K – Nearest Neighbors

Classification Algorithm

Predicts category or class labels for data points

A type of supervised machine learning algorithm

Learns from labeled training data for predictions

Outputs discrete values like "yes" or "no"

Common in spam filters, disease detection, etc.

Key task: classify into predefined categories

Types of Classification Algorithms

Logistic Regression – simple, interpretable, linear boundaries

Decision Trees – splits data into rule-based branches

Random Forest – multiple decision trees combined

Support Vector Machine (SVM) – separates with optimal margin

K-Nearest Neighbors (KNN) – based on closest data points

Naive Bayes – based on Bayes' Theorem assumptions

Binary vs Multiclass Classification

- **Binary**: Two possible classes (e.g., spam or not)
- **Multiclass**: More than two output classes
- Binary is simpler, often a yes/no scenario
- Multiclass includes scenarios like classifying animal types
- One-vs-All technique used in multiclass models
- Algorithms can be adapted for both types

Real-World Use Case Example

Email Spam Detection: spam or not spam

Medical Diagnosis: disease present or absent

Sentiment Analysis: positive, neutral, or negative

Loan Approval: approved or rejected

Image Recognition: identify cats, dogs, etc.

Customer Churn Prediction: stay or leave

Key Benefits of Classification Algorithms

- Automate decision-making tasks efficiently
- Can handle large amounts of data
- Improve with more training data
- Adaptable to various domains and industries
- Useful for real-time predictions and alerts
- Enhance accuracy with proper tuning and features

What is Classification?

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

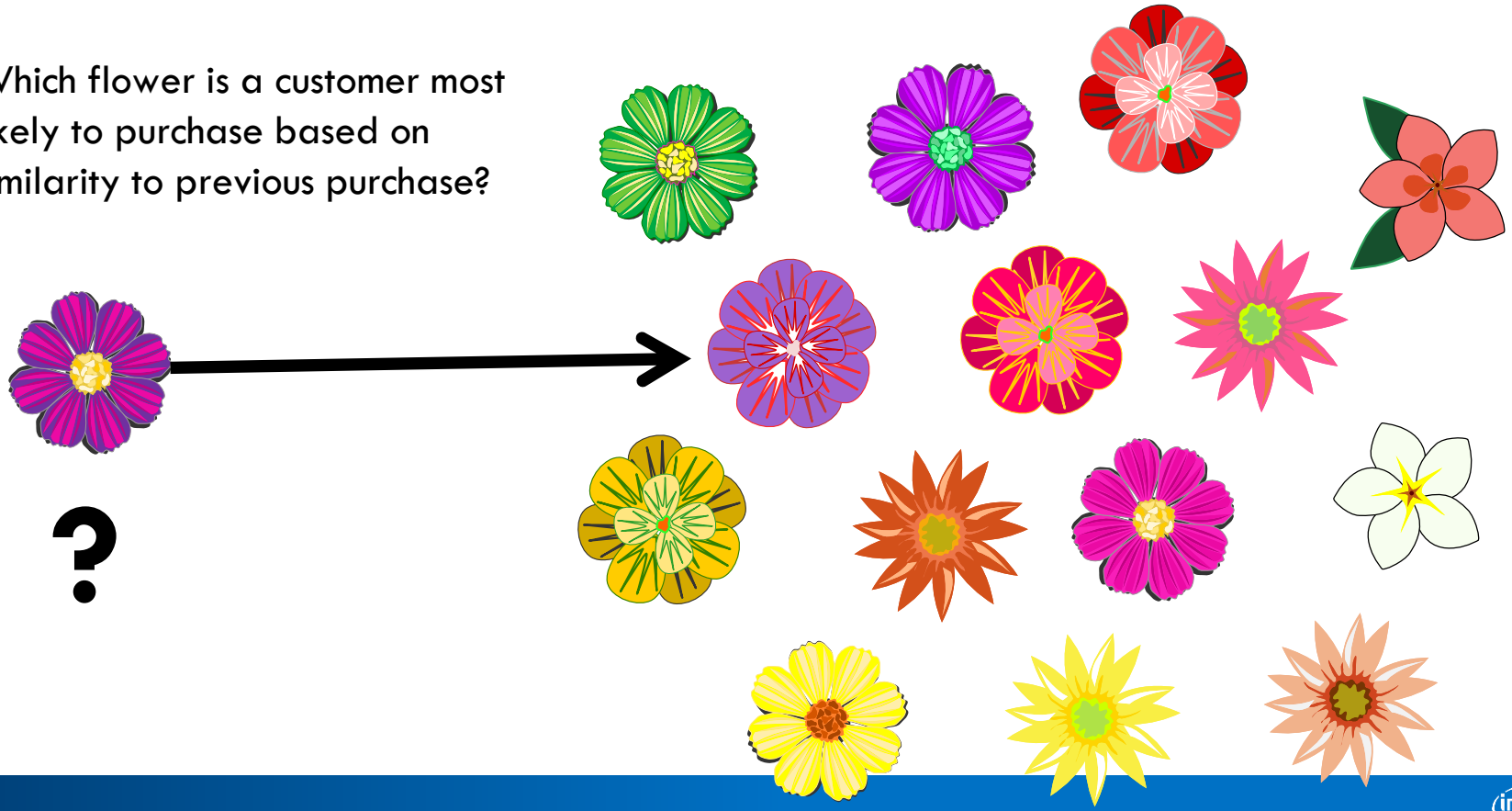
What is Classification?

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

?

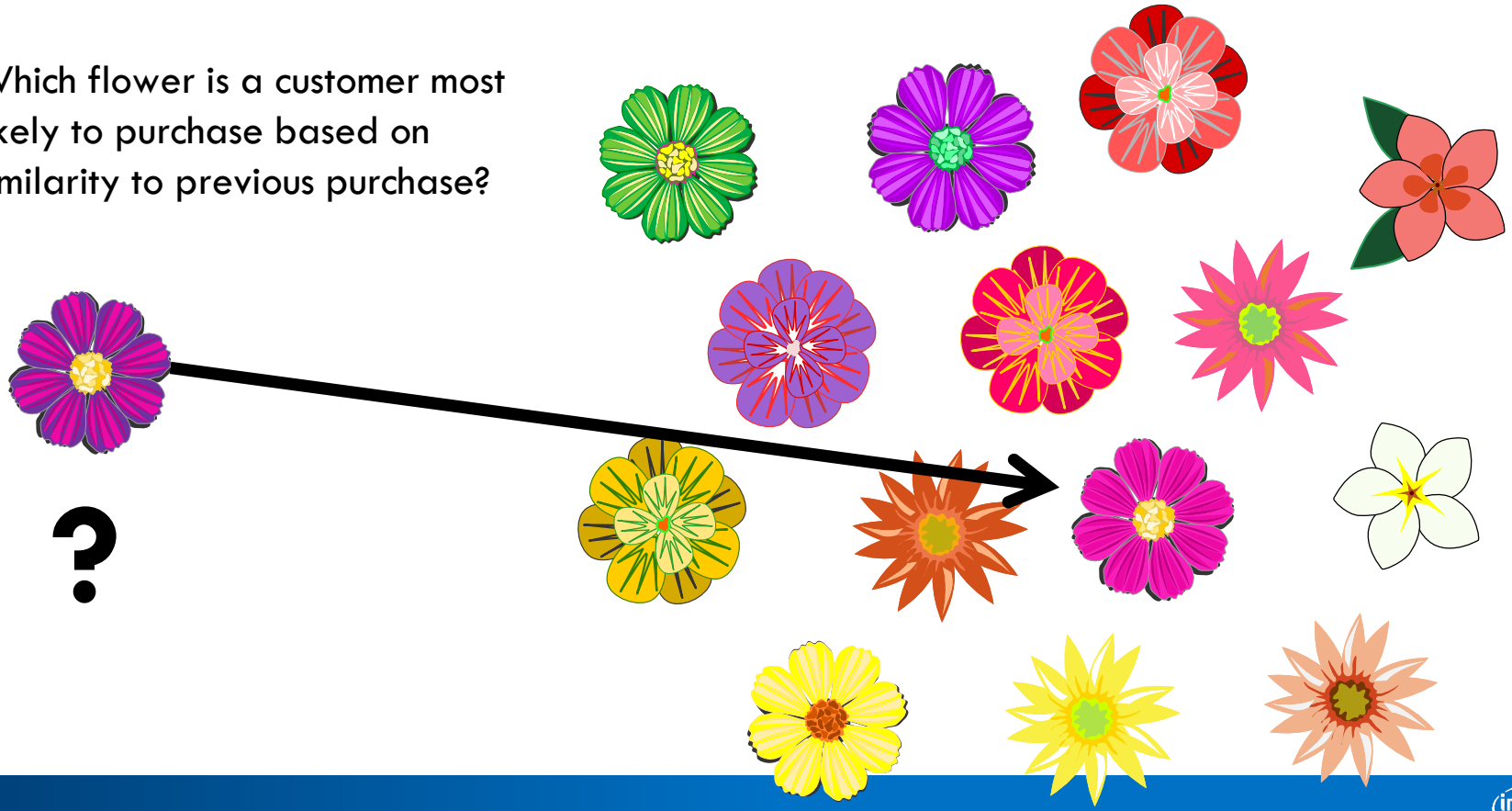
What is Classification?

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



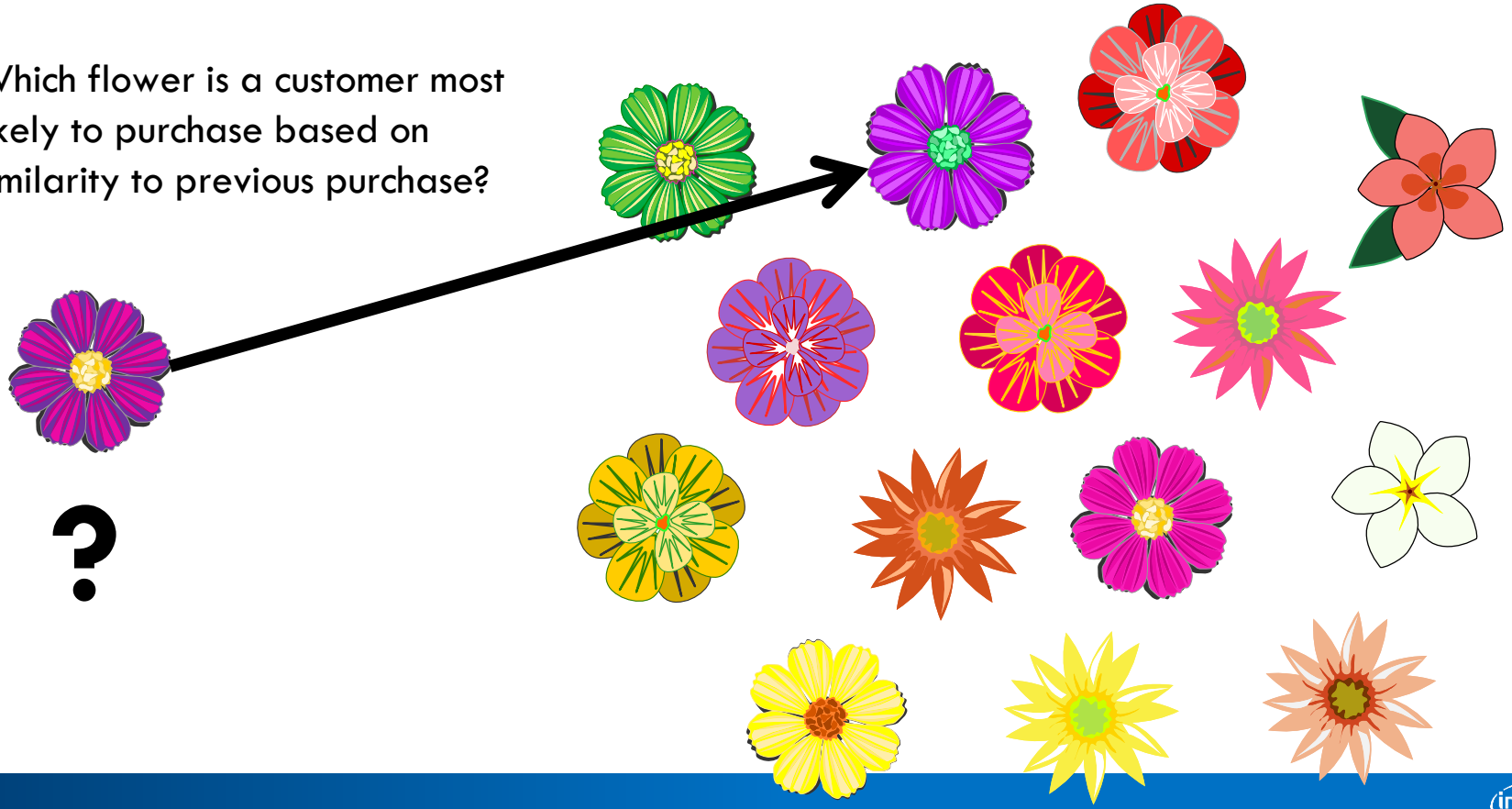
What is Classification?

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

KNN Classification

- A simple, non-parametric classification algorithm
- Classifies based on nearest training data points
- No model training phase — lazy learner
- Stores all training data for comparison
- Works for classification and regression tasks

How KNN Works

Choose a value for **K** (number of neighbors)

Calculate distance to all training points

Common distance: Euclidean distance formula

Select **K** nearest neighbors to test data

Count votes from neighbors' class labels

Assign most common label as prediction

Pros and Cons of KNN

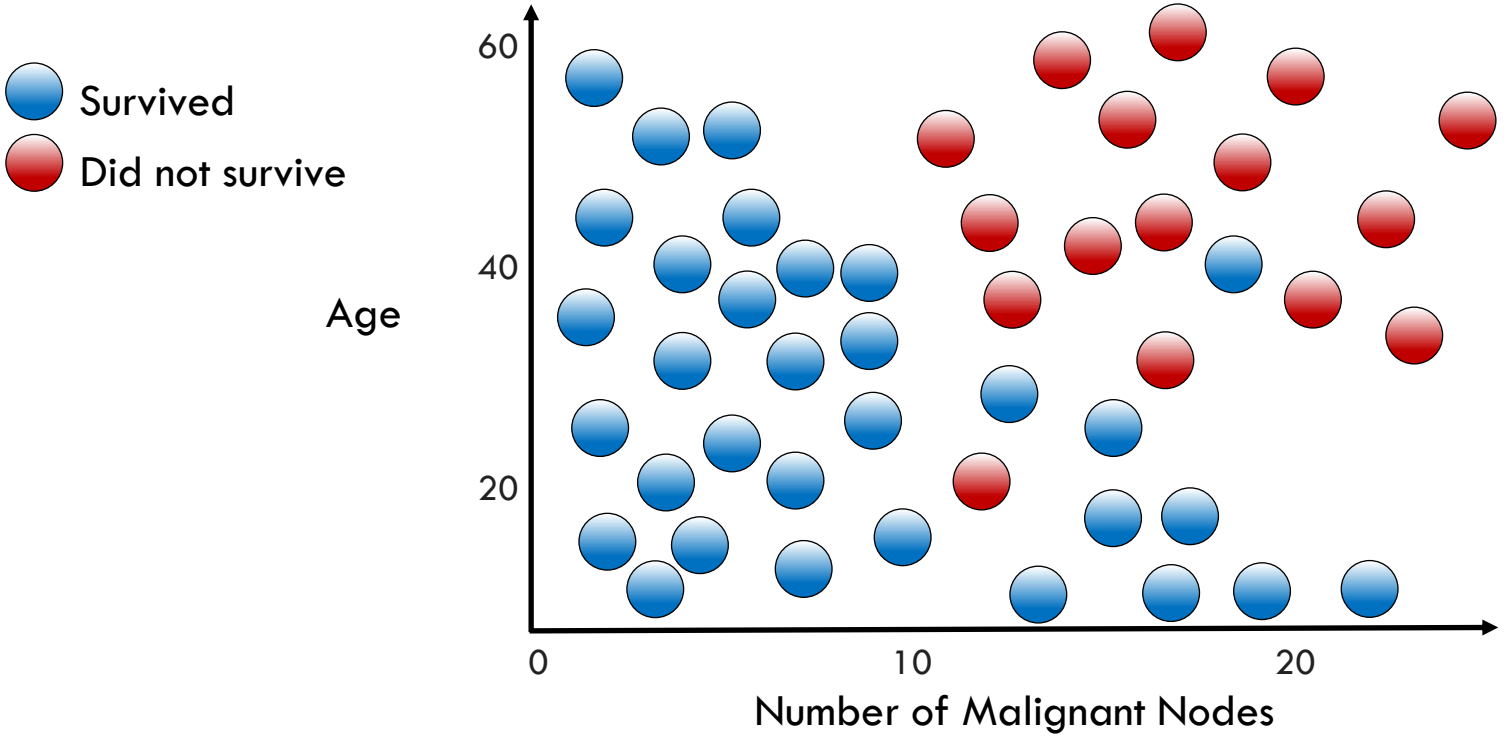
Pros

- Simple and easy to understand
- No training time required
- Adapts well to new data

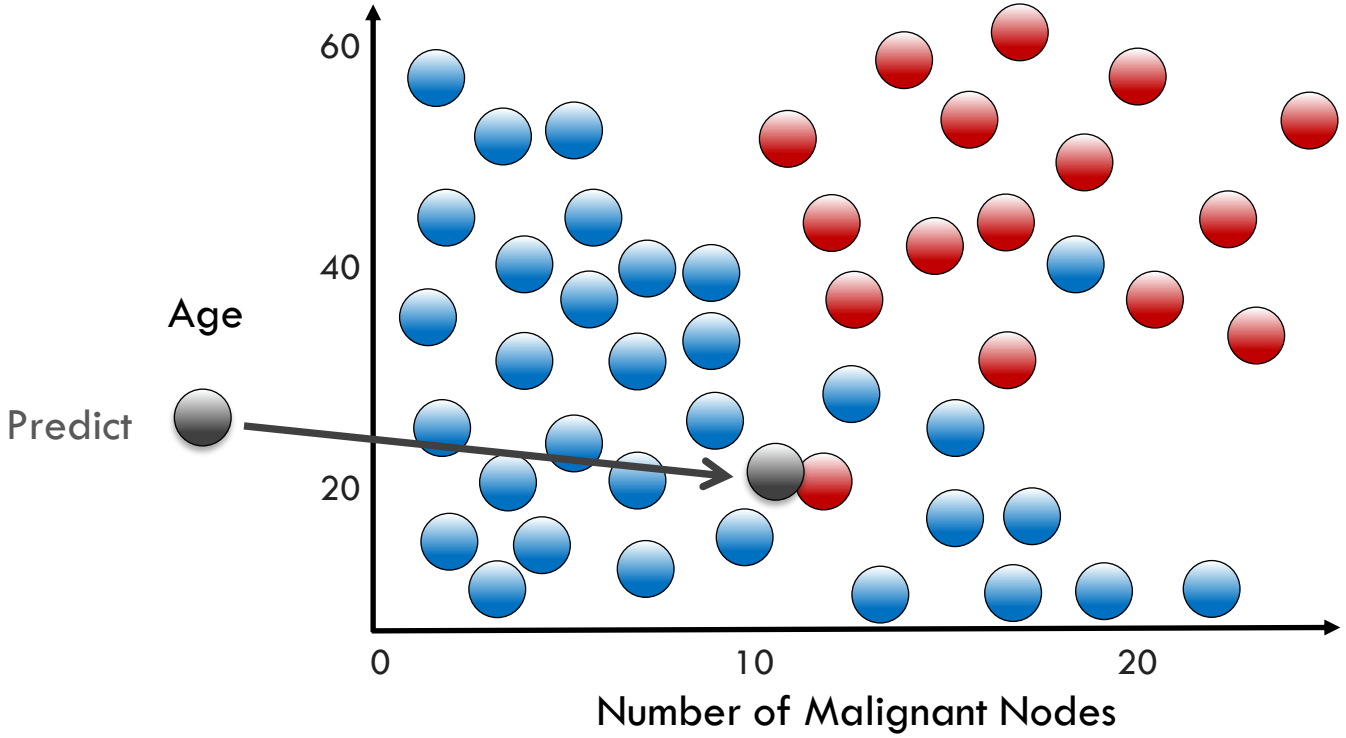
Cons

- Slow with large datasets
- Sensitive to irrelevant features
- Struggles with high-dimensional data

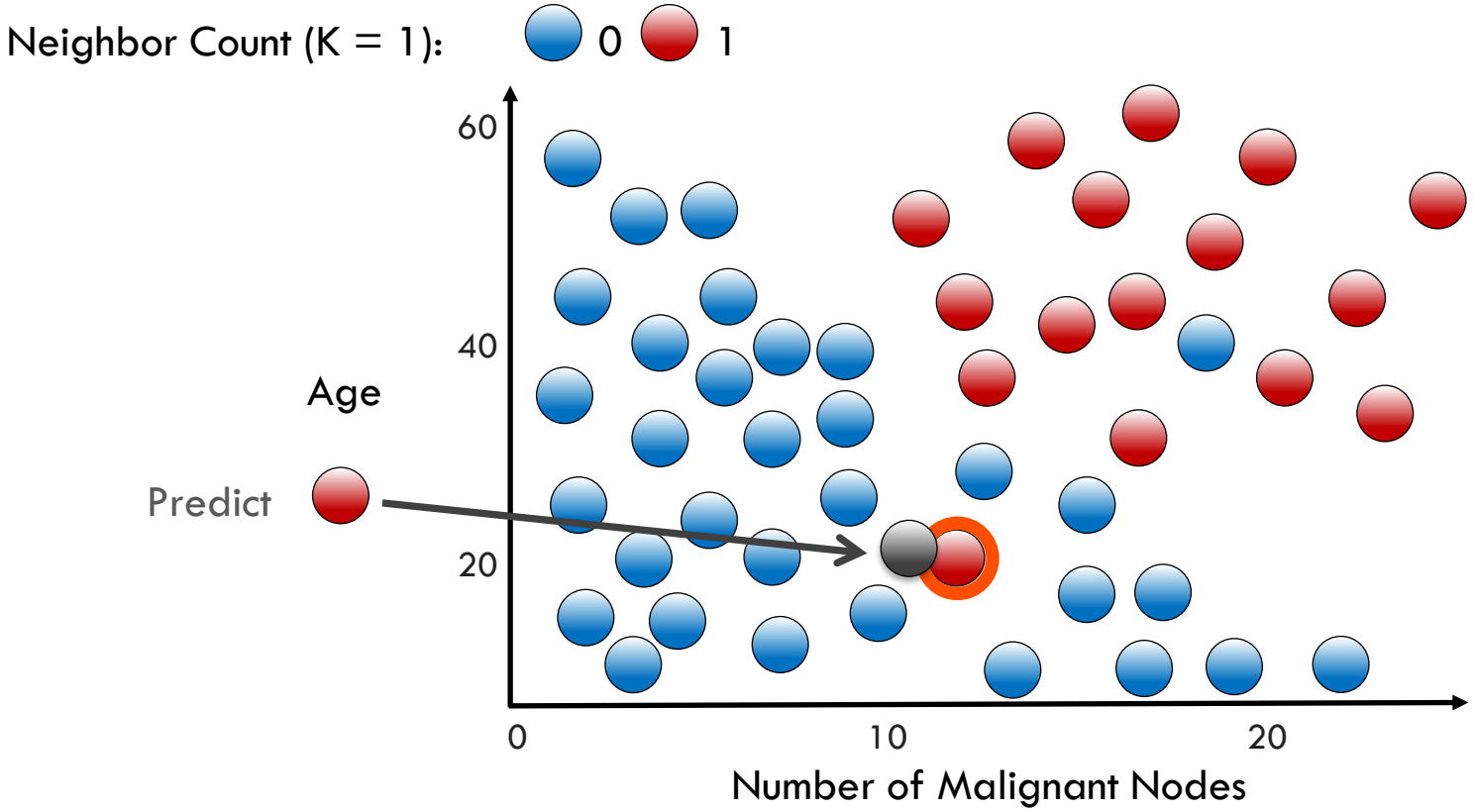
K Nearest Neighbors Classification



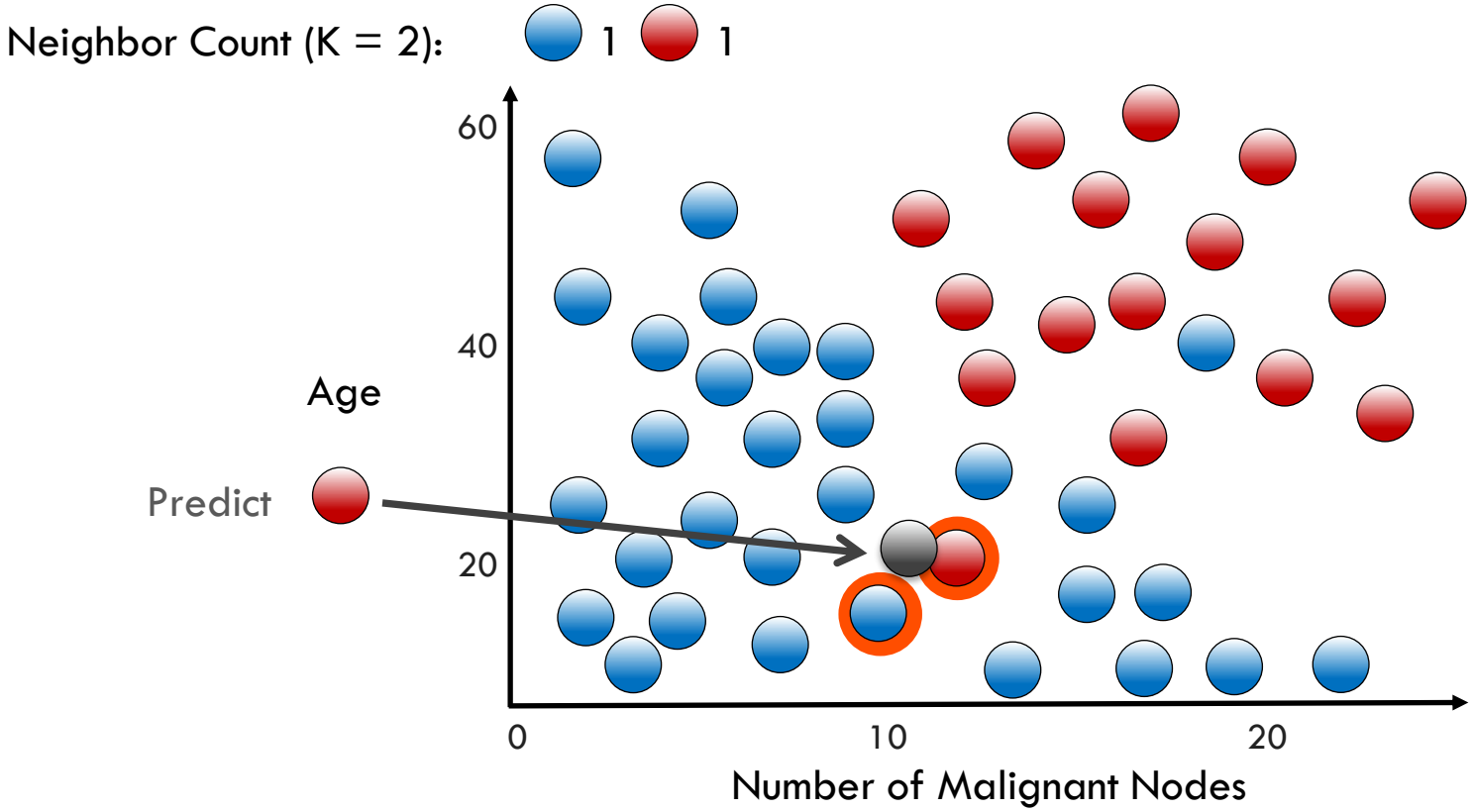
K Nearest Neighbors Classification



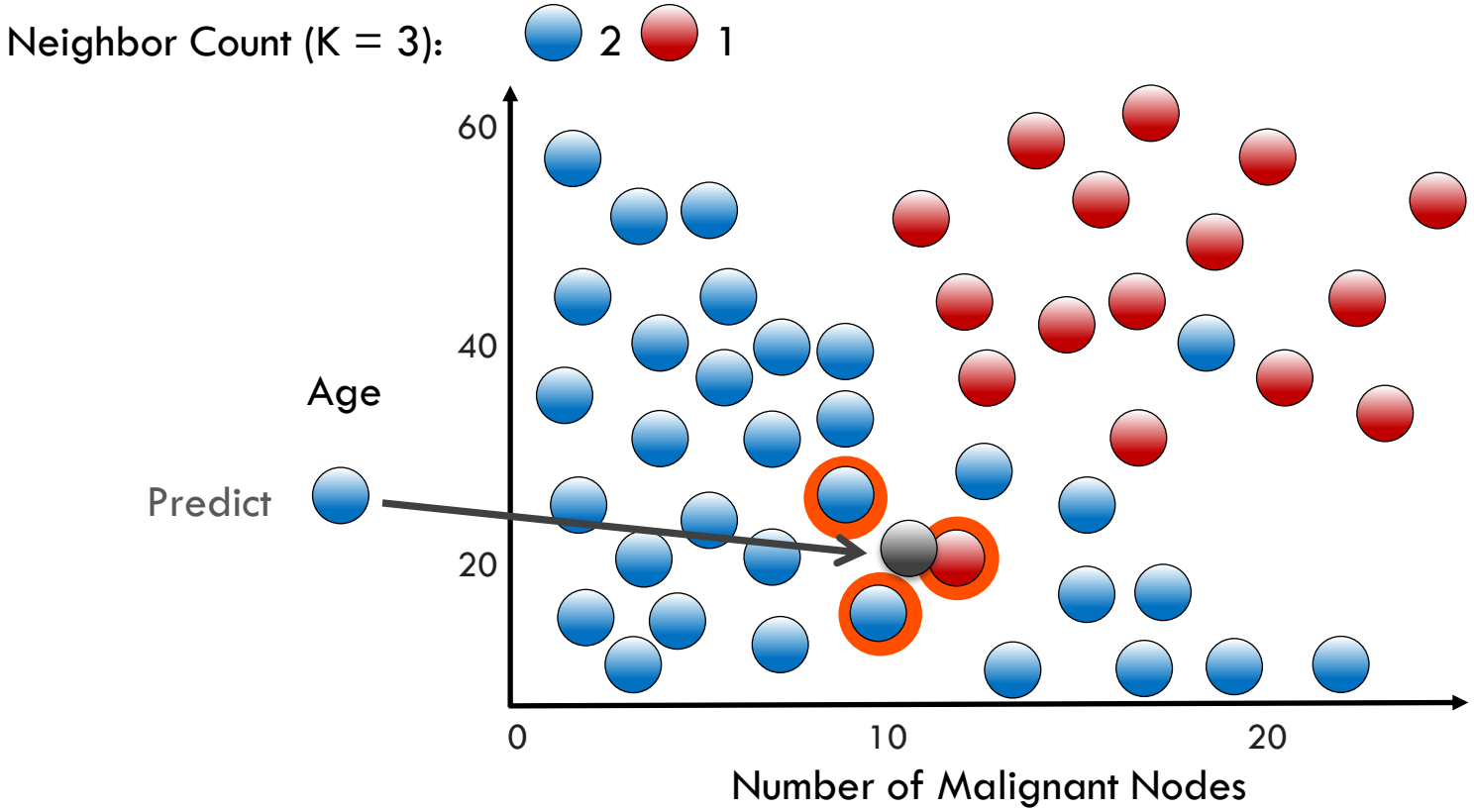
K Nearest Neighbors Classification



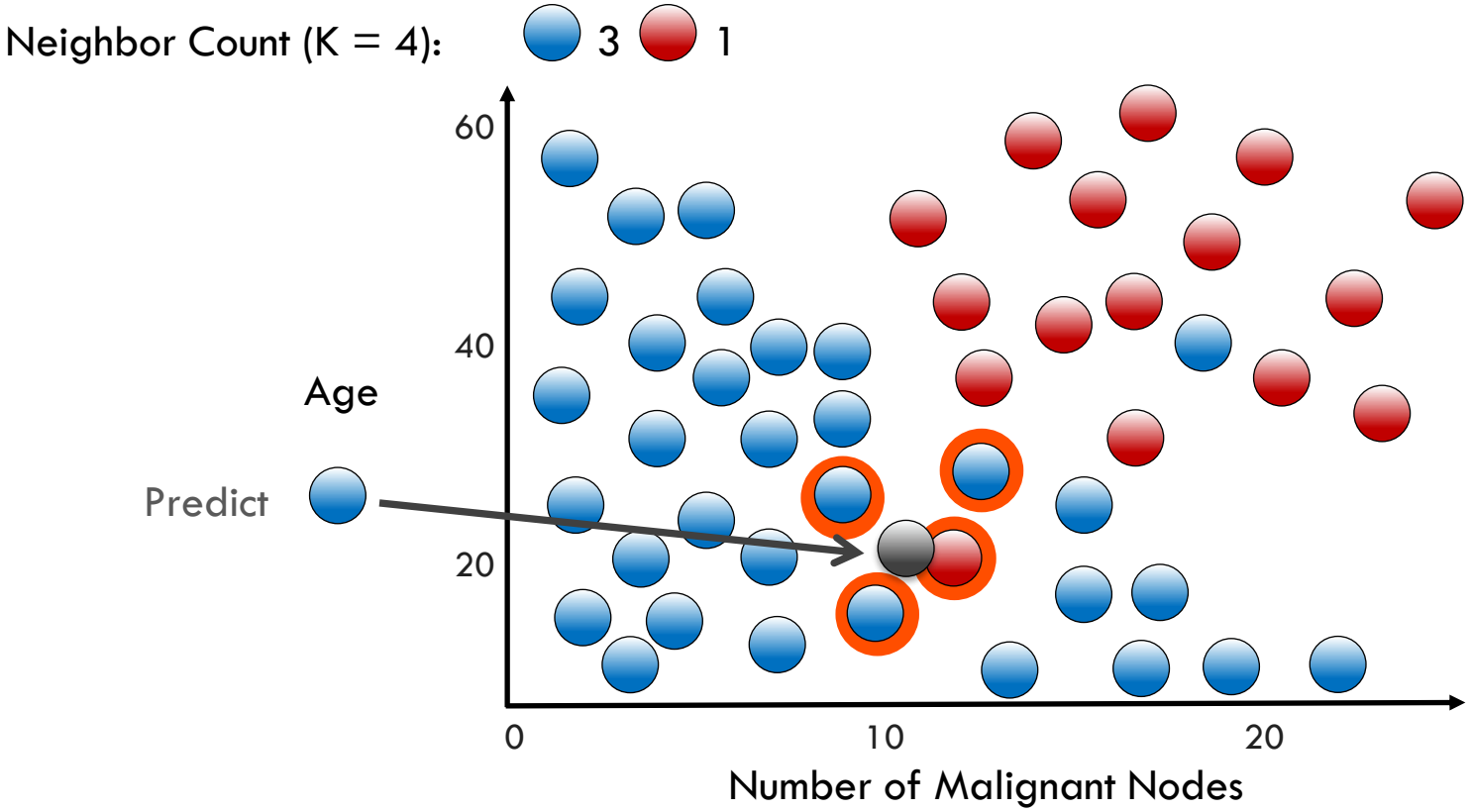
K Nearest Neighbors Classification



K Nearest Neighbors Classification



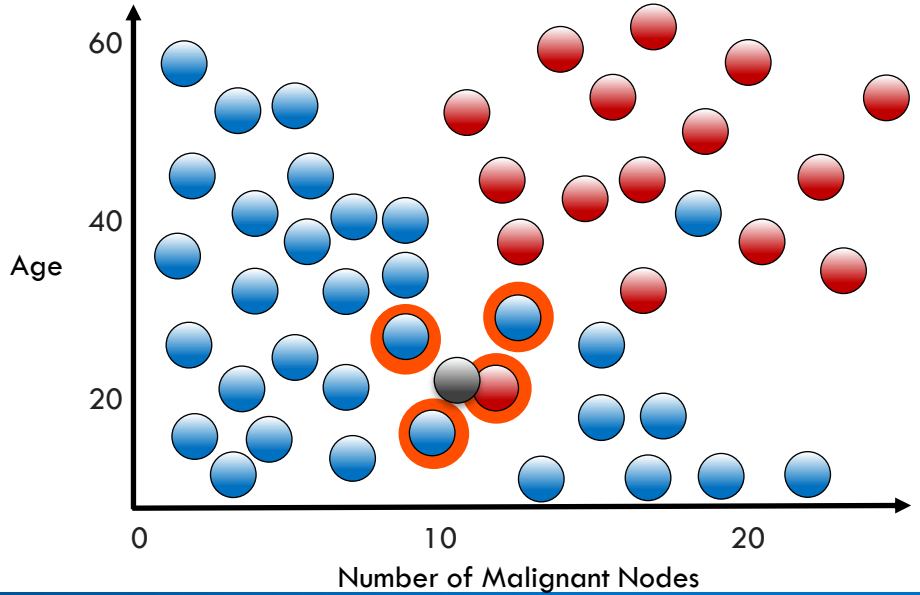
K Nearest Neighbors Classification



What is Needed to Select a KNN Model?

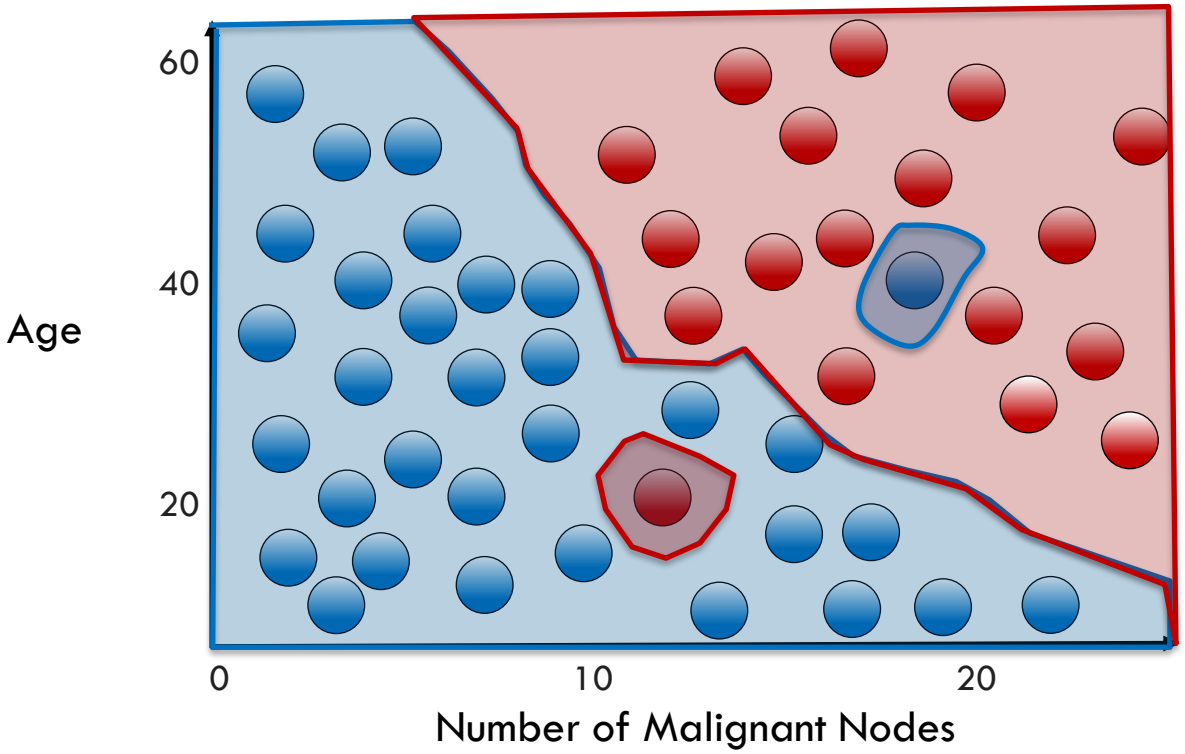
What is Needed to Select a KNN Model?

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



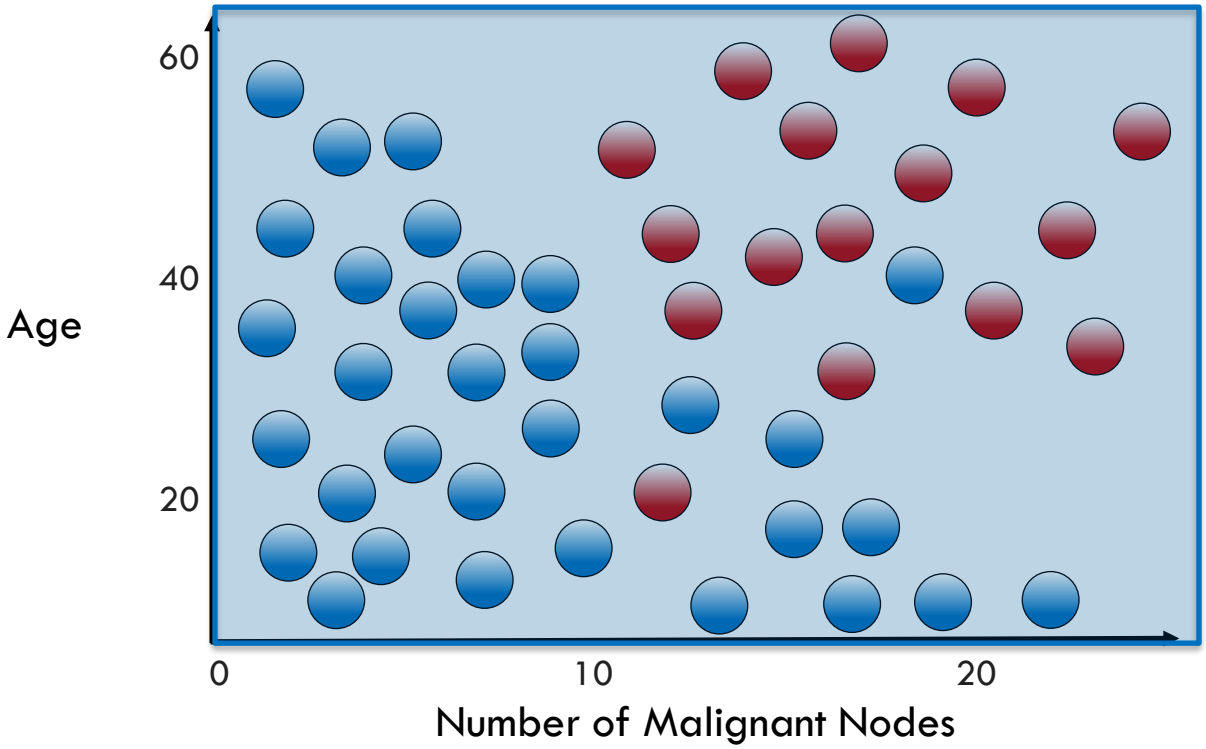
K Nearest Neighbors Decision Boundary

K = 1

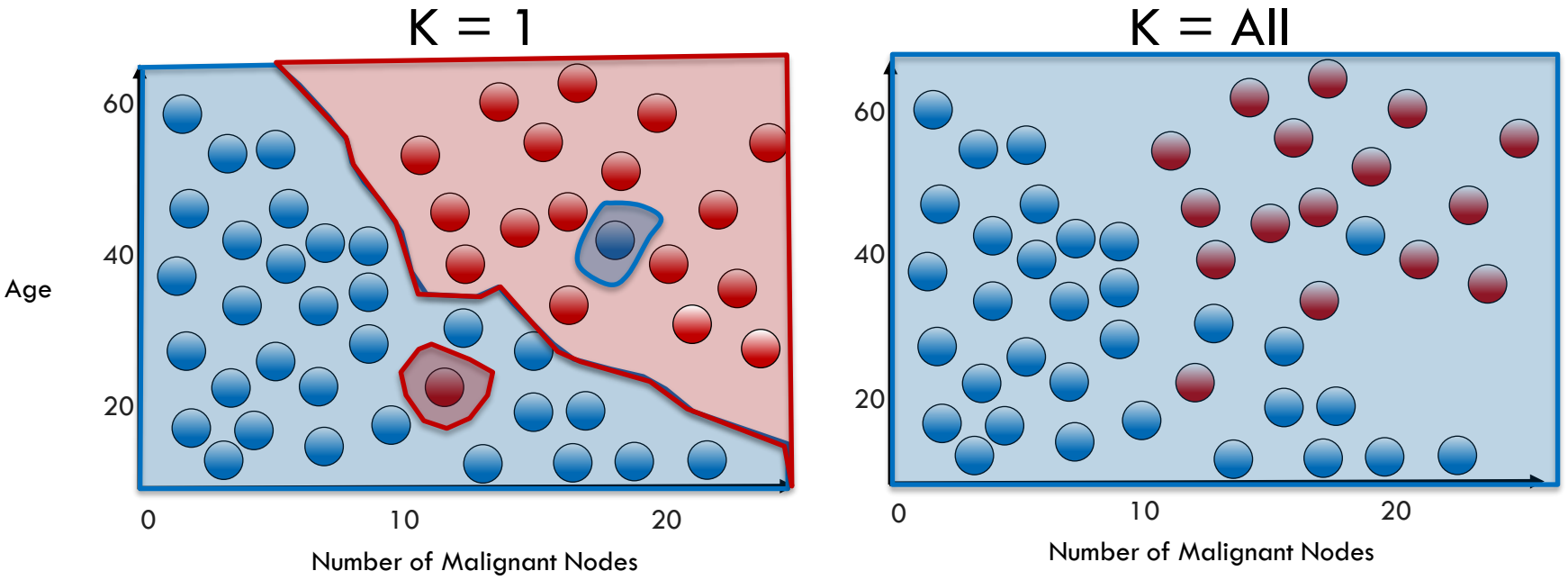


K Nearest Neighbors Decision Boundary

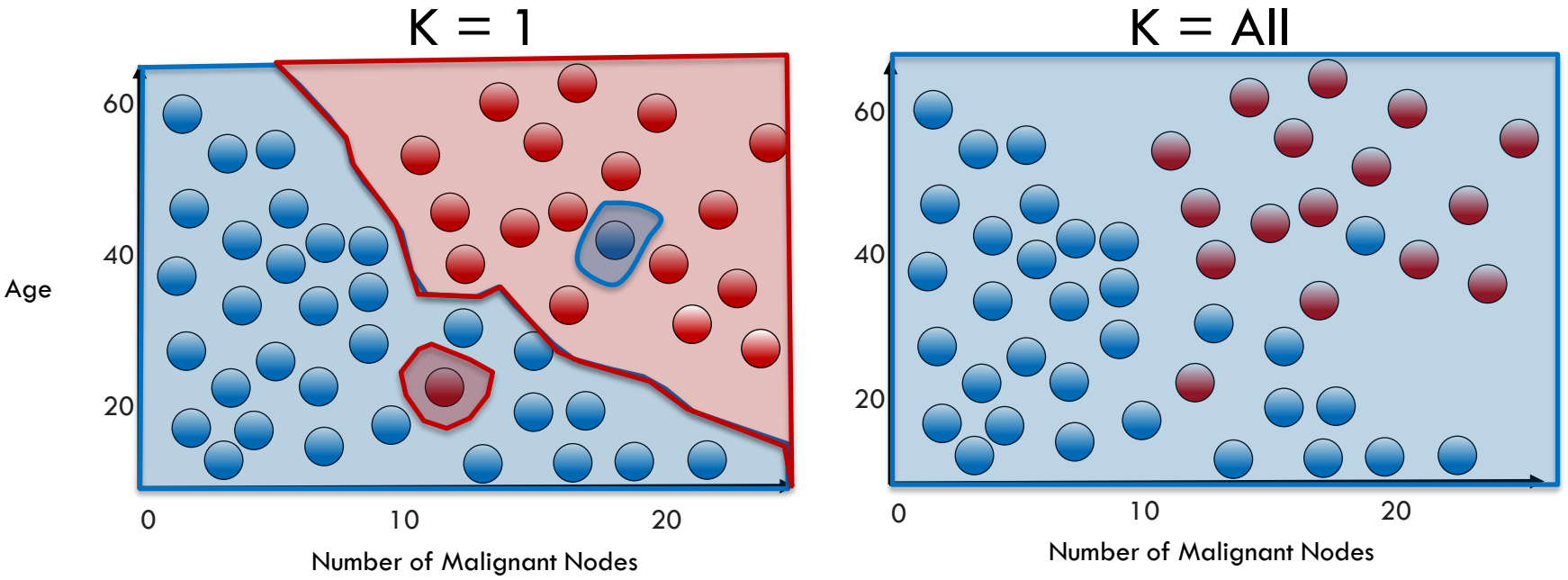
K = All



Value of 'K' Affects Decision Boundary

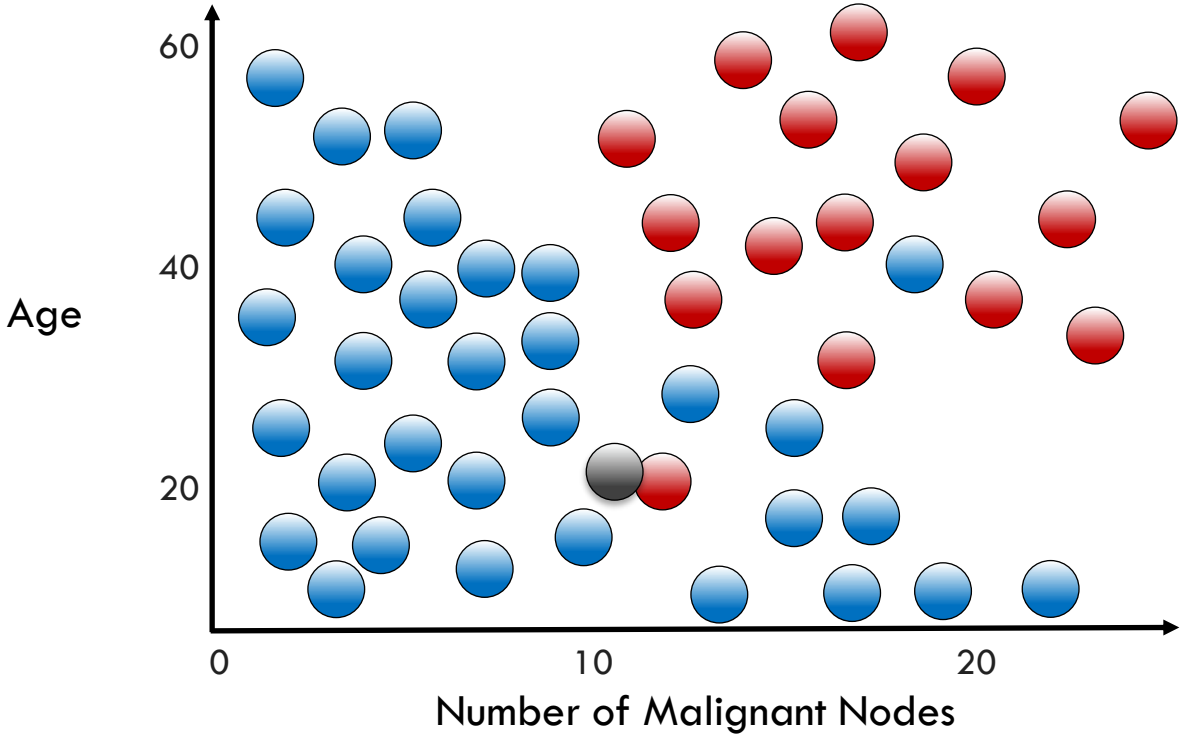


Value of 'K' Affects Decision Boundary

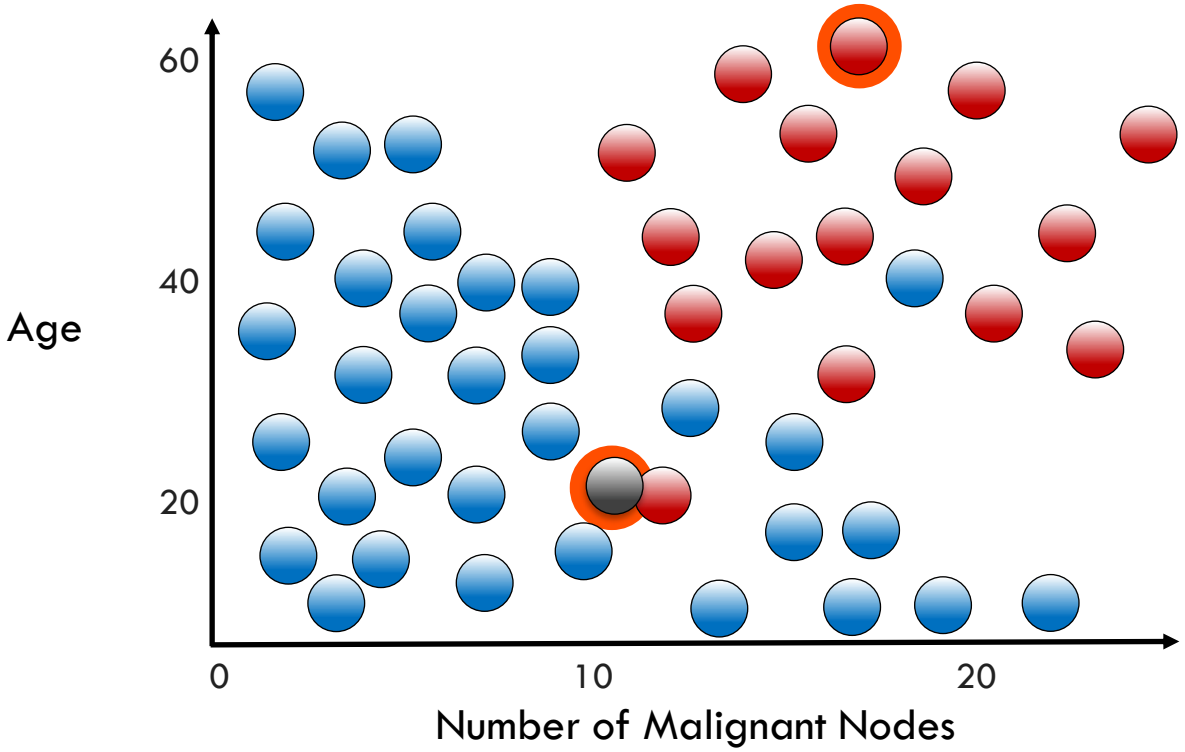


Methods for determining 'K' will be discussed in next lesson

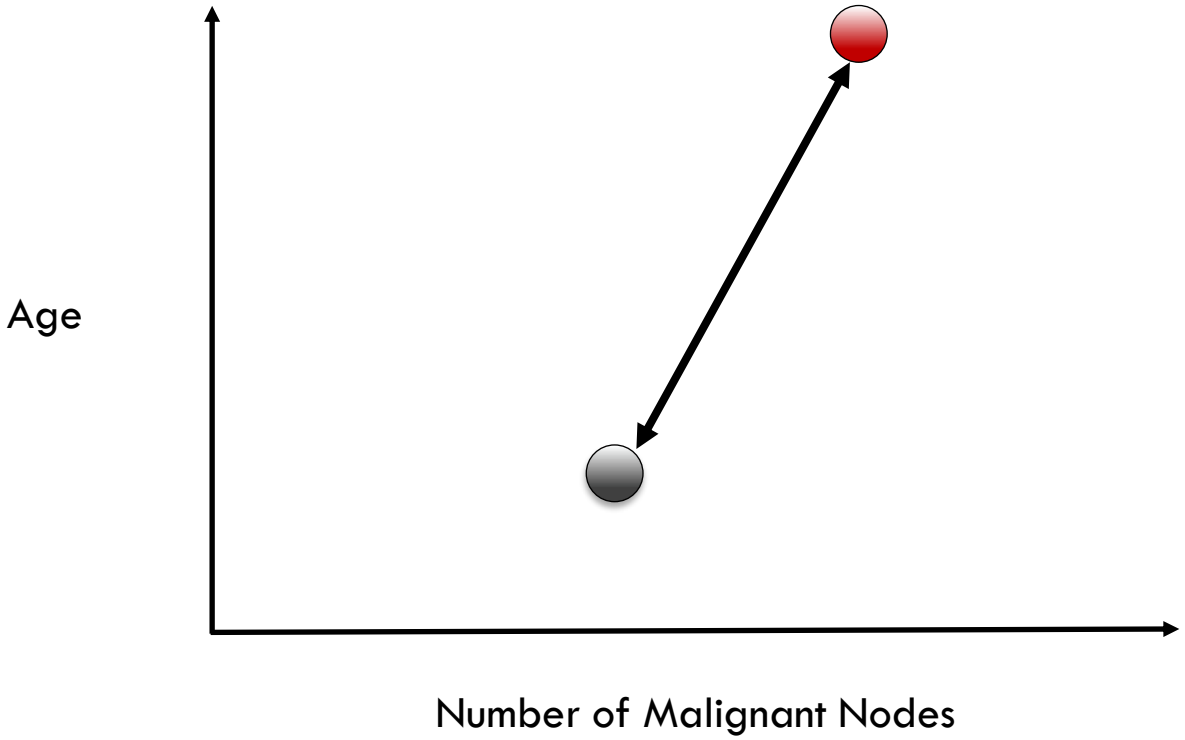
Measurement of Distance in KNN



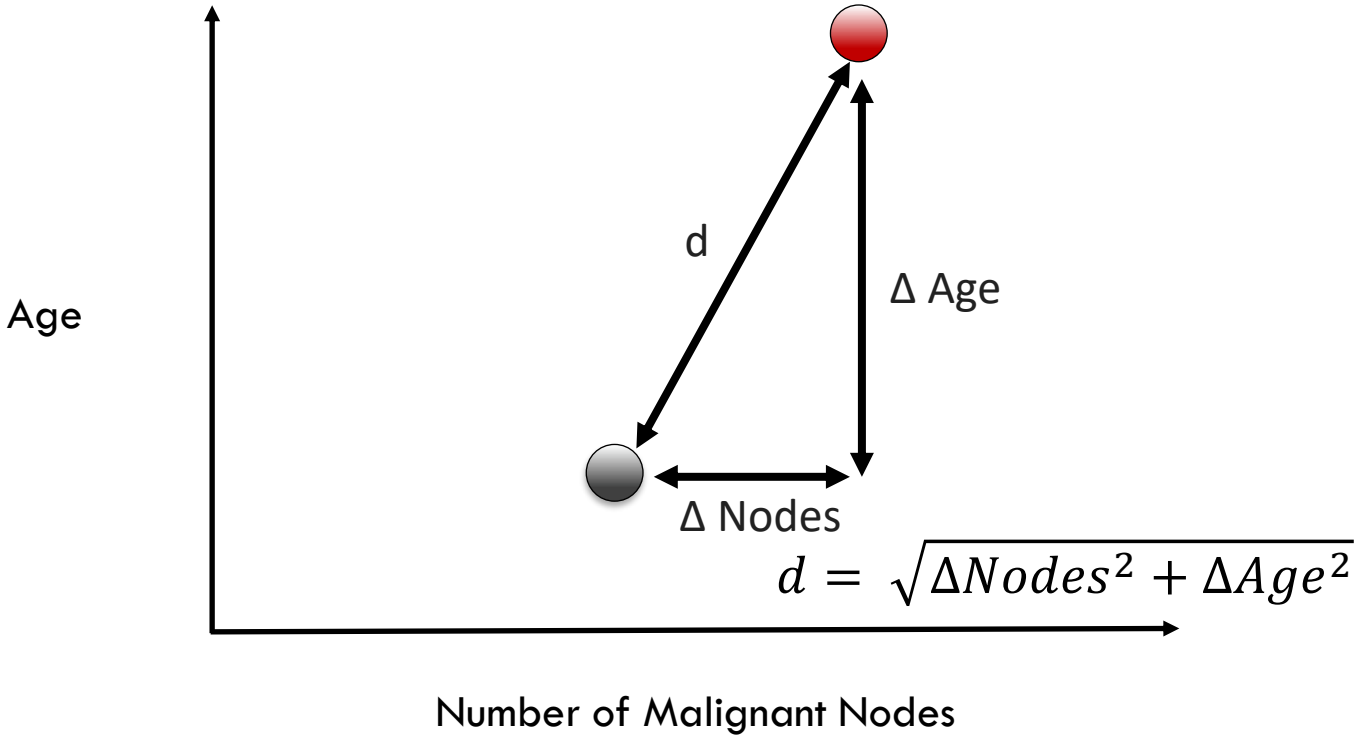
Measurement of Distance in KNN



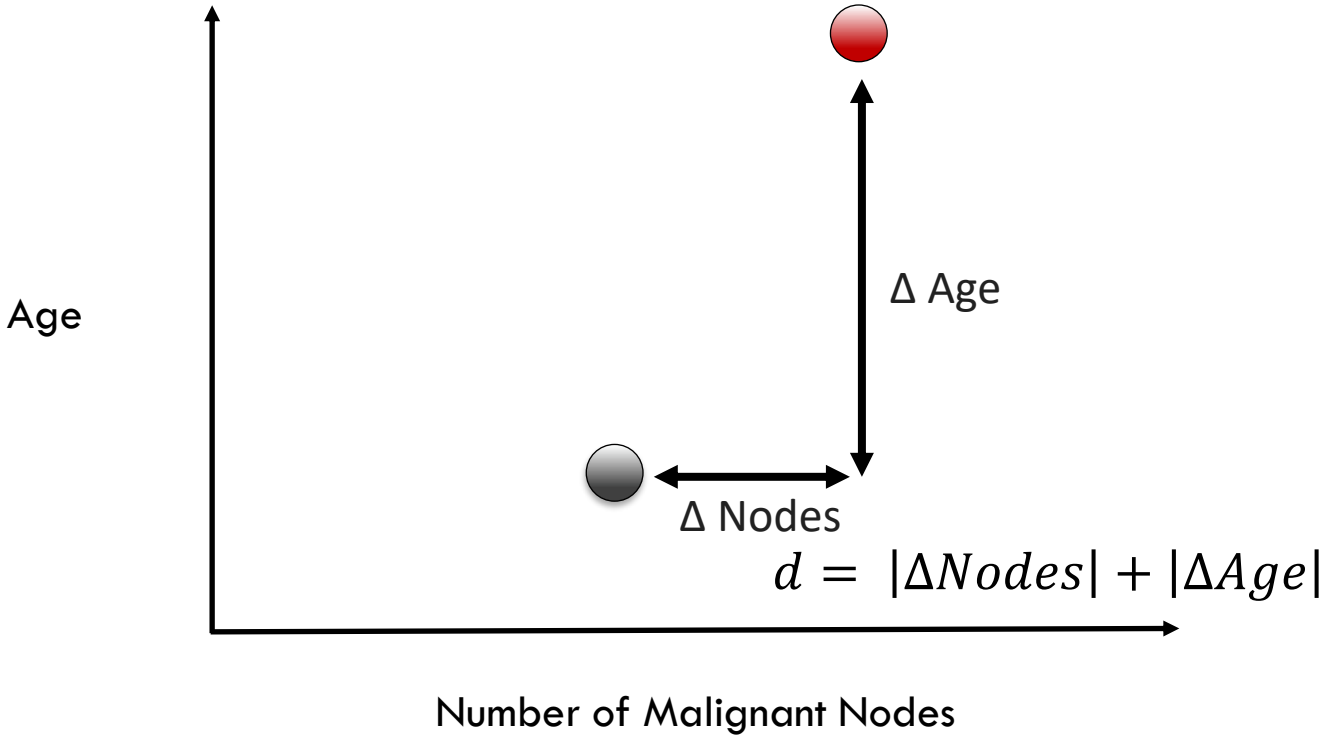
Euclidean Distance



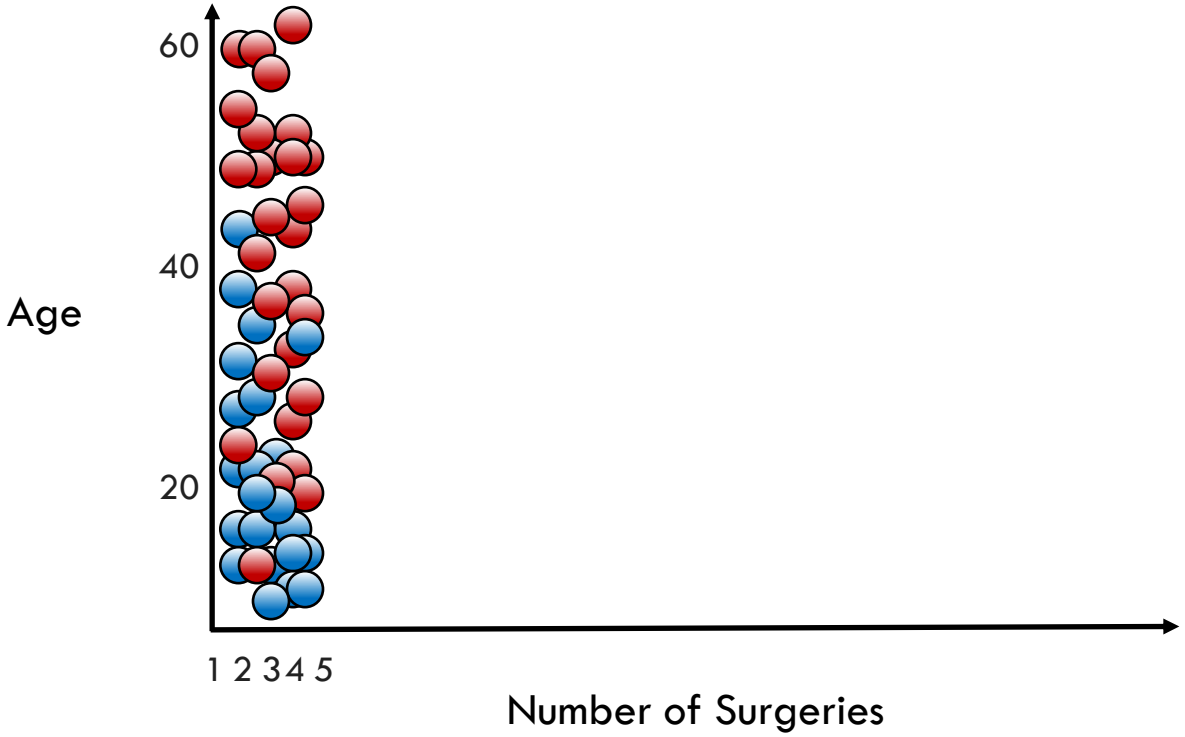
Euclidean Distance (L2 Distance)



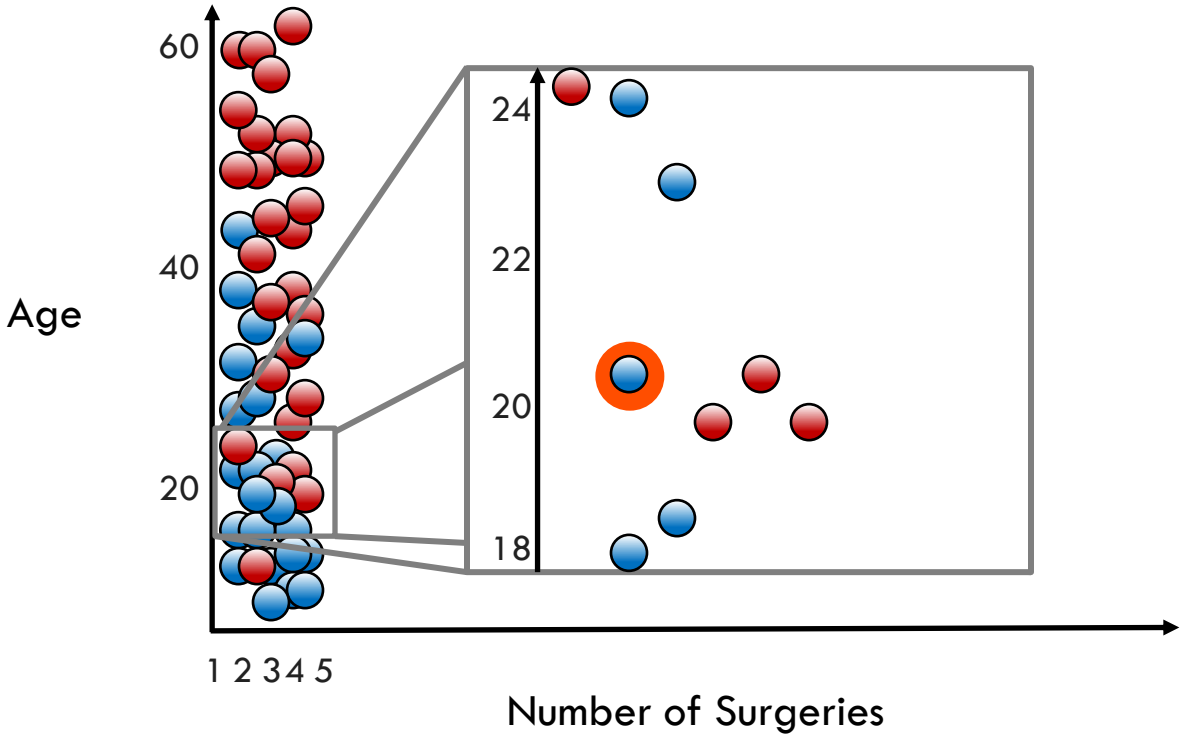
Manhattan Distance (L1 or City Block Distance)



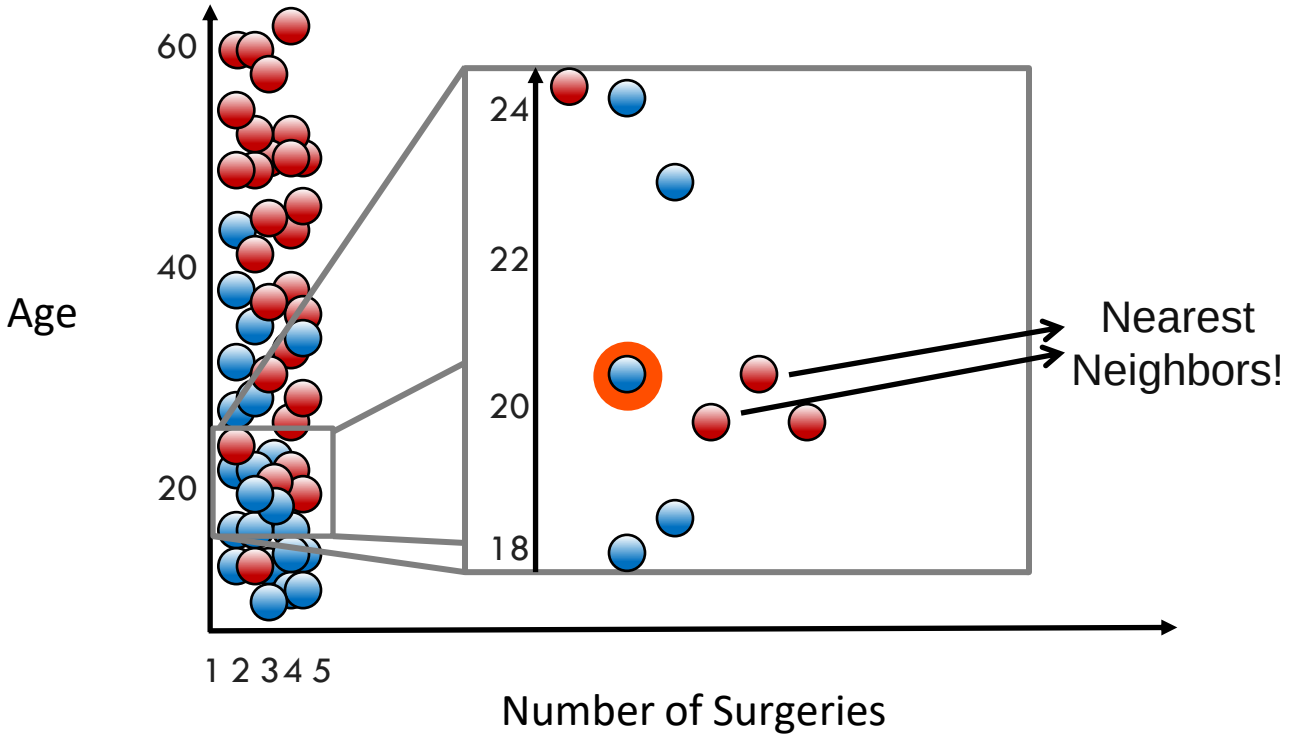
Scale is Important for Distance Measurement



Scale is Important for Distance Measurement

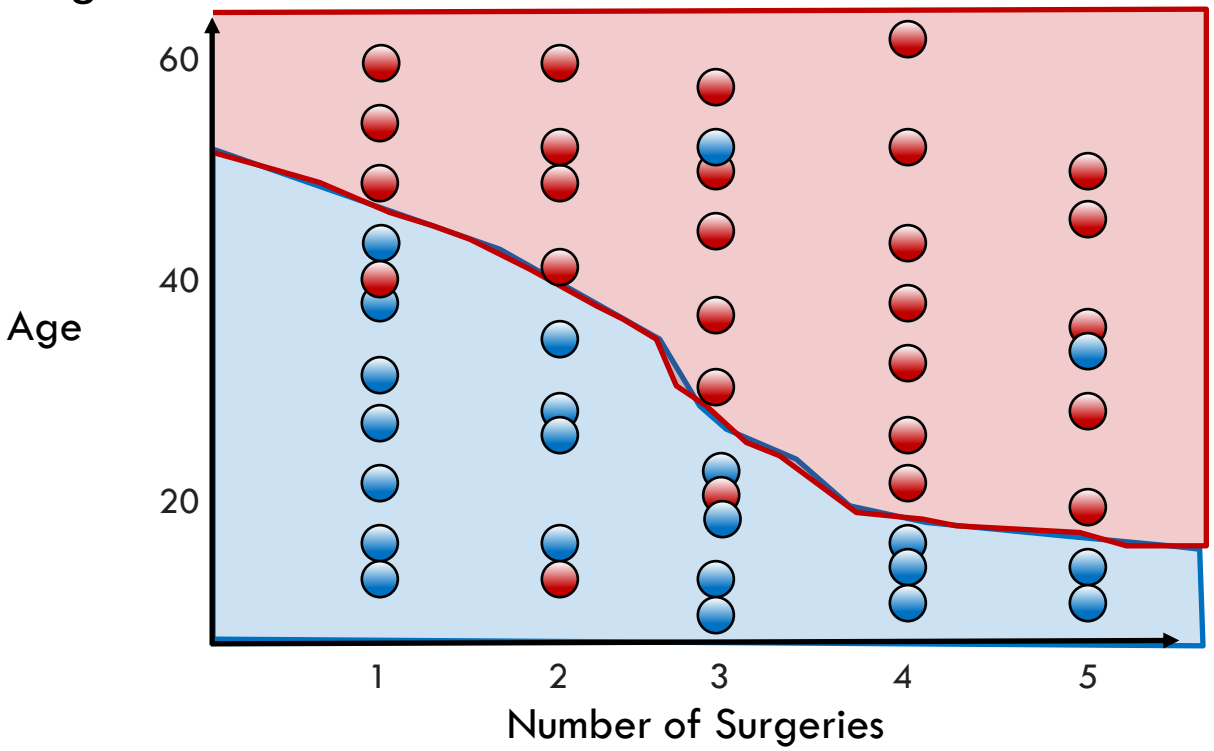


Scale is Important for Distance Measurement



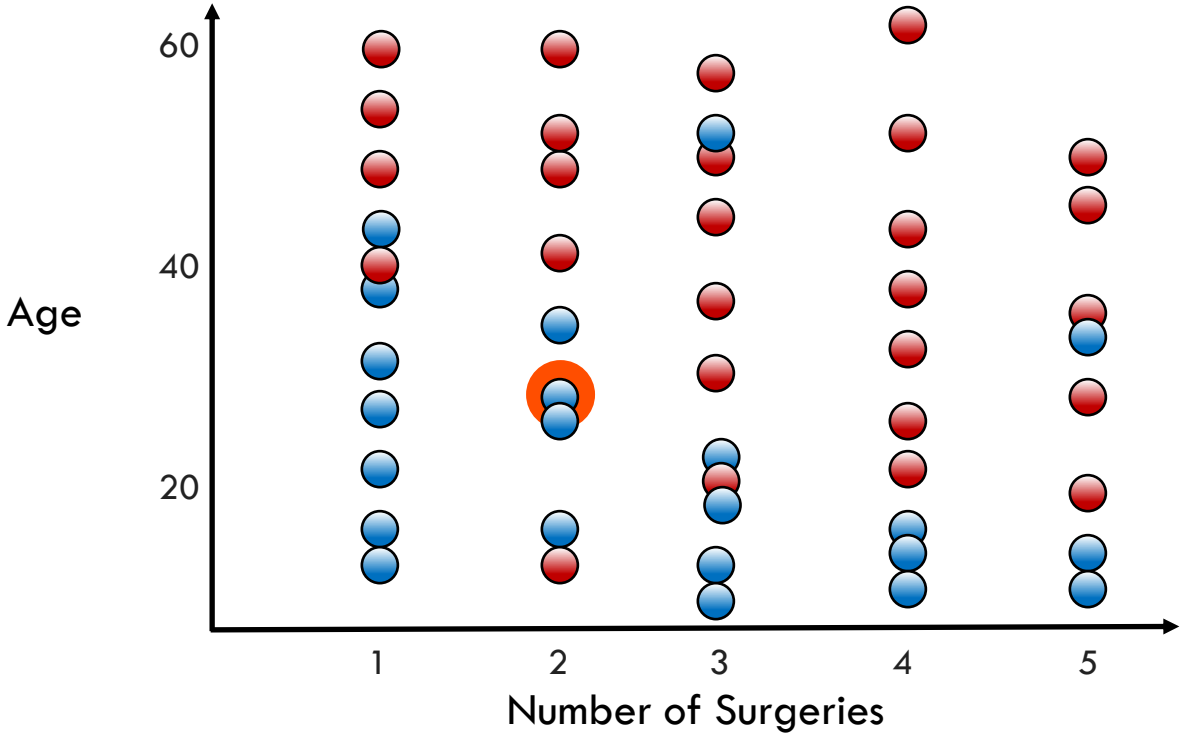
Scale is Important for Distance Measurement

"Feature Scaling"



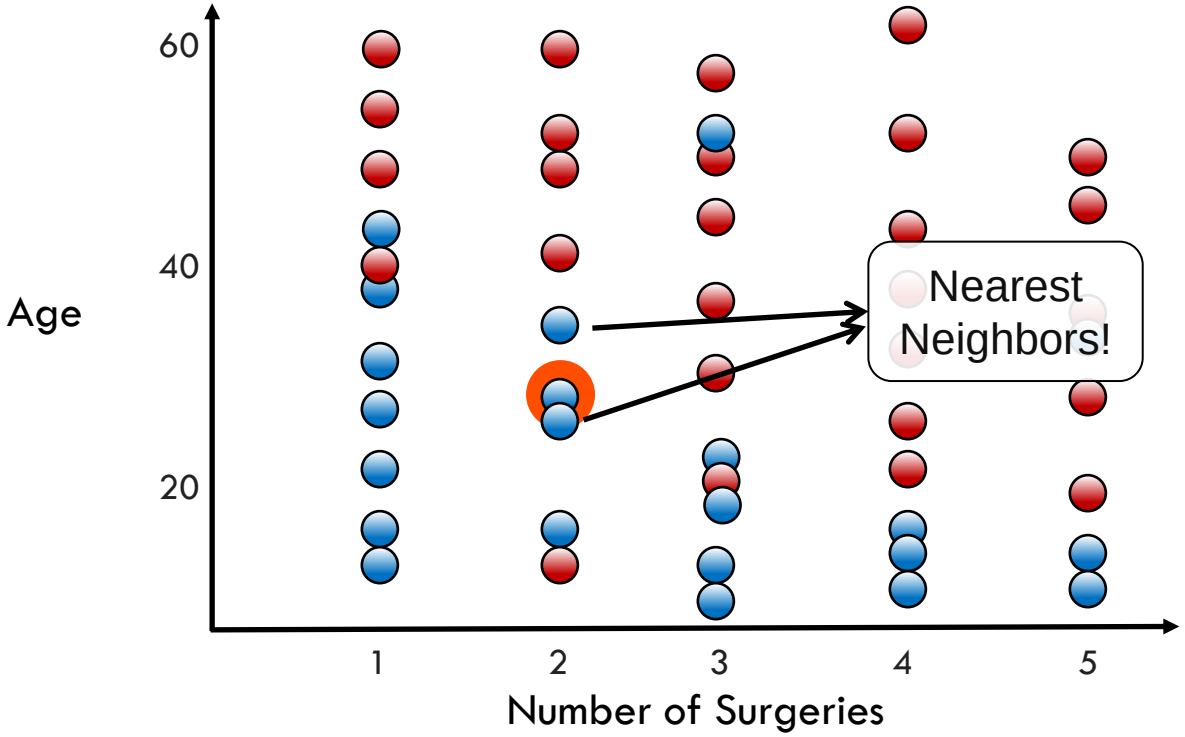
Scale is Important for Distance Measurement

"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 = StdSc.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()
```

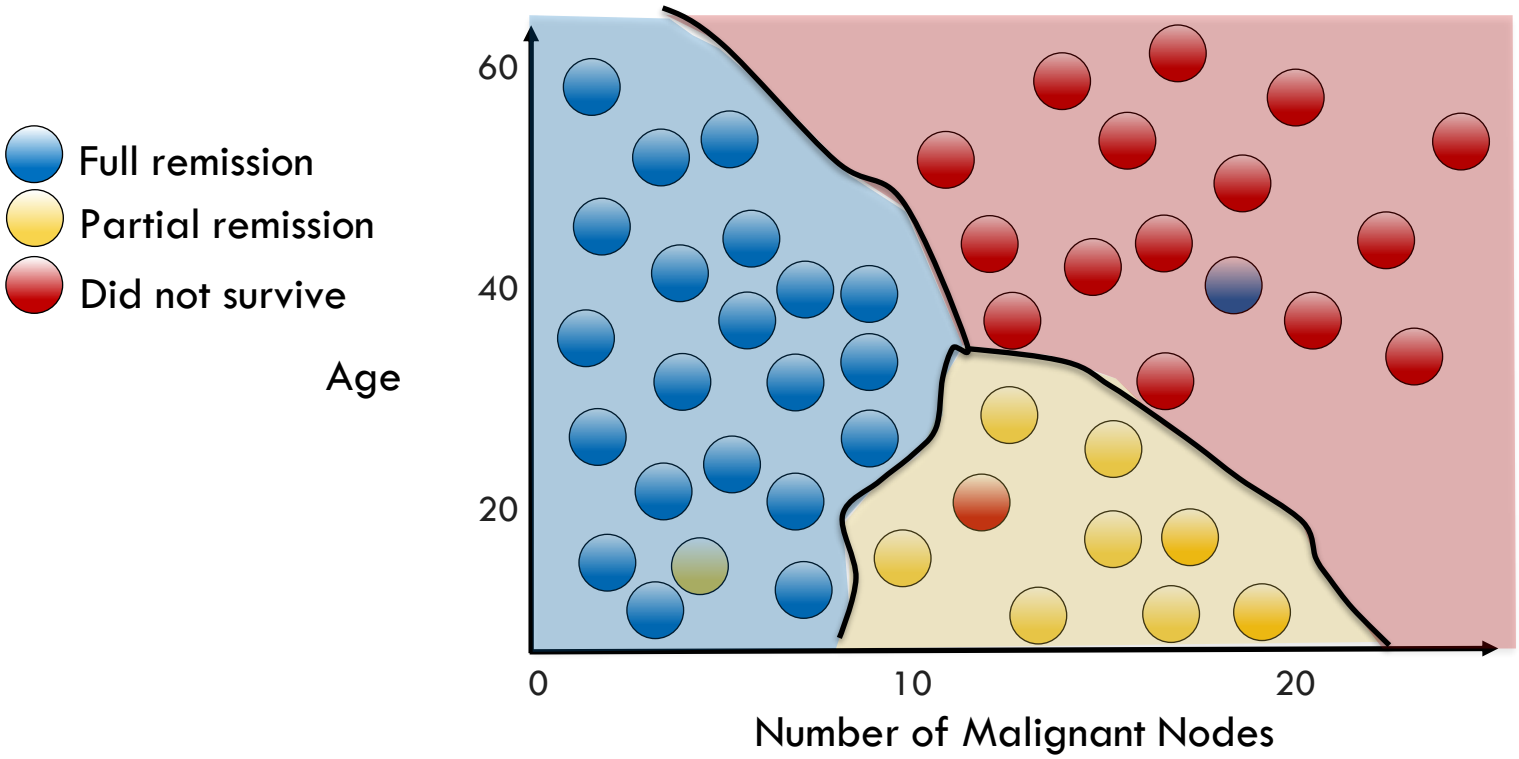
Fit the scaling parameters and then transform the data

```
StdSc = StdSc.fit(X_data)
X_scaled = StdSc.transform(X_data)
```

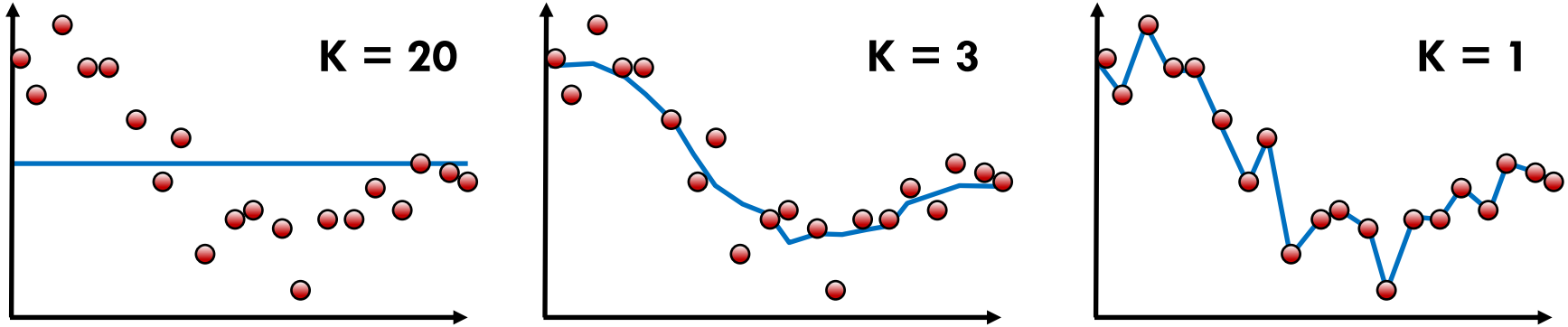
Other scaling methods exist: MinMaxScaler, MaxAbsScaler.

Multiclass KNN Decision Boundary

K = 5



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.

Additional Resources for KNN

Effective k-nearest neighbor models for data classification enhancement

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01137-2#citeas

Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00973-y

KNeighborsClassifier

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

THANK YOU

- Source compiled from intel ai academy