**Fraud Detection**

**Problem Framing:**

|  | Qualitative | Quantitative | Question |
| --- | --- | --- | --- |
| Current State | Too many fraudulent transactions => bad user experience=> less customers=> less revenue | 10% fraudulent transactions=>5% less customers=>5% less revenue | What is the average number of fraudulent transactions at the present time and what can we do about it? |
| Objectives | * Build a model that can flag a transaction as fraudulent before completion * Decrease fraudulent transactions=>improve customer experience=> increase revenue | Flag and reduce bad transactions by at least 20% | How do we flag these transactions? |
| Benefit/Cost Tradeoff and Prioritization | * Errors -   TP - Fraudulent transaction flagged =>protect customers=>more revenue  TN- Fraudulent transaction marked valid=>very bad user experience=> less revenue  FP-valid transaction flagged=>bad user experience=>less revenue  FN-Valid transaction marked valid=>no significant impact on revenue | cost-benefit matrix   | c(TP) | c(FP) | | --- | --- | | c(FN) | c(TN) | | What are the costs of errors/benefits of correct predictions and why? |
| Constraints | Can only afford very little TN rate | At most 10% TN=> very bad | What are the acceptable risks/budgets and why? |
| Desired state | * benefit: significantly lesser fraud transactions=> significantly better user experience => significantly better engagement => significantly better revenue * cost: very few true negatives => limited risk of very bad user experience => limited risk of churn => limited risk to revenue | * at least 50% decrease in fraud (from 20% to 10%) => 5% better engagement => 5% more 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 transaction amount or location of transaction => too many FP and TN => very bad user experience => lesser engagement => loss of revenue | 50% FP 70% TN => not cleaning enough fraud transactions and causing more complaints for misplacing genuine transactions as fraud => 5% 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 TN => much better user experience => much better revenue | 10% FP 50% TN => 50% decrease in fraudulent transcations(from 20% to 10%) 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 solutions and why? |
| ML feasibility  hypothesis | * data: labeled samples of historic sms text data is 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 20% TN | What data and models are good candidates and why? |

**ML Solution Design**

|  | choices | metrics | experiment |
| --- | --- | --- | --- |
| data | (labeled) transaction data | * label imbalance | * randomized 70/15/15 train/validation/test split |
| model | pr(fraud) | * AUCPR   (precision recall curve) | * rule based heuristic * tf-idf + logistic regression * tf-idf + random forest * BERT + logistic regression   train these benchmark models using train data. validate and tune using validation data. select the model with best AUCPR on test data |
| action | if pr(fraud) > threshold: auto take down | * precision * recall * confusion matrix | * choose a threshold to maximize the recall (estimated reward) subject to precision > 90% |
| reward | * decrease in fraud * cost of misclassification | * % decrease in fraud * % increase in daily active users | * A/B test |