Credit Card Fraud Detection

Introduction to Machine Learning (STA S380)

Group 10

- Ammar Mustufa
- Shirley Liu
- Navya Singhal
- Sankeerth Viswanadhuni

Introduction

About the data set
Credit card transaction dataset containing legitimate and fraud transactions*
(Jan 2019 - Dec 2020)

Problem Statement
Design a method to identify and flag potential fraud transactions

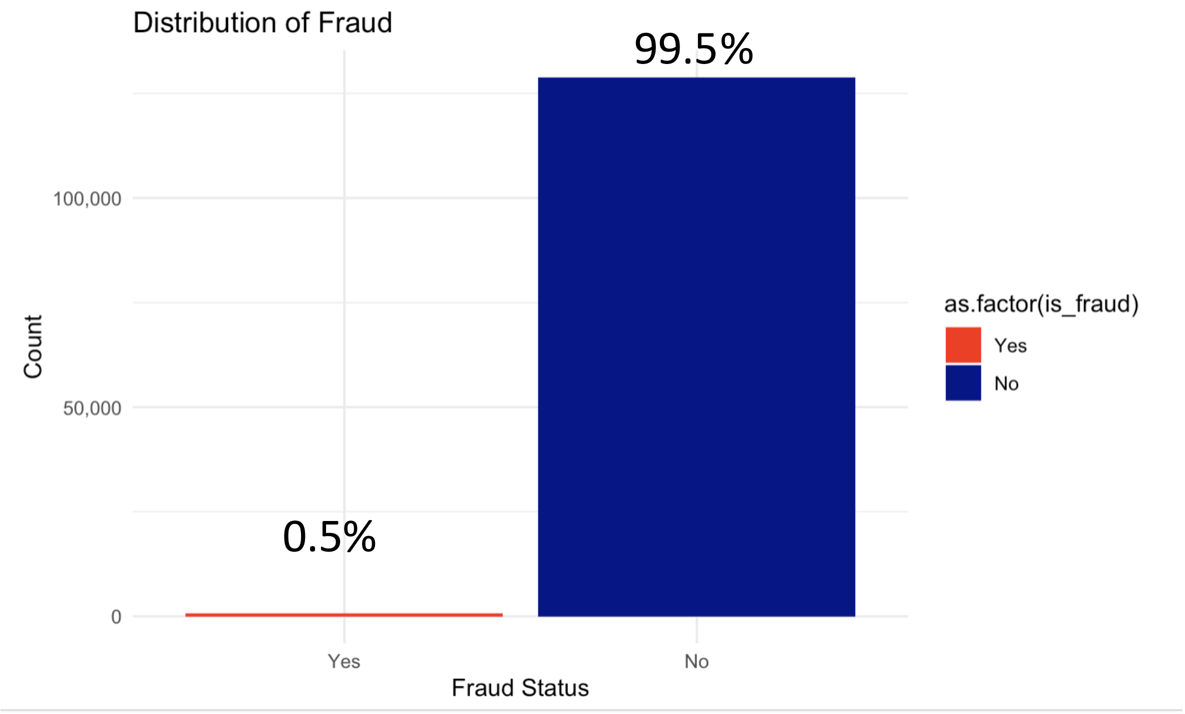
Data preparation
Data cleaning and feature engineering (if reqd.) for building a predictive model

Classification techniques
Random forest, boosting, and bagging

*[*Credit Card Transactions Fraud Detection Dataset (Source: Kaggle)]*

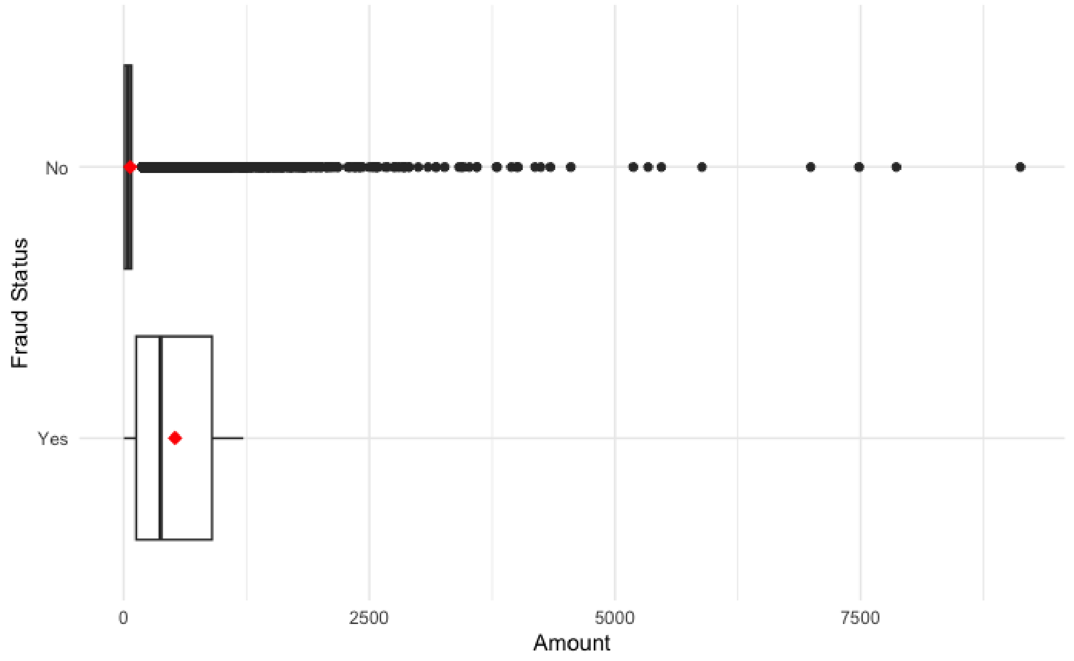
About the dataset

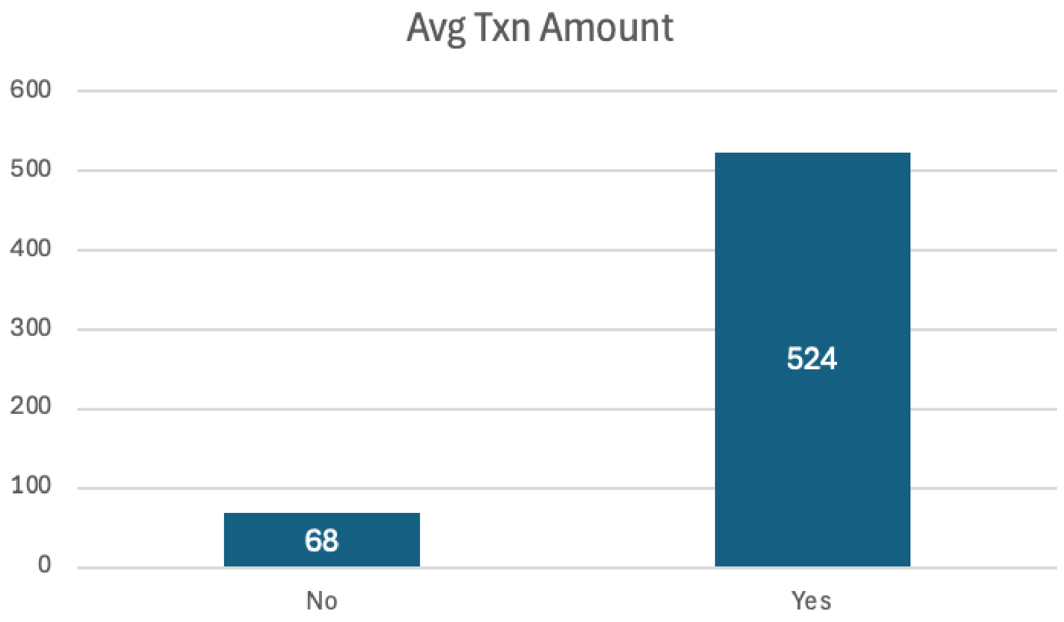
- 129,000 data points with 30 variables
- Y variable : is_fraud (1 / 0)
- Predictors Available : Txn. Amount, Age, Txn. Time Stamp, Category, DOB, Category, Longitude, Latitude etc.



About the dataset

- Avg. transaction amount for a fraudulent transaction is $524 vs legitimate transactions is $68
- Fraudulent transactions are concentrated b/w $10-1200 range
- Legitimate transactions are spread across till $10K

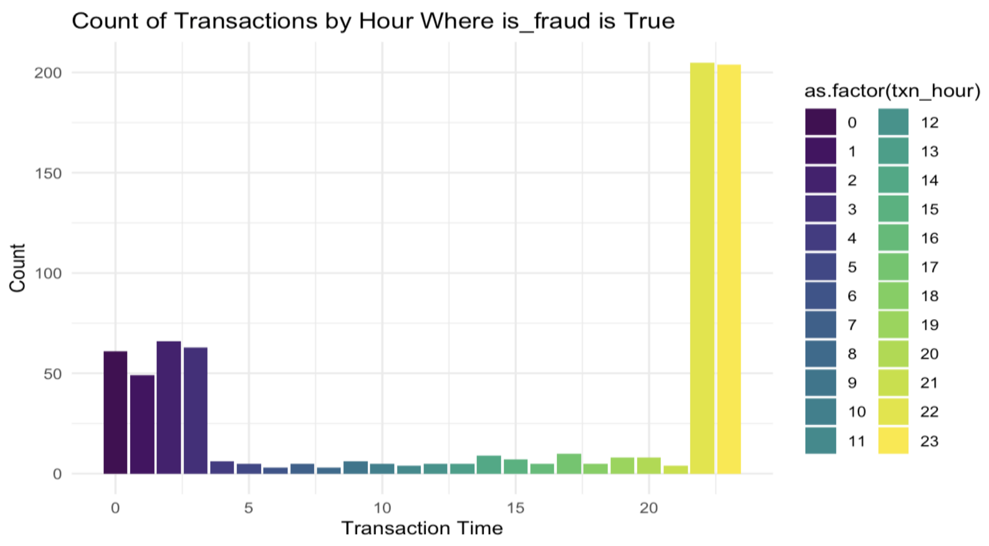


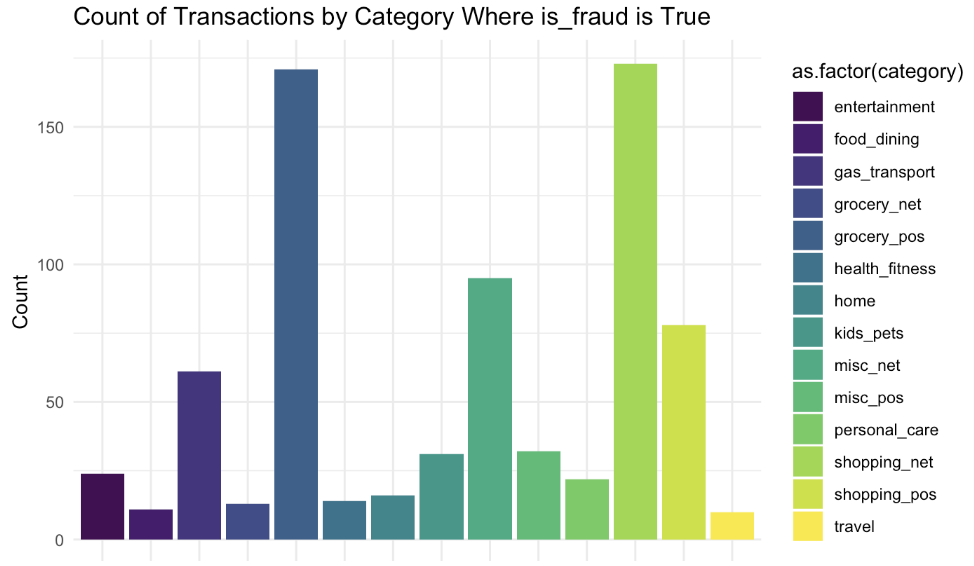


Feature Engineering

- Only 0.5% are fraudulent transactions
- Most default happens at midnight
- More default on online shopping and grocery

Categorize our data





Feature Engineering

Variable 1 - Category of Txn. Hour

- Extracted from Transaction timestamps
- 90% of the fraudulent transactions happen between 10 pm - 3 am
- Categorize into 2 buckets (Fraud Hour / Not Fraud Hour)

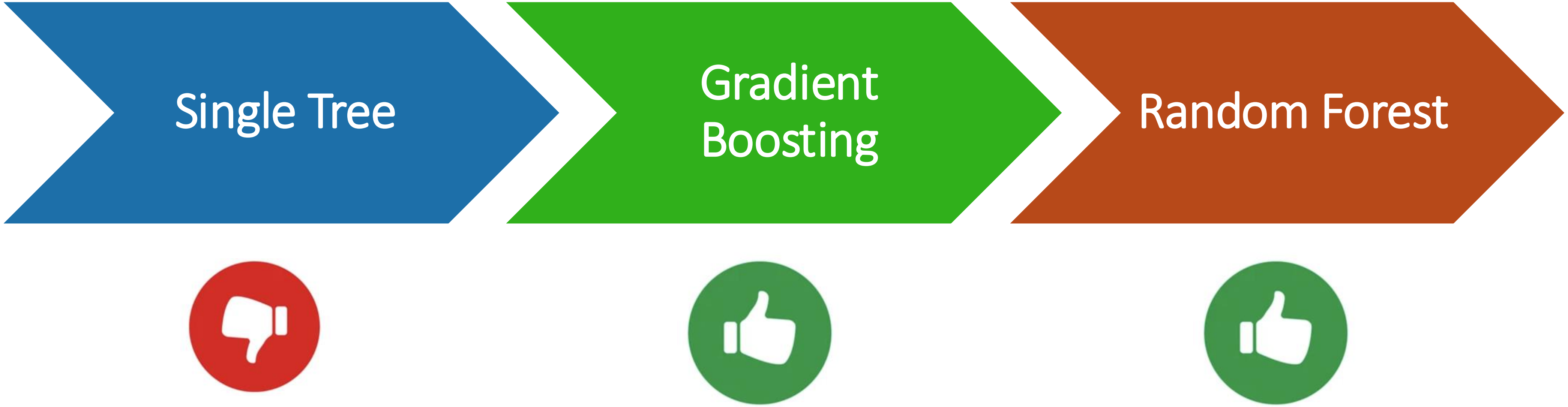
Variable 2 - Category of transaction

- 15 categories of transactions in total
- 45% of the fraudulent transactions are in 2 categories (online Shopping & grocery pos)
- Categorize into 3 buckets (High / Medium / Low Fraud)

Variable Selection

- Final predictors used for the analysis based on predictive quality and EDA
- Predictors used (6 in total) - Txn. Amount, Age, Txn. Hour, Txn. Day, **Txn. Hour category**, **Txn. Type category**

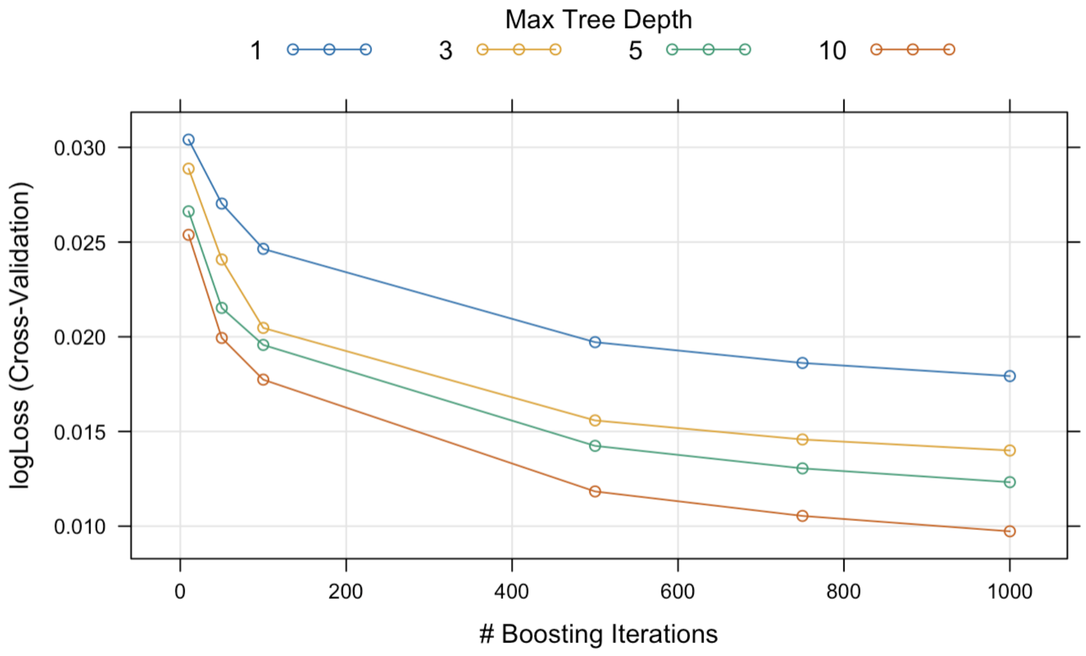
Classification - Model Selection



Oversimplified the model
Used only 3 predictors

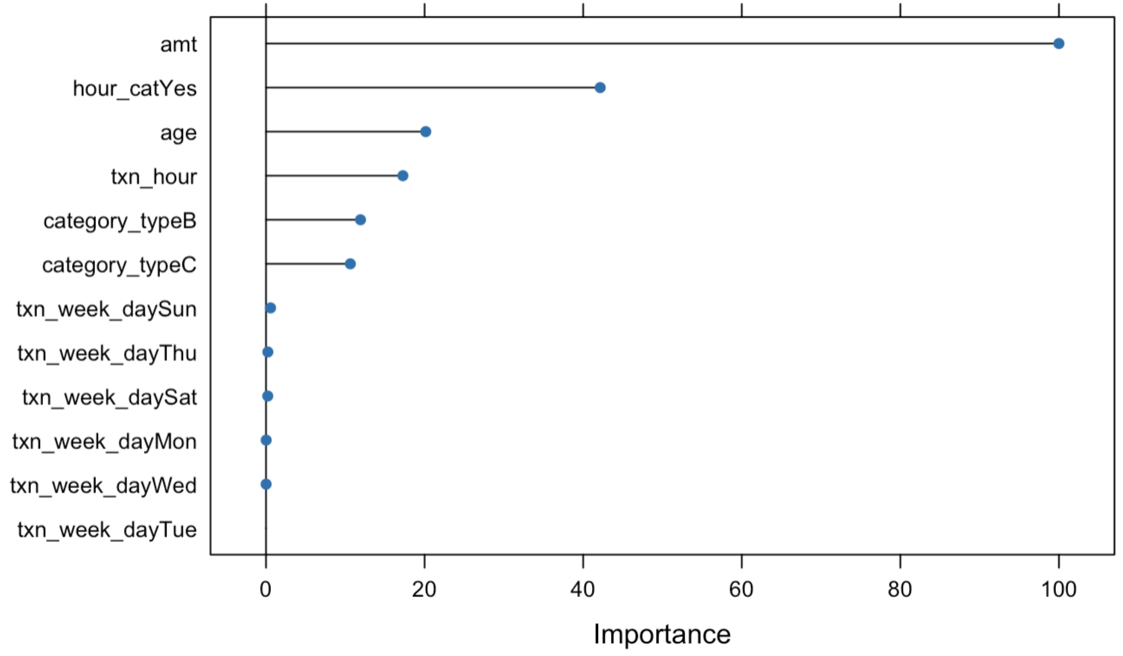
Overfitting but better prediction
Works well on validation data set

Model #1 - Gradient Boosting



Lowest logLoss with depth = 10 , trees =1000

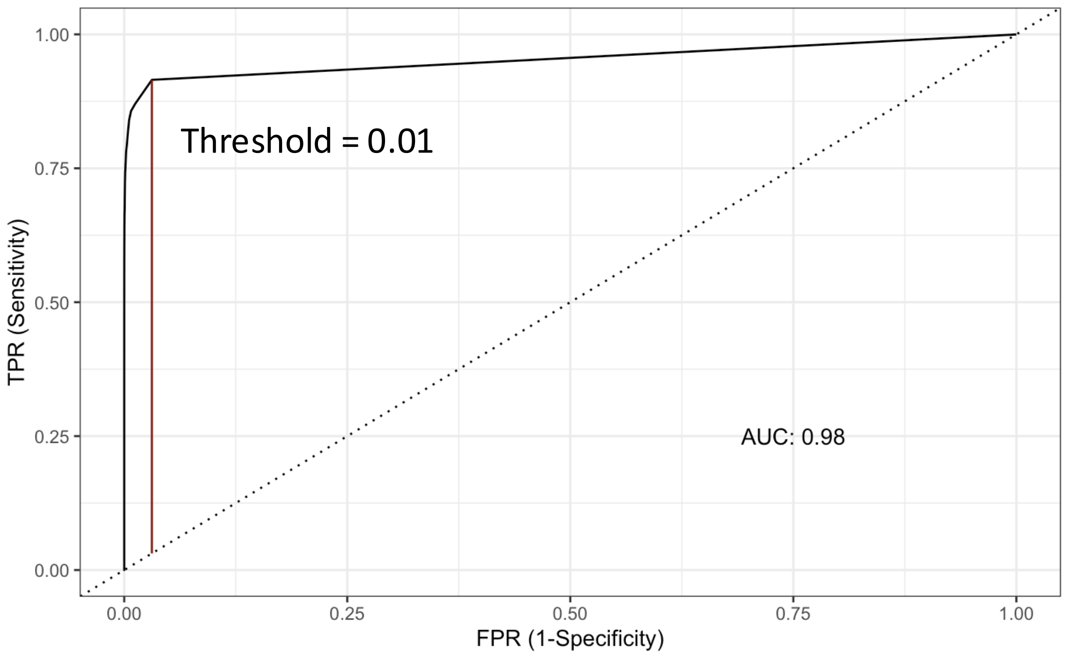
Var Importance



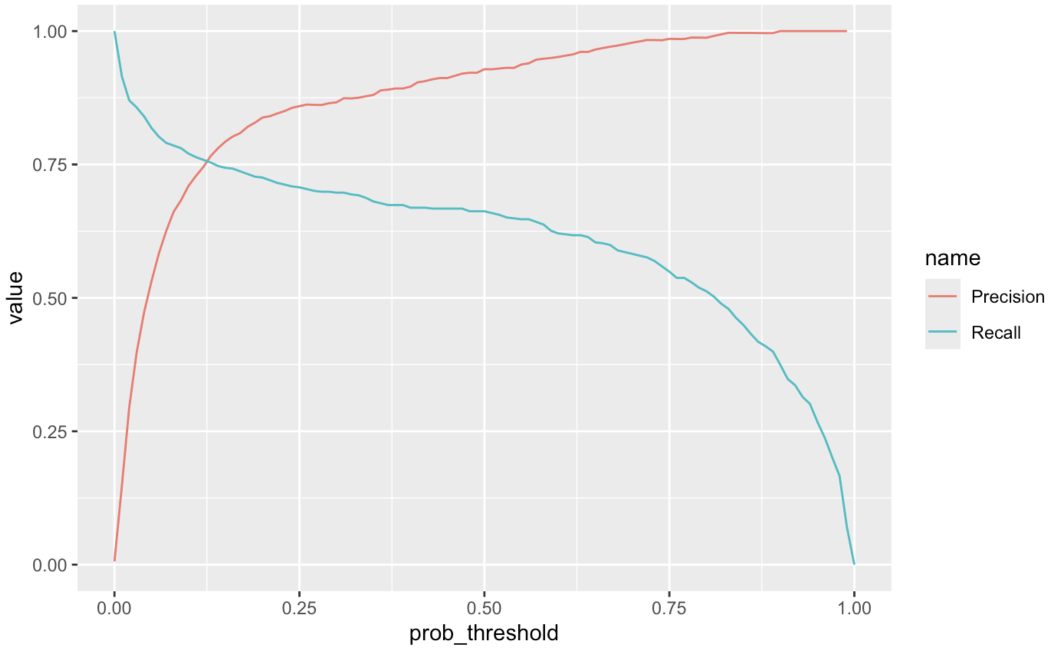
Hour category played an important role

Model #1 - Gradient Boosting

ROC Curve



Threshold: 0.01 vs 0.9

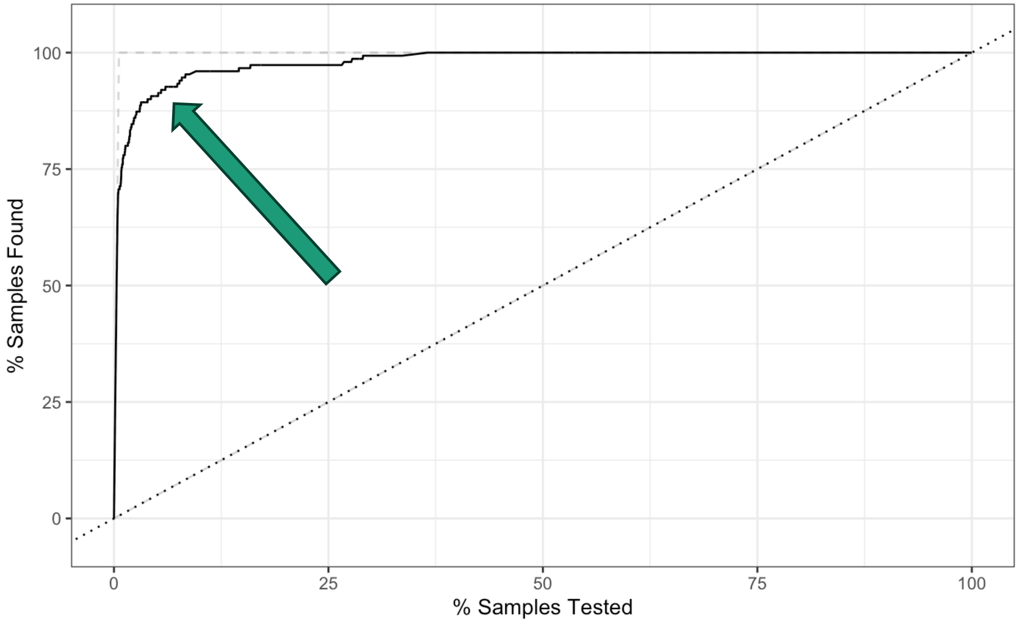


There is a trade-off between TPR and FPR

Different threshold can apply to different business objectives

Model #1 – Gradient Boosting

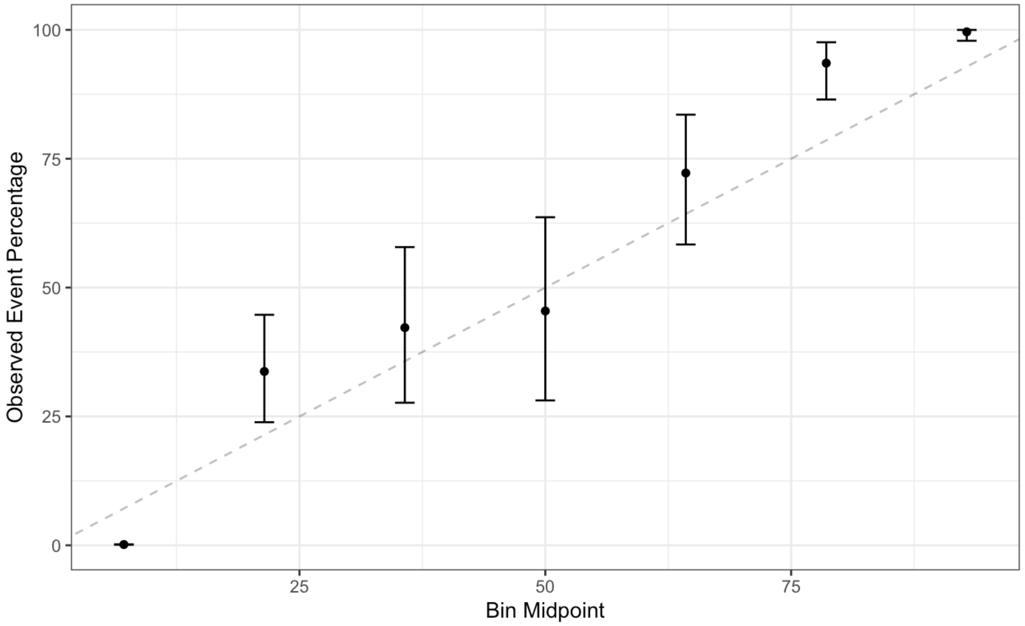
Lift Curve



Good at distinguishing fraud cases from a random guess

Overfitting issue

Calibration



Under confidence in predictions than in reality

Model #2 - Random Forest

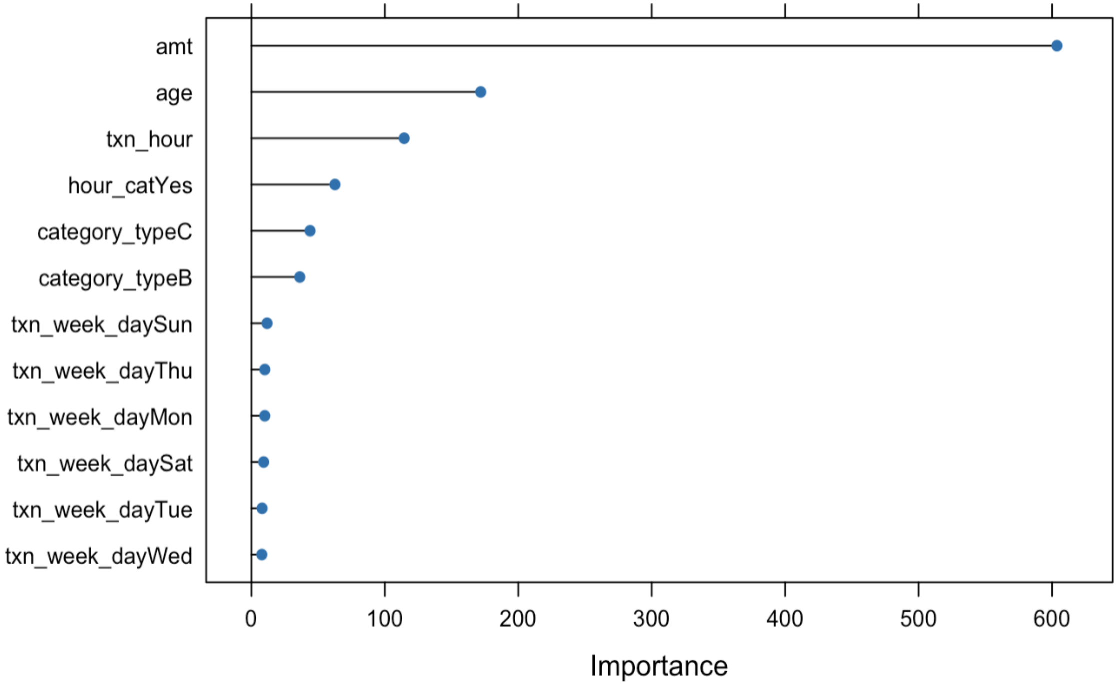
```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 93361, 93362, 93362, 93361, 93362, 93361, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa      AUC_ROC    TPR        FPR           logLoss
  3     0.9976864  0.7547335  0.9958556  0.6207104  1.163542e-04  0.007036978
  4     0.9982359  0.8208410  0.9969377  0.7055738  5.817806e-05  0.005042502
  5     0.9987468  0.8793052  0.9962087  0.7953552  6.787362e-05  0.004618398
  6     0.9993734  0.9428801  0.9962174  0.9051366  7.756918e-05  0.004319694

Accuracy was used to select the optimal model using  the one SE rule.
The final value used for the model was mtry = 6.
Cross-Validated (10 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)
```

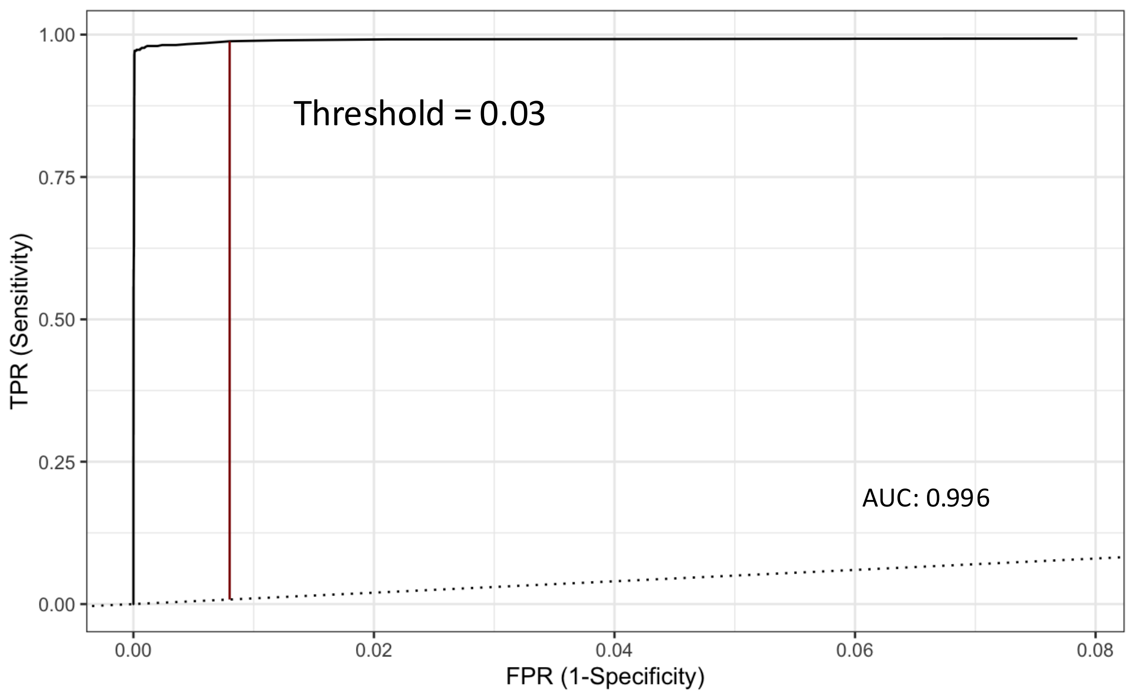
Var Importance



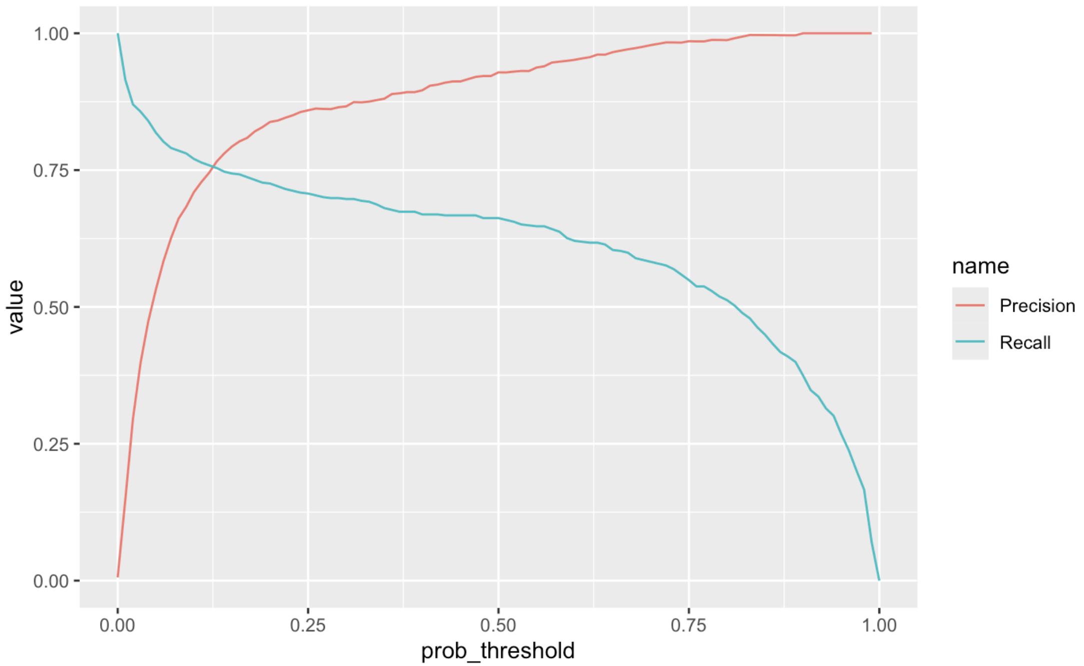
Top predictors are the same as the boosting - different relative order

Model #2 - Random Forest

ROC Curve



Threshold 0.03 vs 0.9

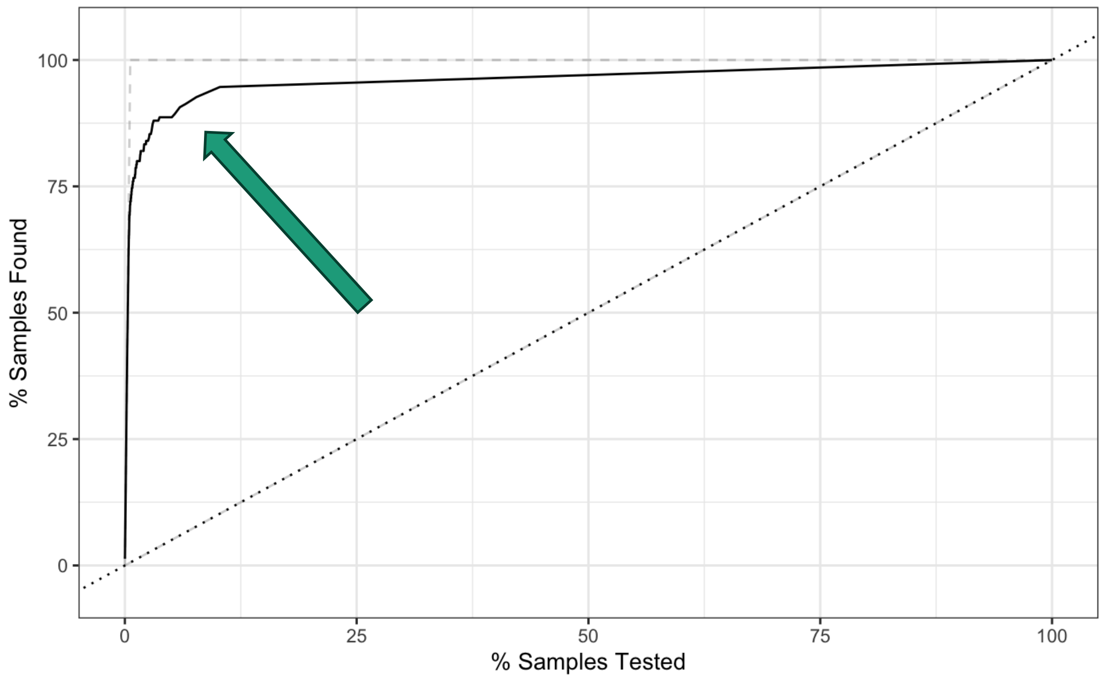


Higher AUC than boosting model

Similar trade-off between precision and recall

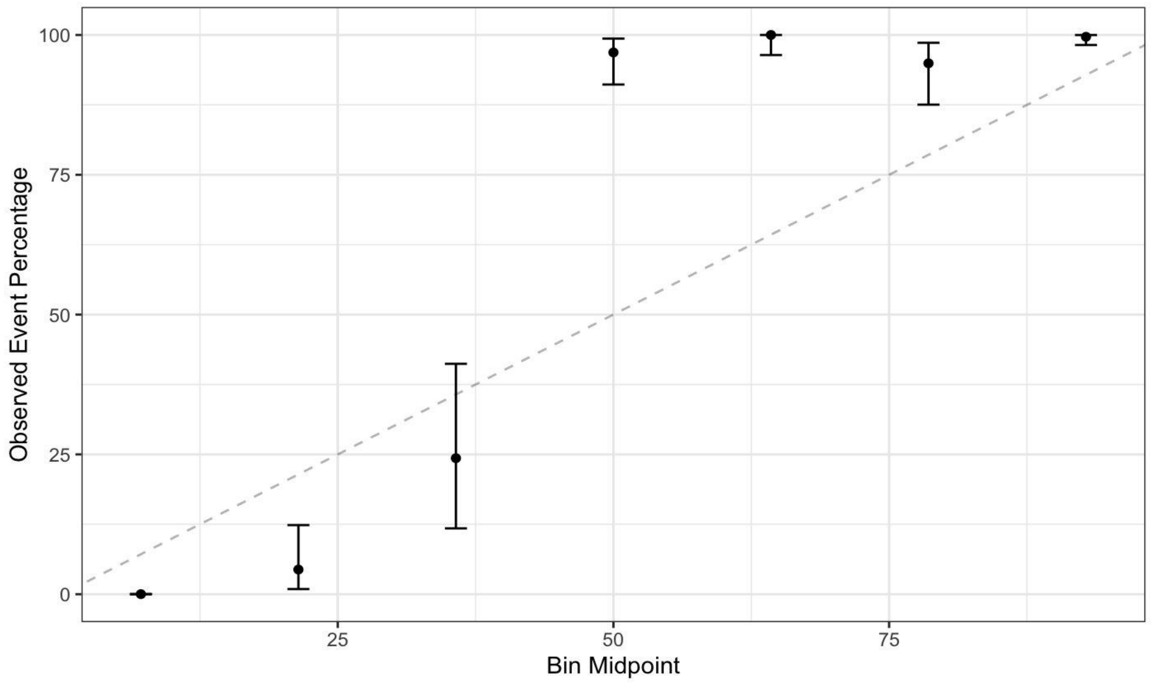
Model #2 – Random Forest

Lift Curve



Still overfitting but expected impact of the class imbalance

Calibration



Under confidence in high probabilities in training

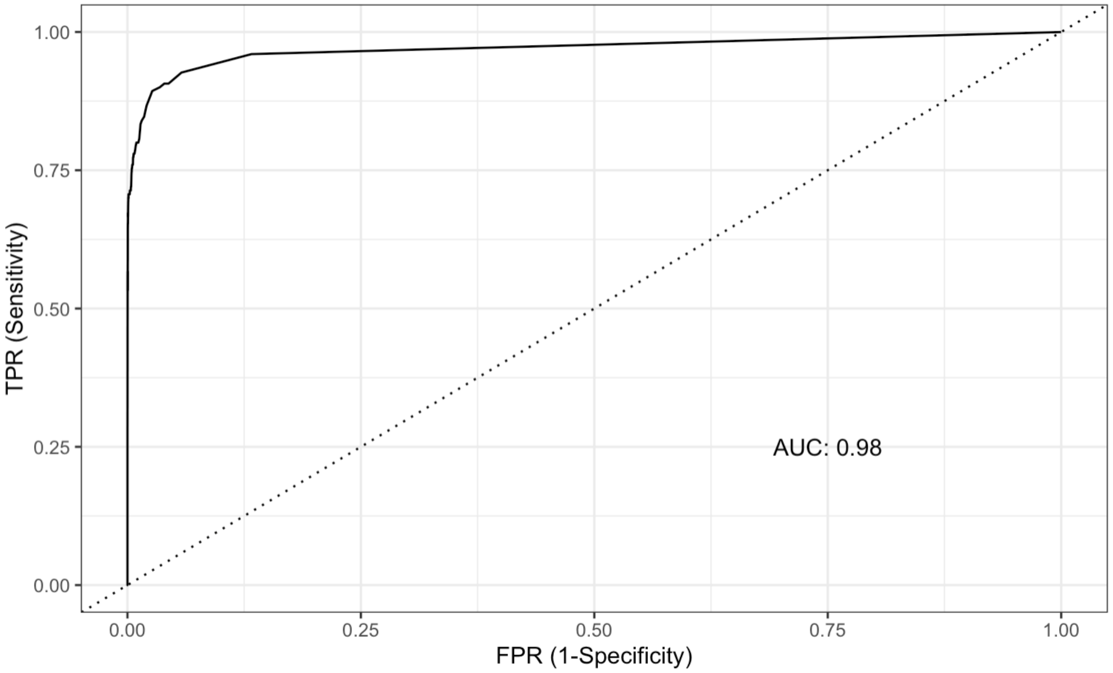
Model Validation Using Test Data

Random Forest

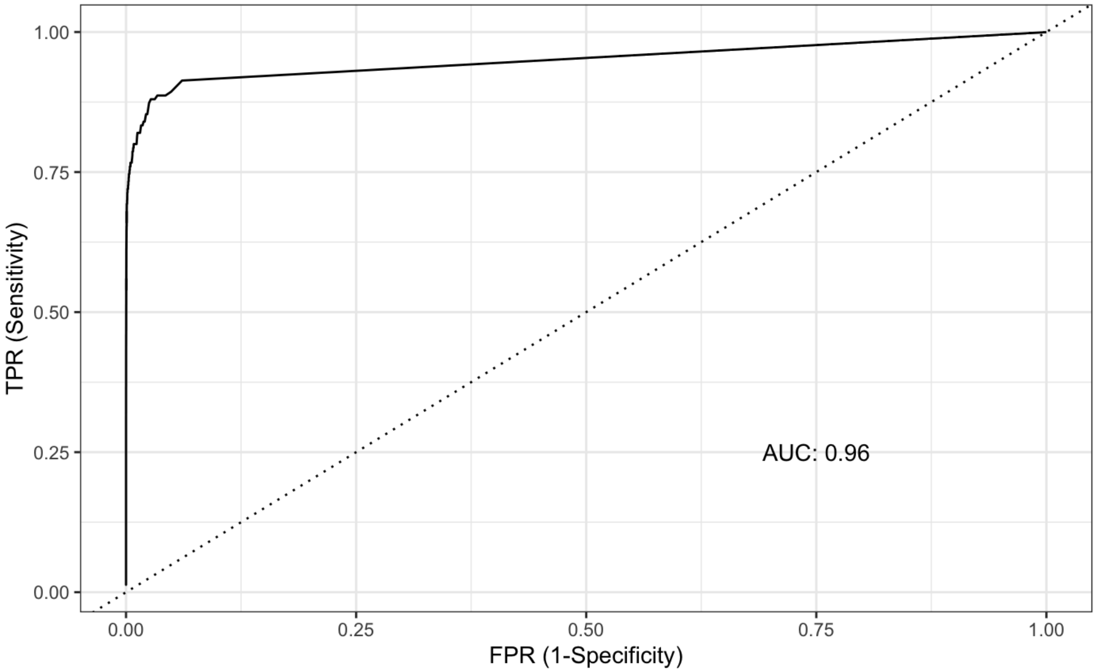
Gradient Boosting

Test Validation - ROC curve

Gradient Boosting



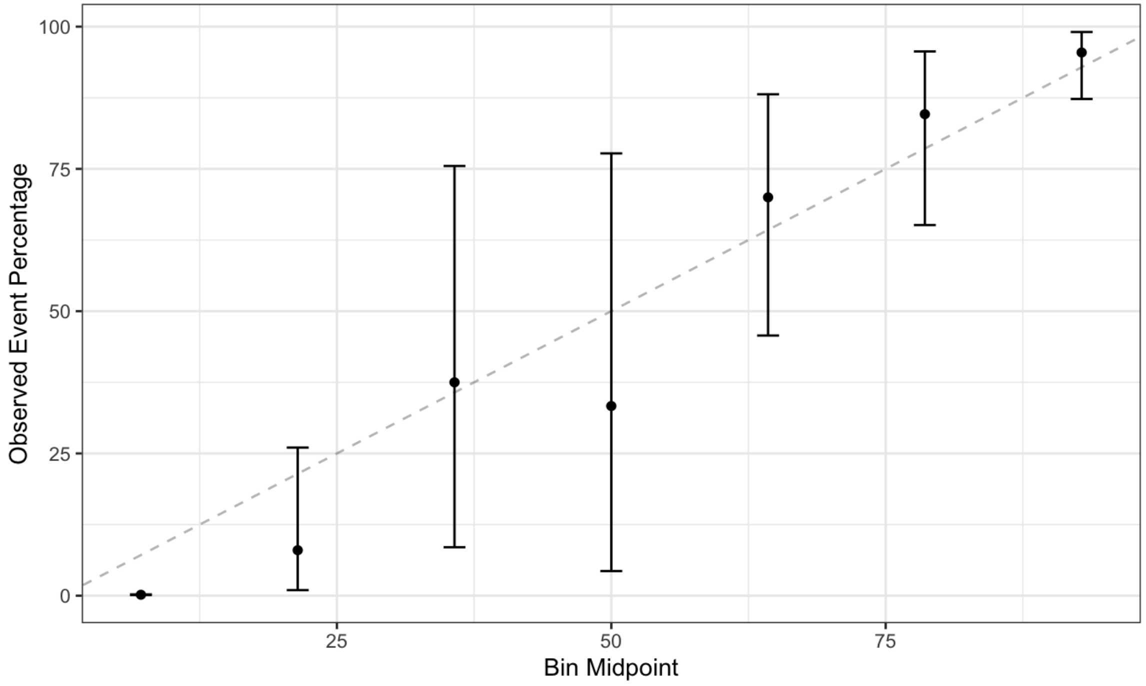
Random Forest



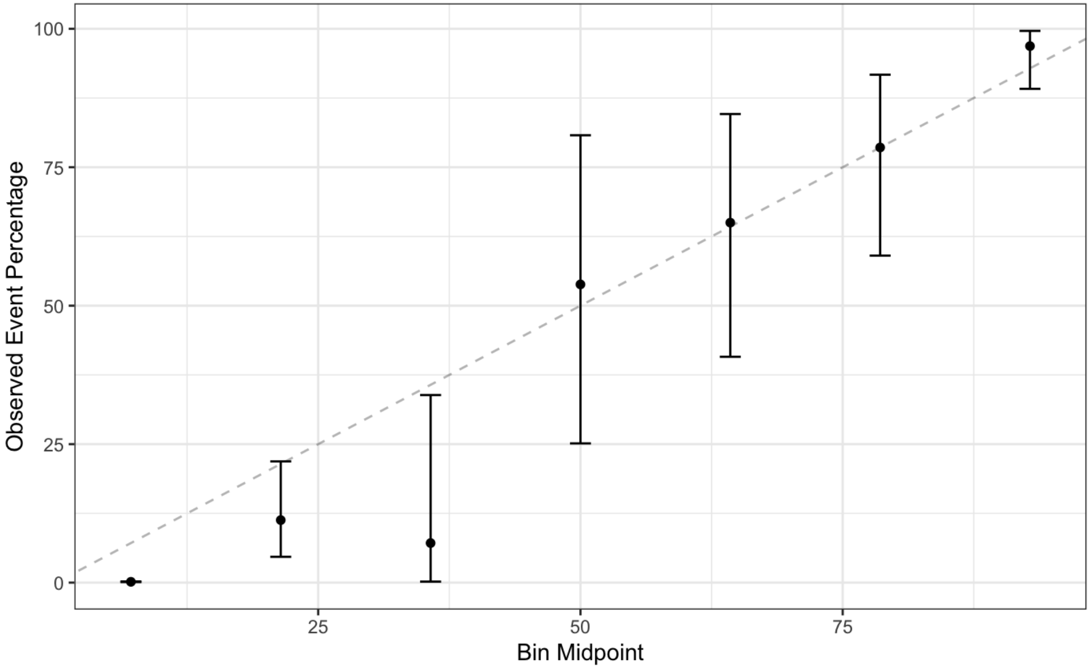
Very similar AUC to make a definite decision!!

Test Validation - Calibration plots

Gradient Boosting

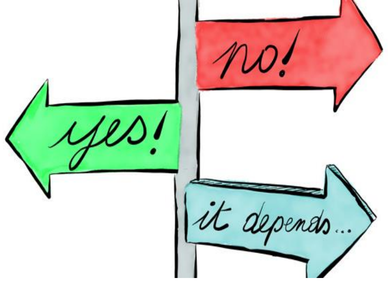


Random Forest



Similar trend on visual inspection as they align with the actual outcome line!!

Conclusion - Model Selection



Metric	Model #1 Gradient Boosting	Model #2 Random Forest
ROC AUC	0.98 ✓	0.96
Accuracy	96.5%	98.2%
Sensitivity (TPR)	90.0% ✓	83.3%
FPR (1 - Specificity)	3.4%	1.75% ✓
Precision	13.1%	21.9% ✓
Log loss	0.011	0.019

Values based on the optimal threshold found using Youden's J during the training stage

Which metric to choose?

- High recall rate – To avoid incorrect classification of fraud transaction as legitimate
- Minimize false positive – To avoid flagging legitimate transactions as fraudulent and disrupt regular business
- Highest ROC AUC -   To minimize the risk of fraud while maintaining customer satisfaction (by reducing false negatives)

Cost Analysis for Fraud Detection Models

Assumptions:

- Total Transactions (N): 1,000,000
- Fraud Prevalence (P_F): 0.5% (as per the data set)
- Cost of a False Positive (C_{FP}): $50 <- Incl. operational costs for manual review, and potential loss of future business
- Cost of a False Negative (C_{FN}): $1,000. <- Incl. direct financial loss, potential regulatory fines, and reputational damagex

Model	TPR	FPR
Model #1 - Gradient Boosting	90.0%	3.4%
Model #2 - Random Forest	83.3%	1.75%

Values based on the optimal threshold found using Youden's J during the training stage

Model #1 - Boosting

- False Positives Cost = FPR * $(1 – P_F)$*N*C_{FP} = $1,693,500
- False Negatives Cost = (1 - TPR)*P_F*N*C_{FN} = $500,000
- Total Cost = **$2,193,500**

Model #2 – Random Forest

- False Positives Cost = FPR * $(1 – P_F)$*N*C_{FP} = $869,125
- False Negatives Cost = (1 - TPR)*P_F*N*C_{FN} = $835,000
- Total Cost = **$1,704,125**

Verdict

Model #2 is significantly more cost-effective.

Higher false positive rate in Model #1 results in substantially higher costs.

Source - Stolfo, S., Fan, N. W., Lee, N. W., Prodromidis, A., & Chan, P. (2002). *Cost-based modeling for fraud and intrusion detection: results from the JAM project*. https://doi.org/10.1109/discex.2000.821515

Scope for further improvement

- Employ **under sampling techniques** to ensure that the model does not favor solely the majority class and prevent overfitting

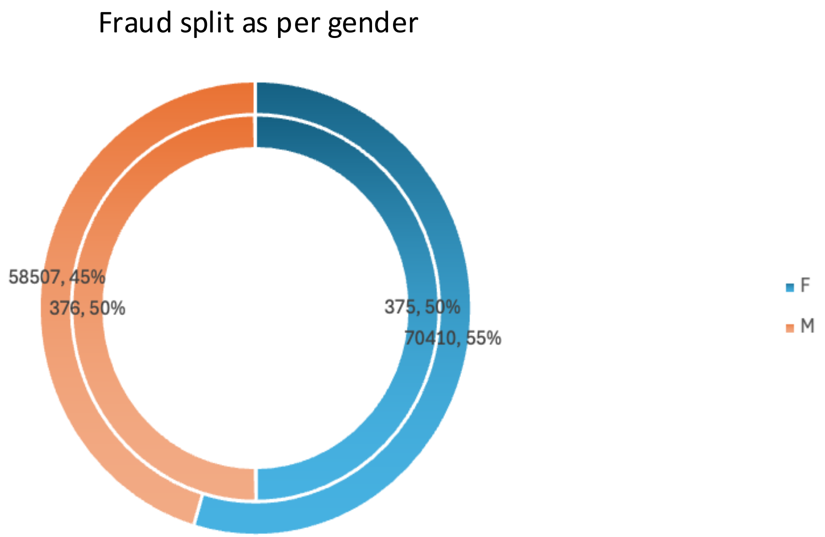
- Further scope for **feature engineering** to simplify the model
  - *Transaction frequency* - Using the time gap between the transactions as a predictor
  - *Geolocation data* - Using the distance between merchant and cardholder as a predictor

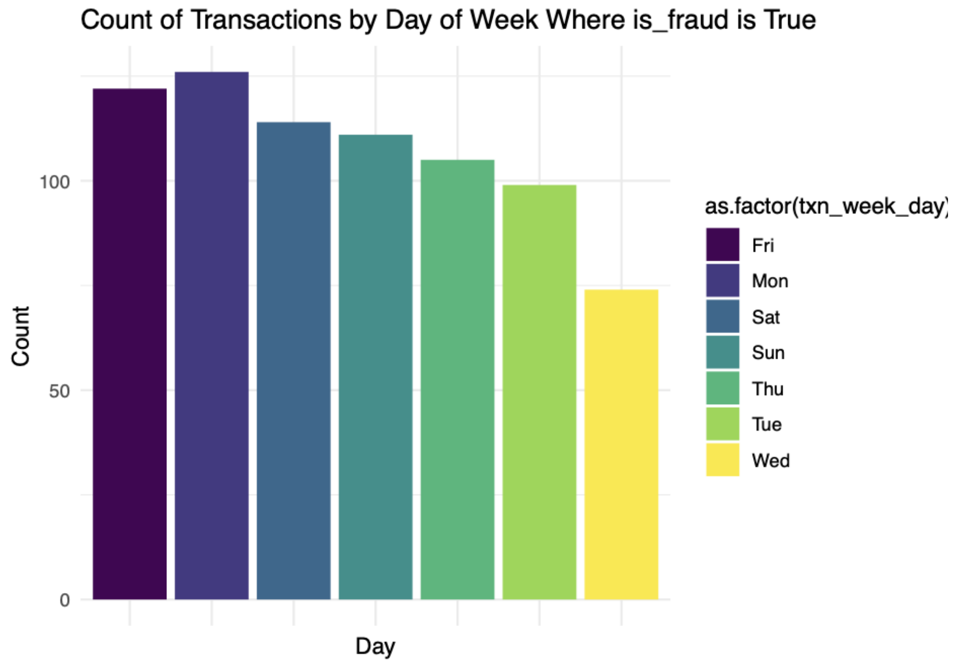
- Perform a detailed cost analysis with **different threshold** (instead of Youden’s J)
  - *Lower threshold* - Ensures high TPR for sending alerts about potentially fraudulent activity
  - *Higher threshold* - Ensures high precision for immediately blocking the card

Questions

Appendix 1.1 - About the dataset

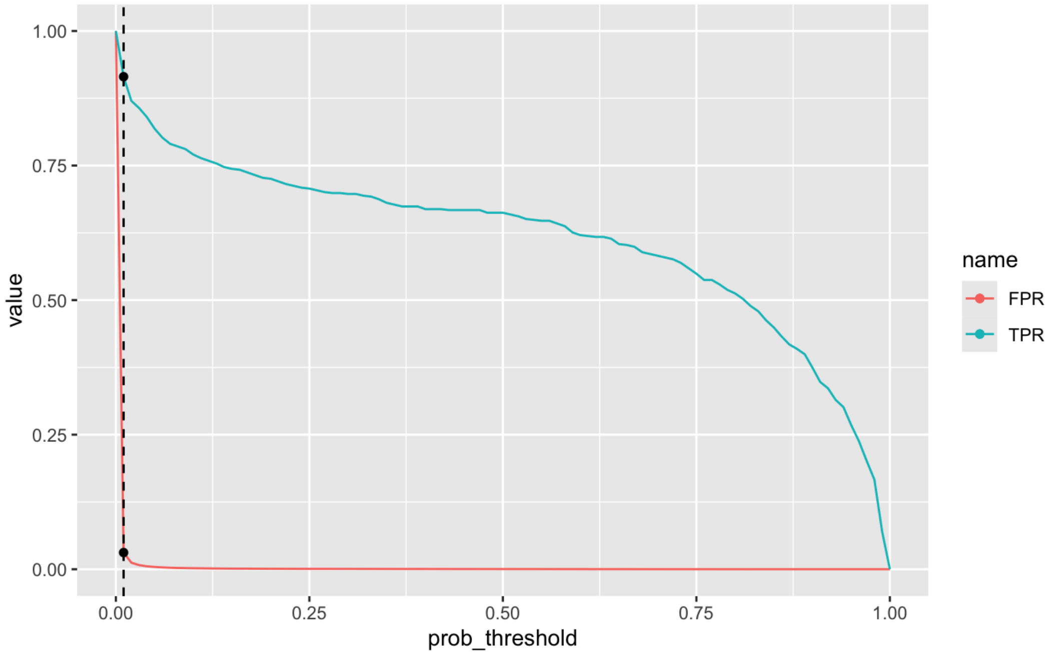
- Gender Wise Split is identical in both Fraudulent and Legitimate Transactions
- Consistent Fraud Activity across all the days of the week



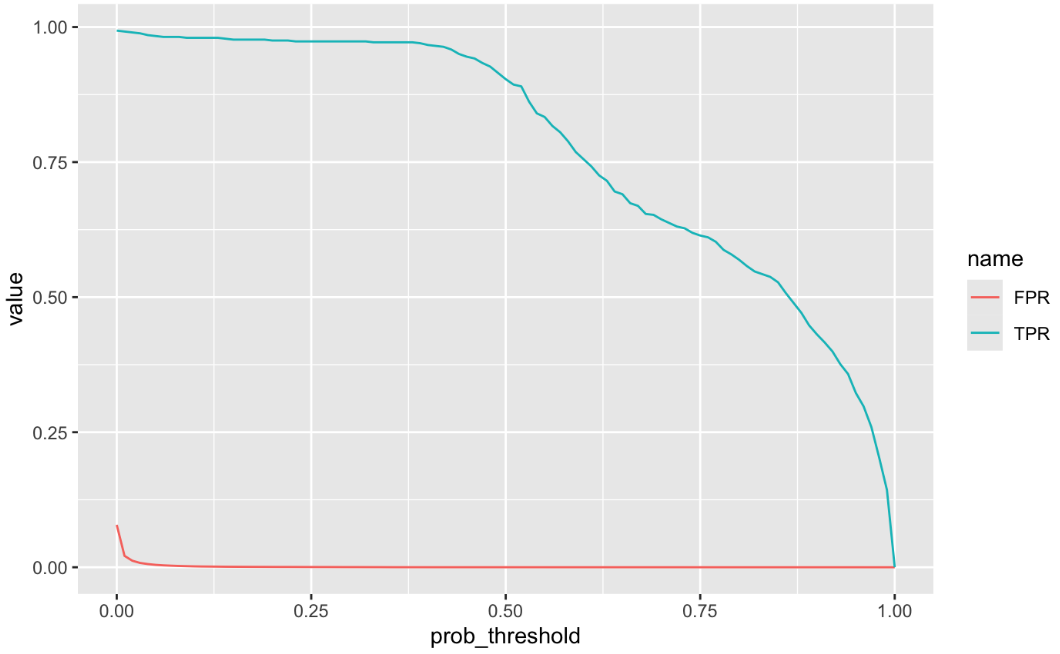


Appendix 1.2 - TPR/FPR comparison (on training data)

Gradient Boosting



Random Forest



FPR is very low across all thresholds - similar to the trend in boosting

Might need to pick different threshold as per the business use case