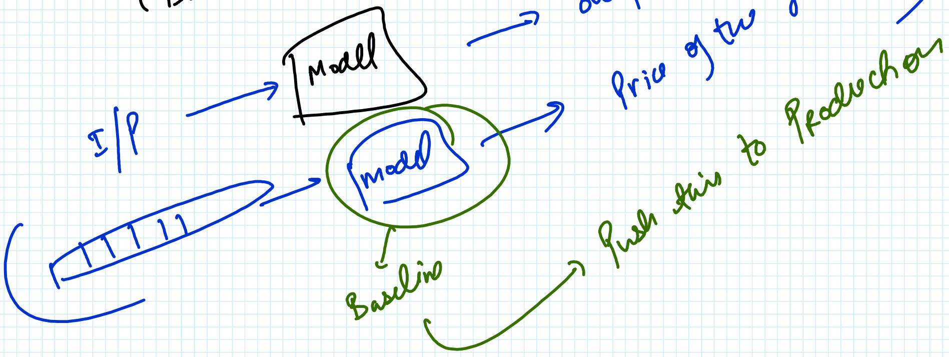
Project

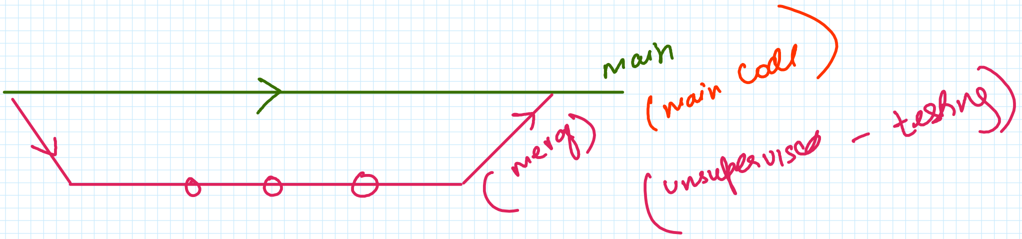
23 December 2024 21:02

Agenda → Project (Supervised ML* Project)

(House Price Prediction)

(Baseline ML model)





(Can we try something Better?)

→(Unsupervised)

(Problem Statement)

Given a set of features, predict the prices of the houses.

performance of the supervised model better by using some ML-2 ideas.

(Part - 1)

ML - 1 *

focus (Unsupervised)

ML → Supervised

ML-2 → Unsup.

Baseline (ML - 1) → Industry things (not see anywhere)

→ Categorical

→ Scaling

(studied in past)

Baseline model → Preprocessing → Scaling

Pipeline

Data → handle missing values → handle categorical column → Preprocessing (Function) → Scaling → Fit the model ←

Handle Categorical → one hot encoding
→ ordinal
→ Frequency encoding
→ Target encoding (multiple column) → Data Leakage

(300 columns)

(10 – 20 columns)

zip code	
22	10
22	10
23	15
23	15
'	'
'	'
23	15
38	
'	

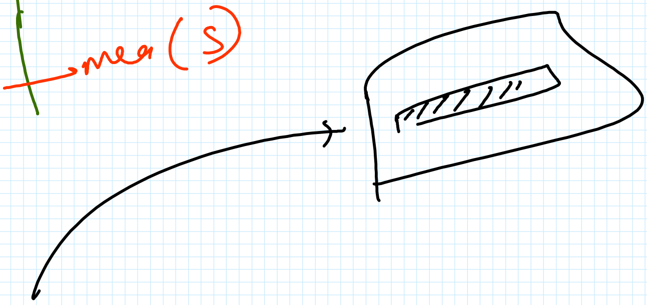
15

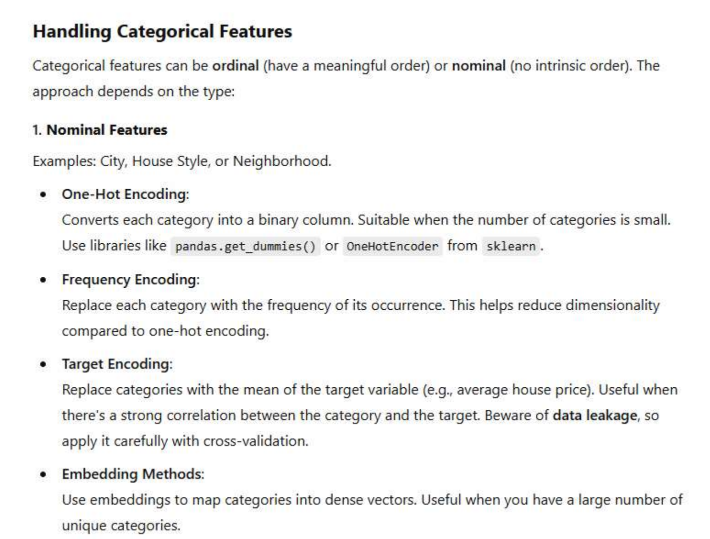
Garage	Price
S	{ } → mean (S)
S	
S	
S	
S	
M	{ } → mean (M)
M	
M	
L	{ } → mea (S)
L	

Advance (NLP)

Embedding method

(D.L)





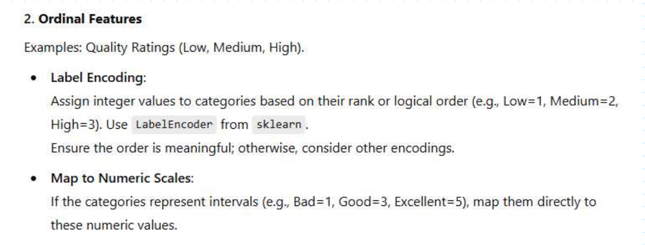
Handling Categorical Features

Categorical features can be **ordinal** (have a meaningful order) or **nominal** (no intrinsic order). The approach depends on the type:

1. Nominal Features

Examples: City, House Style, or Neighborhood.

- **One-Hot Encoding**:
  Converts each category into a binary column. Suitable when the number of categories is small. Use libraries like `pandas.get_dummies()` or `OneHotEncoder` from `sklearn`.
- **Frequency Encoding**:
  Replace each category with the frequency of its occurrence. This helps reduce dimensionality compared to one-hot encoding.
- **Target Encoding**:
  Replace categories with the mean of the target variable (e.g., average house price). Useful when there's a strong correlation between the category and the target. Beware of **data leakage**, so apply it carefully with cross-validation.
- **Embedding Methods**:
  Use embeddings to map categories into dense vectors. Useful when you have a large number of unique categories.



2. Ordinal Features

Examples: Quality Ratings (Low, Medium, High).

- **Label Encoding**:
  Assign integer values to categories based on their rank or logical order (e.g., Low=1, Medium=2, High=3). Use `LabelEncoder` from `sklearn`.
  Ensure the order is meaningful; otherwise, consider other encodings.
- **Map to Numeric Scales**:
  If the categories represent intervals (e.g., Bad=1, Good=3, Excellent=5), map them directly to these numeric values.

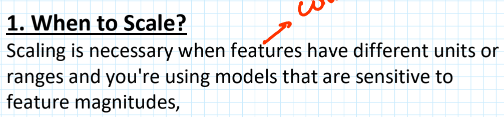
(we want reproducability) → 78% (testing) (Monday)

↓

75%

Scaling

Necessary



1. When to Scale?

Scaling is necessary when features have different units or ranges and you're using models that are sensitive to feature magnitudes,

columns

Distance Based (KNN, clustering, SVM)

Gradient Based (Log Reg, NN)

Regulars" (Lasso, Ridge)

Most Common

{Standard Scaling}

$$\frac{x - \mu}{\sigma}$$

(tries to maintain distribⁿ)

#(gauss/Normal)

→ handles outliers

median

[Min - max Scaling]

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

[0, 1]

[0, 1]

→ Neural Networks

→ when you need all feature in uniform scale

→ handles

feature in scale

Robust Scaling →

⇒ $\boxed{\dfrac{X - \text{median}}{IQR}}$

log transform"

Baseline Model

(H.P.P)

Ideally → (L.R) → (Poly R) → Regular S" [Ideal]

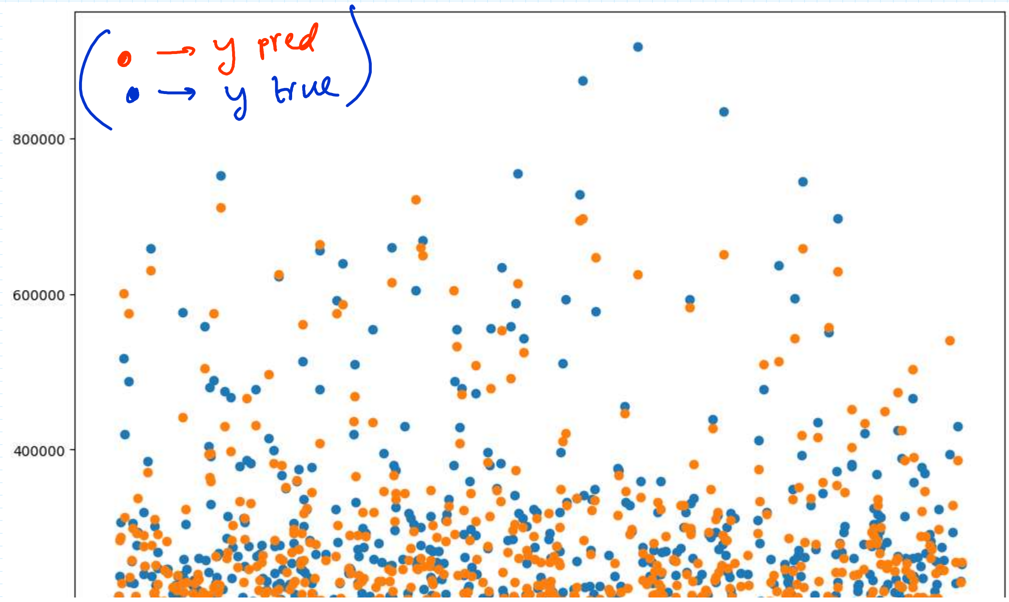
Tree → D.Tree → Random Forest → Boosting

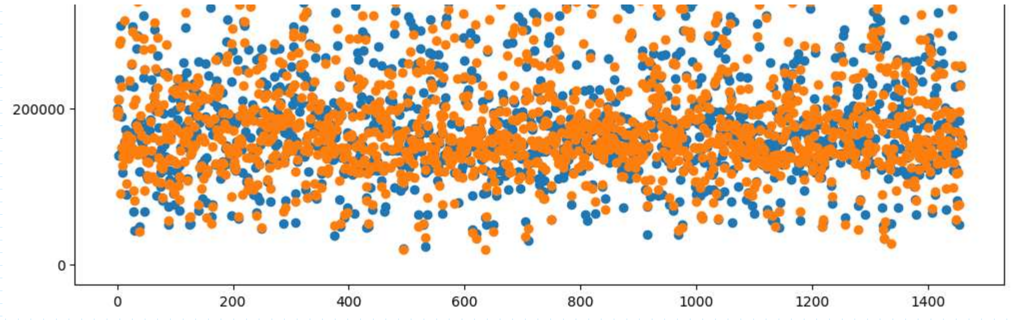
(feature importances)

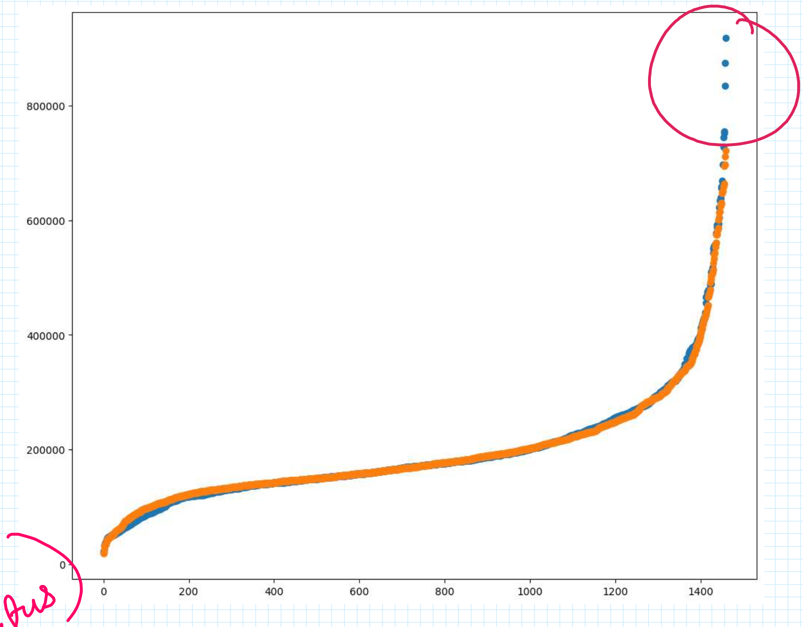
→ (K Fold Cross Valid")

Train mse Test mse

[Apply ML - 2 techniques)







→ y true is high
But y pred is low

(current status)

Baseline Model → good preds for cheap houses
but error for expensive houses is high

(solution?)

→ No more Tr data for expensive houses

→ (Outlier Removal)

(10:09 pm)

ML-2 (Use ML-2 techniques & try to improve the RMSE)

* (There is no guarantee)

What techniques do you know?

columns)

*

(DR techniques)

PCA

Outlier Detection

Clustering

PCA → * (Reduces dimension)

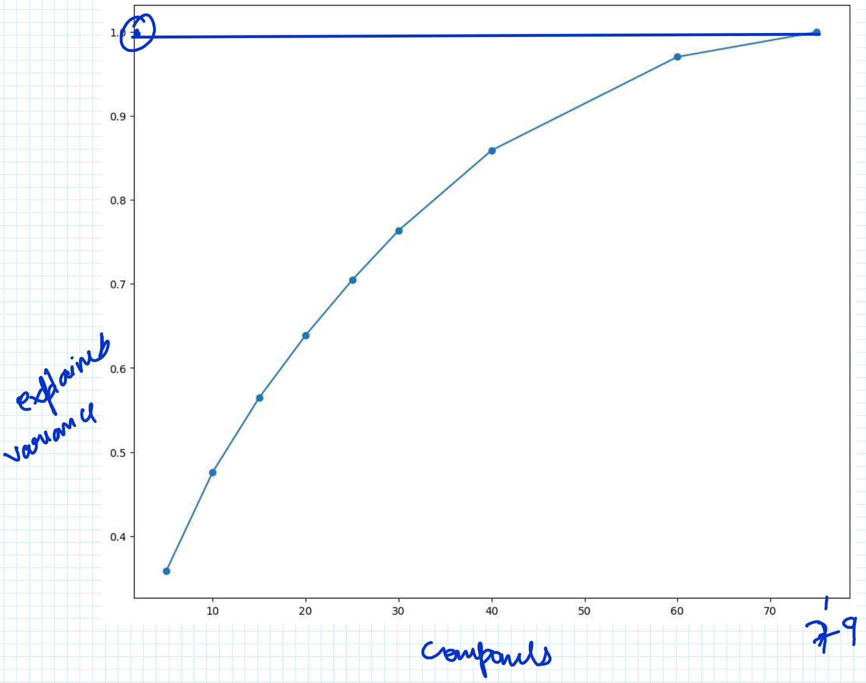
Reduce dim

error (Bug ↑)

(79) → 31,000 $

↓

35 → 33,000 $



(Outlier Detection)

→ Z score

→ IQR (Box Plot)

(Outlier → IQR (Box

# Visualisation → PCA (n-comp = 2)

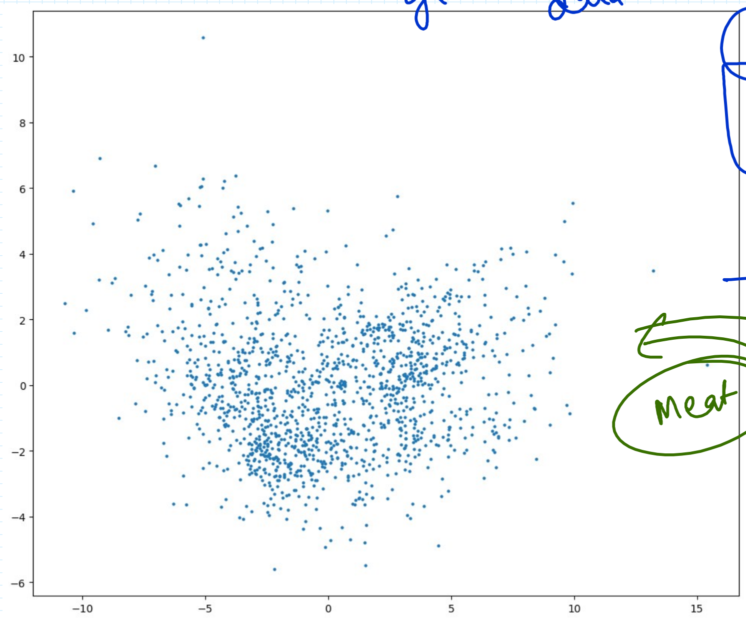
→ t-SNE (checked visually)

→ UMAP

# → (Read 2 research paper) → LOF

→ Isolⁿ forest

global str of data



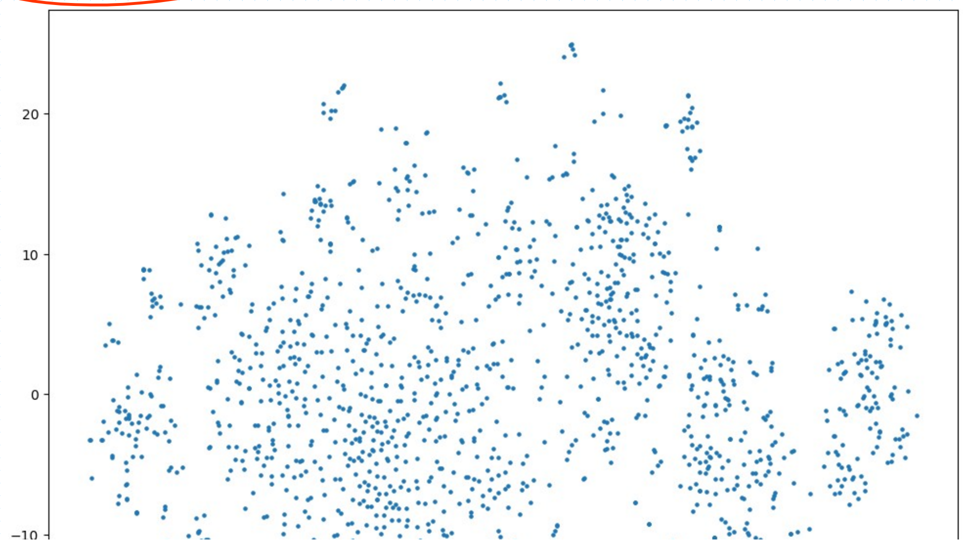
Industry

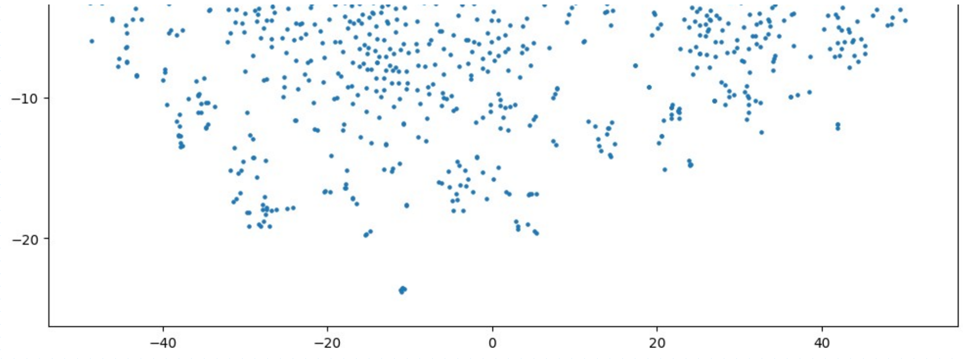
79 → 2 → 2 dim

→ How much variance

Meat [30%]

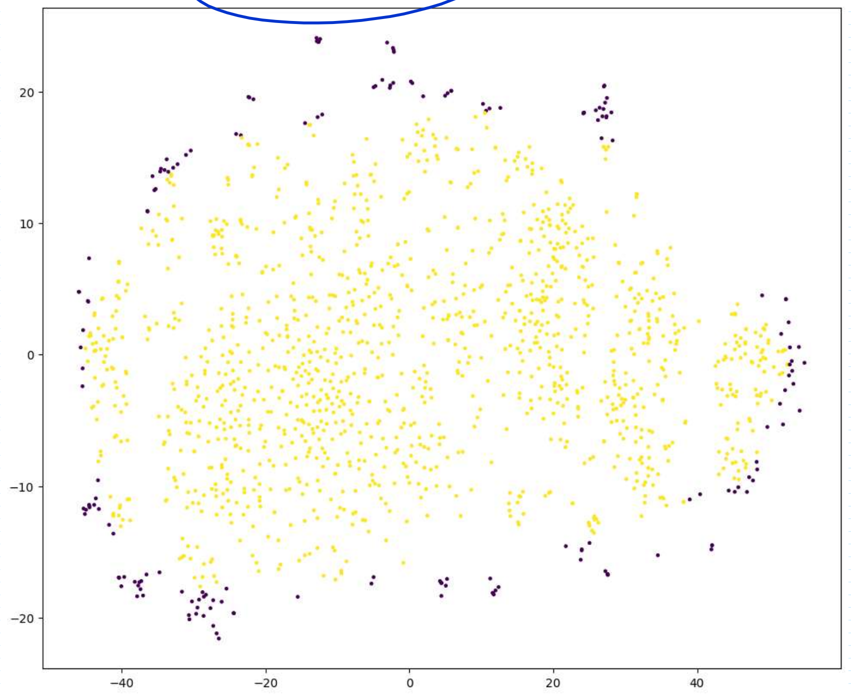
t-SNE



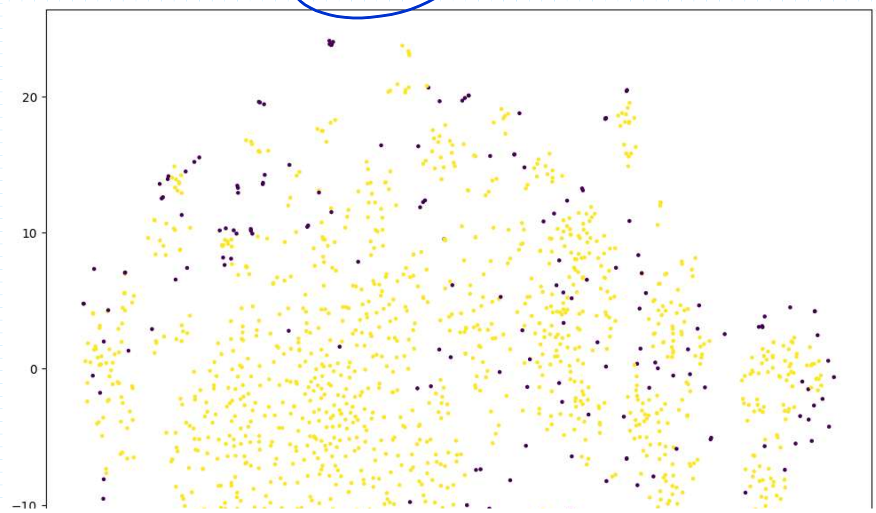


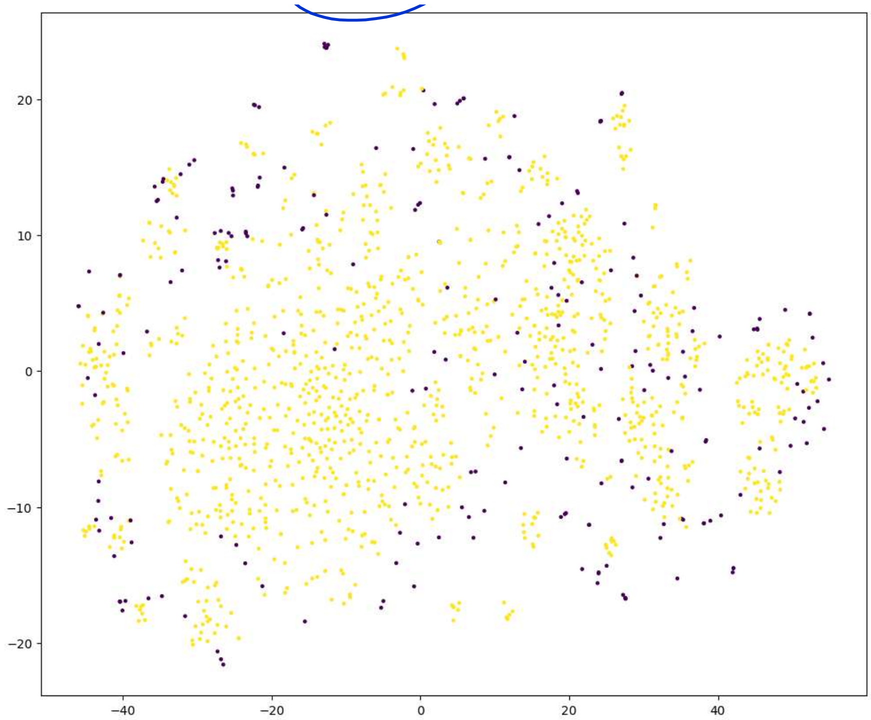
LOF / IForest *

iForest



LOF





Clustering → Next Session

↳ (EDA on clustering → some more steps)

* Come prepared *

* (Not good)

(Dunn Index) & [(Silhoutte Score)]

(global)

↳ one Index

(data point level)

20 data pts

↳ S. scores

Dunn Index

The Dunn Index evaluates the compactness and separation of clusters

$$D = \frac{\min(\text{Inter - cluster})}{\max(\text{Intra cluster distance})}$$

→ Compact Cluster

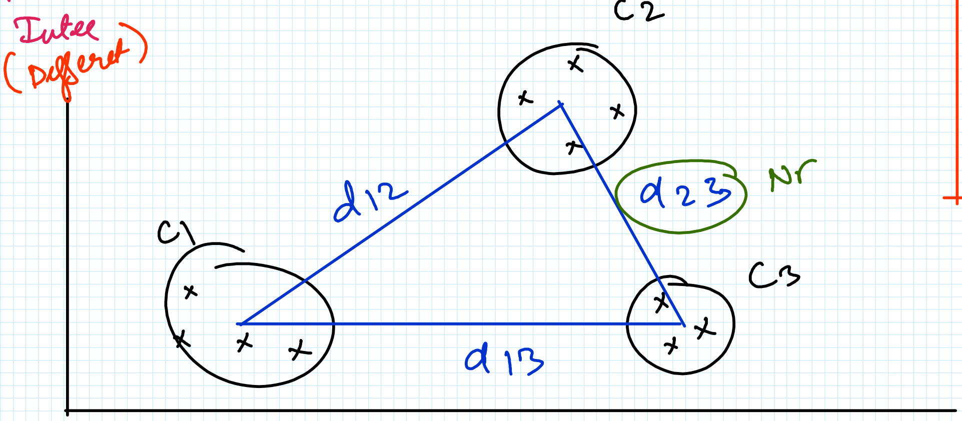
→ well separated clusters

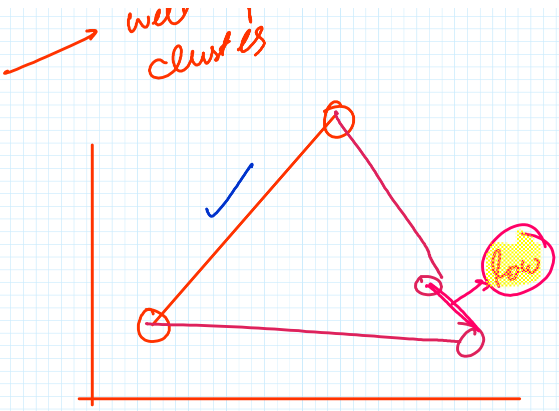
(> 1)

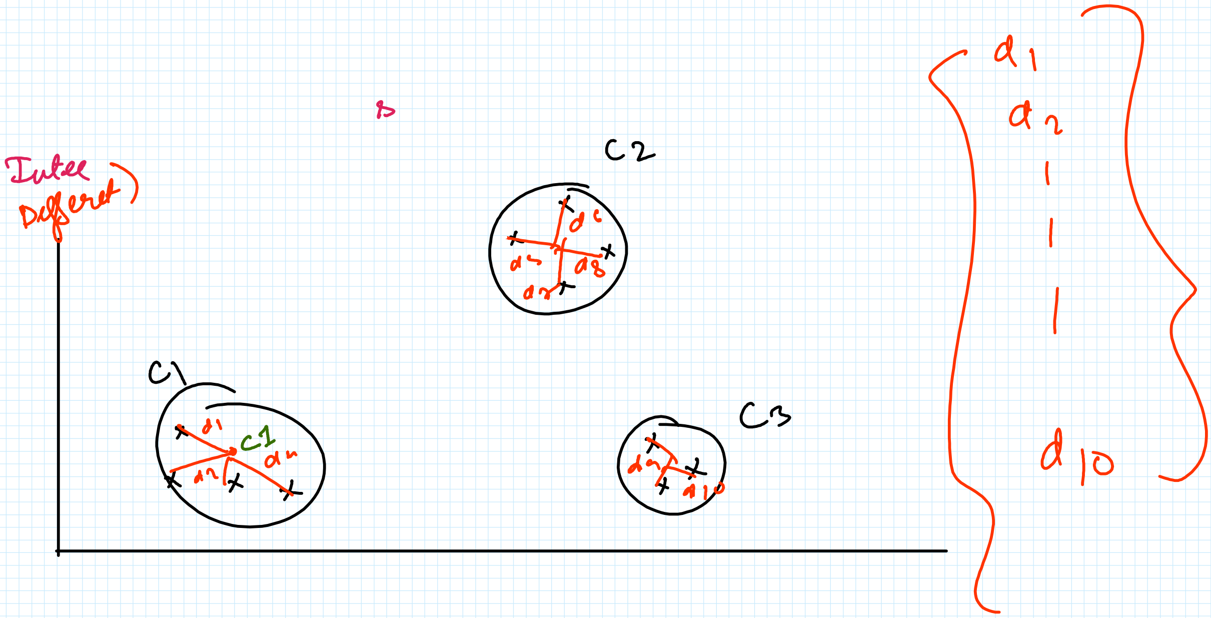
max (Intra cluster distance)

(> 1
> 2
(good)

Intra
(same)



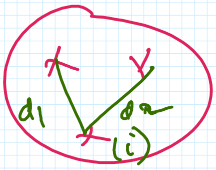




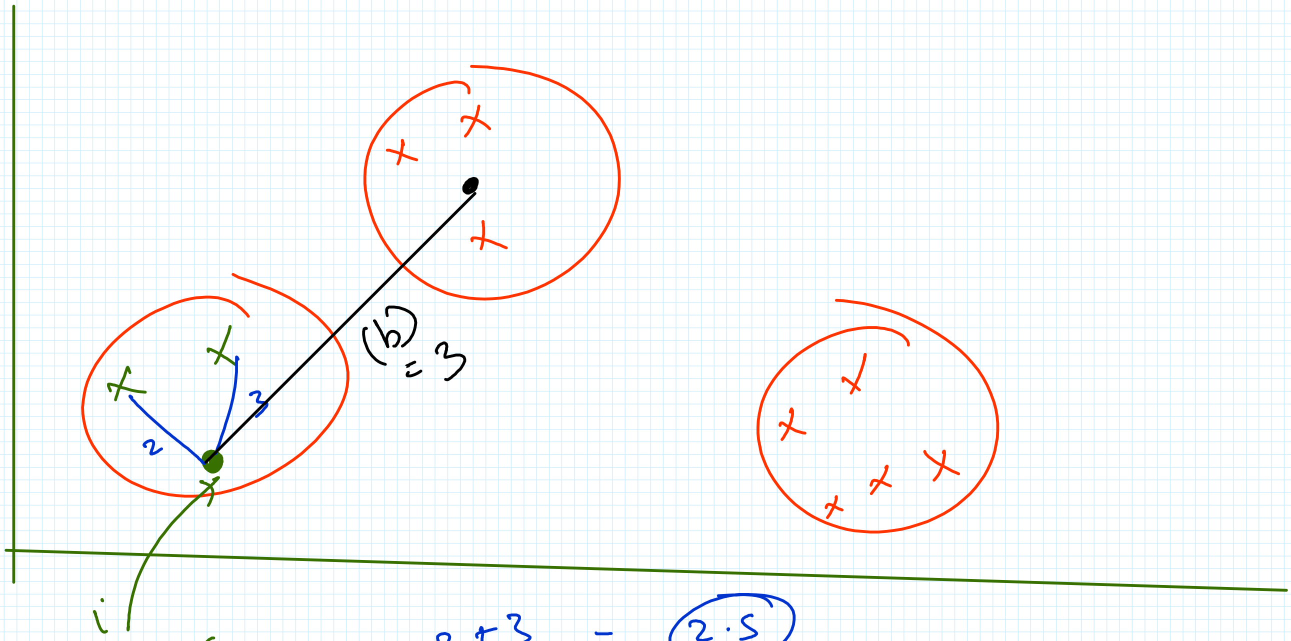
Selhoutt Score (-1, 1)

$$S(i) = \frac{b(i) - a(i)}{\max(b, a)}$$

b = inter cluster distance of that pt i to nearest clusters a = Intra cluster distance



$$a = \frac{d_1 + d_2}{2}$$

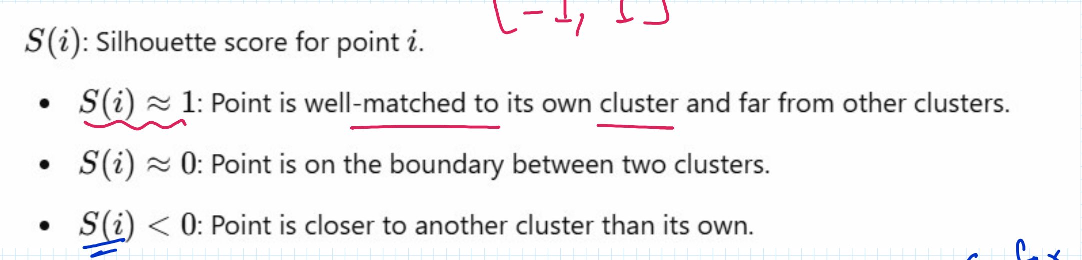


$(a) \rightarrow \frac{2+3}{2} = \boxed{2.5}$

$(b) \rightarrow 3$

$$S.S(i) = \frac{b-a}{\max(a,b)} = \frac{3-2.5}{3} = \frac{0.5}{3}$$

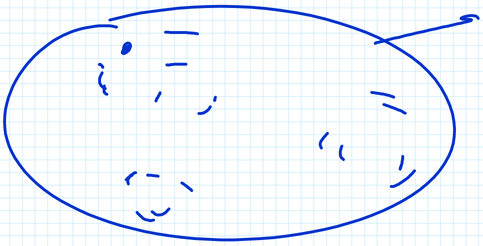
[-1, 1]

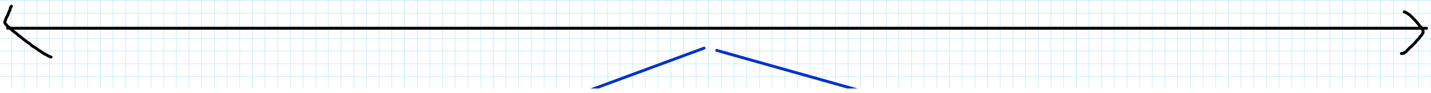


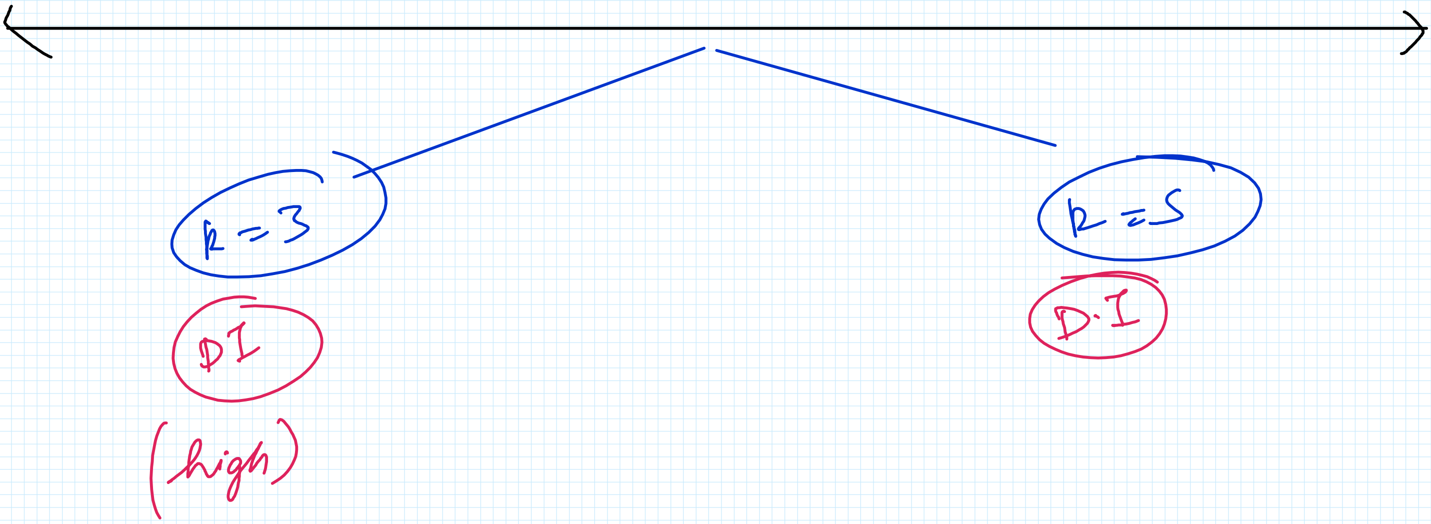
$S(i)$: Silhouette score for point i.

- $S(i) \approx 1$: Point is well-matched to its own cluster and far from other clusters.
- $S(i) \approx 0$: Point is on the boundary between two clusters.
- $S(i) < 0$: Point is closer to another cluster than its own.

final S.S = avg(all S.S for 'i' data pts)






k = 3
DI
(high)
k = 5
D.I