Fraud Detection

Estimating ROI with Machine Learning

The context

Together with the increasing policy acquisition, company received an increase in the number of claims.

Using the current business rules and with resource constraints, the claim function only investigated 10% of claims and found 5% fraud cases among the total claim cases.

The management questions whether the possibility to improve for fraud detection or increase number of claims to be investigated.

Given that:

- Number of claims per month: 200 cases,
- Cost for each investigation is $500 / case,
- Litigation expense is $7,500 / case.

Identify the approach

- With the current business rules, claim function only investigates 10% of 200 claim cases or 20 cases / per month; but 5% (~ 10 cases) are found as fraud. As the quantity of incorrectly investigated cases are still relatively high, we need to find the approach to reduce those cases.
- While we still cannot fully remove current business rules, we can analyze historical data to find additional rules to classify Fraud or Non-Fraud cases.
- A suggested approach is using Machine Learning to classify Fraud / Non-fraud claim in order to categorize which cases should be investigated or not. The following slides aim to estimate the benefits when applying this approach.

Analysing the historical data

- By analysing the historical data, the project team can select the Machine Learning (ML) model to classify Fraud or Non-Fraud case from total claims submission.
- The **confusion matrix** from selected ML model is used to scale by 200, the monthly claims, in order to have the expected confusion matrix for future claim data.

Confusion matrix in our sample

- Confusion matrix is a table used to measure the performance of a classification model.
- Four possible outcomes in our sample
  - True Positive: fraud has occurred, the model correctly predicts frauds – the claim should be investigated
  - True Negative: model corectly predicts non-fraud, the claim should be paid.
  - False Positive: model falsely predicts fraud, the claim is investigated but found unnecessary.
  - Fales Negative: model falsely predicts non-fraud, the claim might eventually end up in litigation.

		Predicted	
		Negative	Positive
Actual	Negative	**TN**	**FP**
	Positive	**FN**	**TP**

Expected Confusion matrix from 200 claim cases

		Predicted		
		Non-Fraud	Fraud	Percent Correct
Actual	Non-Fraud	150	12	0.93
	Fraud	10	28	0.74
			Accuracy	0.89

- By applying the model, we expect the model to accurately predict 89% of fraud and non-fraud case:
  - True Positive Rate: 74% of correctly predicted fraud cases.
  - True Negative Rate: 93% of correctly predicted non-fraud cases.

Estimating cost and benefit

We assign value to the four possible outcomes to estimate ***the Correct Predictions***

- **True Negative:** If the model correctly predicts **non fraud**, the claim should be paid without investigation. The True Negative Rate in our model is quite high, we could concurrent reduce the number of unnecessary investigation. Based on the current business rule, we can filter the overlapped cases which are classified by ML model. Let's say we can **reduce 10** investigation with saving of **$500** each.

save 10 x $500	**FP**
FN	**save 21 x $7,000**

- **True Possitive:** If the model correctly predicts **fraud**, the claim should be investigated: we're going to save $7,500 litigation fee but have to spend $500 for investigation, then we net about **$7,000.** And based on number of investigation we can do a month and the accuracy in our model, we expect to do **additional 21** investigation.

Estimating cost and benefit

And ***the Incorrect Predictions***

- **False Posssitive:** the model falsely predicts **fraud**, the claim is investigated but found unncessary. It means the money for investigation unncessary. Based on the prior investigation rate and the ML mode (between TP/FP), we guess investigating **9** of these which costly **$500** each.

save 10 x $500	**cost 9 x $500**
should be reduced	**save 21 x $7,000**

- **False Negative:** the model falsely predicts **non-fraud**, the claim end up in litigation. We expect these should be reduced as they should be moved from this category into True Possitive category.

Total benefit = $147,000 - $4,500 + $5,000 = $147,500

Conclusions

By applying the Machine Learning to classify *fraud / non-fraud*, we expect:

- Correctly predict 74% of fraud among 19-20% fraud claim submission.
- Correctly predict 93% of non-fraud amond 80-81% non-fraud claim submission.
- Investigate additional 30 cases predicting as *fraud* but reduce 10 cases predicting *non-fraud;* it expects to benefit as **$147,000** + **$5,000 – $4,500 =** **$147,500**

	AS IS	TO BE
Total case to be investigated	20	30
a. Saving litigation when findding of fraud cases	10 * $7,500 = **75,000**	21 * $7,500 = **157,500**
b. Cost of investigation	20 * $500 = **10,000**	30 * $500 = **15,000**
c. Reduce cost by not investigating non-fraud cases	un-known	10 * $500 = **5,000**
Summary = (a+c-b)=	**$65,000**	**$147,500**

Futher actions and improvements

- By increasing the number of investigation cases to 40 which equal to number of cases classifying as fraud, we expect to increase the benefit of **$195,000 (= $196,000** + **$5,000 – $6,000)**
- By optimizing the ML Model, we expect to reduce the False Negative, which reduces the litigation cost paid for falsely predicting non-fraud claims.

save 10 x $500	**cost 12 x $500**
should be reduced	**save 28 x $7,000**

Reference

Inspire from learning course:

- ***Predictive Analytics Essential Traing: Estimating and Ensuring ROI*** by Keith McCormick