Categorization with *k*-nearest-neighbors

CISC 3225
Spring 2024
DSFS 12

Machine learning crash course

- Definition: Creating and using *models* that are *learned from data*.
    - Sometimes called *predictive modeling* or *data mining* in different contexts
- Models can *predict* various outcomes from new data
    - Is an email spam or not?
    - Is a credit card transaction fraudulent?
    - Which advertisement is a shopper most likely to click on?
    - Which football team is going to win the Superbowl?
- Goal: **Use existing data to make predictions about previously unseen data**

Machine learning crash course

Two basic tasks:

1. **Classification**: Apply a label to data
   a. Spam detection
   b. Predict penguin species from physical characteristics
   c. Sentiment analysis: Is an article positive or negative?
2. **Regression**: Predict a continuous value from data
   a. Predict a movie rating (1-10) from its summary
   b. Predict someone's age from a photo of their face
   c. Predict tomorrow's temperature from a week of weather data

Machine learning crash course

Two basic approaches:

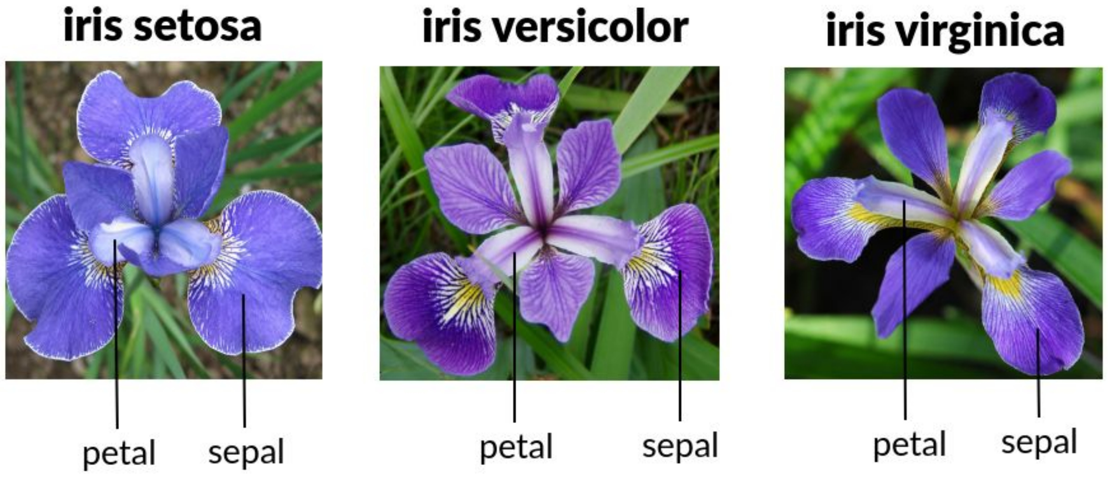
1. **Supervised learning**: We have data labeled with the correct answer
2. **Unsupervised learning**: The data is unlabeled

We will look at examples of both today.

Classic ML problem: Irises

Available in GitHub: https://github.com/CUNY-CISC-3225/datasets/tree/main/iris

Measurements taken from three species of iris:



Irises

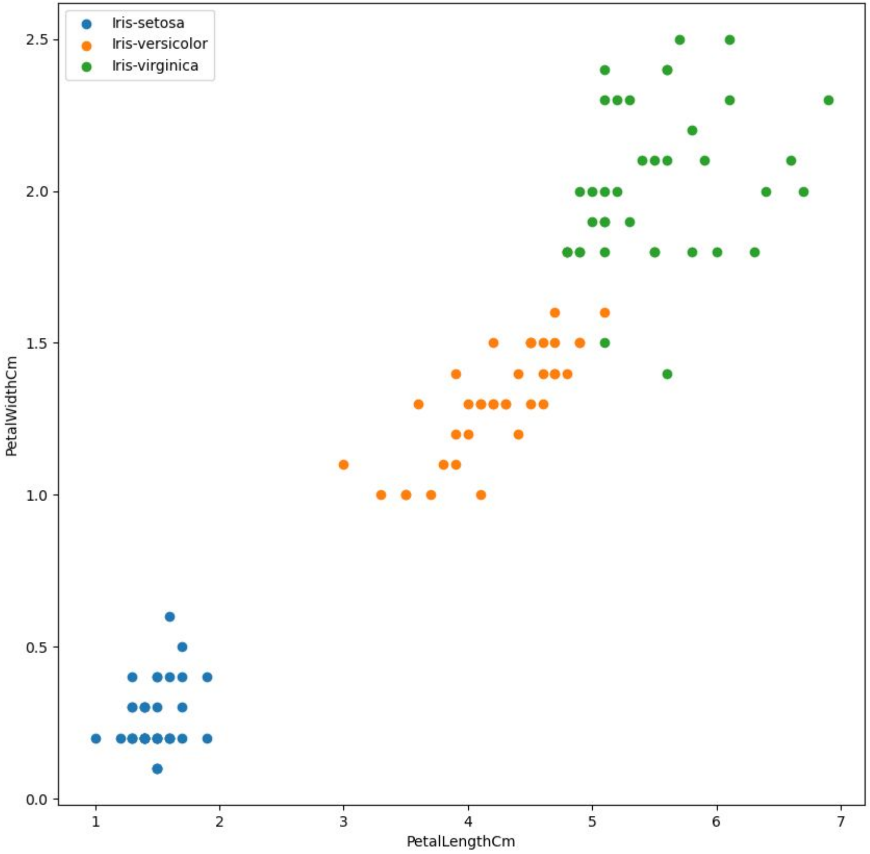
This is a supervised problem!

Observations | Label

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

...

Our data

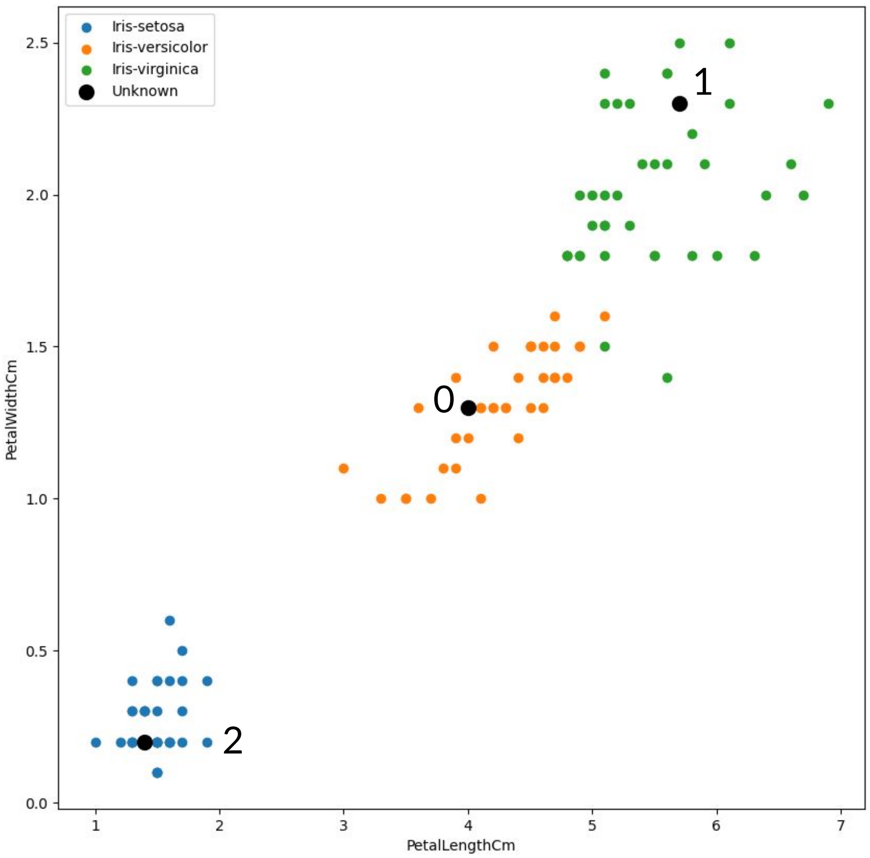


Irises: Classification

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

...

Our data



Irises: Classification

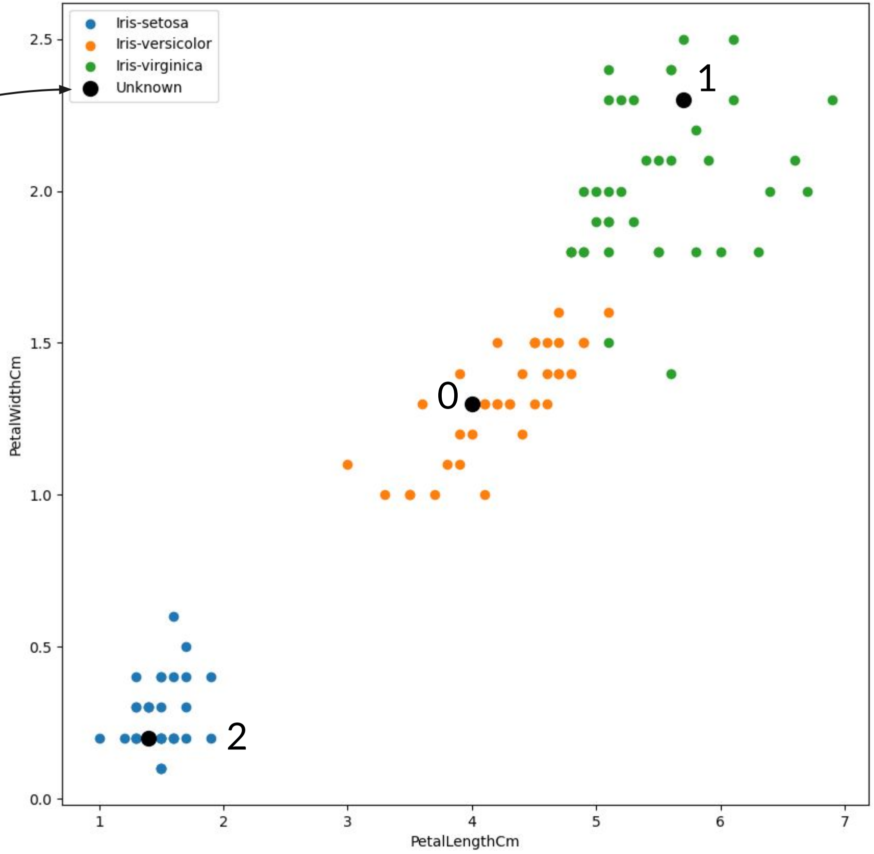
Field observations

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3
1	6.9	3.2	5.7	2.3
2	4.6	3.2	1.4	0.2

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

...

Our data



Irises: Classification

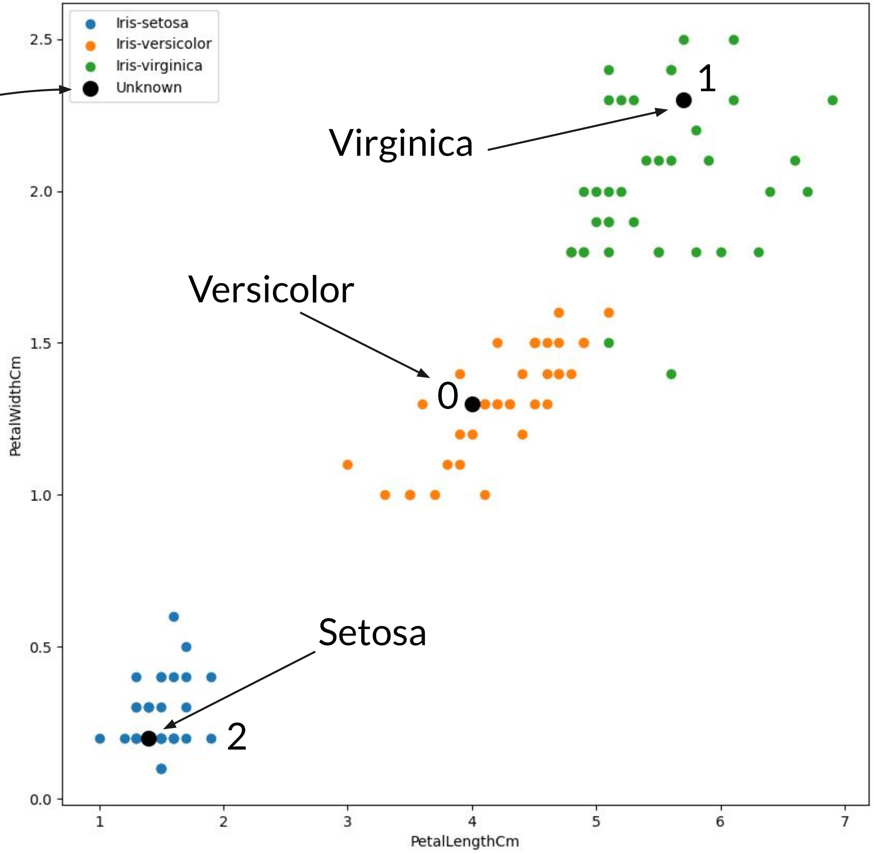
Field observations

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3
1	6.9	3.2	5.7	2.3
2	4.6	3.2	1.4	0.2

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

Our data

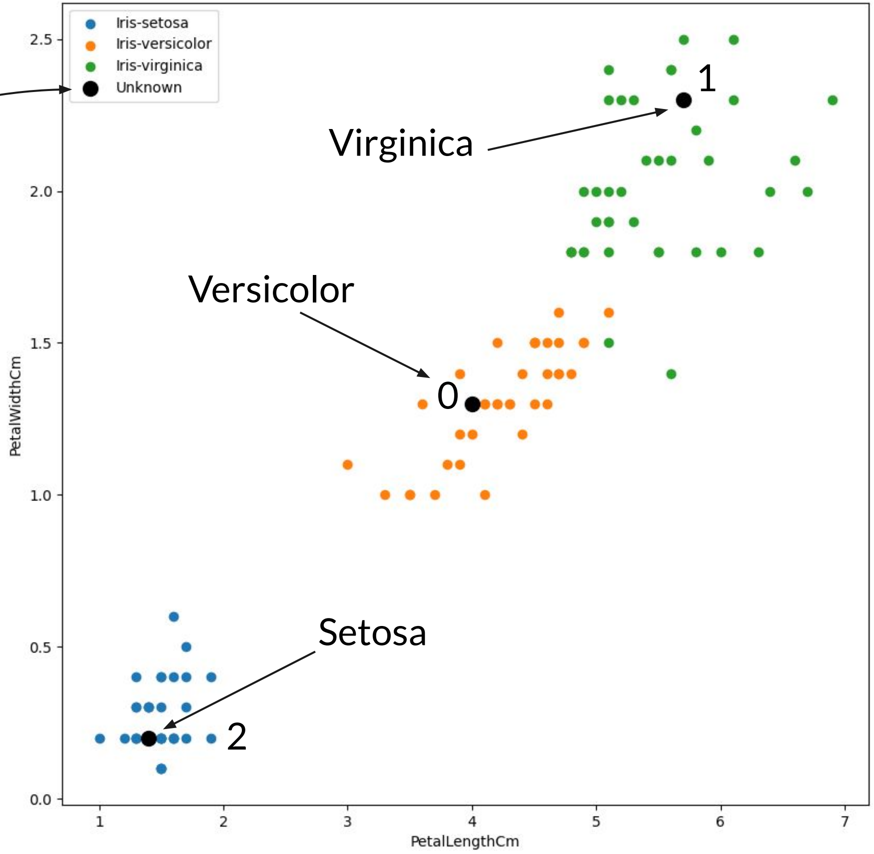
...



Irises: Classification

Field observations

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3
1	6.9	3.2	5.7	2.3
2	4.6	3.2	1.4	0.2



Assumption: Points that are close to one another are similar.

K-Nearest-Neighbors algorithm

Assumption: Points that are close to one another are similar.

We have:

- A dataset of labeled points
- One unlabeled point

Idea: Compute the *distance* between the unlabeled point and every labeled point in the dataset. Select the top *k* closest points and label based on majority voting.

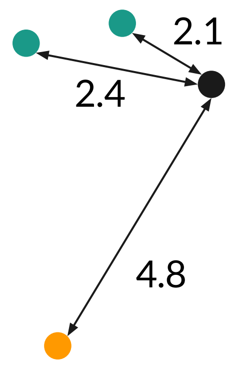
$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

K-Nearest-Neighbors algorithm

Assumption: Points that are close to one another are similar.

We have:

- A dataset of labeled points
- One unlabeled point

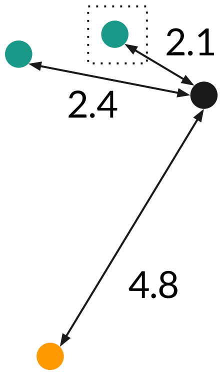


Idea: Compute the *distance* between the unlabeled point and every labeled point in the dataset. Select the top *k* closest points and label based on majority voting.

$$d(\mathbf{p},\ \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

K-Nearest-Neighbors algorithm

k = 1
Label = green



Assumption: Points that are close to one another are similar.

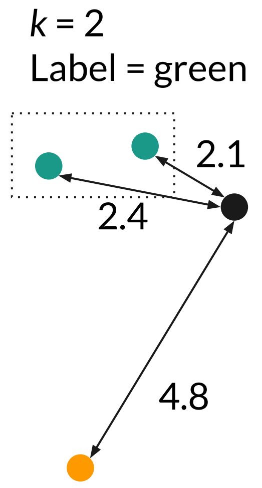
We have:

- A dataset of labeled points
- One unlabeled point

Idea: Compute the *distance* between the unlabeled point and every labeled point in the dataset. Select the top *k* closest points and label based on majority voting.

$$d(\mathbf{p},\ \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

K-Nearest-Neighbors algorithm



Assumption: Points that are close to one another are similar.

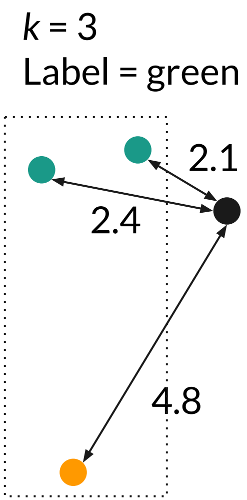
We have:

- A dataset of labeled points
- One unlabeled point

Idea: Compute the *distance* between the unlabeled point and every labeled point in the dataset. Select the top *k* closest points and label based on majority voting.

$$d(\mathbf{p},\ \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

K-Nearest-Neighbors algorithm



Assumption: Points that are close to one another are similar.

We have:

- A dataset of labeled points
- One unlabeled point

Idea: Compute the *distance* between the unlabeled point and every labeled point in the dataset. Select the top *k* closest points and label based on majority voting.

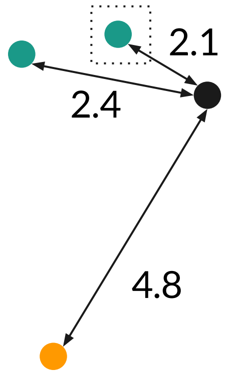
$$d(\mathbf{p},\ \mathbf{q}) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

K-Nearest-Neighbors algorithm

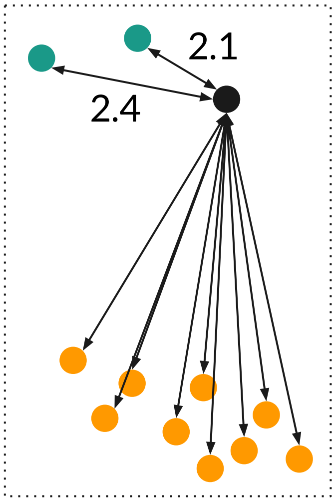
k is important:

- **Too small**: Outliers or other misplaced points may exert too much influence over the prediction.
- **Too big**: Points not in a local area will exert too much influence over the prediction
- **Good starting point**: *k*=3 or *k*=5.

k = 1
Label = green



k = 100
Label = orange



K-Nearest-Neighbors algorithm

$$d(\mathbf{p},\ \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Sum of the squared difference of each column

Our data

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

...

d

Unknown Iris

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3

1. Compute the distance between the new iris and all our data
2. Sort the distances in ascending order
3. Determine which species occurs most frequently in the top *k*

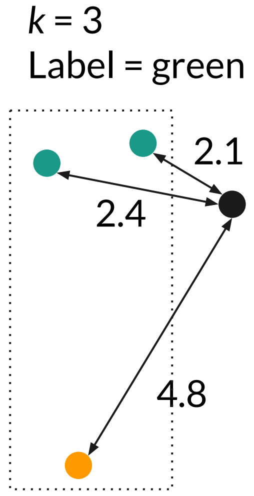
K-Nearest-Neighbors algorithm: Demo

Evaluating a K-Nearest-Neighbors Model

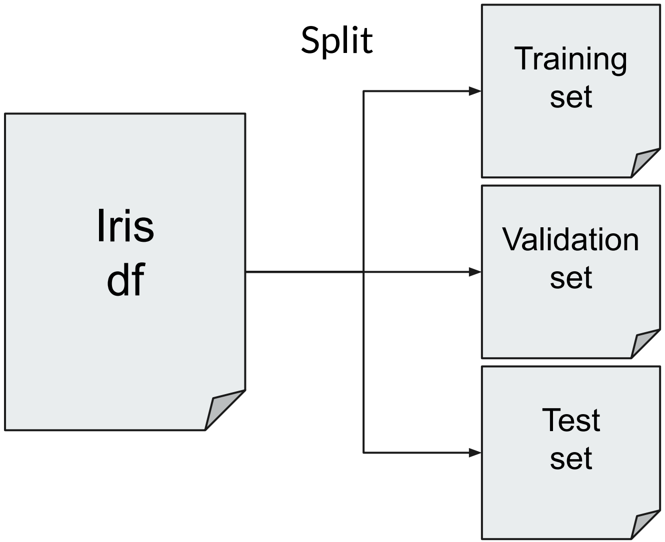
Question: How good is the *k*-nearest-neighbor algorithm at predicting iris species?

i.e., *can we trust its predictions?*

Let's find out.



Evaluating a K-Nearest-Neighbors Model



Compute distances for the *knn* model

Compare different *knn* models
(try different values of *k*)

Evaluate your chosen value of *k*

Evaluating a K-Nearest-Neighbors Model

Training set

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

Model

Evaluating a K-Nearest-Neighbors Model

Training set

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

Model

Validation set

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3
1	6.9	3.2	5.7	2.3
2	4.6	3.2	1.4	0.2

Separate the validation set observations from its labels

Species
Iris-versicolor
Iris-virginica
Iris-setosa

Evaluating a K-Nearest-Neighbors Model

Training set

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	6.9	3.1	5.1	2.3	Iris-virginica
1	6.2	2.2	4.5	1.5	Iris-versicolor
2	6.9	3.1	5.4	2.1	Iris-virginica
3	5.4	3.9	1.3	0.4	Iris-setosa
4	5.1	3.5	1.4	0.2	Iris-setosa

Model

knn Algorithm

Accuracy: 2/3

Validation set

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.5	2.3	4.0	1.3
1	6.9	3.2	5.7	2.3
2	4.6	3.2	1.4	0.2

Predictions:

- Iris-versicolor ✓
- Iris-virginica ✓
- Iris-versicolor ✗

Species
Iris-versicolor
Iris-virginica
Iris-setosa