Bayes Classifier | Standard Scaler | SMOTE | 0.642949 |
| Decision Tree Classifier | Standard Scaler | Random Undersampling | 0.588345 |

# ROC Curves

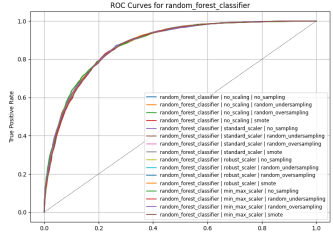
- Below are the ROC curves for all combinations of sampling and scaling for the XG Boost, Random Forest, and Naive Bayes models
- Not a significant amount of variance in the ROC curves when comparing sampling/scaling techniques




(a) XGBoost



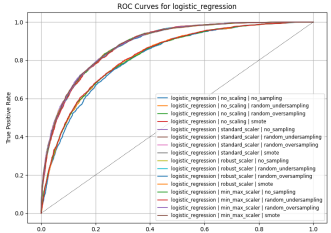

(b) Naive Bayes



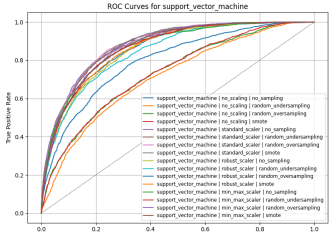

(c) Random Forest

# ROC Curves Continued

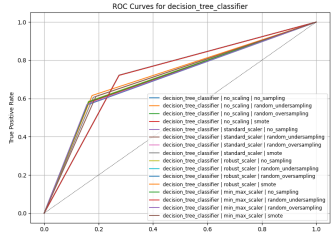
- Below are the ROC curves for all combinations of sampling and scaling for the Logistict Regression, SVM, and the Decision Tree models
- There is significant variation in the ROC curves, indicating that model performance is impacted by sampling and scaling technique used.




(d) Logistic Regression




(e) SVM




(f) Decision Tree

# Hyperparameter Selection

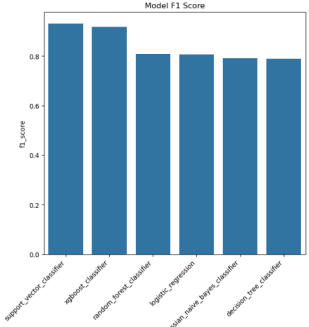
- After identifying the ideal scaling and sampling combinations, we moved on to hyperparameter tuning for each model
- We performed a similar iterative process to identify which hyperparameters led to the highest F1 scores for each model
- The final hyperparameters for each model are below, as well as the average of the validation metrics across a 5-fold cross validation split

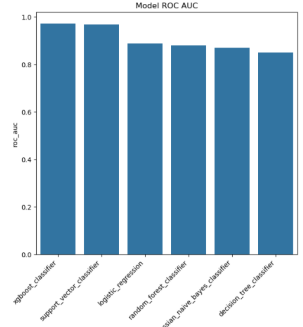
Table: Optimized hyper parameters selected based on the best average F1 Score

| Classification Model | Hyperparameters that maximize F1 Score | Accuracy | Precision | Recall | F1 Score | ROC AUC |
|---|---|---|---|---|---|---|
| Support Vector Classifier | {'C': 10, 'class_weight': None, 'gamma': 1.0, 'kernel': 'rbf'} | 0.933082 | 0.960004 | 0.903858 | 0.931057 | 0.967525 |
| XGBoost Classifier | {'colsample_bytree': 0.8, 'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 300, 'subsample': 0.8} | 0.914294 | 0.874552 | 0.96723 | 0.918555 | 0.971051 |
| Random Forest Classifier | {'class_weight': None, 'criterion': 'entropy', 'max_depth': 10, 'max_features': None, 'min_samples_split': 2, 'n_estimators': 300} | 0.796956 | 0.767085 | 0.854196 | 0.807968 | 0.878474 |
| Logistic Regression | {'C': 1.0, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'lbfgs'} | 0.799632 | 0.781263 | 0.832183 | 0.805904 | 0.887956 |
| Gaussian Naive Bayes Classifier | {'var_smoothing': 1e-09} | 0.787236 | 0.775824 | 0.808266 | 0.791667 | 0.8697 |
| Decision Tree Classifier | {'class_weight': None, 'criterion': 'gini', 'max_depth': 5, 'min_samples_split': 2} | 0.776082 | 0.744953 | 0.842526 | 0.789896 | 0.848311 |

# Final AUC and F1 Scores

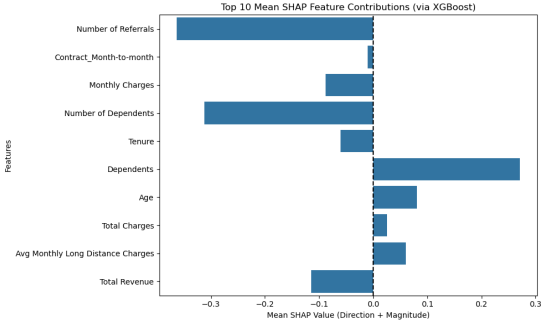
- Below contains graphs of the F1 scores and AUC for each of the 6 models we trained using the optimal scaling and sampling techniques using the hyperparameters detailed earlier
- XGBoost and Support Vector Classifier came in with the two highest AUC and F1 scores
- F1 score and AUC are significantly higher after hyperparameter tuning



# SHAP Feature Importance Analysis

- Below is a plot of the top 10 features sorted by mean absolute SHAP value (average magnitude of impact), regardless of direction.
- The length and direction of the bar indicate whether a feature increases or decreases the probability of churn and how strongly the feature affects that probability.
- For a negative mean SHAP value, higher values of that feature are associated with a lower probability of churn, on average (vice versa).



# Model Application to New Datasets

# Applying models to Dataset 2

- The below table details the results after training the optimized models on the feature engineering dataset.
- The metrics shown are the averages across 5 folds of a cross validation split
- Surprisingly, the overall performance for all models decreased when using engineered features. The support vector classifier experienced a rather extreme decrease in performance which will require further investigation.

| classification_model | scaling_technique | sampling_technique | accuracy | precision | recall | f1_score | log_loss | roc_auc |
|---|---|---|---|---|---|---|---|---|
| random_forest_classifier | standard_scaler | random_undersampling | 0.777302 | 0.552262 | 0.860853 | 0.672624 | 0.434098 | 0.884096 |
| logistic_regression | standard_scaler | random_oversampling | 0.777588 | 0.555098 | 0.821599 | 0.662332 | 0.433754 | 0.881479 |
| xgboost_classifier | no_scaling | random_oversampling | 0.817972 | 0.655546 | 0.667395 | 0.661047 | 0.453056 | 0.882279 |
| decision_tree_classifier | standard_scaler | random_undersampling | 0.747298 | 0.516246 | 0.846182 | 0.640395 | 0.610188 | 0.839351 |
| gaussian_naive_bayes_classifier | standard_scaler | smote | 0.778297 | 0.571152 | 0.687393 | 0.622214 | 2.851317 | 0.838566 |
| support_vector_classifier | standard_scaler | random_oversampling | 0.733077 | 0.459865 | 0.019299 | 0.036977 | 0.753589 | 0.764371 |

# Applying models to Dataset 3

- The below table details the results after training new models on a different telecommunications dataset.
- All models saw significant increases in performance.
- All models except Logistic Regression and Gaussian Naive Bayes had F1 scores over 0.94

| Classification Model | Scaling Technique | SamplingTechnique | Accuracy | Precision | Recall | F1 Score | Log Loss | ROC AUC |
|---|---|---|---|---|---|---|---|---|
| XGBoost Classifier | No Scaling | Random Oversampling | 0.999922 | 0.999869 | 0.999968 | 0.999918 | 0.000582 | 0.999992 |
| Random Forest Classifier | Standard Scaler | Random Undersampling | 0.998742 | 0.998261 | 0.999082 | 0.998671 | 0.004244 | 0.999978 |
| Decision Tree Classifier | Standard Scaler | Random Undersampling | 0.956613 | 0.930023 | 0.982346 | 0.955465 | 0.088584 | 0.99441 |
| Support Vector Classifier | Standard Scaler | Random Oversampling | 0.942228 | 0.934457 | 0.944284 | 0.939341 | 0.143388 | 0.987544 |
| Gaussian Naïve Bayes Classifier | Standard Scaler | SMOTE | 0.835151 | 0.800956 | 0.867575 | 0.832931 | 0.403325 | 0.908518 |
| Logisitic Regression | Standard Scaler | Random Oversampling | 0.825504 | 0.800325 | 0.84158 | 0.820427 | 0.393823 | 0.904097 |

# Applying models to Dataset 4

- Below table details the results after training and testing the models using a banking dataset
- F1 scores were significantly lower across the board then when using original dataset or dataset 2
- This could imply that the scaling/sampling and hyperparameter combinations identified using the first dataset are not necessarily widely applicable in general

| Classification Model | Scaling Technique | Sampling Techinque | Accuracy | Precision | Recall | F1 Score | Log Loss | ROC AUC |
|---|---|---|---|---|---|---|---|---|
| XG Boost Classifier | No Scaling | Random Oversampling | 0.8384 | 0.607759 | 0.584399 | 0.595701 | 0.422642 | 0.836589 |
| Random Forest Classifier | Standard Scaler | Random Undersampling | 0.7810 | 0.476573 | 0.756371 | 0.584504 | 0.474847 | 0.856955 |
| Decision Tree Classifier | Standard Scaler | Random Undersampling | 0.7449 | 0.430041 | 0.761398 | 0.548784 | 0.618796 | 0.830766 |
| Gaussian Naive Bayes Classifier | Standard Scaler | SMOTE | 0.7361 | 0.408822 | 0.661405 | 0.505227 | 0.549287 | 0.781643 |
| Logistic Regression | Standard Scaler | Random Undersampling | 0.7117 | 0.383714 | 0.687141 | 0.492409 | 0.577906 | 0.768511 |
| Support Vector Classifier | Standard Scaler | Random Undersampling | 0.7991 | 0.509257 | 0.361364 | 0.422674 | 0.698014 | 0.764247 |

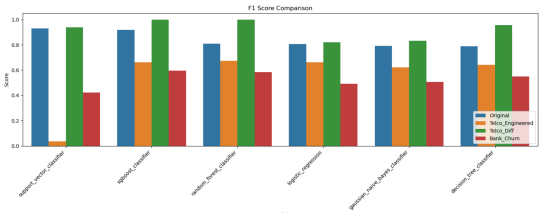
# Conclusion

# Findings

- For XG Boost, Random Forest, and Gaussian Naive Bayes models there was little variance in ROC Curves between different combinations of scaling and sampling methods
- For Logistic Regression, Support Vector Classifier, and Decision Tree Classifiers there was significant variation in ROC Curves between scaling and sampling combinations
- Hyperparameter optimization increased all performance metrics across all models, with an average increase in accuracy of (0.06), precision of (0.25), recall of (0.08), F1 score of (0.19), and ROC AUC score of (0.05).
- The support vector machine classifier saw the most significant increase in F1 score after hyper-parameter tuning (increasing by 0.26), resulting in the highest F1 score across all classification models (0.93).

# Findings Continued

- After applying the model to 3 new datasets, it appears that the configuration combinations we optimized for may not always lead to high F1 scores
- Surprisingly, all models experienced a decrease in F1 score on the engineered features dataset.
- Dataset 3 with different telecommunications data achieved higher F1 scores when using the optimized configurations. On a different note, the models all experienced a decrease in F1 scores in the banking customer churn dataset.
- These combinations may not be as applicable to other industries or datasets, but the process itself may be used to identify the best configuration combinations for a specific dataset.



# The End