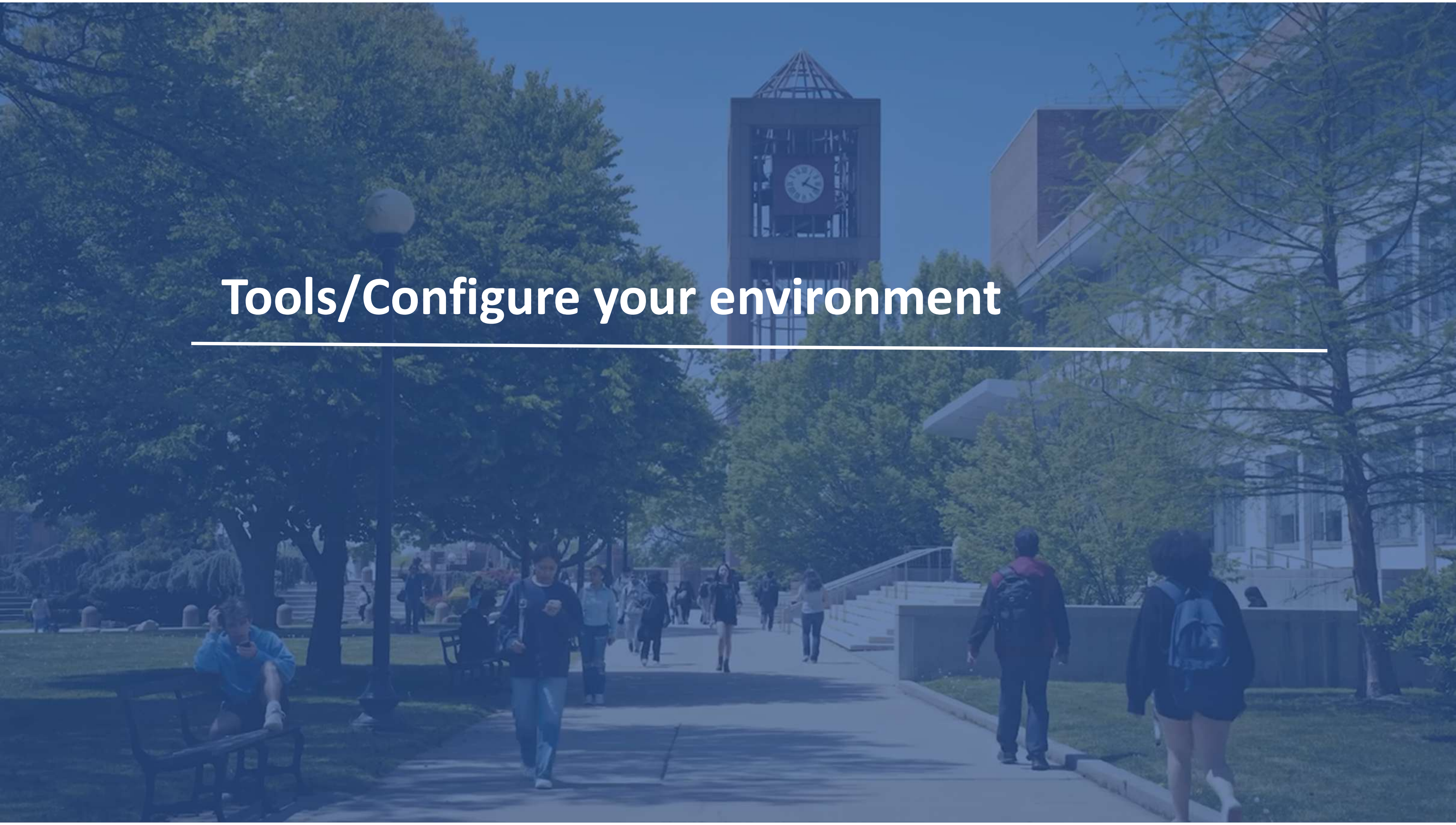
Introduction to Machine Learning (GAI 601)

WEEK 3

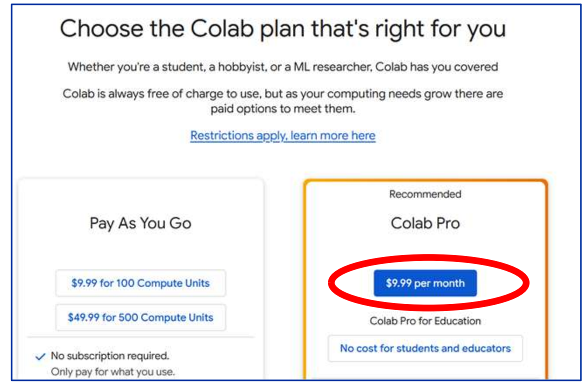
Tools/Configure your environment



Register for free tools

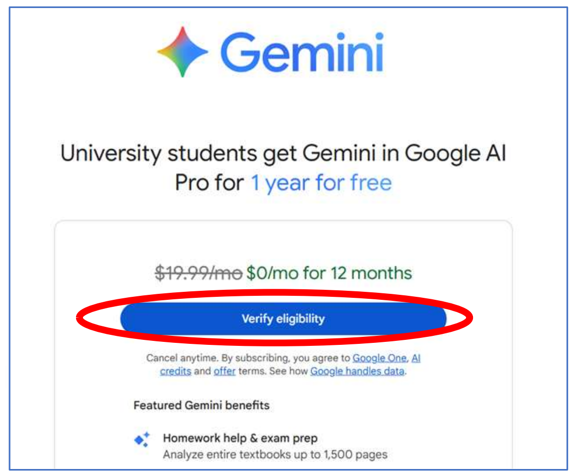
1. Colab

- Go to https://colab.research.google.com/signup
- Create a Google account (if you don't have one)
- Sign up for **Colab Pro for Education** (free for students)



2. GitHub Copilot

- Go to https://github.com/education
- Click on **Join GitHub Education**
- Create a GitHub account (if you don't have one)
- Click on Start an Application to get access



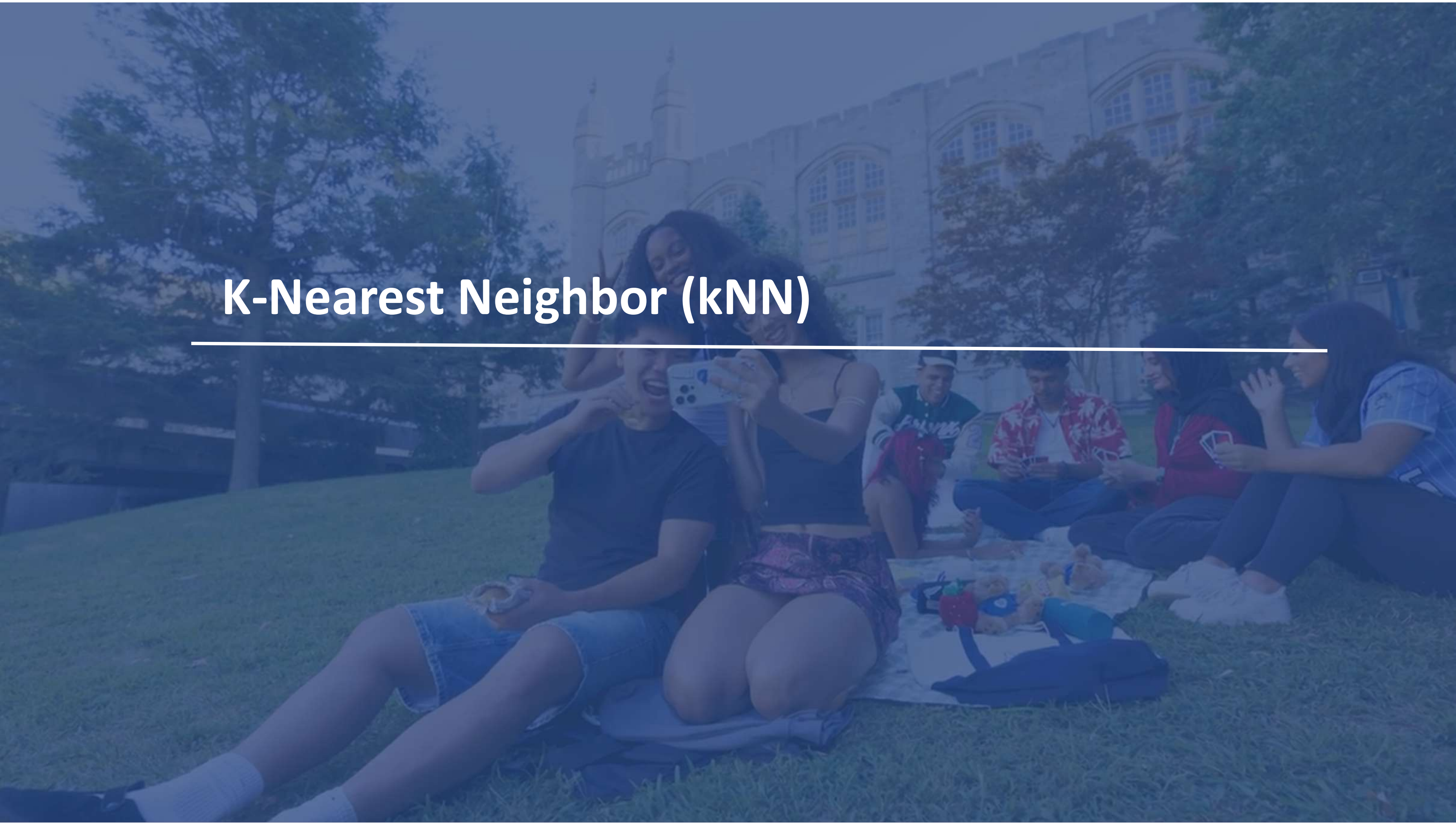
3. Google Gemini

- Go to https://gemini.google/us/students
- Fill out the verification form

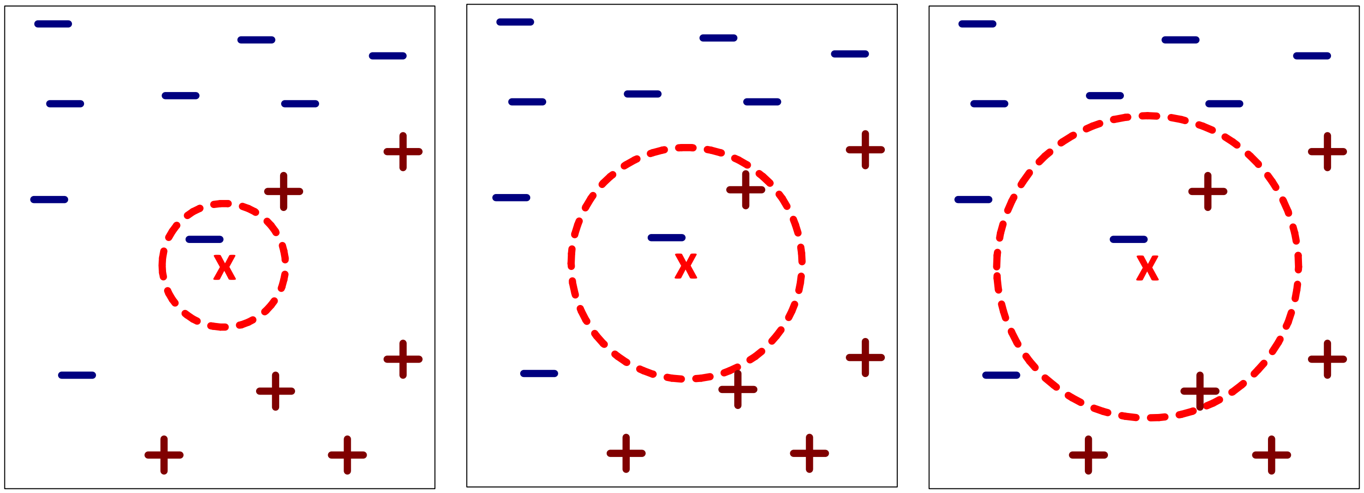
4. Microsoft Copilot 365

- Free with you Office 365 access (via CUNY)

K-Nearest Neighbor (kNN)



K-Nearest Neighbor (kNN)



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

Source: Supervised Learning Algorithms: A Comparison, November 2020Kristu Jayanti Journal of Computational Sciences (KJCS) - DOI:10.59176/kjcs.v1i1.1259. Free access

Identifying k

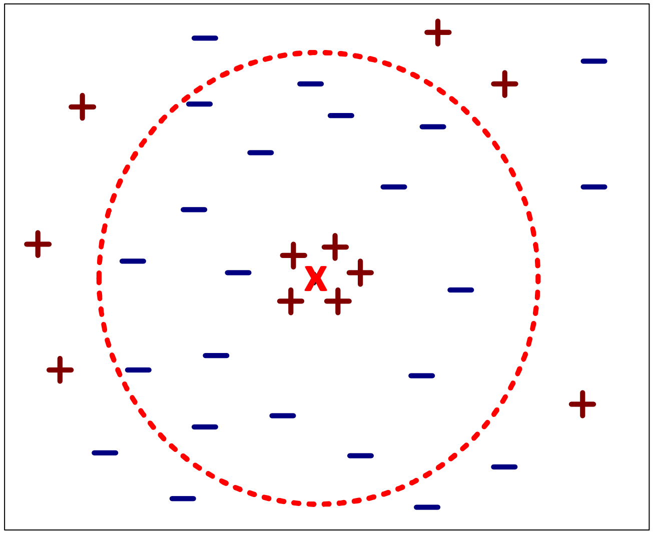
Choosing the value of k:

- **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



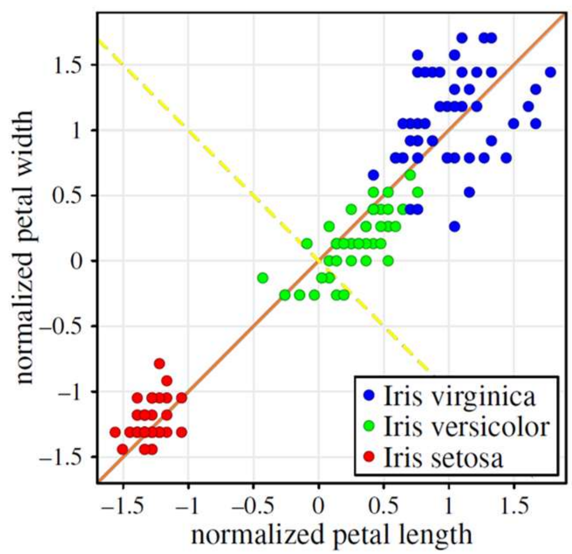
Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

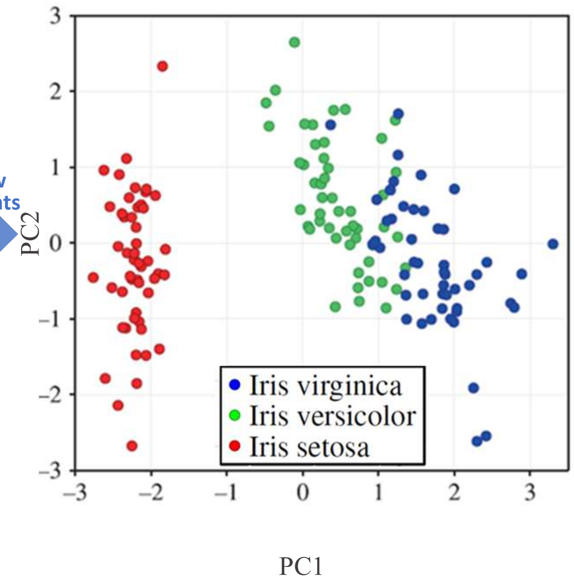
- **Dimensionality reduction technique**
- **Project data from the high-dimensional space to a lower-dimensional space**
- **Criteria: Maximize data variance to construct principal components**

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

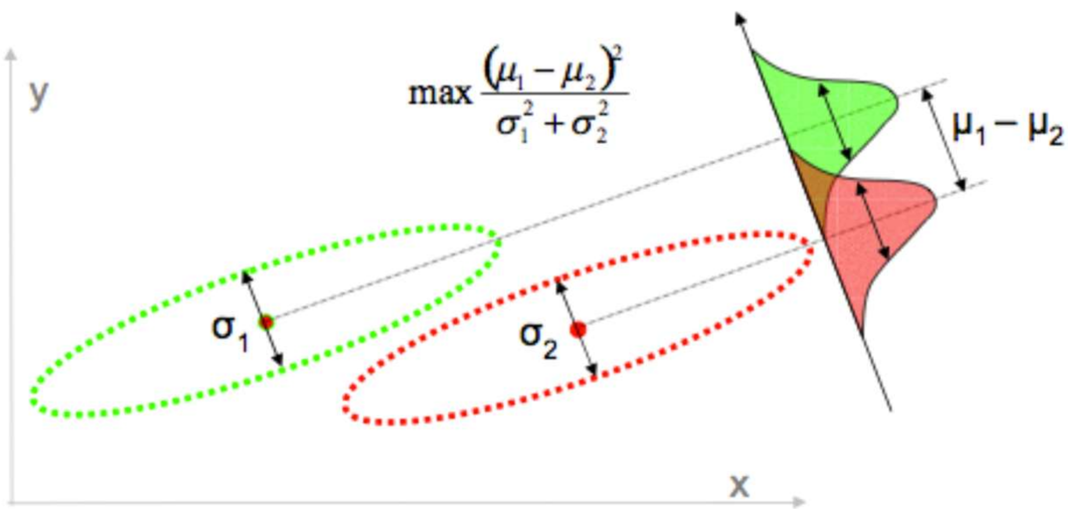


Source: Guide to Intelligent Data Science, Berthold et al

Linear Discriminant Analysis (LDA)

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.

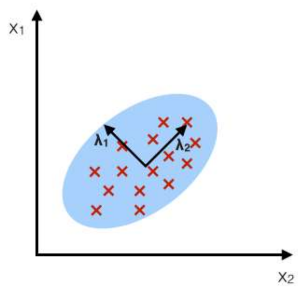


Source: Victor Lavrenko

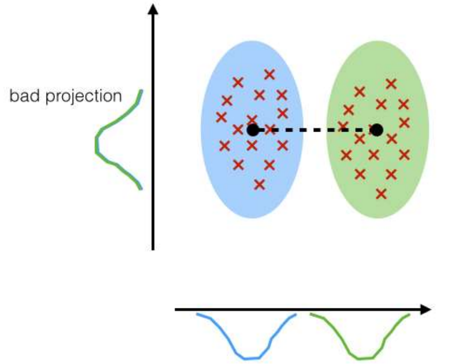
Linear Discriminant Analysis (LDA)

- Discriminants (LDA) maximize the separation ` 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

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation

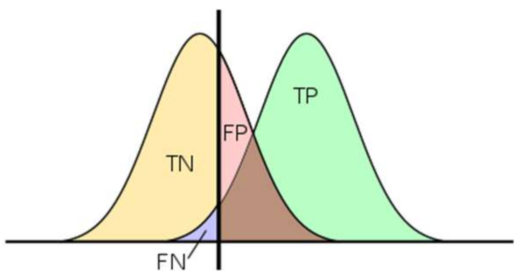


Source: Victor Lavrenko

Classification Metrics

Confusion Matrices

- **Accuracy:** Accuracy is the proportion of all classifications that were correct, whether positive or negative
- **Recall:** Recall is a measure of how many positives your model is able to recall from the data.
- **Precision:** Precision is the ratio of correct positive predictions to the total positive predictions.
- **F1 Score:** F1 score metric is used when you seek a balance between precision and recall.



ACTUAL VALUES	POSITIVE	NEGATIVE
POSITIVE	TP	FN
NEGATIVE	FP	TN

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

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

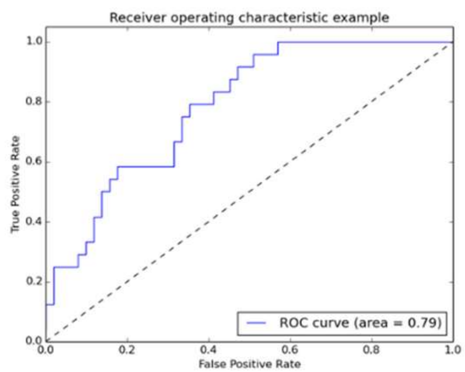
$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

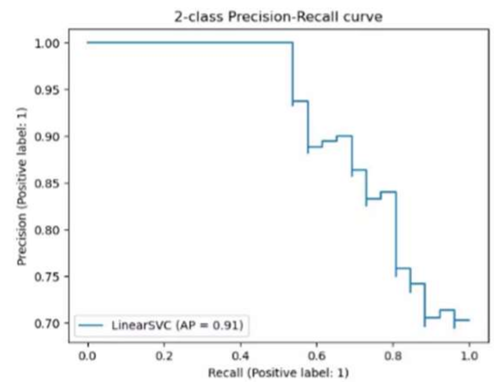
Source: Alleviating Class-Imbalance Data of Semiconductor Equipment Anomaly Detection Study,

ROC & AUC

- **Area under the Curve (AUC):** The AUC represents the probability that the model, if given a randomly chosen positive and negative example, will rank the positive higher than the negative.
- **Receiver-operating characteristic curve (ROC):** A ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR). The ROC curve is drawn by calculating the true positive rate (TPR) and false positive rate (FPR), and then graphing TPR over FPR.

Receiver-operating characteristic curve (ROC):





Both Precision-Recall Curve and ROC-AUC curve are used:

- To explain model goodness of fit
- To identify the correct threshold to map probabilities value to the actual classes 0/1

When to use which one:

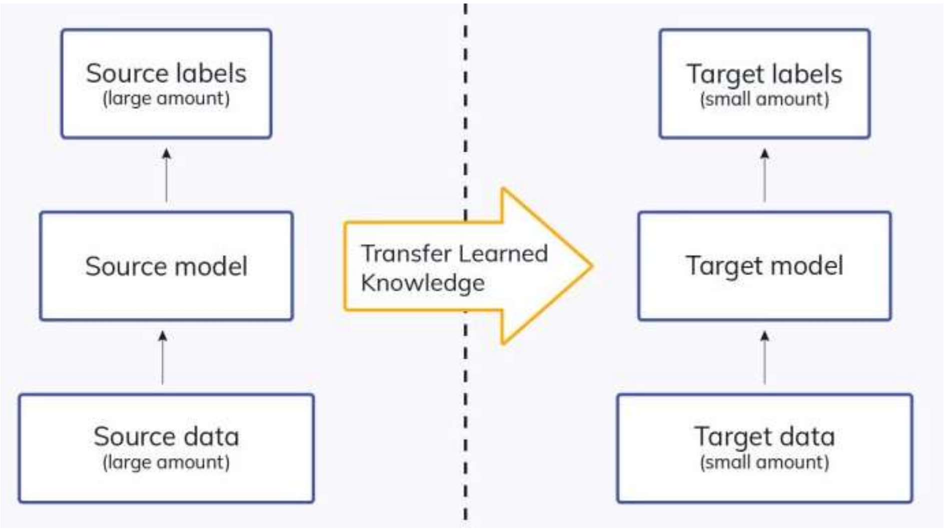
- Precision Recall curve is used when there is imbalance class distribution.
- ROC-AUC curve is used when there is balanced class distribution in data.

Source: https://ashutoshtripathy.com

Transfer Learning

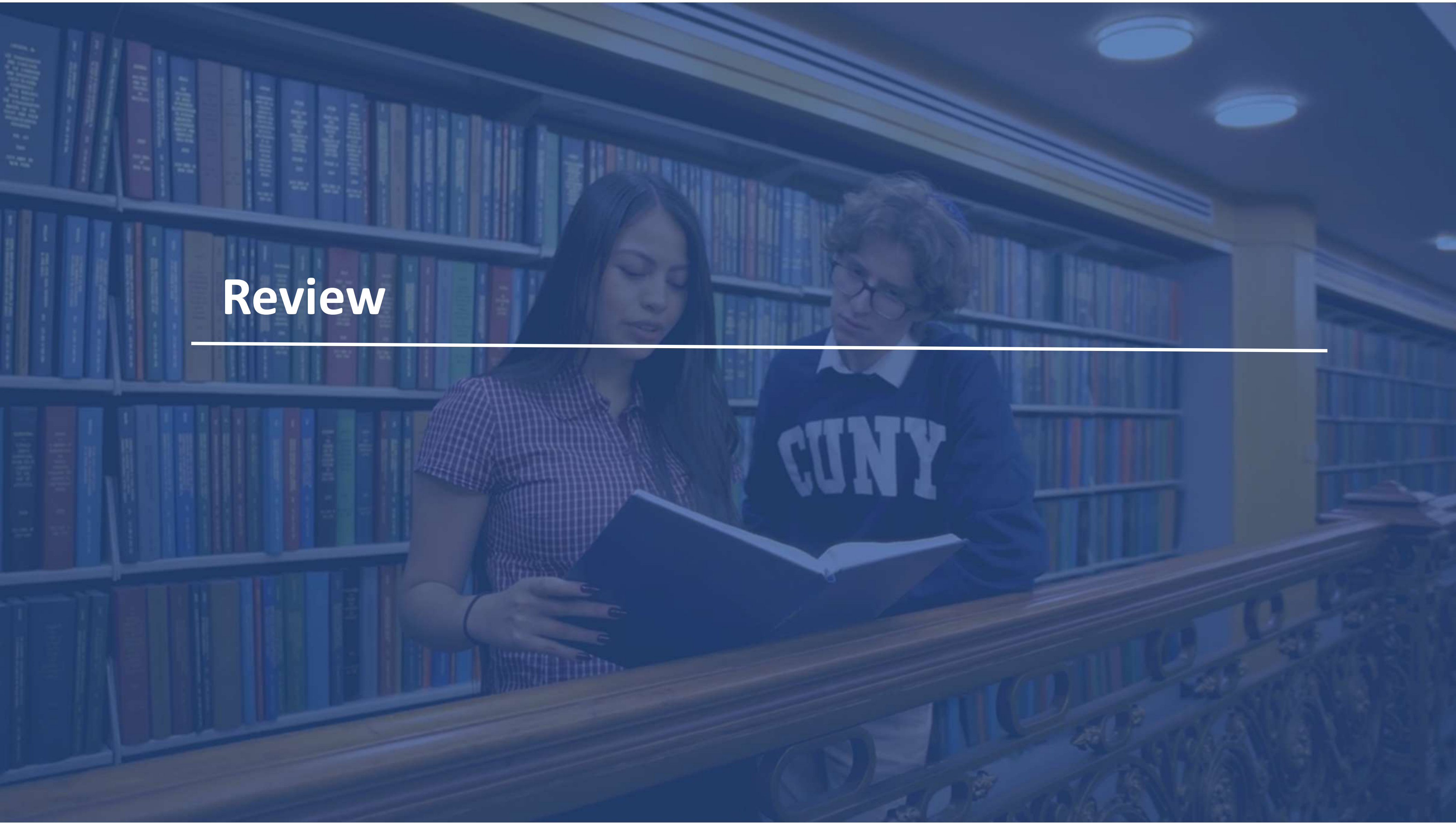
Transfer Learning

- **Transfer learning** is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second task. It allows training to be "jump-started".
- For example, a model trained for self-driving for cars may be useful for self-driving trucks
- **Benefits:**
    1. Reduced data requirements
    2. Faster training
    3. Reduce compute & lower costs



Source: v7labs

Review

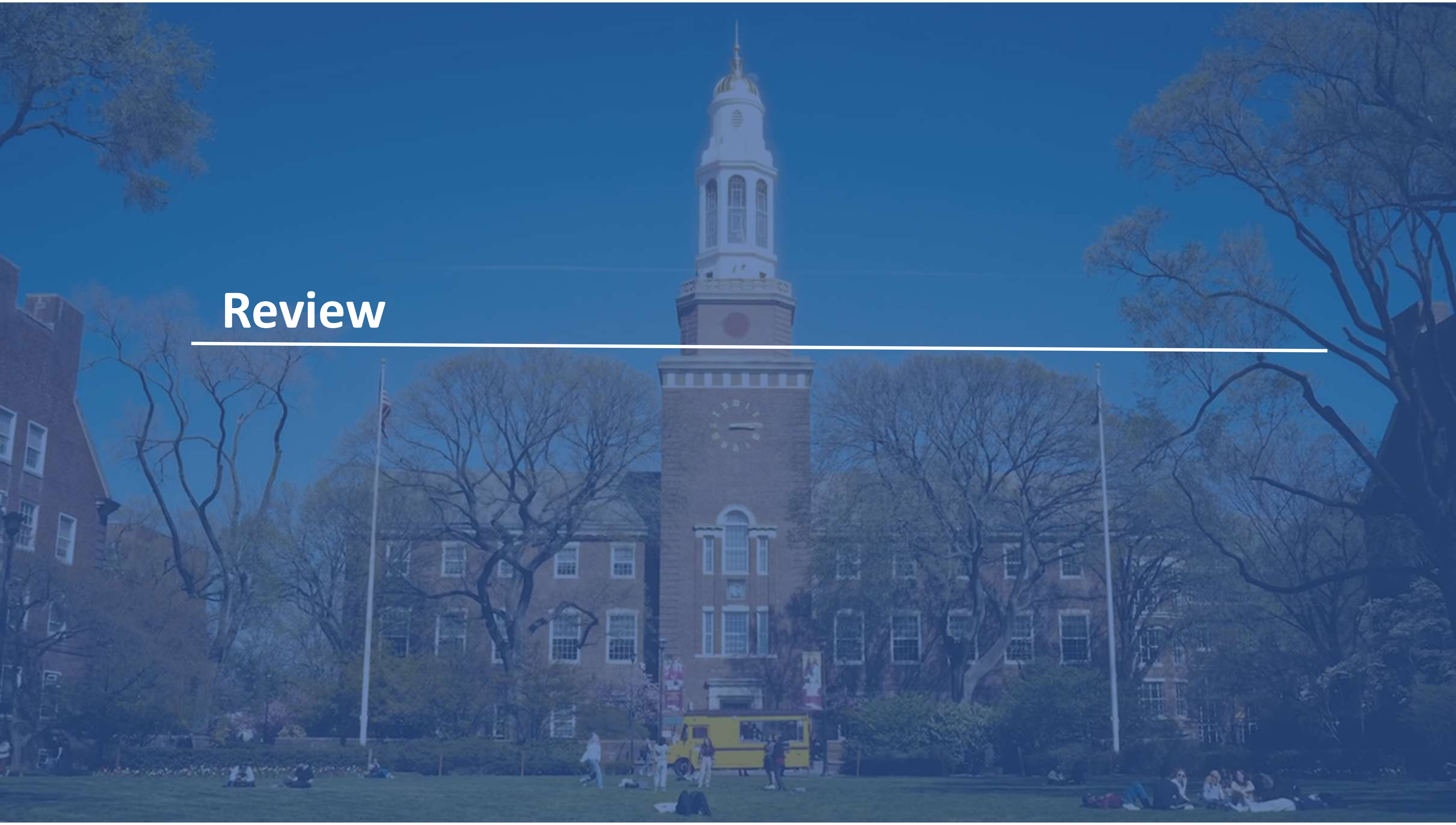


This week we covered

Lesson Objectives/Topics

1. Apply discriminant analysis and k-NN to classify observations in structured datasets.
2. Explain when to choose k-NN over other classification methods in business problems.
3. Prepare and normalize data for non-parametric classification models.
4. Compare classifier accuracy using confusion matrices and cross-validation.

Review



Training Data: Iris data set

Features (inputs) | **Labels (target outputs)**

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

y_1

Iris Setosa

y_2

Iris Versicolor

y_3

Iris Virginica

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

There are 3 potential labels (outputs): y_1, y_2, & y_3

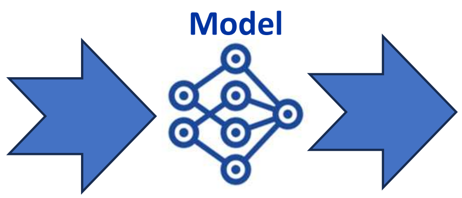
Source: Iris data set, Fisher, et al

Label Space

The trained model will map the Feature space to the Label (target output) space:

Feature space (inputs)

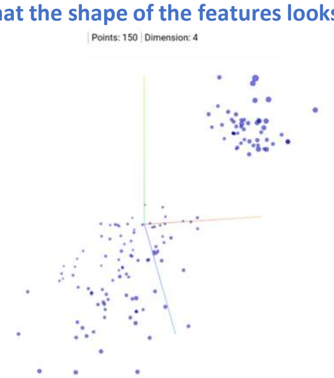
Sepal Length (cm)	Sepal width (cm)	Petal legth (cm)	Petal width (cm)
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.2
5.4	3.9	1.7	0.4
• • •			
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.5	2.8	4.6	1.5
5.7	2.8	4.5	1.3
6.3	3.3	4.7	1.6
• • •			
6.3	3.3	6	2.5
5.8	2.7	5.1	1.9
7.1	3	5.9	2.1
6.3	2.9	5.6	1.8
6.5	3	5.8	2.2
7.6	3	6.6	2.1



Labels (target outputs)

Class
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-setosa
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-versicolor
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica
Iris-virginica

What the shape of the features looks like:



What does the label (target output) space look like?

?

Source: Iris data set, Fisher, et al

One-hot encoding

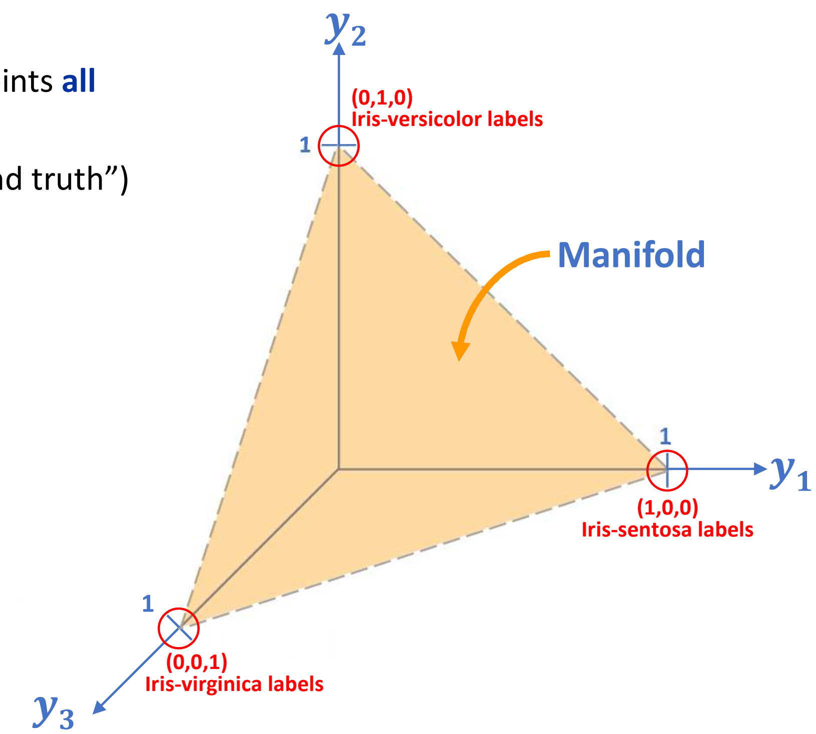
- ML requires numbers: labels must be converted to numbers
- Each class (type of label must be its own dimension)
- The value in each dimension conveys the probability it is of that class
- Training Data Labels always have a probability of 1 (100%) i.e. they are the “Ground Truth”

One-Hot Encoding — **Labels**

Class	One-Hot Encoding	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

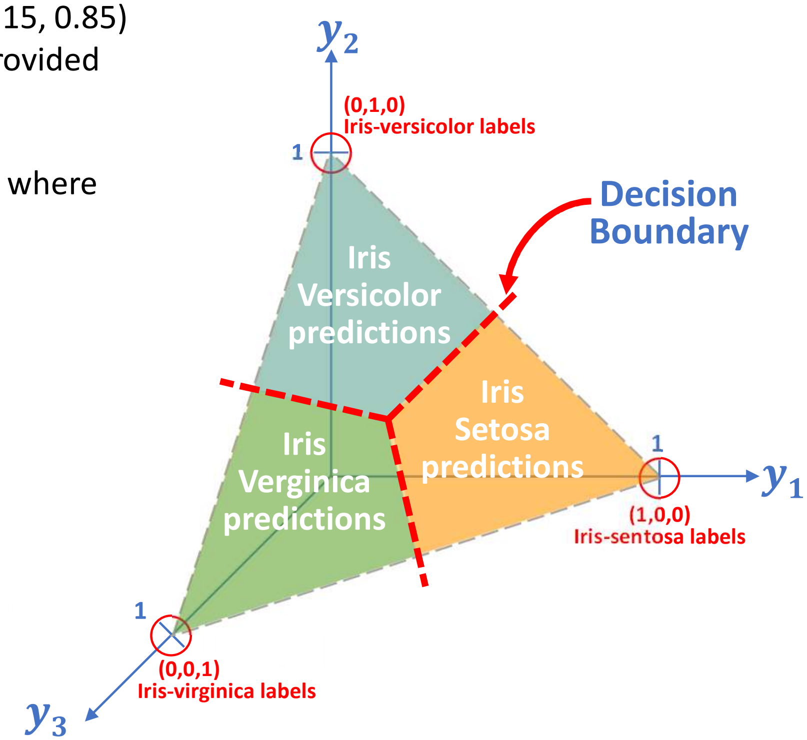
One-hot encoding

- 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 **all points must lie on a manifold**
- Only labels have values of 1 (as they are the "ground truth")



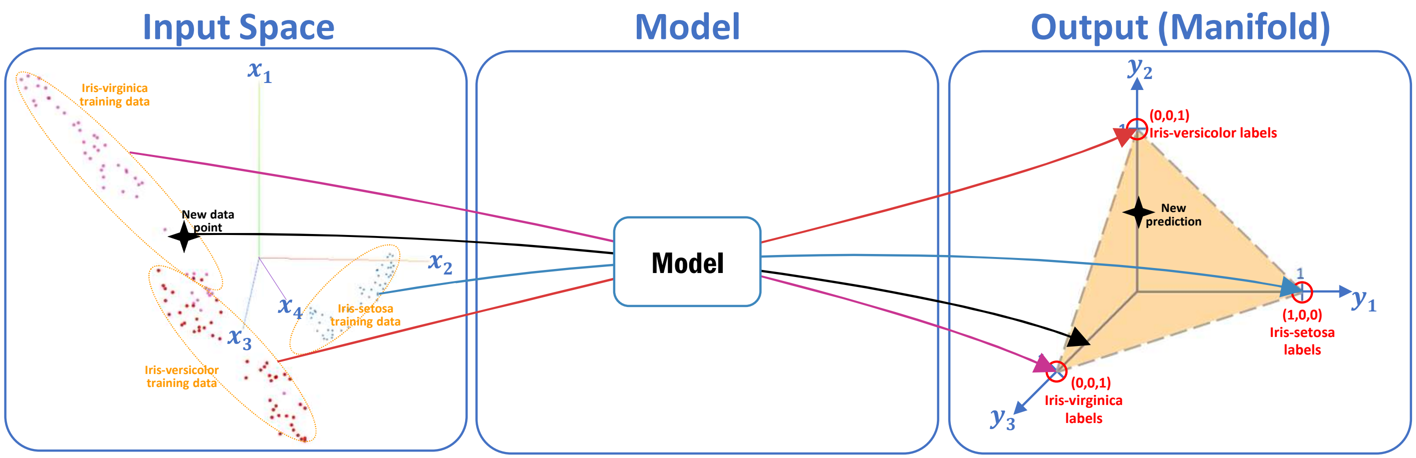
Decision Boundary

- The **prediction generates a probability for each class**, which must add to 1 (100%)
- For example, a prediction that (y_1, y_2, y_3)=(0.1, 0215, 0.85) means that the model predicts that the features provided have a 85% probability of being Iris Virginica.
- A **decision boundary separates the classes** – it lies where there is equal probability between classes.



Putting it all together

A predictive AI model is trained to map the Feature Space to the Target (Label) space.




CUNY
School of
Professional Studies