<u>Use case: Fraud Detection</u>

ML Use case design

Problem Framing

<table>
<tr><th></th><th>Qualitative</th><th>Quantitative</th><th>Question</th></tr>
<tr><td>Current State</td><td>Higher rates of frauds => negative user experience => loss of trust and reputation => reduction in engagement => less revenue</td><td>10% fraud =>10% less engagement => 10% loss in revenue</td><td>what is the current situation (pains/desires) that we want to address and why?</td></tr>
<tr><td>Objectives</td><td>build a model that can detect fraudulent activity as soon as they take place
decrease fraud => improve user experience => gain user trust => more engagement => more revenue => improve topline</td><td>reduce at least 50% of the fraud(from 10% to 5%) => 5% more engagement => 5% more revenue</td><td>what is that we want to do and why? (to improve the topline/bottom line?)</td></tr>
<tr><td>Benefit/ Cost tradeoff And prioritization</td><td>cost of errors:
FN => fraud not detected =>very bad user experience => more complaints => more churn => big loss of revenue
FP => genuine transaction marked as fraud => bad user experience => low engagement => less revenue
benefits of correct</td><td>cost-benefit matrix
<table><tr><td>c(TP)</td><td>c(FP)</td></tr><tr><td>c(FN)</td><td>c(TN)</td></tr></table>
1% TP => 1% engagement => + 1% revenue

1% FP => 0.5% less engagement => 0.5% less revenue

1% FN => 1% very bad</td><td>what is the cost of errors/benefits of correct predictions and why?</td></tr>
</table>

	predictions: TP => correctly detected fraud => better user experience => better engagement => more revenue TN => correctly kept non-fraud => maintained user experience as expected => no significant impact on revenue	experiences => 0.1% risk of churn => 10% less engagement over customer lifetime => 10% less revenue 1% TN => no significant impact on revenue	
Constraints	can only afford a small FN percent => very small percent of very bad user experience => limited risk of churn => limited loss of revenue	at most 1% FN => 1% very bad experiences => 1% churn => 1% risk of revenue loss => acceptable risk for 5% potential upside in revenue	what are the acceptable risks/budgets and why?
Desired state	• benefit: significantly fewer frauds => significantly better user experience => significantly better engagement => significantly better revenue • cost: very few false negatives => limited risk of very bad user experience => limited risk of churn => limited risk to revenue	• at least 50% decrease in fraud (from 10% to 5%) => 2.5% better engagement => 2.5% more revenue • at most 1% false negatives => 1% very bad experiences => 1% churns => 1% risk to revenue	what is the desired outcome (benefits/costs) that we want to see and why?

Why ML

	Qualitative	*Quantitative*	*Question*
Best non-ML alternative hypothesis	classify based on a curated list of keywords => too many FP and FN => very bad user experience => lesser engagement => loss of revenue	30% FP 10% FN => not cleaning enough frauds and causing more complaints for misplacing fraudulent transactions as genuine => 10% revenue loss risk	what are the non-ML alternatives and why are they problematic? (pains/missed gains)?
ML value proposition hypothesis	much fewer FP and FN => much better user experience => better reputation => much better revenue	15% FP 5% FN => 50% decrease in spam (from 10% to 5%) at the expense of 1% bad engagements => 5% increase in revenue at the expense of 0.1% risk	what are the advantages (pain relievers/gain creators) of ML solution and why?
ML feasibility hypothesis	• data: labeled samples of historic bank transactions data are available • model: state-of-the-art review suggests promising candidates are available	• data: around five thousand samples • model: state-of-the-art claim solutions with 10% FP 5% FN	what data and model are good candidates and why?

ML Solution Design

	choices	*metrics*	*experiment*
data	(labeled) bank transaction data	• label imbalance	• randomized 70/15/15 train/validation/test split
model	P(fraud)[Probability of fraud]	• AUCPR (precision-recall curve)	• rule-based heuristic • tf-idf + logistic regression • tf-idf + random forest • Neural Networks train these benchmark models (from simpler to more complex) using train data. validate and tune using validation data. select the model with the best AUCPR on test data
actions	if pr(spam) > threshold: Depending on the probability either a direct takedown or put aside for manual check	• precision • recall • confusion matrix	• choose a threshold to maximize the recall (estimated reward) subject to recall > 95%
reward	• decrease in fraud • cost of misclassification	• % decrease in fraud • % increase in satisfied customers	• shadow test • A/B test