Week 3

Machine Learning and Big Data - DATA622

Fall 2023

CUNY School of Professional Studies

Week 3

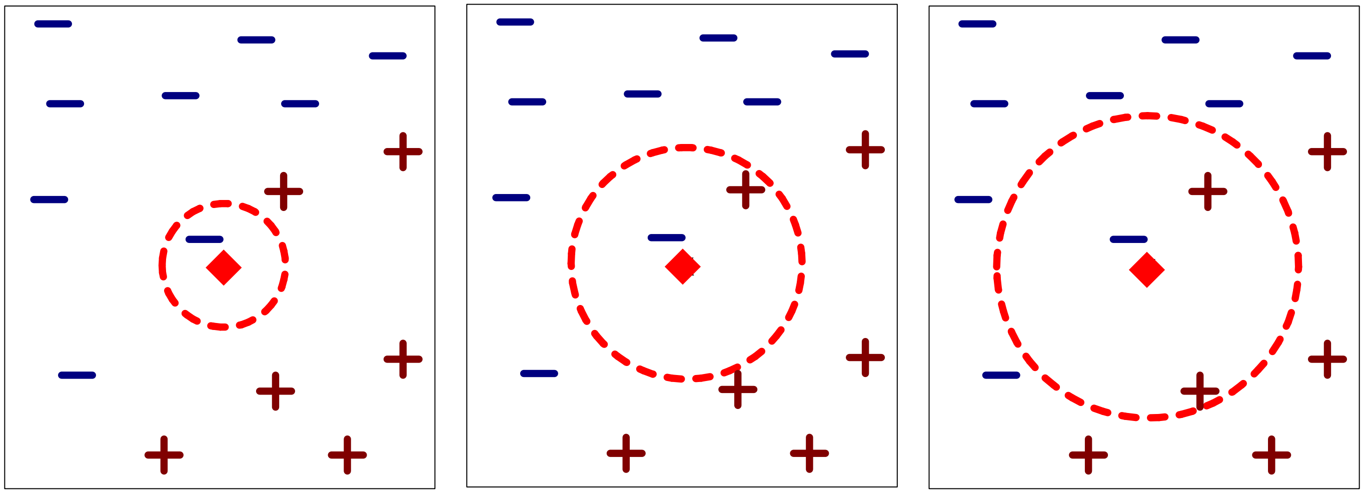
1. Discussion Board Week 3: Machine vs Statistical Learning
2. Reading materials:
   - KNN (k-nearest neighbors):
     - Practical Machine Learning with R Chapter 6 (1 hour)
     - Video: R-lab for kNN: https://youtu.be/OUqMjCfWKiQ (5 mins)
       (Video: Python lab for kNN: https://youtu.be/gGdlYicGIIo)
   - LDA (Linear Discriminant Analysis)
     - Video: Good explanation: https://youtu.be/gPOMLCR1oF8 (7 mins)
     - Video: Simplified explanation of LDA: https://youtu.be/azXCzI57Yfc (15 mins)
     - Video: More mathematical explanation of LDA: https://youtu.be/pZodpSfe9IA (18 mins)
     - Video: R-lab for LDA: https://youtu.be/Wklfvk2BH88 (8 mins)
       (Optional Video: Python lab for LDA: https://www.youtube.com/watch?v=9IDXYHhAfGA)
     - (Optional) How to implement LDA in R: https://datascienceplus.com/how-to-perform-logistic-regression-lda-qda-in-r/#:~:text=LDA%20(Linear%20Discriminant%20Analysis)%20is,for%20all%20class%20is%20normal
       (Python implementation for those who like python: https://hands-on.cloud/implementation-of-linear-discriminant-analysis-lda-using-python/)

K-Nearest Neighbor

Classify data according to its k-closest neighbors

School of Professional Studies

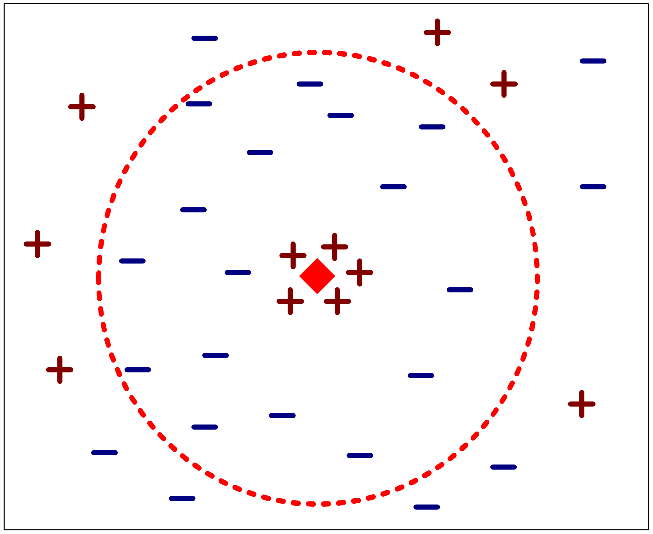
k-Nearest Neighbor (KNN)



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Choice of k

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other



Rule of thumb:

$$k = \sqrt{N}$$

N: number of training points

CUNY School of Professional Studies

Principal Component Analysis (PCA)

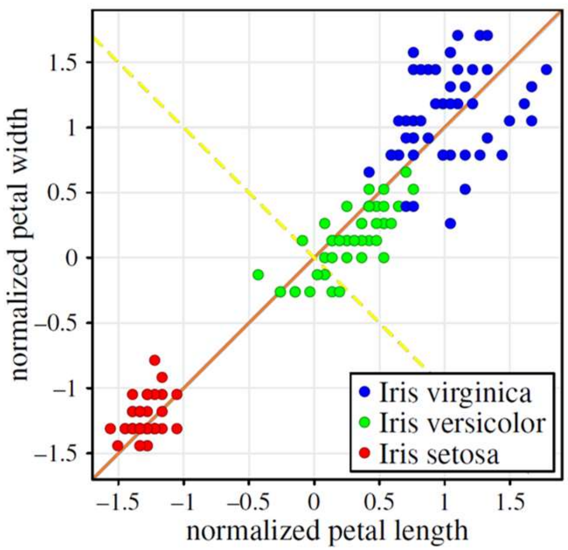
Dimension reduction technique based on maximum variance in data.

Principal Component Analysis (PCA)

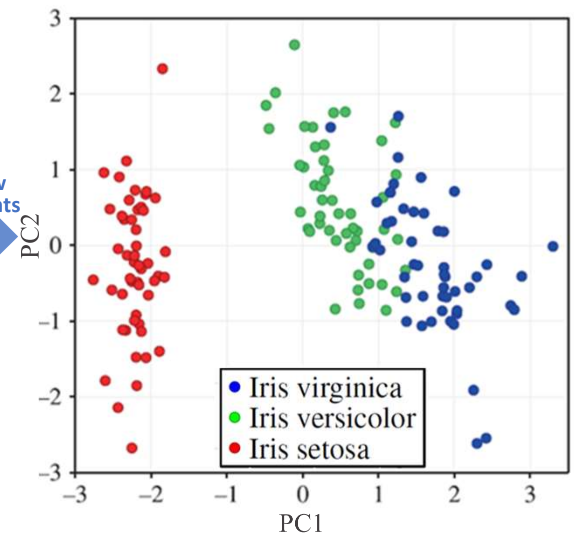
- Dimensionality reduction technique
- Project data from the high-dimensional space to a lower-dimensional space
- Criteria: Maximize data variance to construct principal components

Steps:
1. Construct a line in the direction of maximum variance in data (PC1 = orange line)
2. Next component is orthogonal to the previous component (PC2 = yellow line)

Repeat for as many PC as you need.



Replot onto new components



Source: Guide to Intelligent Data Science, Berthold et al

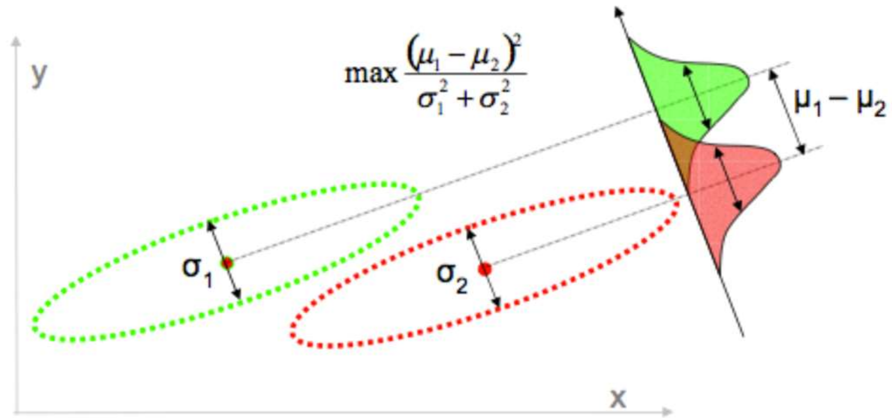
Linear Discriminant Analysis (LDA)

Dimension reduction technique based on maximizing distance between means <u>and</u> minimizing spread.

CUNY School of Professional Studies

Linear Discriminant Analysis (LDA)

- Dimensionality reduction approach
- Two criteria are used by LDA to create a new axis:
    1. Maximize the distance between means of the two classes.
    2. Minimize the variation (spread) within each class.

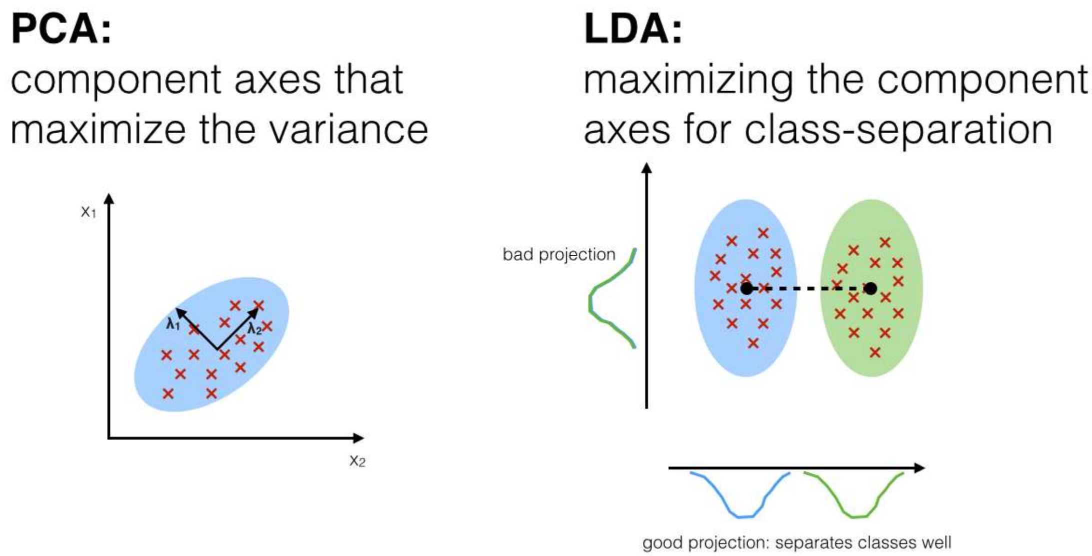


Source: Victor Lavrenko

CUNY School of Professional Studies

PCA vs LDA

	PCA	LDA
Transformation	Linear	Linear
Supervised vs Unsupervised	Un-supervised	Supervised
Objective	Capture variability by finding principal components	Separate classes by identifying a lower dimension which has better discriminatory power
Type	Component: maximize the variance <u>in the data</u>	Discriminant: maximize the separation <u>between classes</u>
Compute requirements	Low	High
Use-cases	Visualization (and classification)	Any classification



PC1

Source: Guide to Intelligent Data Science, Berthold et al

CUNY School of Professional Studies

PCS vs LDA

- Discriminants (LDA) maximize the separation between classes
- Components (PCA) maximize the variance in the data
- Dimensionality reduction technique
- Project data from the high-dimensional space to a lower-dimensional space
- Criteria: Maximize data variance to construct principal components

PC1

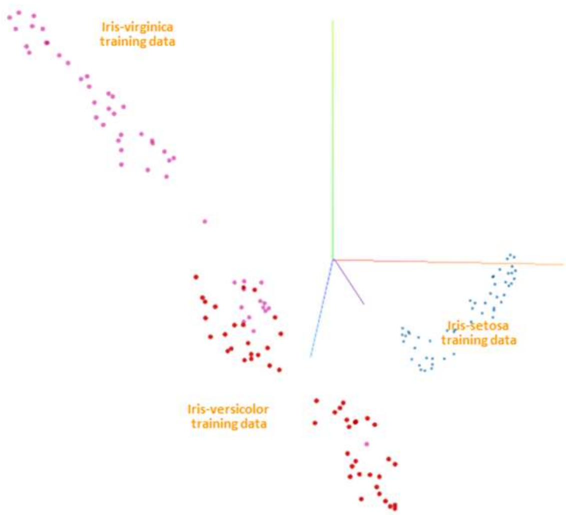
Source: Guide to Intelligent Data Science, Berthold et al

CUNY School of Professional Studies

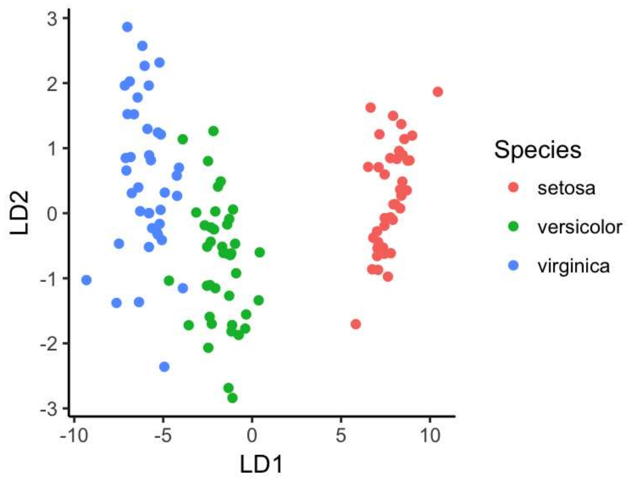
11

Example: Iris data set

4-dimensional input space



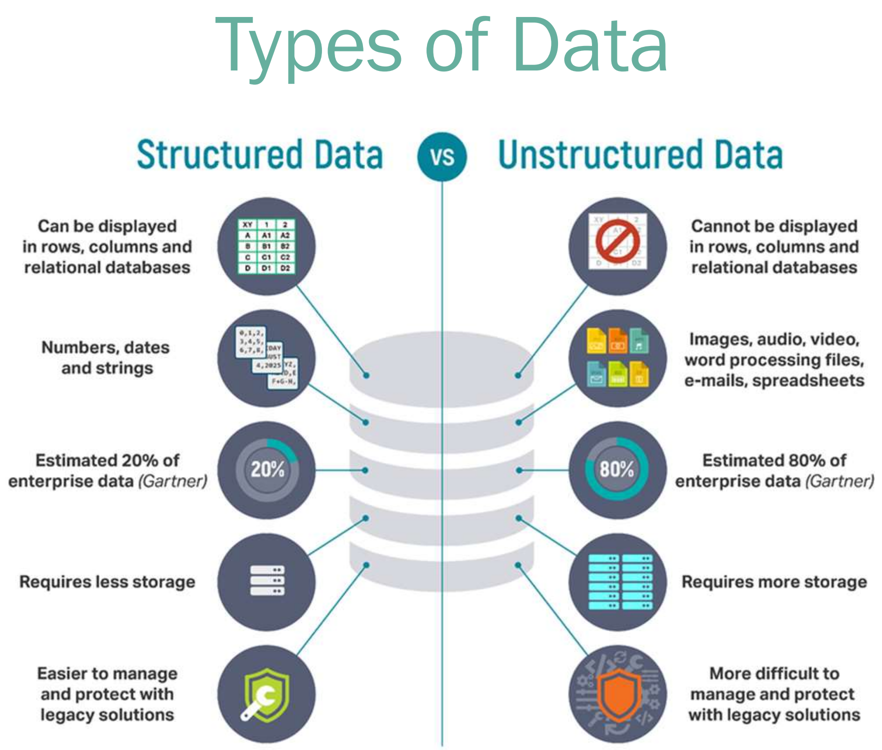
2-components (2-dimensions)



Data point-cloud

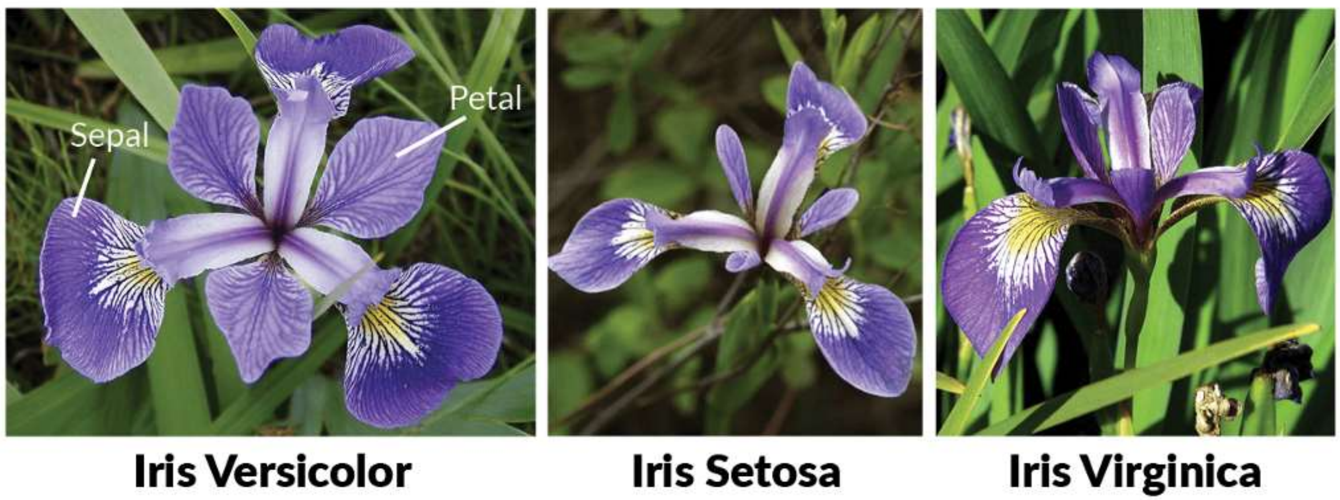
Tabular (structured) data can be plotted on an n-dimensional space (where n=number of input columns in the table). This creates a point-cloud of data points in the table (with each dot a line in the table). Unstructured data can also be plotted – but needs processing first.

Types of Data



Source: igneous.io

Example: Iris data set



Source: Fisher (1936) https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1469-1809.1936.tb02137.x

Iris data

Data is labeled (with 3 classes).

Features (inputs)

Labels

Instance	Sepal Length (cm)	Sepal width (cm)	Petal legth (cm)	Petal width (cm)	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
		• • •			
50	7	3.2	4.7	1.4	Iris-versicolor
51	6.4	3.2	4.5	1.5	Iris-versicolor
52	6.9	3.1	4.9	1.5	Iris-versicolor
53	5.5	2.3	4	1.3	Iris-versicolor
54	6.5	2.8	4.6	1.5	Iris-versicolor
55	5.7	2.8	4.5	1.3	Iris-versicolor
56	6.3	3.3	4.7	1.6	Iris-versicolor
		• • •			
100	6.3	3.3	6	2.5	Iris-virginica
101	5.8	2.7	5.1	1.9	Iris-virginica
102	7.1	3	5.9	2.1	Iris-virginica
103	6.3	2.9	5.6	1.8	Iris-virginica
104	6.5	3	5.8	2.2	Iris-virginica
105	7.6	3	6.6	2.1	Iris-virginica

A single instance

Labels (output) will have 3 classes

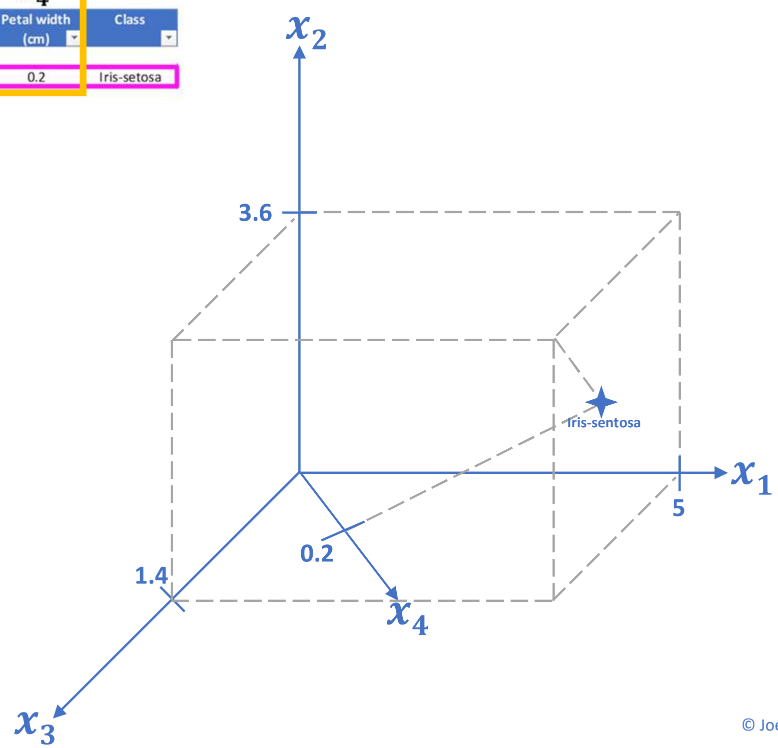
There are 4 features (inputs): x_1, x_2, x_3 & x_4

Features ("Independent inputs")

	x_1	x_2	x_3	x_4	
Instance	Sepal Length (cm)	Sepal width (cm)	Petal legth (cm)	Petal width (cm)	Class
1	5	3.6	1.4	0.2	Iris-setosa

A single instance

- Every feature is a dimension
  4 features = 4 dimensions
- An instance is a <u>single point</u> in that 4-dimensional space
- All of the data forms a <u>point- cloud</u> in that 4-dimensional space



Demo

projector.tensorflow.org

CUNY School of Professional Studies

One-hot encoding

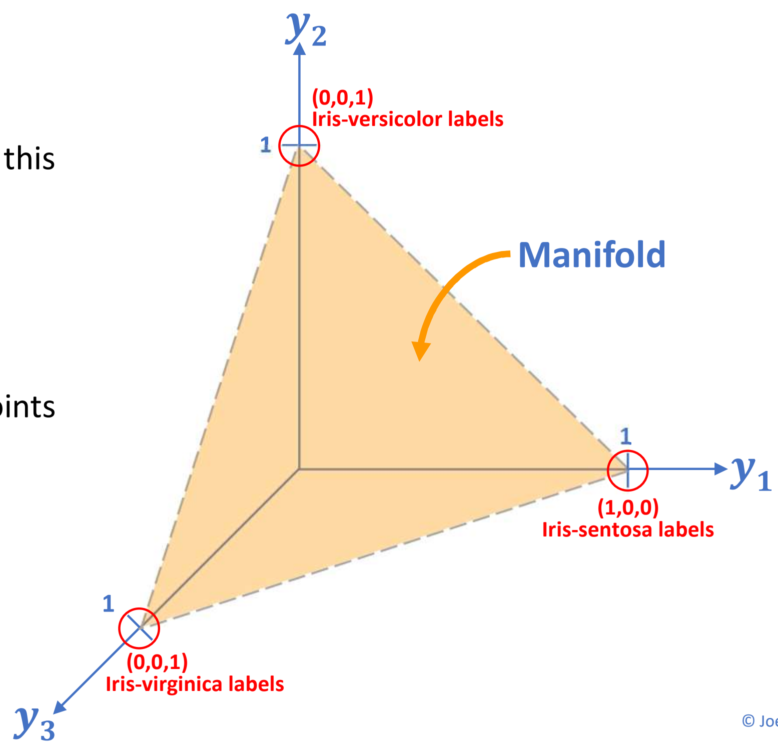
- ML requires numbers: labels must be converted to numbers
- Each class (type of label must be its own dimension)
- The value in each dimension conveys the probability it is of that class
- Training Data Labels always have a probability of 1 (100%) i.e. they are the “Ground Truth”

Labels	One-Hot Encoding	Labels		
Class		Class1 iris-setosa	Class2 iris-versicolo	Class3 iris-virginica
Iris-setosa	$(y_1, y_2, y_3) =$ $(1,0,0)$	1	0	0
Iris-setosa		1	0	0
Iris-setosa		1	0	0
Iris-setosa		1	0	0
Iris-setosa		1	0	0
Iris-setosa		1	0	0
• • •			• • •	
Iris-versicolor	$(0,1,0)$	0	1	0
Iris-versicolor		0	1	0
Iris-versicolor		0	1	0
Iris-versicolor		0	1	0
Iris-versicolor		0	1	0
Iris-versicolor		0	1	0
Iris-versicolor		0	1	0
• • •			• • •	
Iris-virginica	$(0,0,1)$	0	0	1
Iris-virginica		0	0	1
Iris-virginica		0	0	1
Iris-virginica		0	0	1
Iris-virginica		0	0	1
Iris-virginica		0	0	1

CUNY School of Professional Studies

Solution Manifold

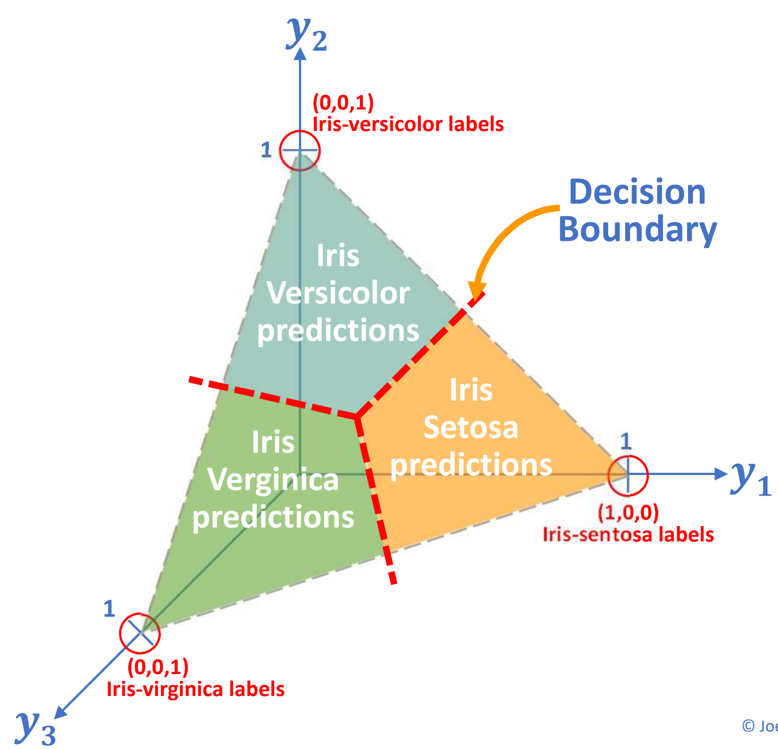
- The number of dimensions = number of classes. In this case 3 dimensions.
- A Label (or prediction) is one data-point in that 3-dimensional space
- Probabilities of all classes add up to 1 (100%) so points lie on a manifold
- Only labels have values of 1



CUNY School of Professional Studies

Decision Boundary

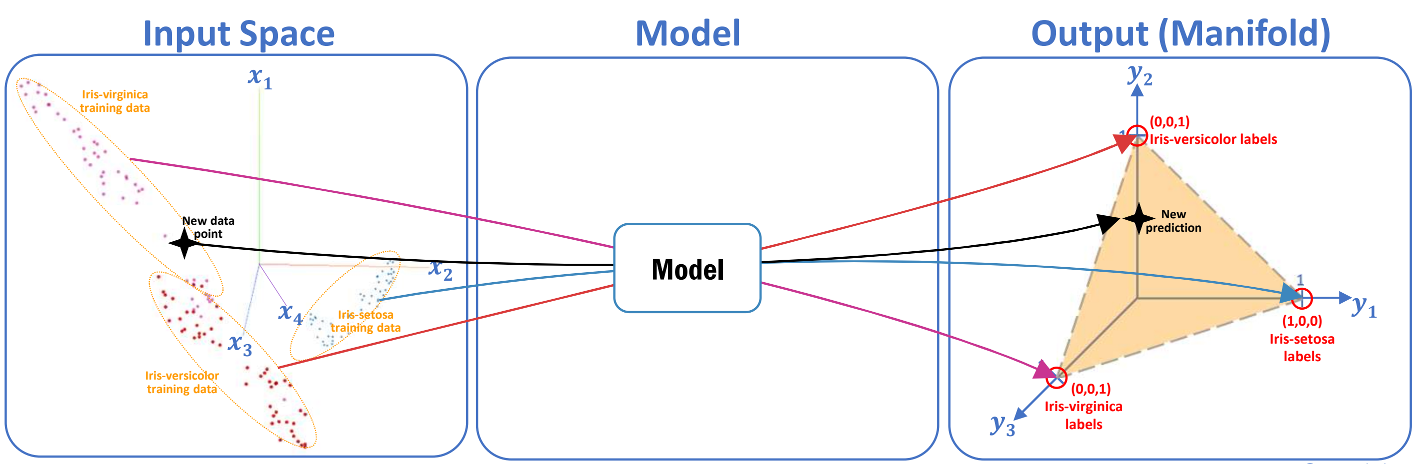
- A decision boundary separates the classes.
- For 2 classes the decision boundary is typically 0.5 (when probability of either class is 50%)
- It may be linear or non-linear



CUNY School of Professional Studies

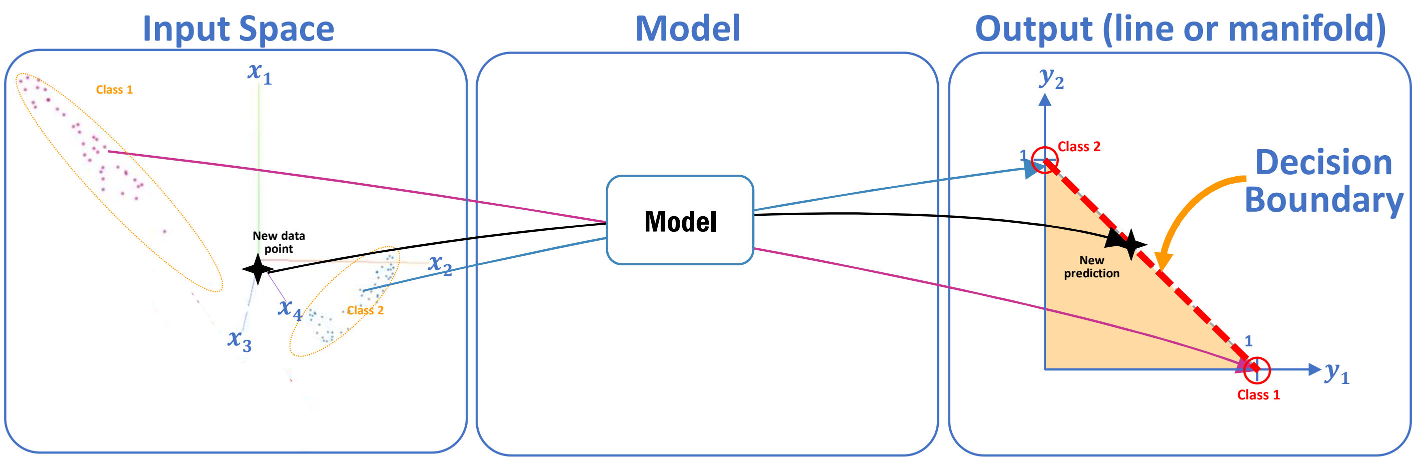
Putting it together

Three class results in a 3-dimensional output space, with the prediction landing on a line where: p(y_1) + p (y_2) + p (y_3) = 1



2 vs 3 classes

Two class results in a 2-dimensional output space, with the prediction landing on a line where: p($\boldsymbol{y_1}$) + p ($\boldsymbol{y_2}$) = 1



CUNY School of Professional Studies

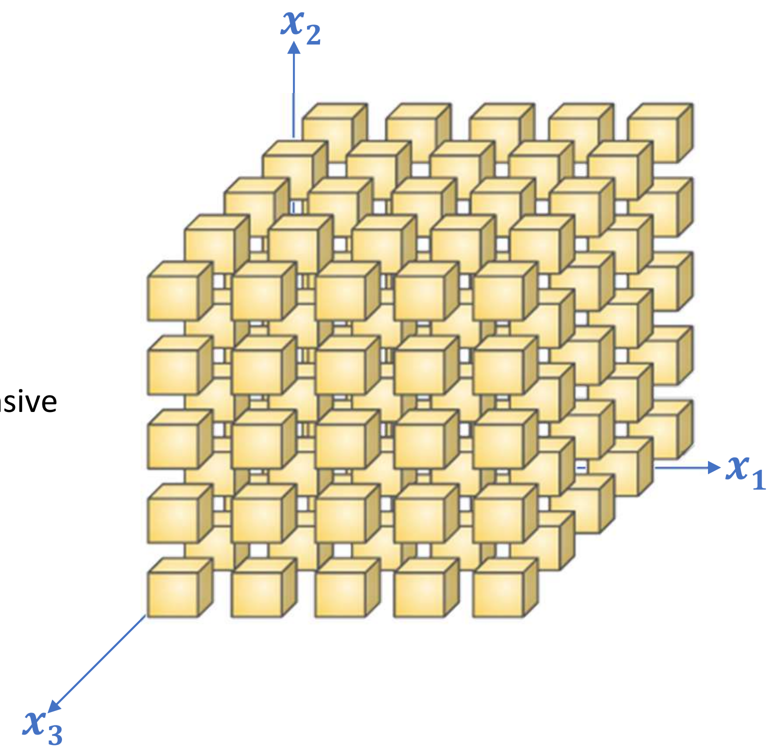
Curse of Dimensionality

As dimensions increase, the data we need to generalize grows exponentially

Curse of dimensionality

- The Iris data set has 150 instances in 4-dimensions: that is ~3.5 values per dimension!

- Labeled data is hard to get and expensive (about $2/instance on average for outsourced labeling services)



Source: therbootcamp.github.io