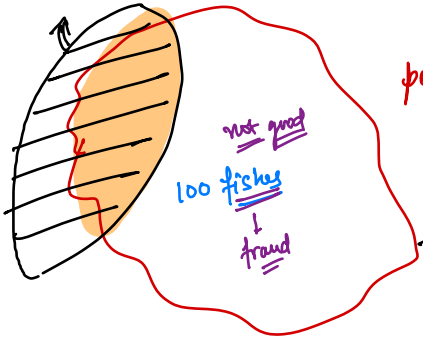


Fraud Analytics

good

Fishing

Accuracy

100 fishes
100 Actually
= Recall 100%

(what you predicted
Actually

Crocodile, Stones, Crabs

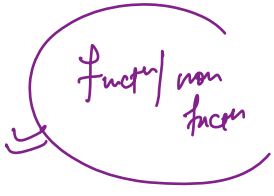
$$= \frac{\text{Intent to captured (what predictor is true)}}{\text{Actual true}}$$

Caught fish

$$\left[\frac{}{100 + 50 + 20 + 20} \Rightarrow \frac{100}{190} \Rightarrow \text{Precision} \right]$$

AWS Service (End-to-End)

Real time → fraud detection



System Component

[10,000 Transaction per Seconds]

✓✓ High throughput.

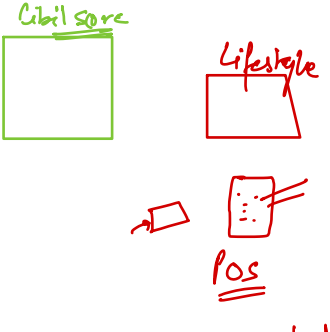
1. Kinesis Data Streams → Ingesting transaction
2. Kinesis Data Analytics (Apache flink)
3. Dynamo DB → low latency lookups

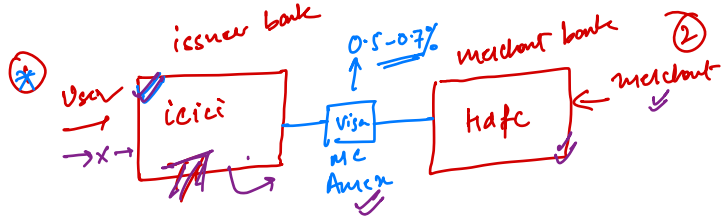
→ Store id = 123 ⇒ fetch the details of store (Metadata)

Data

```
{ "txn id" 1234
  "User id" : "abcd" , "User_1234"
  "amount" : 50,000
  "timestamp" :
  "merchant id" : ____
  "ip address" : ____
  "location" : ← }
```

partition Userid





Max Share → ISSUER Bank → Max risk

[Configuration of Kinesis
+ 4 Shards (1 shard → 1 MB/sec input
1000 records/sec)

+ partition Key = store_id / User-id

② Real time feature engineering (flink)

Agg — User-456 → 1 hr rolling spend
→ Txn happened in last 10 mintues

Enrichment

DynamoDB

fetch details merchant name, account age, past fraud record

↓

Credit Score

5 txn in 10mins
150 km/h

```
{
  "transaction_id": "txn_123",
  "features": {
    "user_hourly_spend": 4500.00,
    "merchant_risk_score": 0.85,
    "transaction_frequency_10min": 5,
    "location_velocity": "150 km/h"  // Distance from last transaction
  }
}
```

A Delhi
B Noida
Kanpur

Amazon S3 (trained models, artifacts)

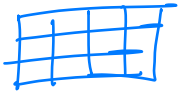
Model training & Deployment

AWS

Training, hosting & monitoring model
↳ Sagemaker

EMR (spark) Batch feature engineering for historical data [Optional]

Transaction

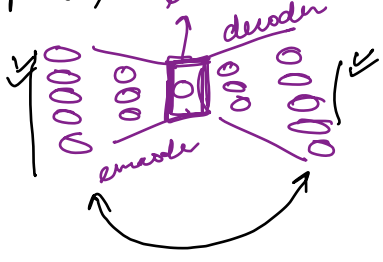
↳ Redshift (AWS) → 

DB [Optional]
Athena
↳ S3

Label → 0, 1

↳ isolation forest (Unsupervised)

↳ autoencoders (Un)



→ XGBOOST (Supervised)

Configuration

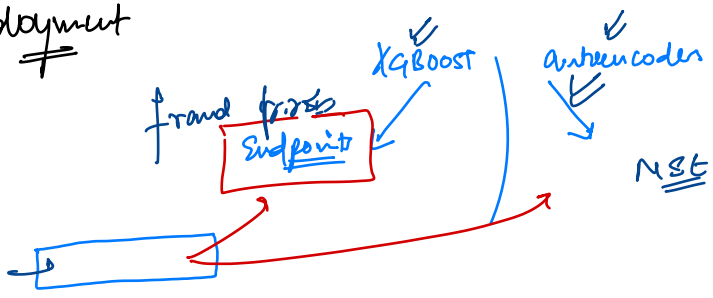
+ Deep learning → GPU

Precision | Recall

ml.p3.2xlarge]

+ XGBOOST → CPU ml.m5.4xlarge ↳ cost efficient

Deployment



Autoencoders → Computing reconstruction error (MSE)

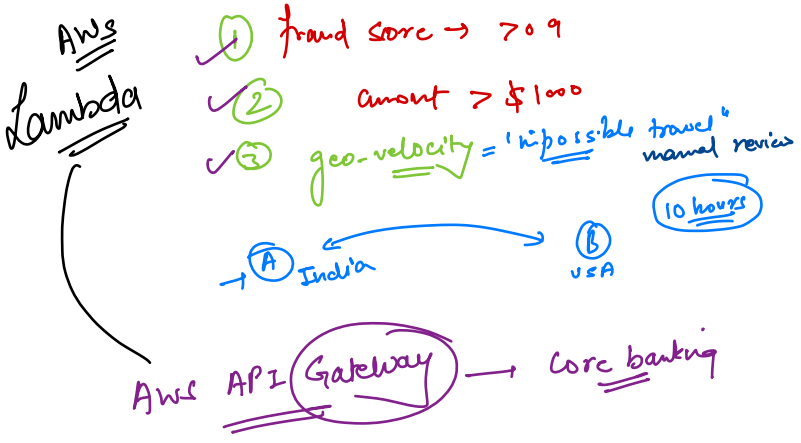
↳ Anomaly MSE > 0.5]

Combined fraud score

↳ 0.7 * XGBboost_prob + 0.3 (autoencoder_mse / max_mse)

Real time - decision making

(1) Block transactions



SMS / email
↳ Amazons SNS

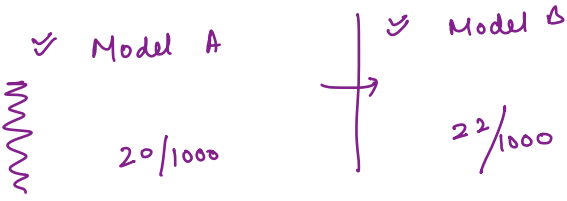
Monitoring / Retraining phase

(1) Data Drift
Statistical test

training | production data

1. KS test
2. Chi-Sq
3. KL divergence

Experimental



→ t-test

→ two-proportion Z-test

Q

Historically → 0.1%

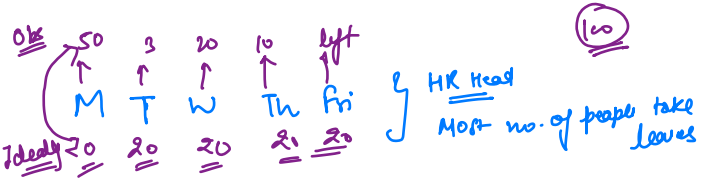
Today → 10,000

Fraud → 100

Observed value → 120

→ t-test

→ z-test



① Real time data processing

↳ Kinesis + Flink

② Hybrid model → Combined score

(Uns + Sup)

③ Automated trigger

① → Metric shift (Precision, Recall)

② → Data drift (KL, KS, Chi)

③ Model A → Model B (t-test, z-test, two-prob z test)

④ Cost Efficiency
- GPU
- CPU

⑤ Alert System → Lambda + API Gateway + SNS

https://machinelearningmastery.com/statistical-hypothesis-tests-in-python-cheat-sheet/