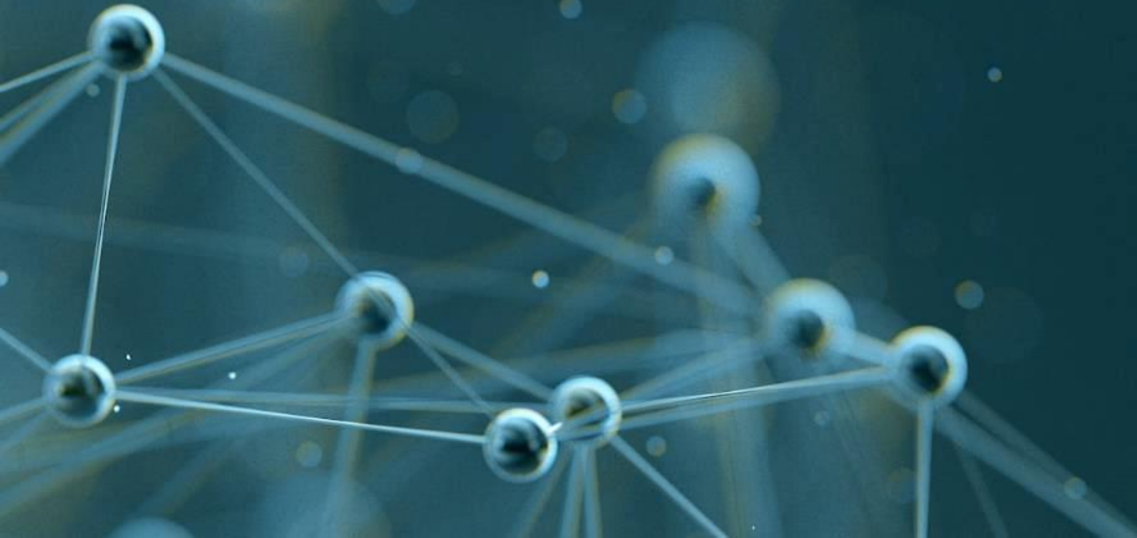
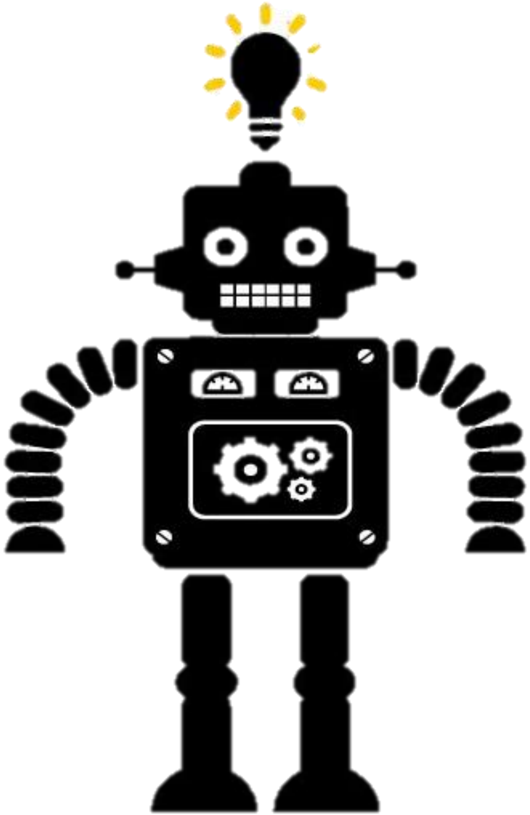
intel Nervana™ AI Academy

INTRODUCTION TO SUPERVISED LEARNING



WHAT IS MACHINE LEARNING?

Machine learning allows computers to learn and infer from data.



MACHINE LEARNING IN OUR DAILY LIVES

SPAM FILTERING

MACHINE LEARNING IN OUR DAILY LIVES

SPAM FILTERING

WEB SEARCH

MACHINE LEARNING IN OUR DAILY LIVES

SPAM FILTERING

WEB SEARCH

POSTAL MAIL ROUTING

MACHINE LEARNING IN OUR DAILY LIVES

SPAM FILTERING

WEB SEARCH

POSTAL MAIL ROUTING

FRAUD DETECTION

MOVIE RECOMMENDATIONS

VEHICLE DRIVER ASSISTANCE

WEB ADVERTISEMENTS

SOCIAL NETWORKS

SPEECH RECOGNITION

TYPES OF MACHINE LEARNING

SUPERVISED	Data points have known outcome

TYPES OF MACHINE LEARNING

SUPERVISED	Data points have known outcome
UNSUPERVISED	Data points have unknown outcome

TYPES OF MACHINE LEARNING

SUPERVISED	Data points have known outcome
UNSUPERVISED	Data points have unknown outcome

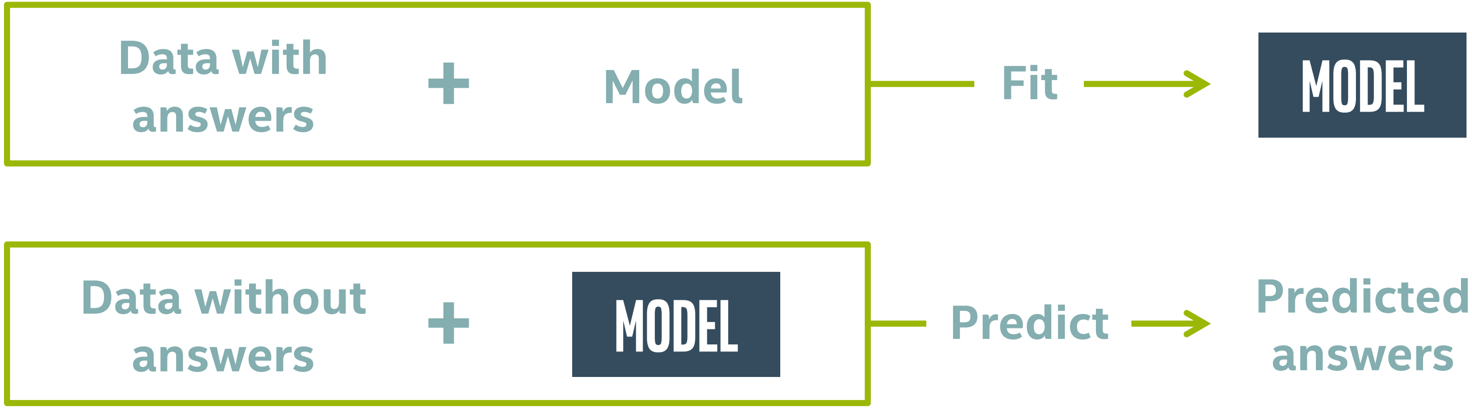
TYPES OF SUPERVISED LEARNING

REGRESSION	Outcome is continuous (numerical)

TYPES OF SUPERVISED LEARNING

REGRESSION	Outcome is continuous (numerical)
CLASSIFICATION	Outcome is a category

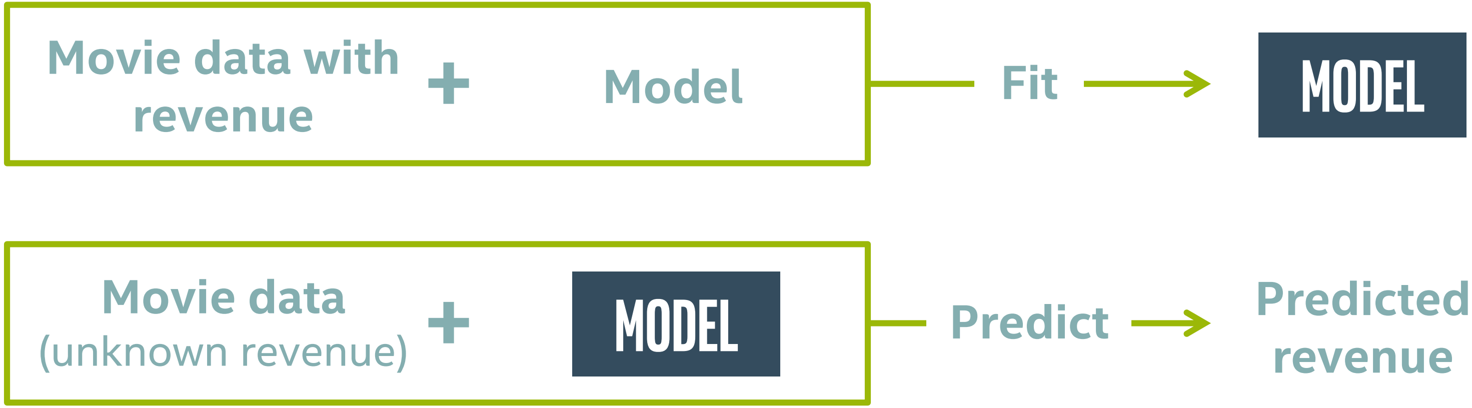
SUPERVISED LEARNING OVERVIEW



Data with answers + Model — Fit → MODEL

Data without answers + MODEL — Predict → Predicted answers

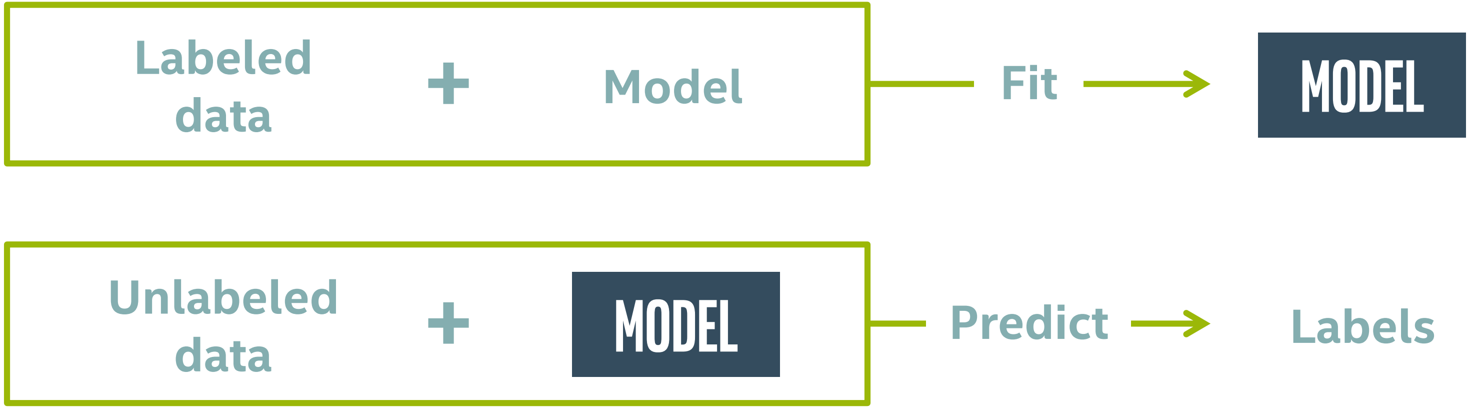
REGRESSION: NUMERICAL ANSWERS



Movie data with revenue + Model — Fit → MODEL

Movie data (unknown revenue) + MODEL — Predict → Predicted revenue

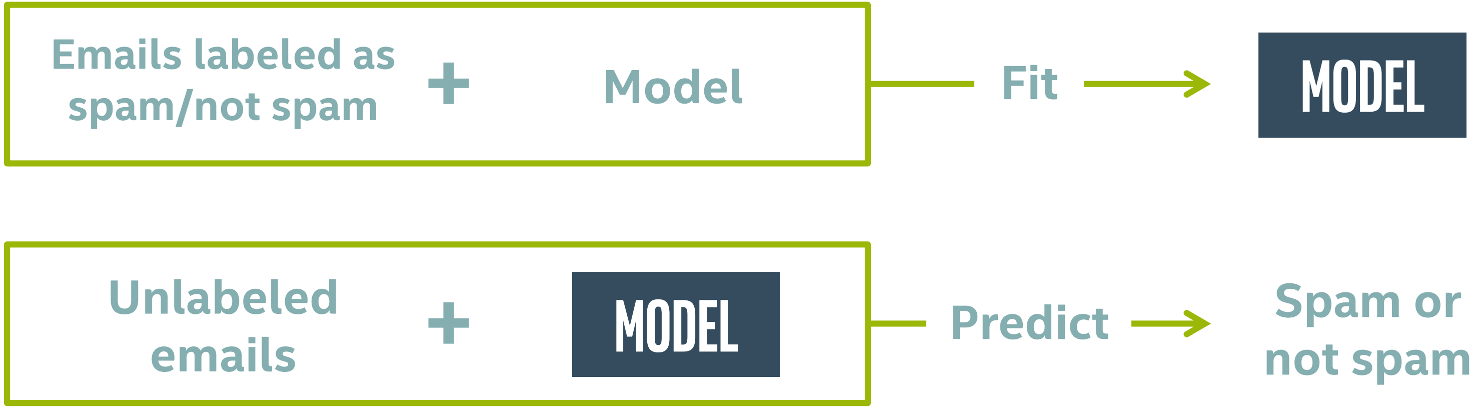
CLASSIFICATION: CATEGORICAL ANSWERS



Labeled data + Model — Fit → MODEL

Unlabeled data + MODEL — Predict → Labels

CLASSIFICATION: CATEGORICAL ANSWERS



Emails labeled as spam/not spam + Model

Fit → MODEL

Unlabeled emails + MODEL

Predict → Spam or not spam

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

Target

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)

- **Features: properties of the data used for prediction**
  (non-target columns)

MACHINE LEARNING VOCABULARY

Features

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)
- **Features: properties of the data used for prediction**
  (non-target columns)
- **Example: a single data point within the data**
  (one row)

MACHINE LEARNING VOCABULARY

Examples →

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

MACHINE LEARNING VOCABULARY

- **Target: predicted category or value of the data**
  (column to predict)

- **Features: properties of the data used for prediction**
  (non-target columns)

- **Example: a single data point within the data**
  (one row)

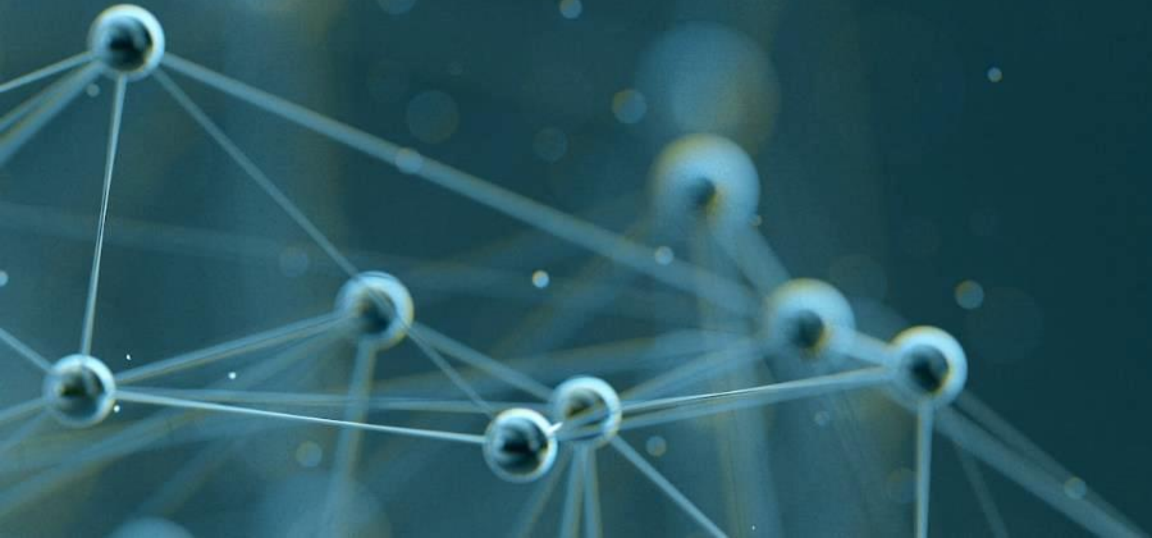
- **Label: the target value for a single data point**

MACHINE LEARNING VOCABULARY

Sepal length	Sepal width	Petal length	Petal width	Species
6.7	3.0	5.2	2.3	Virginica
6.4	2.8	5.6	2.1	Virginica
4.6	3.4	1.4	0.3	Setosa
6.9	3.1	4.9	1.5	Versicolor
4.4	2.9	1.4	0.2	Setosa
4.8	3.0	1.4	0.1	Setosa
5.9	3.0	5.1	1.8	Virginica
5.4	3.9	1.3	0.4	Setosa
4.9	3.0	1.4	0.2	Setosa
5.4	3.4	1.7	0.2	Setosa

Label

K – NEAREST NEIGHBORS



WHAT IS CLASSIFICATION?

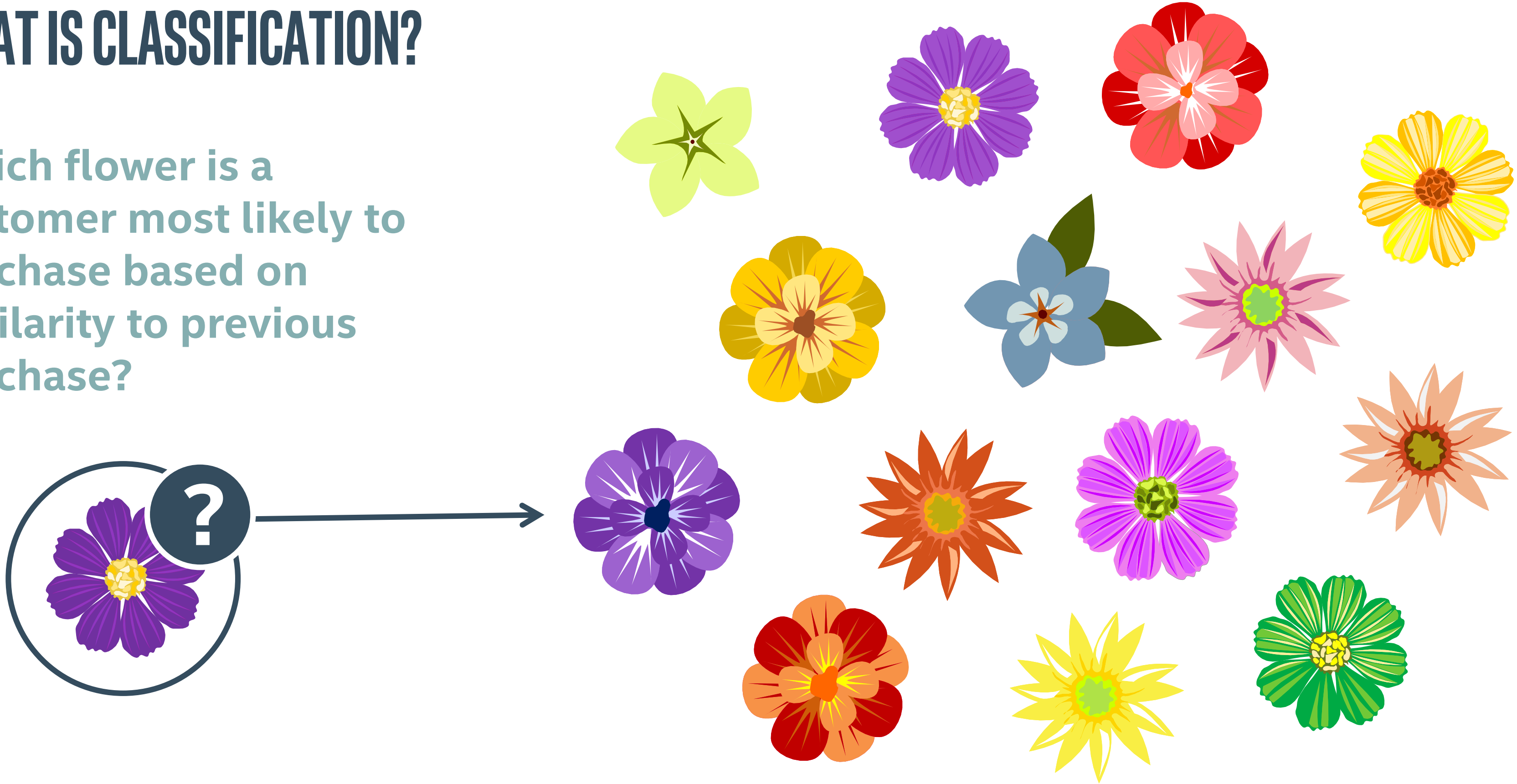
A flower shop wants to guess a customer's purchase from similarity to most recent purchase.

WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?

WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?

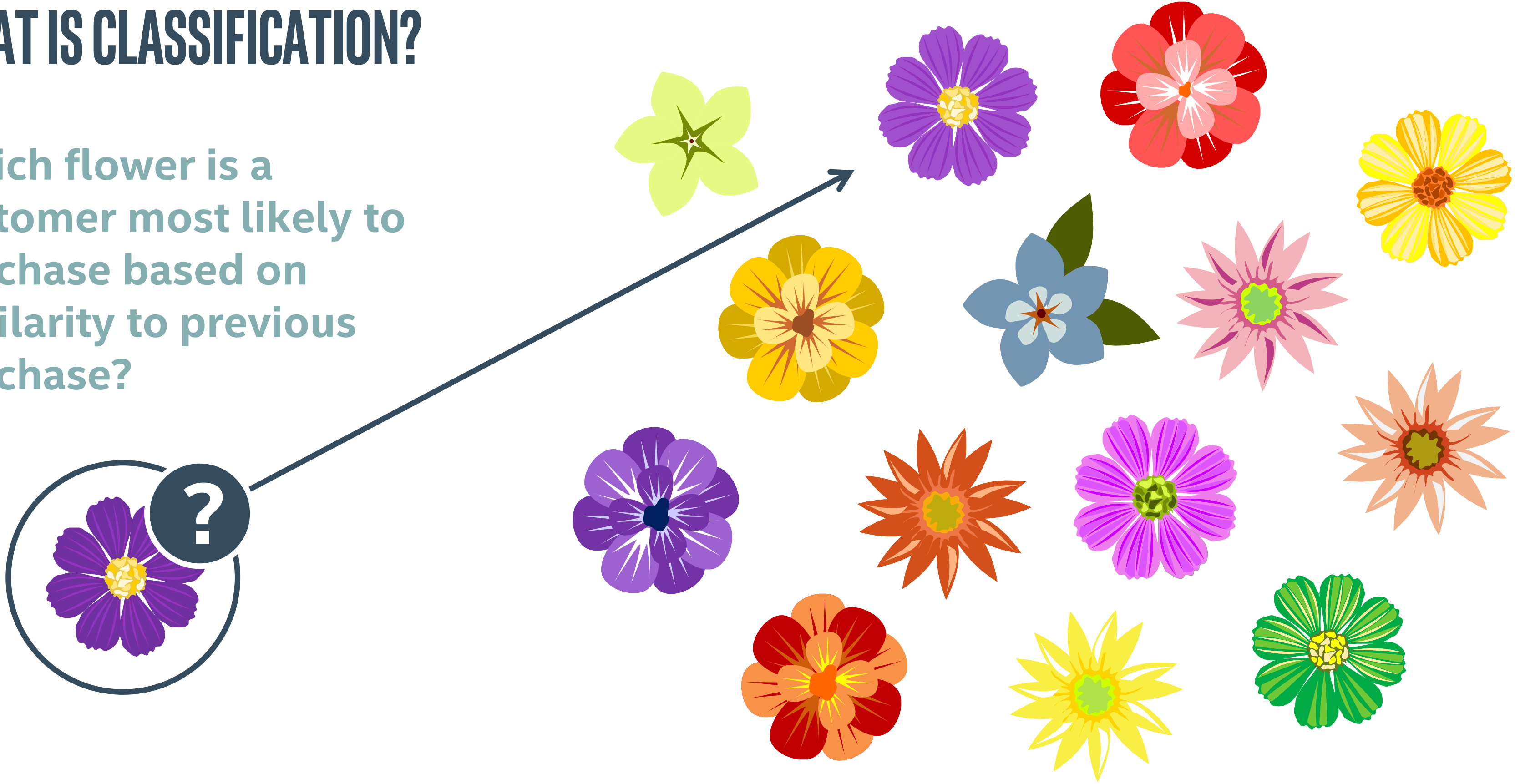


WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?

WHAT IS CLASSIFICATION?

Which flower is a customer most likely to purchase based on similarity to previous purchase?



WHAT IS NEEDED FOR CLASSIFICATION?

- **Model data with:**
  - Features that can be quantitated

WHAT IS NEEDED FOR CLASSIFICATION?

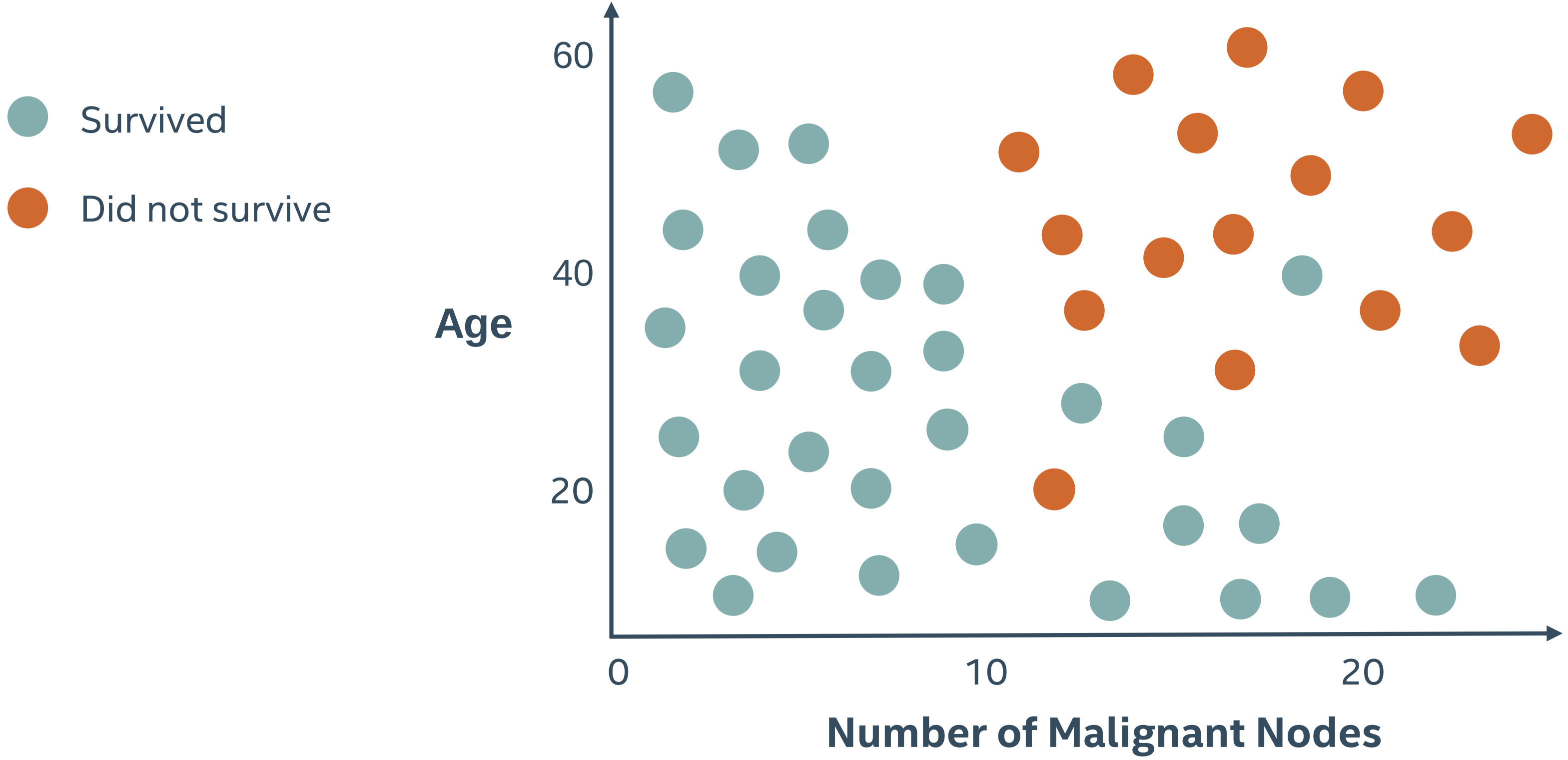
- **Model data with:**
  - Features that can be quantitated
  - Labels that are known

WHAT IS NEEDED FOR CLASSIFICATION?

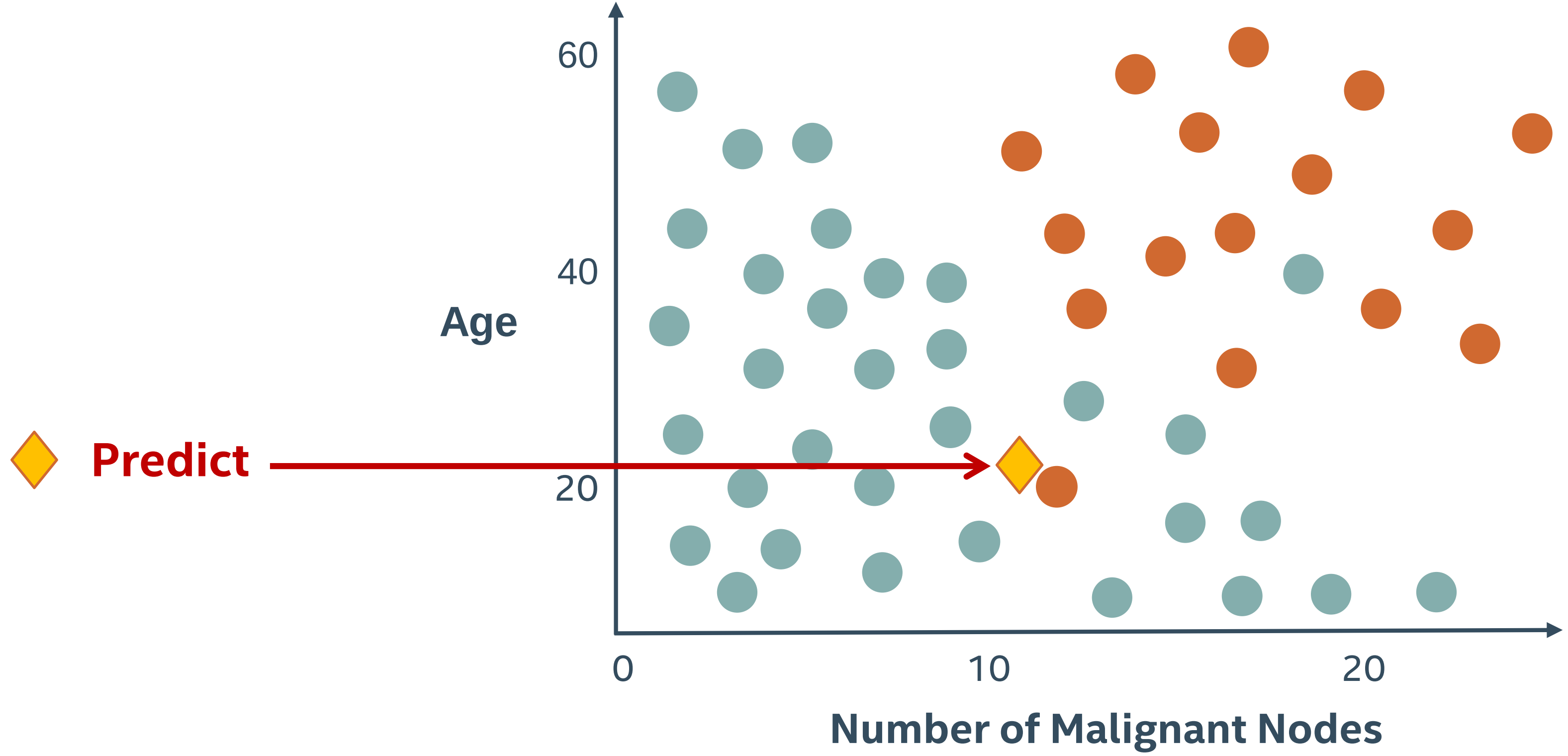
- **Model data with:**
  - Features that can be quantitated
  - Labels that are known
- **Method to measure similarity**

K NEAREST NEIGHBORS CLASSIFICATION

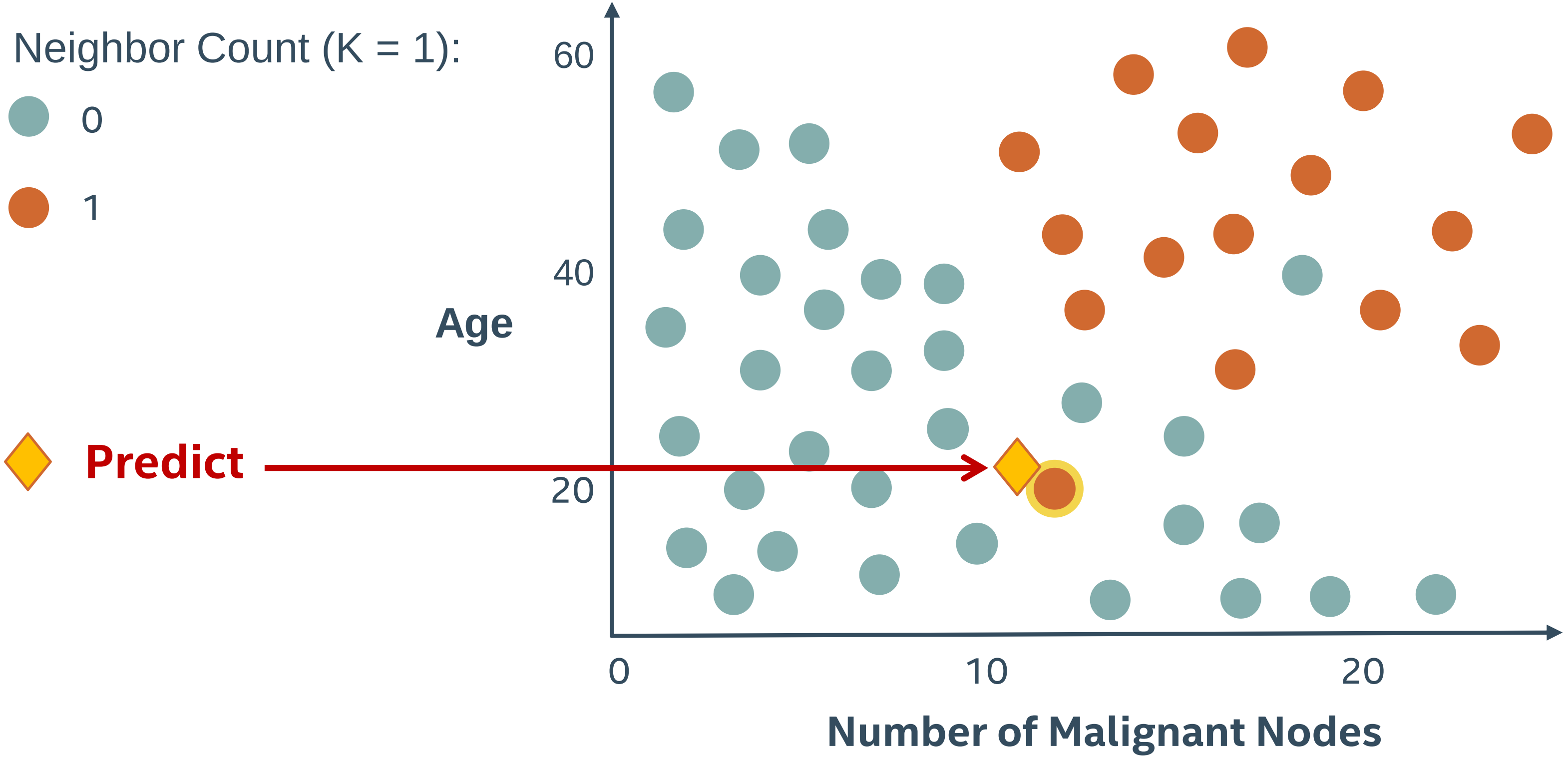
K NEAREST NEIGHBORS CLASSIFICATION



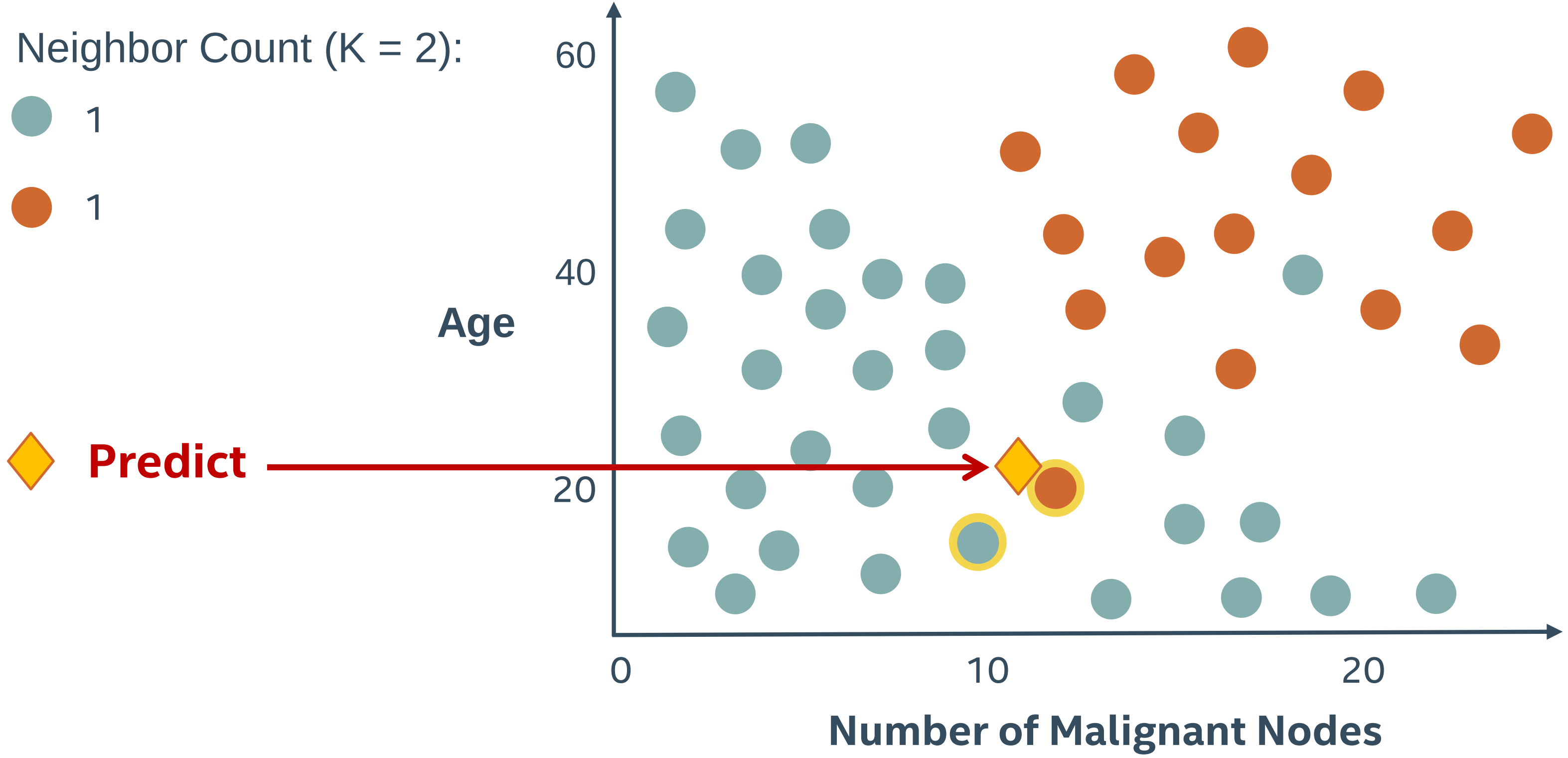
K NEAREST NEIGHBORS CLASSIFICATION



K NEAREST NEIGHBORS CLASSIFICATION



K NEAREST NEIGHBORS CLASSIFICATION



Neighbor Count (K = 2):

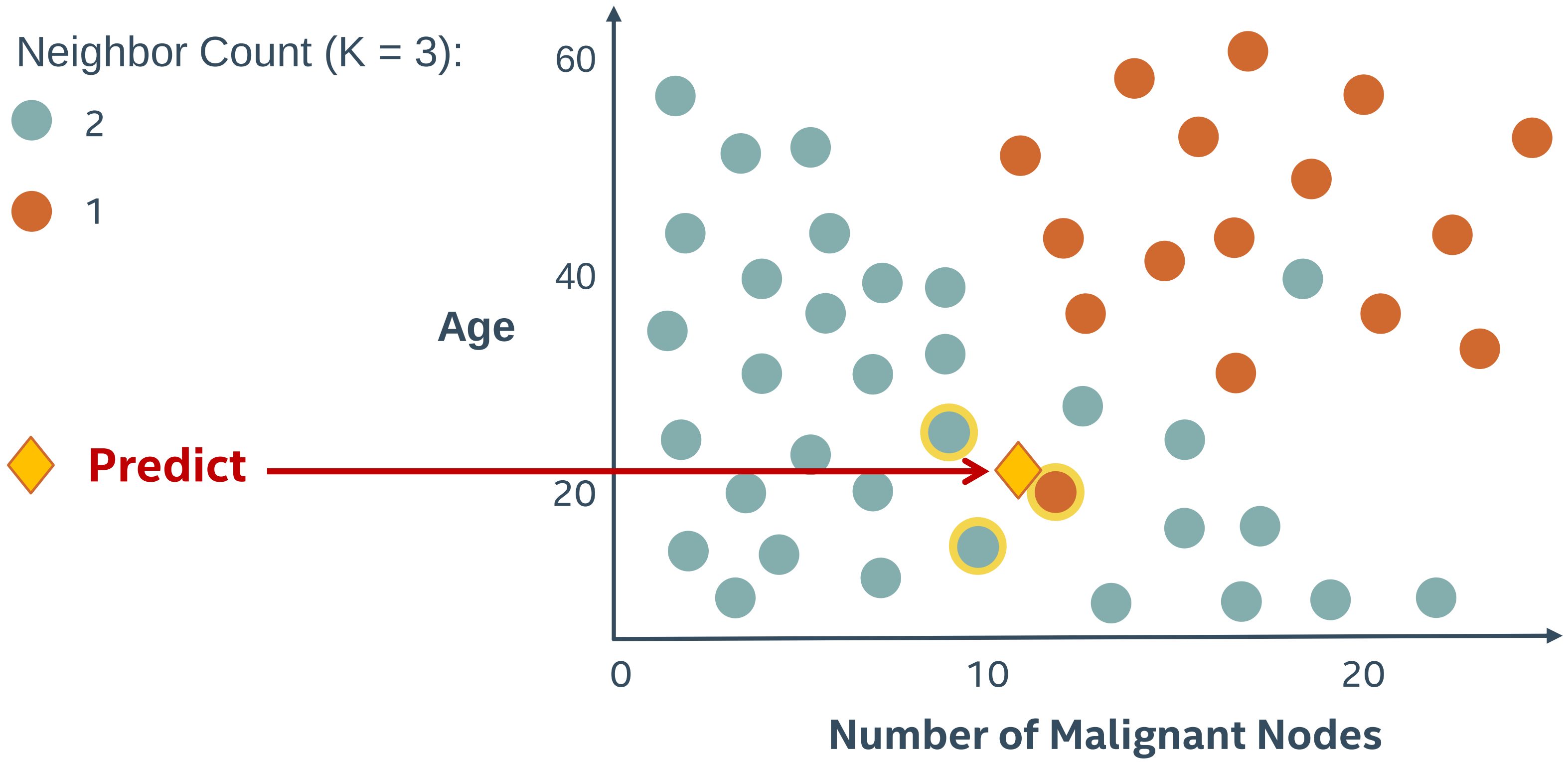
- ● 1
- ● 1

K NEAREST NEIGHBORS CLASSIFICATION

Neighbor Count (K = 3):

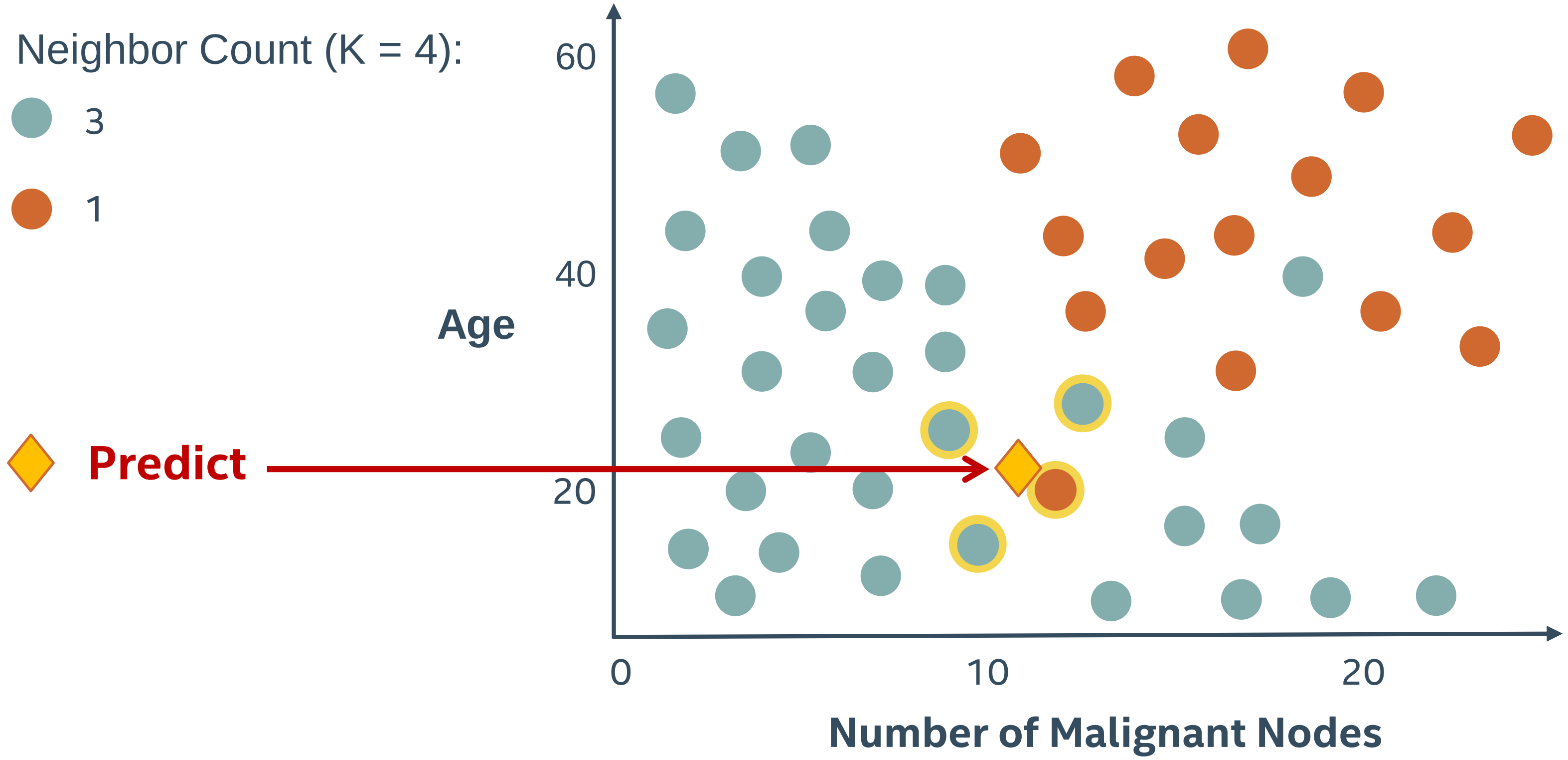
● 2

● 1



◆ Predict

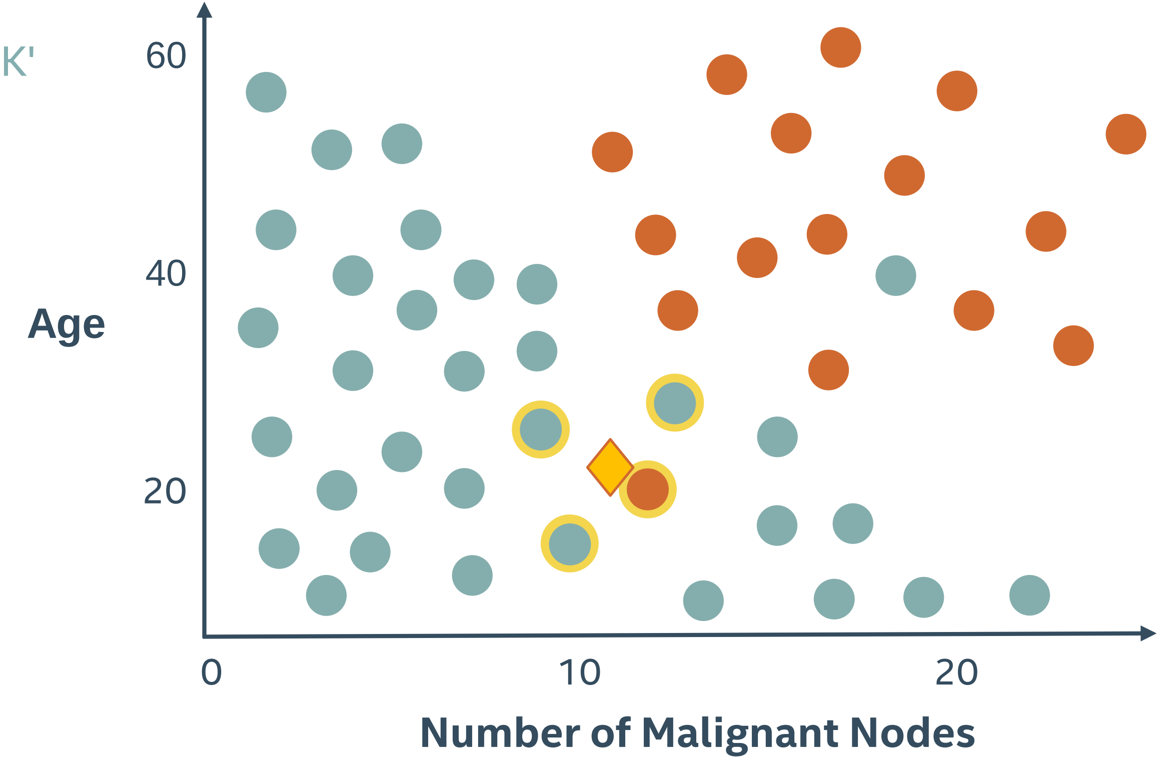
K NEAREST NEIGHBORS CLASSIFICATION



WHAT IS NEEDED TO SELECT A KNN MODEL?

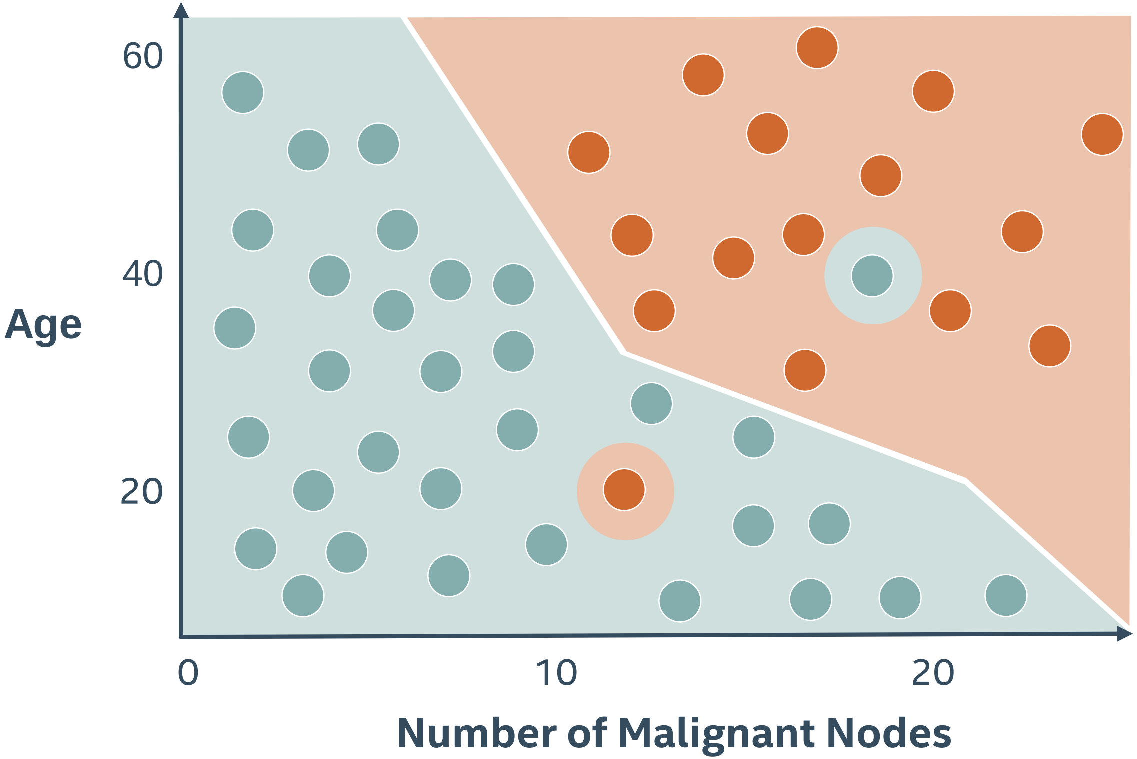
WHAT IS NEEDED TO SELECT A KNN MODEL?

- Correct value for 'K'
- How to measure closeness of neighbors?



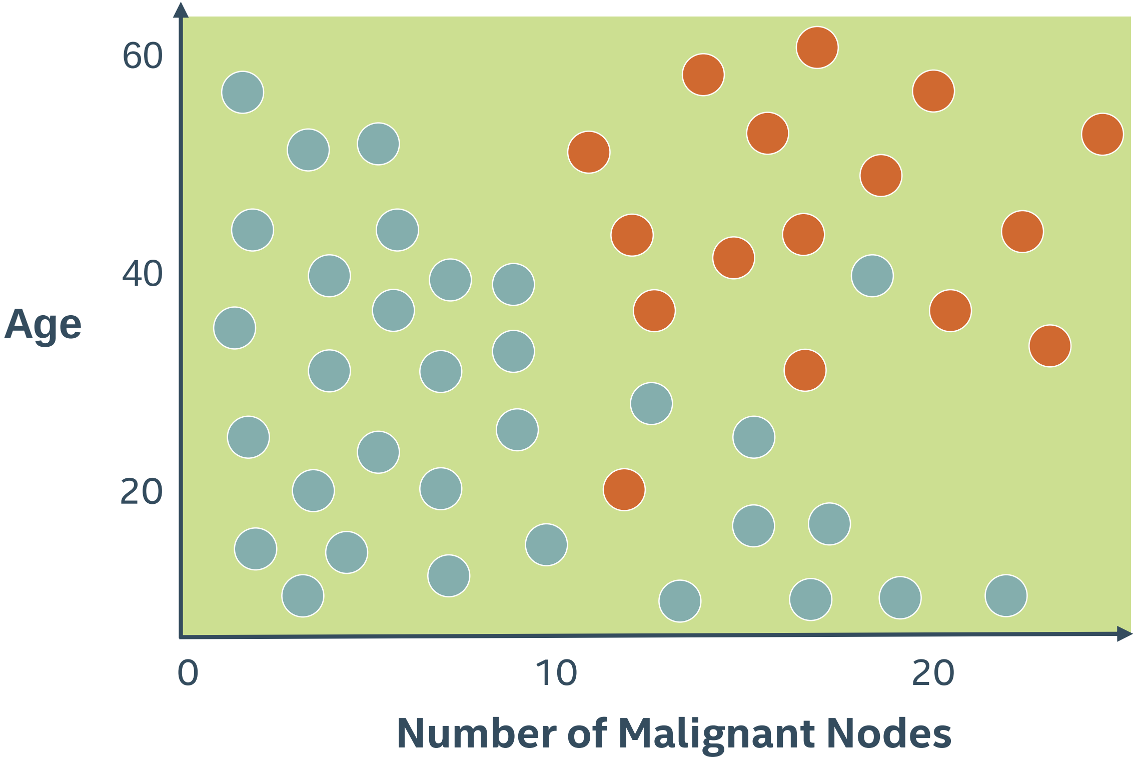
K NEAREST NEIGHBORS DECISION BOUNDARY

K=1

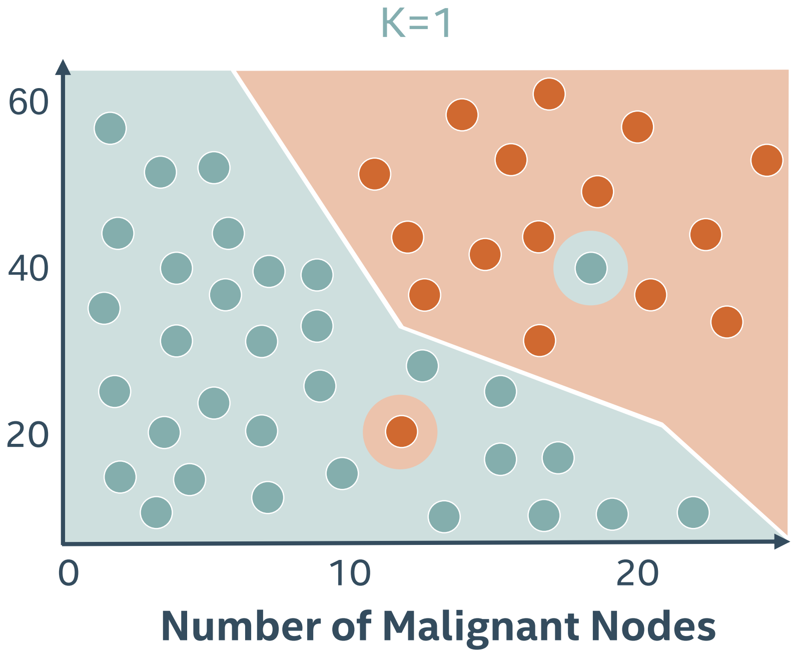


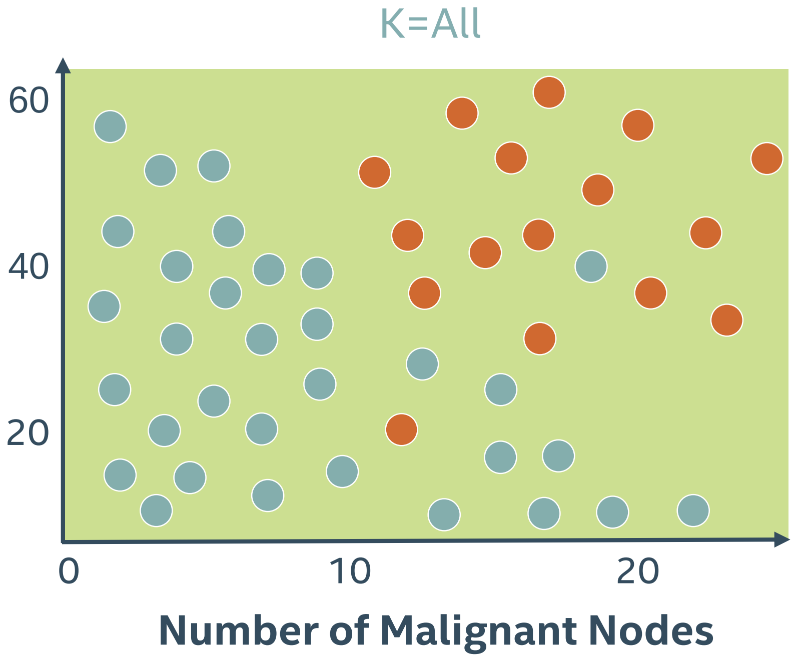
K NEAREST NEIGHBORS DECISION BOUNDARY

K = All

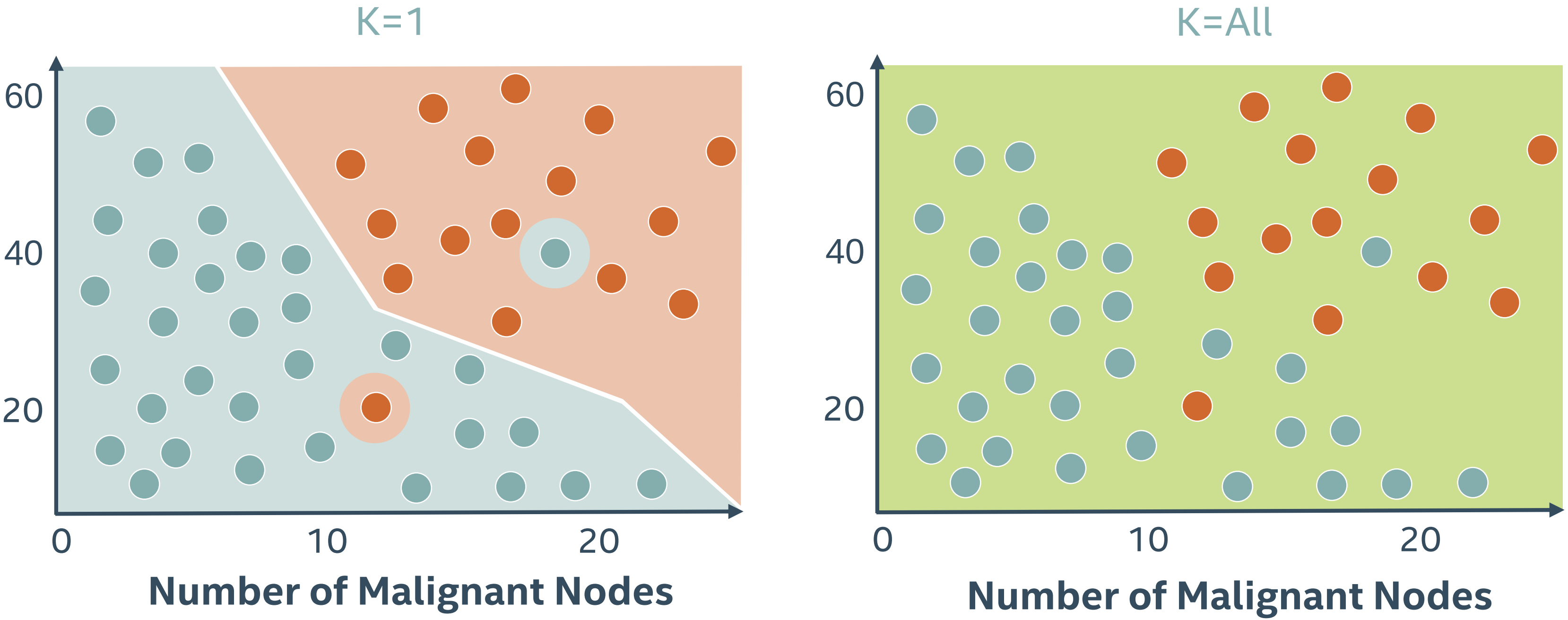


VALUE OF 'K' AFFECTS DECISION BOUNDARY



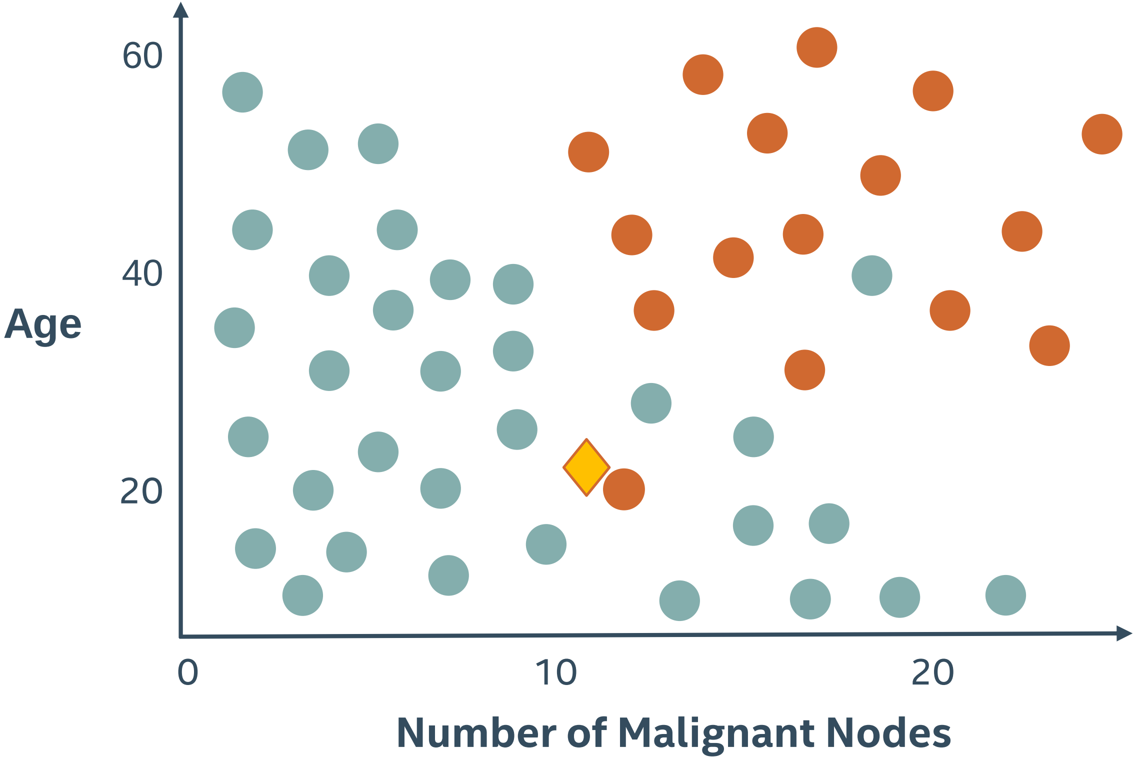


VALUE OF 'K' AFFECTS DECISION BOUNDARY

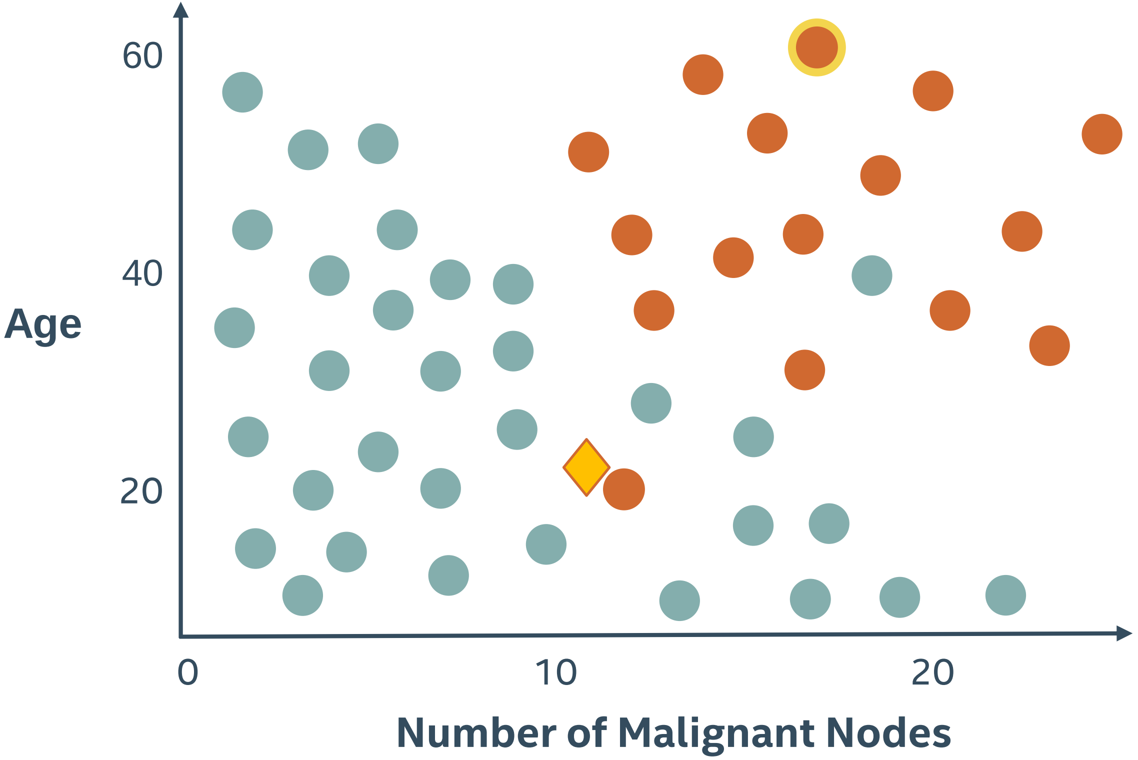


Methods for determining 'K' will be discussed in next lesson

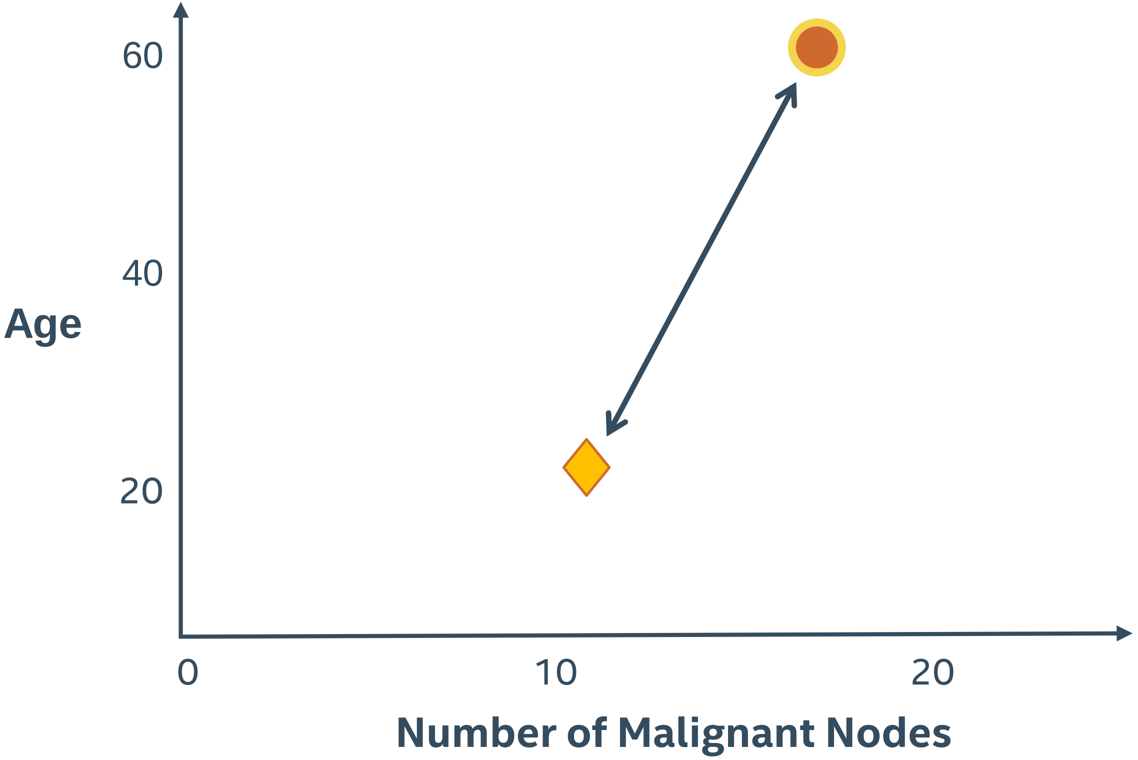
MEASUREMENT OF DISTANCE IN KNN



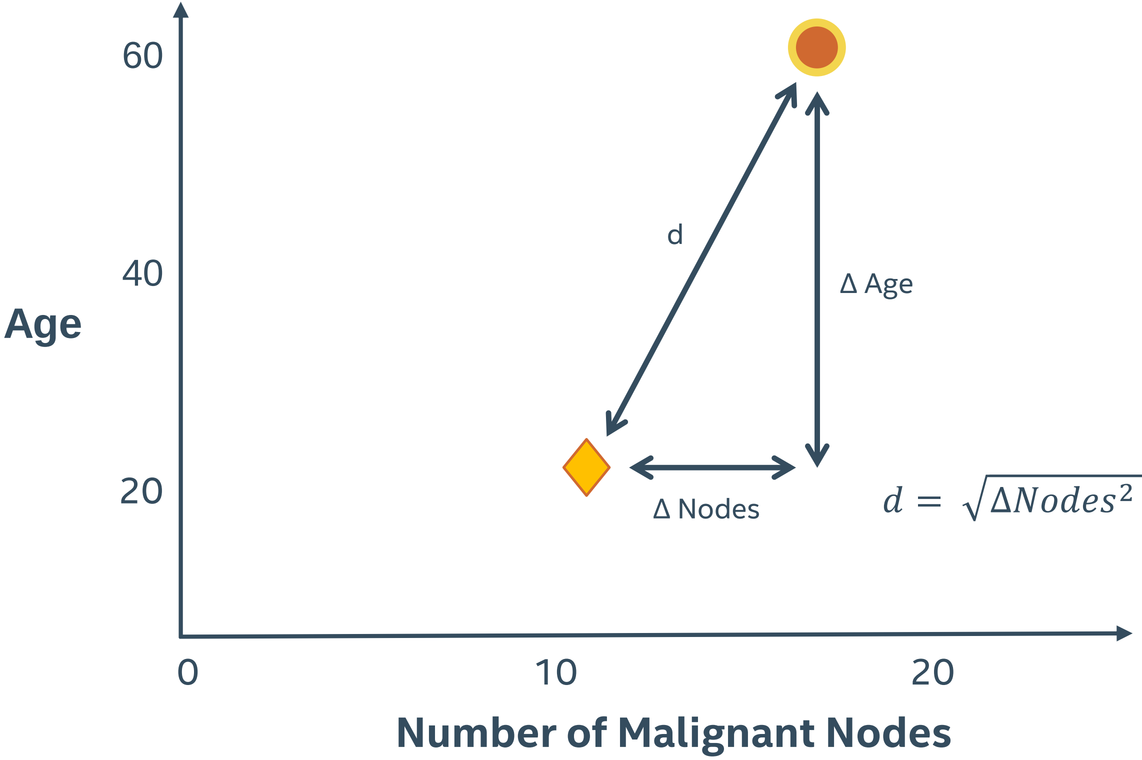
MEASUREMENT OF DISTANCE IN KNN



EUCLIDEAN DISTANCE

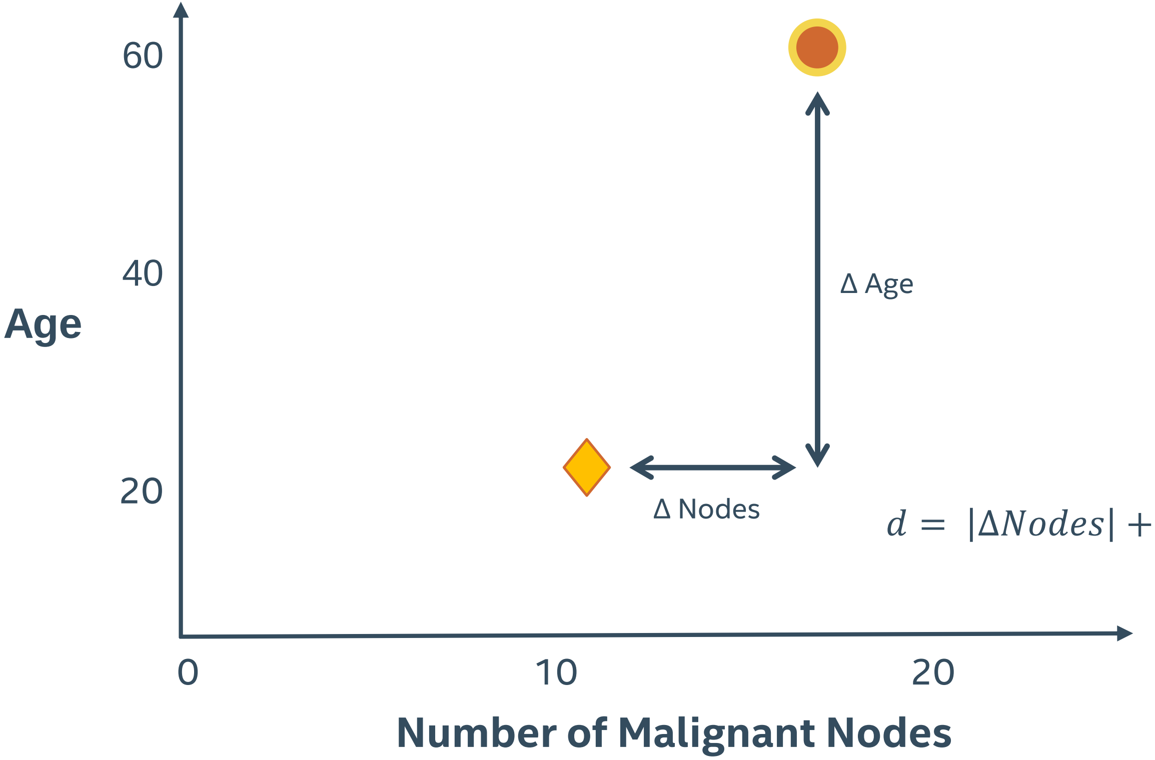


EUCLIDEAN DISTANCE (L2 DISTANCE)



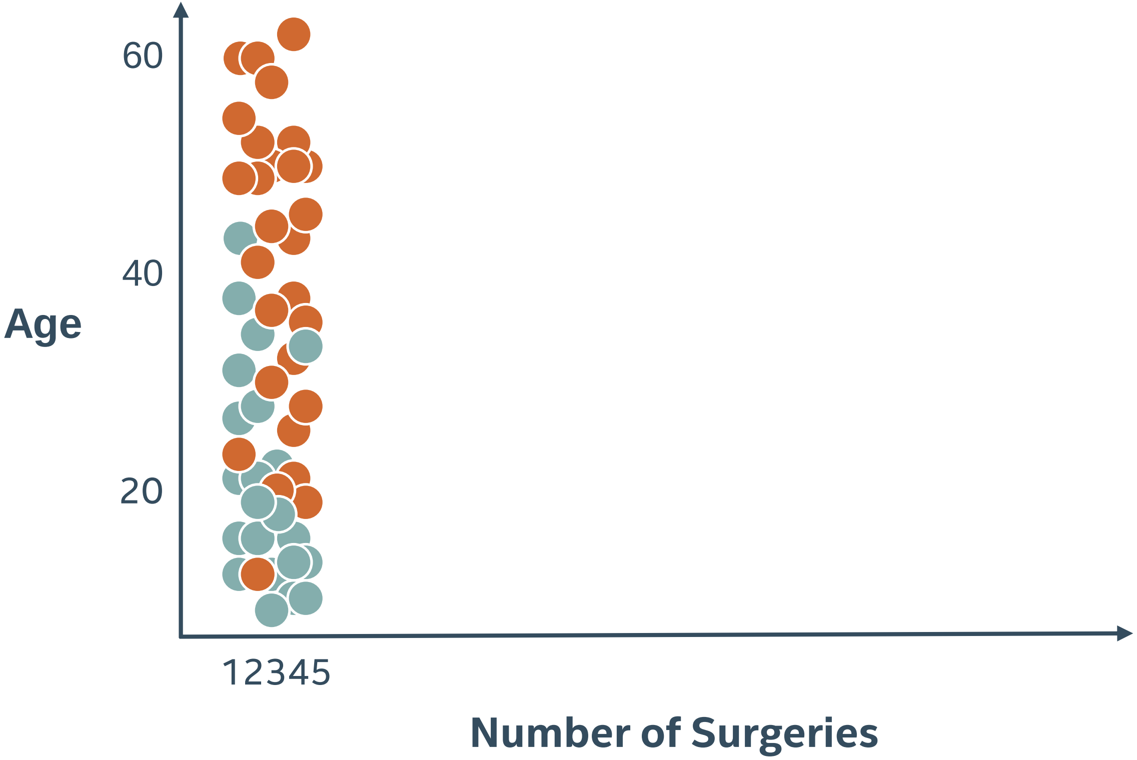
$$d = \sqrt{\Delta Nodes^2 + \Delta Age^2}$$

MANHATTAN DISTANCE (L1 OR CITY BLOCK DISTANCE)

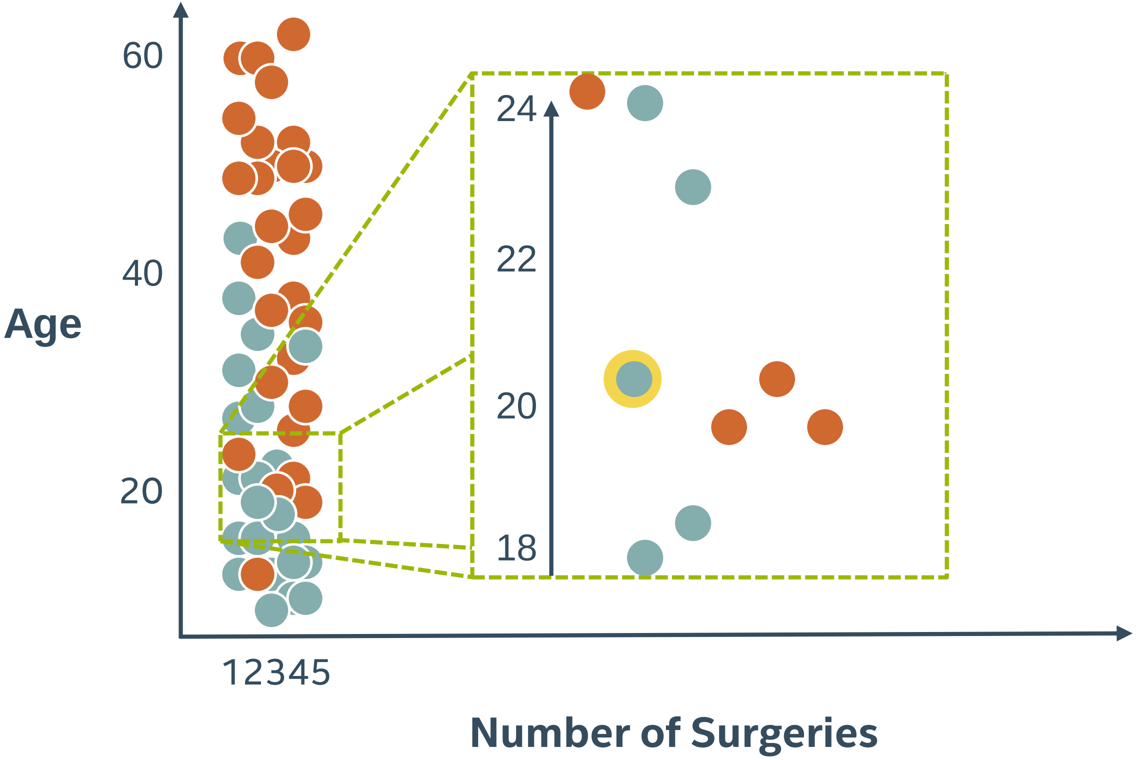


$$d = |\Delta Nodes| + |\Delta Age|$$

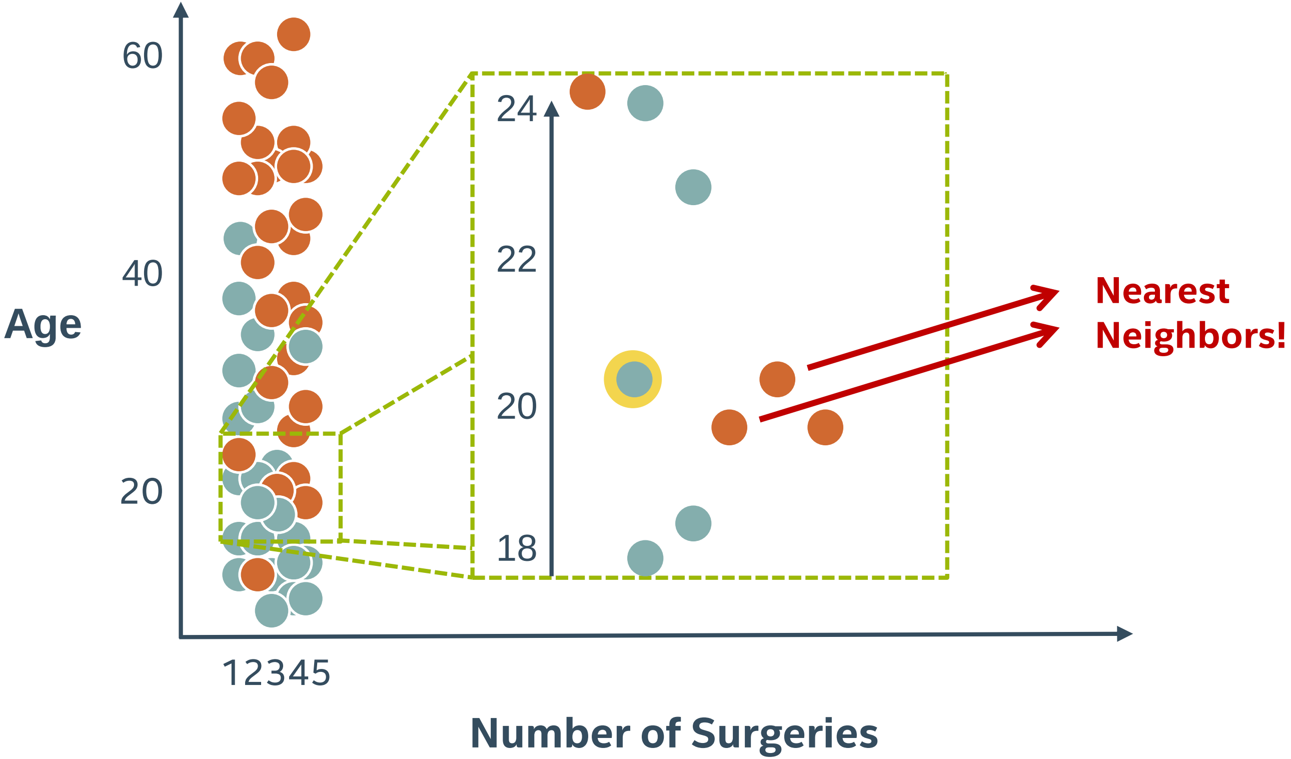
SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT



SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

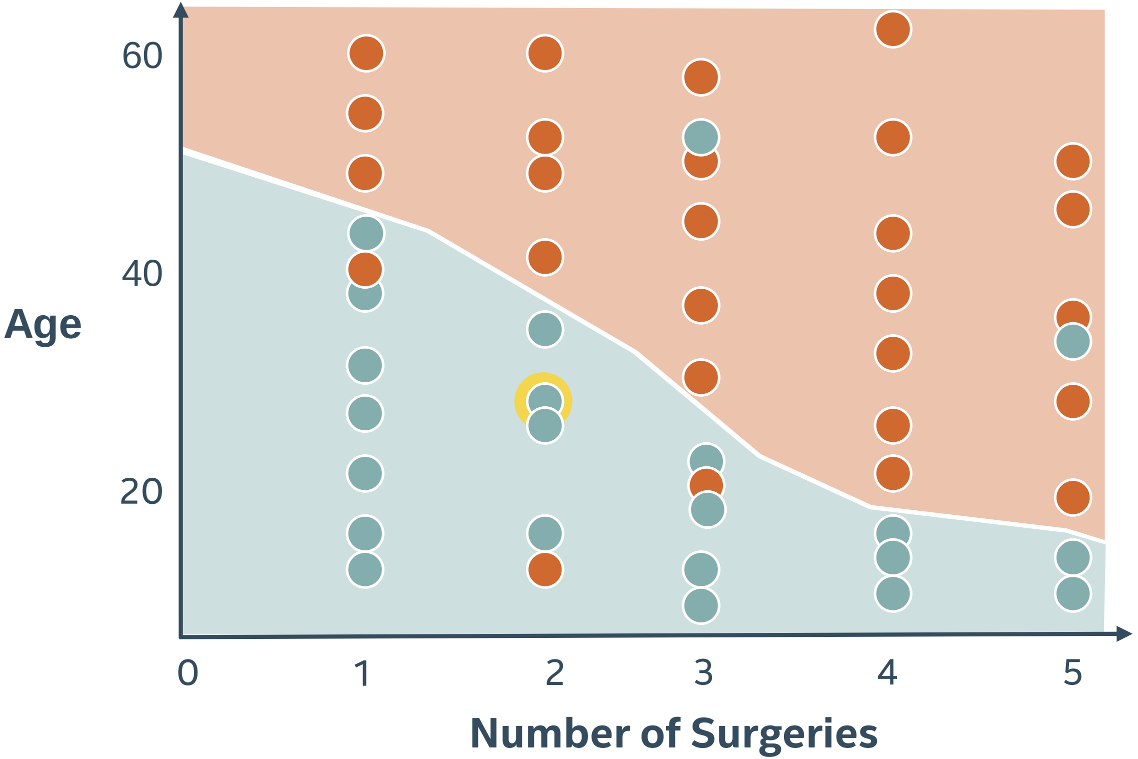


SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT



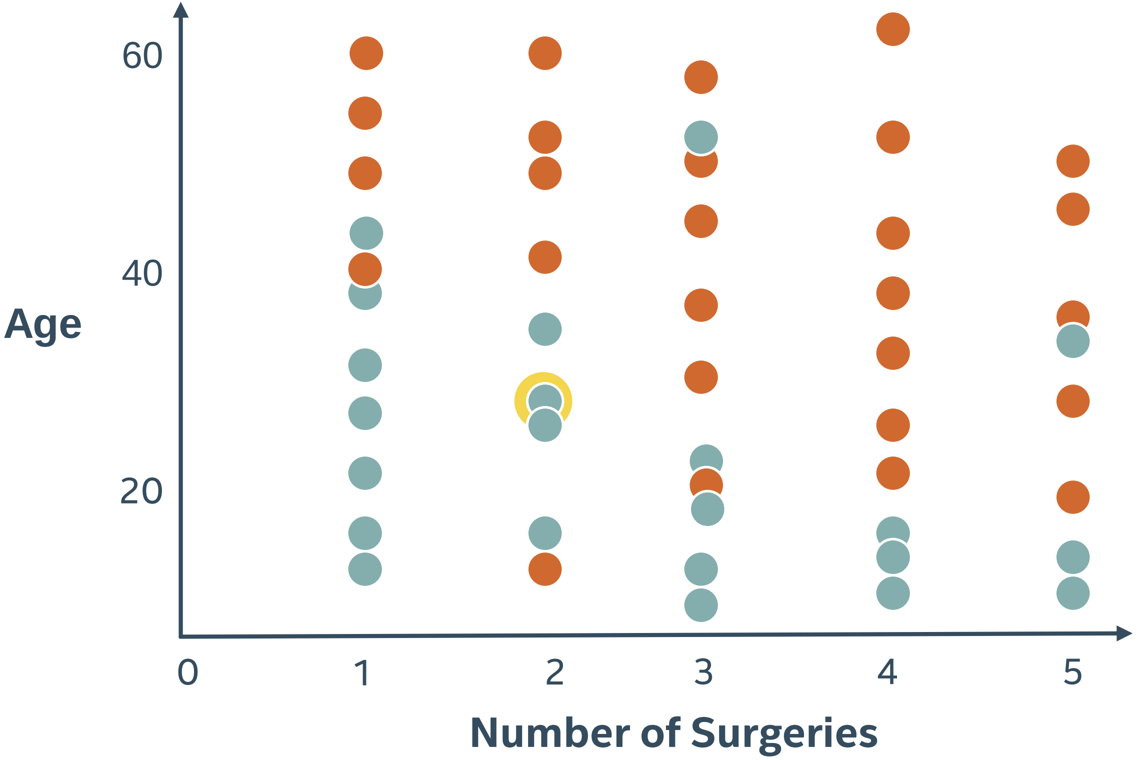
SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

"Feature Scaling"



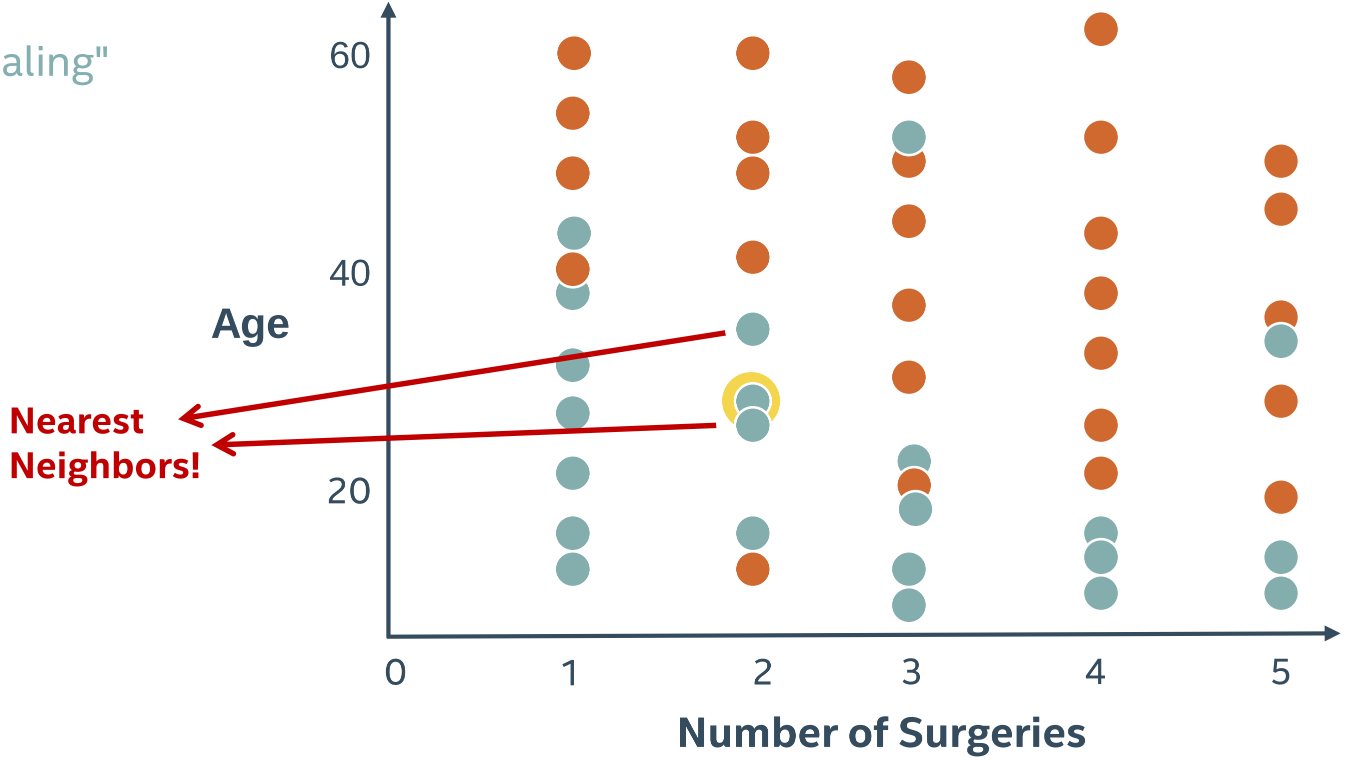
SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

"Feature Scaling"



SCALE IS IMPORTANT FOR DISTANCE MEASUREMENT

"Feature Scaling"



COMPARISON OF FEATURE SCALING METHODS

- **Standard Scaler:** Mean center data and scale to unit **variance**
- **Minimum-Maximum Scaler:** Scale data to fixed range (usually 0–1)
- **Maximum Absolute Value Scaler:** Scale maximum absolute value

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
    from sklearn.preprocessing import StandardScaler
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

Fit the scaling parameters and then transform the data

```
StdSc = StdSc.fit(X_data)

X_scaled = KNN.transform(X_data)
```

FEATURE SCALING: THE SYNTAX

Import the class containing the scaling method

```
from sklearn.preprocessing import StandardScaler
```

Create an instance of the class

```
StdSc = StandardScaler()
```

Fit the scaling parameters and then transform the data

```
StdSc = StdSc.fit(X_data)

X_scaled = KNN.transform(X_data)
```

Other scaling methods exist: MinMaxScaler, MaxAbsScaler.

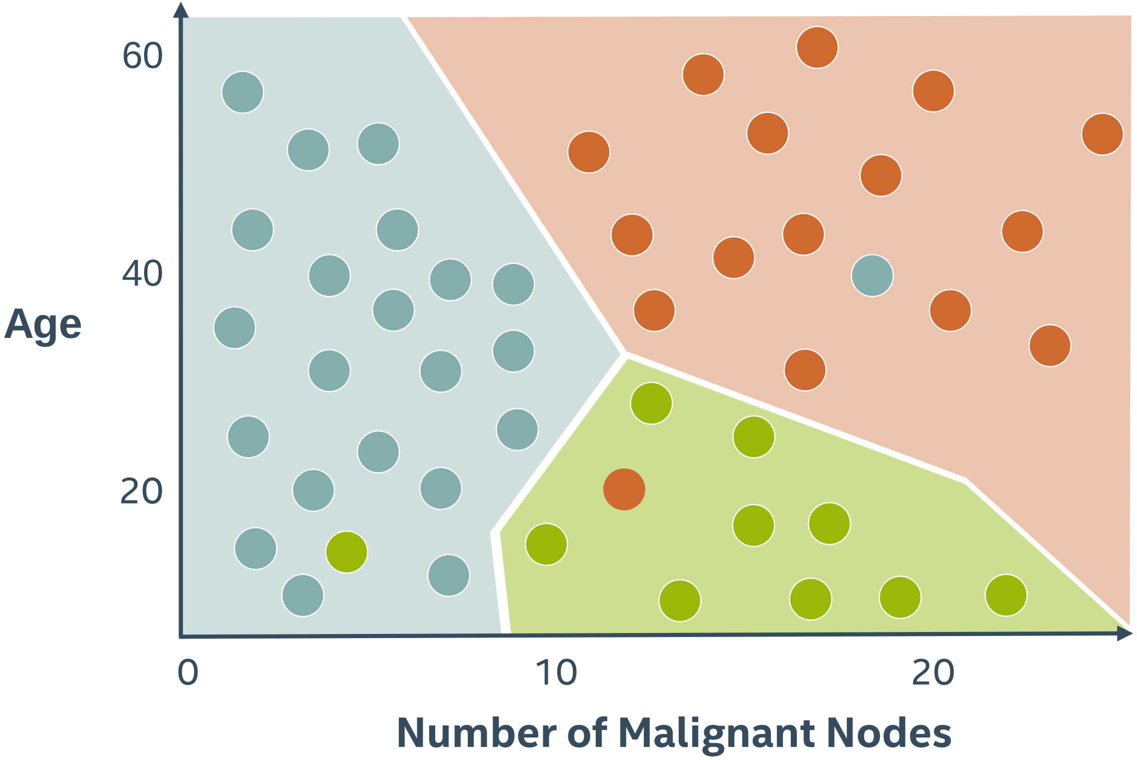
MULTICLASS KNN DECISION BOUNDARY

K=5

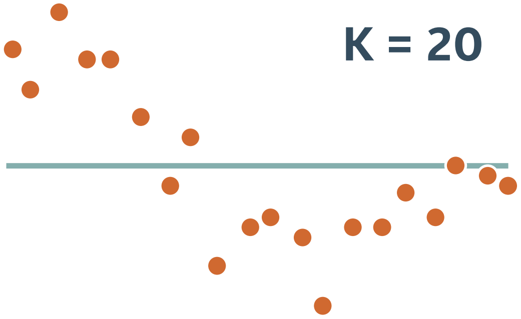
● Full remission

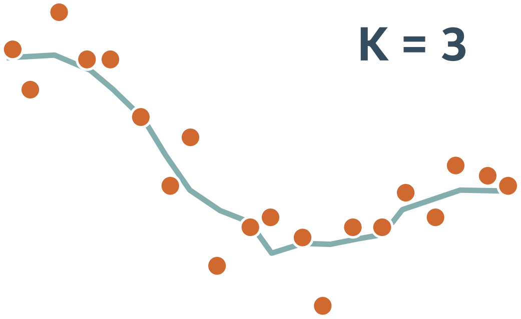
● Partial remission

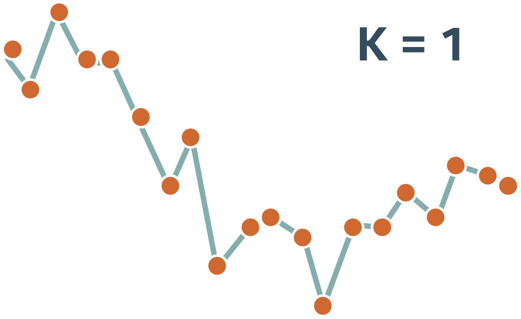
● Did not survive



REGRESSION WITH KNN







CHARACTERISTICS OF A KNN MODEL

- Fast to create model because it simply stores data
- Slow to predict because many distance calculations
- Can require lots of memory if data set is large

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
    from sklearn.neighbors import KNeighborsClassifier
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

The fit and predict/transform syntax will show up throughout the course.

K NEAREST NEIGHBORS: THE SYNTAX

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN = KNN.fit(X_data, y_data)

y_predict = KNN.predict(X_data)
```

Regression can be done with KNeighborsRegressor.

intel® Nervana™ AI Academy