Introduction

Customer Churn Modeling

Connex Communications

Connex is a communications company that provides phone, internet, and streaming services. In addition, they offer supportive services like online security, device protection, etc.

Connex would like to utilize machine learning methodologies to create a model that gives them the ability to get ahead of customer churn.

Key Questions

What model can best predict churn?

Test and optimize a variety of machine learning algorithms

Use product subscription status, account, and demographic information as inputs into the modeling process

Which features (inputs) are more likely to predict churn?

Utilize logistic regression feature coefficients and the feature_importances_ tool in sklearn's random forest

Process

Getting to Answers

The Data

Kaggle Telco Dataset

7,032 observations

30 Features

Products

- Phone
- Multiple Lines
- Internet
- Online Security
- Online Backup
- Device Protection
- Tech Support
- Streaming TV
- Streaming Movie

Account Info

- Time as Customer
- Contract Type
- Payment Method
- Paperless Billing
- Monthly Charges
- Total Charges

Demographics

- Senior
- Dependents
- Gender
- Partner

Churn

Last month

- Yes
- No

Approach

EDA + Benchmark

- Logistic Regression
- Evaluate key metrics
- Address class imbalance

Test More Models

- Decision Trees, Random Forest, XGBoost
- Tune hyperparameters

Determine Winner

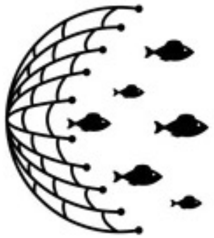
- Best performing model
- Most important features

Identify Further Optimizations

- Identify further tuning opportunities
- Explore other ideas

Tools: Pandas – Sklearn – Imblearn - Numpy

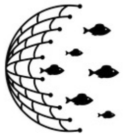
Which metric should we use to optimize the model?



Recall

It’s better for the business to cast the net wider and capture more potential churn customers than to use precision metric and miss potential churn customers.

Recall



$$\frac{\text{\# correctly classified as churn}}{\text{\# of actual churn in dataset}}$$

Precision

$$\frac{\text{\# of actual churn in dataset}}{\text{\# correctly classified as churn}}$$

F1

The harmonic mean of precision and recall. Penalize situations where precision or recall is significantly better than other

Results

Model Performance

Model Settings:
- Default hyperparameters
- Oversampling

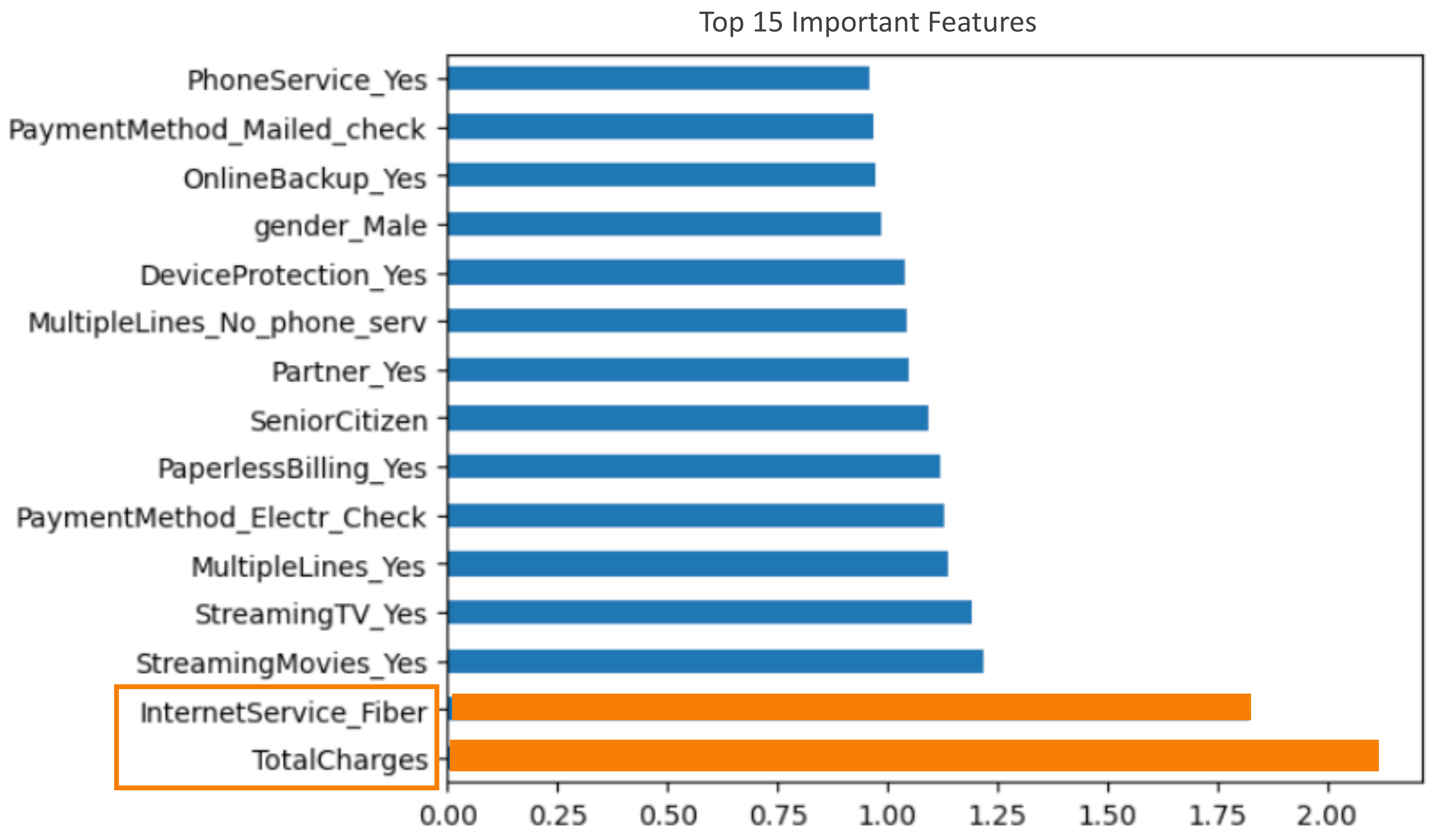
Benchmark

Started with Logistic Regression

Model	Recall	Precision	F1
Logistic Regression	80%	48%	60%

Feature Probability

Coefficient Odds Review: Feature Probability of Churn (Logistic Regression)



Highest Probability of Indicating Churn

- TotalCharges
- Fiber Internet Service

Lowest Probability of Indicating Churn

- Tenure
- Two Year Contract

Model Settings:
- Default hyperparameters
- Oversampling

Test More Models

Additional algorithms

Model	**Recall**	**Precision**	**F1**
Logistic Regression	80%	48%	60%
XGBoost	74%	50%	60%
Random Forest	60%	57%	58%
Decision Tree	48%	47%	48%

Tune Hyperparameters

Random Forest + Decision Tree

- Default hyperparameters
- Oversampling

- Tuned hyperparameters
- Oversampling

Model	Recall	Precision	F1
Logistic Regression	80%	48%	60%
Random Forest Hyperparameters Tuned	76%		
XGBoost	74%	50%	60%
Random Forest	60%	57%	58%
Decision Tree	48%	47%	48%
Decision Tree Hyperparameters Tuned	48%		

Random Forest Gridsearch Best Parameters: max_features: 23, n_estimators: 119

Decision Tree Gridsearch Best Parameters: max_depth: 1, min_samples_leaf: 1

Winners

(thus far in the process)

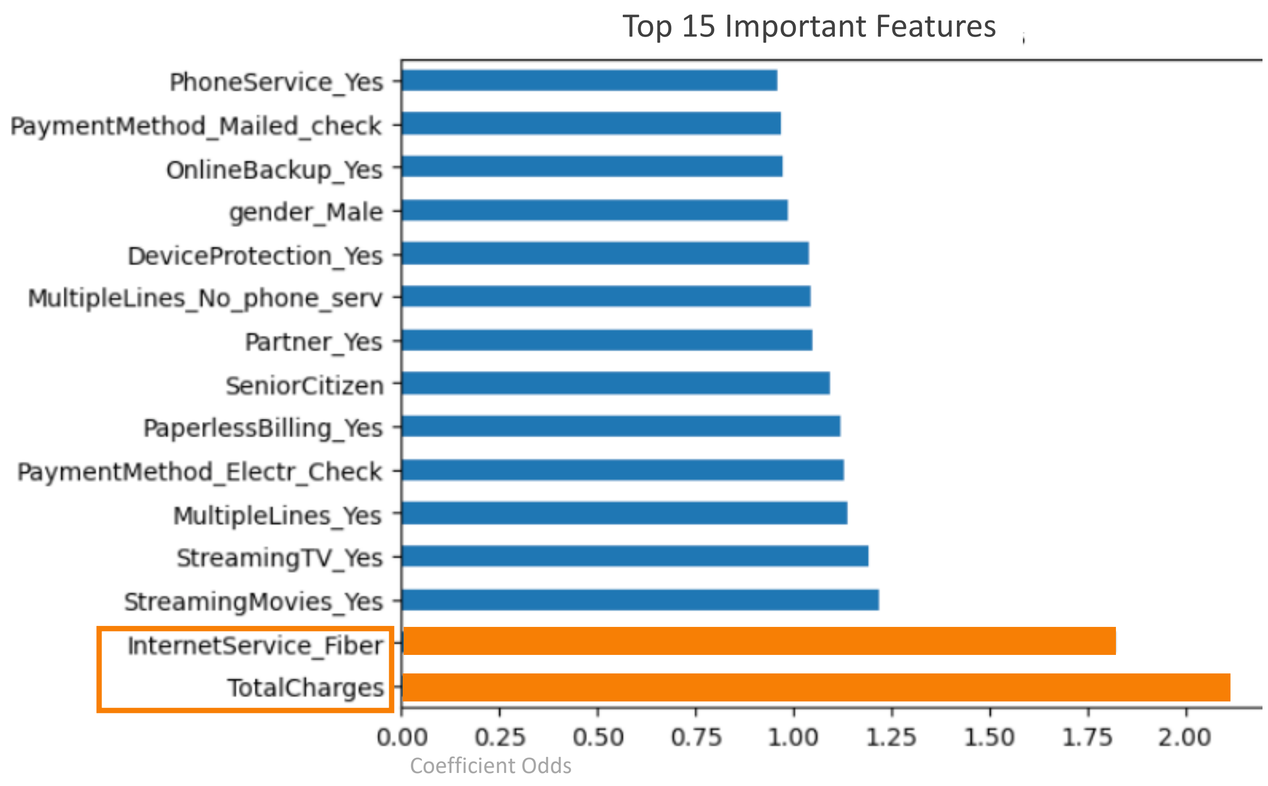
Model	Recall
Logistic Regression	80%
Random Forest Hyperparameters Tuned	76%

Let’s compare how the features performed for each model

Feature Comparison by Model

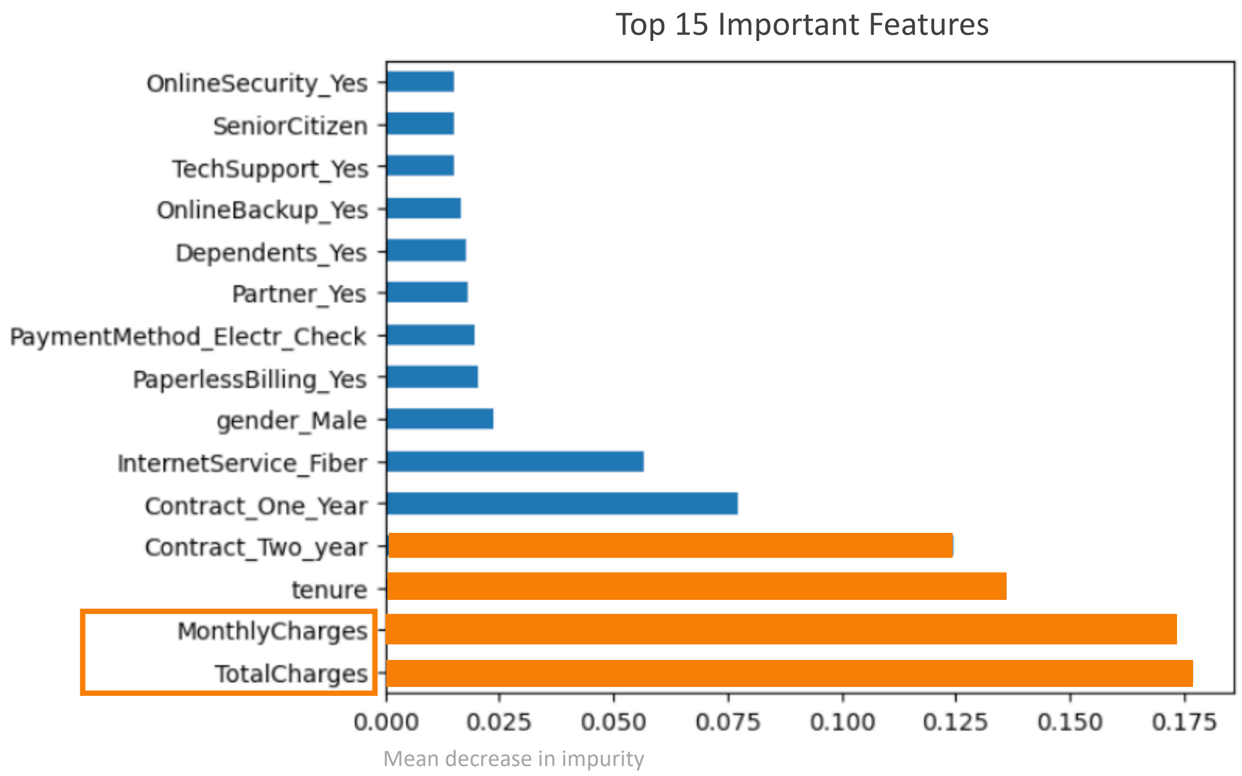
Both models agree that Total Charges is the most important, differ on others

Logistic Regression, Defaults



Recall: 80%

Random Forest, Tuned Hyperparameters



Recall: 76%

Future Work

Modeling

Try Voting + Stacking Ensemble Methods

- To learn the different aspects of the data with each model
- Add naïve bayes into the mix

Further hyperparameter tuning

- Experiment with probability threshold
- XGBoost could be a winning model with tuning
- Logistic Regression:
  - Grid search with lasso regression to reduce feature space and determine impact on model score

Explore 'Total Charges' feature

- Feature engineer: bin by range to determine specific total charges that indicate churn
- Look for price sensitivity research within the company. Findings could contribute to feature engineering.

Coding

Update pipeline to work with Gridsearch

Consider utilizing some of the flows discovered while doing this project

Discussion