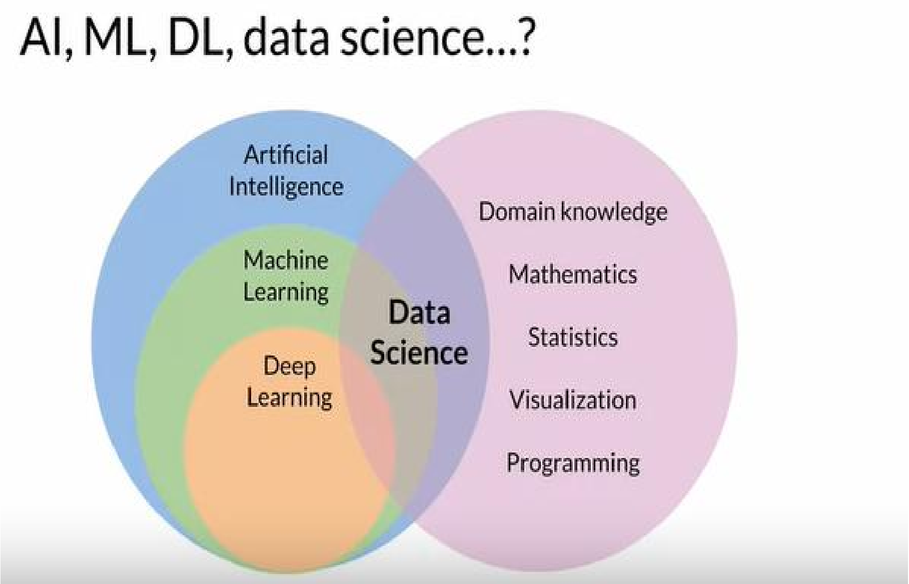
Practical Data Science with AWS Cloud

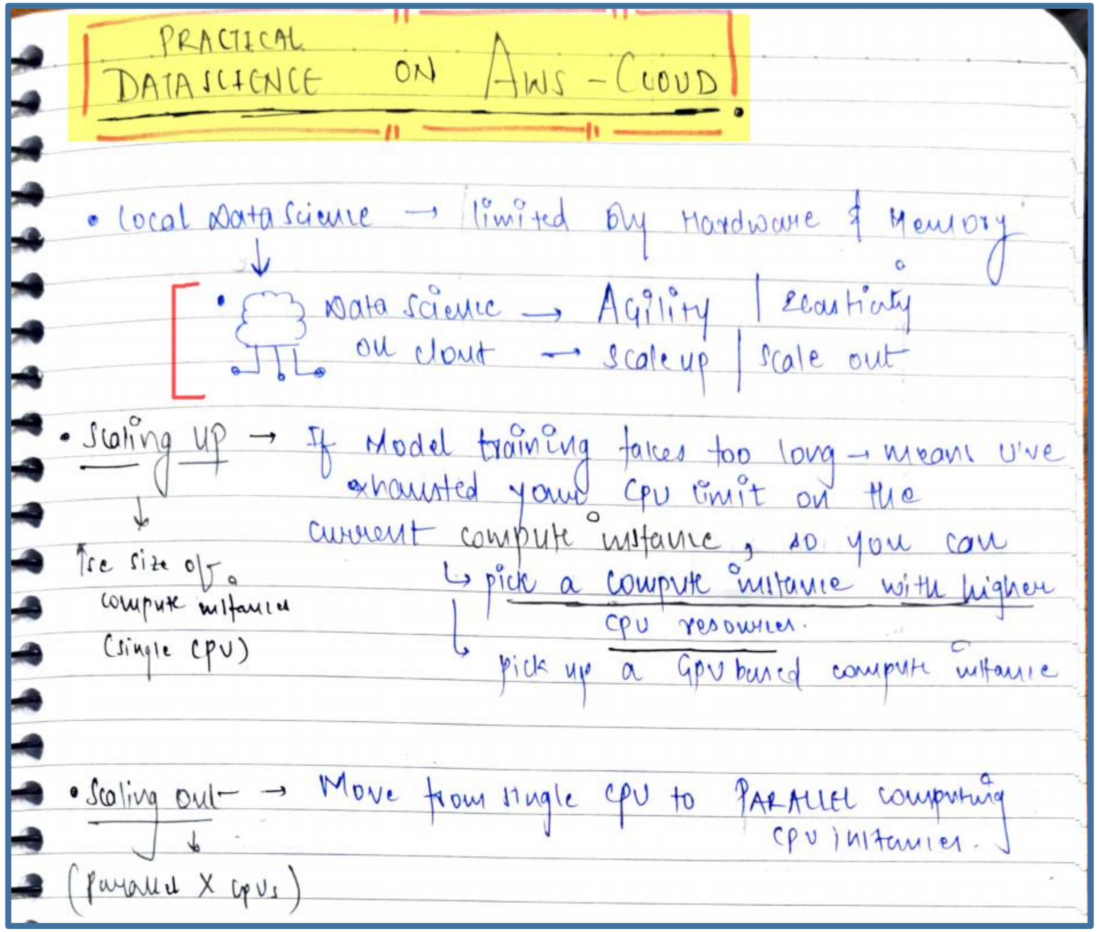
(Hand-written notes and Course Gist)

Divyanshu Vyas
Energy Data Scientist, Petroleum From Scratch



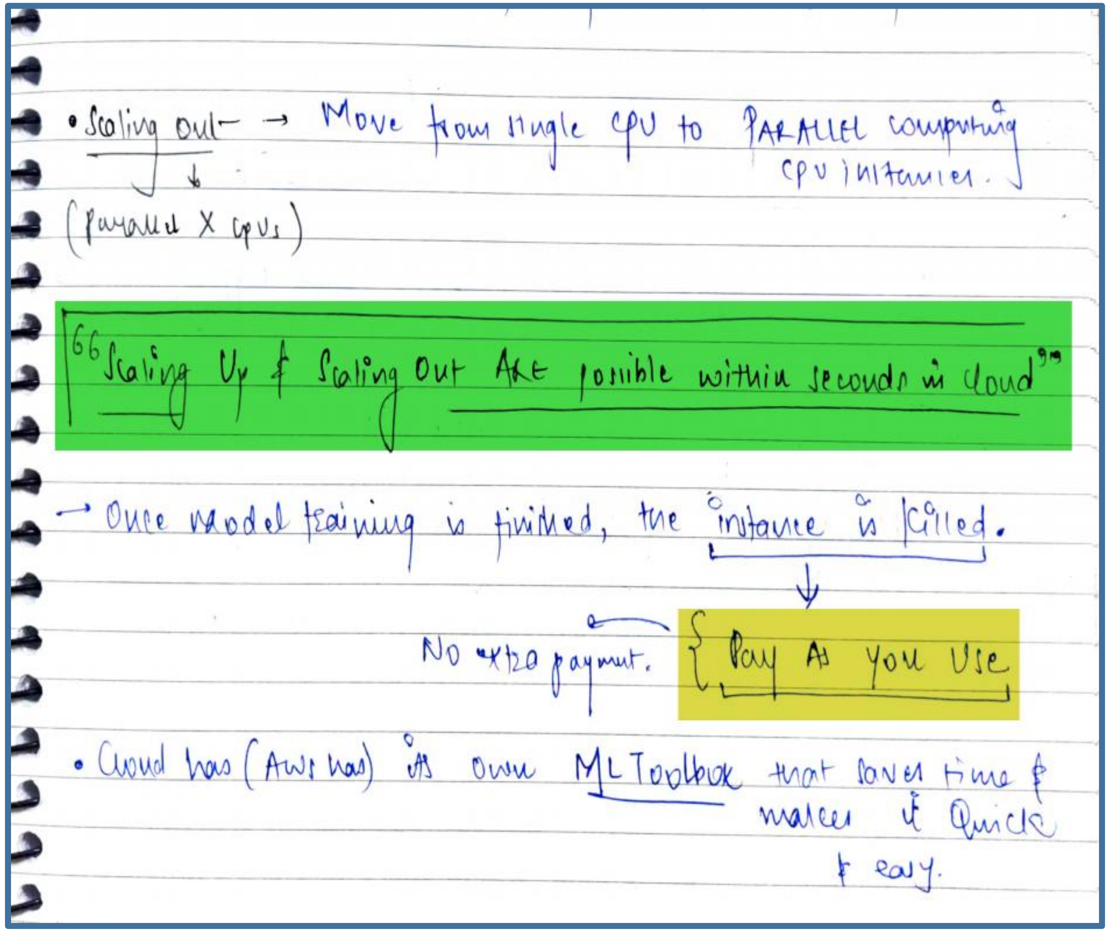
DAY 1

1. Baiscs about AWS cloud : Scaling Up & Scaling Out abilities



PRACTICAL DATASCIENCE ON AWS - CLOUD

- Local Data Science → limited by Hardware & Memory
  - ↓
  - Data Science on cloud → Agility | Elasticity → Scale up | Scale out
- Scaling up → If Model training takes too long → means u've exhausted your CPU limit on the current compute instance, so you can
  - ↳ pick a compute instance with higher CPU resources.
  - ↳ pick up a GPU based compute instance
  - ↓ Inc size of compute instances (single CPU)
- Scaling out → Move from single CPU to PARALLEL computing CPU instances.
  - ↓ (parallel X CPUs)



- Scaling out → Move from single CPU to PARALLEL computing CPU instances.
  - ↓ (parallel X cpus)

"Scaling Up & Scaling Out Are possible within seconds in cloud"

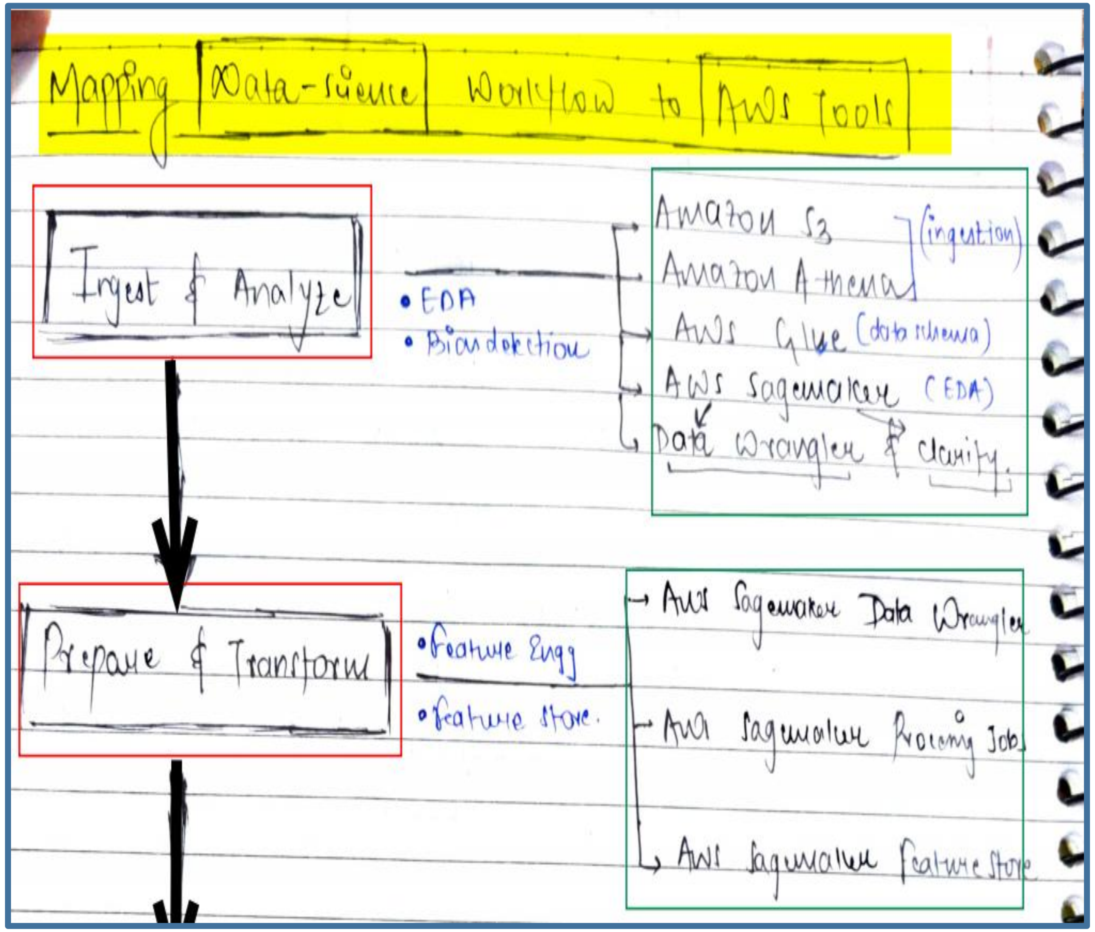
→ Once model training is finished, the instance is killed.

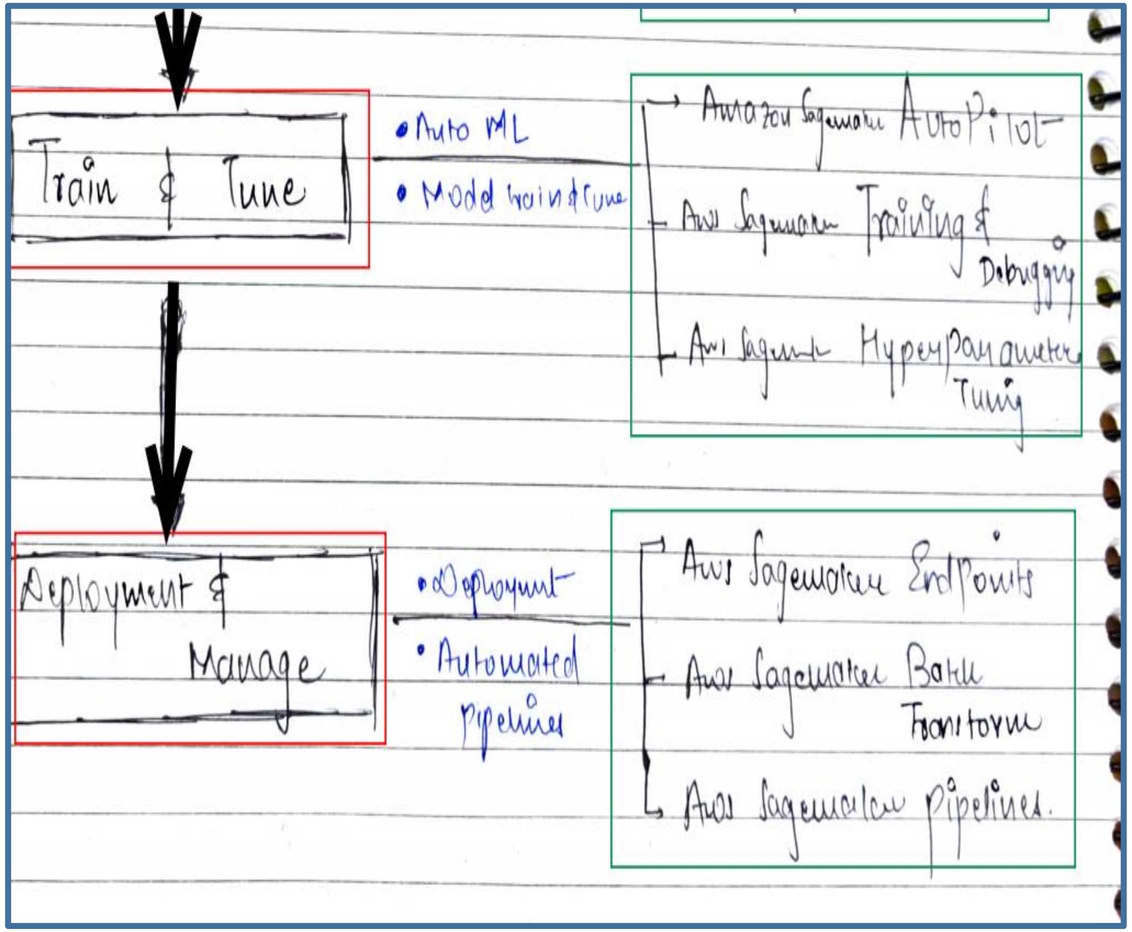
↓

{Pay As you Use} → No extra payment.

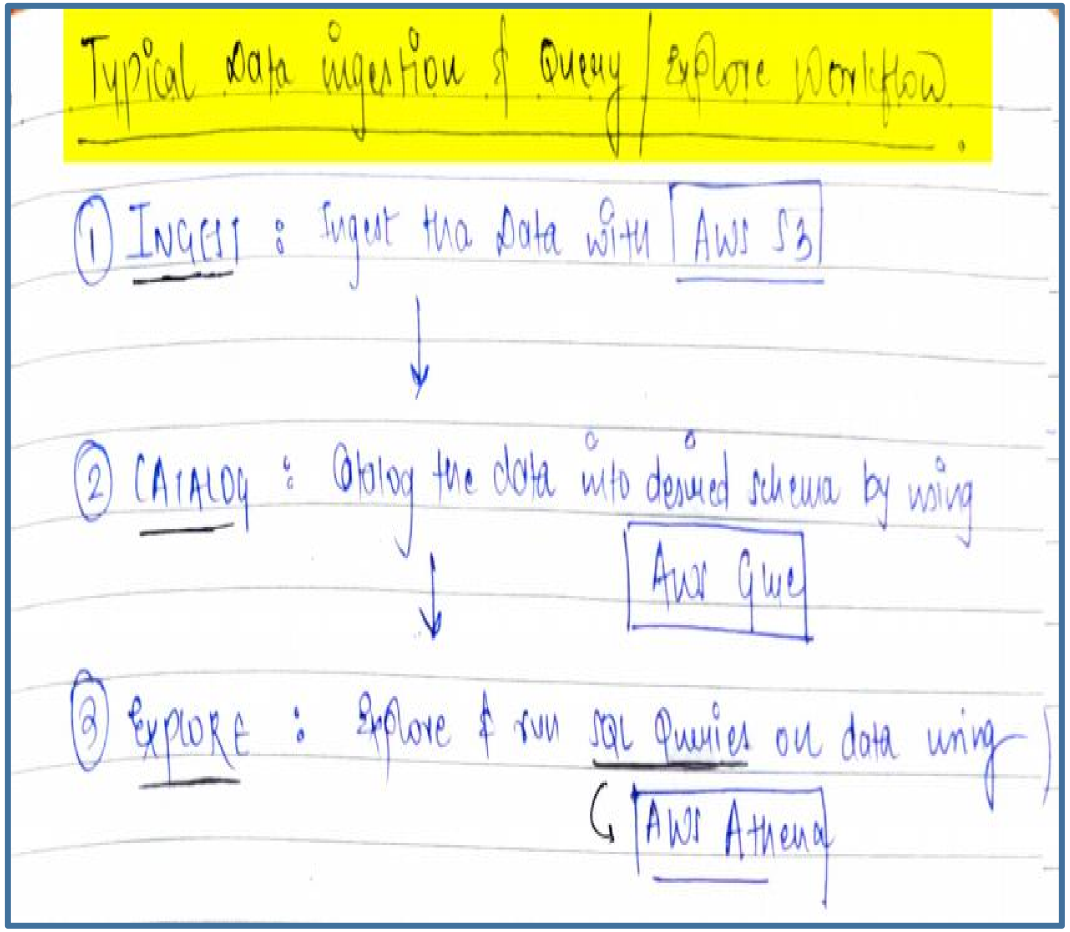
- Cloud has (AWS has) its own ML Toolbox that saves time & makes it Quick & easy.

2. Mapping Data Science workflow with AWS Tools





3. Mapping Data Science workflow with AWS Tools



Typical Data ingestion & Query / Explore Workflow.

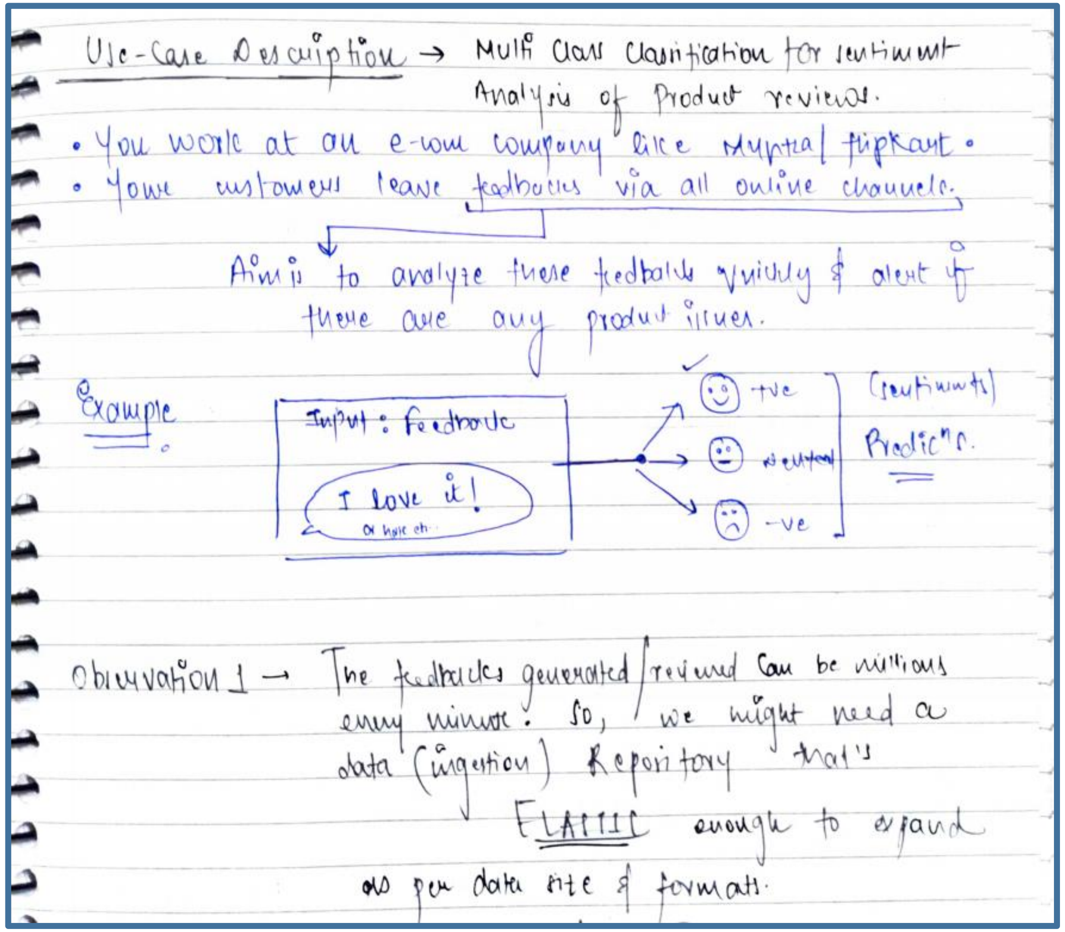
① INGEST : Ingest the Data with AWS S3

↓

② CATALOG : Catalog the data into desired schema by using AWS Glue

↓

③ EXPLORE : Explore & run SQL Queries on data using AWS Athena



Use-Case Description → Multi Class Classification for sentiment Analysis of Product reviews.

- You work at an e-com company like Myntra/ flipkart.
- Your customers leave feedbacks via all online channels.

Aim is to analyze these feedbacks quickly & alert if there are any product issues.

Example.

Input : Feedback

I love it! or hate etc

+ve

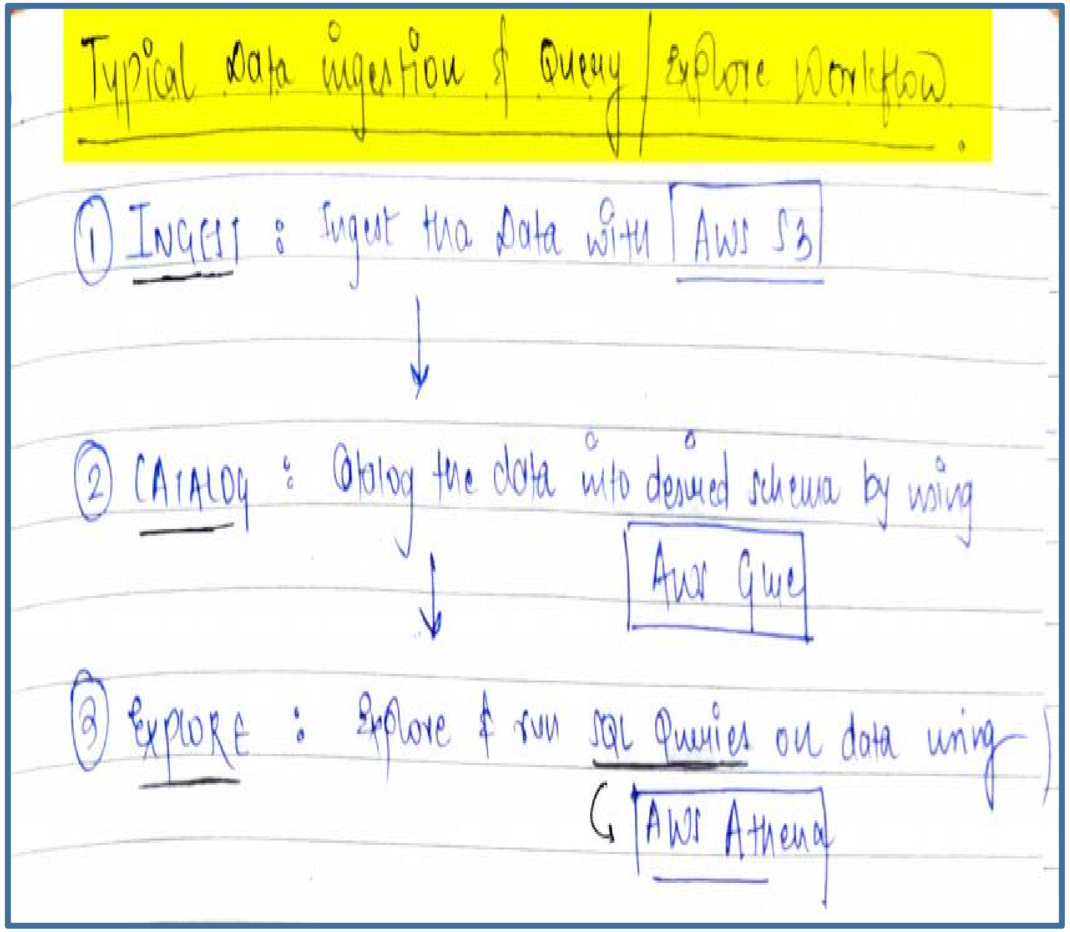
neutral

-ve

(sentiments) Predic^n^s.

Observation 1 → The feedbacks generated/reviewed can be millions every minute. So, we might need a data (ingestion) Repository that's ELASTIC enough to expand as per data size & formats.

4. Example Use case



Typical Data ingestion & Query / Explore Workflow.

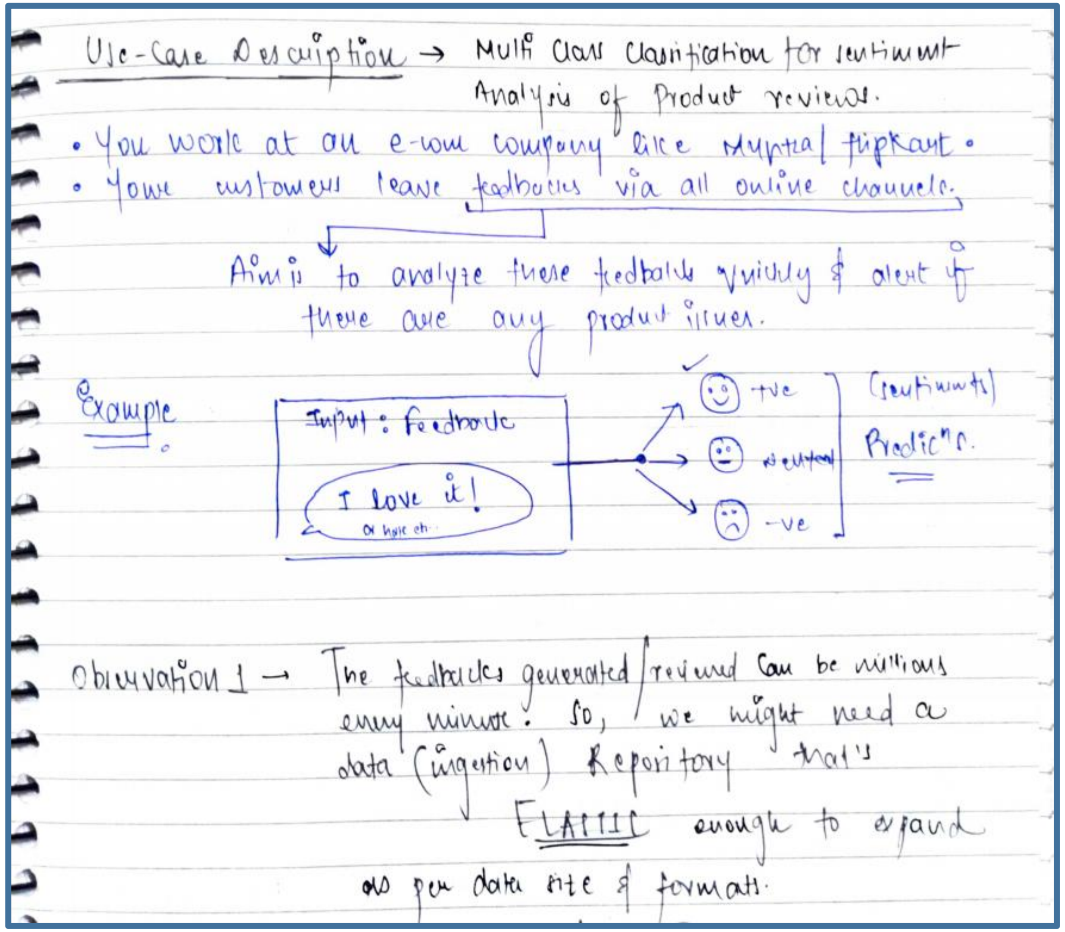
① INGEST : Ingest the Data with AWS S3

↓

② CATALOG : Catalog the data into desired schema by using AWS Glue

↓

③ EXPLORE : Explore & run SQL Queries on data using AWS Athena



Use-Case Description → Multi Class Classification for sentiment Analysis of Product reviews.

- You work at an e-com company like Myntra / flipkart.
- Your customers leave feedbacks via all online channels.

Aim is to analyze these feedbacks quickly & alert if there are any product issues.

Example.

Input : Feedback

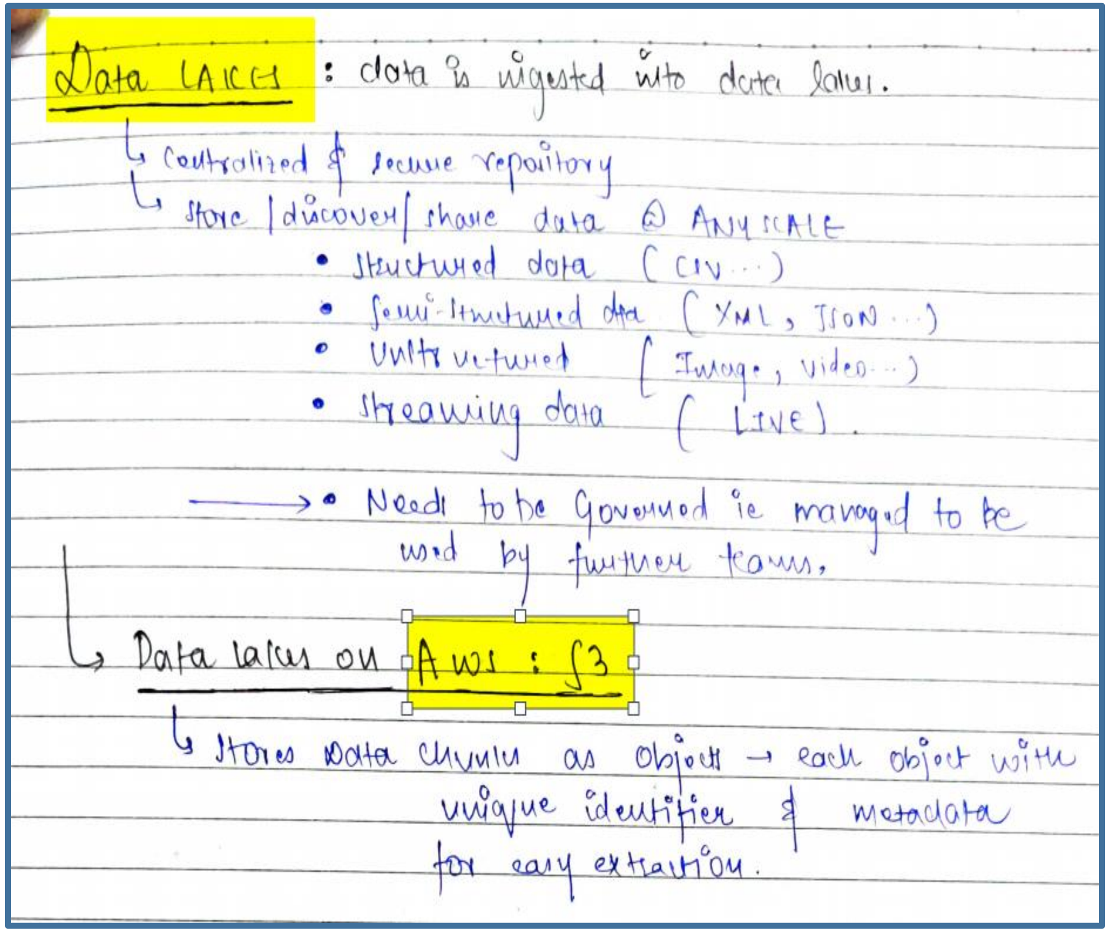
I love it!
Or hate it

→ ☺ +ve ✓
😐 Neutral
☹ -ve

(sentiments) Predic^{ns}.

Observation 1 → The feedbacks generated / reviewed can be millions every minute. So, we might need a data (ingestion) Repository that's ELASTIC enough to expand as per data size & formats.

5. Data Lake (AWS S3) , Data Wrangler, AWS Athena & AWS Glue



Data LAKES : data is ingested into data lakes.

↳ Centralized & secure repository

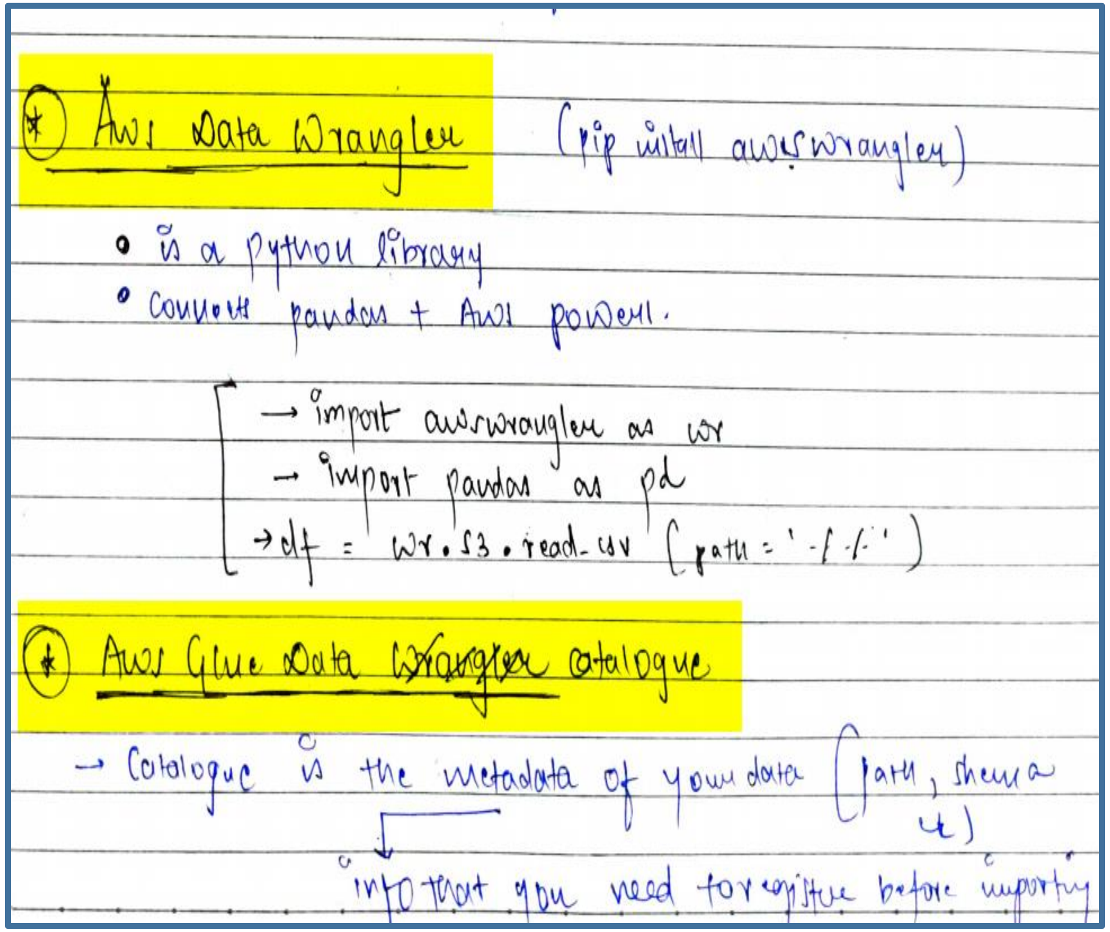
↳ Store | discover | share data @ ANY SCALE

- Structured data (CSV ...)
- Semi-structured data (XML, JSON ...)
- Unstructured (Image, video ...)
- Streaming data (LIVE).

→ • Needs to be Governed ie managed to be used by further teams.

↳ **Data lakes on AWS : S3**

↳ Stores data chunks as objects → each object with unique identifier & metadata for easy extraction.



⊛ **AWS Data Wrangler** (pip install awswrangler)

- is a python library
- Connects pandas + AWS powers.

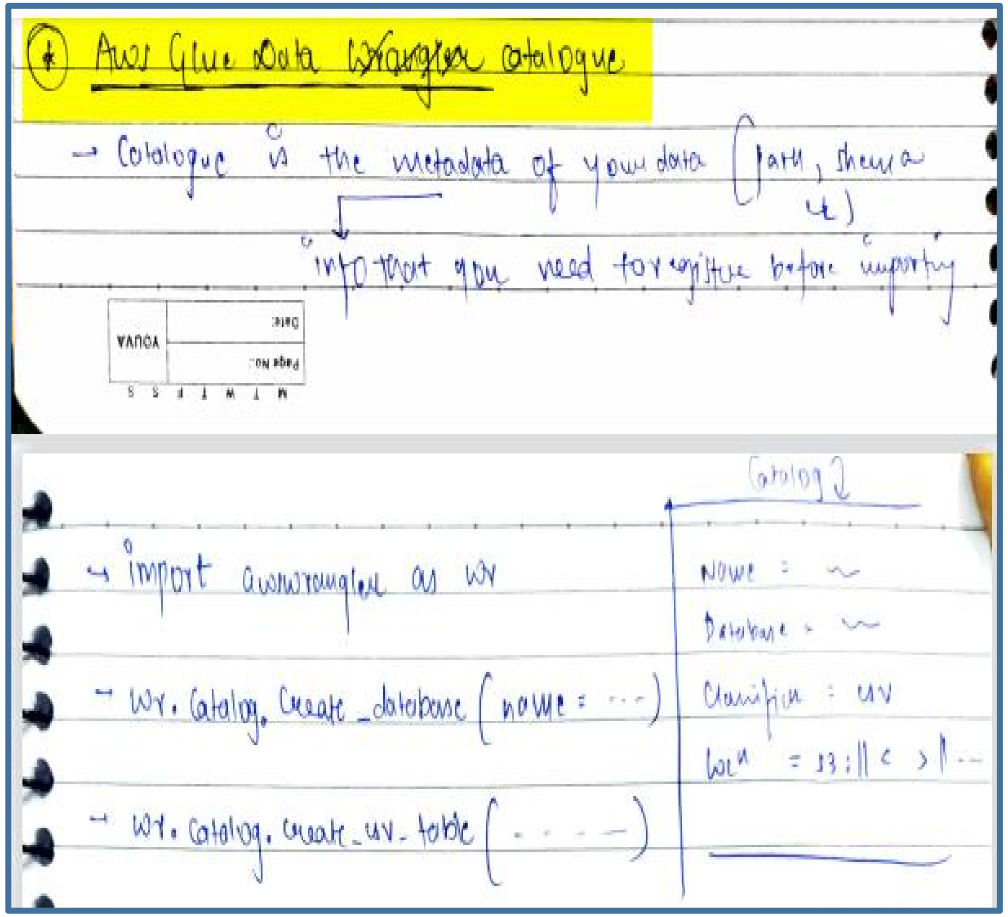
```
→ import awswrangler as wr
→ import pandas as pd
→ df = wr.s3.read_csv (path = '-/-/-')
```

⊛ **AWS Glue Data Wrangler catalogue**

→ Catalogue is the metadata of your data (path, schema etc)

↳ info that you need to register before importing

5. Data Lake (AWS S3) , Data Wrangler, AWS Athena & AWS Glue



(*) Aws Glue Data Wrangler catalogue

→ Catalogue is the metadata of your data (path, schema etc)

info that you need to register before importing

Catalog 2

→ import awswrangler as wr

→ wr.catalog.create_database(name = ...)

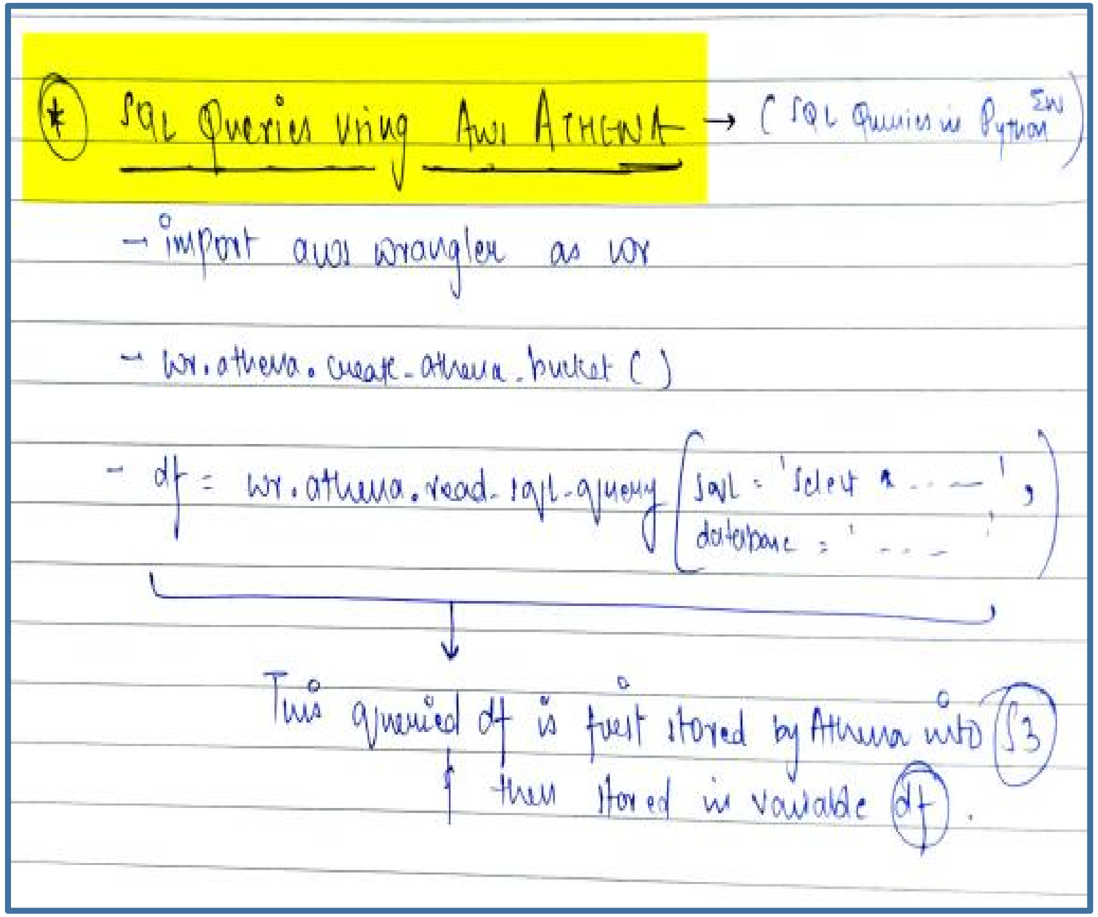
→ wr.catalog.create_csv_table(. . . .)

Name : ~

Database : ~

Classification : csv

Locn = s3://< >/...



(*) SQL Queries using AWS ATHENA → (SQL Queries in Python Env)

→ import aws wrangler as wr

→ wr.athena.create_athena_bucket ( )

→ df = wr.athena.read_sql_query(sql = 'select * ...', database = '...')

This queried df is first stored by Athena into S3 & then stored in variable df.