Recap

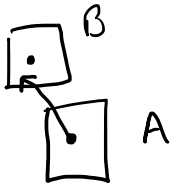
Demand Prediction → NYC taxi dataset

Rows → 1 Ride → Pickup location
→ Drop location
↓
Pickup time
↓
No. of passenger

Predict → demand → At a given time interval → 15 min
and
at a particular region
↓
No. of pickups

Time → 8:00 pm

8:01 pm → demand α



next minute

More rides → more share

We can predict / forecast more accurately

25°C → 24°C → 26°C

25°C → 18°C

25°C → confidence level ↓

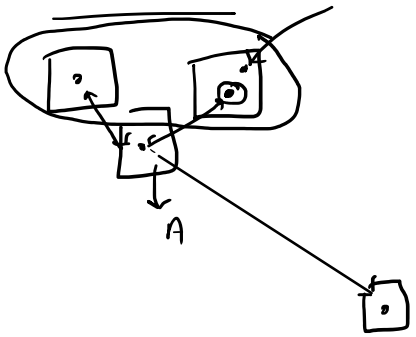
15 min interval → Enough time to move to a nearby region

↓ More accurate

Region → NYC → unsupervised ML → Clustering

↙ k Means clustering

Centroid



distances → Region center

Sort on the basis of distances.

Centroids

↓

Identity of our region

nearby locations

EDA → How we can divide → regions (Task 1)

How we can divide the time axis (Task 2)

↓

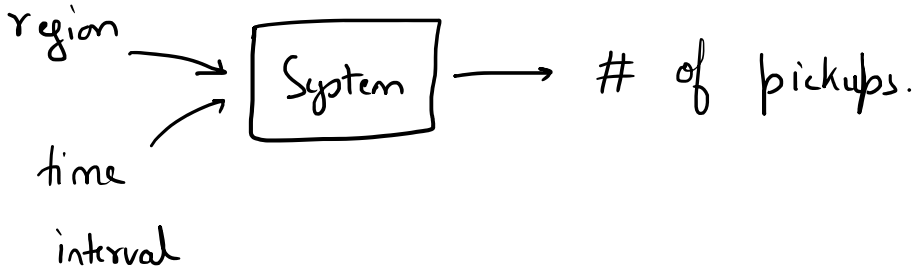
← intervals

15 min intervals

for a given region and for a given time interval

↓

Predict # of pickups

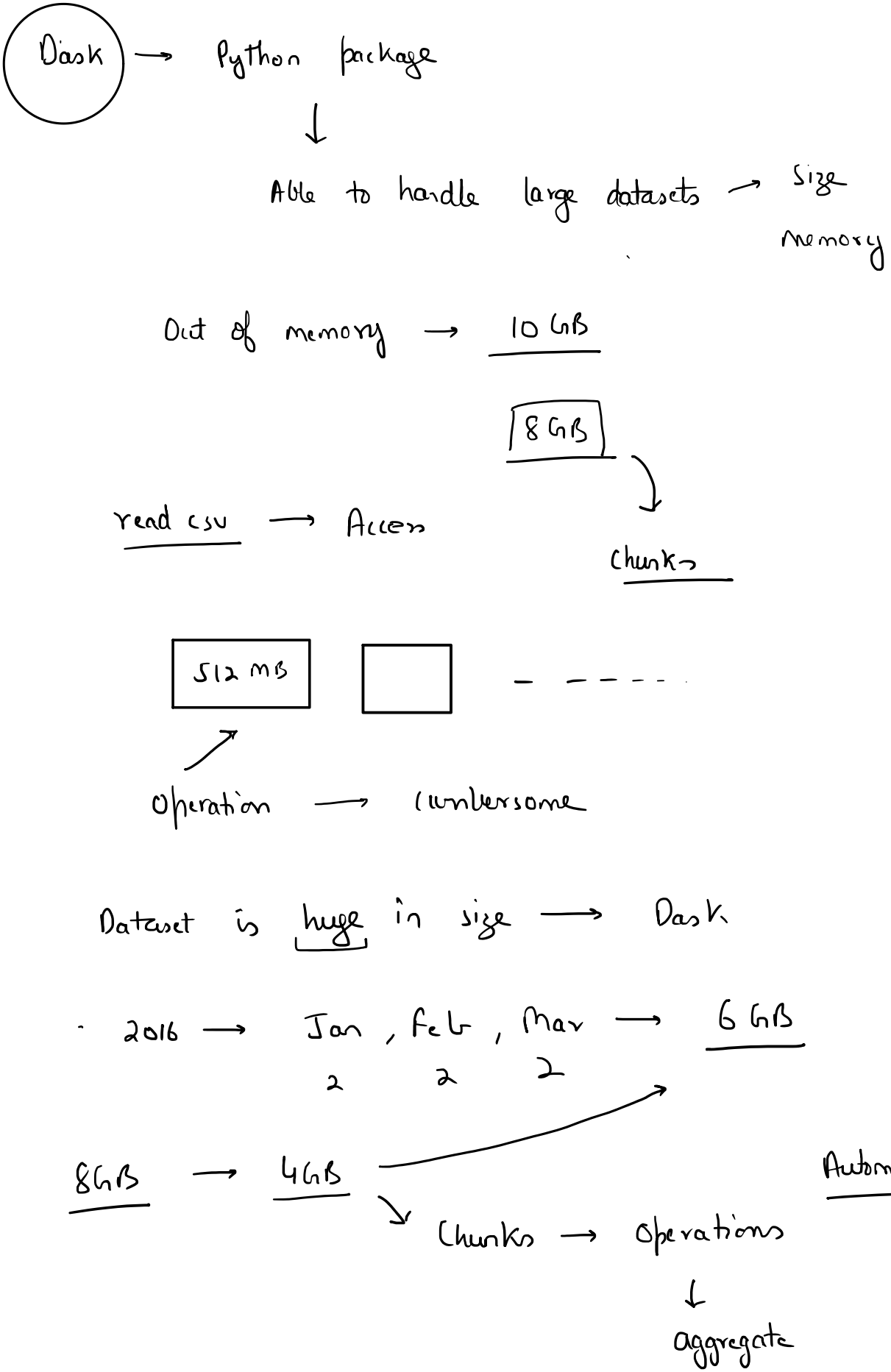


Metric → MAPE

App → Streamlit app

location
time interval
→ Graph → NYC → Regions
↓
Demand predict
↙
More darker ↑ ← Color coding
More lighter ↓

What is Dask?


Dask → Python package
Able to handle large datasets → Size
Memory
Out of memory → 10 GB
8 GB
read csv → Access
Chunks
512 MB
Operation → cumbersome
Dataset is huge in size → Dask
2016 → Jan, Feb, Mar → 6 GB
2 2 2
8 GB → 4 GB
Automatically
Chunks → Operations
aggregate

Why use it

Dataset → 2016 → Jan, Feb, Mar

↓

2GB each → 6GB

Chunking → Parts

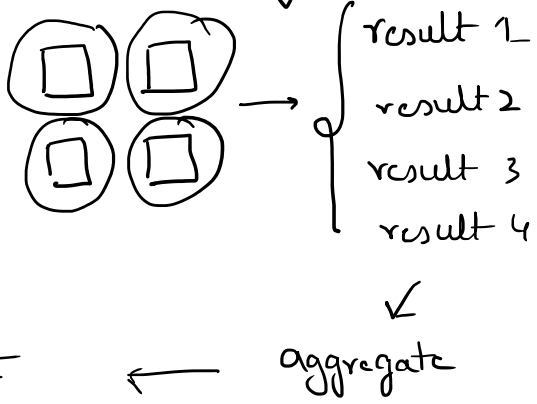
Dask dataframes → Pandas dataframe

Dask arrays → numpy arrays

Handles all the chunking operations automatically

+

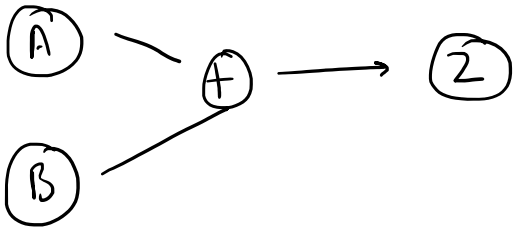
Parallelize → CPU core



Dask → Pytorch computation graphs

Graph store

A + B → Z



Dask → Task graphs

Jan 2016 → load as a dask df.

2GB → metadata store

- No. of Columns
- Column dtype
- Index

dask df →

- Chunks
- No. of partitions

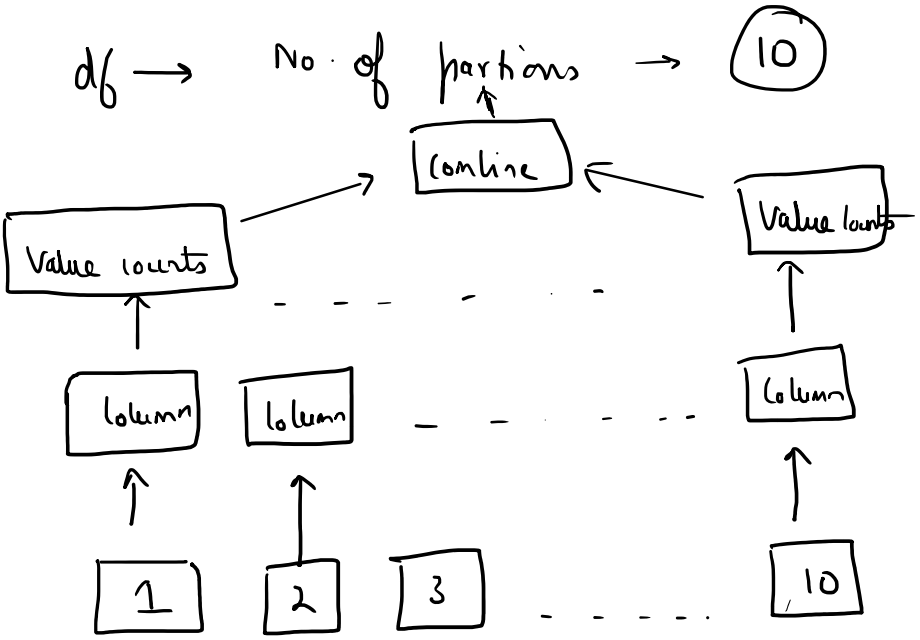
	Column A	Column B	Column C
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

Categorical → value counts

df['column name'].value_counts().sum()

Index

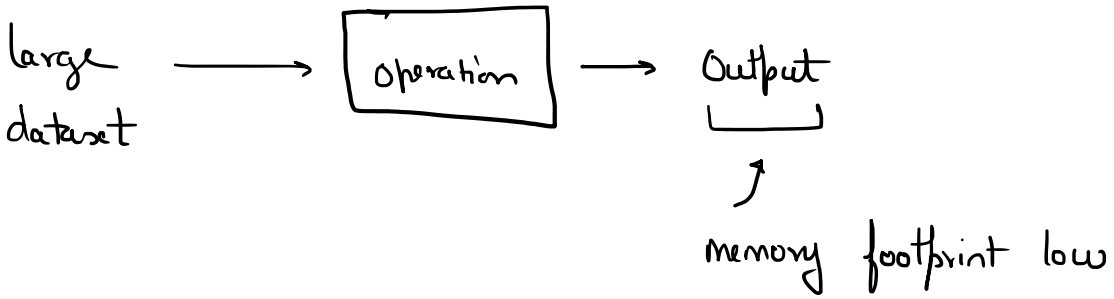
category A —
B —
C —



Return a series

Dask → compute() → scalar value

Chain operations


large dataset
Operation
Output
memory footprint low

Things to avoid

1) Compute → Uses a lot of compute

2) Try to use pandas → 6GB ← Dask