Recap

Demand Prediction → # of pickups

- Overview
- EDA → Dask, compute
- Break into regions

Sample → Pickup pattern on NYC

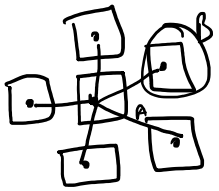
Outliers — lat, long
Fare
Trip distance

3.3 crore → lat, long → Scale

K means clustering → Euclidean distances

Partial fit + chunking in pandas

Mini Batch K means



→ inter cluster distance

K → 10, 90

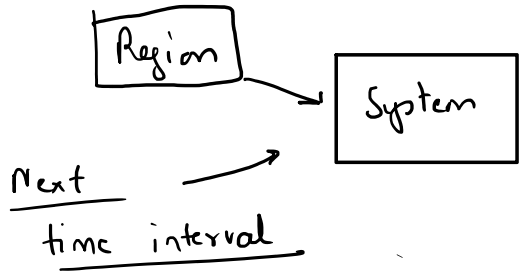
no. of neighbors = 8, Avg inter cluster distances

K → No. of regions

K = 30

1 – 1.5 miles → 15 min intervals → 1 - 1.5 miles in NYC

Region



Neighboring regions

$t = 0$	9:00 pm	region A	
$t = 1$	9:15 pm	region B	→ demand
			# of pickups

Task 1 → Breaking NYC into regions

30 regions

Task 2 → Calculating historical data.

for every region.

15 min intervals → # of pickups

Region	Time interval	# of pickups
A	5:15 pm	100

Each row = 1 separate ride.

Pickup time, lat, long

↓

instance

Time-Binning.

5:07 pm.
5:03 pm.
5:04 pm.

→ 5:00 pm – 5:15 pm.

↑ 3 pickups

↗ 2

5:20 pm.
5:17 pm] → 5:15 pm – 5:30 pm

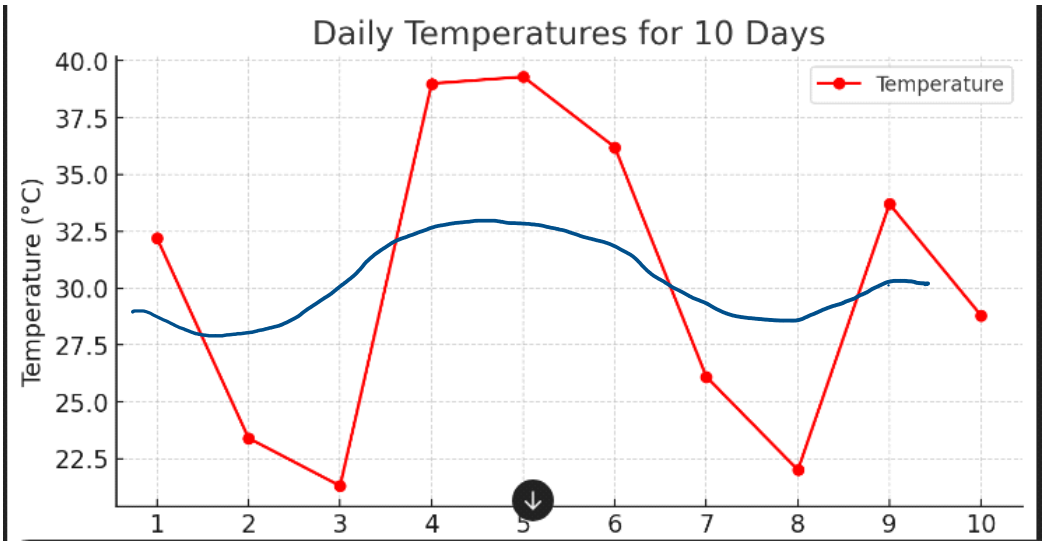
intervals → Count the # of pickups

data can fluctuate. Trend ∝

We cannot predict with high accuracy for the next time interval.

Delhi Variations → Trends

	Day	Temperature (°C)
0	1	32.2
1	2	23.4
2	3	21.3
3	4	39.0
4	5	39.3
5	6	36.2
6	7	26.1
7	8	22.0
8	9	33.7
9	10	28.8



less variations. Constant trend

Day 11 → Trend → Predictions more accurate

Smoothing → more accurate prediction

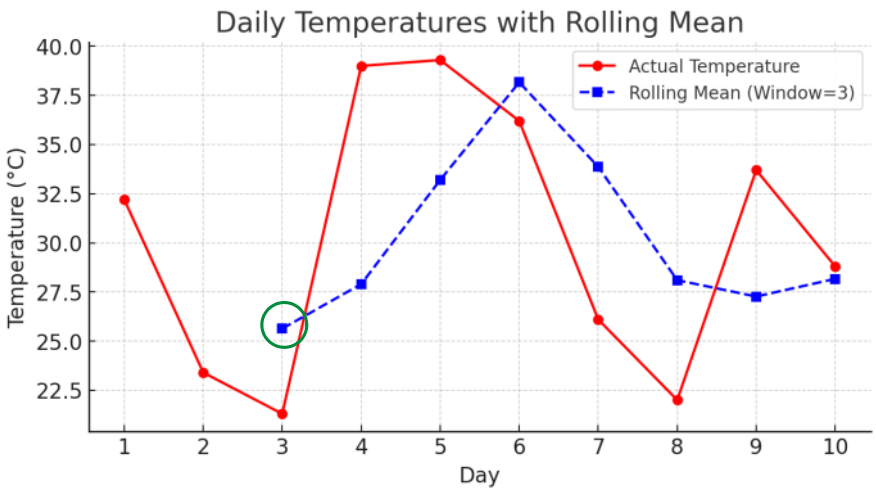
Trend →

MA → Moving average

EWMA → Exponentially weighted MA.

Moving Averages (MA)

	Day	Temperature (°C)	Rolling Mean (°C)
0	1	32.2	NaN
1	2	23.4	NaN
2	3	21.3	25.633333
3	4	39.0	27.900000
4	5	39.3	33.200000
5	6	36.2	38.166667
6	7	26.1	33.866667
7	8	22.0	28.100000
8	9	33.7	27.266667
9	10	28.8	28.166667



Smooth out the data

MA → Window = 3 days.

avg along this window

Window ↑ → smoothing factor ↑

Window = 3

t_0

t_0, t_{-1}, t_{-2}

window = 5

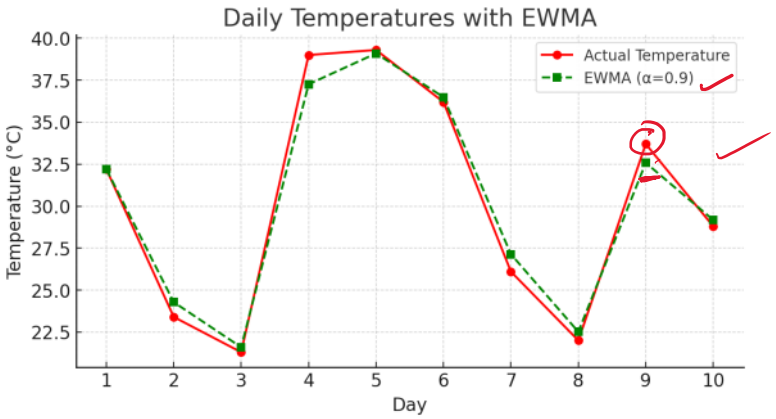
t_0 → smoothing factor

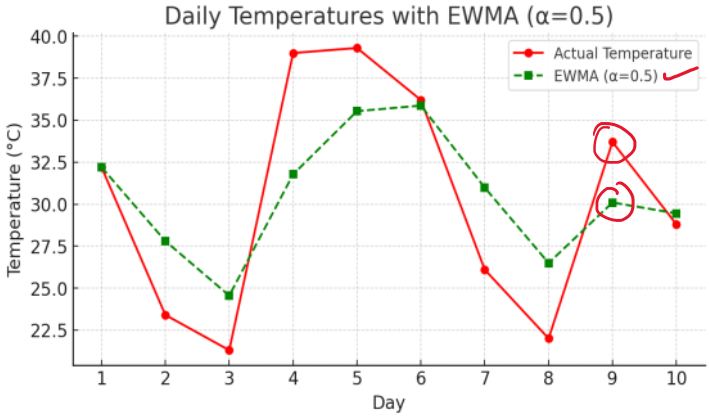
$t_0, t_{-1}, t_{-2}, t_{-3}, t_{-4}$

EWMA

100 days of DL

	Day	Temperature (°C)	Rolling Mean (°C)	EWMA (α=0.9)	EWMA (α=0.5)
0	1	32.2	NaN	32.200000	32.200000
1	2	23.4	NaN	24.280000	27.800000
2	3	21.3	25.633333	21.598000	24.550000
3	4	39.0	27.900000	37.259800	31.775000
4	5	39.3	33.200000	39.095980	35.537500
5	6	36.2	38.166667	36.489598	35.868750
6	7	26.1	33.866667	27.138960	30.984375
7	8	22.0	28.100000	22.513896	26.492188
8	9	33.7	27.266667	32.581390	30.096094
9	10	28.8	28.166667	29.178139	29.448047





$$
\begin{aligned}
s_t &= \alpha x_t + (1-\alpha)s_{t-1} \qquad \alpha = 0.9 \\
&= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2 s_{t-2} \\
&= \alpha\left[x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + (1-\alpha)^3 x_{t-3} + \cdots + (1-\alpha)^{t-1} x_1\right] + (1-\alpha)^t x_0.
\end{aligned}
$$

Weighted moving average.

Recent observations → Equal weight

Past observations

Window = 3

Day 3 → (Day 1 + Day 2 + Day 3) / 3

Equal weight.

Current observation ↑

Past observation ↓

Stocks

NVIDIA → (x) ↓ → MA

avg ↓

Current trend → EWMA

Weighing my current observation ↑

past observation ↓.

Exponentially decaying in nature.

Day $t-1 = x$

$t-2 = x^2$

$t-3 = x^3$ . . .

$S_t = \alpha x_t + (1-\alpha) S_{t-1}$

$x_t \rightarrow$ current data point t

$S_{t-1} = \alpha x_{t-1} + (1-\alpha) S_{t-2}$.

$S_{t-1} =$ $t-1$ prediction.

$S_{t-2} = \alpha x_{t-2} + (1-\alpha) S_{t-3}$

$\alpha = 0.9$

$0.9(x_t) + (0.1) S_{t-1}$

$$S_t = 0.9 x_t + 0.1 \left(0.9 x_{t-1} + 0.1 S_{t-2}\right)$$

$$= 0.9 x_t + \left(\overset{1}{0.1} \times 0.9 x_{t-1}\right) + (0.1)^2 S_{t-2}$$

$\alpha \uparrow \longrightarrow$ focus is more on current observations

More weightage to current observations

$\alpha \downarrow \longrightarrow$ Some more weightage to past observations

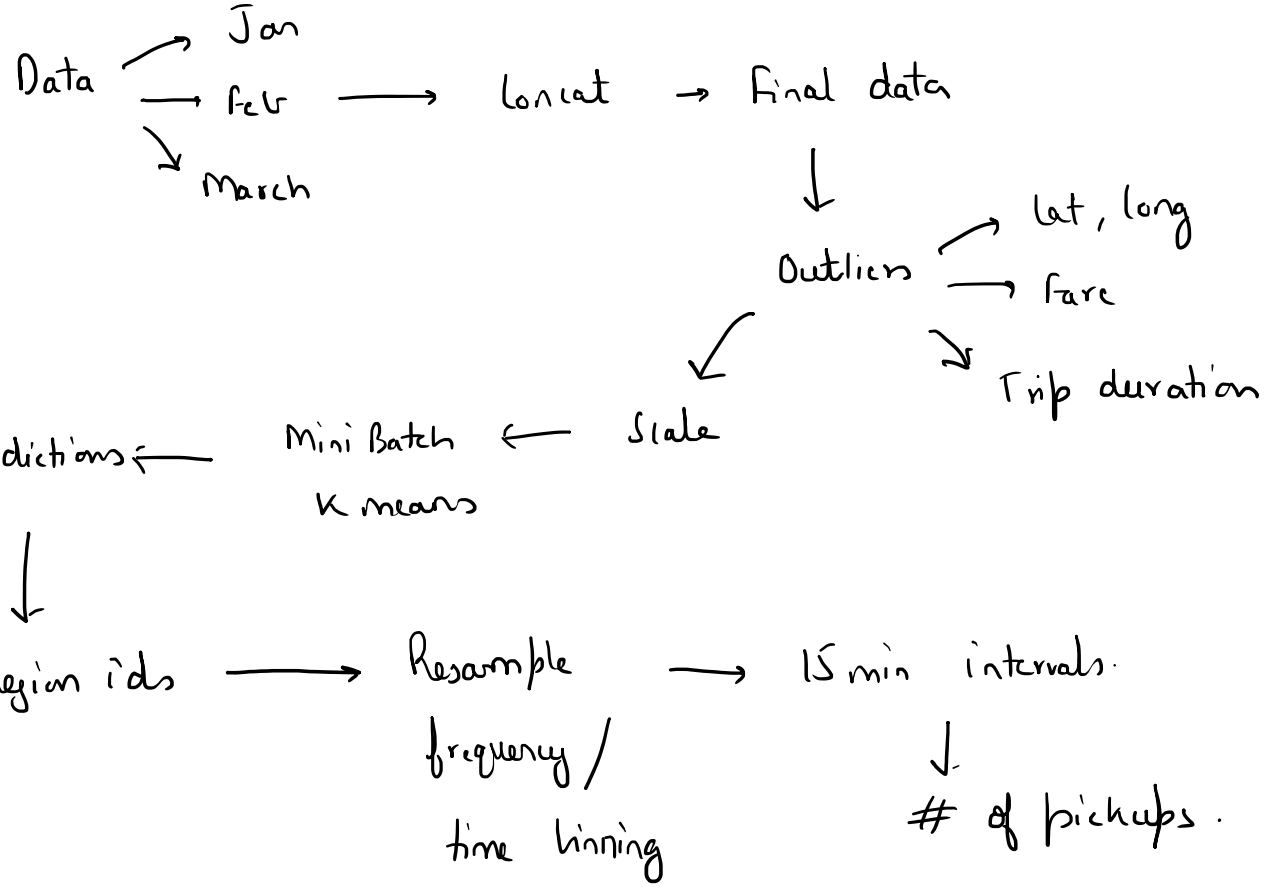
α towards 1 $\longrightarrow$ less smoothing factor

α towards 0 $\longrightarrow$ More smoothing factor

Flow

1) Dividing our data into time intervals → 15 min

2) Smooth out the # of pickups for each interval.



For each region and for each 15 min time interval

# of pickups.

Smooth out the data → MA → window ✓

EWMA → α values ✓