Imagine you were working on iPhone. Everytime users open their phones, you want to suggest one app they are most likely to open first with 90% accuracy. How would you do that?

Functional Requirement

1. Real time prediction — 100 ms of phone unlock
2. Personalised recommendations → Morning app usages / Evening app usage

   Adapt with user habit
3. offline availability → offline fallback → cache predictions

   ↳ 24 hrs on-device
4. Privacy → ~~GPS~~, user-id hashed

Non-Requirements

1. Latency low (Unlock → prediction) < 100 ms
2. Scalability → Handle 10Mn daily active users

   (1100+ req/sec at peak)
3. Available → 99.99%
4. Security → Encrypt
5. Cost → ↓ 0.001 (Sagemaker, ECS, EC2)

Data Ingestion & Storage Layer.

1. User Behaviour

- Apps during diff times
- Apps opened after unlocking (food → Swiggy, fintech → ≡)
- Time since last unlock
- session duration

2. Contextual Signals

- Time of day, location
- Device Status (ios version, Battery, wifi vs Cellular data, storage space ...)

  ios version → OS

Historical Patterns

- Most used app last 24 hours, week . . .
- frequency of app switches (imessages, slack)

Storage (AWS)

Real-time data → Kafka / Amazon Kinesis Data Streams → Raw data archives, processed S3 (partition using date/hour)

Batch → S3 → Glue Data Catalog.

Features Engineering & Processing

Real-time

- → Compute time since last app used.
  {type}

[User_123 → 8:30 AM]

{"hour" : 8 , location : "New Delhi",
→ "last_opened_app2" : "Calendar"}

Batch

- → Aggregated daily app usage
  [Messages 15 times in a week]
- → Pyspark SQL → Ranking
  time spent on the app
  over user id , hour

S3 → store batch features

Store real-time info (key-value)

+ user 8:00 AM
+ user 9:00 AM

(DynamoDB)

https://aws.amazon.com/dynamodb/

Machine Learning Pipeline

90%

1. Data Versioning → Track datasets S3 Object versioning

2. Training → Sagemaker → ml.c5.2xlarge (spot instance)

+ Hyperparameter tuning (Dept... , ,)

3. Validation

+ Holdout validation 30 days time-series split)

AUC-ROC

4. Model Registry

+ Version model in Sagemaker Model Registry

Production / Staging

5. Deployment

- Sagemaker training
  - validation → deployment.

Step function

90% Accuracy

- trigger retraining if accuracy drops < 85%

https://aws.amazon.com/step-functions/

Model → ML model selection

1. LightGBM
   - → < 10ms
   - → interpretable
   - → SHAP values

2. XGBOOST
   - → < 10 ms
   - < interpretable
   - → sequential data (struggles) → app usage history.

3. Transformer Models
   - app sequences (Calendar → [ ] [ ] → Zerodha → Drivent)
   - GPU inference
   - latency could be higher

+ complex patterns

Optional Extension

A/B testing → Sagemaker AB Testing

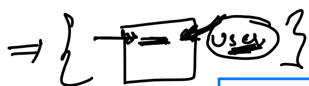
↳ 5% traffic to challenger model

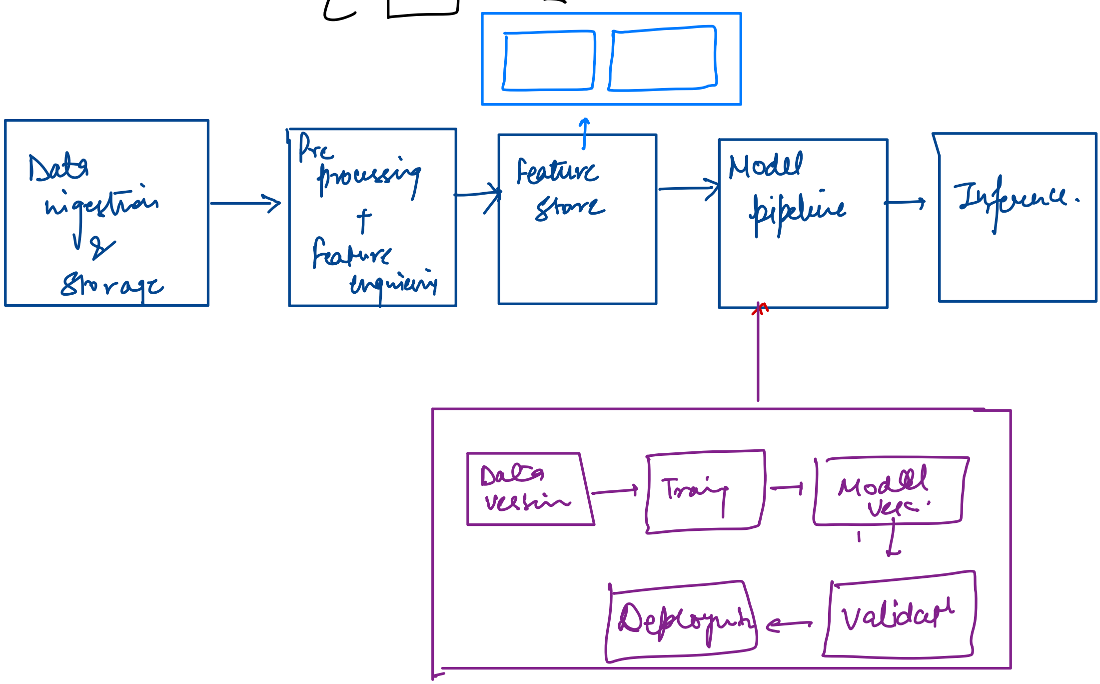
https://aws.amazon.com/blogs/machine-learning/a-b-testing-ml-models-in-production-using-amazon-sagemaker/

Canary deployment

↳ 1% users in specific region

Federated learning →





User → API Gateway → Lambda (input validation) → Sagemaker endpoint

Example req → { User id: 123
timestamp:
features: { hour: 8, location: new Delhi,
last_app_used: Calendar } }

{ battery: 75%, network: – }

{ score: 10 }

H/W ↓

An e-commerce company is trying to minimize the time it takes customers to purchase their selected items. As a machine learning engineer, what can you do to help them?