

The background of the slide features a semi-transparent world map in a light blue color, centered over a dark blue sky. Below the map, a vibrant city skyline at night is visible, with numerous skyscrapers illuminated in various colors like yellow, orange, and blue. The city lights are reflected on a body of water in the foreground, which also shows some small boats. The overall composition is a blend of global connectivity and urban technology.

25Credit

Big Bytes 2019

Meet Team Jingle Jangle



Hong Jun

Nanyang Technological University
Accountancy
Year 2



Jiang Shen

Nanyang Technological University
Business Analytics & Accountancy
Year 2



Jordan

Nanyang Technological University
Renaissance Engineering Program
Computer Science
Year 2

Agenda

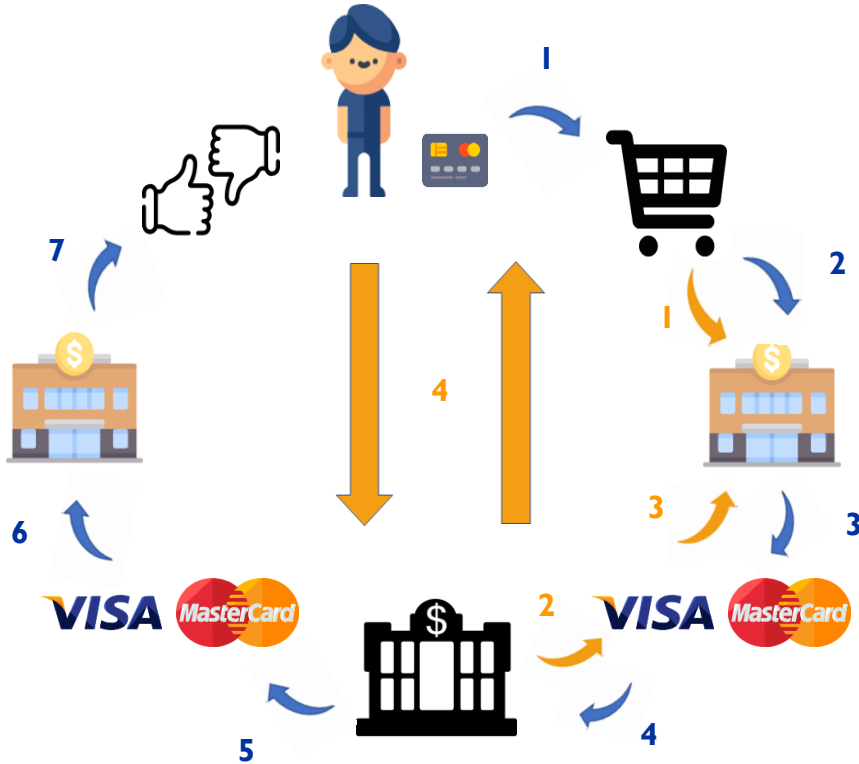
- 1 Business Problem
- 2 Data Exploration
- 3 Costing Analysis
- 4 Performance Metric
- 5 Model Building
- 6 Revisit Our Approach
- 7 Business Recommendations & Deployment
- 8 Limitations & Further Research



Problem Statement

“What are the solutions to reduce the time and effort for investigations as well as improve on the fraud recovery through early identification and proactive management of suspicious transactions?”

How Credit Card Works



Authorisation

1. Customer pays with credit card
2. Merchant transmits the credit card information and details of transaction to its acquiring bank.
3. Acquiring bank (or the bank processor) routes it through the appropriate card network to the cardholder's issuing bank.
4. Visa transactions are routed through Visa's VisaNet network
5. The issuing bank responds by approving or declining the transaction after checking to ensure that the transaction information is valid etc.
6. Visa helps route the response code back through its network to the acquiring bank.
7. The approval code is delivered to the merchant's point of sale device.

Clearing & Settlement

1. At the end of the day, merchant will send out its authorisation batch of transactions to the acquiring bank for money to be deposited into the merchant's account.
2. Visa debits the issuing bank's account.
3. Visa then credits the acquiring bank's account for the net amount. Essentially, the card issuing bank pays the acquiring bank for the cardholder's purchase in advance.
4. The issuing bank would then bill the cardholder on a monthly basis for his credit card expenditure.

Types of Credit Card Fraud



Friendly Fraud

Friendly Fraud occurs when a consumer makes an online shopping purchase with their own credit card, and then requests a chargeback from the issuing bank after receiving the purchased goods or services, regardless intentional or with other reasons.



Account Takeover

Account takeover occurs when a fraudster uses another person's account information (e.g., username and password) to gain full control of the account (by changing password etc), subsequently obtaining products and services using that person's existing accounts.



Unauthorised Transactions

Unauthorized transactions occurs when fraudsters use stolen card or payment credentials to pose as the customer to purchase merchandise on the stolen card or payment account.

Fig. 1.1: Average Fraud Loss by Type

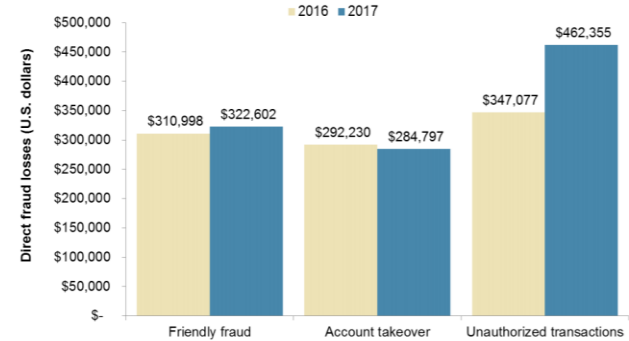
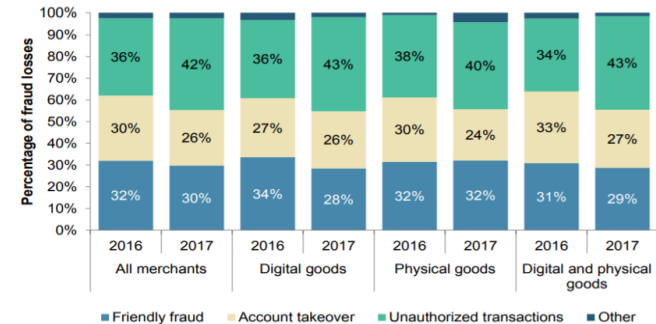


Fig. 1.2: Percentage Fraud Loss by Type



Source: Javelin Strategy & Research, 2018

Current Challenges of Credit Card Fraud Detection



Heavily Imbalanced Data

With less than 1% of fraud transactions, methods and metrics must be adjusted to meet the goals.



Time lag

Unauthorised transactions only discovered when cardholders reviewed their monthly bills



Estimation of Cost

- Difficulty in estimating intangible costs suffered in false alarm cases
- Difficulty in estimating the difference in misclassification cost between False Negative and False Positive



Non-stationarity of data

Customer spending behaviour and fraud behaviour evolve over time. E.g. Point of Sale fraud to halve but card-not-present (CNP) fraud to increase over next few years. (Javelin, 2018)



Chargeback Frauds

Merchants are forced to absorb the friendly fraud losses as it is time consuming and expensive for merchants to fight a chargeback claim.

Business Problem

Credit card frauds are so rare yet they incur massive losses for the credit card companies. The problem largely revolves around how to timely **detect frauds** without annoying the other ~99% of legitimate transactions from loyal cardholders and minimise the cost in fraud handling.



7

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

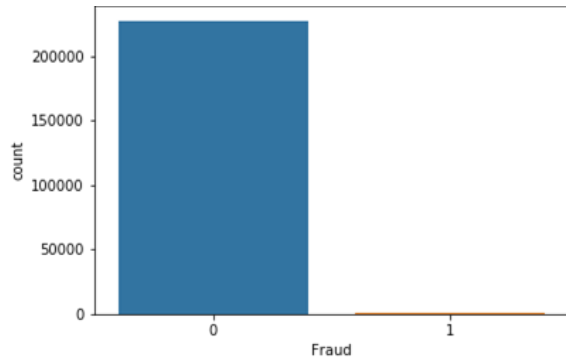
Limitation &
Further research

Data Exploration - Class, Time and Amount

Data Size = 227,844, 31 Features (PCA)

Fig. 2.1: Class Distributions

(0: No Fraud || 1: Fraud)



No frauds: 99.82% of the dataset

Frauds: 0.18% of the dataset

The dataset is imbalanced and skewed towards no-frauds. Pre-processing of the data is required.

Fig. 2.2: Time Distribution

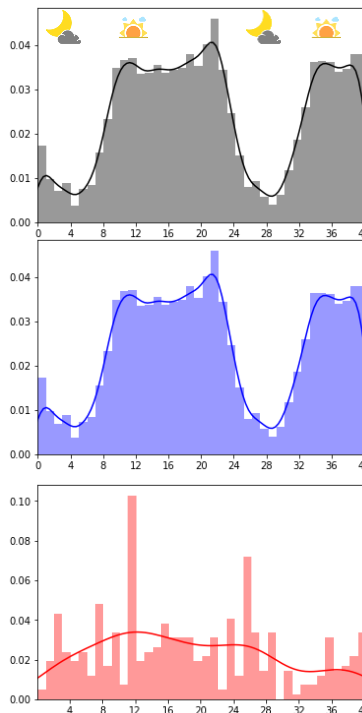
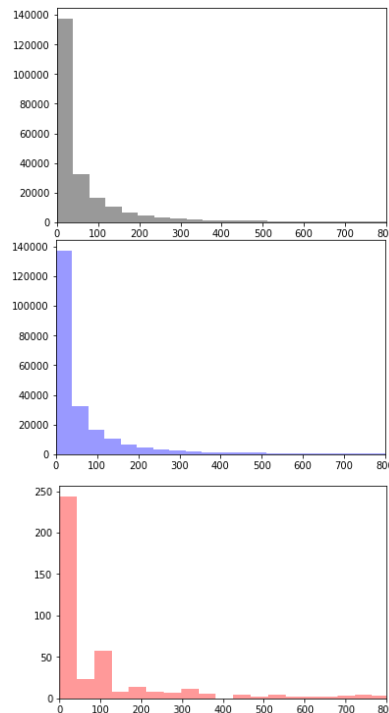


Fig. 2.3: Amount Distribution



■ All data
■ Non-Fraud
■ Fraud

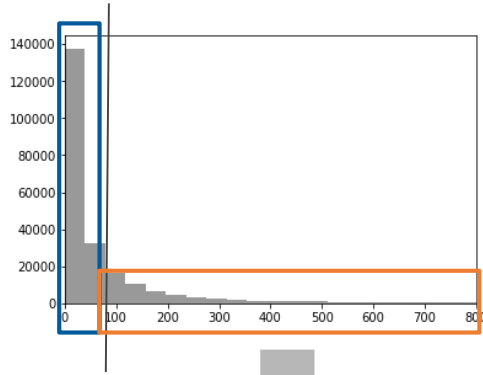
Time: There is a pattern in frequency of transaction over time for non-fraud transactions. However, no visible trend can be identified for fraud transactions. Tests are required to find out the significance of the variance.

Amount: About 80% and 70% of the transactions are below \$100 for non-fraud and fraud transactions respectively. The number of transactions decreases as the amount increases.

Pareto Principle (20:80 Rule)

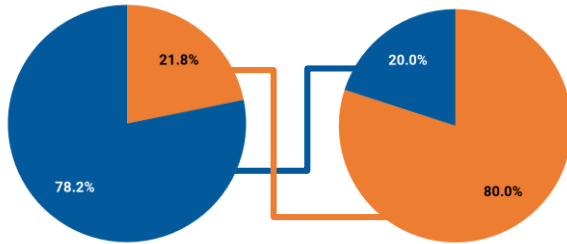
Pareto Principle

Fig 3.1 All Data Amount Distribution

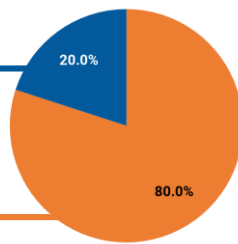


	All Data		Fraud	
	≤ \$95	> \$95	≤ \$150	> \$150
No. of Transaction	178,125 (78%)	49,591 (22%)	331 (79%)	86 (21%)
Amount	4.1M (20%)	16.6M (80%)	9.6K (28%)	43K (82%)

Transaction Frequency



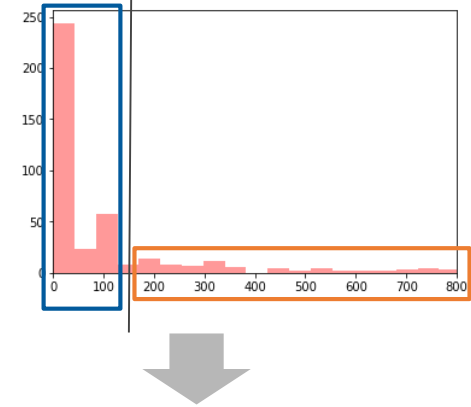
Transaction Amount



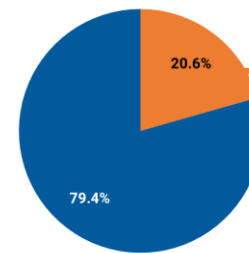
● High
● Low

Although 20% of total number of transactions may not seem as significant, they add up to 80% of total amount of transactions.

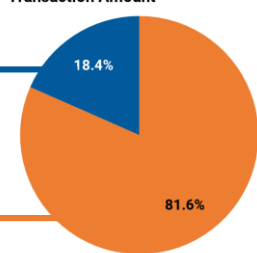
Fig 3.2 Fraud Data Amount Distribution



Transaction Frequency



Transaction Amount



● High
● Low

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

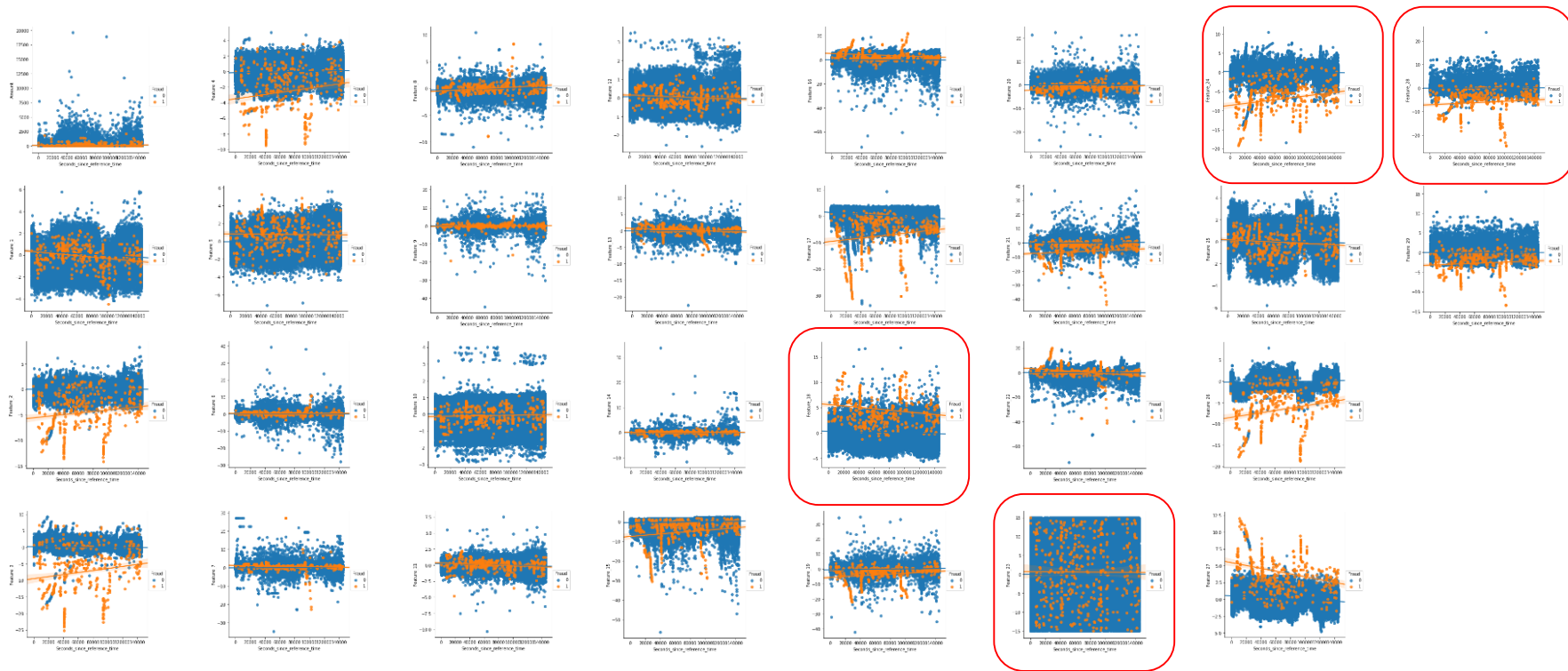
Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

Data Exploration - Feature Scatter Plots



10

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

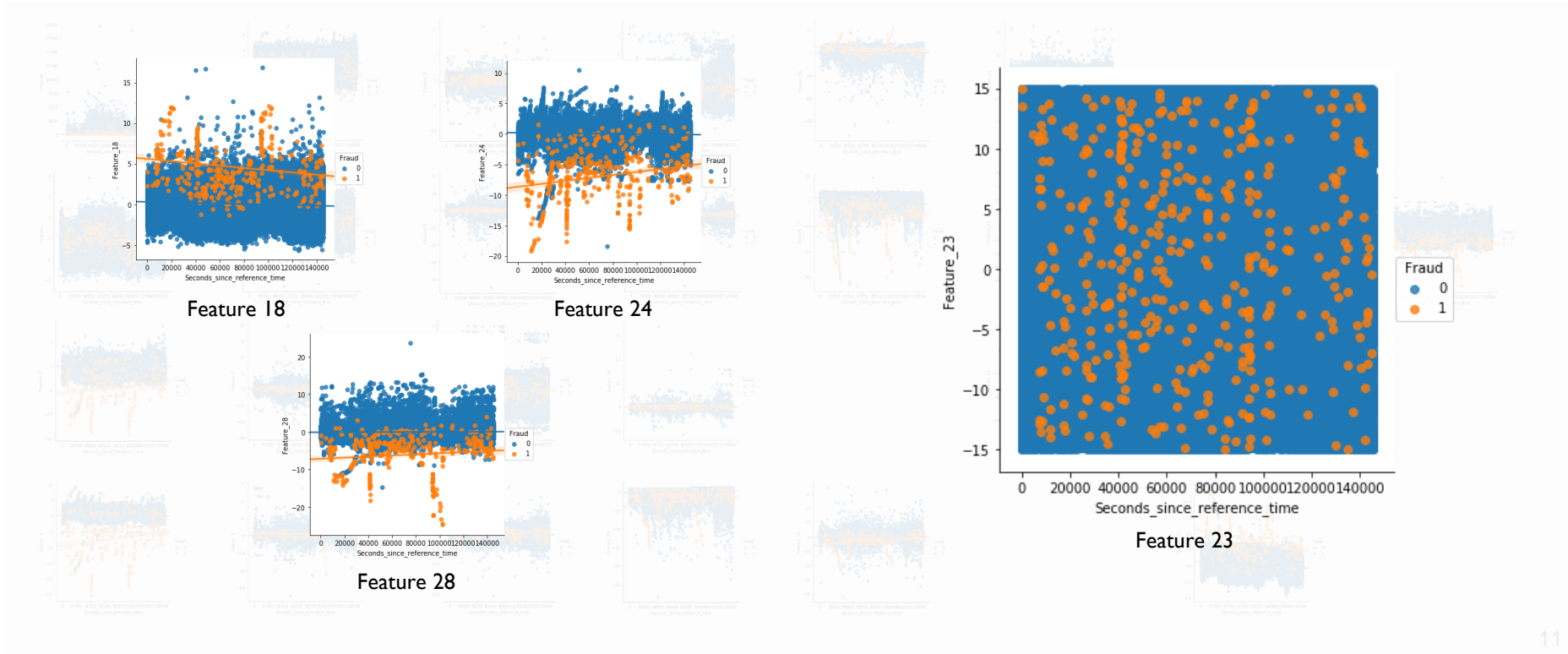
Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

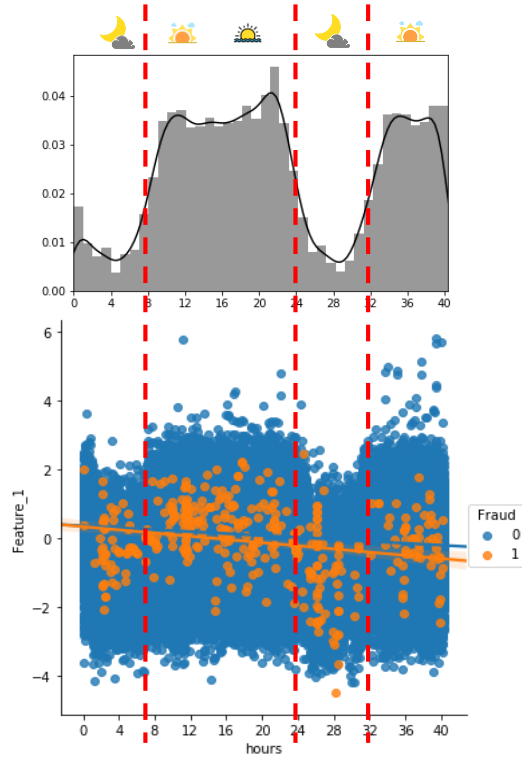
Limitation &
Further research

Data Exploration - Feature Scatter Plot

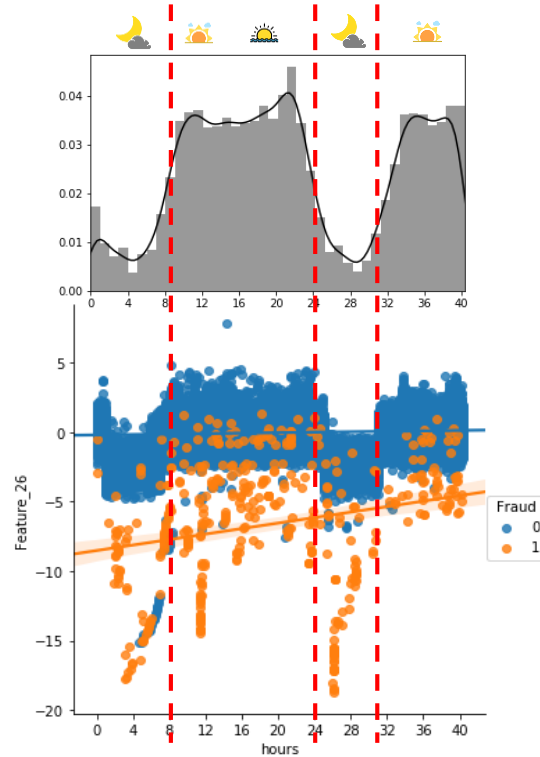


Data Exploration - Feature Scatter Plot

Feature 1



Feature 26



Data Exploration - Penny Scams

Penny Scam ($\leq \$5$)

Fig 4.1: By Frequency

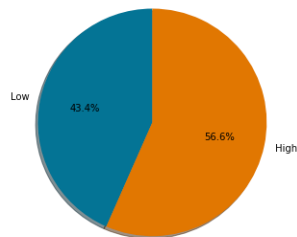
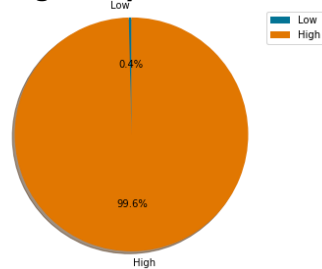


Fig 4.2: By Amount

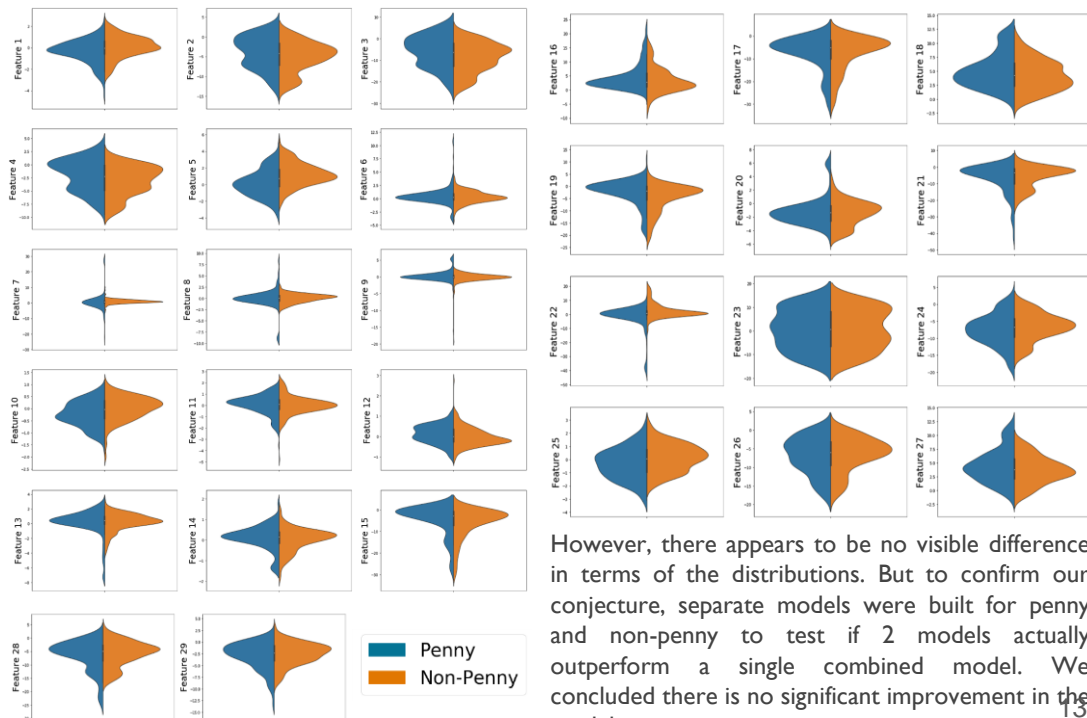


- Half of frauds are penny scams, but only adds up to 0.4% of total fraud
- Still Significant because:
- Used by fraudsters to identify the vulnerable credit card users who do not check their monthly statements thus making them easy targets for future higher value frauds (Simon, 2006)
- Cases where fraudsters stole 14,000 credit card information which saw pennies stolen from each cardholder

Different decisions will be reached depending on whether a transaction is a penny scam.

Our **hypothesis**: features differ in distribution based on whether a fraud is a penny or non-penny one.

Fig. 4.3: Comparison of distribution of 29 Features (Penny vs Non-Penny)



However, there appears to be no visible difference in terms of the distributions. But to confirm our conjecture, separate models were built for penny and non-penny to test if 2 models actually outperform a single combined model. We concluded there is no significant improvement in the models.

Misclassification Costs of FN vs FP

Cost Breakdown

		<u>Prediction</u>	
		Positive	Negative
<u>Actual</u>	Positive	<u>True Positive (TP)</u> <i>Intervention Cost</i>	<u>False Negative (FN)</u> <i>Txn Amt + Potential Reputational Damage</i>
	Negative	<u>False Positive (FP)</u> <i>Intervention Cost + Customer Frustration - Revenue</i>	<u>True Negative (TN)</u> <i>- Revenue</i>

True Positive (TP) Detected Fraud

False Negative (FN) Undetected Fraud

False Positive (FP) Wrongly Flagged Fraud

True Negative (TN) Genuine Transactions

Intervention Cost: When the model predicts True, the merchant conducts inspections on the transaction. Cost depends on the actions taken by the company. It can be as little as SMS cost and it increases as the amount of efforts spent

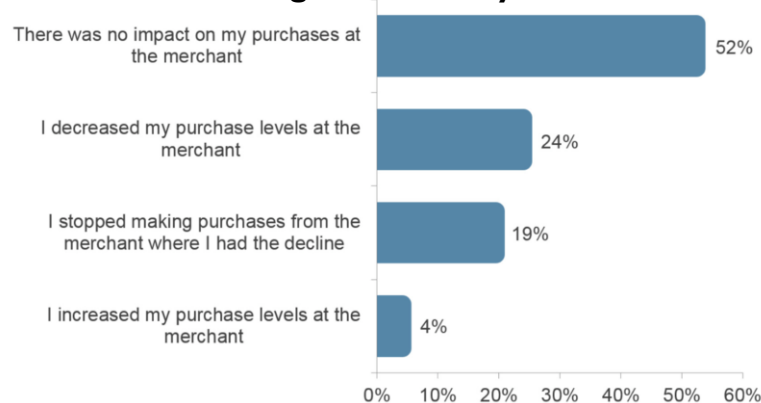
Transaction Amount: In the case of FN, the full transaction amount is refunded back to the credit card holder.

Potential Reputational Damage: When credit card company fails to detect a fraudulent transaction, the reputation may be affected, hence there is an intangible cost associated to the reputational damage.

Customer Frustration: In the case of FP, when a genuine transaction is declined, there is a potential cost to the company. According to Javelin's report (2018), 43% of the affected credit card holders reacted negatively (Figure 4)

Revenue: Revenue is earned in the cases of genuine transactions. *Formula: Transaction amount x Merchant Fee Rate*

Fig. 5: False Positive Resulting in Consumers Shifting to Secondary Card



Source: Javelin Strategy & Research, 2018

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

Costing Analysis

Computing the cost of FP and FN

Based on the dataset, there are **417** fraud transactions, amounting to **\$52,399**.

Out of these transactions, there are 182 and 235 **penny** and **non-penny** fraud transactions respectively and they amount to \$198.20 and \$52,200.89.

Therefore, **average transaction amount** is \$1.09 and \$222.13 per transaction for penny and non-penny respectively.

Assumptions

- Intervention for penny frauds is in the form of SMS push notification where each costs \$0.05
- Merchant rate charged by 25Credit is 2% of transaction amount
- Negligible customer frustration cost incurred.
- Ignore any cost incurred from reputational damage for now.

	Penny Frauds (≤ \$5)		Non-Penny Frauds	
False Negative / Positive	FN	FP	FN	FP
Transaction Amount	\$1.09	-	222.13	-
Intervention Cost	-	\$0.05	-	\$5
Merchant Fee	-	(\$0.02)	-	(\$4.44)
Customer Frustration Cost	-	\$0	-	\$0
Potential Reputational Damage	\$0	-	\$0	-
Avg Cost Incurred	\$1.09	\$0.03	\$222.13	\$0.56

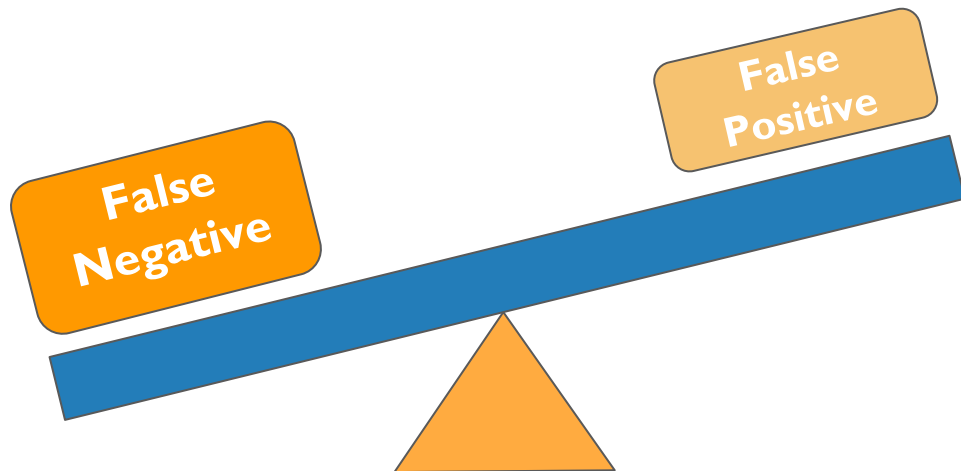
Naturally, **non-penny frauds** incurs higher and more significant intervention. Customer frustration cost is likely to be significant as well, but we will assume 0 for now to simplify our calculations.

Choice of Performance Metric

It is apparent from the table that misclassification cost incurred for a FN case is significantly higher than that for a FP case, regardless of the type of fraud involved.

Indeed, contextually fraud cases that go undetected incur the heaviest cost on the credit card company.

Hence, recall score will be the key performance metric of our ML model for now, while keeping an eye on minimizing the total cost for 25Credit.



Choice of Performance Metric (Cont')

TPR & FPR

$$\text{TPR (sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR (1-specificity)} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Low False Positive Rate (FPR) masks the lacklustre precision score and high number of FP occurrences.

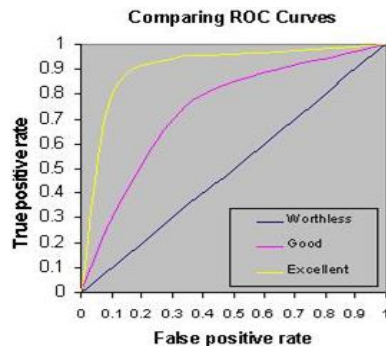
Refer to Appendix 3 for detailed calculations

F1 Score

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Harmonic average between recall and precision, that incorporates both recall and precision. However, due to **very skewed** cost ratio of FP : FN, we will focus on maximising recall for now.

ROC AUC



Does not account for prevalence or different misclassification costs arising from false-negative and false-positive.

F-beta score/ F_2

- A weighted harmonic mean of Recall and Precision
- Gives heavier weight to Recall
- Current cost ratio of FP : FN too skewed towards FN
- How to determine the appropriate weight ratio? Arbitrary?

17

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach





Business
Recommendation
& Deployment

Limitation &
Further research

Data Preprocessing

- Remove NaN values in *Features 5 and 6* by replacing with mean of each feature.
- Scaled *Amount* using mean and std. dev.

Various Rebalancing Techniques

			
Undersample Majority	Oversample Minority	SMOTE	EasyEnsemble + SMOTE
<ul style="list-style-type: none">• Throwing away potentially important info about the non-fraud cases	<ul style="list-style-type: none">• Introducing duplicate instances, could lead to model overfitting	<ul style="list-style-type: none">• Artificially synthesizing minority instances in its neighbourhood	<p>An integrated approach comprising EasyEnsemble training a learner for each subset and SMOTE to generate minority class</p>

Number of NaNs by variables

Seconds elapsed	0
Amount	0
Fraud	0
Feature 1	0
Feature 2	0
Feature 3	0
Feature 4	0
Feature 5	27
Feature 6	59
Feature 7	0
Feature 8	0
Feature 9	0
Feature 10	0
Feature 11	0
Feature 12	0
Feature 13	0
Feature 14	0
Feature 15	0
Feature 16	0
Feature 17	0
Feature 18	0
Feature 19	0
Feature 20	0
Feature 21	0
Feature 22	0
Feature 23	0
Feature 24	0
Feature 25	0
Feature 26	0
Feature 27	0
Feature 28	0
Feature 29	0

Data Leak - A Cause For Concern



“When the data you are using to train a machine learning algorithm happens to have the information you are trying to predict.”

Preprocess the whole dataset (e.g. MinMax Scaler)



Train/Test split

Problem

- Test data is ‘leaked’ into training data as the scaling occurs based on the distribution of the whole dataset
- Test data should be information hidden from our model when training
- Create overly optimistic/“too good to be true” models that are practically useless and give dismal predictions on completely new unseen data



```
# train_set:test_set = 80:20
train_data_value, test_data_value_clean, train_data_target, test_data_target
    = train_test_split(data_value, data_target, train_size=0.8)

# train and test set processed separately
train_data_value = preprocessing_data(train_data_value)
test_data_value = preprocessing_data(test_data_value_clean)
```

**Preprocess the data separately
(scaling, compressing etc.) to
provide an accurate assessment of
the model performance.**

19

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

Model Pipeline

SMOTE

Generates synthetic minority instances
in each subset to match majority class.

New samples generated are not
duplicated - they are the average of
their k-Nearest Neighbours.

EasyEnsemble

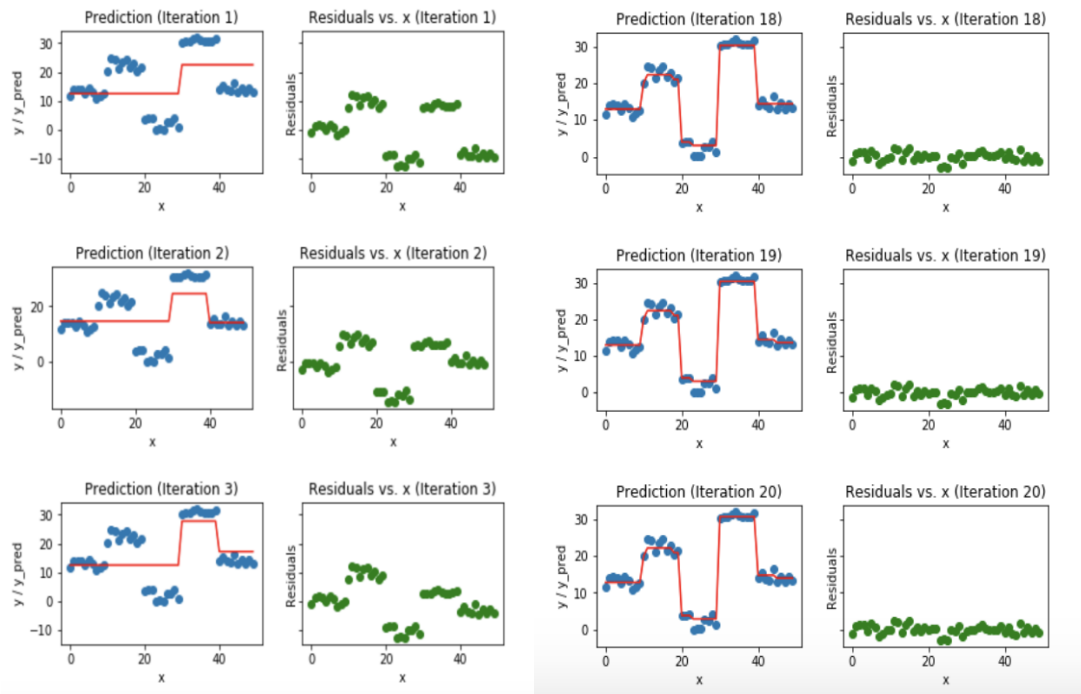
Sample the dataset to form several subsets with
smaller differences in number of minority and
majority instances. On top of random
undersampling of the majority class in each
subset, we used SMOTE to upsample the
minority class.



Training using ensemble of XGBoost

A different XGBoost is trained using each data subset and
the outputs of these learners are then combined to form
the overall model.

What is XGBoost?



Credits: <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>

- Sequential classifiers (Decision Tree)
- Contains a regularized (L1 and L2) objective function that combines a convex loss function and penalty term for model complexity
- Training runs iteratively
- Where new trees are created to predict the residuals (errors) of prior trees
- Then combined with previous trees to make the final prediction
- Boosting effectively combines these many weaker learners into a strong learner
- That lowers both bias and variance

Other Model Considerations

	<i>Ideal</i>	Naïve Bayes	Non-linear SVM	Neural Network	Random Forest	XGBoost
Interpretability	<i>Yes</i>	Yes	No	No	Yes	Yes
Predictive Power	<i>High</i>	Low	Medium	High	Medium-High	High
Training Speed	<i>Fast</i>	Fast	Medium	Slow	Medium	Medium
Amount of Training Data Required	<i>High</i>	Low	Medium	High	Medium	Medium

Why XGBoost?



Parallelisation

Allows for distributed computing which greatly reduces training time for large and growing datasets. Reduces computing resources required for the models.



Continued Training

XGBoost can be refitted with new incoming data when necessary. As the nature of transactions shifts to CNP and fraudsters deploying novel methods to conduct fraud, models have to be retrained. XGBoost allows for continued training to account for changing conditions.



Ability to identify feature importance

The importance of each feature in the model's decision making process can be identified through XGBoost weightages. Businesses can use these weightages to discover valuable insights and come up with targeted solutions.

Fig. 6.1: CNP Fraud to Dominate Fraud Landscape

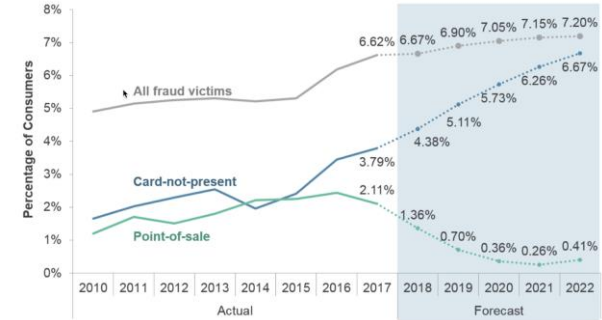
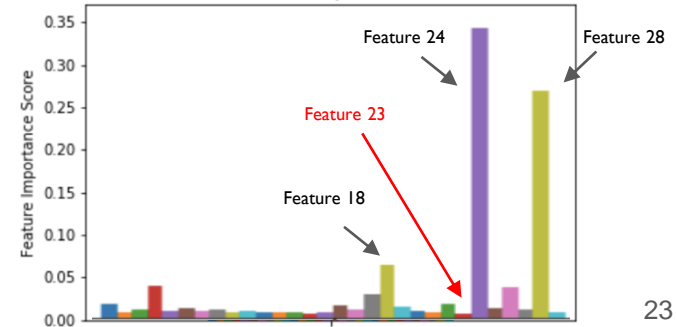
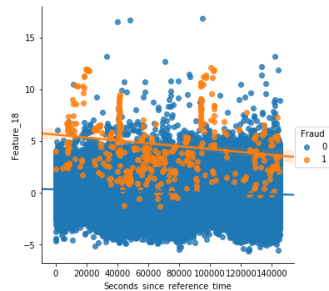


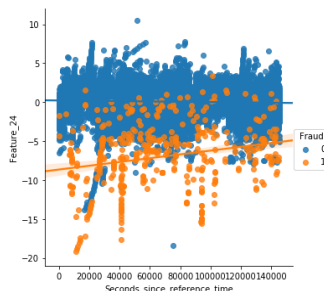
Fig. 6.2: Feature Importance



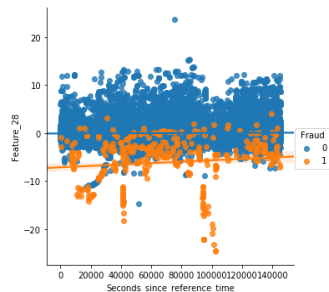
Why XGBoost? (Cont')



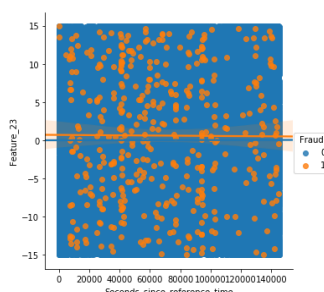
Feature 18



Feature 24



Feature 28



Feature 23

Fig. 6.1: CNP Fraud to Dominate Fraud Landscape

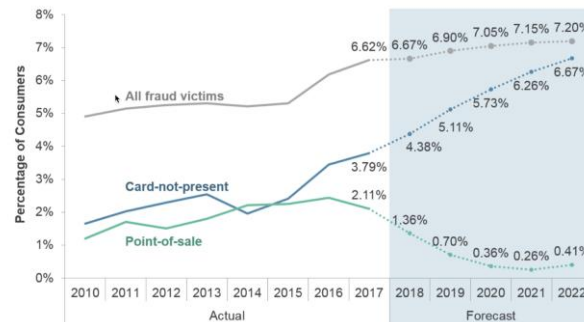
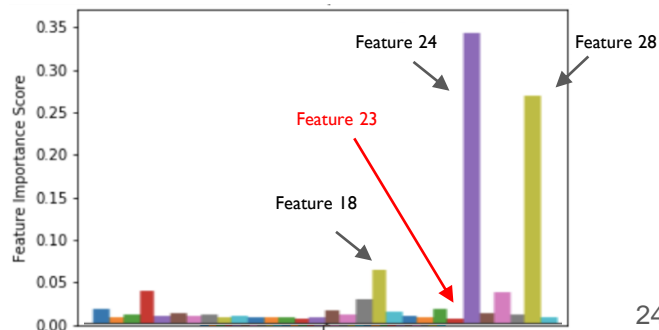


Fig. 6.2: Feature Importance



Hyperparameter Tuning

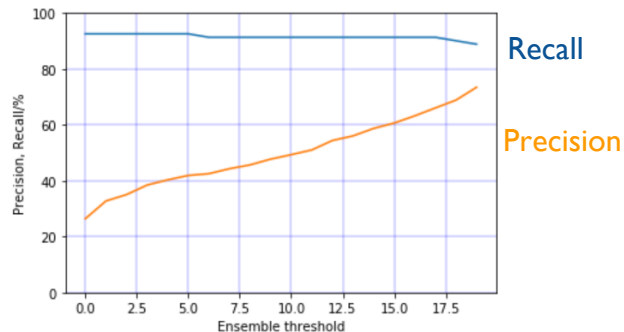
Grid Search		
	Coarse	Fine
Max. Depth	3, 5, 7, 9	8, 9 , 10, 11
Min. Child Weight	1 , 3, 5, 7, 9, 11, 13	1 , 2
Gamma	0.0 , 0.1, 0.2, 0.3, 0.4	-
Subsample	0.5, 0.6, 0.7, 0.8 , 0.9	0.75, 0.80 , 0.85
Column Sample by Tree	0.1, 0.2, 0.3 , 0.4, 0.5 0.6, 0.7, 0.8, 0.9	0.25, 0.30 , 0.35
Regularisation Alpha	1E-5, 0.01 , 0.1, 1.0, 100	-

Hyperparameter Tuning (Cont')

	<code>trained_model = XGBClassifier(</code>	
	<code>learning_rate=0.1,</code>	he
Max. Depth	<code>n_estimators=1000,</code>	9, 10, 11
Min. Child Weight	<code>max_depth=9,</code>	2
Gamma	<code>min_child_weight=1,</code>	
Subsample	<code>gamma=0.0,</code>	
Column Sample by Tree	<code>colsample_bytree=0.30,</code>	75, 0.80, 0.85
Regularisation Alpha	<code>reg_alpha=0.01,</code>	25, 0.30, 0.35
	<code>objective='binary:logistic',</code>	
	<code>nthread=4,</code>	
	<code>scale_pos_weight=1,</code>	
	<code>seed=27</code>	
	<code>)</code>	

Model Evaluation

Fig. 7.1: Recall and Precision VS Ensemble Threshold



Ensemble Threshold refers to the number of weak classifiers agreeing to a positive before the model outputs positive




Optimal Recall: 92.6%

From our model, we obtain a **high recall** of 92.6% and a precision of 41.9%. In addition, the stability of recall as ensemble threshold increases proves the **robustness** of our model. This also shows the effectiveness of our data preprocessing (EasyEnsemble + SMOTE) in the training of our weak classifiers, allowing them to effectively identify fraud cases.

* While the results show an **optimal ensemble threshold** of 5, it is likely because of the small number of fraud cases in the test set which creates a considerable drop in recall/precision with one less correct prediction. For larger datasets, we believe recall to slide slowly as ensemble threshold is increased. Therefore, we will recommend the use of Ensemble Threshold 0 if the FN Cost to FP cost ratio is greater than the Recall to Precision ratio.

Optimal Precision: 41.9%

Revisit Our Approach - Growing Cost of FP

	Penny Frauds (≤ \$5)		Non-Penny Frauds	
	FP	FN	FP	FN
Avg Cost Incurred	\$0.03	\$1.09	\$0.56	\$222.13
E-commerce False Positives			Cardholder Behavior & Habit	
	Millennials - The Future Revenue Source			

Previously, we established the skewed cost ratio of FP to FN as shown in the table. However, upon further research this gap might be closing and fast.

Today, genuine transactions are declined at an alarming rate. These genuine users might then refrain from using that credit card altogether, thus incurring higher than expected FP cost for card companies (Massachusetts Institute of Technology, 2018; Marchini & Pascual, 2018).

Vesta Corporation (2016) even went on to suggest that losing a customer and transaction through false positive decline is perhaps worse than suffering outright fraud.

Due to greater internal cost visibility and the 3 key drivers on the left, we believe that this gap between FN cost and FP cost will further close in the future and hence more attention must be given to precision score when constructing our ML model.

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

1st Driver: Cardholder Behavior & Habit

- Friction in checkout experience teaches cardholders to avoid potential embarrassment or inconvenience from a repeated decline
- By habitually switching to a secondary card i.e. a competitor's card
- Poor user experience could spread through word-of-mouth
- The existing credit card company could lose its spot as the 1st choice, go-to card for consumers
- Hence, FP might in fact be more costly than estimated in the long run

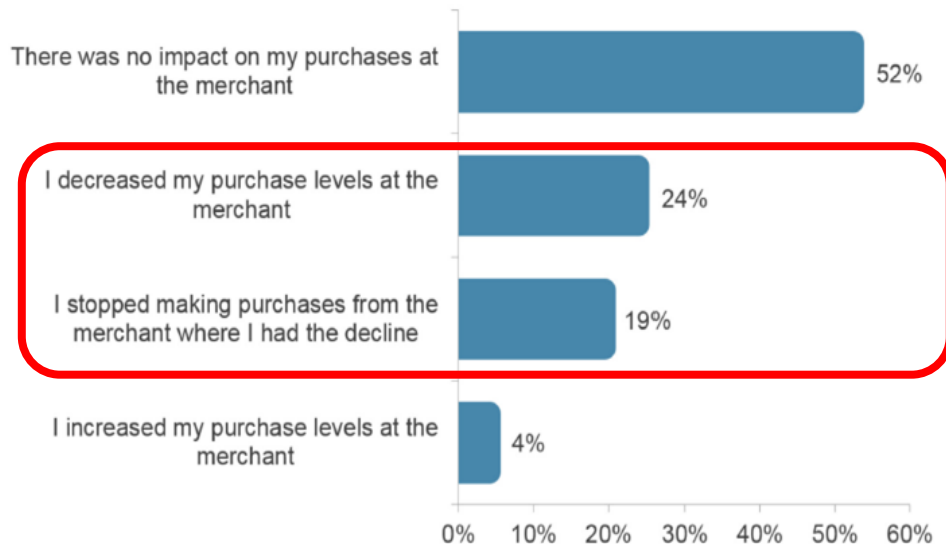


Fig. 8: How Most Recent False Positive Decline Was Resolved

Source: Javelin Strategy & Research, 2018

2nd Driver: Millennials - The Future Revenue Source

- Younger cardholders more prone to declined transactions due to
 - More erratic/ less predictable spending habits;
 - Their liking for products associated with high fraud risk i.e. high resale value products such as e-gift cards/tickets
- They react most adversely to declined transactions:
 - Higher proportion who reduce or stop their spending all together at the merchants
 - Based on our FP formula:
$$FP = \frac{\text{Intervention Cost} - \text{Txn Amt} \times \text{Merchant Fee} + \text{Customer Frustration}}{\text{Txn Amt} \times \text{Merchant Fee}}$$
Once the customer ceases the transaction, (Txn Amt x Merchant Fee) = 0, FP will definitely increase.
- Credit card companies' future source of revenue is threatened, signifying a more expensive FP cost

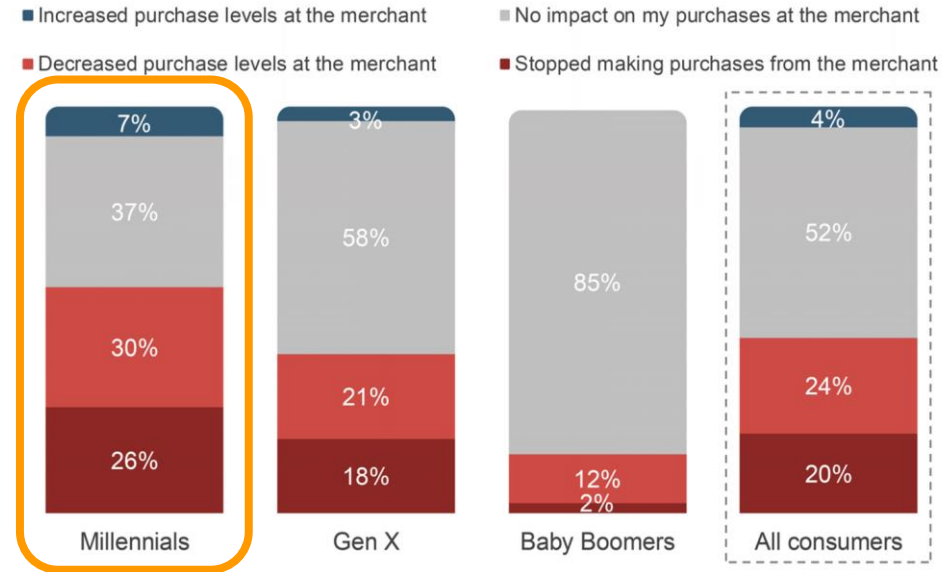


Fig. 9: Impact of False Positive Declines on Merchant Patronage, by Generation

Source: Javelin Strategy & Research, 2018

3rd Driver: E-commerce False Positives

- Against the backdrop of growing volumes of online/mobile transactions i.e. card-not-present (CNP) transactions
- E-commerce FPs have a greater impact on the affluent customer segment
- Resort to a secondary card, dampens their transaction experience with the existing credit card company
- As a result, cost of FP is very likely to escalate in the future

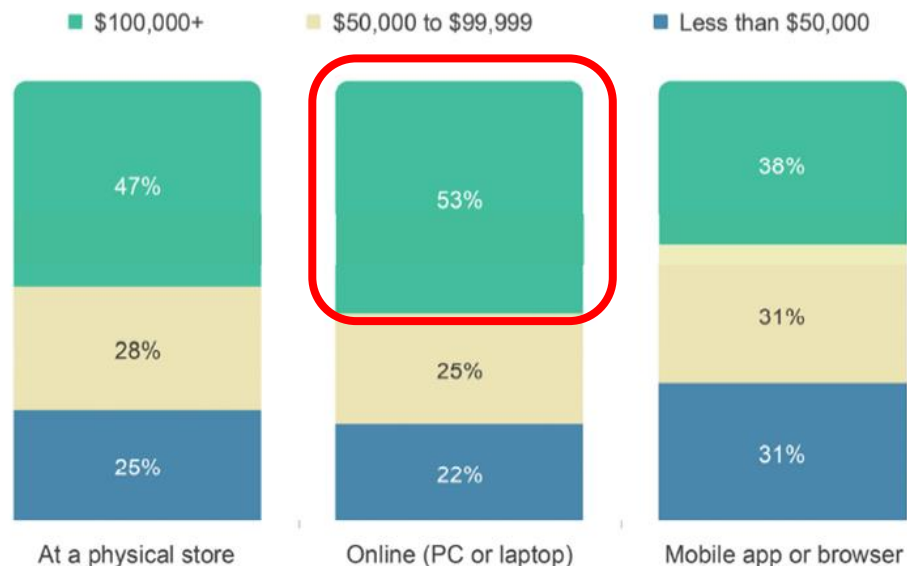


Fig. 10: Household Income of Declined Cardholders, by Channel Where Most Recent Decline Occurred

Source: Javelin Strategy & Research, 2018

New Costing

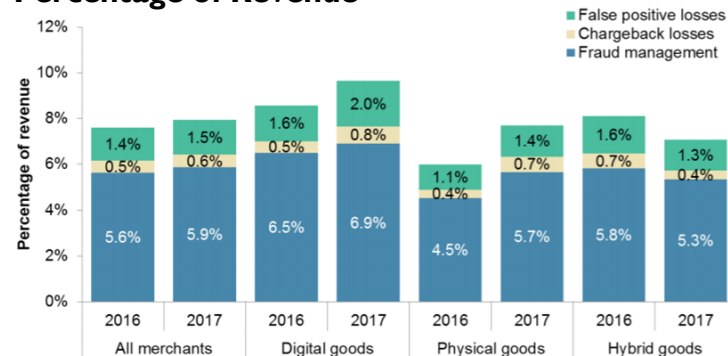
According to a 2017 report from Javelin Strategy & Research, on average FP cost takes up 1.5% of the credit card companies' revenue. We are able to derive the total FP cost for all the non-penny frauds.

Subsequently, according to (Bizarro, 2015), FDS in the market generally prioritise minimising FN and have a recall score of about 95%. If we were to deploy a similar FDS, we will be able to detect 223 TP cases out of the 235 non-penny fraud cases in this dataset.

According to another report from Javelin Strategy & Research in 2015, it is estimated that only 1 in 5 fraud predictions is correct. This suggests a precision of 20%. Using this figure, we can estimate to have 503 FP cases from 223 TP non-penny cases.

Hence, we will be able to compute an average cost of \$12.34 incurred for the credit card company for every FP non-penny case.

Fig. 11: Breakdown of Fraud-Related Costs as Percentage of Revenue



Source: Javelin Strategy & Research, 2017

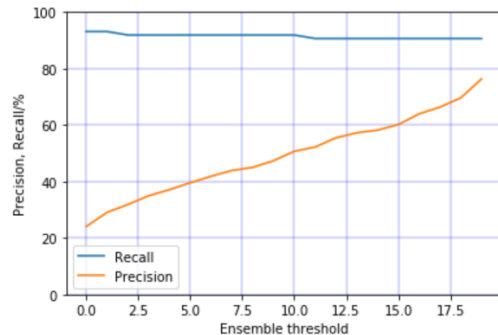
Total FP Costs (Non-Penny Frauds)	\$6,208.14
Estimated no. of TP cases (Non-Penny Frauds)	223
Estimated no. of FP cases (Non-Penny Fraud)	503
Avg Cost of FP (Non-Penny Fraud)	\$12.34

New Costing (Cont')

	Penny Frauds (≤ \$5)		Non-Penny Frauds	
	FP	FN	FP	FN
Transaction Amount	-	\$1.09	-	\$222.13
Intervention Cost	\$0.05	-	-	-
Merchant Fee	(\$0.02)	-	-	-
Customer Frustration Cost	\$0	-	-	-
Potential Reputational Damage	-	\$0	-	\$0
Avg Cost Incurred	\$0.03	\$1.09	\$12.34	\$222.13
Ratio of FP cost: FN cost	1 : 39		1 : 18	

Model Evaluation - Negligible FP cost

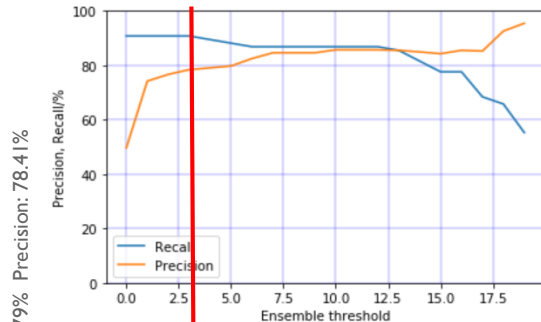
SMTEnsemble + XGBoost



Optimal: 1

Minimal Cost: \$702.64

SMOTE + Ensemble XGBoost



Optimal: 3

Minimal Cost: \$913.06

SMOTE + XGBoost

Recall: 88.61%

Precision: 87.50%

Minimal Cost: \$915.27

34

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

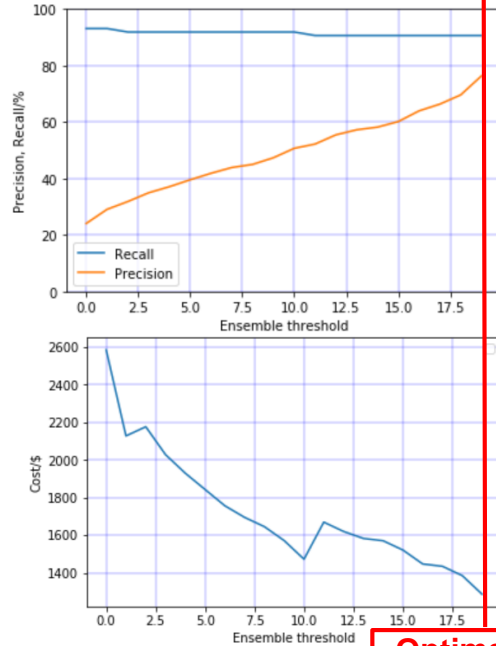
Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

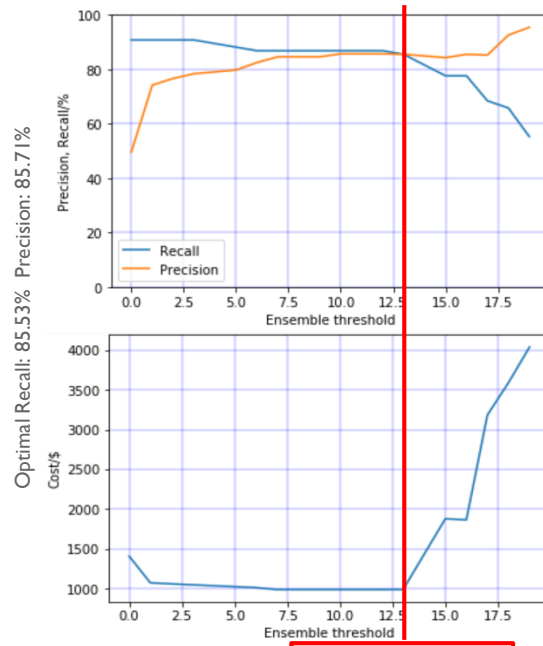
Model Evaluation - 1:18 FP:FN Ratio

SMTEnsemble + XGBoost



Minimal Cost: \$1285.78

SMOTE + Ensemble XGBoost



Minimal Cost: \$989.05

SMOTE + XGBoost

Recall: 88.61%

Precision: 87.50%

Minimal Cost: \$964.51

35

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

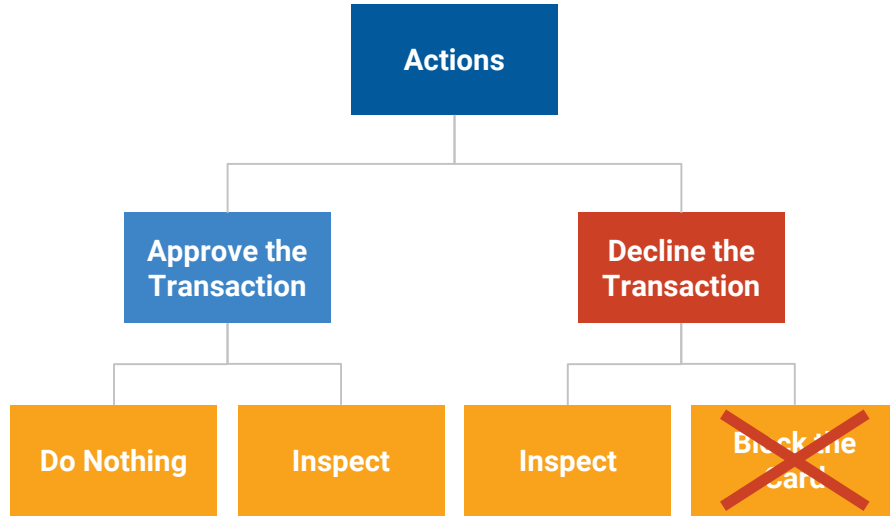
Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

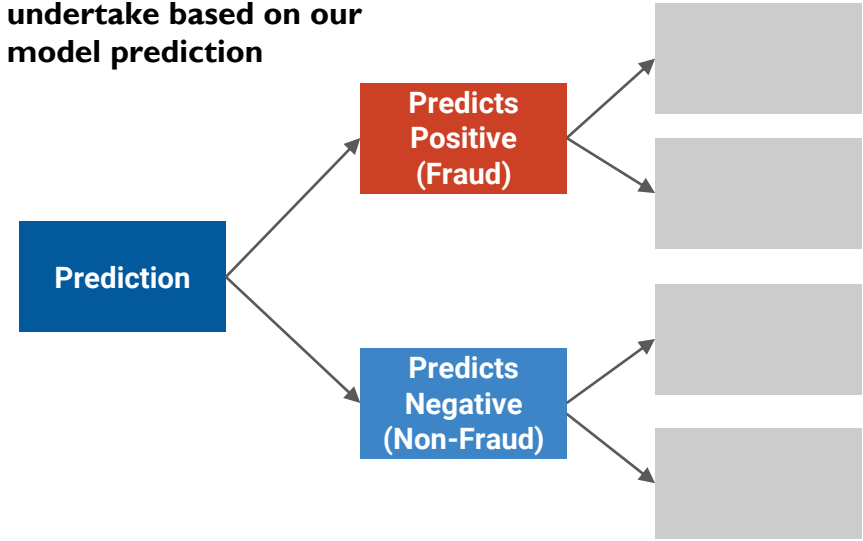
Business Implications & Recommendations

4 Courses of Action



There are 4 courses of actions that can be taken based on the result of the prediction model. However, due to predictably massive cost of customer frustration and risk of losing customer from blocking the card, we decided against this action.

Course of action to undertake based on our model prediction



With the model's prediction and its confidence, we would then incorporate another factor - dollar value of the transaction - to prescribe a cost-effective course of action to undertake by the credit card company.

36

Business Problem

Data Exploration

Costing Analysis

Performance Metric

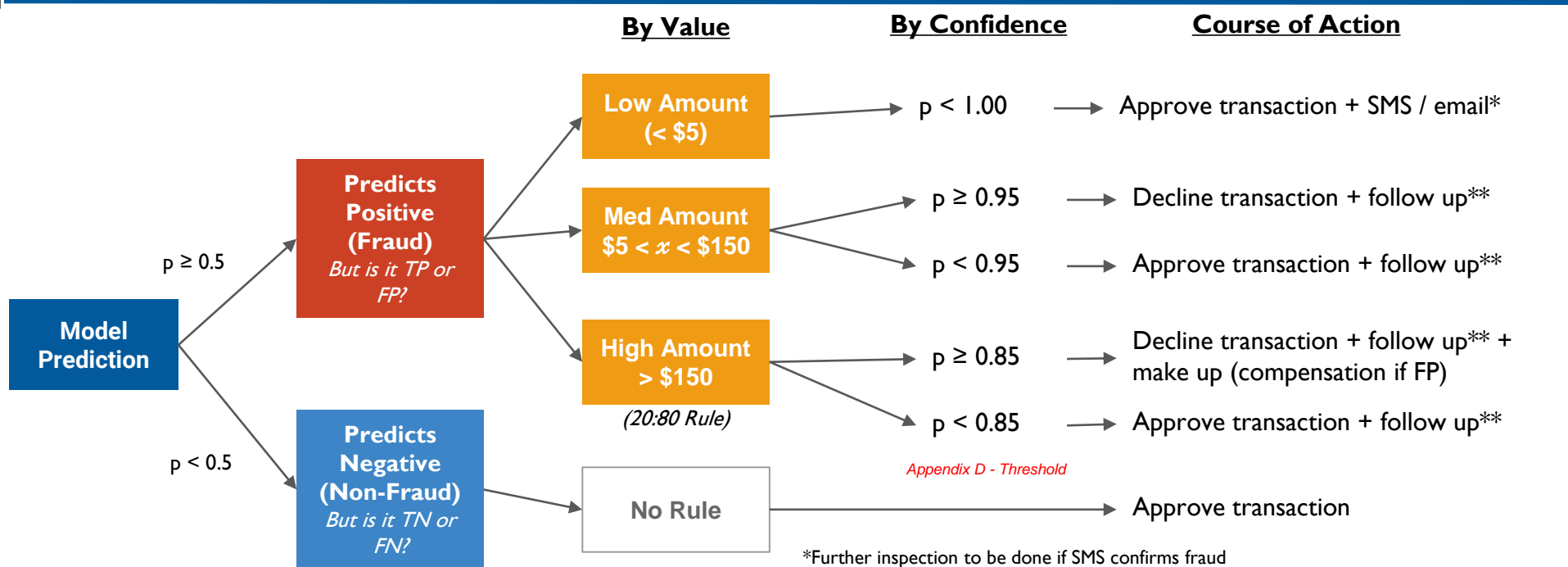
Model Building

Revisit Our Approach

Business Recommendation & Deployment

Limitation & Further research

Business Implications & Recommendations



Costs

Cost of FP: \$intervention + \$frustration - revenue

Cost of TP: \$intervention

*Further inspection to be done if SMS confirms fraud

** Investigation officer calls the card holder to verify

Note that confidence threshold varies from company to company and may be revised periodically.

Business Problem	Data Exploration	Costing Analysis	Performance Metric	Model Building	Revisit Our Approach	Business Recommendation & Deployment	Limitation & Further research
------------------	------------------	------------------	--------------------	----------------	----------------------	--------------------------------------	-------------------------------

Business Implications & Recommendations

By Value

By Confidence

Course of Action

Low Amount

$p < 1.00$

Approve transaction + SMS / email*

```
if amount < 5:
    print('Transaction Success. Send SMS')
elif amount[i] < 150:
    if confidence > 0.95:
        print('Decline Transaction. Follow up required.')
    else:
        print('Approve Transaction. Follow up required.')
else:
    if confidence > 0.85:
        print('Decline Transaction. Follow up required. Compensation if not fraud.')
    else:
        print('Approve Transaction. Follow up required.')
```

Model
Prediction

Costs

Cost of FP: \$intervention + \$frustration - revenue

Cost of TP: \$intervention

*Further inspection to be done if SMS confirms fraud

** Investigation officer calls the card holder to verify

Note that confidence threshold varies from company to company and may be revised periodically.

38

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

Business Implications & Recommendations

Current Cost (\$'000)

	\$
Total Transaction	4,531,987
Revenue	90,640
Loss from Fraud	(11,475)
Total Profit	79,164

New Cost (\$'000)

	\$	\$
Total Transaction		4,531,987
Revenue		90,640
Cost of Fraud Handling		
- FN (Small)	(3.65)	
- FN (Medium)	(173.45)	
- FN (Large)	(1,088.98)	
- FP (Small)	(0.215)	
- FP (Medium)	(72.36)	
- FP (Large)	(41.75)	
- TP	(2.30)	(1,382.70)
Total Profit		89,257

The cost of fraud handling is expected to reduce by \$10,092,682 (▼ 87.9%)(#).

The profit is expected to rise by 12.7%.

#There is a limitation on estimation of cost, and there are other costs such as data scientists' salary and cost of maintaining the model
+ Limitation of sampling

Results Based on Test run on 20% of test data (n = 45,569, 77 frauds)

True Positive: 70 x 5
True Negative: 45477 x 5
False Positive: 15 x 5
False Negative: 7 x 5

Based on Average Cost

FN (S) : \$1.09 FP (S): \$0.03
FN (M): \$63.33 FP (M): \$12.34
FN (L): \$497.25 FP (L): \$12.34
TP: \$0.03

39

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

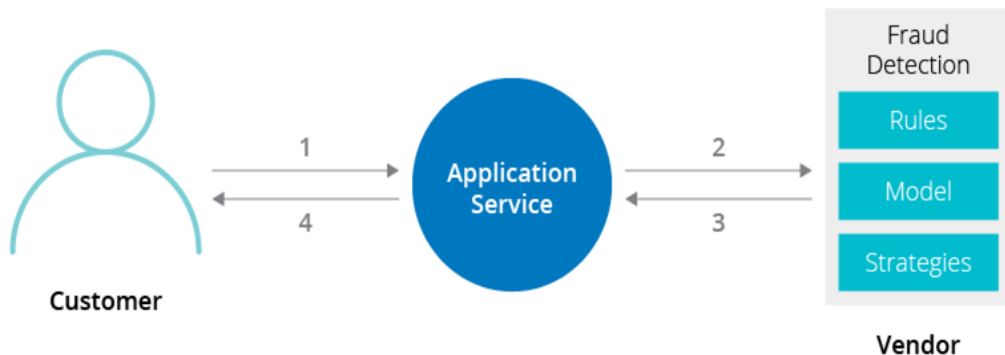
Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

Deploying Our FDS



Credits: <https://www.thoughtworks.com/insights/blog/using-cd-machine-learning-models-tackle-fraud>

Considerations

- Ever-changing fraud landscape as fraudsters continually try to outsmart the FDS
- Rules and models to be adjusted to stay ahead of the crooks
- For 3rd party vendor, change often requires a governance process to ensure that change is not biased/discriminative towards certain profile of cardholders (e.g. age, location, race)

How to overcome the lengthy governance process? Otherwise, time lags in FDS would be exploited by fraudster

40

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

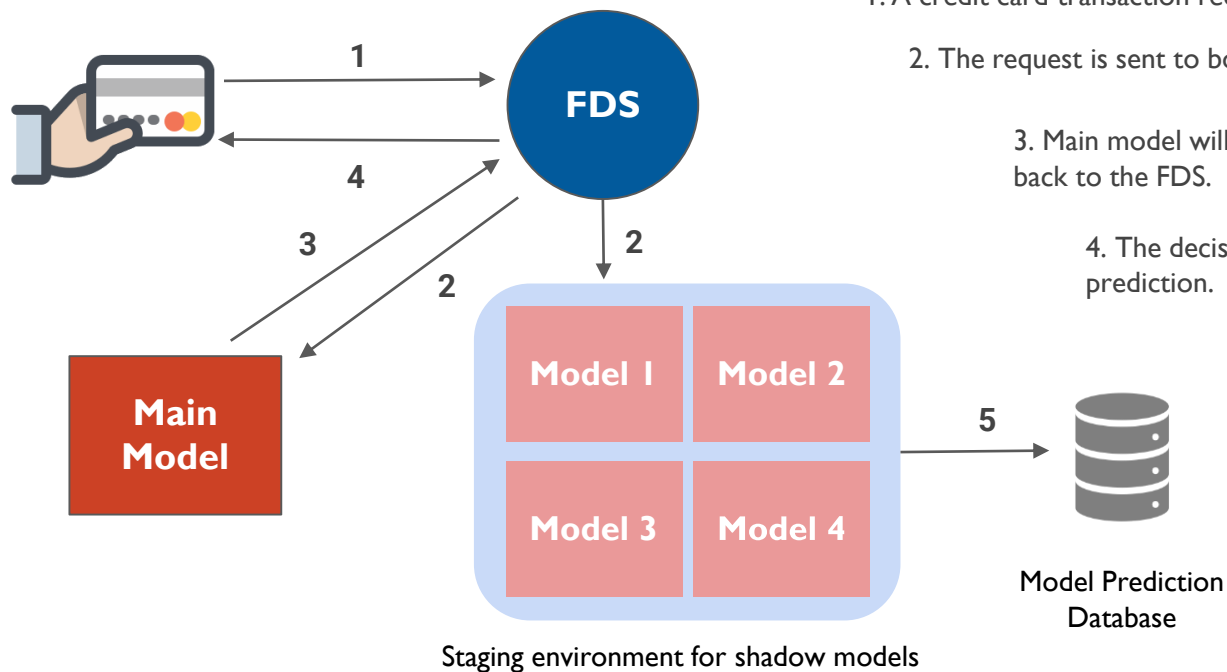
Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

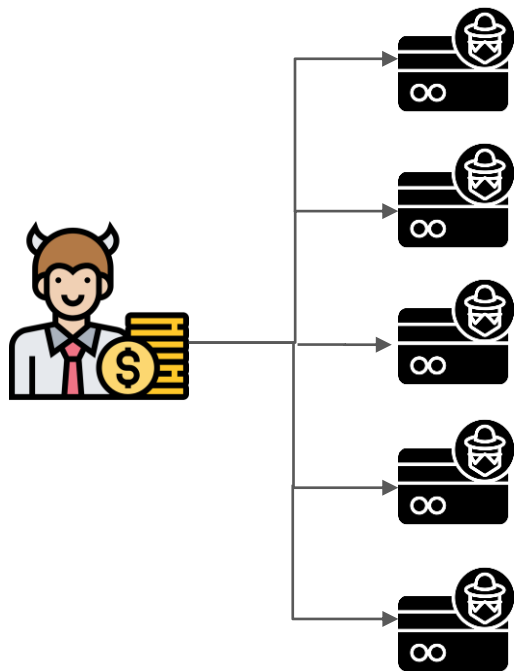
Limitation &
Further research

Deploying Our FDS (Cont')



** Shadow models that outperforms the main model consistently will be promoted and replace the main model*

Deploying Our FDS - Benefits



Adversarial attacks are when fraudsters intentionally probe the FDS to find out its loopholes or even cause it to malfunction altogether. In doing so, they know how to evade the system defense and generate fraudulent transactions that go undetected.

For example, a fraudster could purchase thousands of credit card numbers and social security numbers as a means of **testing and learning the current FDS defense**. Currently many FDS are well built to flag fraudulent transaction, but **few actively incorporate an adversary's potential strategies**

With cunning fraudsters and ever-evolving fraud fraud behavior, we need an **adaptive and pre-emptive detection system** that can match this rate of change.

In our deployment, we create a **staging environment** for the suite of other models where we observe how they perform but avoid the risk of them making the wrong decision. With the above deployment, we are enabling our system to **keep up with the fraudsters**.

Limitations & Further Research

Limitations

Further Research

Anonymisation of Features



01

Due to anonymised features, limited insights are extracted. More meaningful and actionable recommendations pertaining to the business context can be made with known inputs.

- Behavioural analytics can be used and model can be trained at individual account holder's level. The model can learn each account holder's behaviour and detect any inconsistencies in behaviour.
- For instance, behavioural biometrics can be used to assess the user's interactions with input devices (Javelin, 2018).

Lack of Cost Visibility



02

Due to lack of information and expertise in estimating costs such as frustration and intervention cost, cost ratio can be more intimate and applicable for the company.

- More accurate projection of cost can be made with better costing information and estimates.
- More cost efficient decision rules can be designed to minimise loss from fraud.

Non-representative data



03

The sample data of 40 hours may not be representative of the entire population. The characteristic of data may vary across different time of the year or in other years, especially when fraudsters adapt over time. Also, 417 fraud data points are not sufficient to build a robust model.

- With continued training of the model with more data, the model will become stronger with a larger minority class size.
- A larger dataset would also allow us to observe the tradeoffs and intersection of the recall and precision curves more closely.

43

Business
Problem

Data
Exploration

Costing
Analysis

Performance
Metric

Model
Building

Revisit Our
Approach

Business
Recommendation
& Deployment

Limitation &
Further research

A nighttime photograph of a city skyline, likely Singapore, with numerous skyscrapers illuminated by various lights. The lights reflect on the water in the foreground. The sky is dark blue with some clouds. The text "Thank You" is overlaid in the center in a white, italicized serif font.

Thank You

References

Brown, E. (2018). What is the Value of Machine Learning? The Case of Credit Card Fraud Detection - RCG Global Services. Retrieved from <https://rcgglobalservices.com/what-is-the-value-of-machine-learning-the-case-of-credit-card-fraud-detection/>

Bizarro, D. (2015). In Fraud Detection, Size Matters. Retrieved from <https://www.rtinsights.com/in-fraud-detection-size-matters/>

Massachusetts Institute of Technology. (2018, September 20). Reducing false positives in credit card fraud detection: Model extracts granular behavioral patterns from transaction data to more accurately flag suspicious activity. *ScienceDaily*. Retrieved March 16, 2019 from www.sciencedaily.com/releases/2018/09/180920131513.htm

Marchini, K., & Pascual, A. (2018). *Addressing the Threat of False Positive Declines*. Javelin Strategy. Retrieved from <https://www.javelinstrategy.com/coverage-area/addressing-threat-false-positive-declines>

Javelin Strategy & Research. (2016). *The Financial Impact of Fraud*. Vesta Corporation. Retrieved from http://info.trustvesta.com/research2016?utm_campaign=Javelin%20Research%20Power%20Play%2010/2016&utm_source=ebook&utm_medium=cta

Elmery, H., & LeBlanc, S. (2018). Using CD with machine learning models to tackle fraud. Retrieved from <https://www.thoughtworks.com/insights/blog/using-cd-machine-learning-models-tackle-fraud>

Appendix A: Misclassification Cost Breakdown

Case	Description	(Mis) classification Costs
True Negative (TN)	Actual genuine txn as predicted by model	= -1 x Txn Amt x Merchant Fee
False Positive (FP)	Model predicted fraud, but the transaction is actually genuine	= Intervention Cost - Txn Amt x Merchant Fee + Customer Frustration
False Negative (FN)	Model predicted no fraud, but the transaction is actually fraudulent	= Txn Amt + Potential Reputational Damage
True Positive (TP)	Actual fraud as predicted by model	= Intervention Cost
Column1	Count	Dollar Value (\$)
All txns	227844	\$20,694,004
Genuine txns	227427	\$20,641,605
Fraud txns (<i>actual frauds as confirmed by the cardholders</i>)	417	\$52,399

Appendix B: Alternative Costing Method for FP

Column1	Column2	Assumptions
Total FP Costs (Non-Penny Frauds)	\$6,208.14	Javelin Report - FP costs is 1.5% of total revenue
Estimated no. of TP cases	223	95% recall for DFS on the current market (Dr Pedro Bizarro)
Estimated no. of FP cases (Non-Penny Fraud)	503	Javelin Report - 20% precision
Avg Cost of FP (Non-Penny Fraud)	\$12.34	

Column1	Penny Frauds (defined as \$5 and below)	Non-Penny Frauds
Intervention Cost	0.05	-
Merchant Fee	2.00%	2.00%
Count	182	235
Total Dollar Value	\$198.20	\$52,199.89
Average Txn Amt	\$1.09	\$222.13
Customer Frustration Cost	0	-
Potential Reputational Damage	0	0
Avg Cost of FP	\$0.03	-
Avg Cost of FN	\$1.09	\$222.13
Ratio of FN cost vs FP cost	39	18

Appendix C: Other Performance Metric Considerations

$$\text{TPR (sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

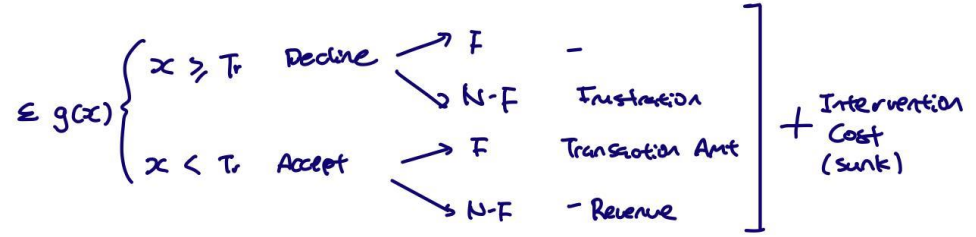
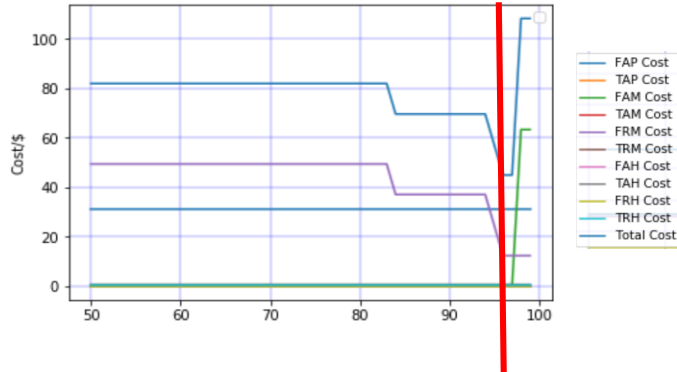
$$\text{FPR (1-specificity)} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Bizarro, 2015 suggests the use of both true positive rate (TPR) i.e. recall and false positive rate (FPR). In his study, the model achieved a 95% TPR and 0.6% FPR. However,

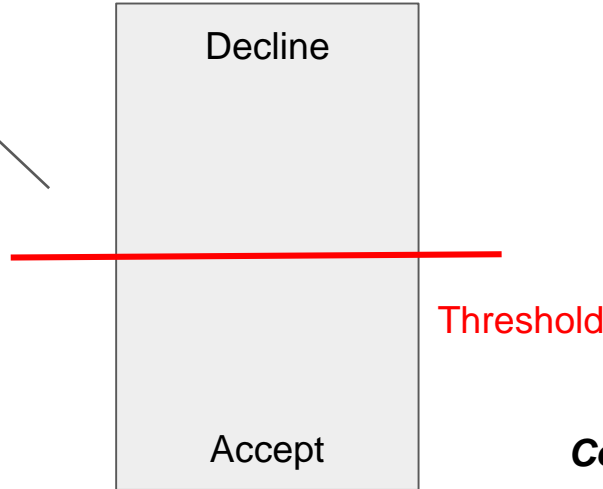
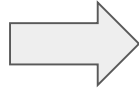
- applying the 0.6% FPR, out of the 227,427 genuine transactions, we will have 1,365 FP cases
- applying the 95% TPR, out of the 417 fraud transactions, we will have 396 TP cases

We would then compute the precision score of his model, which is $\text{TP}/(\text{TP} + \text{FP})$, giving us a mere 22.5% precision. Hence, we conclude that the use of FPR as a performance metric is not ideal for imbalanced dataset.

Appendix D: Threshold Optimisation



New Data



To find the best threshold that minimises the cost for each low, medium and high amount level, loop was run to generate the cost graph. For each level of threshold, total cost was computed based on the current data. However, with 40 hours of dataset, there is very limited number of datapoints to find the optimal point with the lowest cost.

Codes in the following slide

Appendix D: Threshold Optimisation (Codes)

```
1 def costing_v2(predictions, label, amount, m_thres=0.5, h_thres=0.5, intervention_cost=0.03,  
2   p_transaction_amt=1.09, m_transaction_amt=63.33, h_transaction_amt=497.25,  
3   frustraion_cost=11112.34):  
4     fa_p_cost = 0  
5     ta_p_cost = 0  
6     fa_m_cost = 0  
7     ta_m_cost = 0  
8     fr_m_cost = 0  
9     tr_m_cost = 0  
10    fa_h_cost = 0  
11    ta_h_cost = 0  
12    fr_h_cost = 0  
13    tr_h_cost = 0  
14  
15    for i, pred in enumerate(predictions):  
16        if pred[1] > 0.5:  
17            if amount[i] < 5:  
18                if label[i] == 1:  
19                    fa_p_cost += p_transaction_amt + intervention_cost  
20                else:  
21                    ta_p_cost += intervention_cost  
22  
23            elif amount[i] < 150:  
24                if pred[1] < m_thres: # accept  
25                    if label[i] == 1: # false  
26                        fa_m_cost += m_transaction_amt + intervention_cost  
27                    else:  
28                        ta_m_cost += intervention_cost  
29                else: #decline  
30                    if label[i] == 0: # frustration  
31                        fr_m_cost += frustraion_cost + intervention_cost  
32                    else:  
33                        tr_m_cost += intervention_cost  
34
```

```
5     else: # high value  
6         if pred[1] < h_thres:  
7             if label[i] == 1:  
8                 fa_h_cost += h_transaction_amt + intervention_cost  
9             else:  
10                ta_h_cost += intervention_cost  
11        else:  
12            if label[i] == 0:  
13                fr_h_cost += frustraion_cost + intervention_cost  
14            else:  
15                tr_h_cost += intervention_cost  
16  
17    return [fa_p_cost,  
18            ta_p_cost,  
19            fa_m_cost,  
20            ta_m_cost,  
21            fr_m_cost,  
22            tr_m_cost,  
23            fa_h_cost,  
24            ta_h_cost,  
25            fr_h_cost,  
26            tr_h_cost, sum([fa_p_cost,  
27                            ta_p_cost,  
28                            fa_m_cost,  
29                            ta_m_cost,  
30                            fr_m_cost,  
31                            tr_m_cost,  
32                            fa_h_cost,  
33                            ta_h_cost,  
34                            fr_h_cost,  
35                            tr_h_cost])]
```

```
1 print(predictions[:5])
```

```
1 results_list = []  
2 for threshold in range(99980, 100000):  
3     results_list.append(costing_v2(predictions, np.array(test_data_target), test_data_amount, threshold/100000))  
4 print(results_list[len(results_list)-5:])
```

```
1 plt.plot(results_list)  
2 plt.ylabel('Cost/$')  
3 plt.xlabel('Confidence threshold')  
4 plt.grid(which='both', color='b', linestyle='--', linewidth=0.3)  
5 plt.legend()
```