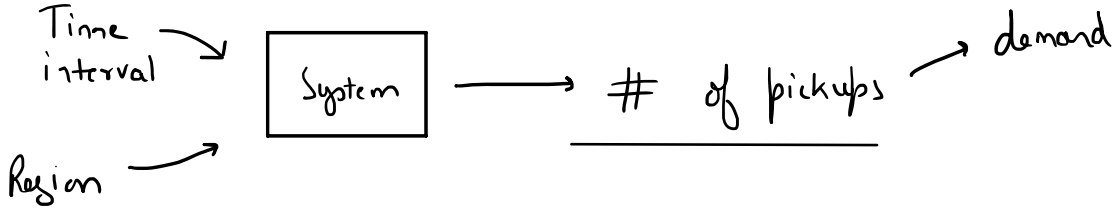
Recap

demand prediction →



NYC Taxi dataset → 2016 → Jan, Feb, March

Jan

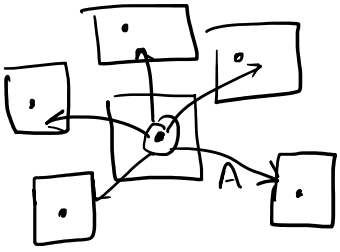
2GB

dask dataframes → Chunking, Operations, task graphs

.compute() ✓

Row → Information about a ride → Distance, Fare, Drop coord, Pickup coord, No. of passenger, Datetime

App → Driver Partner



Migrates to a region where demand ↑

10:00 am → 10:15 AM

revenue ↑

Driver rides ↑

Driver wait times

# of pickups

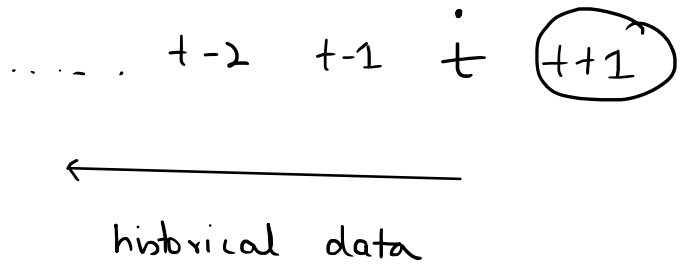
ride → share

EDA 1) Datetime columns ✓ with day / time of day # of pickups

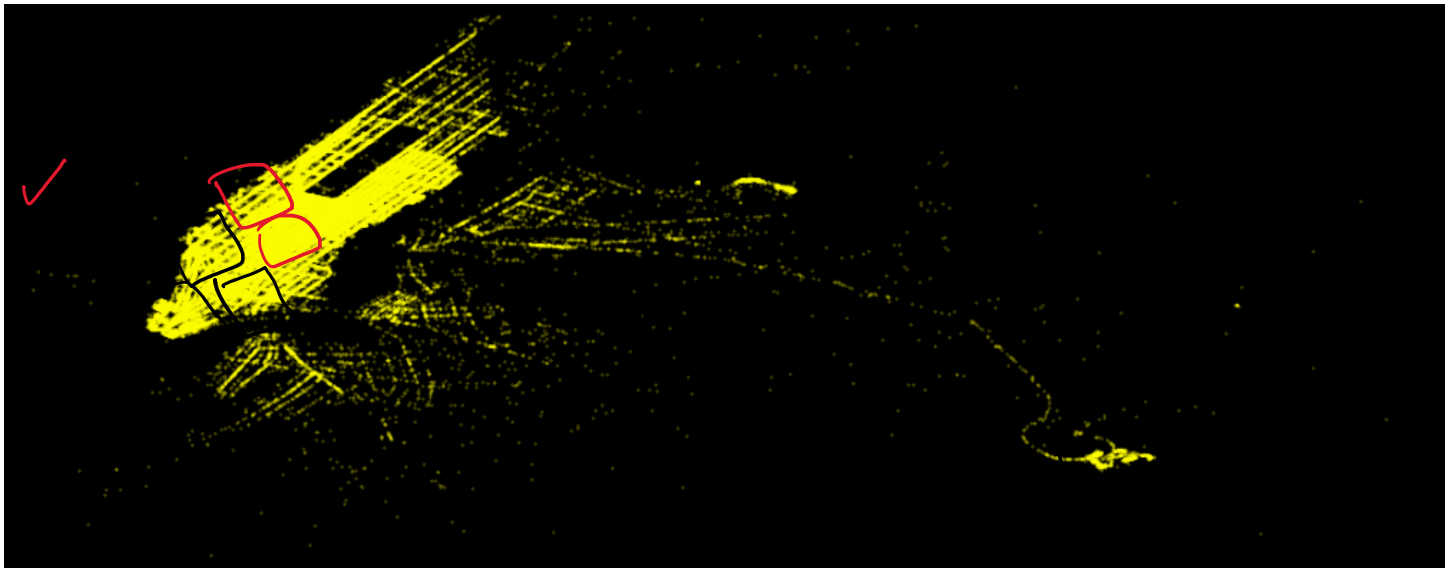
2) Outliers → loord, distance, fare

Task 1 → Break our NYC into regions ✓

Task 2 → historical data for pickups for each region in the map

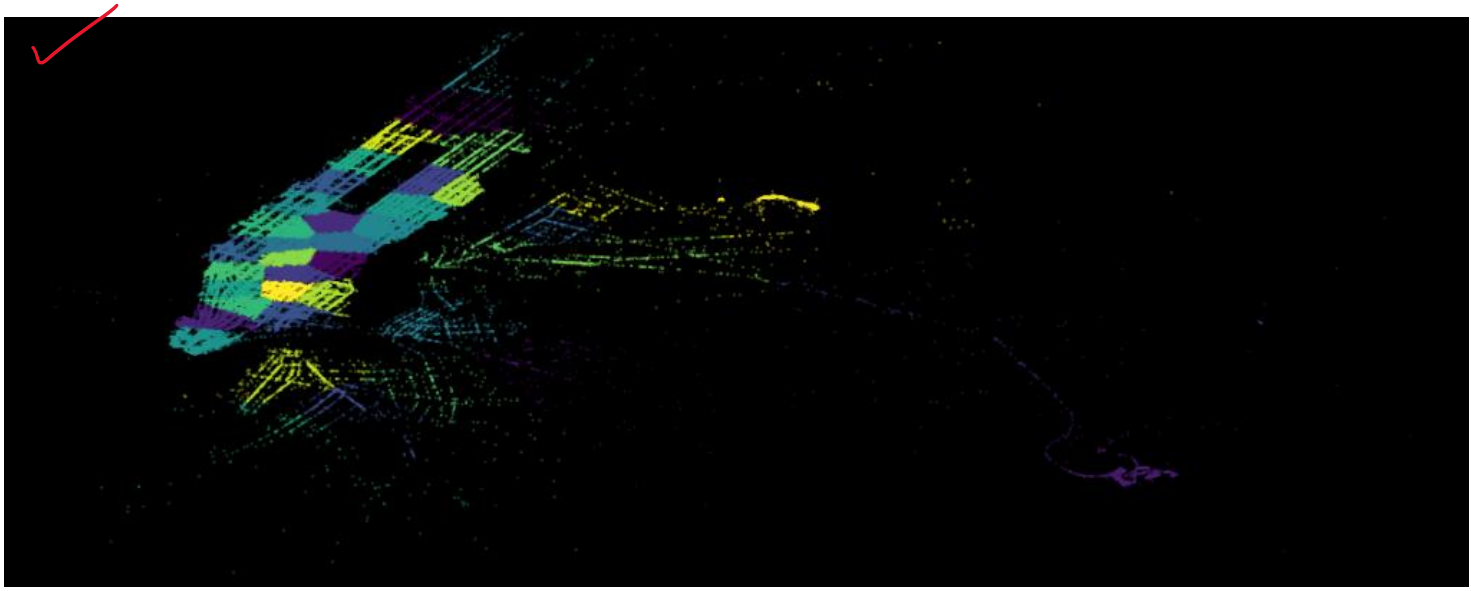


Goal



Scatterplot → Pickups for NYC

Regions → Break Considerations



Considerations

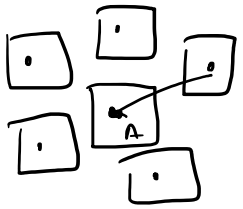
Outliers → coordinates → extreme

↓

Treat outliers.

1mile in NYC → 10 - 15 minutes

travel a lot



→ Proximity ≃ 1mile

Move from current region to neighboring region.

15 minute interval

10:00 am

10:15 am

1mile away

optimum no. of regions

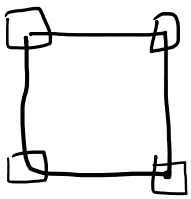
2 Parts → Remove outliers

Jan, Feb, Mar → Dask

Coordinate, fare amount, distance → remove

↓

4 columns



outliers → remove

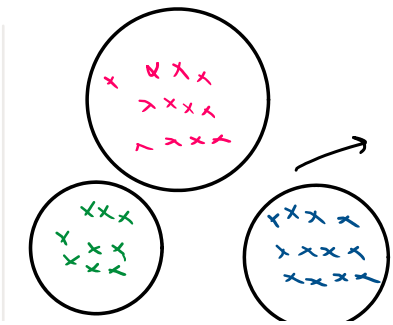
pick up coordinates → pickup lat | pickup long → Regions

Save as pandas dataframe

2)

→ K Means

K = 3



→ Initialize → Centroids

1) Each data point will be assigned a cluster.
2) Move centroid
3) Reassign cluster

K → We have to find the value of K

Dataset ↑ K = 5

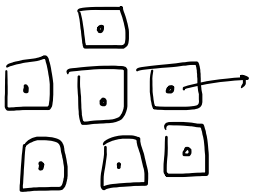
Points = Cores

Mini Batch K Means → Batch of data points

↓

Stochastic speed ↑ accuracy ↓

Centroids → Region Centers



Defines a region

Location data → Pandas, chunksize

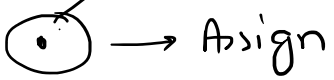
Mini Batch K means → Partial fit

chunk 1 —

chunk 2 —
chunk 3 —

final trained model.

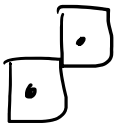
Centroids → Region centers.



1 mile

K = 10, 20, 30, 40. ----

centroids.



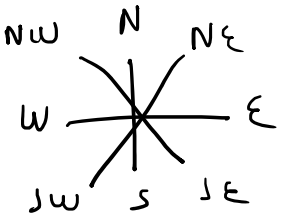
K = 10

(10,10)

	1	2	3	4	-----
1	0	—	—	—	
2	—	—	—	—	
3	—	—	—	—	
4	—	—	—		
:					

Sort → Ascending order

& → Directions



avg → 1 → 1.5 mile

✓ Good Region