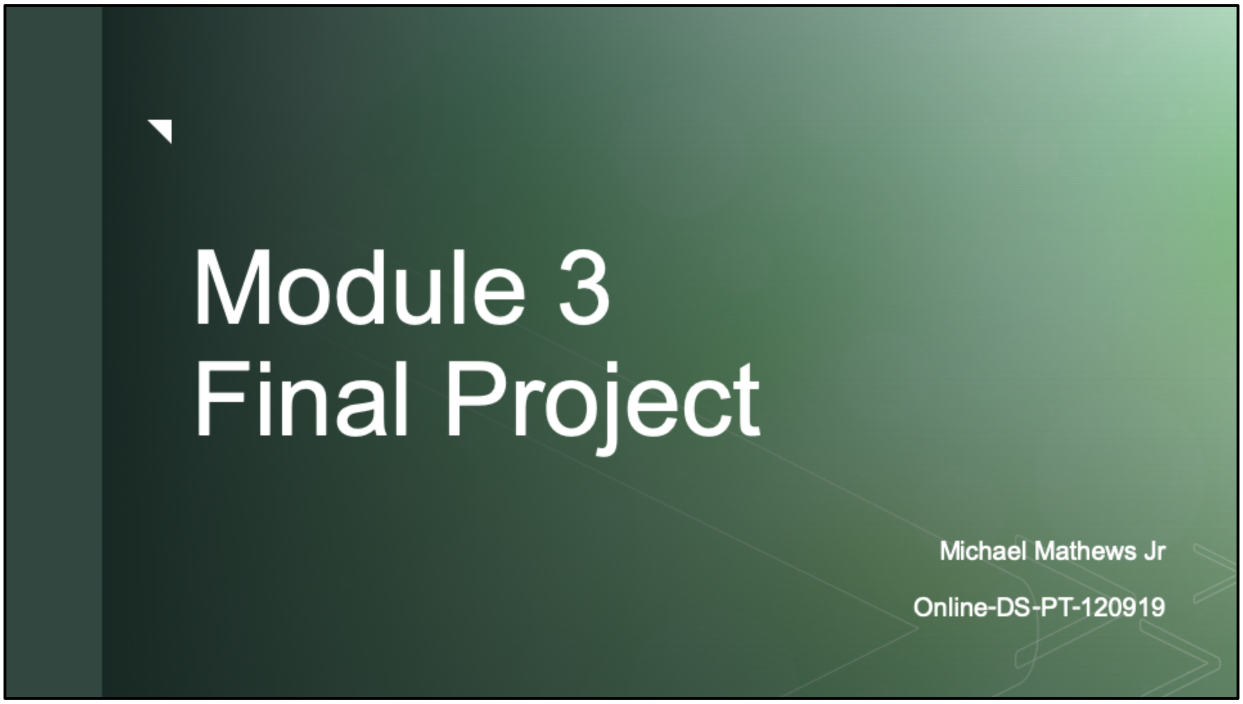
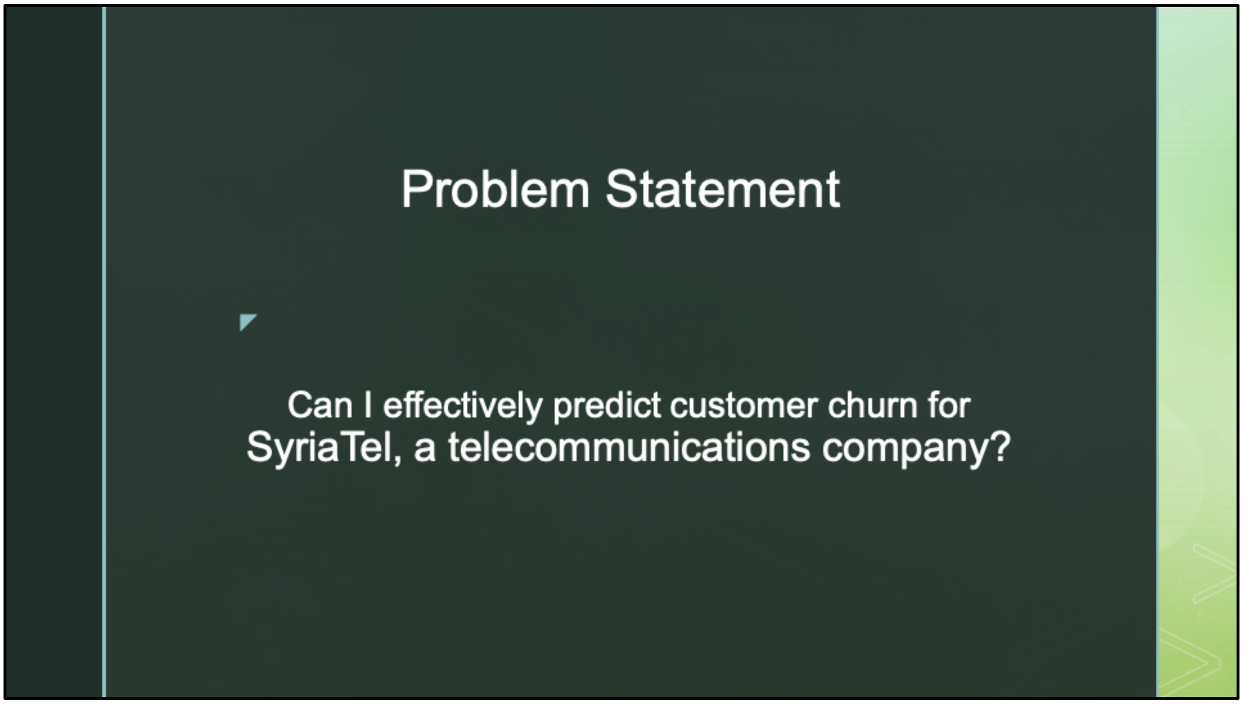


Module 3 Final Project

Michael Mathews Jr

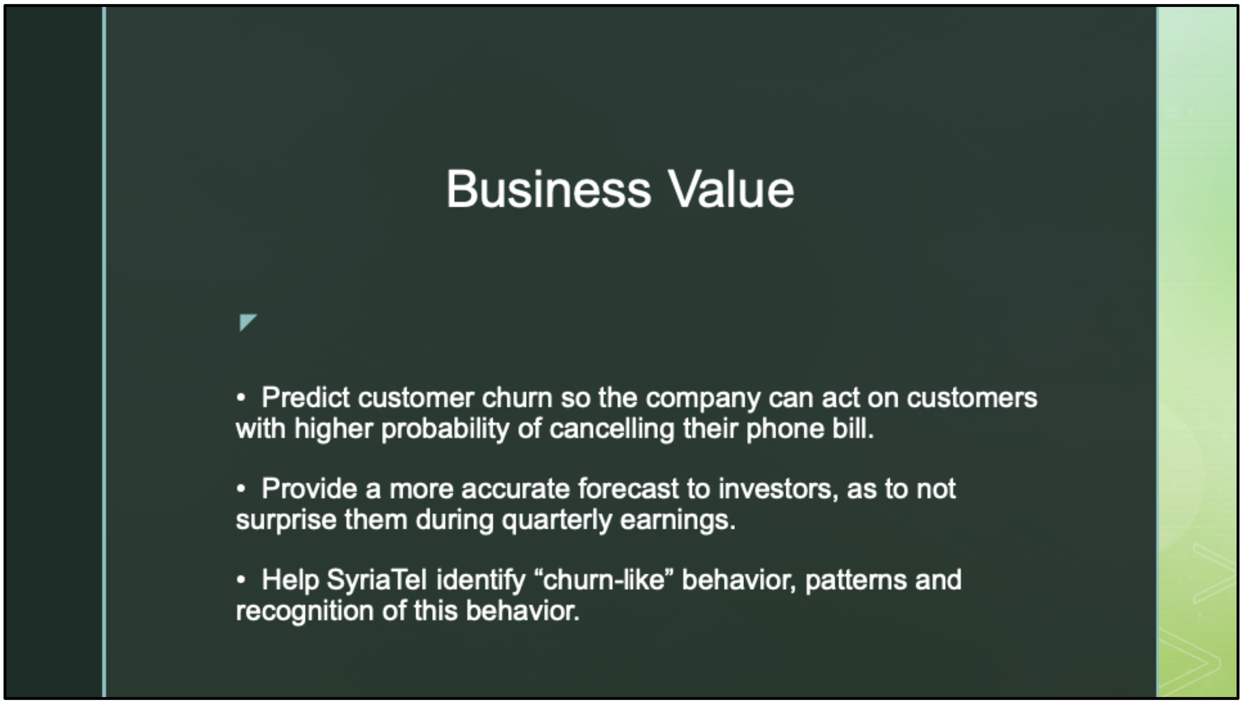
Online-DS-PT-120919



Problem Statement

Can I effectively predict customer churn for SyriaTel, a telecommunications company?

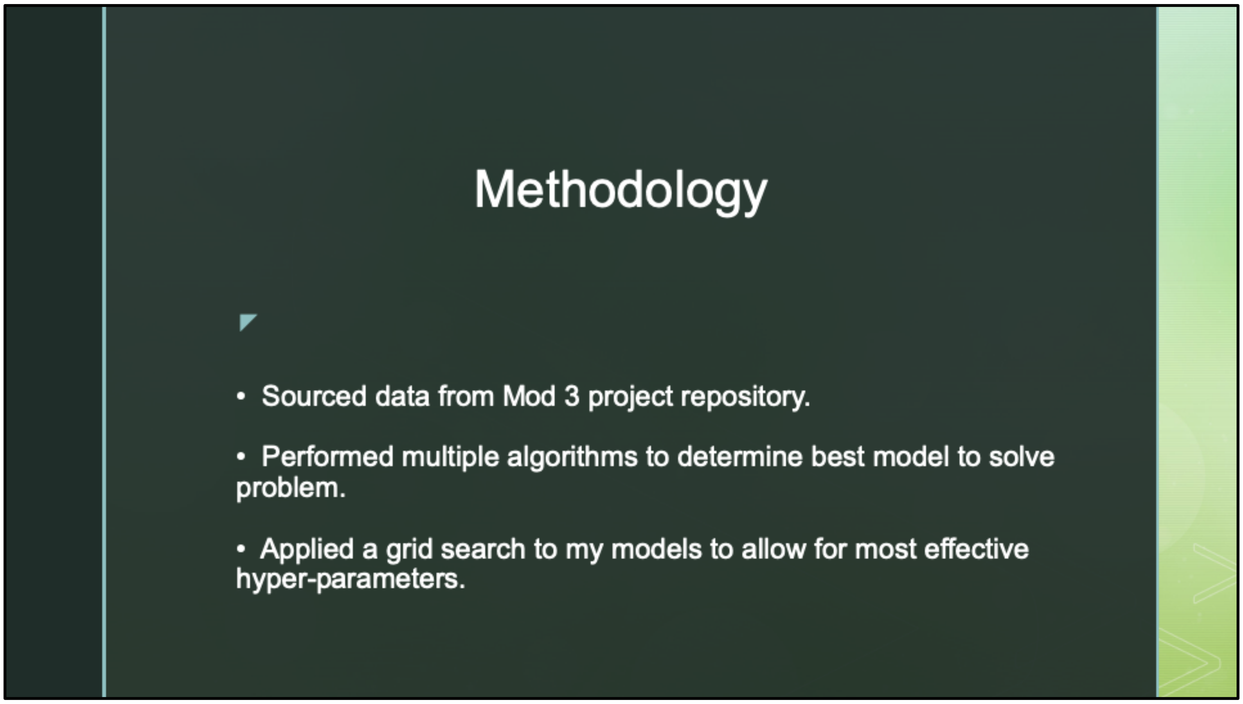
Customer churn is an important metric in any B2C company. Does SyriaTel have a rich enough dataset that allows for accurate predictions?



Business Value

- Predict customer churn so the company can act on customers with higher probability of cancelling their phone bill.
- Provide a more accurate forecast to investors, as to not surprise them during quarterly earnings.
- Help SyriaTel identify “churn-like” behavior, patterns and recognition of this behavior.

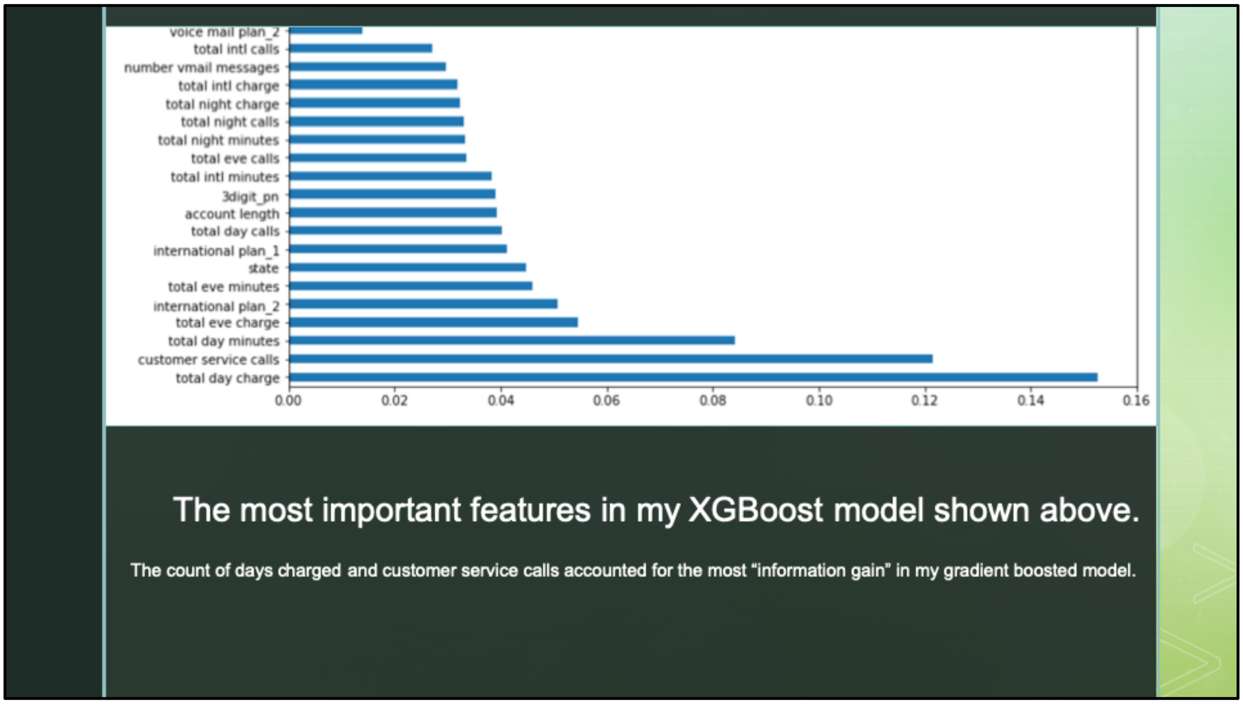
- The accuracy of the model should allow the business to leverage the information for financial gain.
- Making business decisions using advanced statistics leaves little room for gut feelings and opinions. Get a statistical advantage over the competition by utilizing a machine learning model.
- Make the whole business aware of the information and why it is significant. Allow employees to incorporate the information into their day to day.



Methodology

- Sourced data from Mod 3 project repository.
- Performed multiple algorithms to determine best model to solve problem.
- Applied a grid search to my models to allow for most effective hyper-parameters.

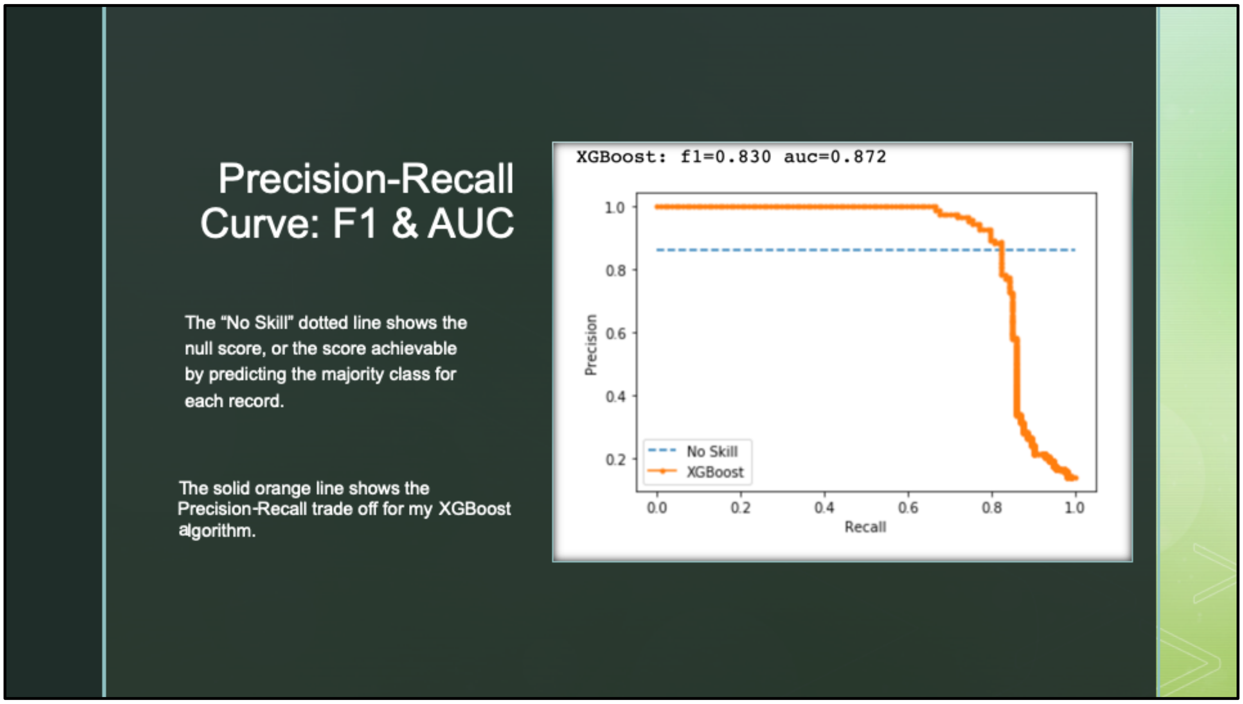
- Please check my repo for the dataset used in this analysis.
- I tested many algorithms to determine the best model to solve this problem. This includes tree based algos (random forest, xgboost), logistic regression, and even a neural network. Gradient boosted trees in combination with a grid search provided the best results.
- I was most concerned with the recall score for the minority target variable (1 = ‘churn’). I tuned the model to return the best recall for churn as possible. I didn’t even have to make a large trade-off, as precision for the minority target was 99%!



The most important features in my XGBoost model shown above.

The count of days charged and customer service calls accounted for the most "information gain" in my gradient boosted model.

Its important for SyriaTel to understand the behavior that leads to customer churn. By providing a bar chart showing feature importance, SyriaTel can use this information to take action.

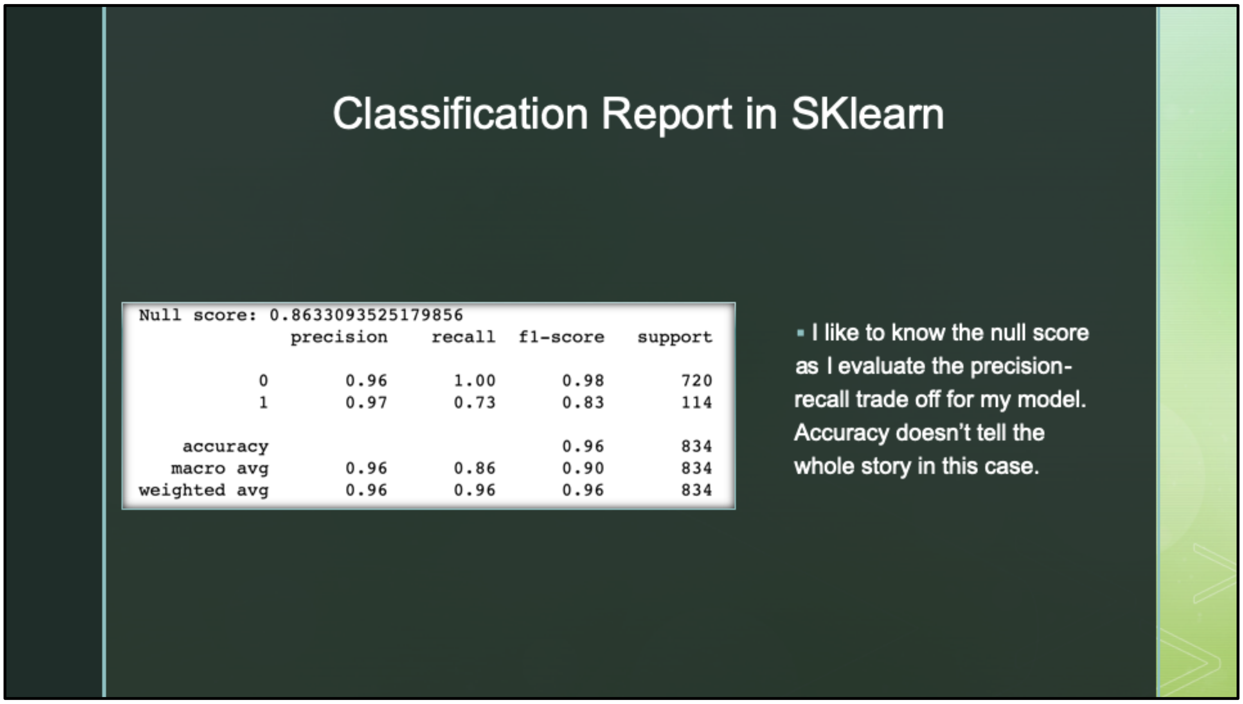


Precision-Recall Curve: F1 & AUC

The "No Skill" dotted line shows the null score, or the score achievable by predicting the majority class for each record.

The solid orange line shows the Precision-Recall trade off for my XGBoost algorithm.

The precision-recall graph is used when the minority target is most important to accurately predict. In this case, we need high recall for churn customers so we don't lose them before its too late. Every machine learning solution needs to evaluate this trade off.



Classification Report in SKlearn

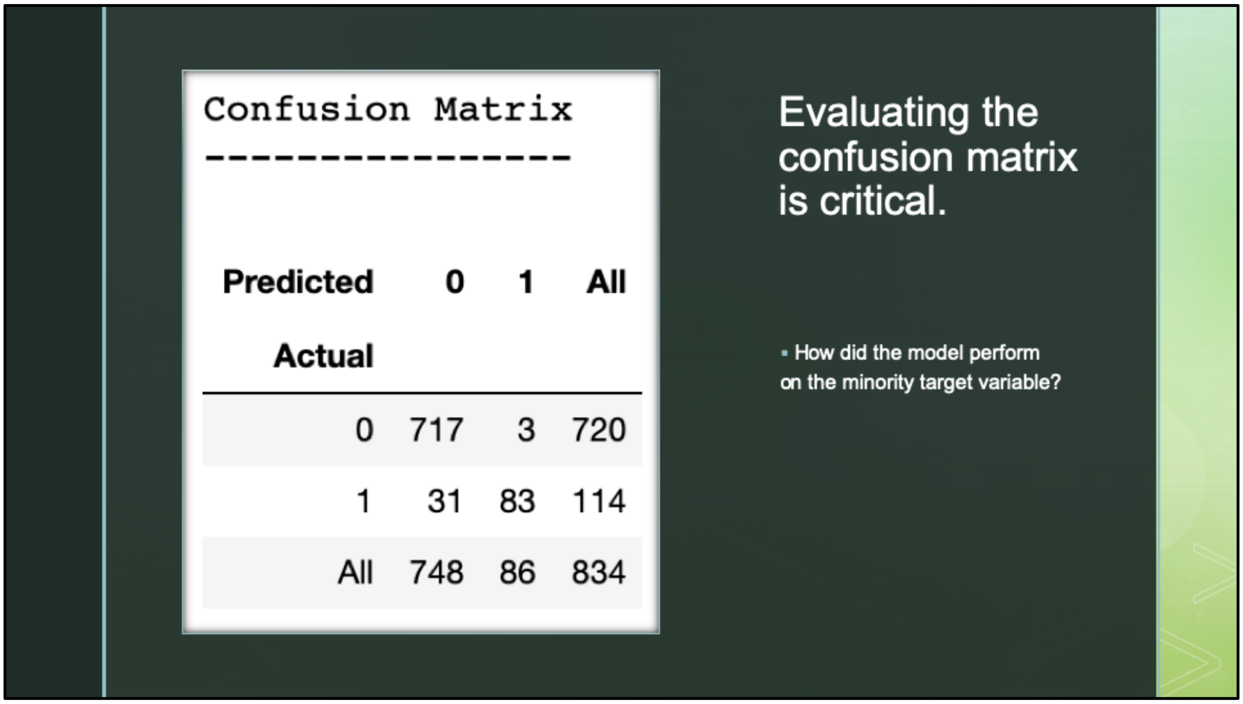
```
Null score: 0.8633093525179856
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       720
           1       0.97      0.73      0.83       114

    accuracy                           0.96       834
   macro avg       0.96      0.86      0.90       834
weighted avg       0.96      0.96      0.96       834
```

- I like to know the null score as I evaluate the precision-recall trade off for my model. Accuracy doesn’t tell the whole story in this case.

Classification reports are important to evaluate because it offers metrics that are critical to any binary classification problem, such as this one. Accuracy in the mid 90% range looks great on paper, but it may not be acceptable if the target variable is heavily imbalanced.



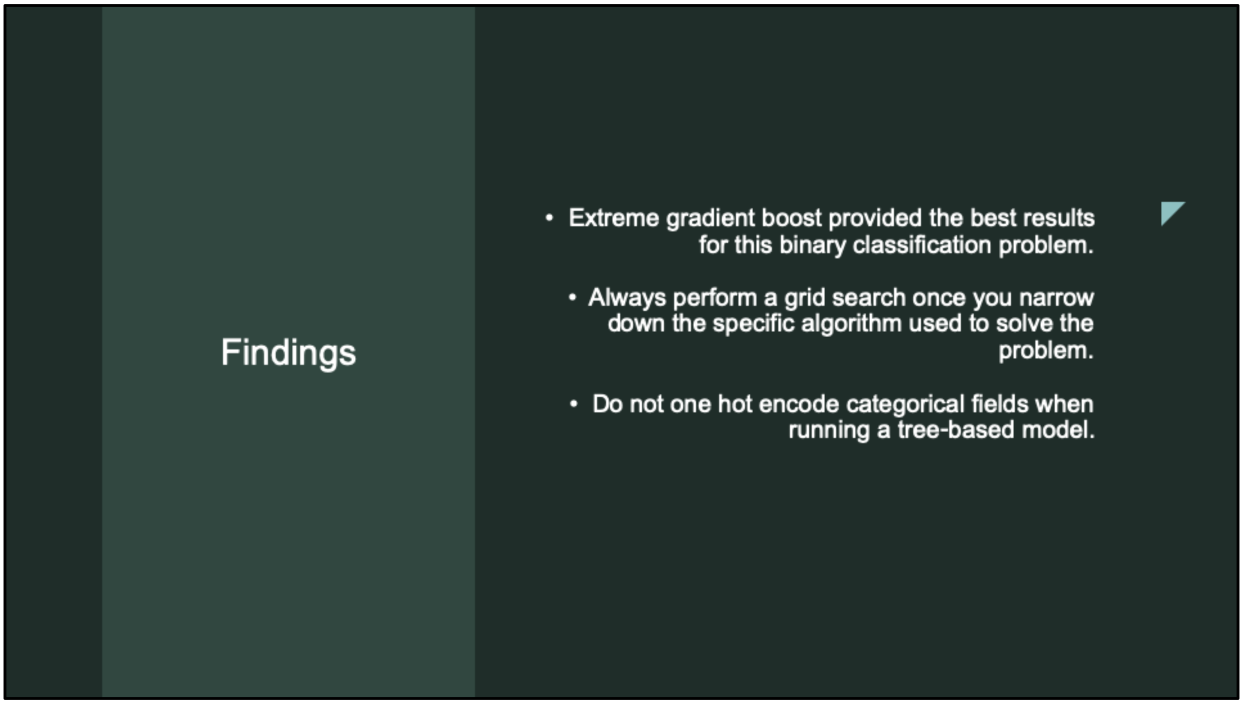
```
Confusion Matrix
----------------
```

Predicted	0	1	All
Actual			
0	717	3	720
1	31	83	114
All	748	86	834

Evaluating the confusion matrix is critical.

- How did the model perform on the minority target variable?

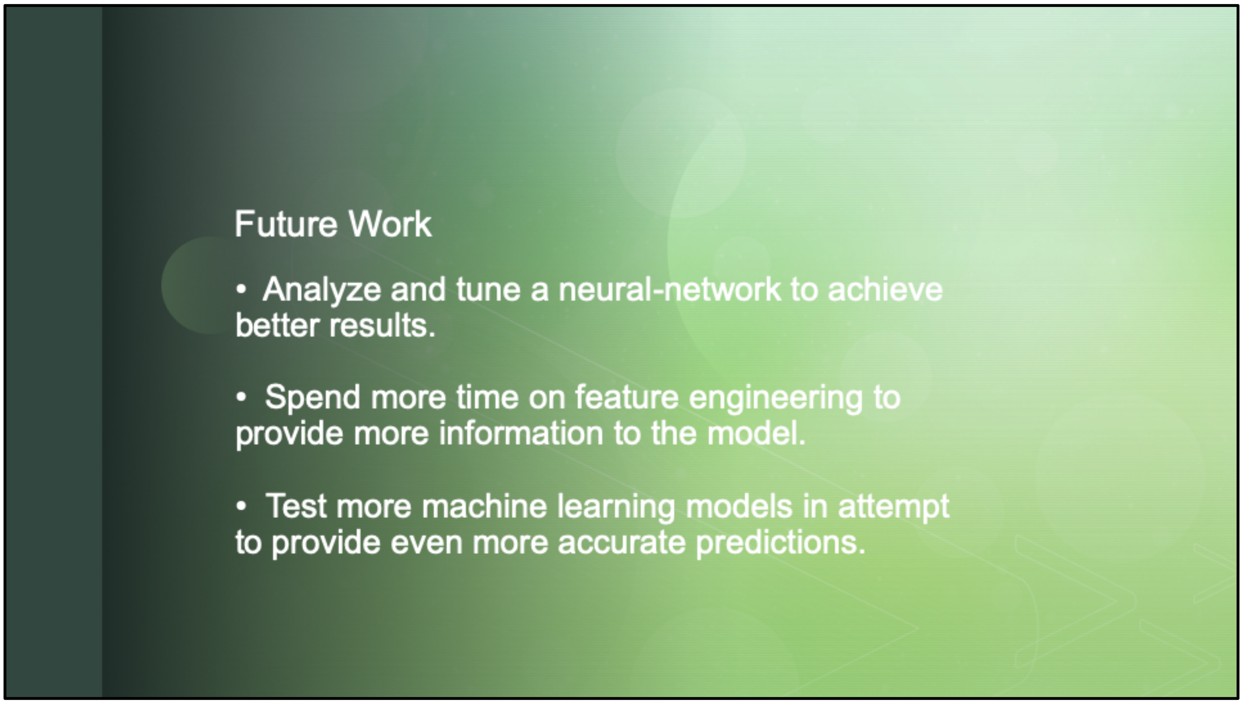
Just like the classification report, the confusion matrix must be evaluated for any classification problem. This gives you the real number results of how the model performed.



Findings

- Extreme gradient boost provided the best results for this binary classification problem.
- Always perform a grid search once you narrow down the specific algorithm used to solve the problem.
- Do not one hot encode categorical fields when running a tree-based model.

- XGBoost model works best for this specific binary classification problem.
- Grid search continues to improve the model by tuning the hyper-parameters.
- One hot encoding the categorical features negatively impacted the model. Tree-based algorithms do not handle sparse matrix datasets well.



Future Work

- Analyze and tune a neural-network to achieve better results.
- Spend more time on feature engineering to provide more information to the model.
- Test more machine learning models in attempt to provide even more accurate predictions.

- Neural-networks, although often a black box can provide even more accurate results. This type of model should be tested and tuned to possibly achieve an increase in accuracy.

Thank You

- Questions?
- Comments?
- Feedback?

- Questions, comments and/or feedback? Thank you for following along!