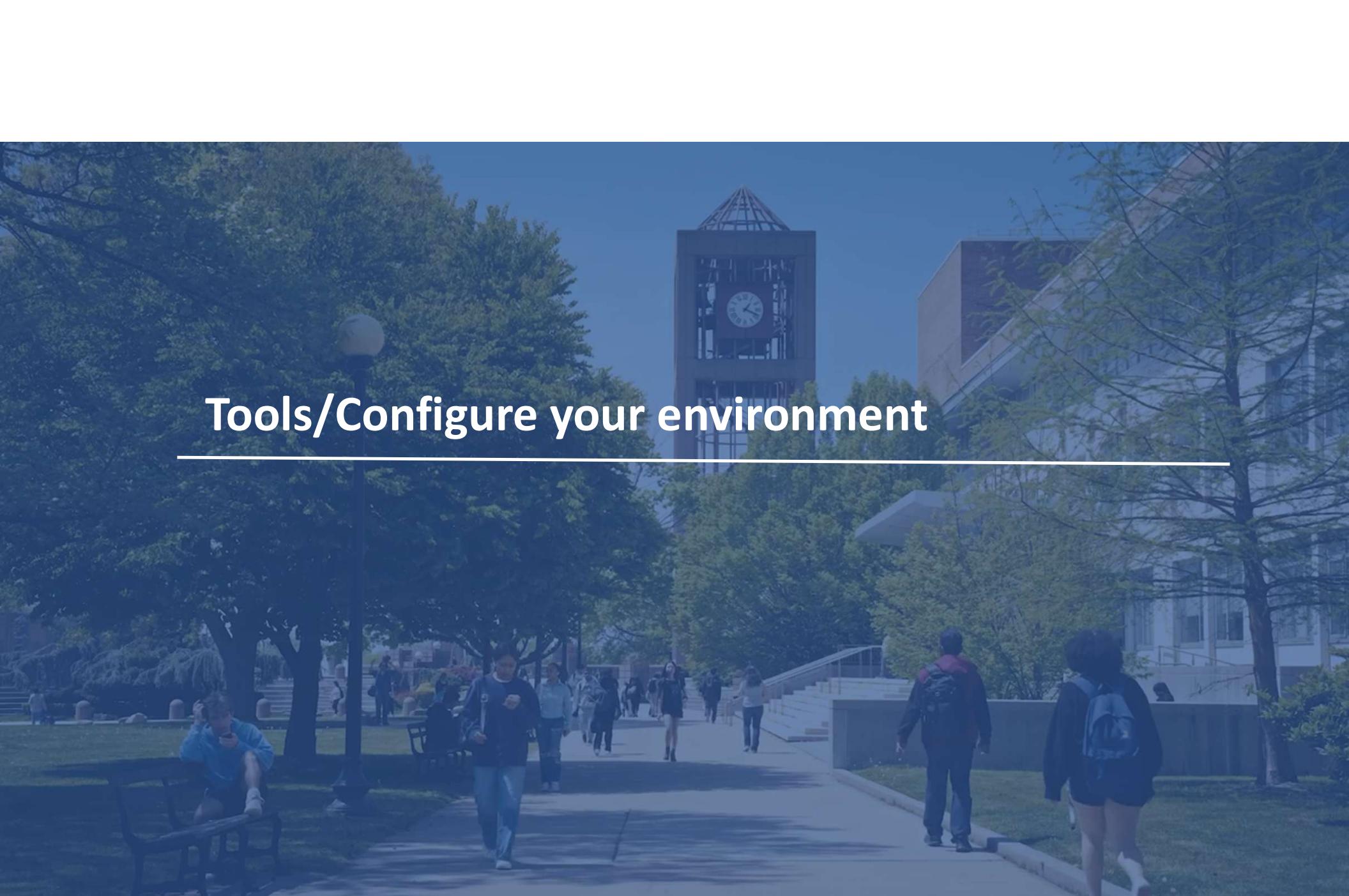


# Introduction to Machine Learning (GAI 601)

## WEEK 3



Tools/Configure your environment

---

# Register for free tools

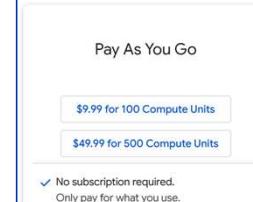
## 1. Colab

- 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)

### Choose the Colab plan that's right for you

Whether you're a student, a hobbyist, or a ML researcher, Colab has you covered. Colab is always free of charge to use, but as your computing needs grow there are paid options to meet them.

[Restrictions apply, learn more here](#)



## 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



University students get Gemini in Google AI Pro for **1 year for free**

\$19.99/mo \$0/mo for 12 months

[Verify eligibility](#)

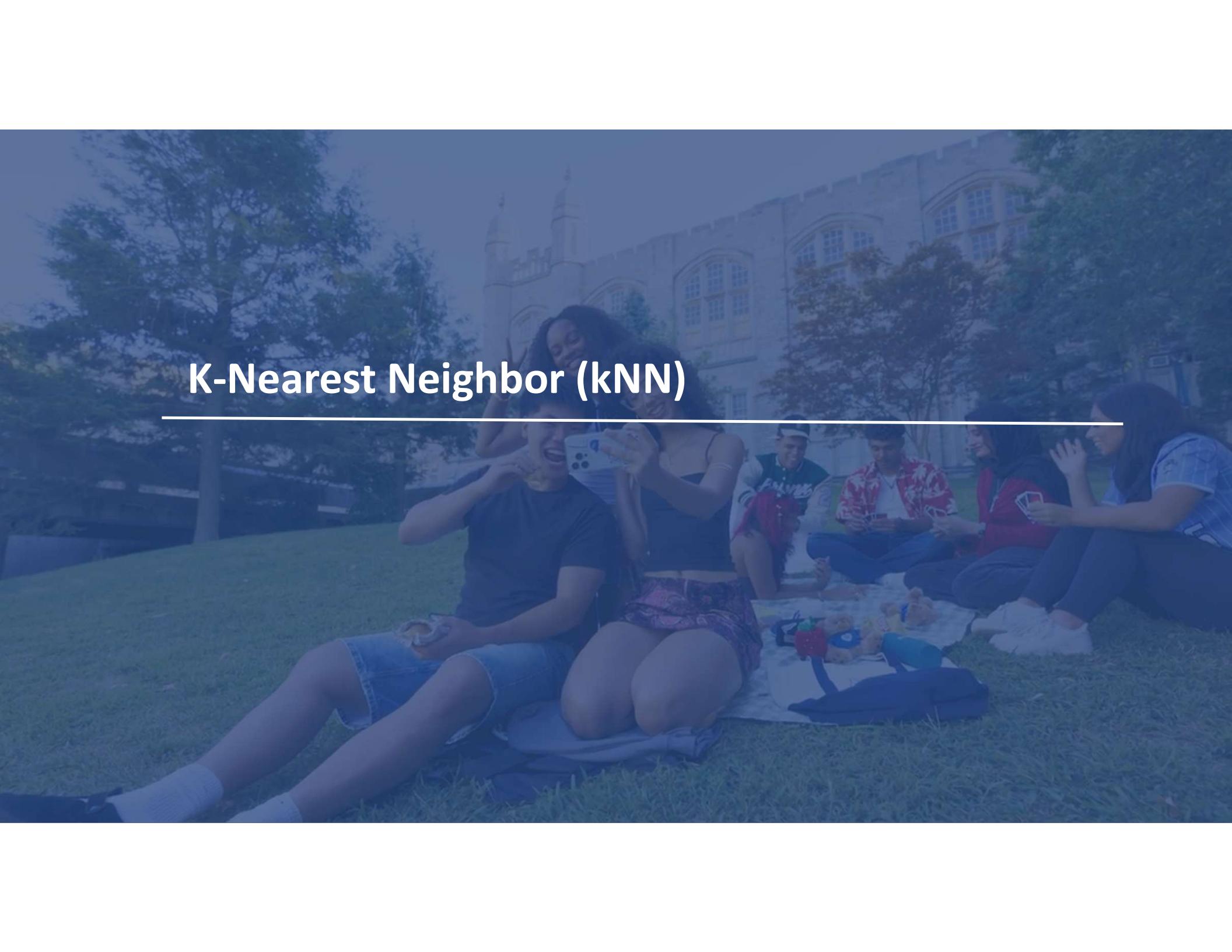
Cancel anytime. By subscribing, you agree to [Google One](#), [AI credits](#) and [offer terms](#). See how [Google handles data](#).

#### Featured Gemini benefits

- Homework help & exam prep  
Analyze entire textbooks up to 1,500 pages

## 4. Microsoft Copilot 365

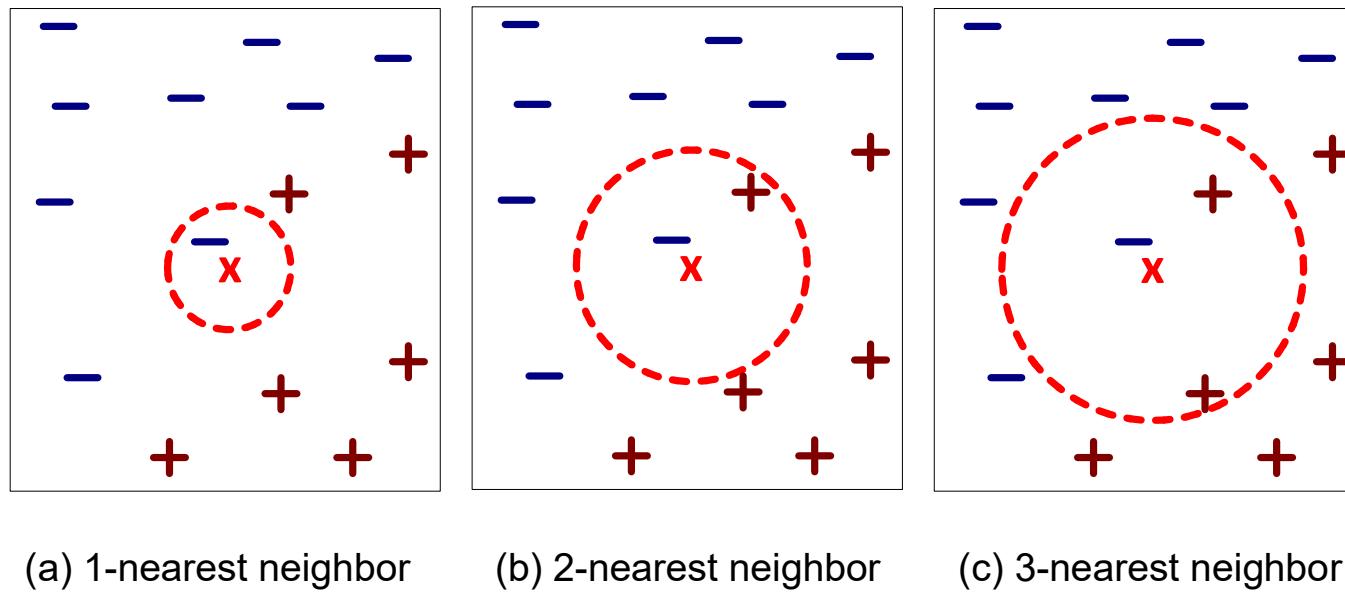
- Free with your Office 365 access (via CUNY)

A photograph of a group of approximately ten young adults sitting on a blue and white checkered picnic blanket on a grassy lawn. They are all looking at their smartphones, which are held up in front of them. Some are laughing and smiling. In the background, there is a large, light-colored stone building with multiple windows and a prominent gabled roofline. Large trees are visible behind the building and in the foreground.

# 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 2020 Kristu Jayanti Journal of Computational Sciences (KJCS) - DOI:10.59176/kjcs.v1i1.1259. Free access

# Identifying k

---

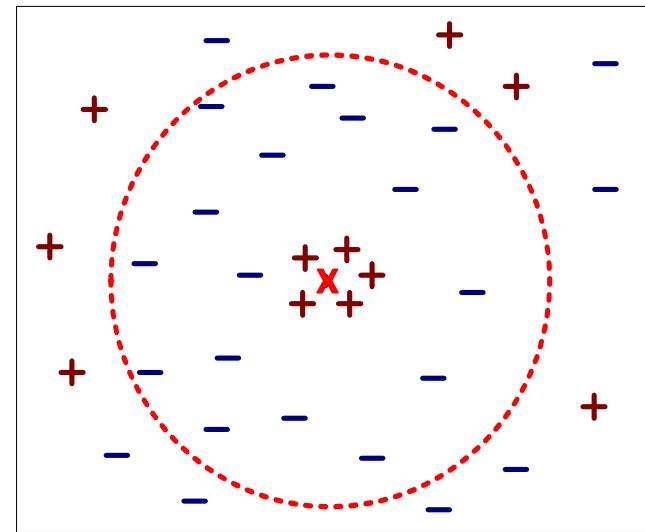
Choosing the value of k:

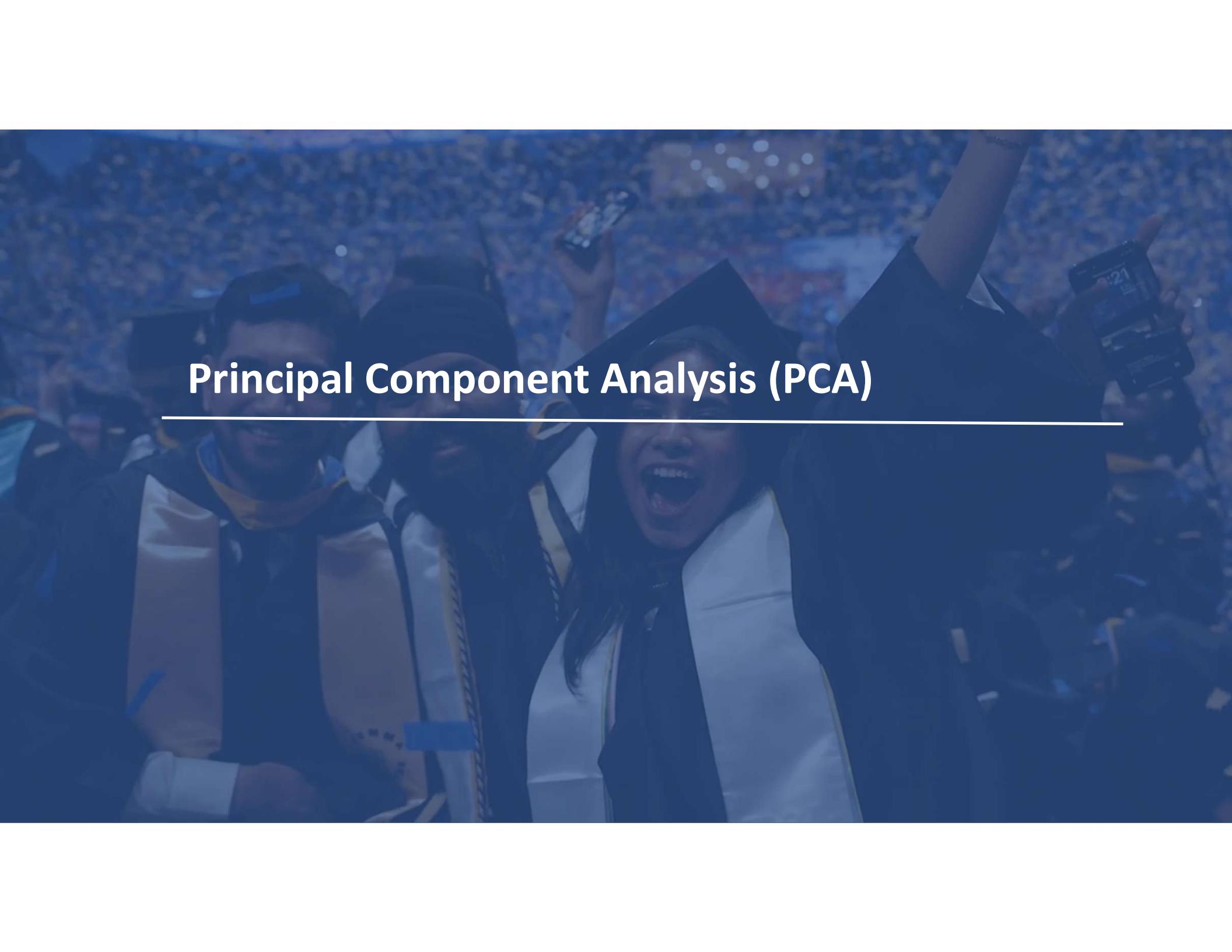
- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other

Rule of thumb:

$$k = \sqrt{N}$$

N: number of training points



A large, dense crowd of people is gathered in a stadium, filling the background. In the foreground, several individuals are visible, their faces illuminated by the screens of their smartphones. One person in the center-right is shouting or cheering with their mouth wide open. Another person to the left is wearing a blue and yellow scarf. The overall atmosphere is energetic and suggests a major event or game is taking place.

# Principal Component Analysis (PCA)

---

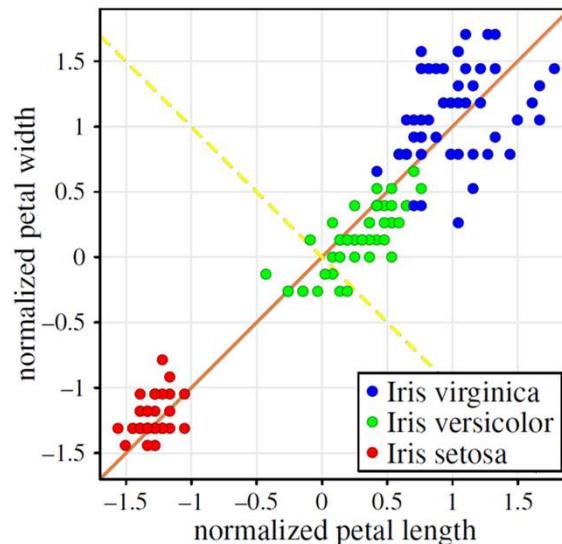
# Principal Component Analysis (PCA)

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

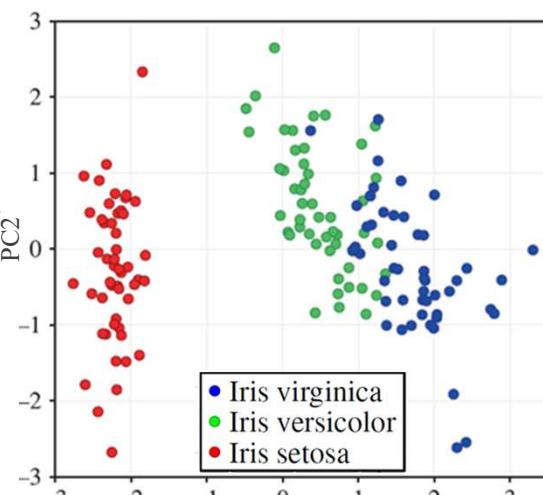
Steps:

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



PC1

Source: Guide to Intelligent Data Science, Berthold et al

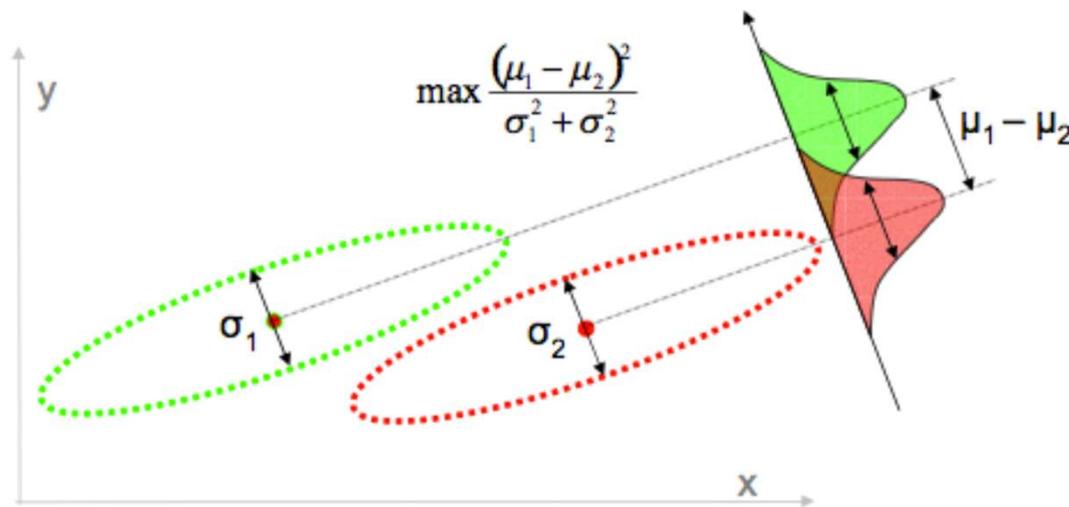
A blue-tinted photograph of a university campus. In the background, a tall, modern clock tower with a glass facade and a geometric, pyramid-like roof stands prominently. To the right, a large, light-colored building with a glass-enclosed staircase is visible. In the foreground, several students are walking along a paved path. One student in a blue hoodie is sitting on a bench on the left. The scene is set against a clear blue sky.

# 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.



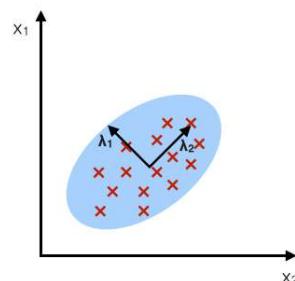
Source: Victor Lavrenko

# Linear Discriminant Analysis (LDA)

- Discriminants (LDA) maximize the separation of 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

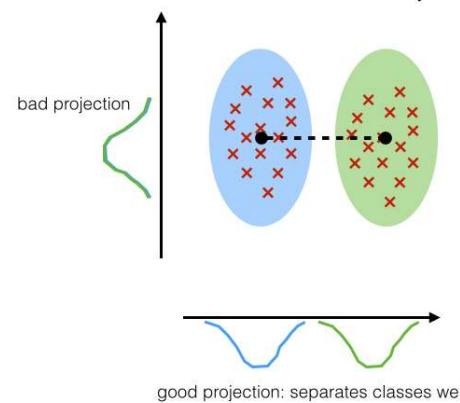
## PCA:

component axes that maximize the variance



## LDA:

maximizing the component axes for class-separation



Source: Victor Lavrenko

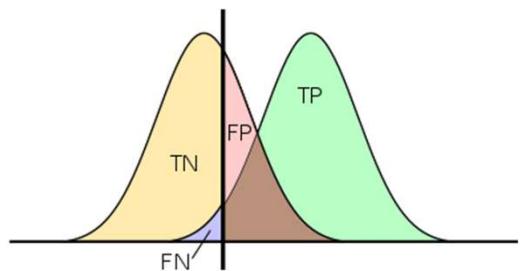
A photograph of a group of students sitting on bleachers in a large stadium or arena. They are looking down at their mobile devices. The background shows a city skyline through large windows.

# 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.



		POSITIVE	NEGATIVE
ACTUAL VALUES	POSITIVE	TP	FN
	NEGATIVE	FP	TN

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

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

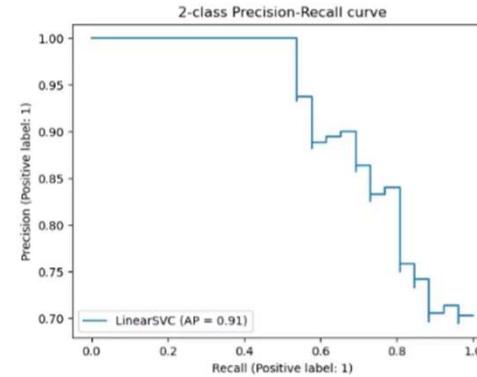
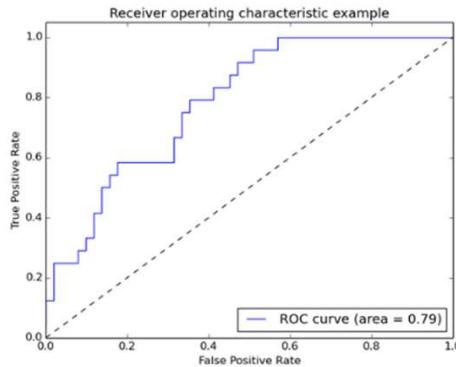
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

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):**



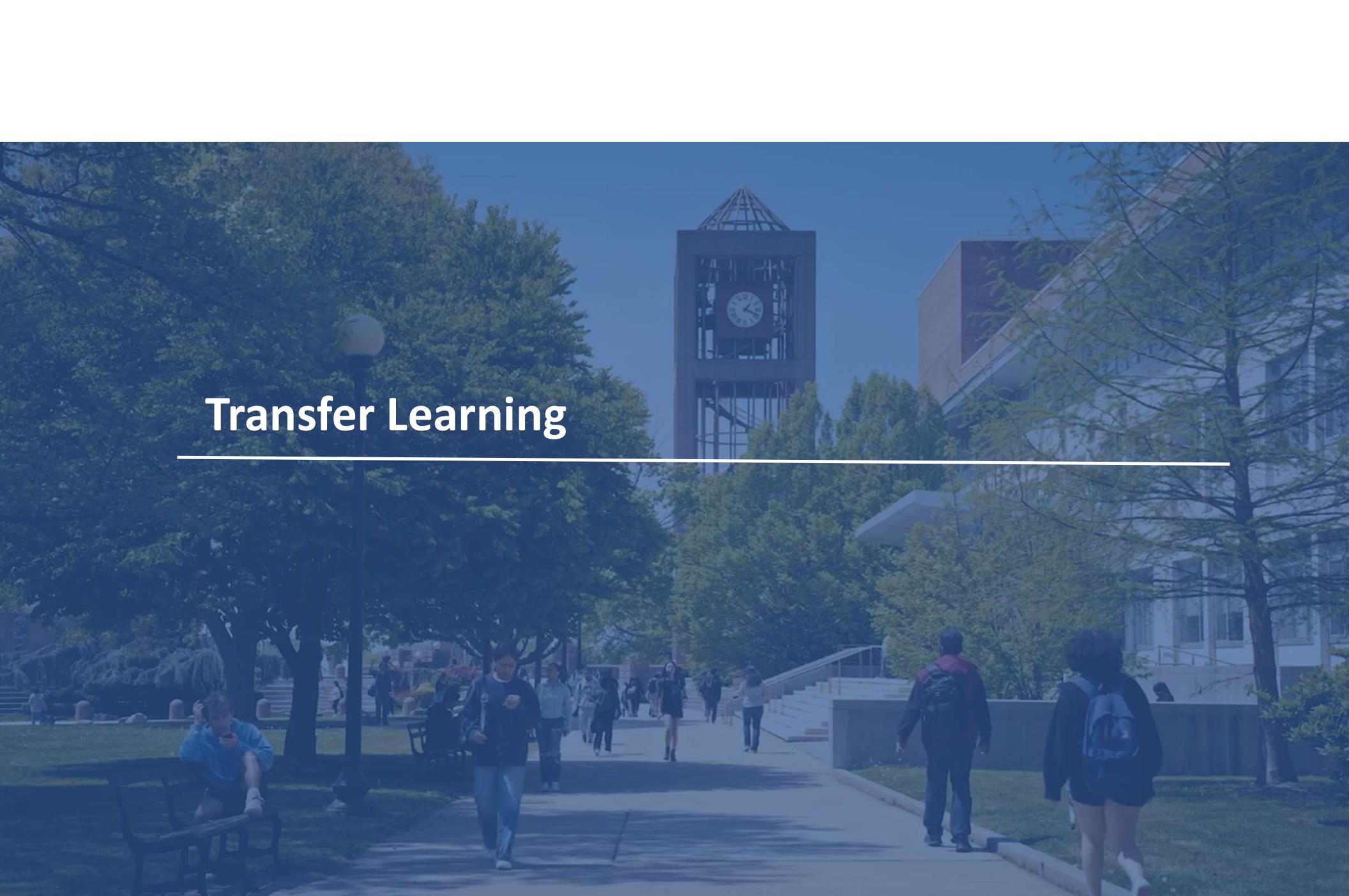
**Both Precision-Recall Curve and ROC-AUC curve are used:**

- 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>

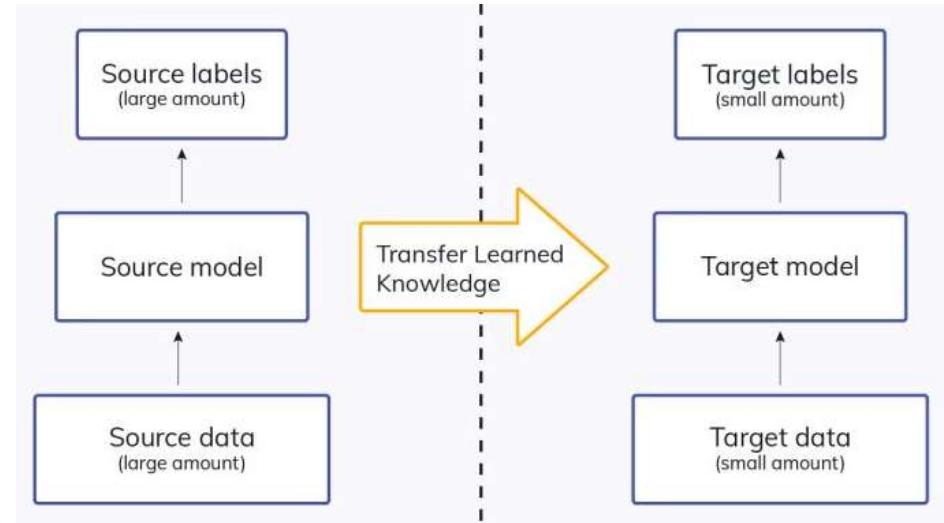


# Transfer Learning

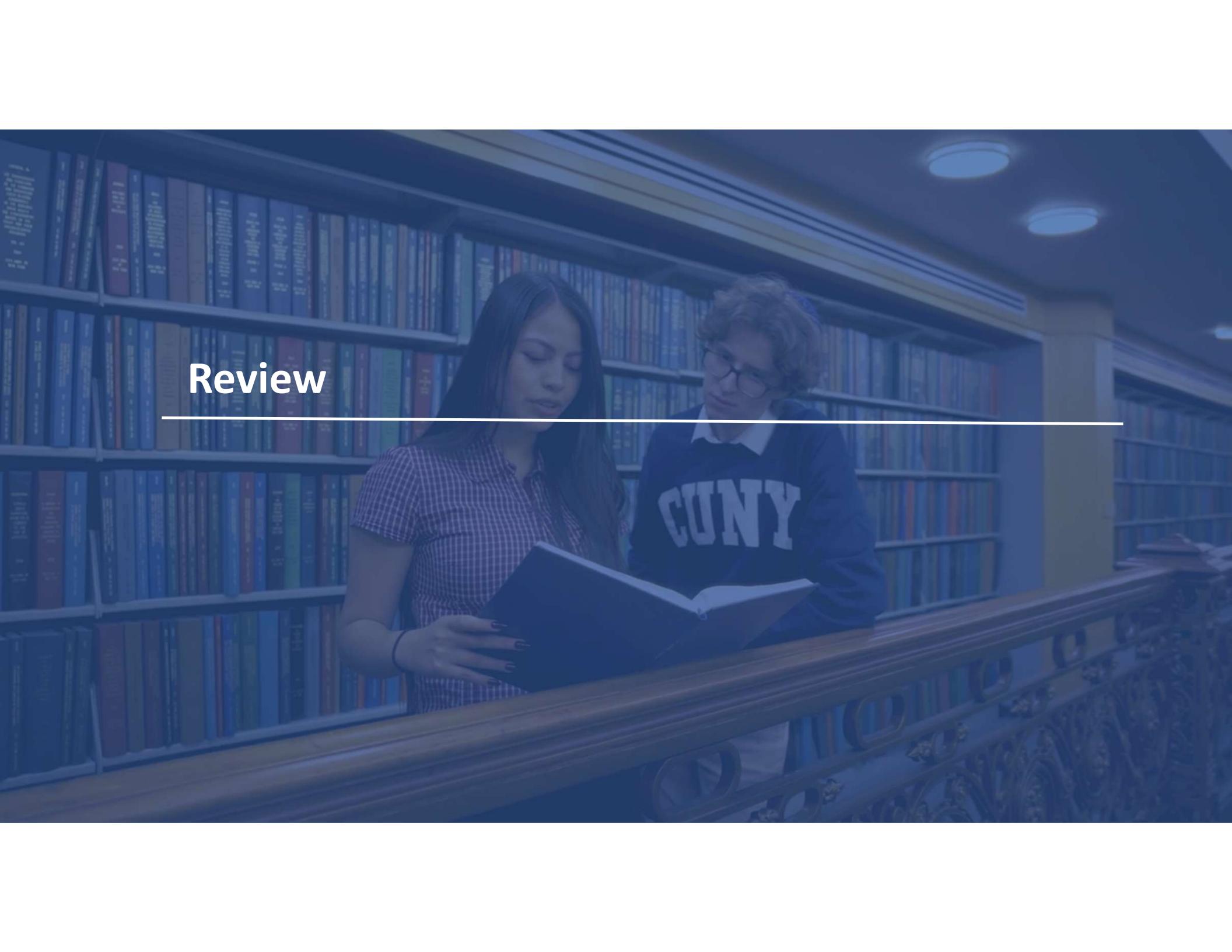
---

# 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

A photograph of a man and a woman in a library. The woman, on the left, has long dark hair and is wearing a white and grey checkered shirt. She is looking down at an open book she is holding. The man, on the right, has short blonde hair and is wearing glasses and a dark blue sweatshirt with "CUNY" printed in white. He is also looking at the book. They are standing in front of tall bookshelves filled with books. The lighting is soft, and the overall atmosphere is quiet and studious.

# Review

---

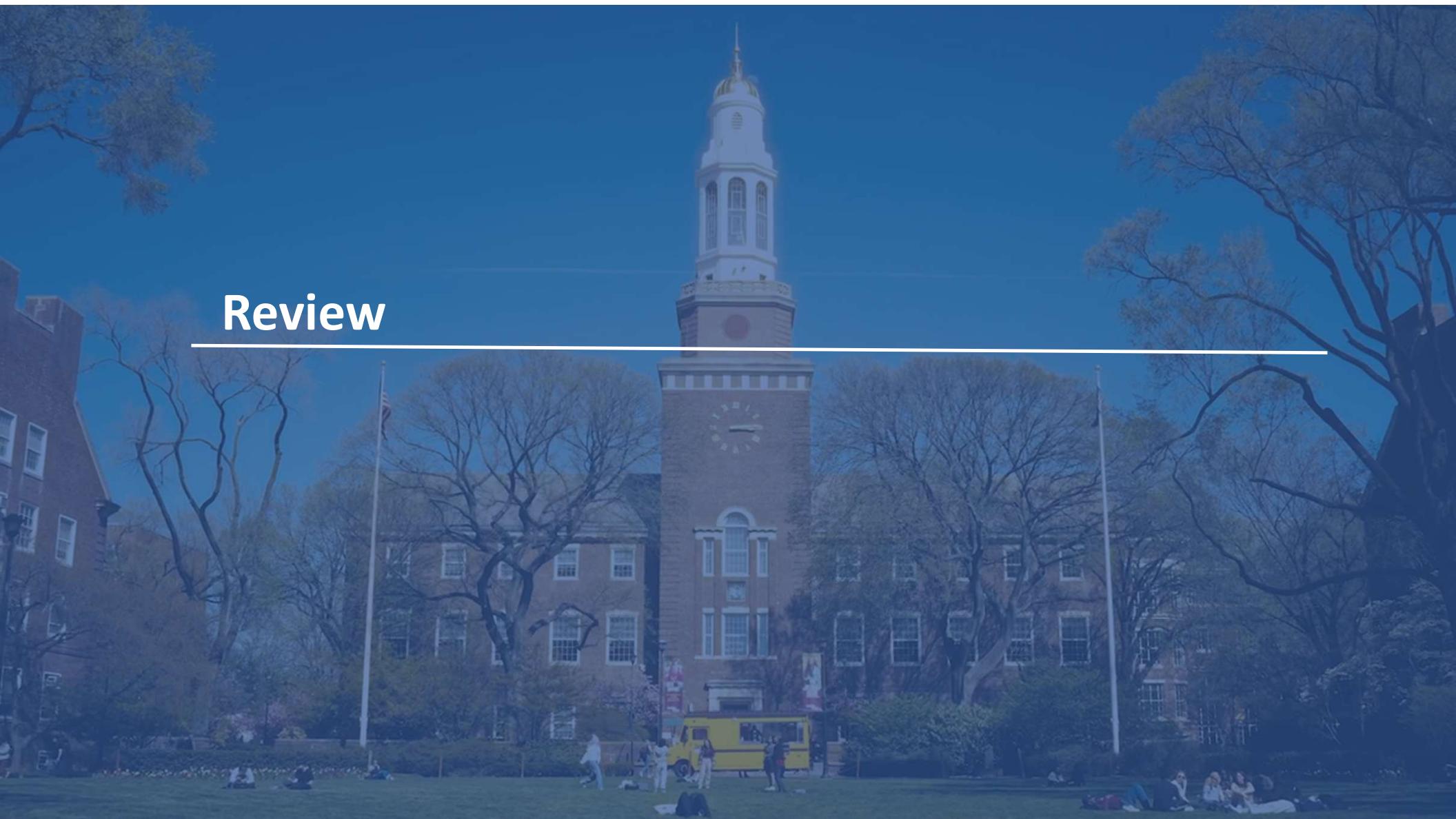
# This week we covered

## **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

Instance	Features (inputs)				Class	Labels (target outputs)	
	Sepal Length (cm)	Sepal width (cm)	Petal length (cm)	Petal width (cm)			
0	5.1	3.5	1.4	0.2	Iris-setosa		$y_1$
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		$y_2$
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		$y_3$
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		

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

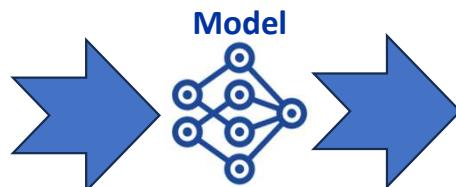
The trained model will map the Feature space to the Label (target output) space:

Feature space (inputs)

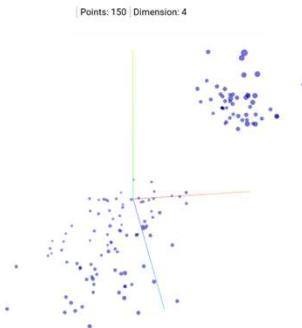
Sepal Length (cm)	Sepal width (cm)	Petal length (cm)	Petal width (cm)
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5	3.6	1.4	0.2
5.4	3.9	1.7	0.4
...	...	...	...
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.5	2.8	4.6	1.5
5.7	2.8	4.5	1.3
6.3	3.3	4.7	1.6
...	...	...	...
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-versicolor
Iris-virginica



What the shape of the features looks like:



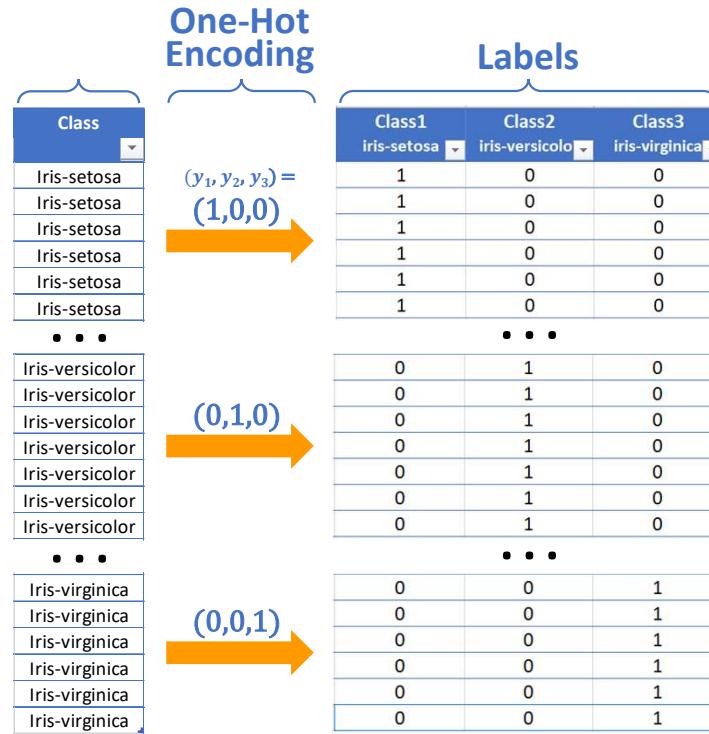
Source: Iris data set, Fisher, et al

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



# One-hot encoding

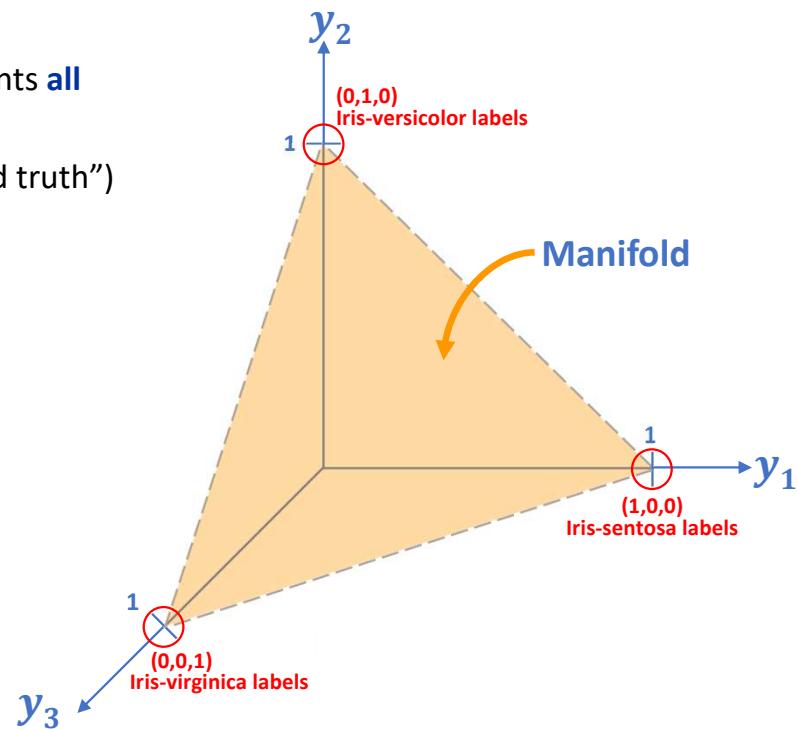
- ML requires numbers: labels must be converted to numbers
- Each class (type of label) must be its own dimension
- The value in each dimension conveys the probability it is of that class
- Training Data Labels always have a probability of 1 (100%) i.e. they are the “Ground Truth”



# One-hot encoding

---

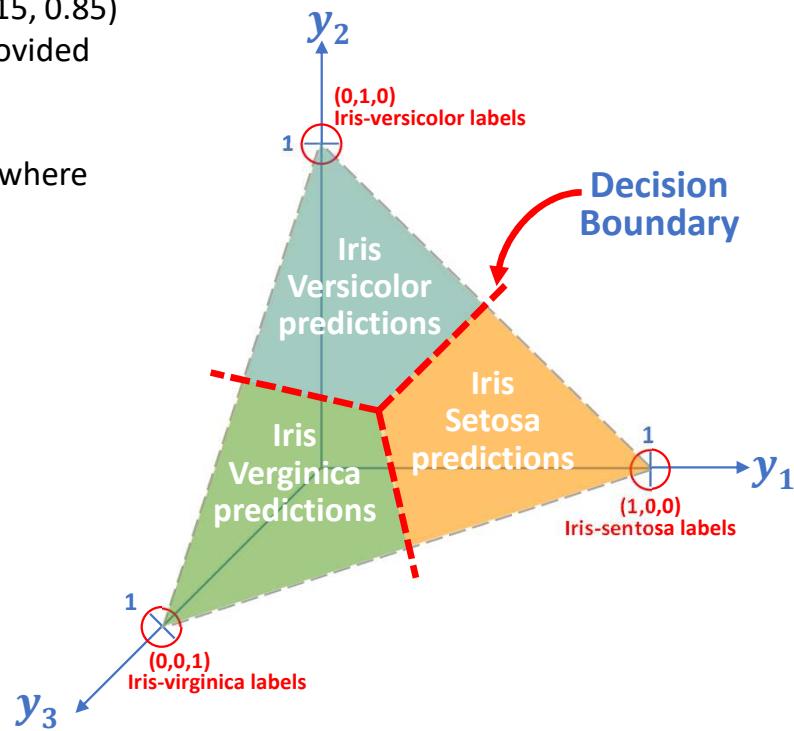
- The **number of dimensions = number of classes**.  
In this case 3 dimensions.
- A Label (or prediction) is one data-point in that 3-dimensional space
- Probabilities of all classes add up to 1 (100%) so points **all points must lie on a manifold**
- Only labels have values of 1 (as they are the “ground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



# Decision Boundary

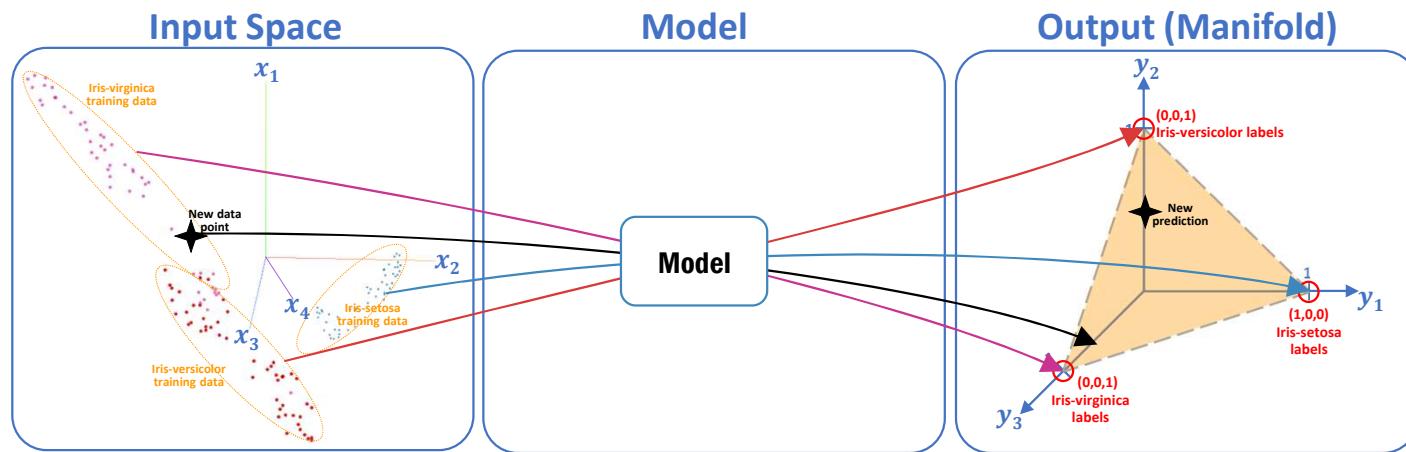
---

- The **prediction generates a probability for each class**, which must add to 1 (100%)
- For example, a prediction that  $(y_1, y_2, y_3) = (0.1, 0.215, 0.85)$  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.



The background image shows a panoramic view of the New York City skyline at sunset or sunrise. The Brooklyn Bridge is prominent in the lower-left foreground, stretching across the East River. The Manhattan skyline, with its numerous skyscrapers, rises in the background under a clear blue sky.

**CU** | School of  
**NY** | Professional Studies