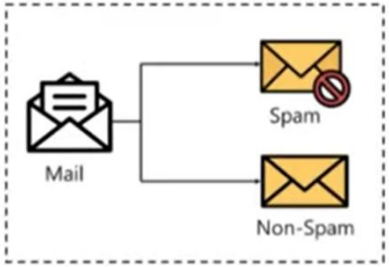
Classifications in Machine Learning

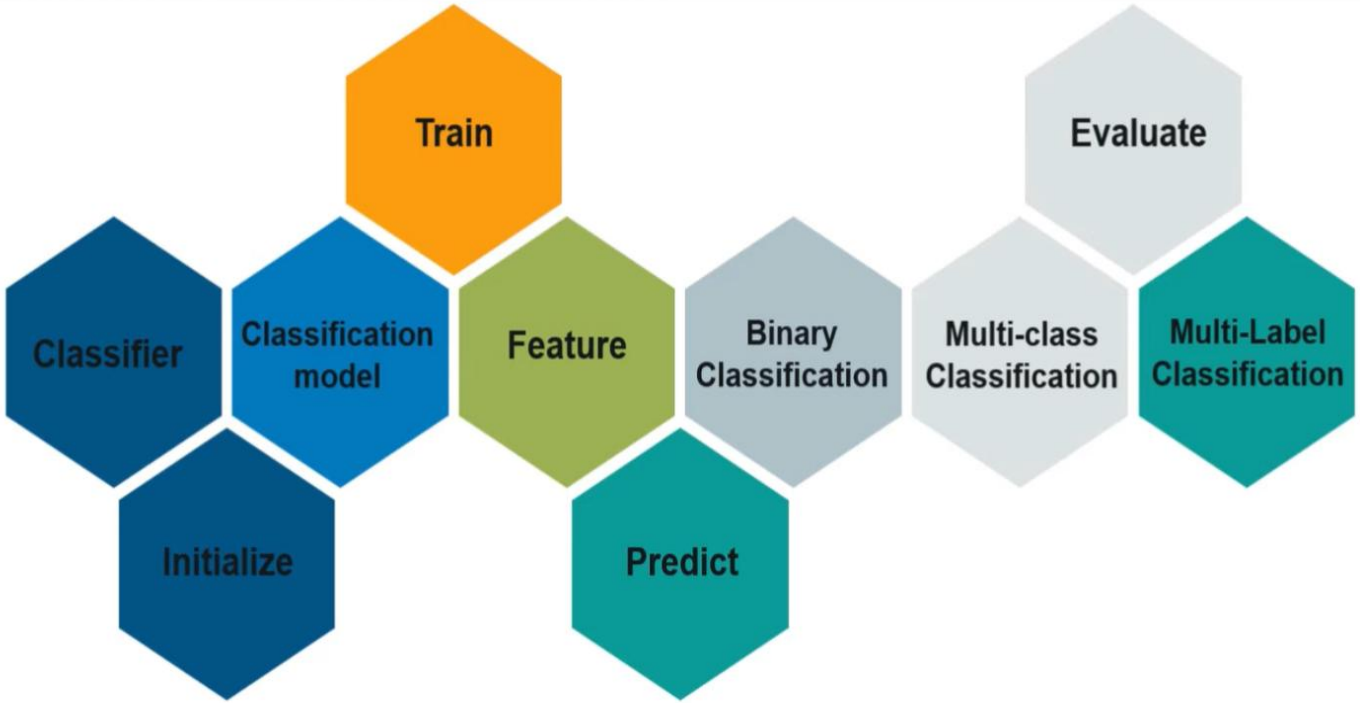
Dr. Dipak Kumar Mohanty
KIIT University, Bhubaneswar

What is Classification In Machine Learning?



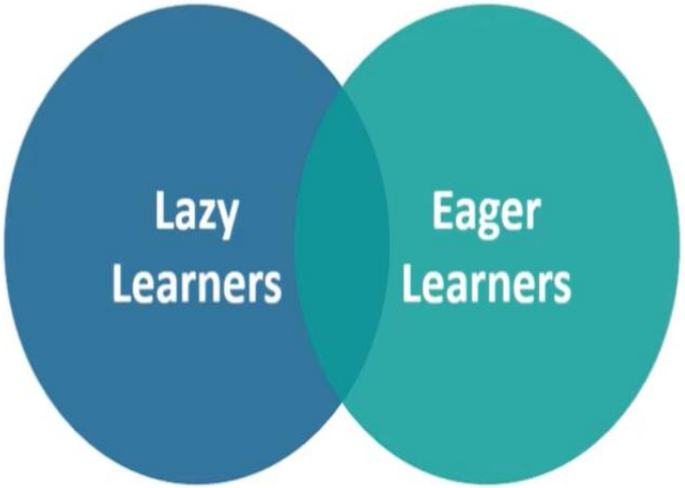
Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

Classification Terminologies



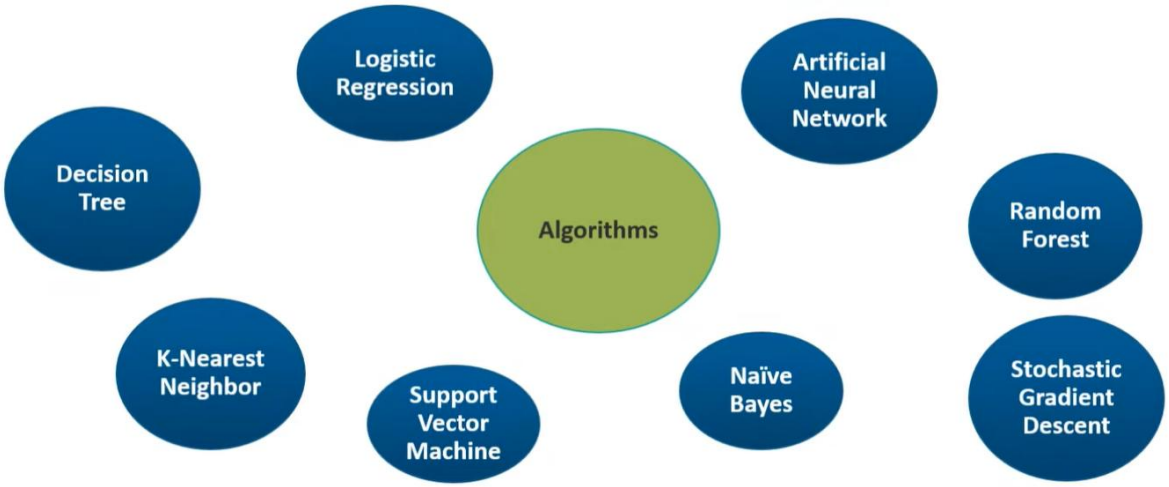
Types Of Learners In Classification

Lazy learners simply store the training data and wait until a testing data appears.



Eager learners construct a classification model based on the given training data before getting data for predictions

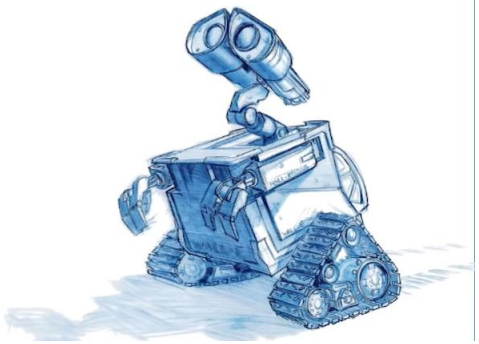
Classification Algorithms



KNN Algorithm

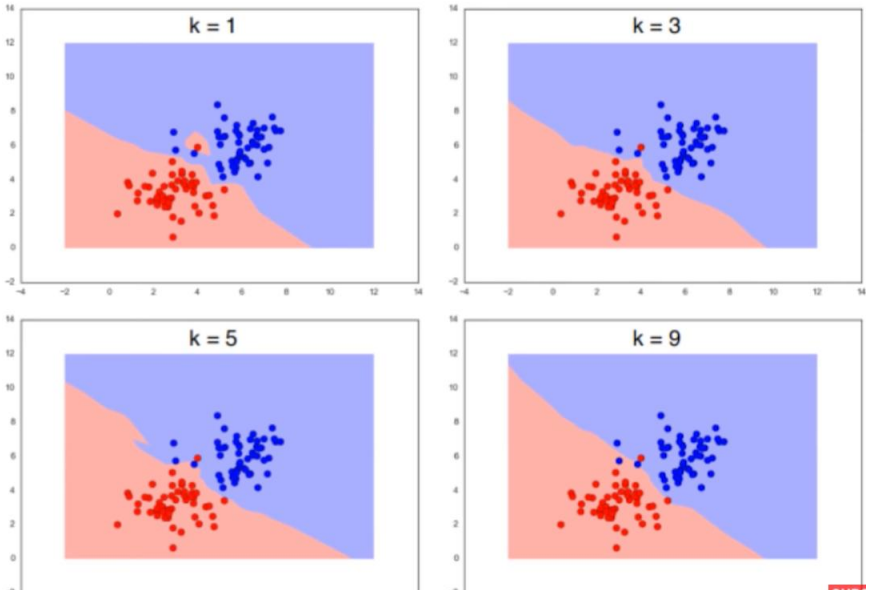
What is KNN Algorithm?

"K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure."

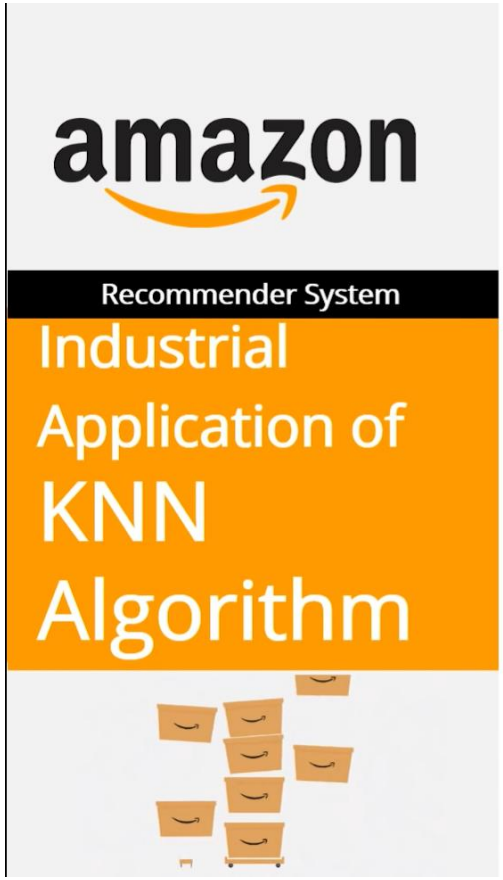


What is K in KNN Algorithm?

K = Number of Nearest Neighbors



Example: Recommender System



Apple

Apple iPhone 8 (Space Grey, 64GB)

281 customer reviews | 195 answered questions

M.R.P.: ~~₹ 7,499,922.00~~

Price: ₹ 62,450.00 FREE Delivery.

You Save: ₹ 7,437,472.00 (99%)

Inclusive of all taxes

EMI starts at ₹2,969 per month. Options

Want it faster? Available with Fulfilled FREE delivery from another seller at ₹ 63,925.

Only 1 left in stock.

Delivery to pincode 603203 - Kanchipuram between Jul 11 - 13. Details

Sold and fulfilled by National-shoppe (4.5 out of 5 | 160 ratings).

6 offers from ₹ 63,925.00

Colour: Space Grey

Size name: 64GB

64GB | 256GB

- 11.93 centimeters (4.7-inch) capacitive touchscreen with 1334 x 750 pixels resolution
- iOS v11 operating system with 1.2GHz Apple A11 Bionic hexa core processor, 2GB RAM, 64GB internal memory and single SIM
- 1821mAH lithium-ion battery
- 1 year manufacturer warranty for device and in-box accessories including batteries from the date of purchase

› See more product details

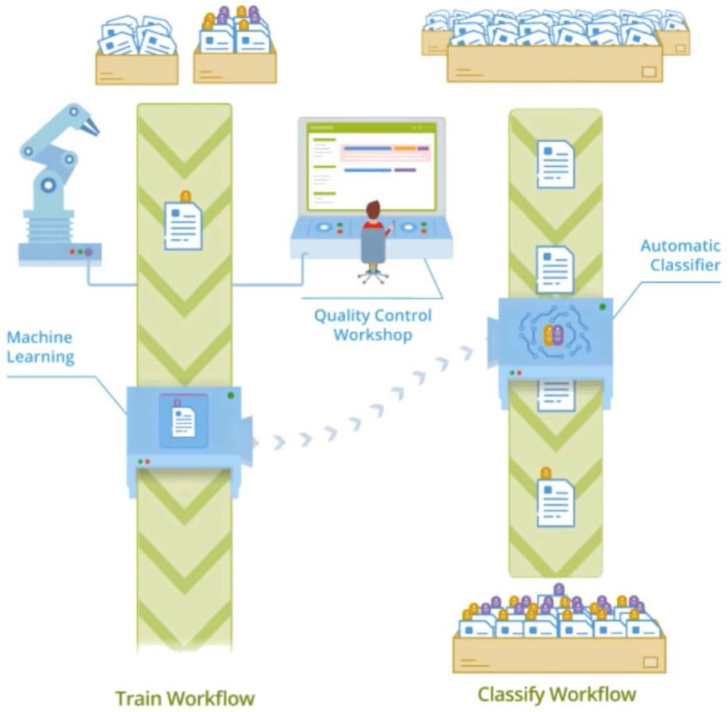
Report incorrect product information.

LIMITED QUANTITY — The order quantity for this product is limited to 1 unit per customer. Please note that orders which exceed the quantity limit will be auto-canceled. This is applicable across sellers.

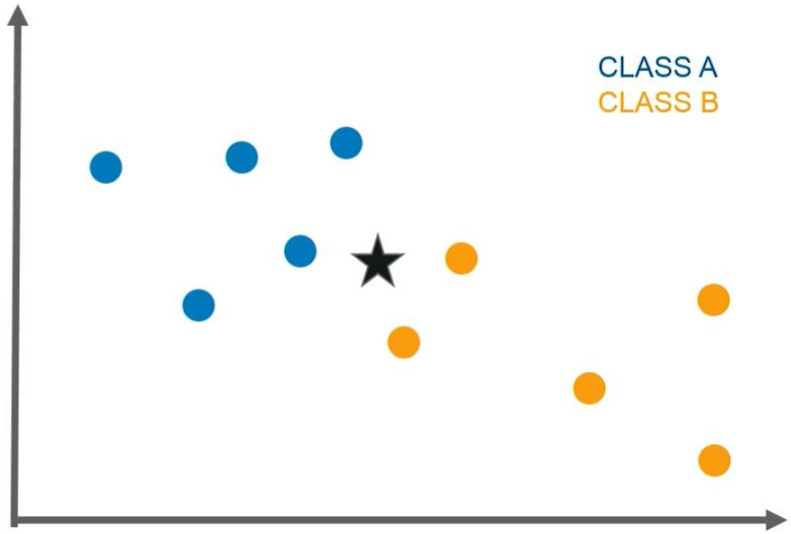
Customers who bought this item also bought

Page 1 of 10

Example: Handwritten detection, Image/video recognization ,

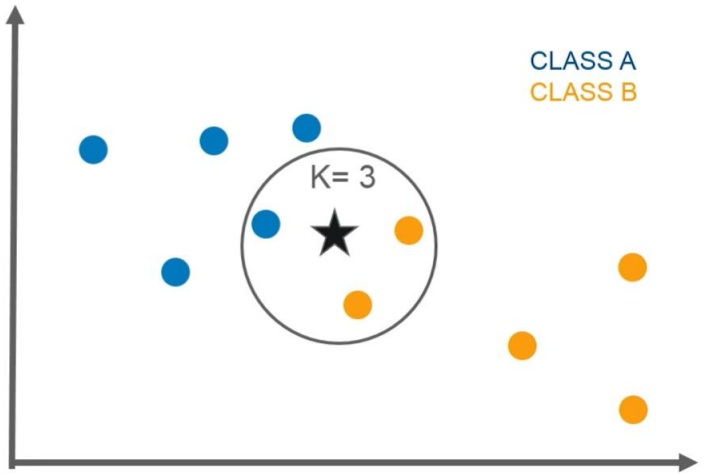


How does a KNN Algorithm work?



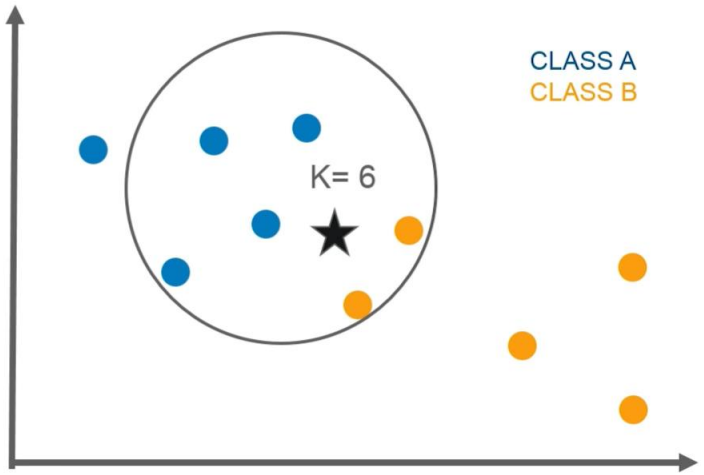
Here, Let k=3

How does a KNN Algorithm work?



What happens if k=6 ?

How does a KNN Algorithm work?



As majority of points are blue, so we can say Star belongs to A

A few thoughts on picking a value for "K"

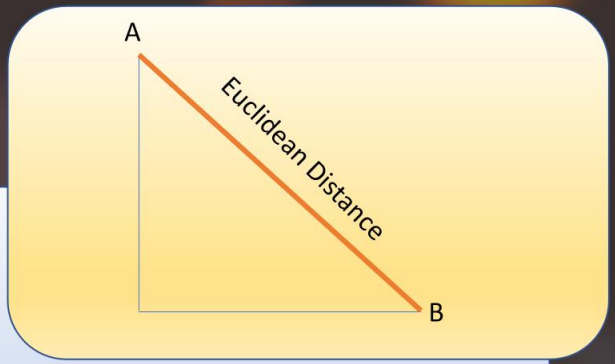
- There is no physical or biological way to determine the best value for "K", so you may have to try out a few values before settling on one. Do this by pretending part of the training data is "unknown".
- Low values for K (like K=1 or K=2) can be noisy and subject to the effects of outliers.
- Large values for K smooth over things, but you don't want K to be so large that a category with only a few samples in it will always be out voted by other categories.

Types of distances typically we use for ML models.

Minkowski Distance

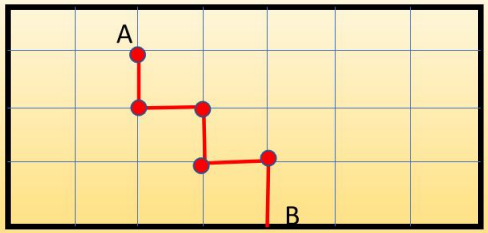
- $D(X_i, X_j) = \left(\sum_{d=1}^{D} |X_{id} - X_{jd}|^p\right)^{1/p}$
- Minkowski distance is defined as the similarity metric between two points in the normed vector space.
- Here, Norm is defined as a function that assigns a strictly positive length to each vector in a vector-space (exception is zero vector)
- $It\ is\ also\ known\ as\ L_p\ norm$

Euclidean Distance (p=2)



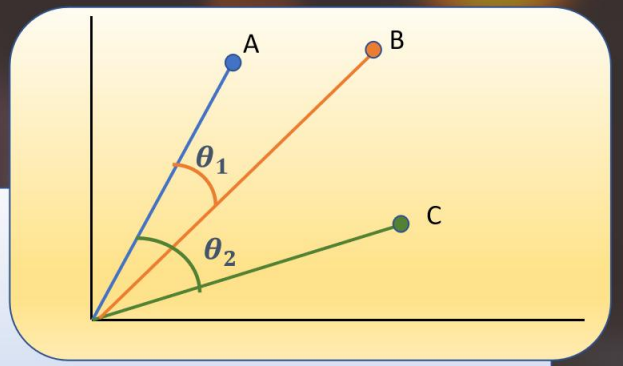
- $D(X_i, X_j) = \left(\sum_{d=1}^{D} |X_{id} - X_{jd}|^2\right)^{1/2}$
- Euclidean distance is a special form of Minkowski distance with the value of p=2
- Euclidean distance is the general distance we talk about in our real life situation.
- It is actually the distance we calculate from the Pythagoras theorem.
- $It\ is\ also\ known\ as\ L_2\ norm$

Manhattan Distance (p=1)



- $D(X_i, X_j) = \left(\sum_{d=1}^{D} |X_{id} - X_{jd}|^1\right)^{1/1}$
- Manhattan distance is also a special form of Minkowski distance with the value of p=1
- Manhattan distance is not the straight distance like Euclidean distance, it's similar to a grid like distance we consider while traveling from one place to other.
- $It\ is\ also\ known\ as\ L_1\ norm$

Cosine Distance
Cosine Similarity



- Cosine Similarity is cosine of the angle between two vectors.
- ***Cosine Similarity = $cos\ \theta$***
- ***Cosine Distance = 1-$cos\ \theta$***
- Cosine Distance and Cosine Similarity are inversely proportional to each other.
- ***Cosine Similarity*** $\epsilon\ [-1,\ +1]$
- Two dissimilar vectors will have cosine similarity approaching to -1 and two similar vectors will have cosine similarity approaching to +1

When to use each Distance

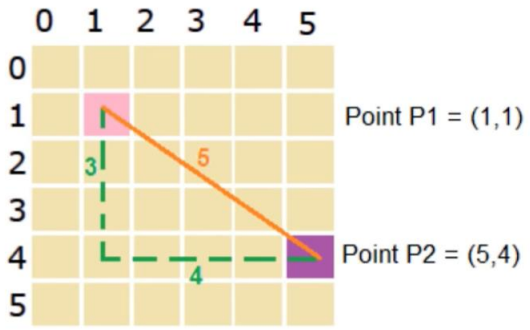
- Both Manhattan and Euclidean distances use in regression and classification problems.
- Euclidean distance doesn't perform well due to 'curse of dimensionality'
- Manhattan distance is linear in nature so used in high dimensional data as well as pointing outliers.
- Cosine Similarity is mainly used in recommendation systems, to calculate similar textual data.

https://colab.research.google.com/drive/1xKxNyTb7BCtXSUD2larqunIMQsFdcHt1?usp=sharing

For code Link : Types of distances typically we use for ML models.

How things are predicted using KNN Algorithm?

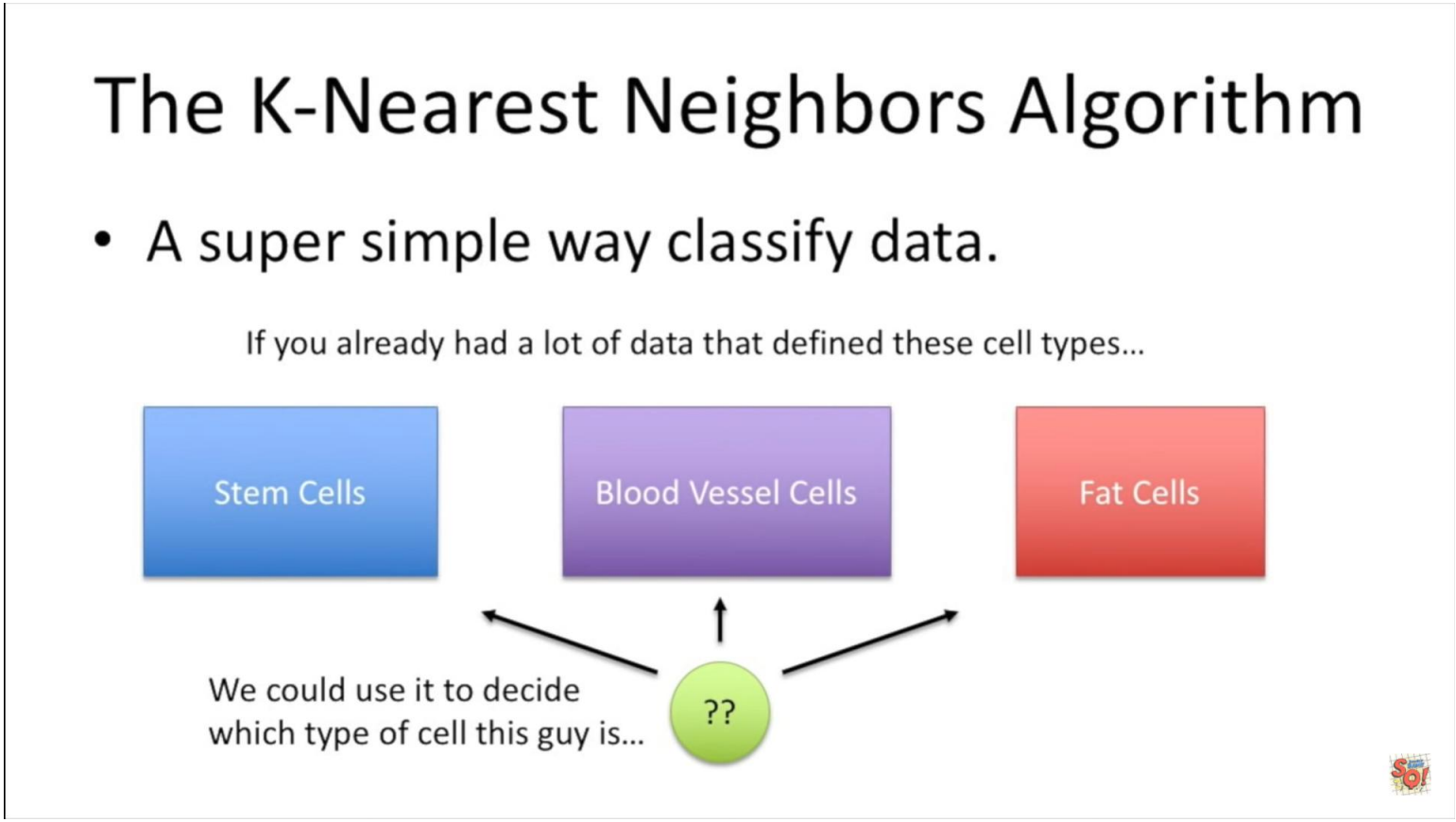
Manhattan Distance



Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

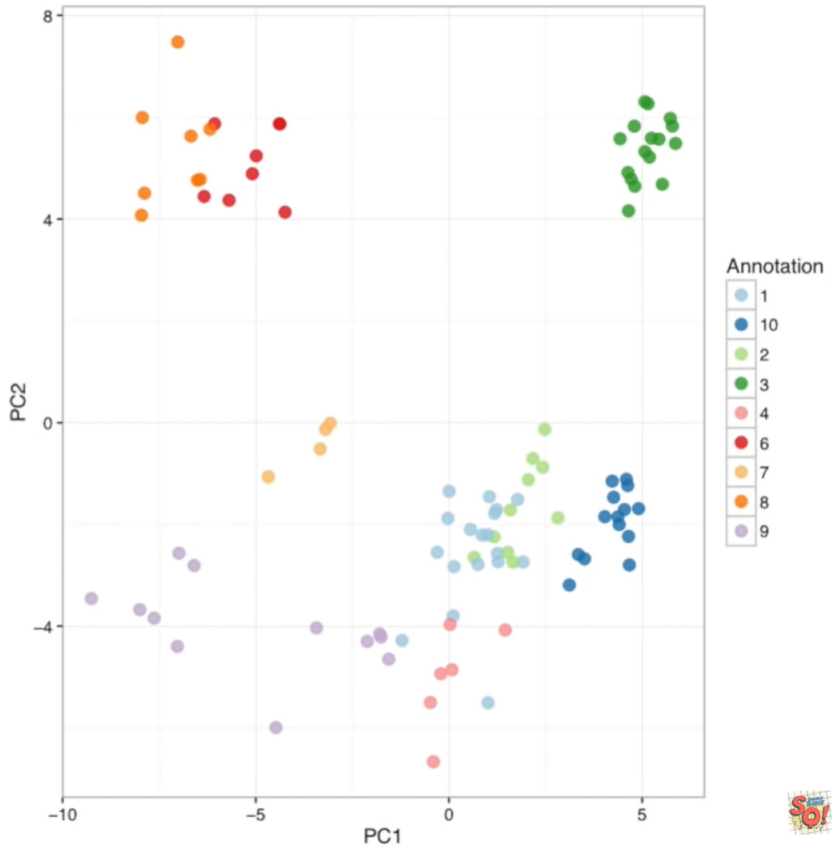
Example : KNN Algorithm



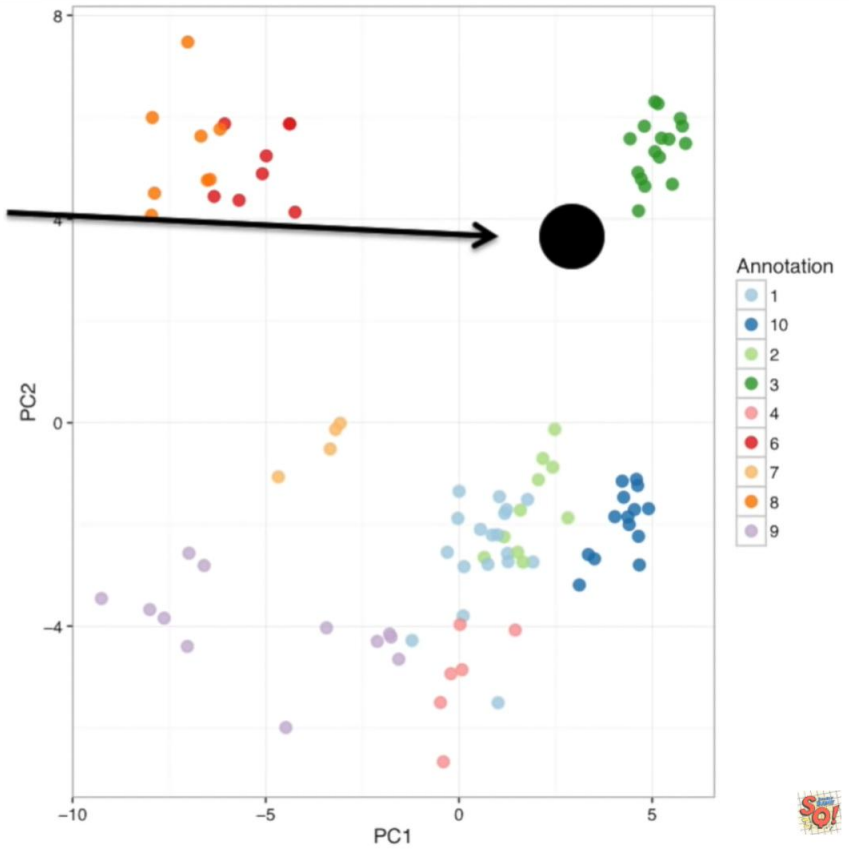
The K-Nearest Neighbors Algorithm

- A super simple way classify data.
- Let's see it in action…

Step 1: Start with a dataset with known categories. In this case, we have different cell types from an intestinal tumor. Then cluster that data. In this case, we used PCA.



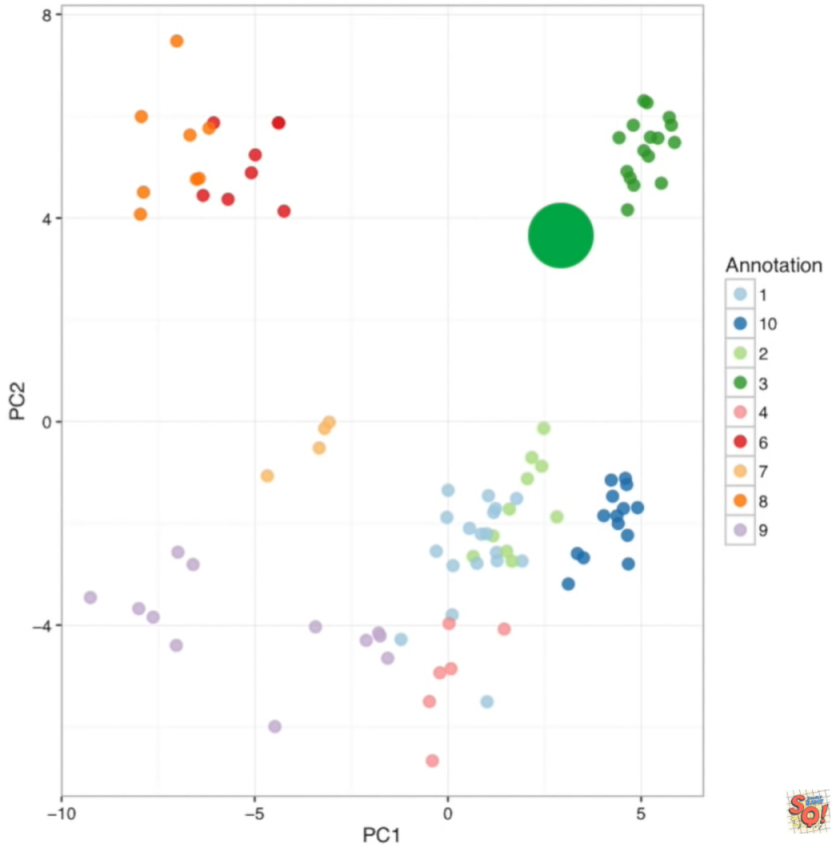
Step 2: Add a new cell, with unknown category, to the PCA plot. We don't know this cell's category because it was taken from another tumor where the cells were not properly sorted.



Step 3: We classify the new cell by looking at the nearest annotated cells. (i.e. the “nearest neighbors”).

If the “K” in “K-nearest neighbors” is equal to 1, then we only use the nearest neighbor to define the category.

In this case, the category is **GREEN**.



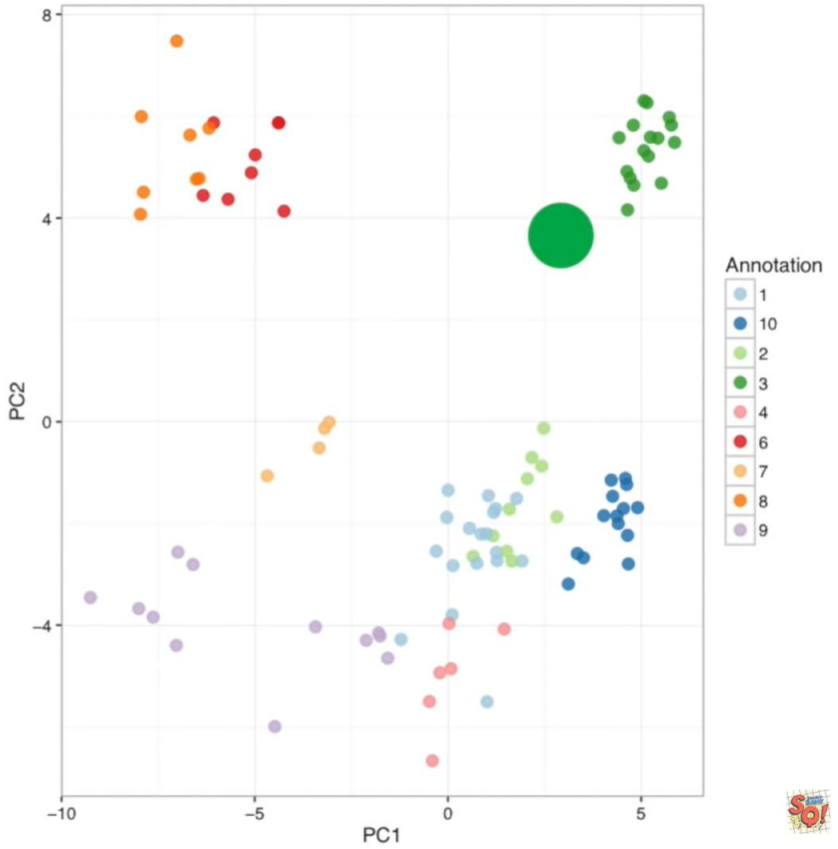
Step 3: We classify the new cell by looking at the nearest annotated cells. (i.e. the “nearest neighbors”).

If the “K” in “K-nearest neighbors” is equal to 1, then we only use the nearest neighbor to define the category.

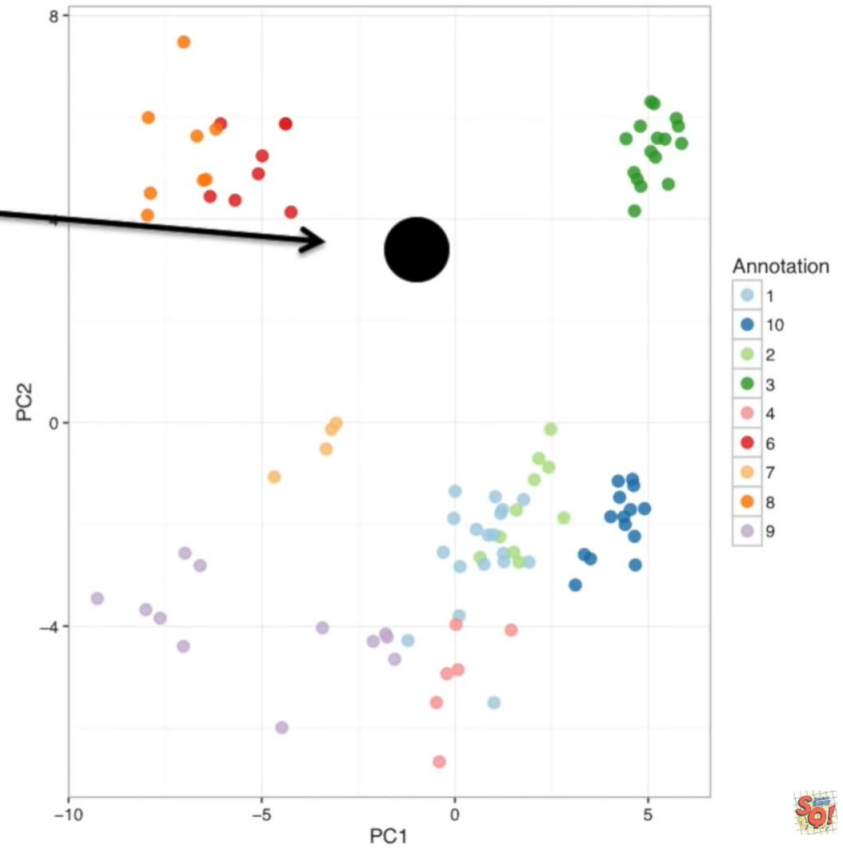
In this case, the category is **GREEN**.

If K=11, we would use the 11 nearest neighbors.

In this case, the category is still **GREEN**.



Now the new cell is somewhere more interesting…



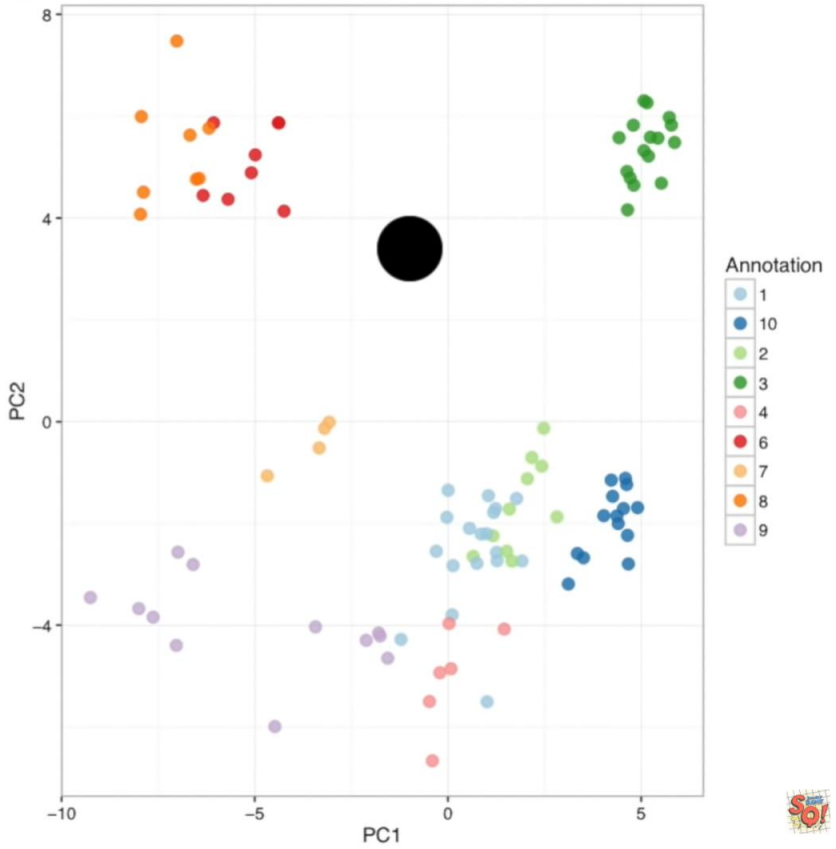
If K=11 and the new cell is between two (or more) categories, we simply pick the category that "gets the most votes".

In this case….

7 nearest neighbors are **RED**.

3 nearest neighbors are **ORANGE**.

1 nearest neighbor is **GREEN**.



If K=11 and the new cell is between two (or more) categories, we simply pick the category that "gets the most votes".

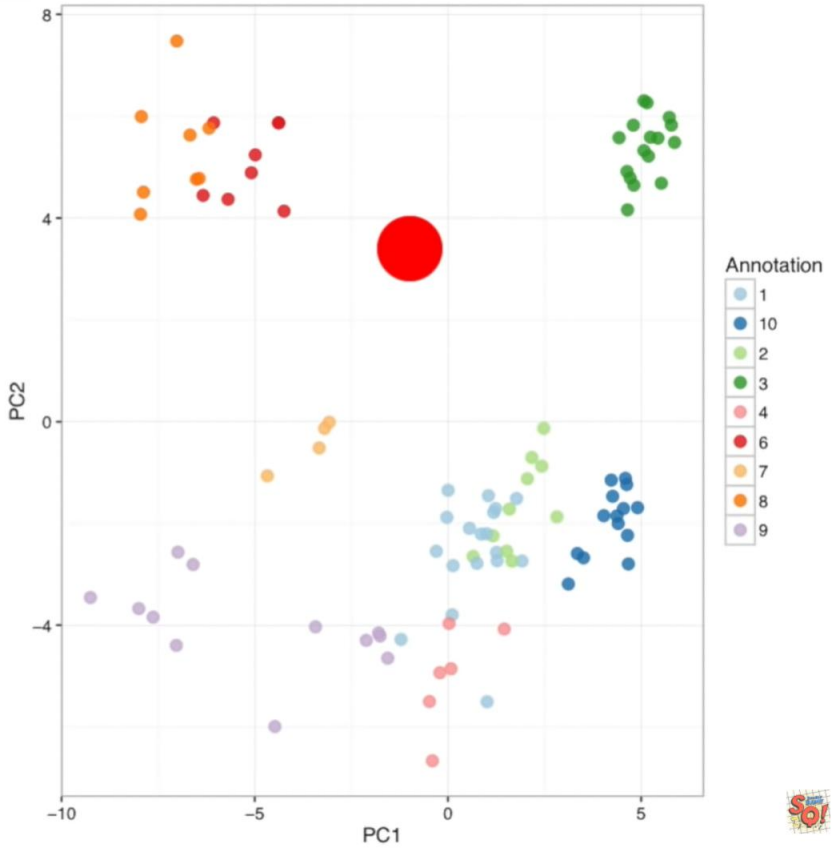
In this case….

7 nearest neighbors are **RED**.

3 nearest neighbors are **ORANGE**.

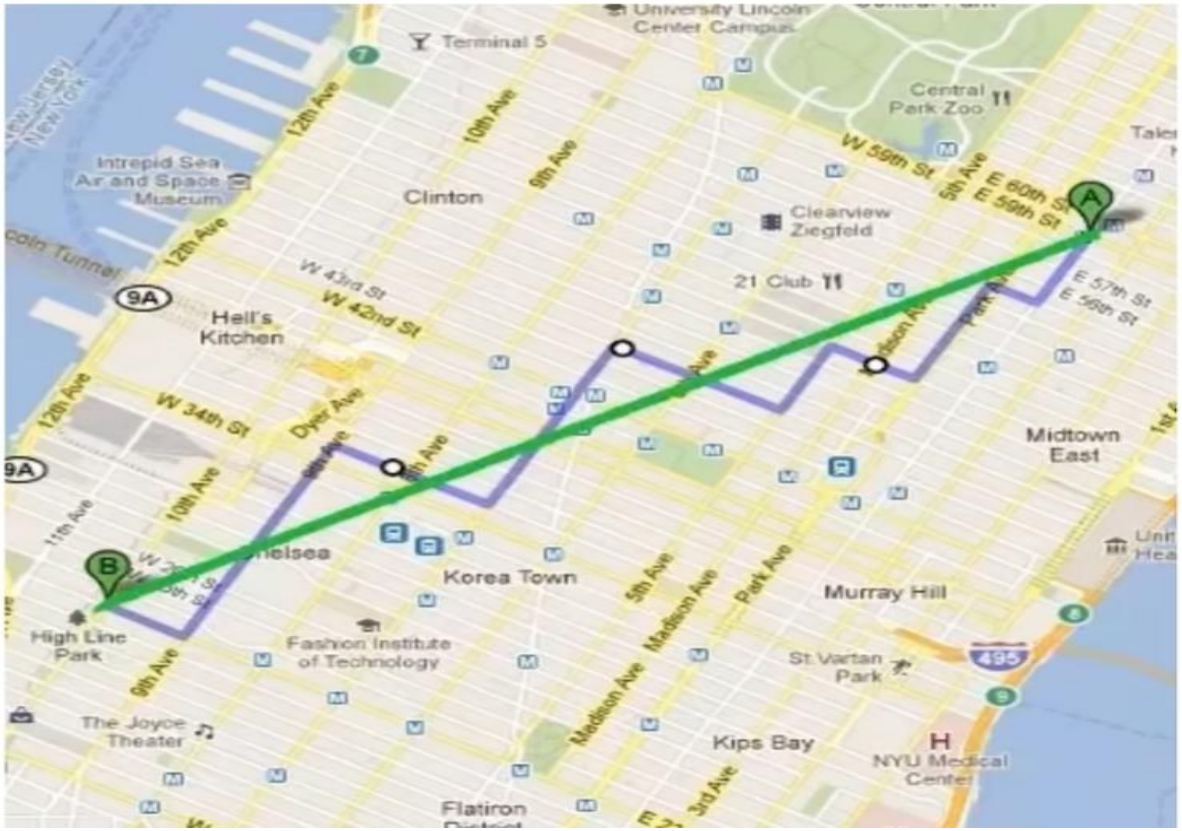
1 nearest neighbor is **GREEN**.

Since **RED** got the most votes, the final assignment is **RED**.



Man-distance majored along axes at right angle.

Manhattan Distance
vs
Euclidean Distance



KNN _algorithm used for both classification and regression purpose.

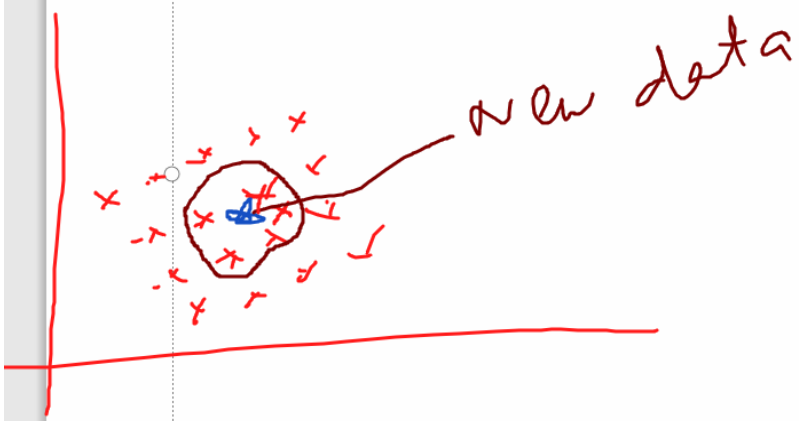
For Regression: let k=5

The predicted value = mean value of all 5 neighbors.

KNN _algorithm used for both classification and regression purpose.

For Regression: let k=5

The predicted value = mean value of all 5 neighbors.



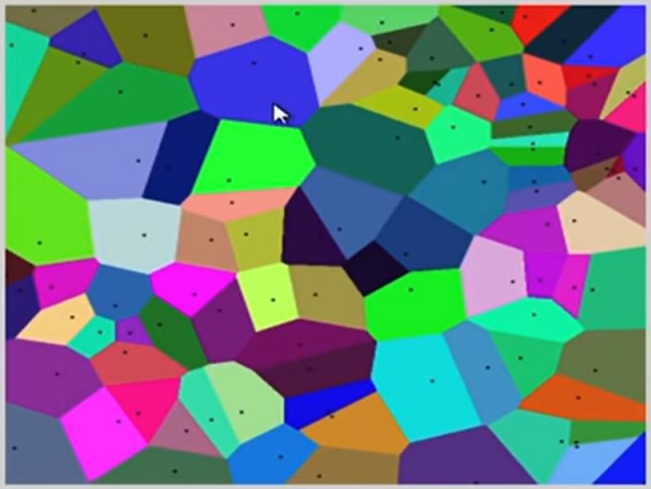
Carla P. Gomes
CS4700

Basic k-nearest neighbor classification

- Training method:
  - Save the training examples
- At prediction time:
  - Find the k training examples $(x_1, y_1), \ldots (x_k, y_k)$ that are closest to the test example x
  - Predict the most frequent class among those y_i's.

What is the decision boundary?

Voronoi diagram



The KNN Algorithm

The KNN Algorithm

1. Load the data

2. Initialize K to your chosen number of neighbors

3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances

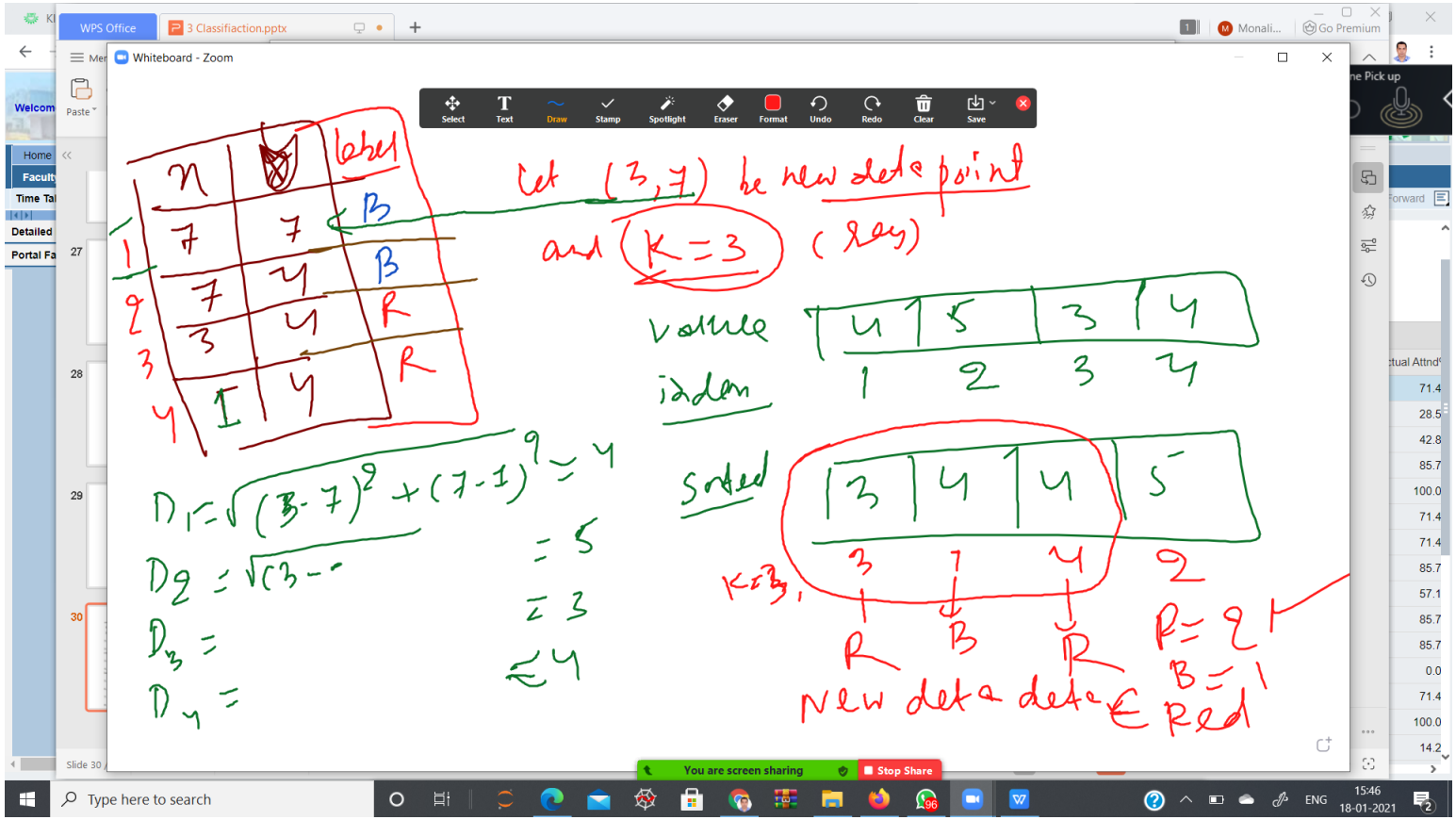
5. Pick the first K entries from the sorted collection

6. Get the labels of the selected K entries

7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

Example:


Whiteboard - Zoom
Let (3,7) be new data point
and K=3 (say)
values
index
sorted
New data data ∈ Red
You are screen sharing
Stop Share

Basic k-nearest neighbor classification

- Training method:
  - Save the training examples
- At prediction time:
  - Find the k training examples $(x_1,y_1),\ldots(x_k,y_k)$ that are closest to the test example x
  - Classification: Predict the most frequent class among those y_i's.
  - Regression: Predict the average of among the yi's.
- Improvements:
  - *Weighting* examples from the neighborhood
  - Measuring "*closeness*"
  - Finding "close" examples in a large training set *quickly*

How to select the value of K in the K-NN Algorithm?

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Summary of KNN-Algorithm

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the categor is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new dat appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the da and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a categ

Weighted KNN

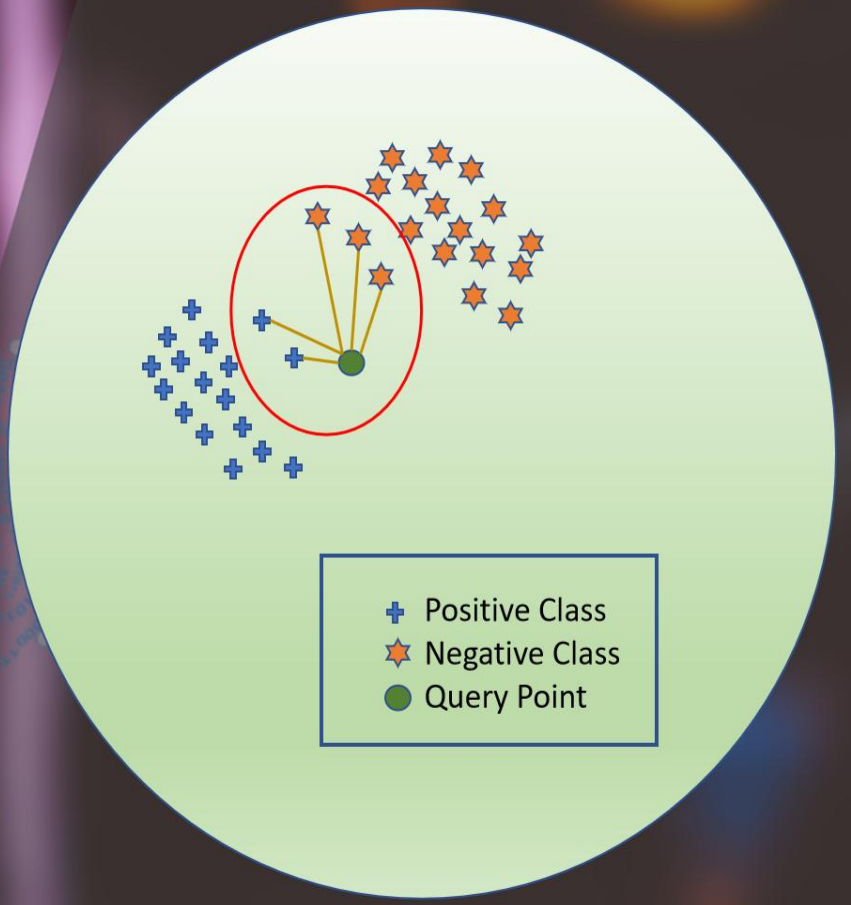
Weights as a function of distances

$$weight = \mathrm{F}(distance) = \frac{1}{distance}$$

Give Weights based on distance

Neighbour	True Label	Distance	Weight	Sum of Weights
X_1	Positive	0.1	10	13.3
X_2	Positive	0.3	3.3	
X_3	Negative	1	1	1.8
X_4	Negative	2	0.5	
X_5	Negative	3	0.3	

Predict Class labels based on the weighted-sum not on majority vote

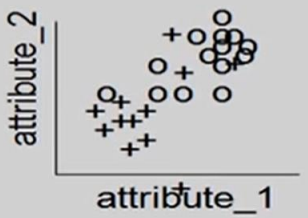


k-Nearest Neighbor

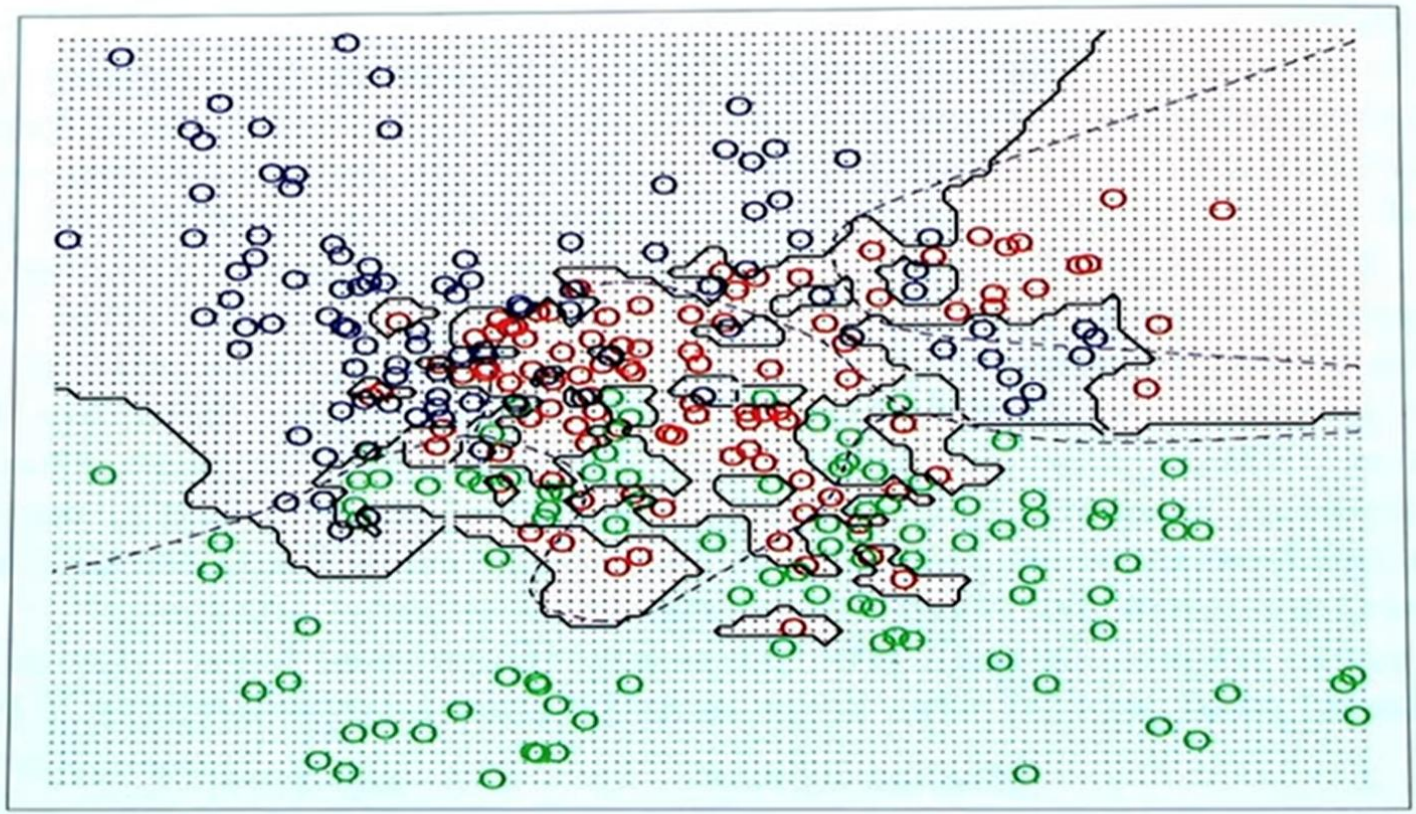
$$Dist(c_1, c_2) = \sqrt{\sum_{i=1}^{N} \left(attr_i(c_1) - attr_i(c_2)\right)^2}$$

$$prediction_{test} = ?$$

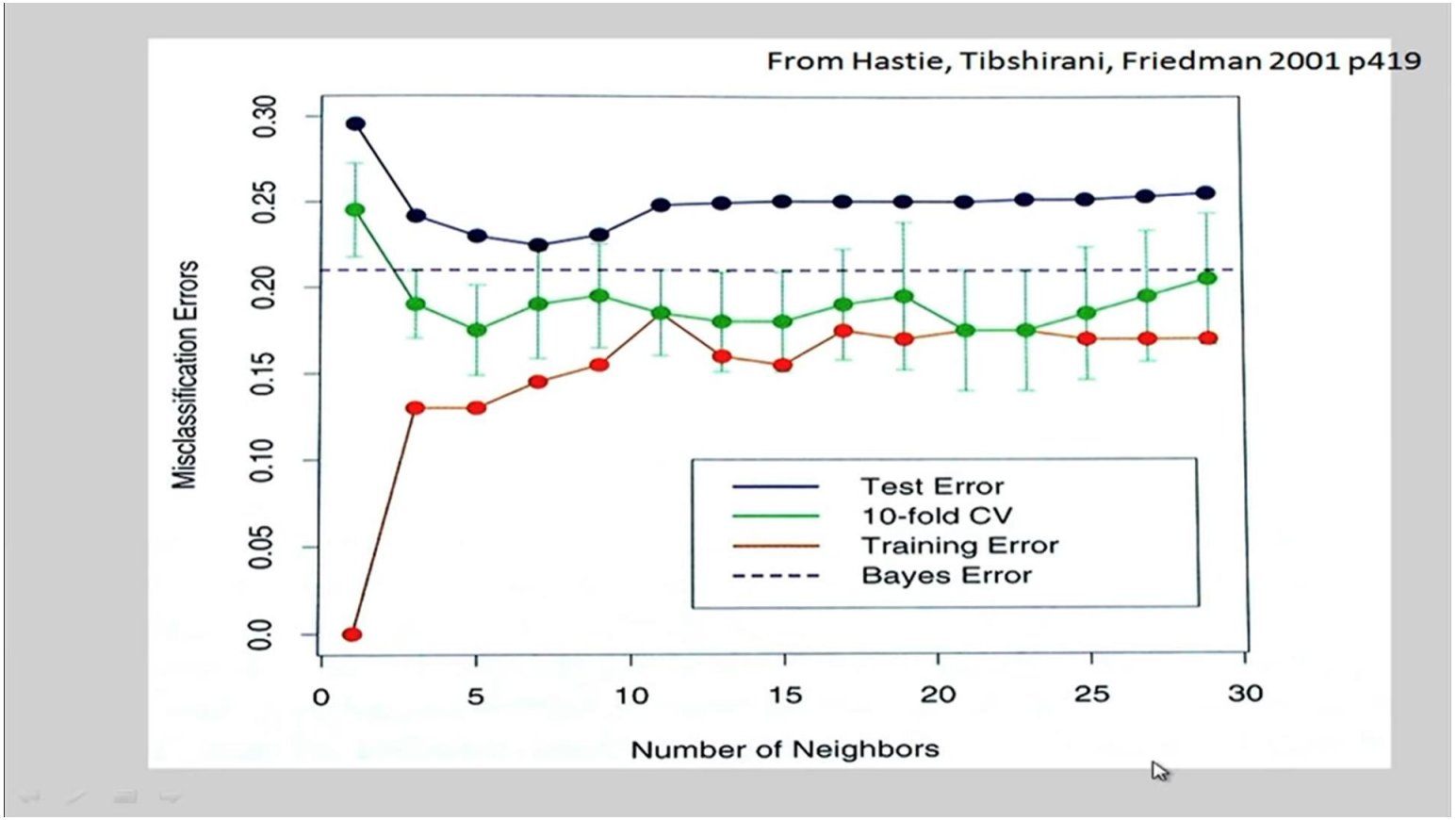
- Average of k points more reliable when:
  - noise in attributes
  - noise in class labels
  - classes partially overlap



1-Nearest Neighbor



From Hastie, Tibshirani, Friedman 2001 p418


From Hastie, Tibshirani, Friedman 2001 p419
Misclassification Errors
0.0
0.05
0.10
0.15
0.20
0.25
0.30
0
5
10
15
20
25
30
Number of Neighbors
Test Error
10-fold CV
Training Error
Bayes Error

Weighted Euclidean Distance

$$D(c1, c2) = \sqrt{\sum_{i=1}^{N} w_i \cdot \left(attr_i(c1) - attr_i(c2)\right)^2}$$

- large weights => attribute is more important
- small weights => attribute is less important
- zero weights => attribute doesn't matter

- Weights allow kNN to be effective with axis-parallel elliptical classes
- Where do weights come from?

Distance-Weighted kNN

- tradeoff between small and large k can be difficult
  - use large k, but more emphasis on nearer neighbors?

$$prediction_{test} = \frac{\sum_{i=1}^{k} w_i * class_i}{\sum_{i=1}^{k} w_i} \left(or \frac{\sum_{i=1}^{k} w_i * value_i}{\sum_{i=1}^{k} w_i} \right)$$

$$w_k = \frac{1}{Dist(c_k, c_{test})}$$

Locally Weighted Averaging

- Let k = number of training points
- Let weight fall-off rapidly with distance

$$prediction_{test} = \frac{\sum_{i=1}^{k} w_i * class_i}{\sum_{i=1}^{k} w_i} \left(or \frac{\sum_{i=1}^{k} w_i * value_i}{\sum_{i=1}^{k} w_i} \right)$$

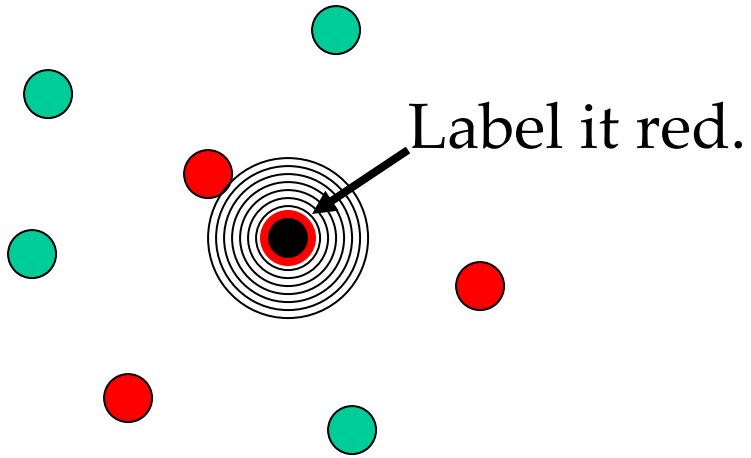
$$w_k = \frac{1}{e^{KernelWidth \cdot Dist(c_k, c_{test})}}$$

- KernelWidth controls size of neighborhood that has large effect on value (analogous to k)

1-Nearest Neighbor

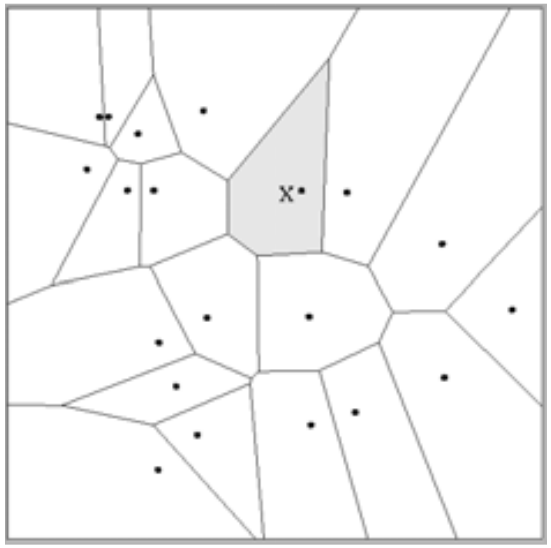
One of the simplest of all machine learning classifiers

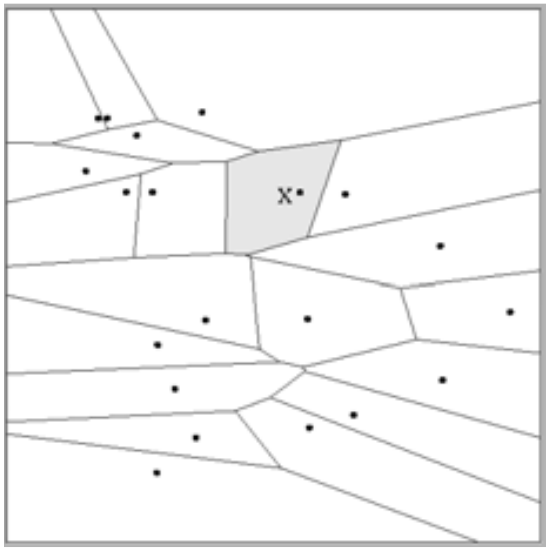
Simple idea: label a new point the same as the closest known point



Distance Metrics

Different metrics can change the decision surface





Dist(**a,b**) $=(a_1 - b_1)^2 + (a_2 - b_2)^2$

Dist(**a,b**) $=(a_1 - b_1)^2 + (3a_2 - 3b_2)^2$

Standard Euclidean distance metric:

- Two-dimensional: Dist(a,b) = sqrt($(a_1 - b_1)^2 + (a_2 - b_2)^2$)
- Multivariate: Dist(a,b) = sqrt($\sum (a_i - b_i)^2$)

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

1-NN's Aspects as an Instance-Based Learner:

A distance metric

- Euclidean
- When different units are used for each dimension
  → normalize each dimension by standard deviation
- For discrete data, can use hamming distance
  → D(x1,x2) =number of features on which x1 and x2 differ
- Others (e.g., normal, cosine)

How many nearby neighbors to look at?

- One

How to fit with the local points?

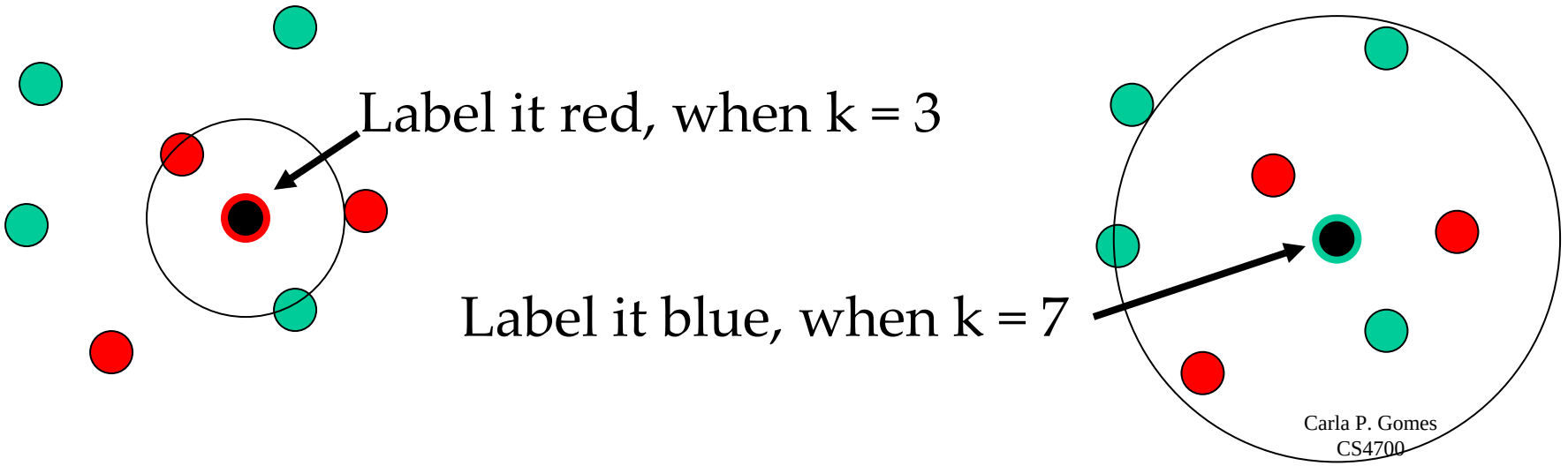
- Just predict the same output as the nearest neighbor.

Adapted from "Instance-Based Learning" lecture slides by Andrew Moore, CMU.

k – Nearest Neighbor

Generalizes 1-NN to smooth away noise in the labels

A new point is now assigned the most frequent label of its *k* nearest neighbors



KNN Example

	Food (3)	Chat (2)	Fast (2)	Price (3)	Bar (2)	BigTip
1	great	yes	yes	normal	no	yes
2	great	no	yes	normal	no	yes
3	mediocre	yes	no	high	no	no
4	great	yes	yes	normal	yes	yes

Similarity metric: Number of matching attributes (k=2)

New examples:

- Example 1 (great, no, no, normal, no) Yes

  → most similar: number 2 (1 mismatch, 4 match) → yes

  →Second most similar example: number 1 (2 mismatch, 3 match) → yes

- Example 2 (mediocre, yes, no, normal, no) Yes/No

  → Most similar: number 3 (1 mismatch, 4 match) → no

  →Second most similar example: number 1 (2 mismatch, 3 match) → yes

Selecting the Number of Neighbors

Increase k:

- Makes KNN less sensitive to noise

Decrease k:

- Allows capturing finer structure of space

➔Pick k not too large, but not too small (depends on data)

Curse-of-Dimensionality

Prediction accuracy can quickly degrade when number of attributes grows.

- Irrelevant attributes easily “swamp” information from relevant attributes
- When many irrelevant attributes, similarity/distance measure becomes less reliable

Remedy

- Try to remove irrelevant attributes in pre-processing step
- Weight attributes differently
- Increase k (but not too much)

Advantages and Disadvantages of KNN

Need distance/similarity measure and attributes that "match" target function.

For large training sets,

→ Must make a pass through the entire dataset for each classification. This can be prohibitive for large data sets.

Prediction accuracy can quickly degrade when number of attributes grows.

Simple to implement algorithm;

Requires little tuning;

Often performs quite weel!

(Try it first on a new learning problem).