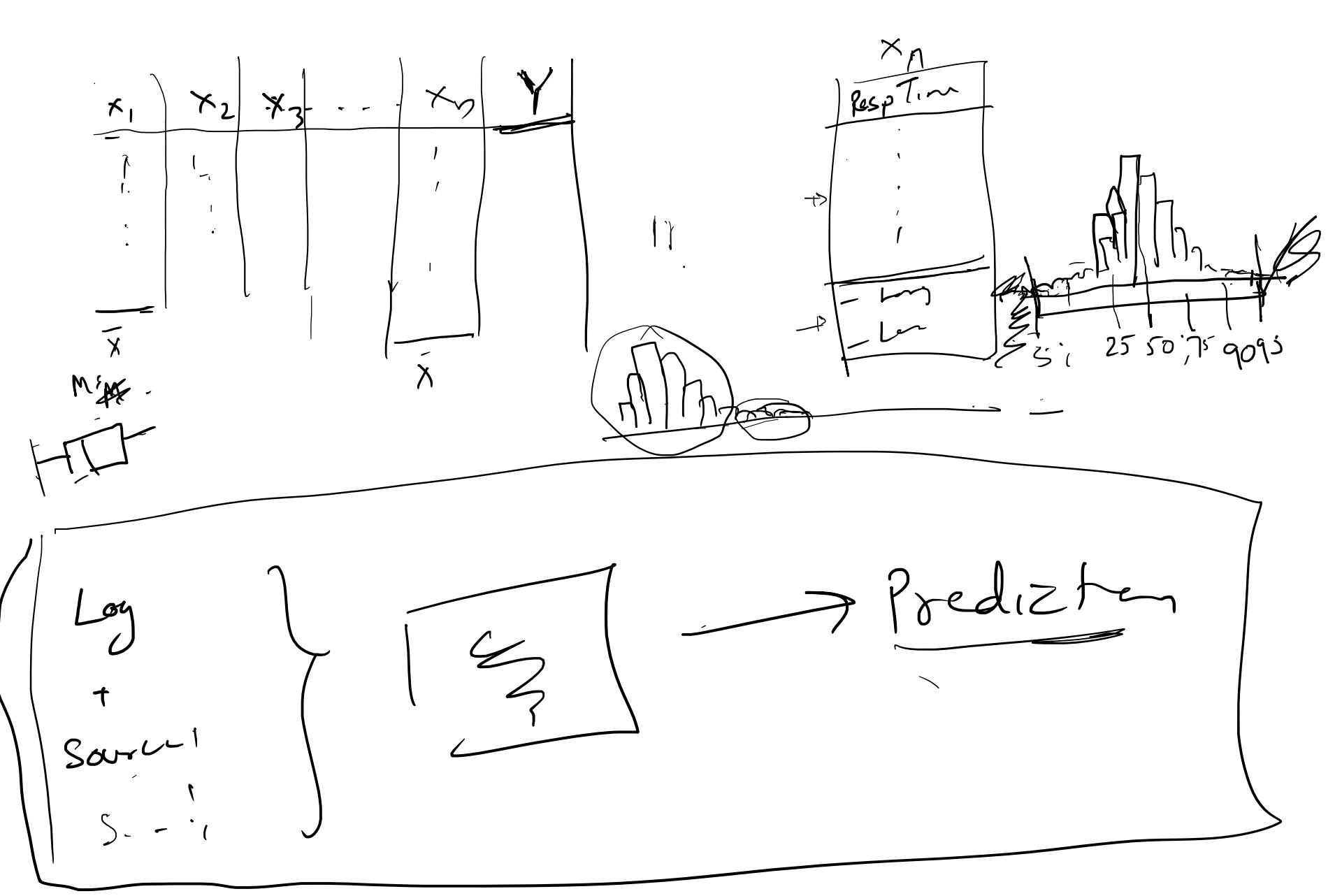
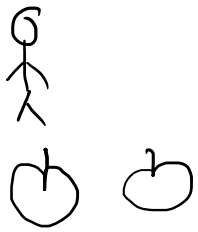
Machine Learning

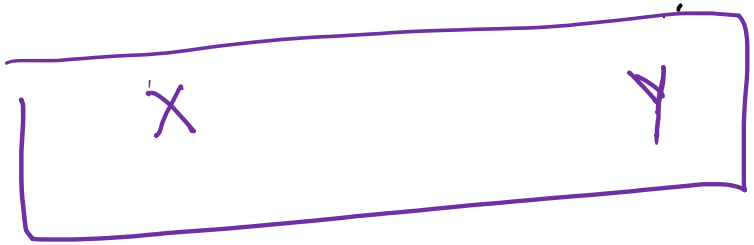
K Nearest Neighbors


Resp Time
25 50 75 90 95
Log
+
Source
Prediction

TRAINING

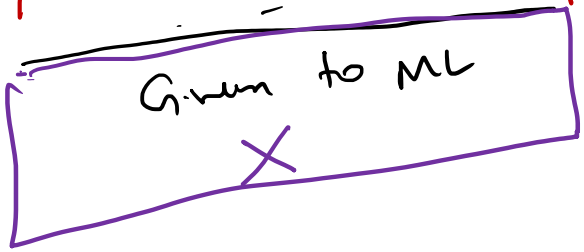


X_1 Col	X_2 Size	X_3 Shape	X_4 ... Texture	Y
–	–	–	–	Apple
–	–	–	–	Apl
–	–			

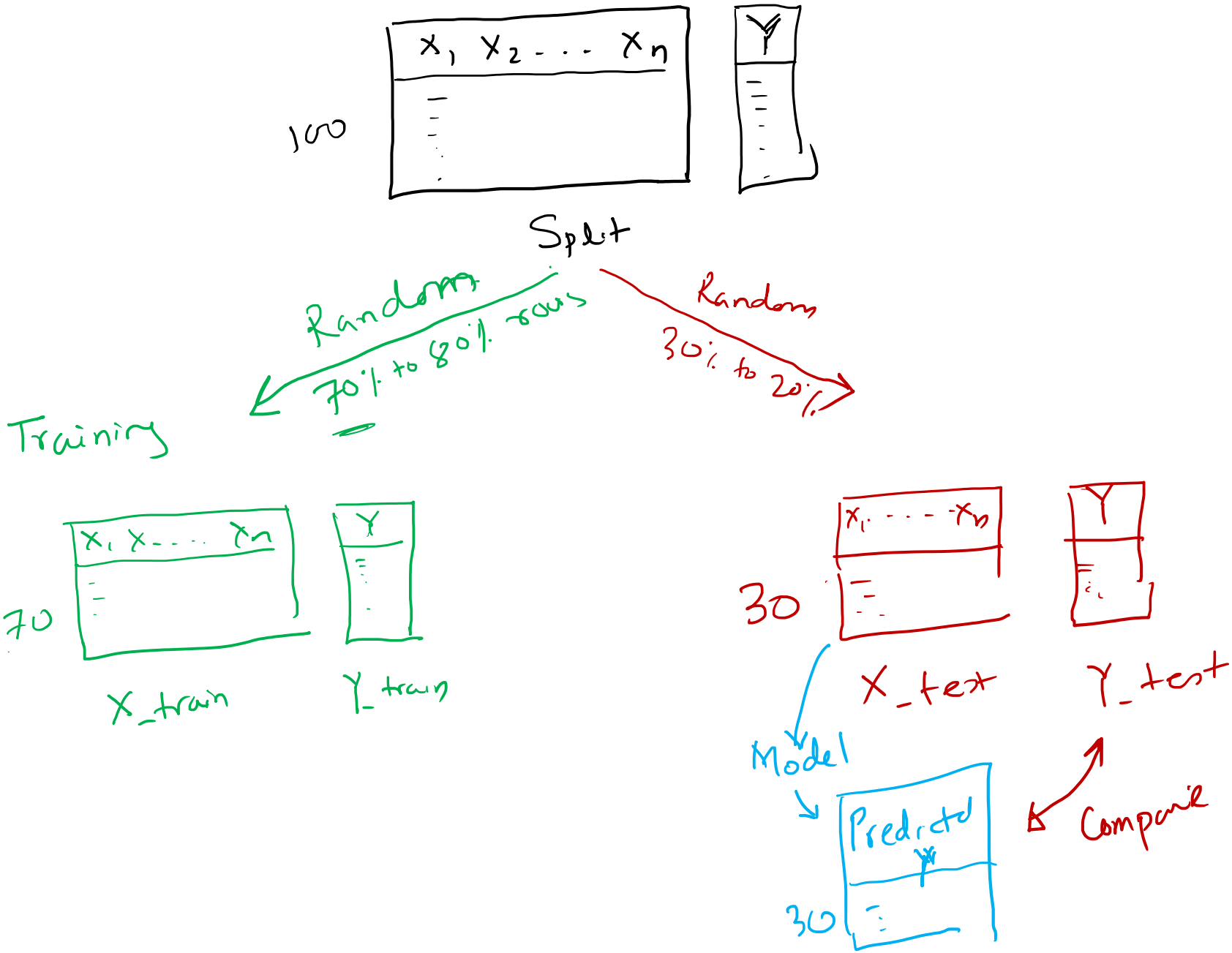


Testing

X_1 Col	X_2 Size	X Shy	X_4 ...	Pred Y	Actual Y
–	–	–		Appl–	Appl
–	–	–		Apple	Appl
–	–			No Ap	N.A.
–	–				



Not given to ML.


100
x_1 x_2 ... x_n
Y
Split
Random
70% to 80% rows
Random
30% to 20%
Training
70
x_1 x_2 ... x_n
Y
X_train
Y_train
30
x_1 ... x_n
Y
X_test
Y_test
Model
Predicted
30
Compare

K Nearest Neighbors

- A supervised learning method
- Though it is a supervised learning method, it is a ‘lazy learner’, i.e. does not construct a model using training data
- Classification is determined based on a majority vote of the nearest neighbors of each point
- Suitable for classification where items in a class tend to be fairly homogenous on the values of attributes
- Not suitable if the data is too noisy or the target classes do not have clear demarcation in terms of attribute values

K Nearest Neighbors

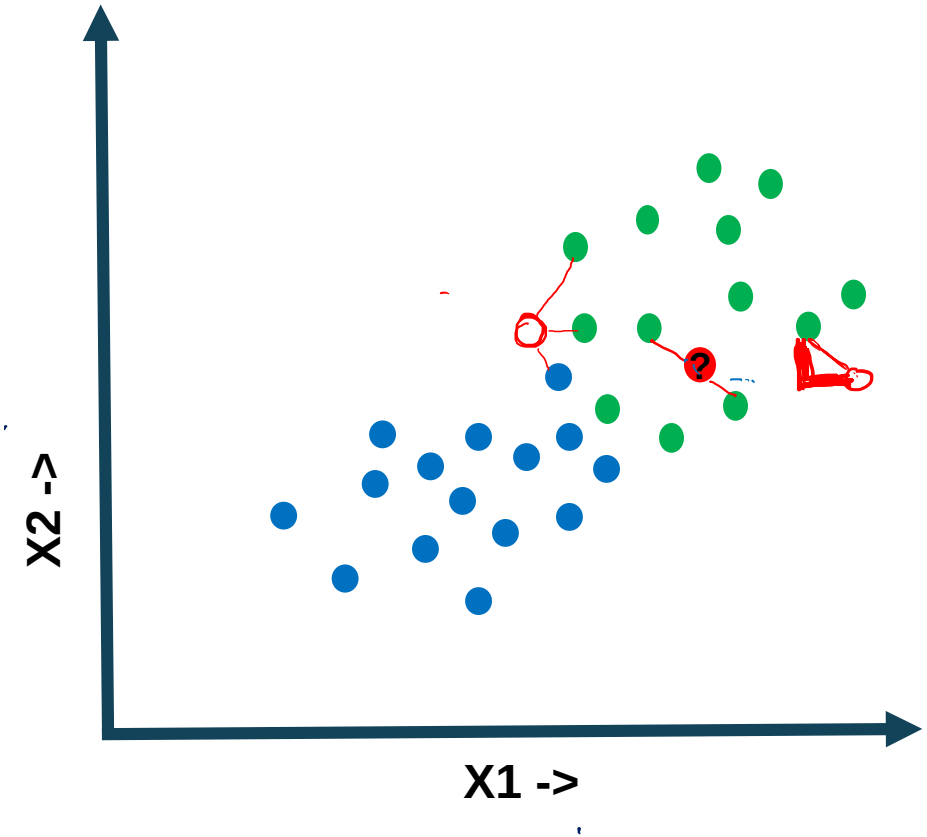
- Nonparametric model: distribution-free tests because no assumption of the data needing to follow a specific distribution
- Commonly used for classification
- Can also be used for regression

K Nearest Neighbors

3-Class

K=5

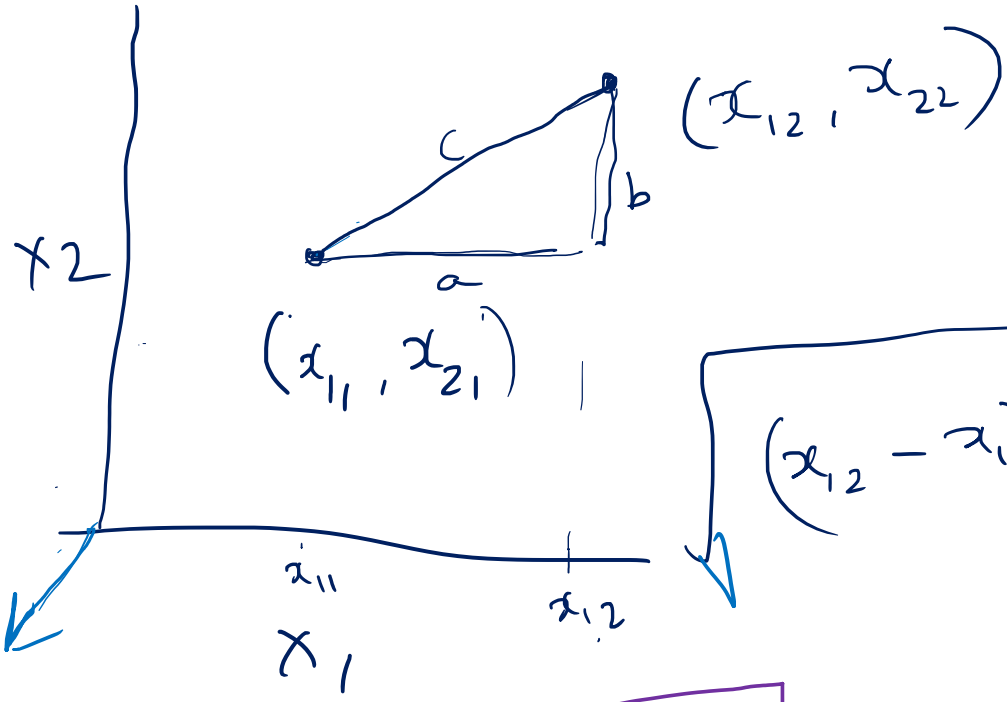
- New data point is assigned a class which has the most data points in the nearest neighbors of the point



Similar

More Similar = Smaller dist. between them

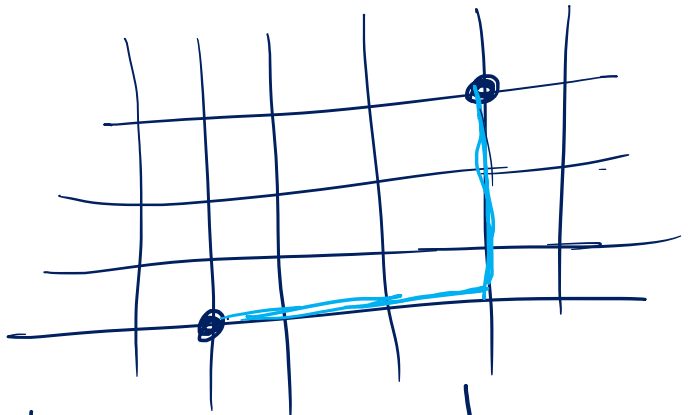
X_1	X_2	X_3	...
C=	C=	C=	



$$\sqrt{(x_{12}-x_{11})^2+(x_{21}-x_{22})^2+(x_{31}-x_{32})^2+(x_{41}-x_{42})^2}$$

Minkowiski 2

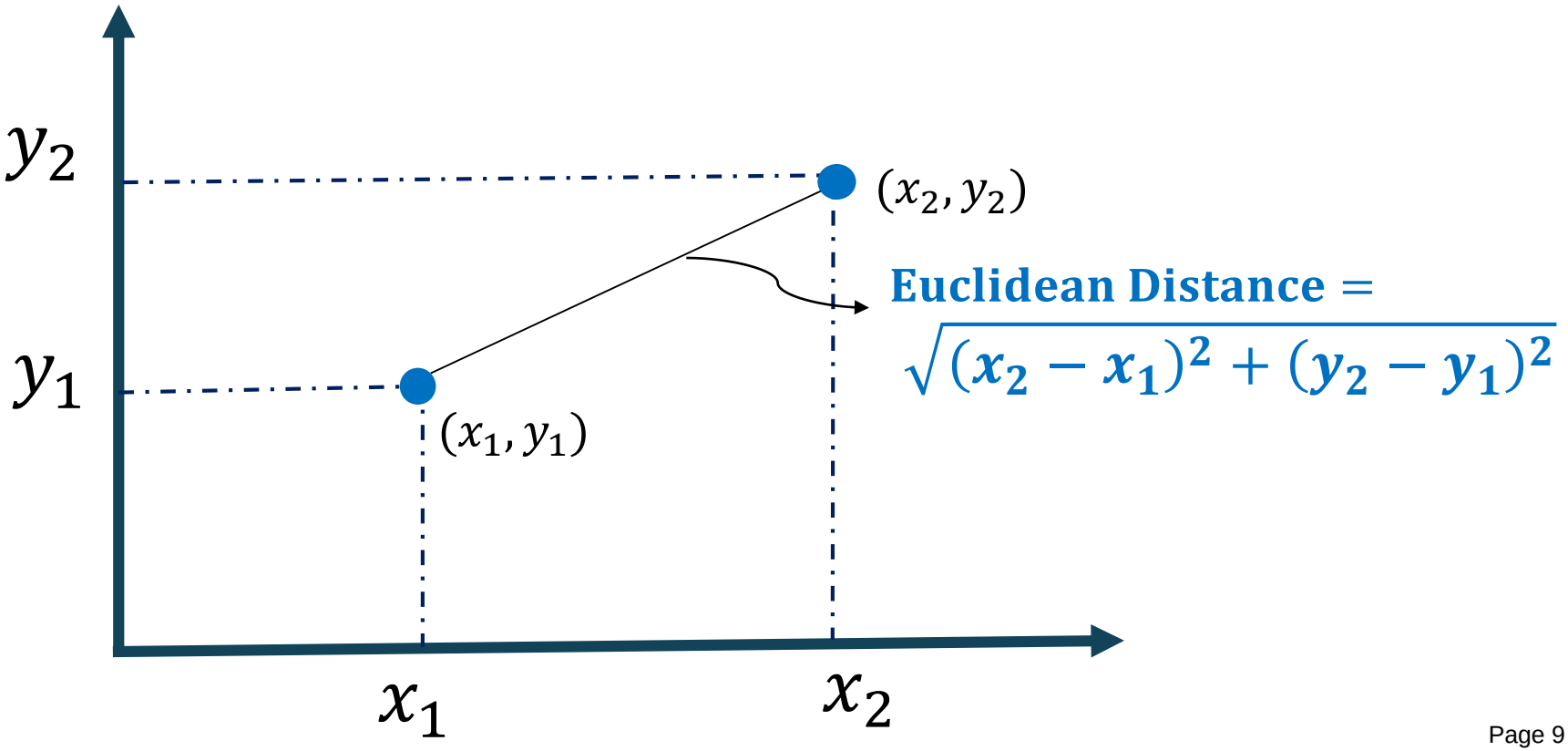
$$\left(|x_{12}-x_{11}|^p+|x_{21}-x_{22}|^p\right)^{1/p}$$



$$|x_{12}-x_{11}|+|x_{21}-x_{22}|$$

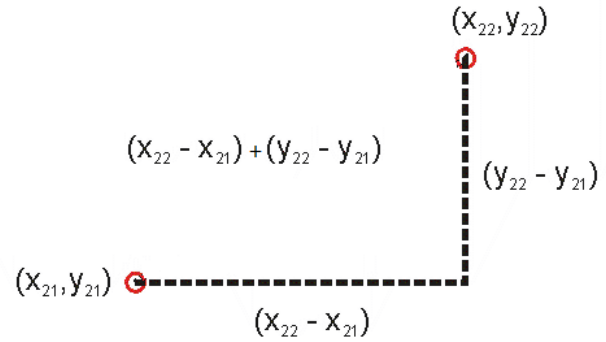
K Nearest Neighbors

- Nearest neighbors are calculated based on shortest distance.
- Most commonly used distance measure is Euclidean distance (default)

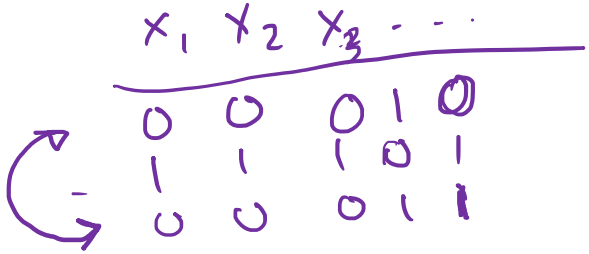


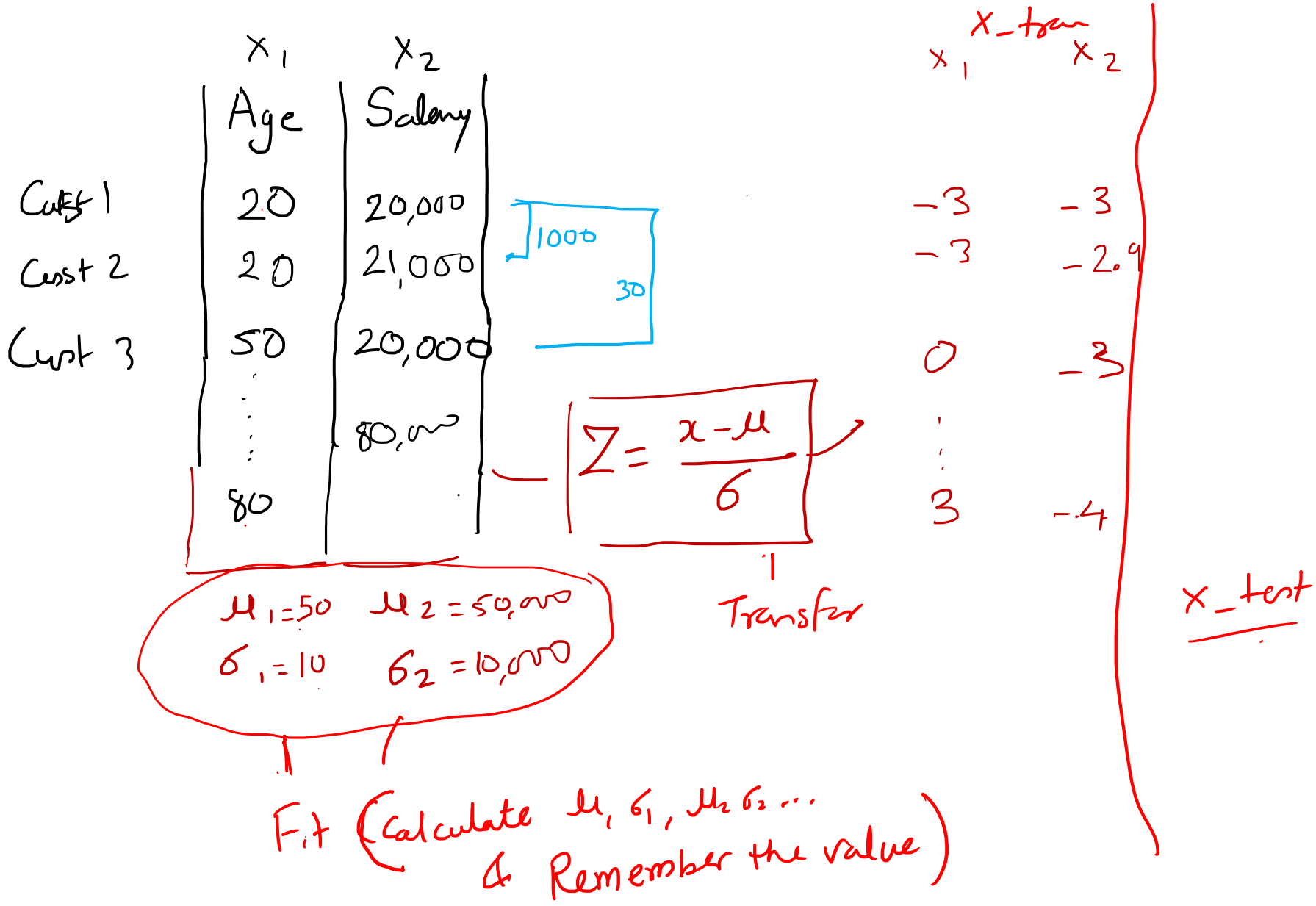
K Nearest Neighbors

- Manhattan / Taxi Distance: Also called L1 norm. It is the sum of the differences in each dimension.



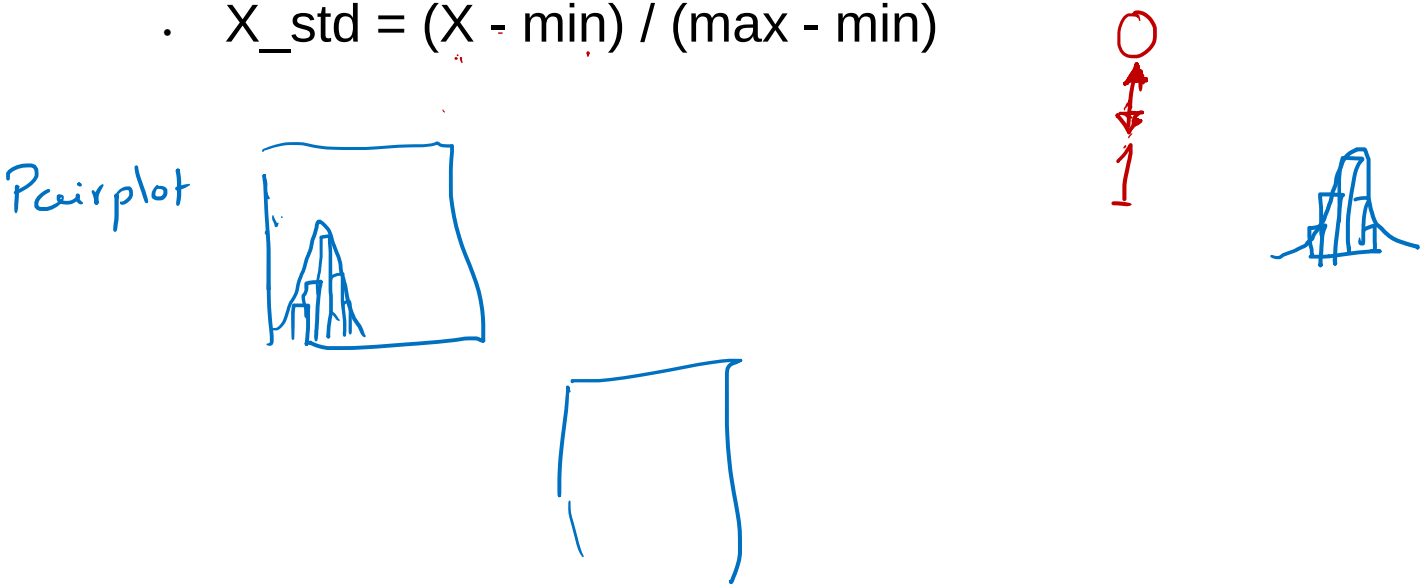
- Mahalanobis distance – takes into account the covariance between attributes
- Jaccard distance (for Boolean data) = number of non-equal dimensions / number of nonzero dimensions.
- Matching distance (for Boolean data) = number of non-equal dimensions / number of dimensions




x_1 x_2
Age Salary
Cust 1 20 20,000
Cust 2 20 21,000
Cust 3 50 20,000
80
1000
30
$Z = \frac{x-\mu}{\sigma}$
Transfer
$\mu_1 = 50$ $\mu_2 = 50,000$
$\sigma_1 = 10$ $\sigma_2 = 10,000$
Fit (Calculate $\mu_1, \sigma_1, \mu_2 \sigma_2 \ldots$ & Remember the value)
x_train
x_1 x_2
-3 -3
-3 -2.9
0 -3
3 -4
x_test

K Nearest Neighbors

- To ensure all the dimensions have similar scale, it is necessary to normalize the data. Common way to normalize data is
  - Z-score standardization using formula $z_i = \frac{x_i - \bar{x}}{s}$
  - Min-max scaler
    - X_std = (X - min) / (max - min)

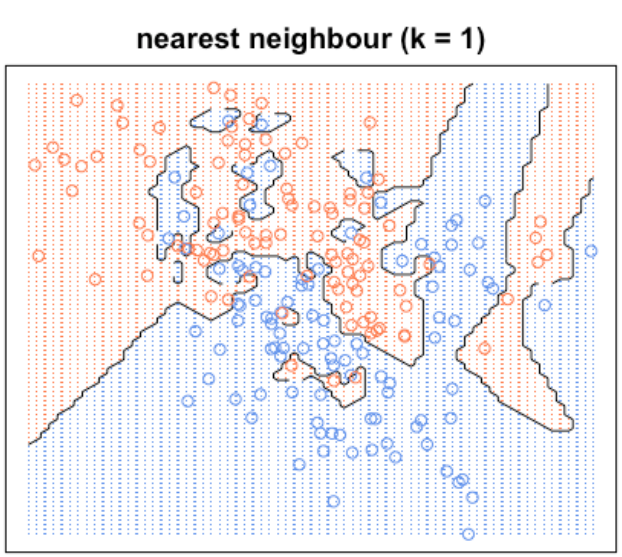


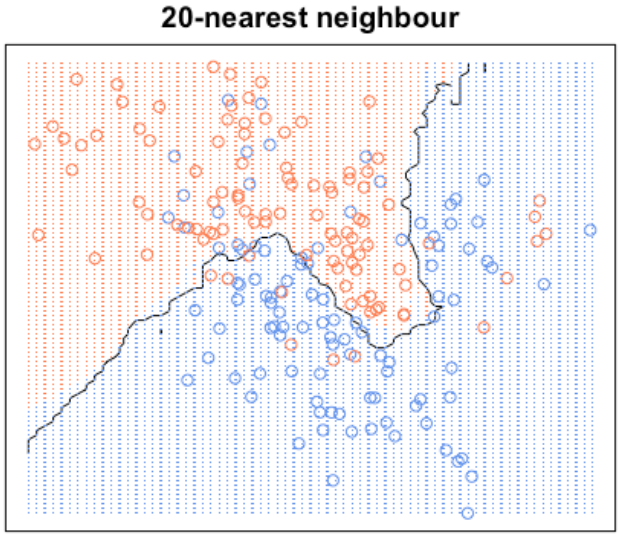
K Nearest Neighbors

- Advantages -
  - Makes no assumptions about distributions of data
  - Easy to understand and implement
  - Not impacted by outliers
  - Able to segregate classes using non-linear boundary

- Dis-advantages -
  - Determining the optimal value of K is a challenge
  - Not effective when the class distributions have large overlap
  - Does not output any models. Calculates distances for every new point. Hence very computation intensive

Selecting value of K





- K is a hyperparameter must be picked in order to get the best possible fit for the data set
- K controls the shape of the decision boundary we talked about earlier
- A small value of k means that noise will have a higher influence on the result

K Nearest Neighbors

- Hands-on exercise

Confusion Matrix

		Predicted			
		A	B	C	
Actual	A	15	0	0	1
	B	0	19	2	19/21
	C	0	0	17	1
		1	1	17/19	

- Classification accuracy = correct predictions / total predictions
- Precision is the proportion of the predicted positive cases that were correct.
  - Precision of C = 17 / (17+2)
- Recall is the proportion of positive cases that were correctly identified
  - Recall for B = 19 / (19+2)
- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

$$\frac{1}{\frac{1}{2}\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}\right)}$$

Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \qquad \textbf{Precision} = \frac{TP}{TP + FP}$$

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

- Note that in binary classification, recall of the positive class is also known as “sensitivity”; recall of the negative class is “specificity”.
- High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.
- Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)