Fraud Detection Within The Hut Group

Thomas Pinder, Nicholas Abad, Julie Sun, Omar Khan, Luke Lorenzi, Mengnan Sun

Data Science Audience

Project Background

- What is The Hut Group?
  - E-commerce company that sells a wide range of products to customers all over the world
- Why is detecting fraud important?
  - Helps limit the amount of money lost through fraudulent behaviour
- How does the fraud process work at The Hut Group?
  - Automated program flags potential fraud
  - Potential fraud referred on for manual investigation

Aims & Objectives

Aims

- Identify key variables associated to fraudulent activity
- Produce a classifier to identify a transaction as fraudulent

Objectives

- Engineer a set of new variables
- Quantify variable importance
- Construct and compare logistic regression and random forest models
    - Aim to maximise precision

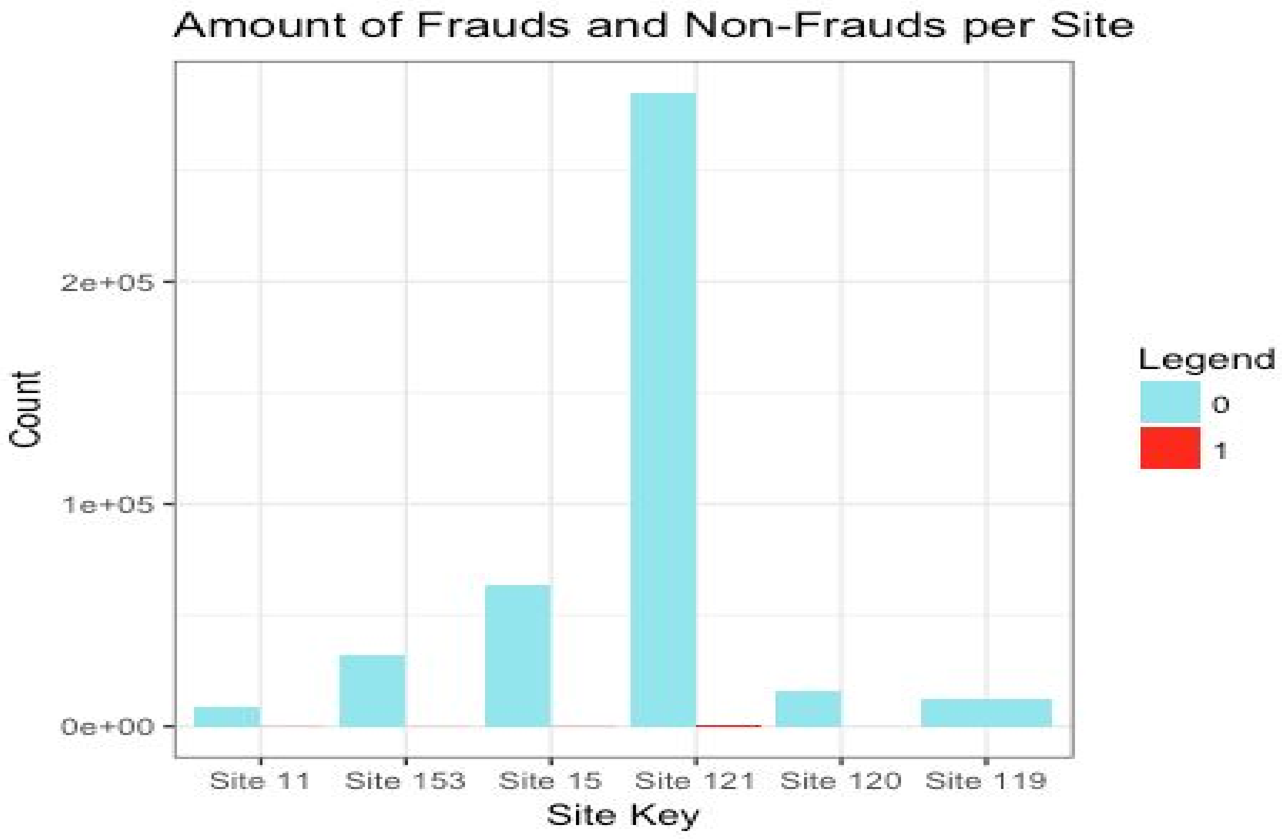
Overview of steps taken to detect fraud

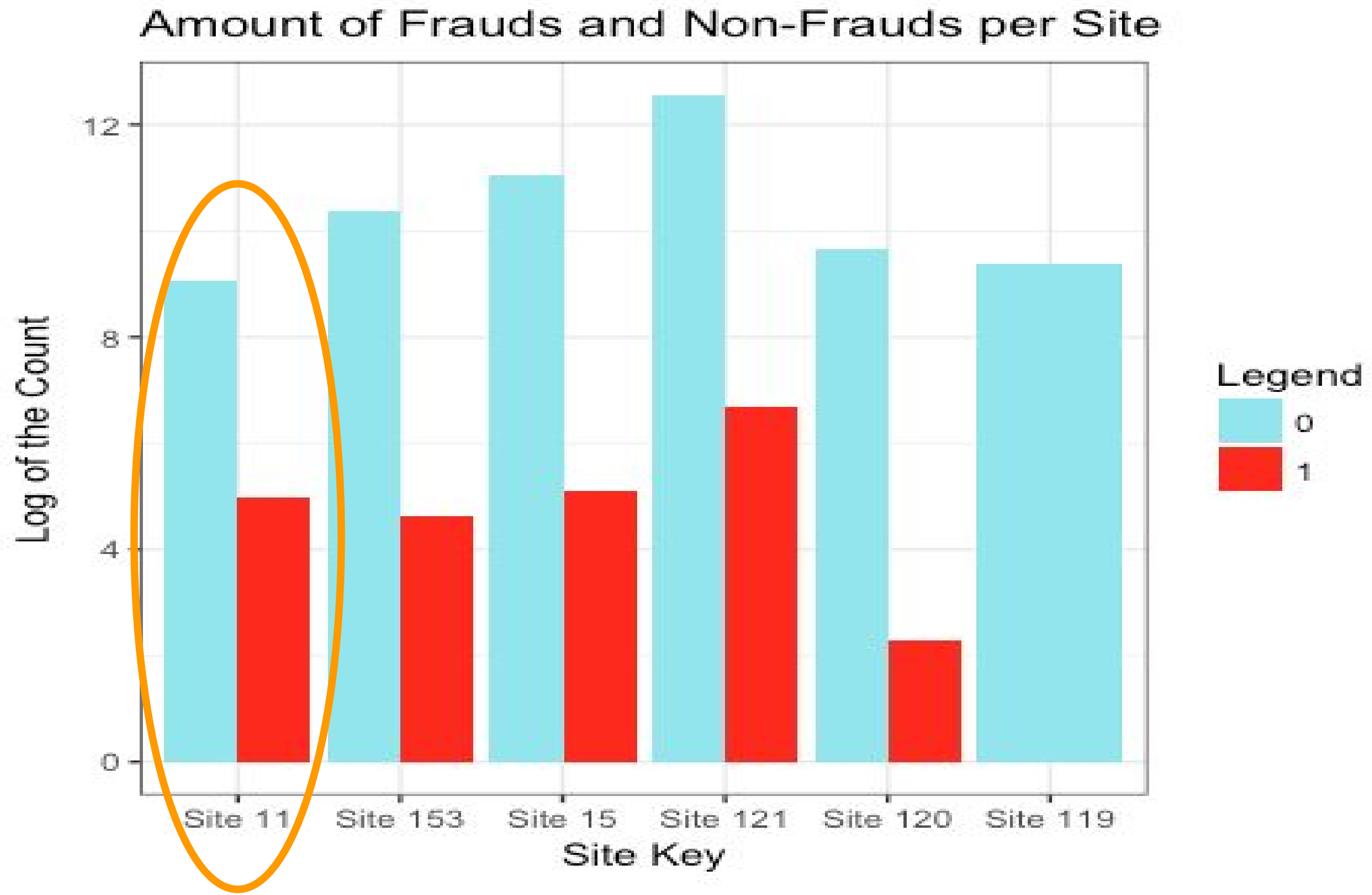
- Discuss fraud with The Hut Group
- Explore existing fraud detection methods
- Decide on technologies to use
- Feature engineering & exploratory analysis
- Determine feature importance
- Account for class imbalances
- Model using logistic regression and random forests

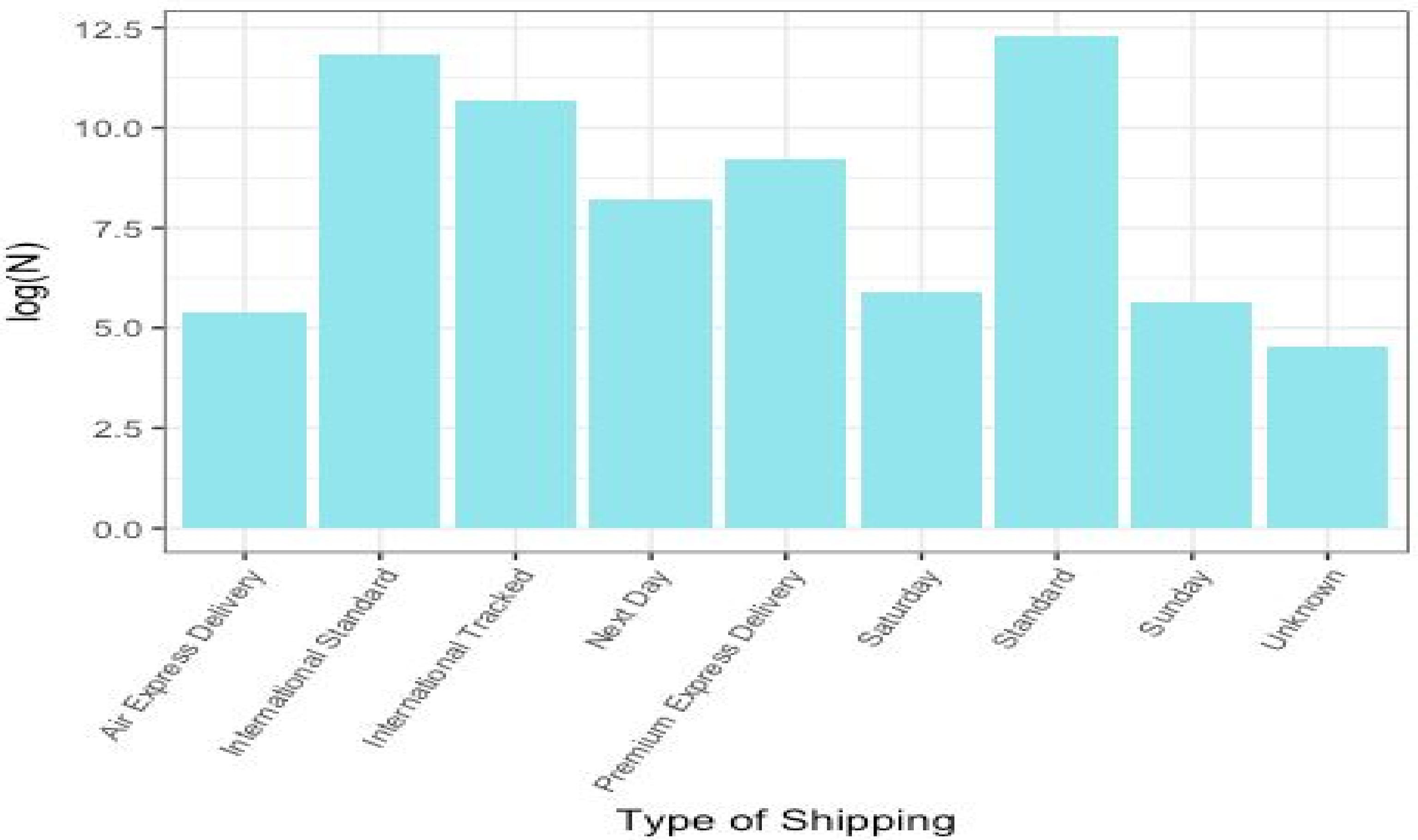
Understanding the Data and Current Fraud Research

First Steps

- 3 primary tables & auxiliary lookup tables
  - 418,000 observations within a 3-month period
  - 0.3% fraud rate
- Not much publicly available literature on current fraud research
- Decided on Git, Python, R and Google Slides

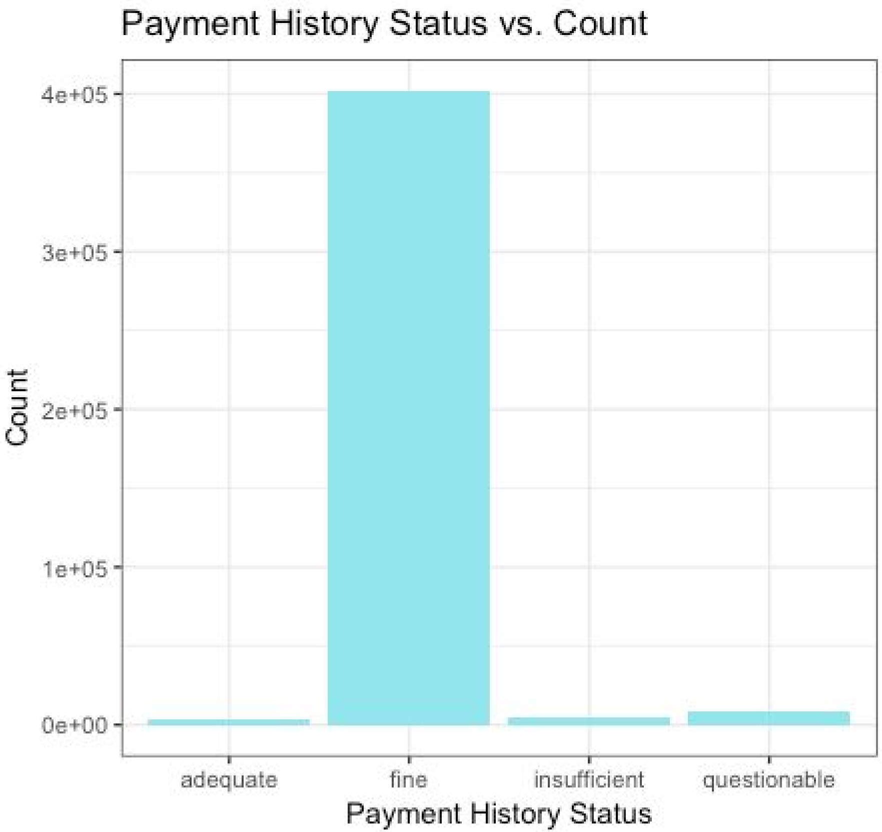

Amount of Frauds and Non-Frauds per Site
Count
2e+05
1e+05
0e+00
Site 11
Site 153
Site 15
Site 121
Site 120
Site 119
Site Key
Legend
0
1

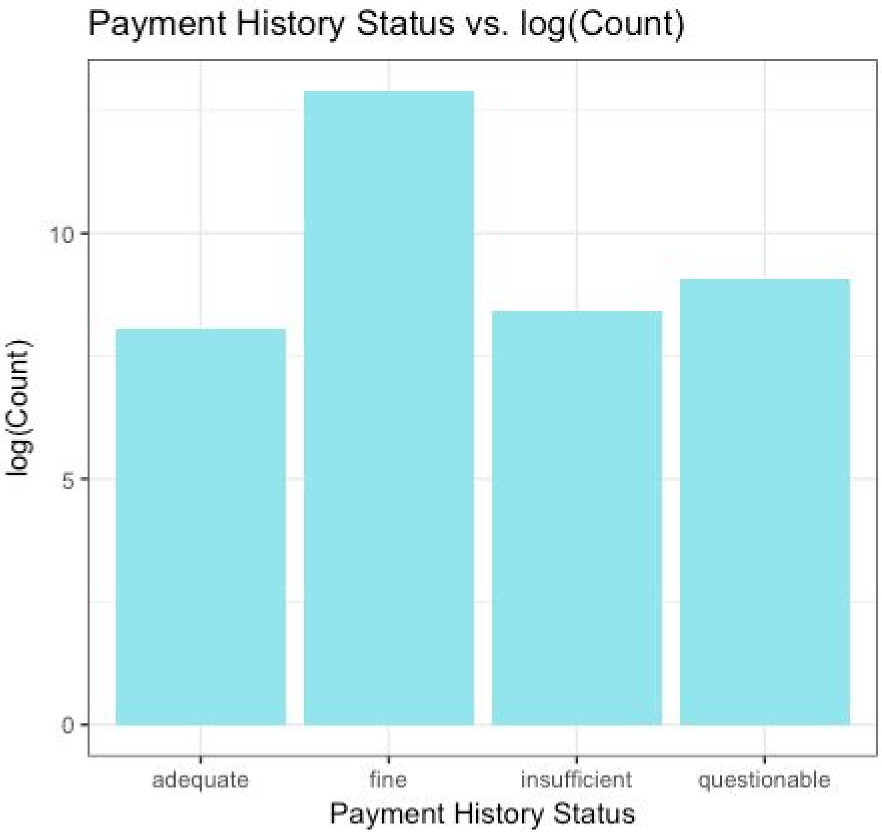

Amount of Frauds and Non-Frauds per Site
Log of the Count
12
8
4
0
Site 11
Site 153
Site 15
Site 121
Site 120
Site 119
Site Key
Legend
0
1



12.5
10.0
7.5
5.0
2.5
0.0
log(N)
Air Express Delivery
International Standard
International Tracked
Next Day
Premium Express Delivery
Saturday
Standard
Sunday
Unknown
Type of Shipping

Cleaning Data and Feature Engineering

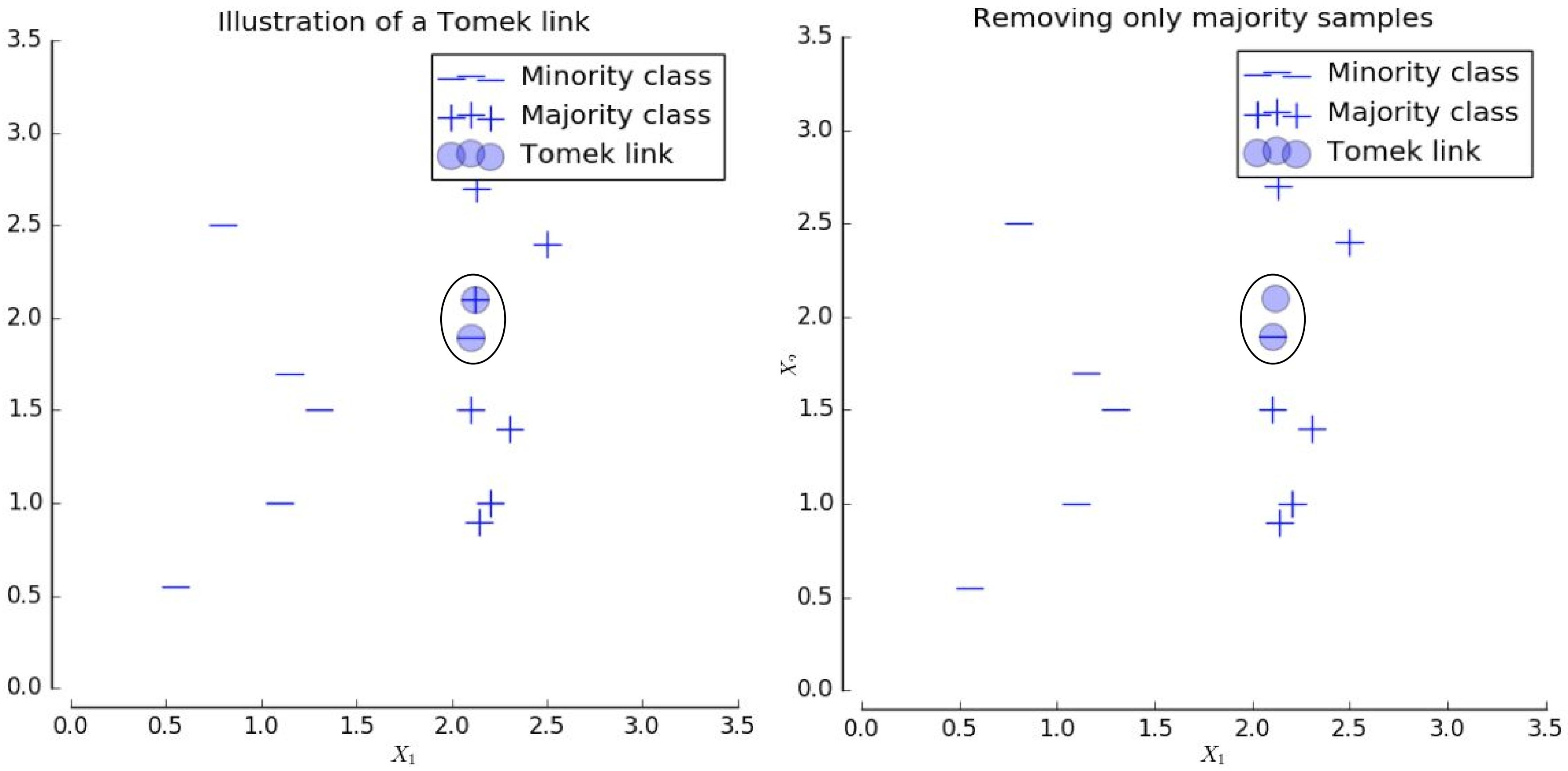

Payment History Status vs. Count
Count
4e+05
3e+05
2e+05
1e+05
0e+00
adequate
fine
insufficient
questionable
Payment History Status


Payment History Status vs. log(Count)
log(Count)
10
5
0
adequate
fine
insufficient
questionable
Payment History Status

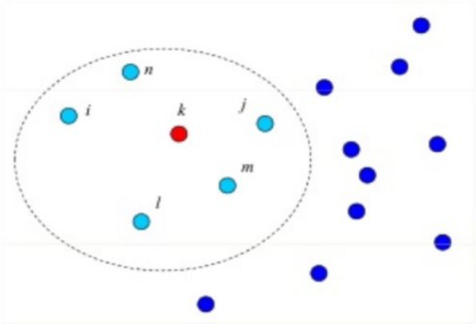
- Examples of *important* new variables created
  - Proportion of cancelled orders
  - Customer status
  - International delivery boolean
  - Priority delivery boolean
  - One-hot encoded product category

- Problems with data:
  - Missing data and received NA for several values
    - Checked if fraudulent
    - If non-fraudulent, observation dropped
  - Class Imbalance
    - Tested random under and over-sampling
    - Found SMOTE with Tomek links best

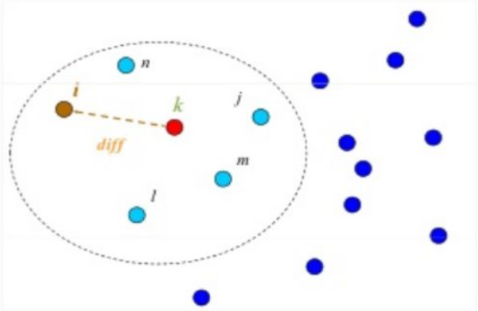
Modeling



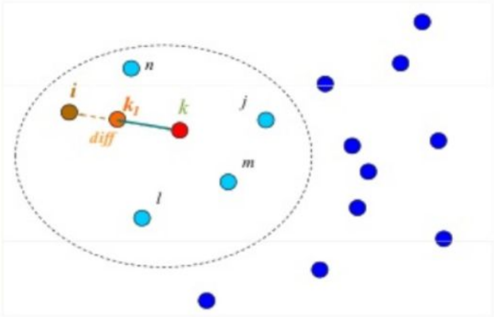
Code for graphs sourced from Sci-Kit Learn - https://tinyurl.com/y8fon3eg



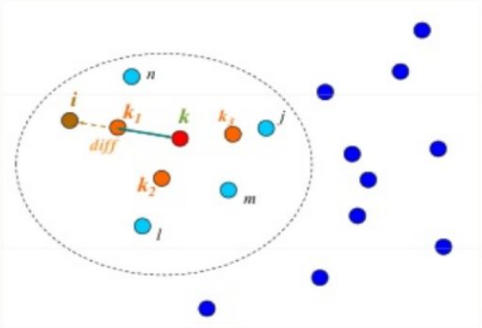
1. For each minority example k compute nearest minority class examples (i, j, l, n, m)



2. Randomly choose an example out of 5 closest points



3. Synthetically generate event k_1, such that k_1 lies between k and i



4. Dataset after applying SMOTE 3 times

Graphs sourced from SCC403 Lecture 9

Choice of Classifiers

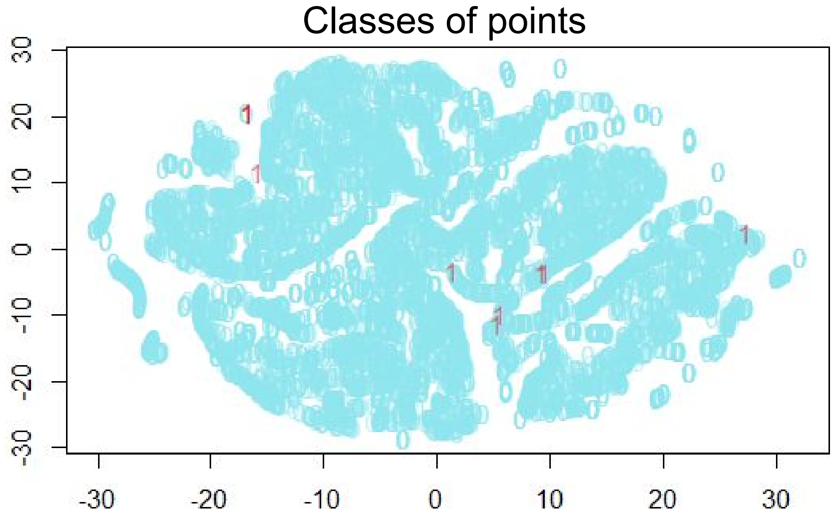
- Logistic regression
  - Easy to interpret model output
  - Frequently used in fraud literature
- Random Forest
  - Reduces chance of overfitting
  - Allows for key variables to easily be identified

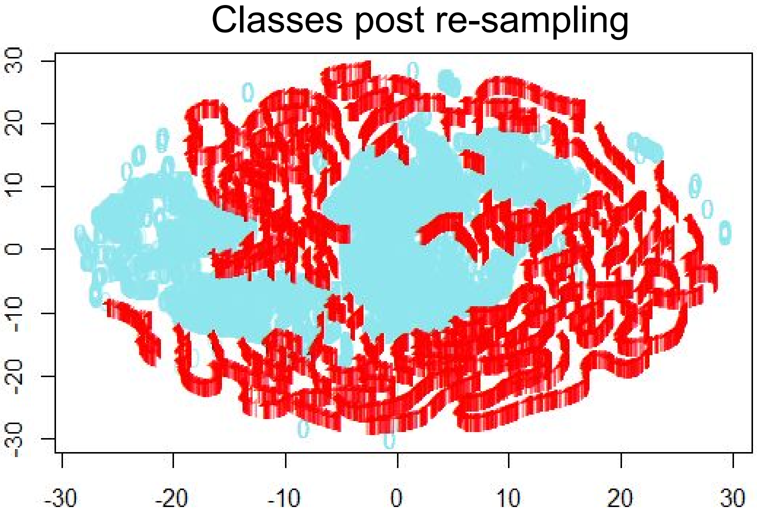
Steps of Analysis

- Stratified on the 5 site keys
- Split data into train and test, 60:40 split
- Applied SMOTE + Tomek links to training split
- Ran classifier on training
- Assess performance using 10-fold cross-validation
- Remove unimportant variables, tune and re-fit
- Tested model on testing split
- Computed model metrics

Results

Results of SMOTE + Tomek Links





0: Majority Class

1: Minority Class

	Accuracy	Recall	Precision	F-Score	AUC
Logistic Regression	98.3	0	0	0	76.3
Random Forest	98.8	43.7	**73.8**	54.8	92.9
Logistic Regression (with SMOTE + Tomek)	89.6	53.5	8.4	14.5	78.3
Random Forest (with SMOTE + Tomek)	**98.9**	**59.2**	71.2	**64.6**	**95.0**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

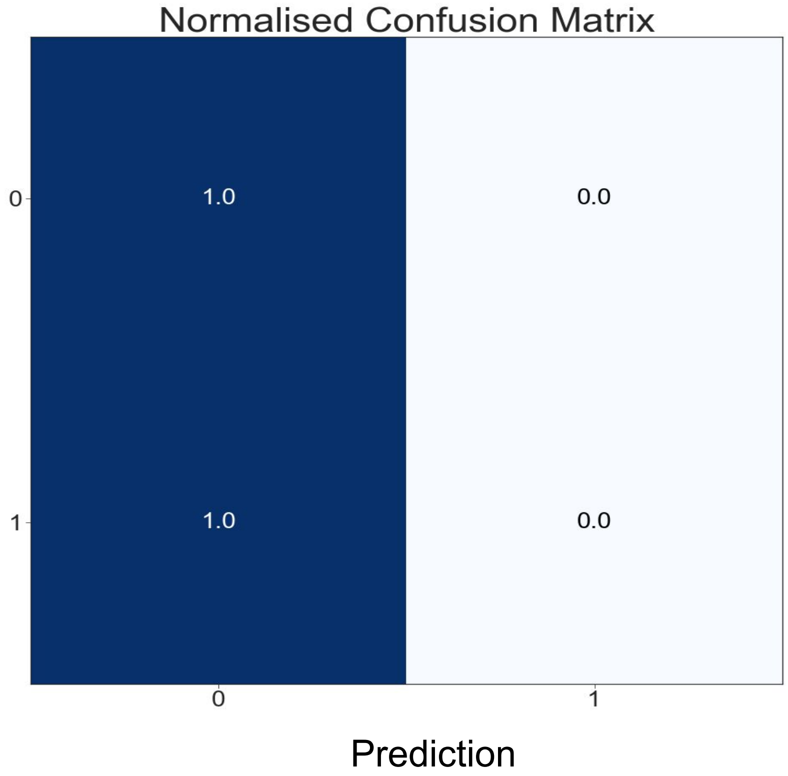
$$F - Score = 2 \star \frac{Precision \star Recall}{Precision + Recall}$$

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

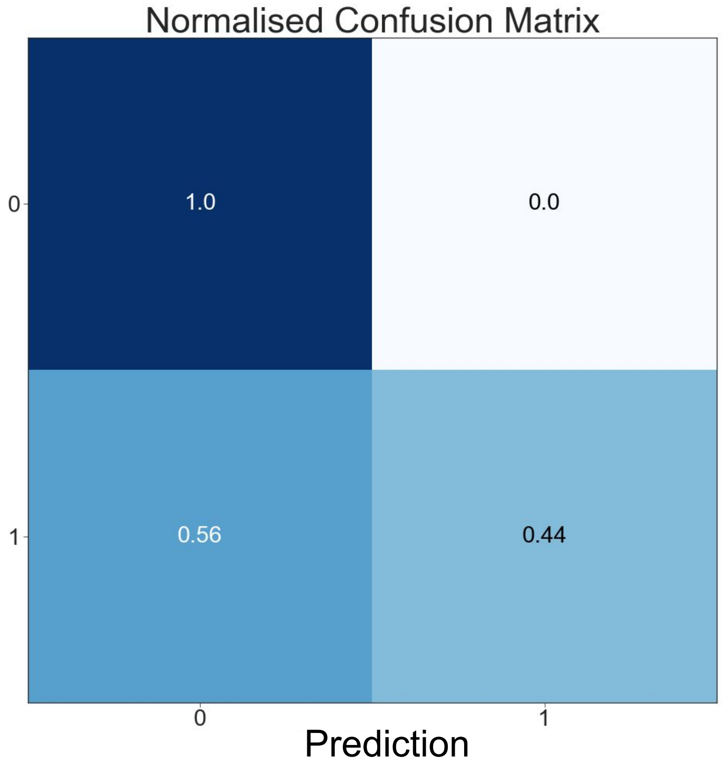
$$Precision = \frac{TP}{TP + FP}$$

Normalised Confusion Matrix

Logistic Regression

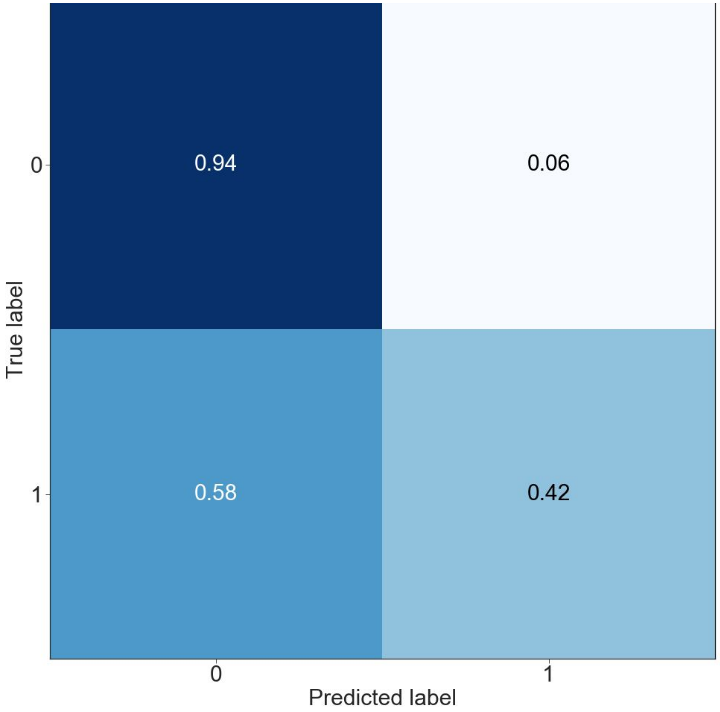


Random Forest

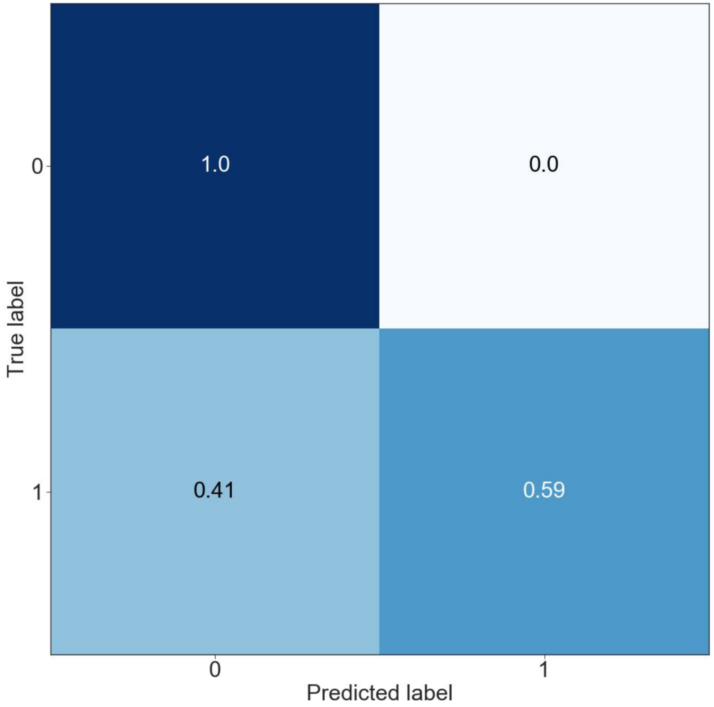


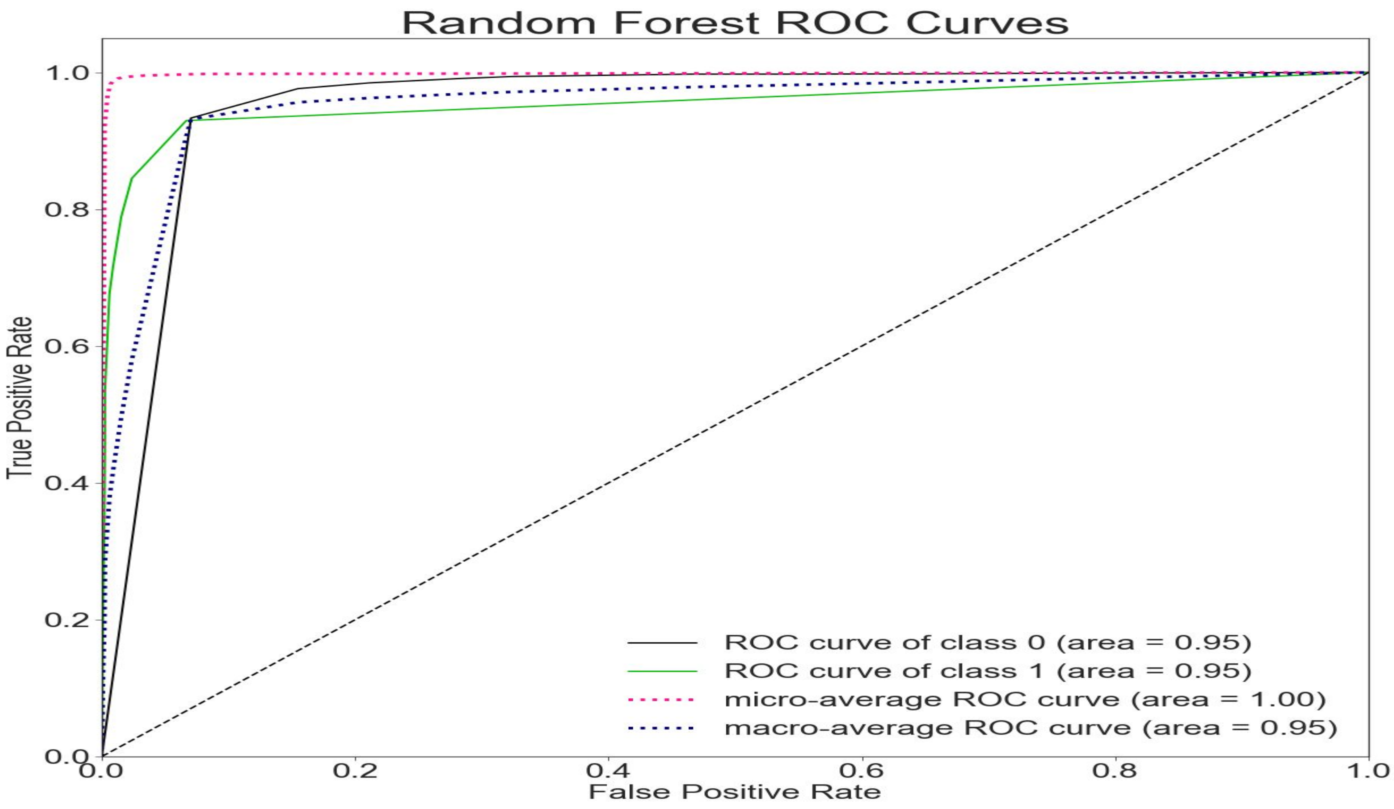
Normalised Confusion Matrix With SMOTE + TOMEK

Logistic Regression



Random Forest





Random Forest ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class 0 (area = 0.95)
ROC curve of class 1 (area = 0.95)
micro-average ROC curve (area = 1.00)
macro-average ROC curve (area = 0.95)

Bias and Validity concerns

Class Imbalances

- Under sampled non-fraudulent & over sampled fraudulent transactions
  - SMOTE + Tomek links
  - Can “cheat” the classifier
  - Only re-sampled training data

External Validity & Generalisability

- Only received data during a three month period
- Excluded Periods of High Fraud
  - Black Friday, Cyber Monday, Christmas...
- Stratifying on the 6 unique account keys rather than using the entire data set

Measurement Error

- Cannot Guarantee All Fraud Is Accounted For
- Variables strongly correlated to fraud may not have been measured
  - User time on site, user behaviour

Conclusions & Further Work

Conclusions

- Key variables
  - Payment method
  - Charge price
  - Proportion of cancelled orders
- Random forest performed best
  - Re-sampling minority class via SMOTE with Tomek links enhanced performance

Further Work

- Test different classifiers
- Introduce new cost function
  - Account for charge price in cost function
- Obtain full year data
  - Test for seasonality and trends across time

Thank You For Listening.

Are There Any Questions?