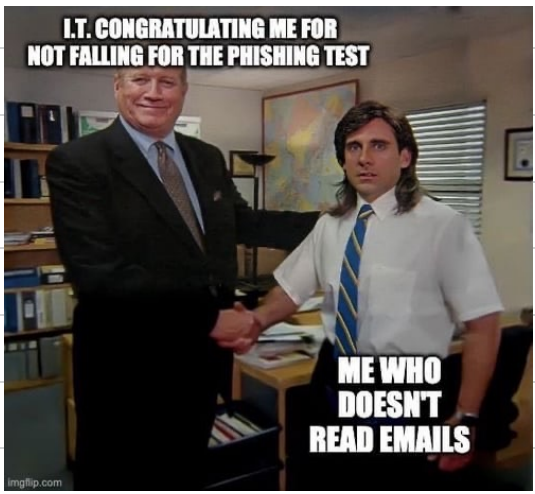
ML-System Design-2
Fraud Detection

Session-8

Jan 22, 2024



Agenda

* Design an ML system that can analyze all txns & classify Fraud / Not Fraud

Where do we start??

Q: Manager says

" Traffic is down by 20% on CRED"
why?!

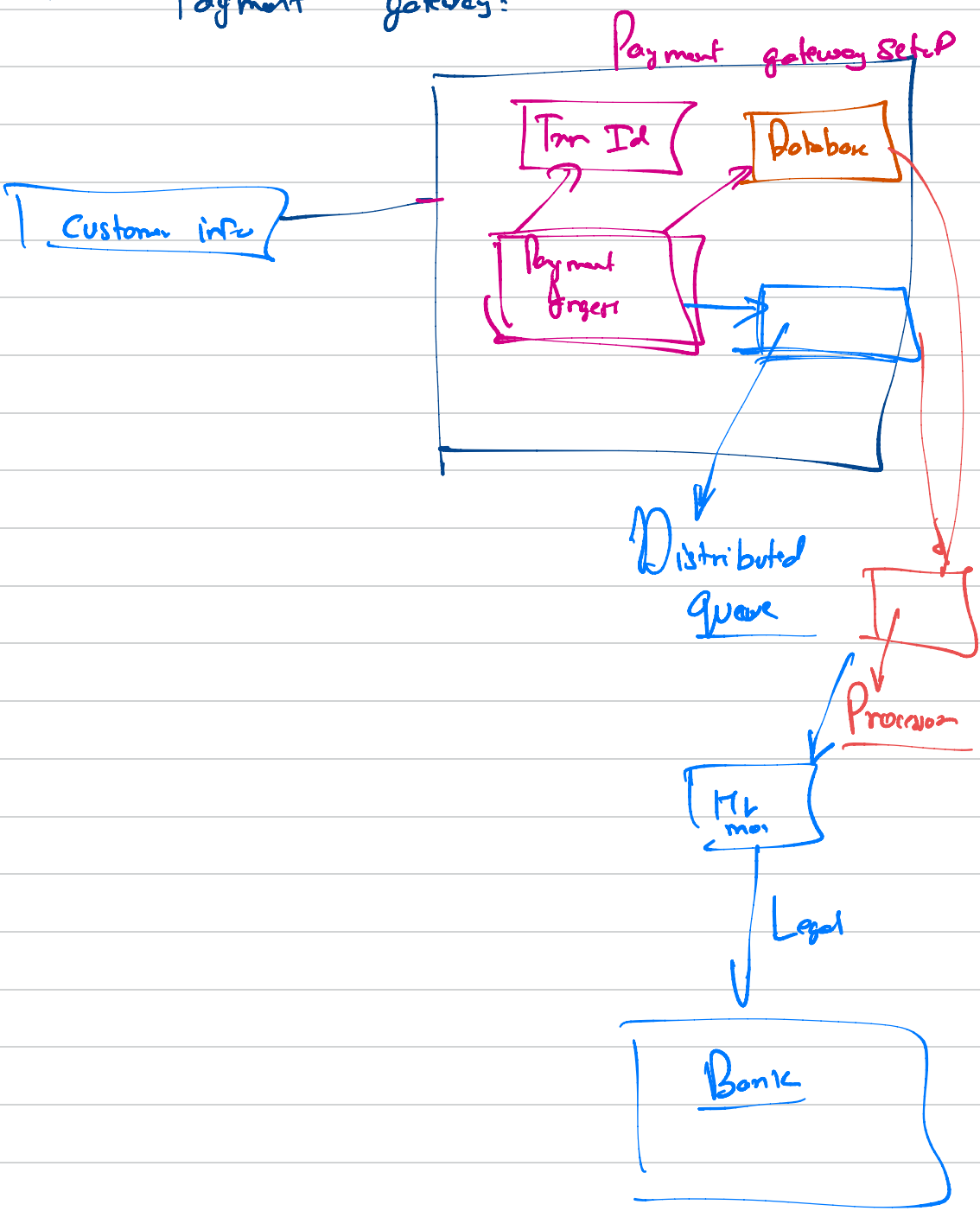
① Clarify → What metric?!
→ On what interval?!
→ Down by 20% w.r.t. what??

Current work Flow of Razorpay

1. Checkout Initialization.
2. Payment Information entry
3. Payment gateway:

4. Authorization Req. <- OTP
5. Confirmation to Custom.
6. Settlement / Fund transfer

→ Payment gateway:



Supervised → Data

1. Account related features
   - → high balance
   - → Type of account
   - → ~~Account no.~~
   - → " status
   - Last used date
   - City
   - Internet banking
2. Txn Related (Fraud)
   - → txn no
   - → Amount
   - → location
   - → time of the day
3. User related
   - → CIBIL Score
   - → location
   - → low-profile / high-profile
   - → Past victim
   - → Education

→ Data → extremely imbalanced

1. Re-sampling / Under sampling
   → SMOTE

2. Ensemble methods → RF

3. Class weights → {'0': 0.95, '1': 0.1}

4. Evaluation metric → ROC-AUC

5. Anomaly detection

   → train auto encoder → compare flaws with test of illegal one → compare & classify as anomalous

   train with legal data

6 Cost sensitive learning

→ Penalize more (For incorrect classification of smaller class)

7 Custom loss functions → more on minority class

8 Domain Knowledge

↳ Create new feature ↓ Focus on minority class

→ Models we can use ??

1. RF ✓✓
2. Logistic Reg.
3. GBM

4. SUM

5. ANN

Please use exp. tracking ML Flow

→ Some nice metrics

1. Confusion matrix
2. Precision
3. Recall → Cost of missing fraud is v-high
4. ROC-AUC
5. AP - Average Precision.
   → Weighted means of precision

6. Matthews Correlation Coeff
7. Cohen's Kappa

} H.W

Scoping

1. Data Engineer: → Pipelin
2. ML Engineer: → Modelling + DE / 10
3. Devops → Deploy/ Scaling
4. Product team → QA / A + Sanity Check

→ drift ?? Consistently train on new data.

Some Custom logic ??
+
Manual intervention

1000 False positive → 1/2 ↓ actual Perf

→ How to reduce latency:

Quantization.

① Pre - Quant

② Post - Quant

GPT - 2 6 Billion params

Post Quantization

32 bits → 8 bits

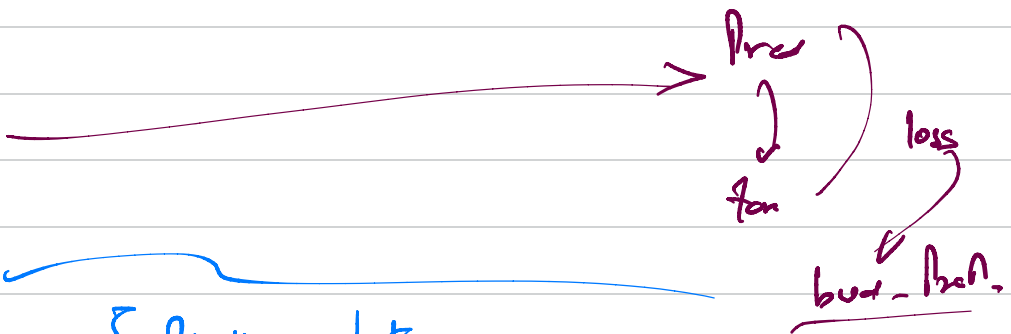
0.8123 45681 23

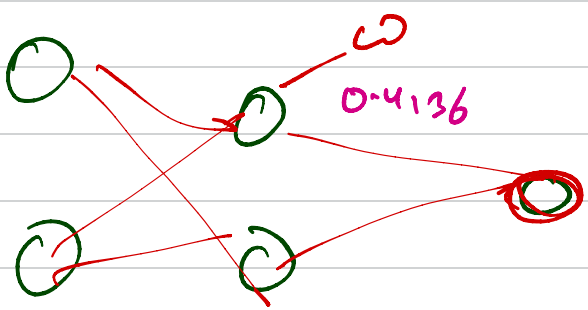
Reduces model size a lot

deploy on small devices

→ Pre Quantization

* Train model





→ Adversarial attack

* tweak my input data → more Regularized Robust model.

What we do, large no DP

FP

50000, Current, High, 9900000 → NF

10000, Current, low, 99999999 → F

9000, Current, low, 999 → F