Intro to Supervised Learning & KNN Classifier

What is Supervised Learning?

Supervised Learning

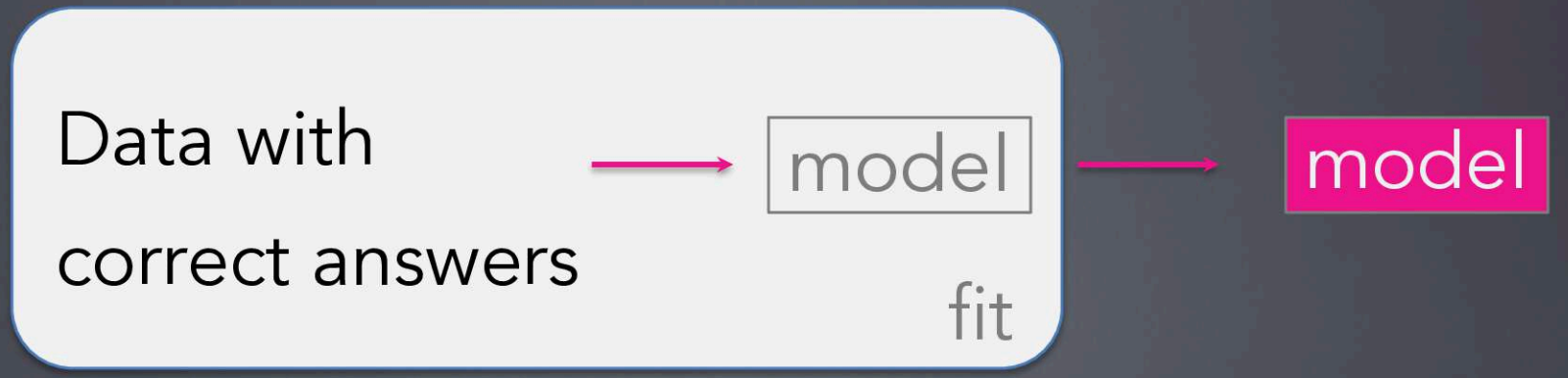
The main goal of supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen or future data.

Here the term supervised refers to a set of observations where the desired output signals (labels) are already known.

A classic example of a supervised learning algorithm is categorizing emails as spam or not-spam.

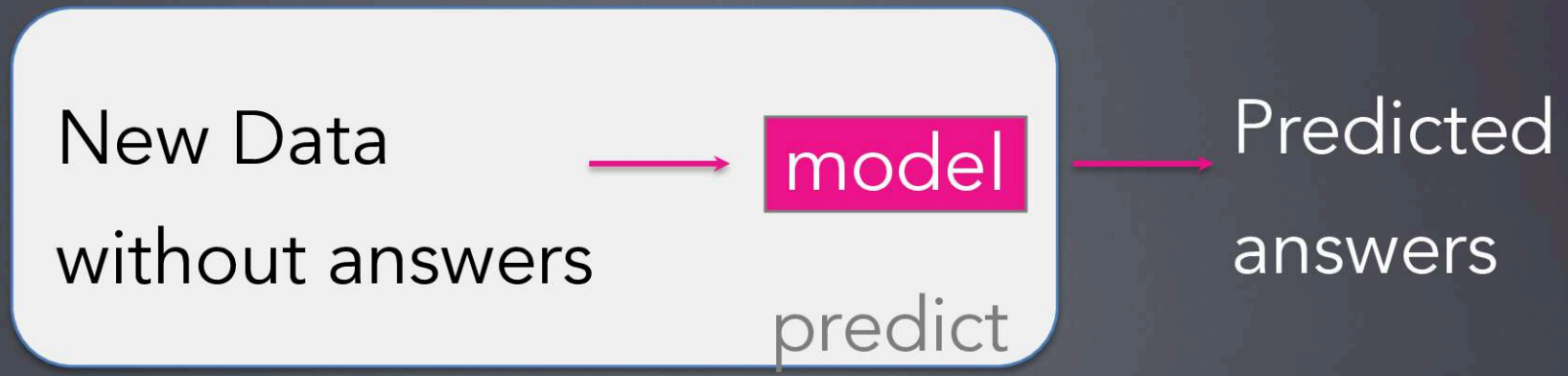
Supervised Learning

Use the labeled data to fit a machine learning model



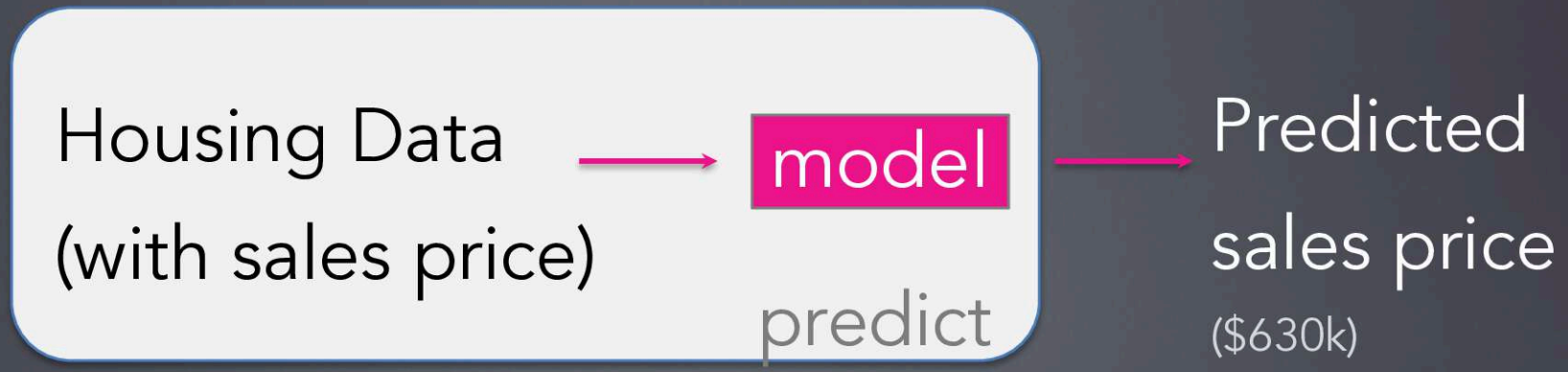
Supervised Learning

Once our model has been trained (and tested!), we can use it on unlabeled data to make new predictions.



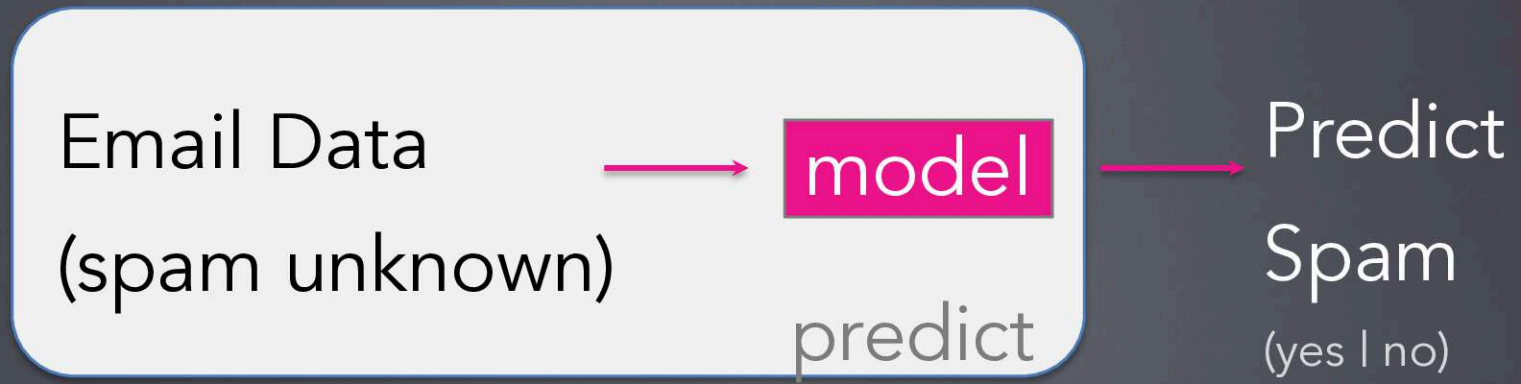
Supervised Learning

Regression: "Answers" from model are numeric



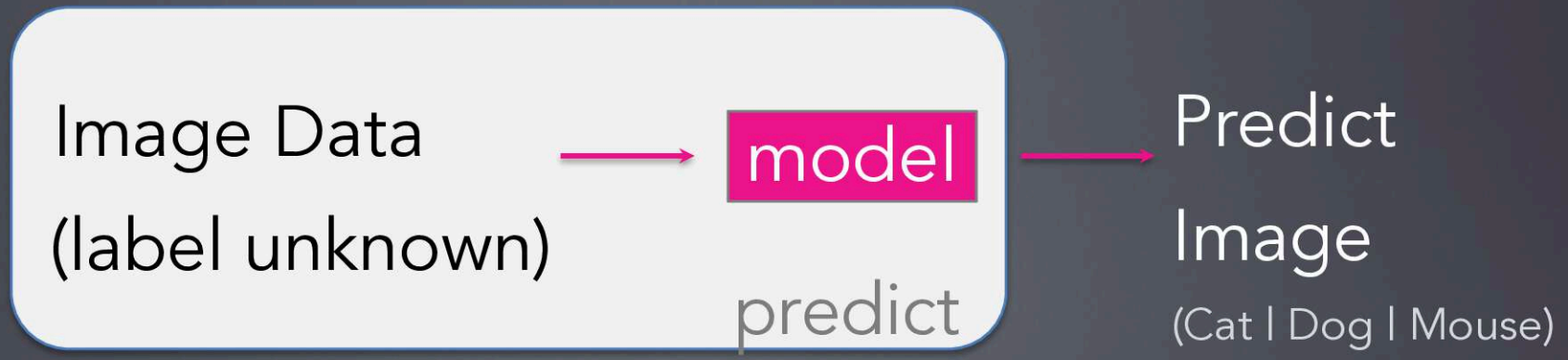
Supervised Learning

Classification: "Answers" from model are categories



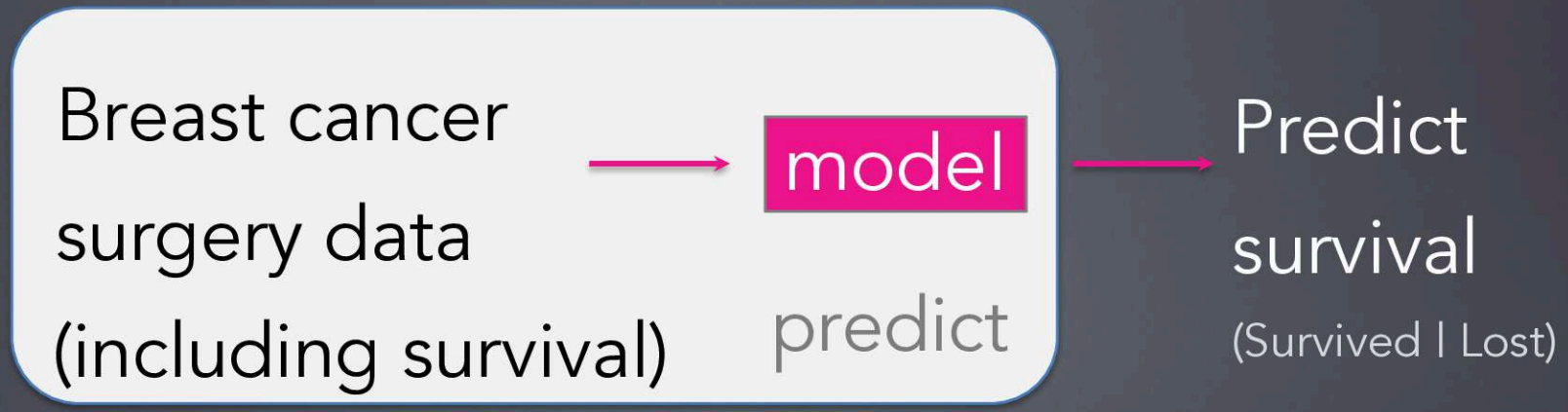
Supervised Learning

Classification: “Answers” from model are categories

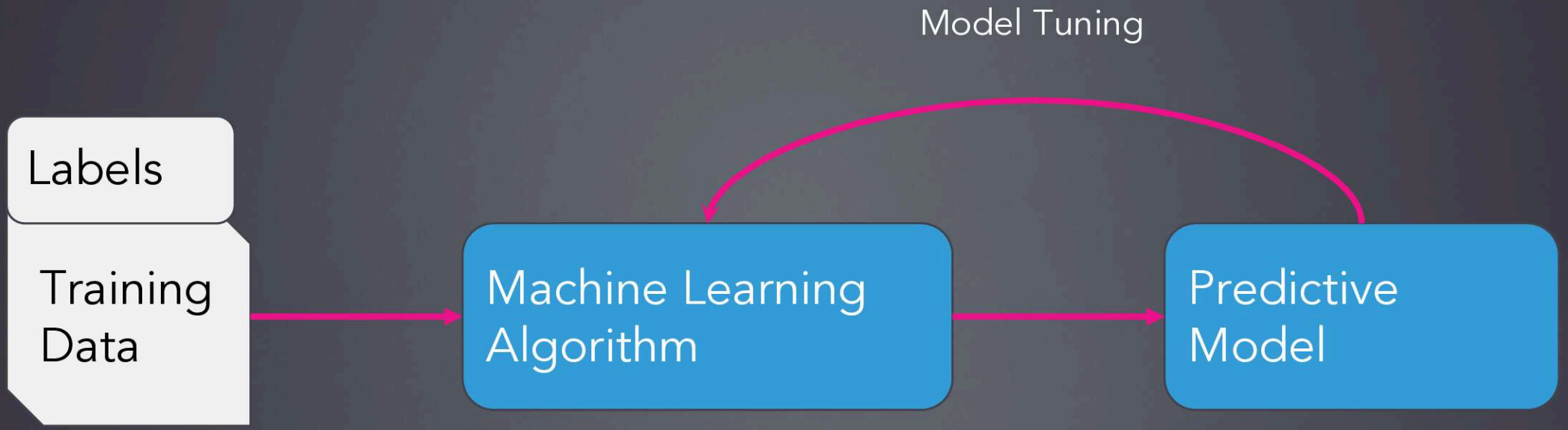


Supervised Learning

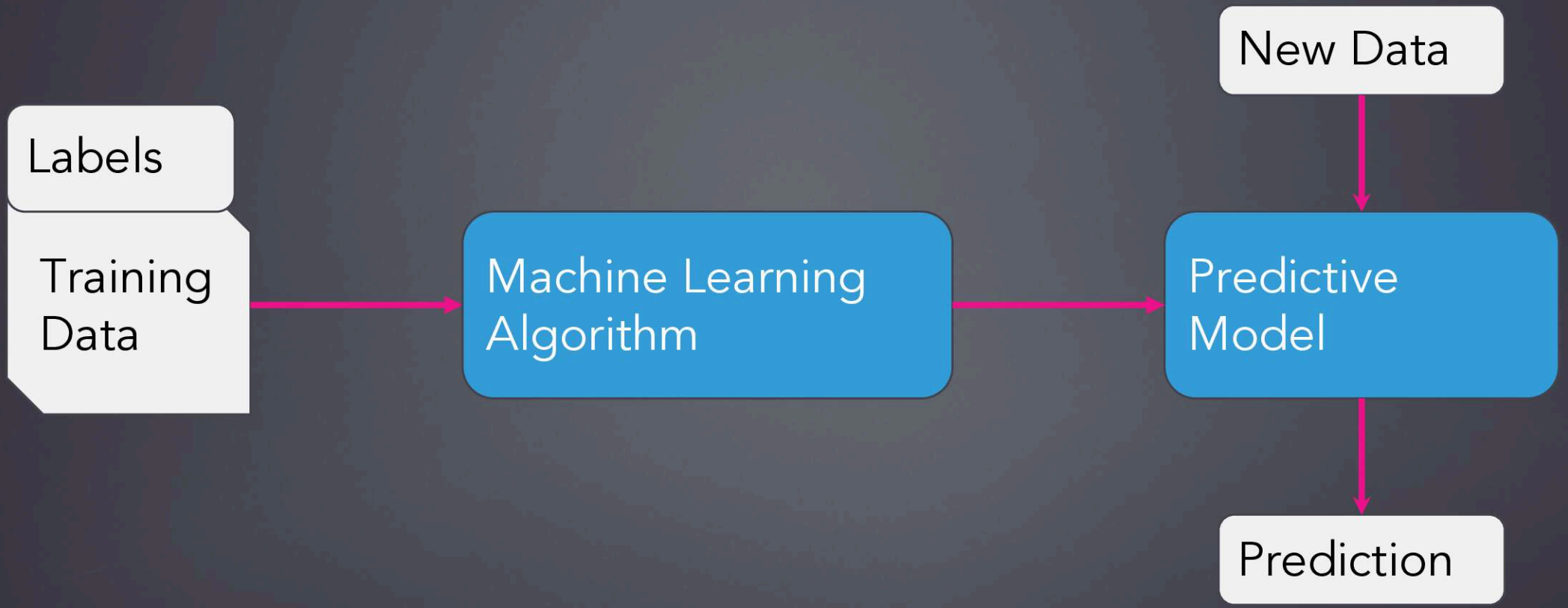
Classification: "Answers" from model are categories



Supervised Learning



Supervised Learning



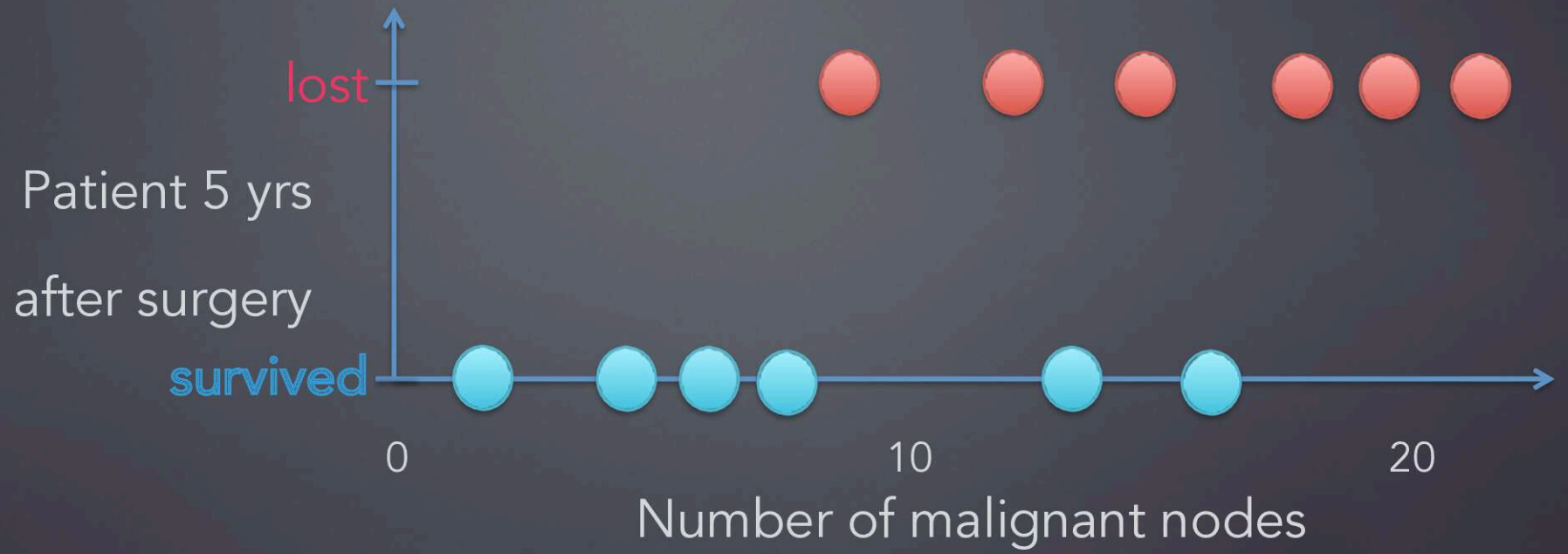
Supervised Learning - Vernacular

- **Observation** - each data point (one row)
- **Target** - Predicted property (column to predict)
- **Label** - Target / Category of the observation (value of target column)
- **Feature** - A property of the observation used for prediction (non-target columns in the data)

Supervised Learning - Vernacular

1 Feature. Number of malignant nodes

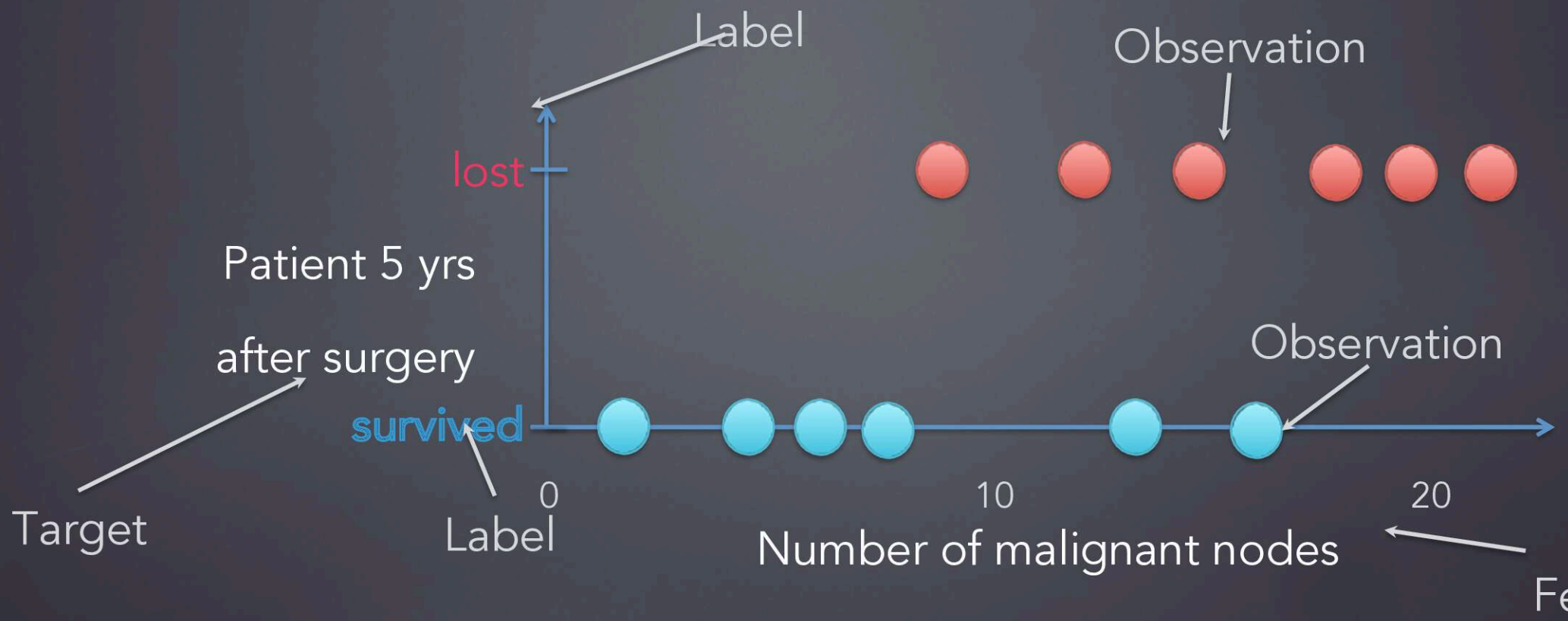
2 Labels. Survived / Lost



Supervised Learning - Vernacular

1 Feature. Number of malignant nodes

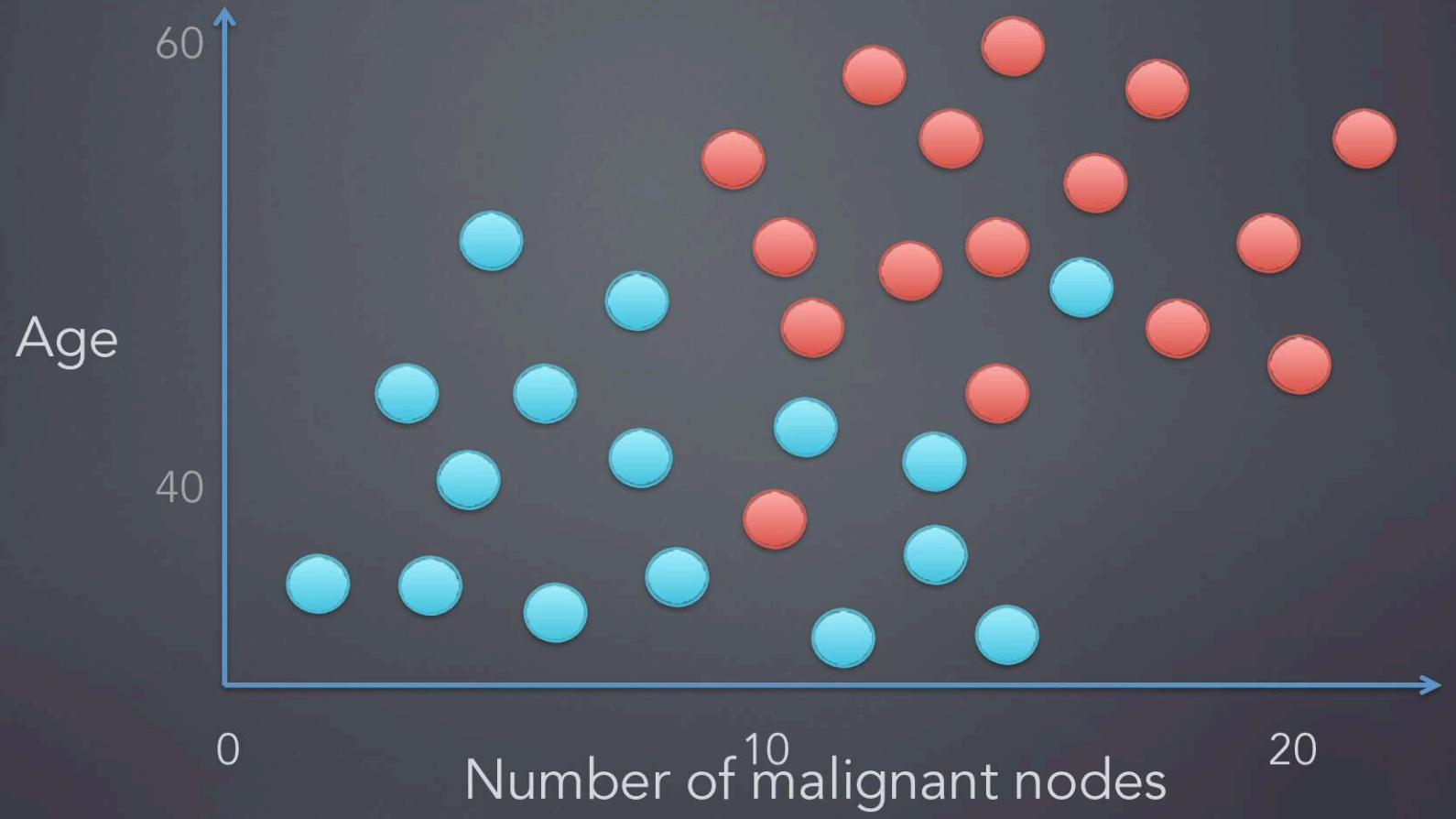
2 Labels. Survived / Lost



Supervised Learning - Example

2 Features. No of malignant nodes / Age

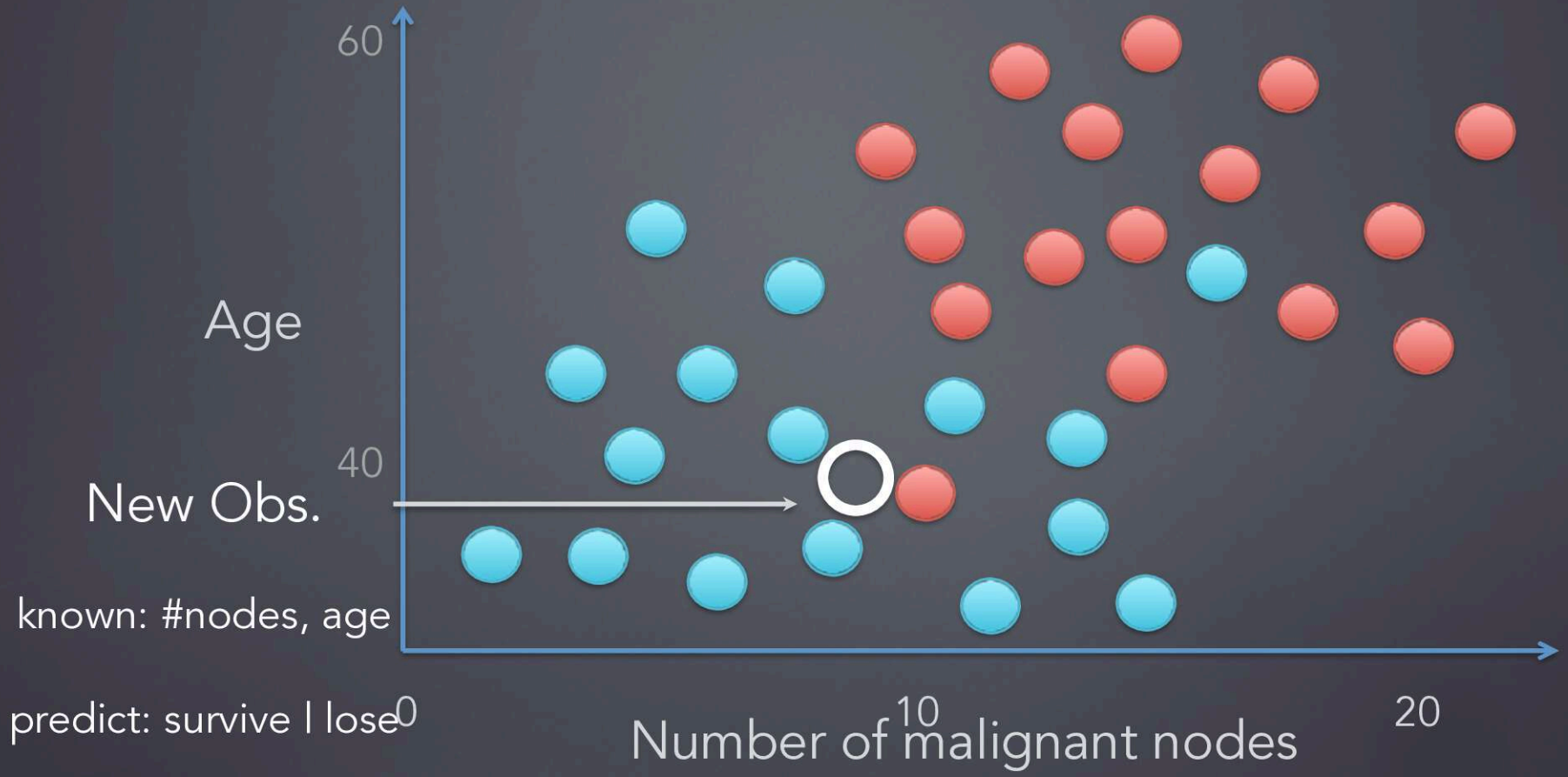
2 Labels. Survived / Lost



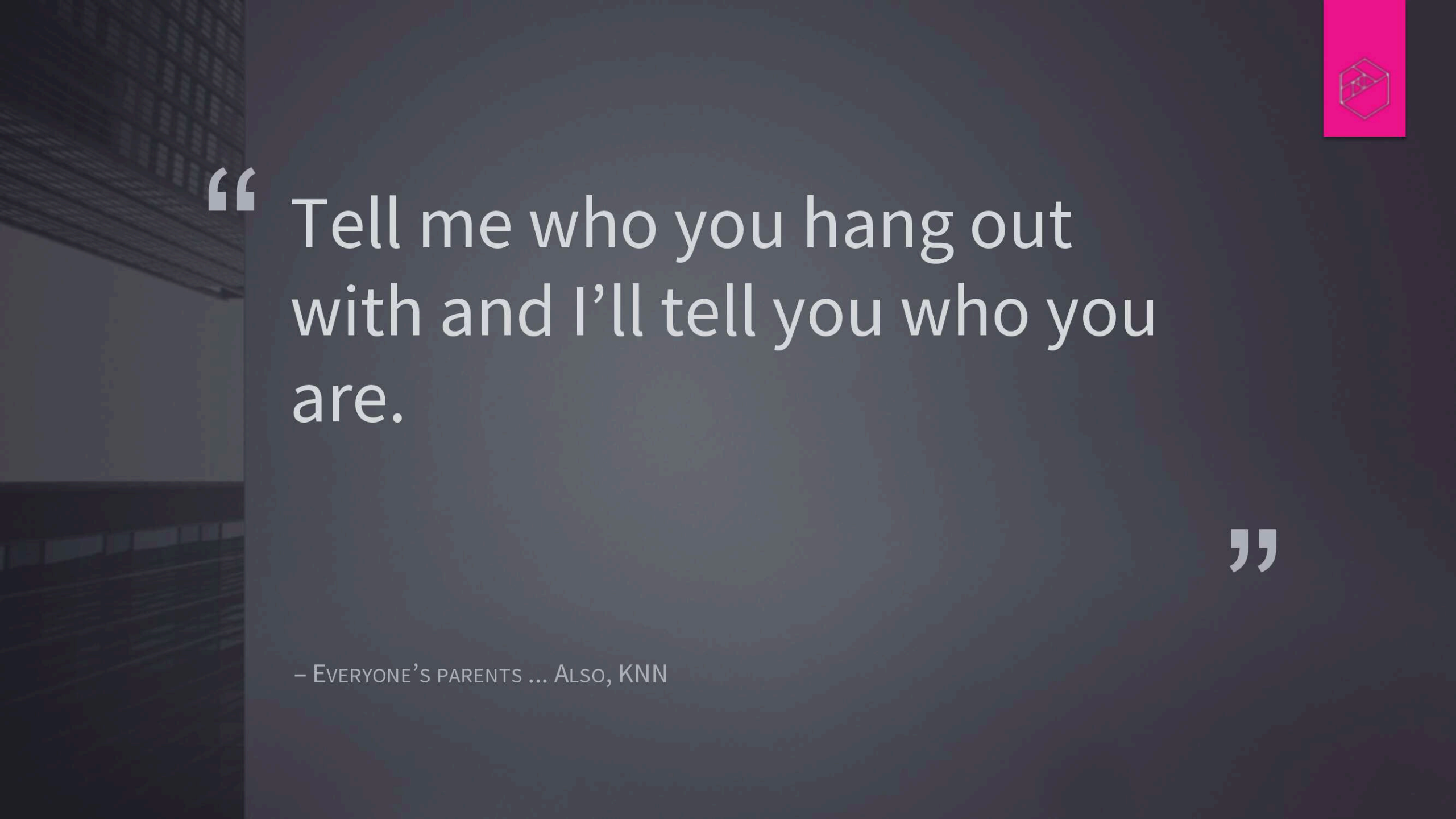
Supervised Learning - Example

2 Features. No of malignant nodes / Age

2 Labels. Survived / Lost



K-Nearest Neighbors Algorithm


“
Tell me who you hang out with and I’ll tell you who you are.
”
– Everyone’s parents … Also, KNN

KNN - Overview

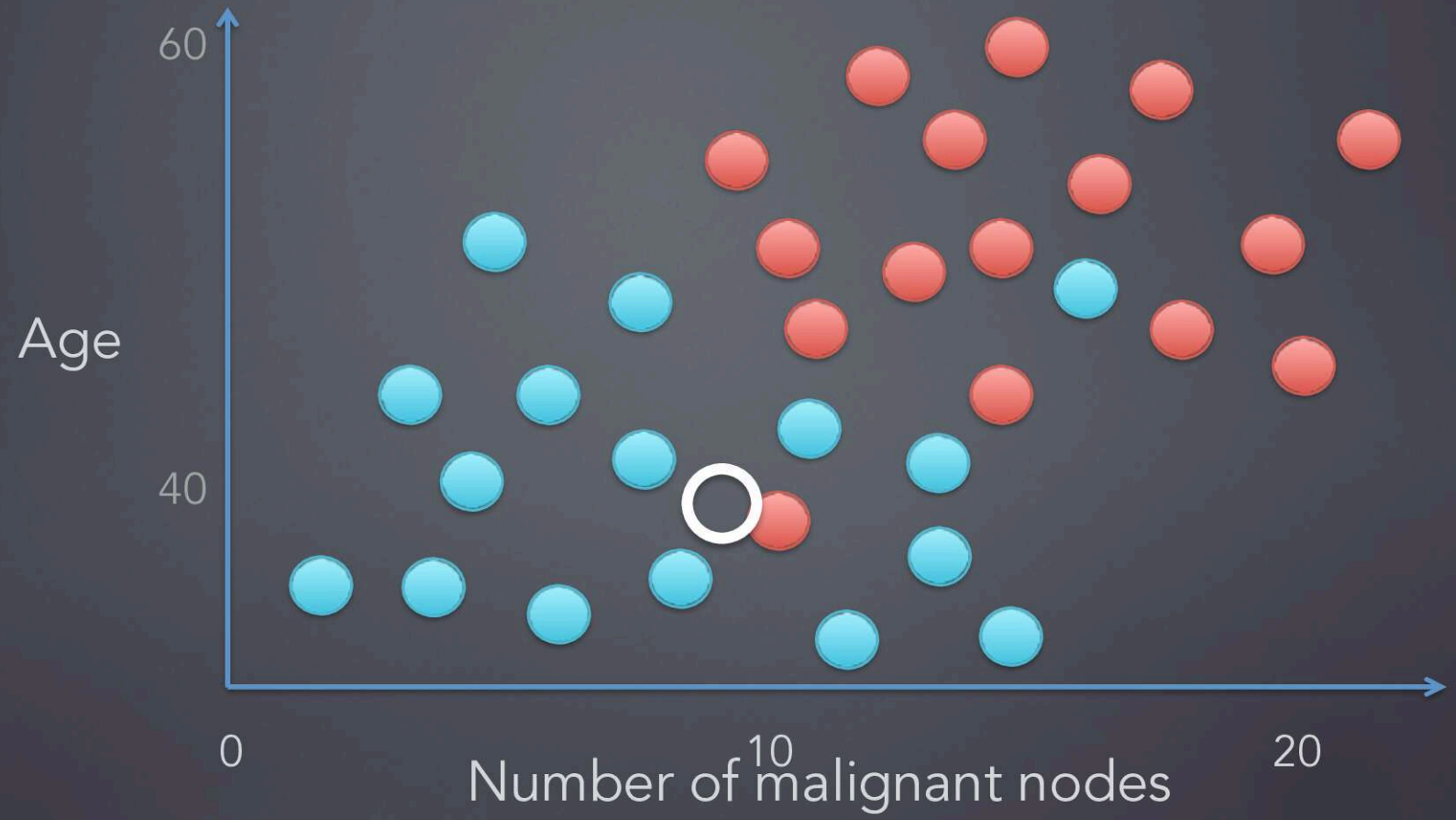
The KNN Algorithm is fairly straightforward and can be summarized by the following steps:

- Choose the number *k* and the distance metric (Euclidean is the most common)
- Find the K-nearest neighbors of the observation we want to classify
- Assign the class label by majority vote

Supervised Learning - Example

Look at the nearest neighbor, predict their label

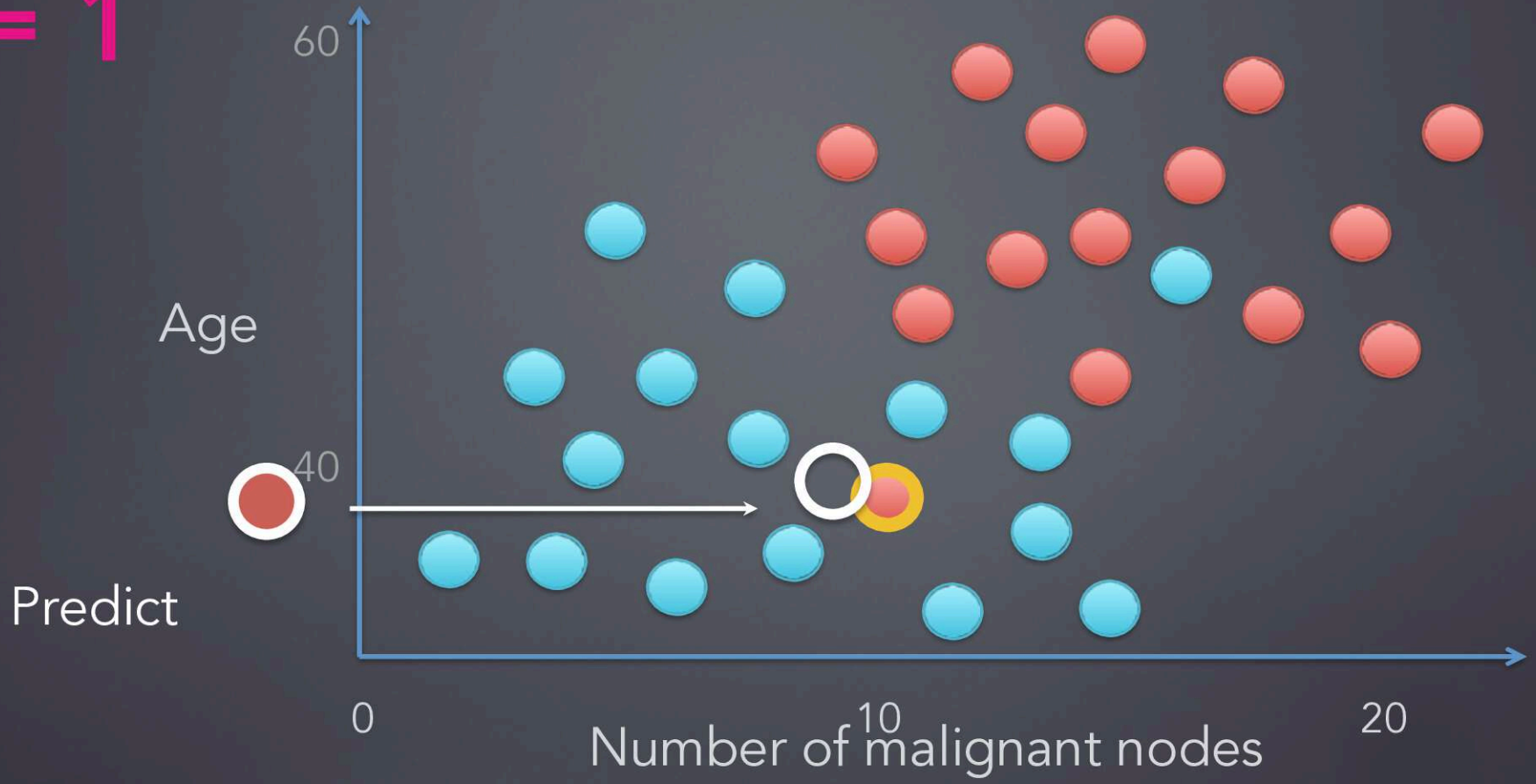
K = 1



Supervised Learning - Example

Look at the nearest neighbor, predict their label

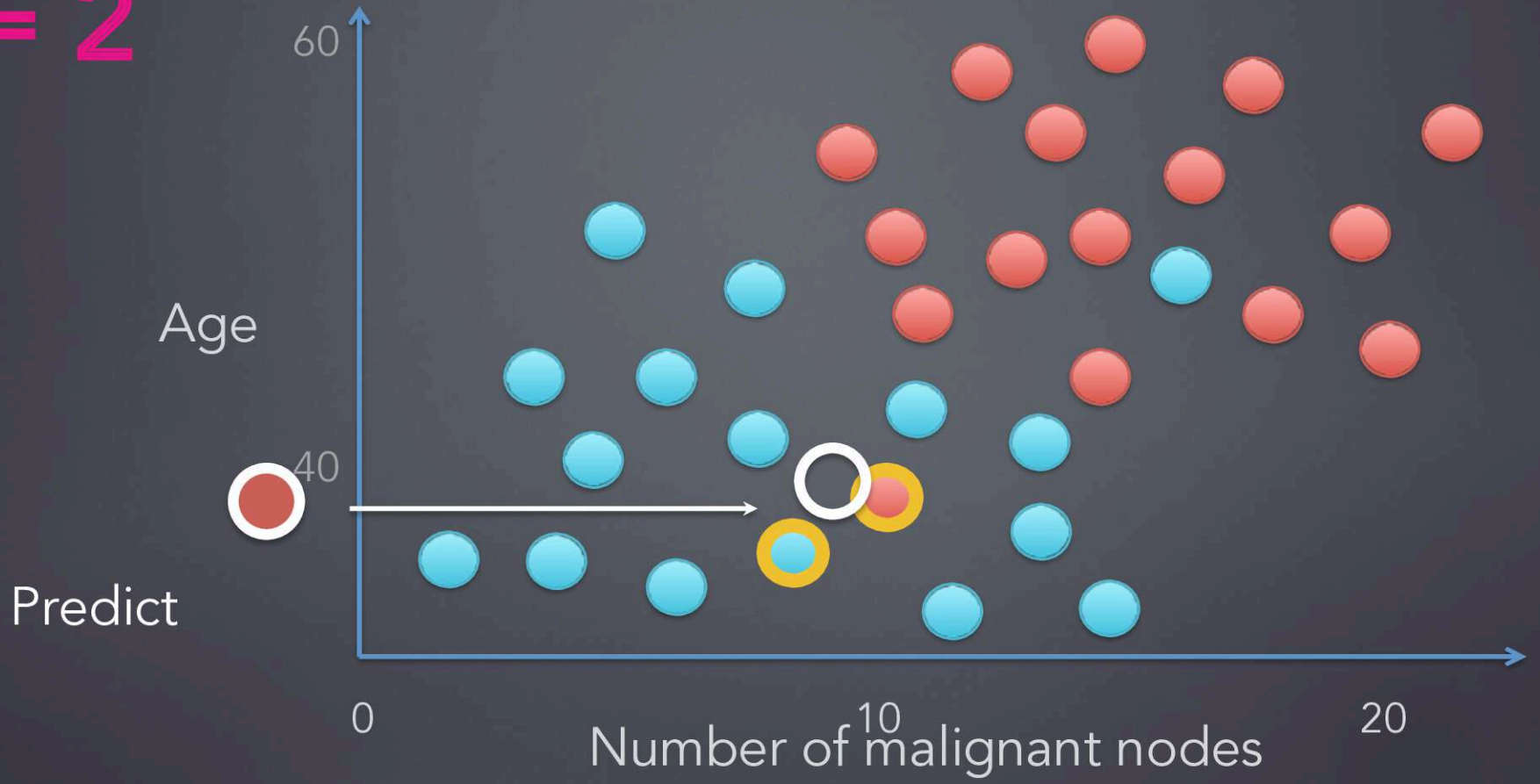
K = 1



Supervised Learning - Example

Look at the two nearest neighbors, predict the label you see most

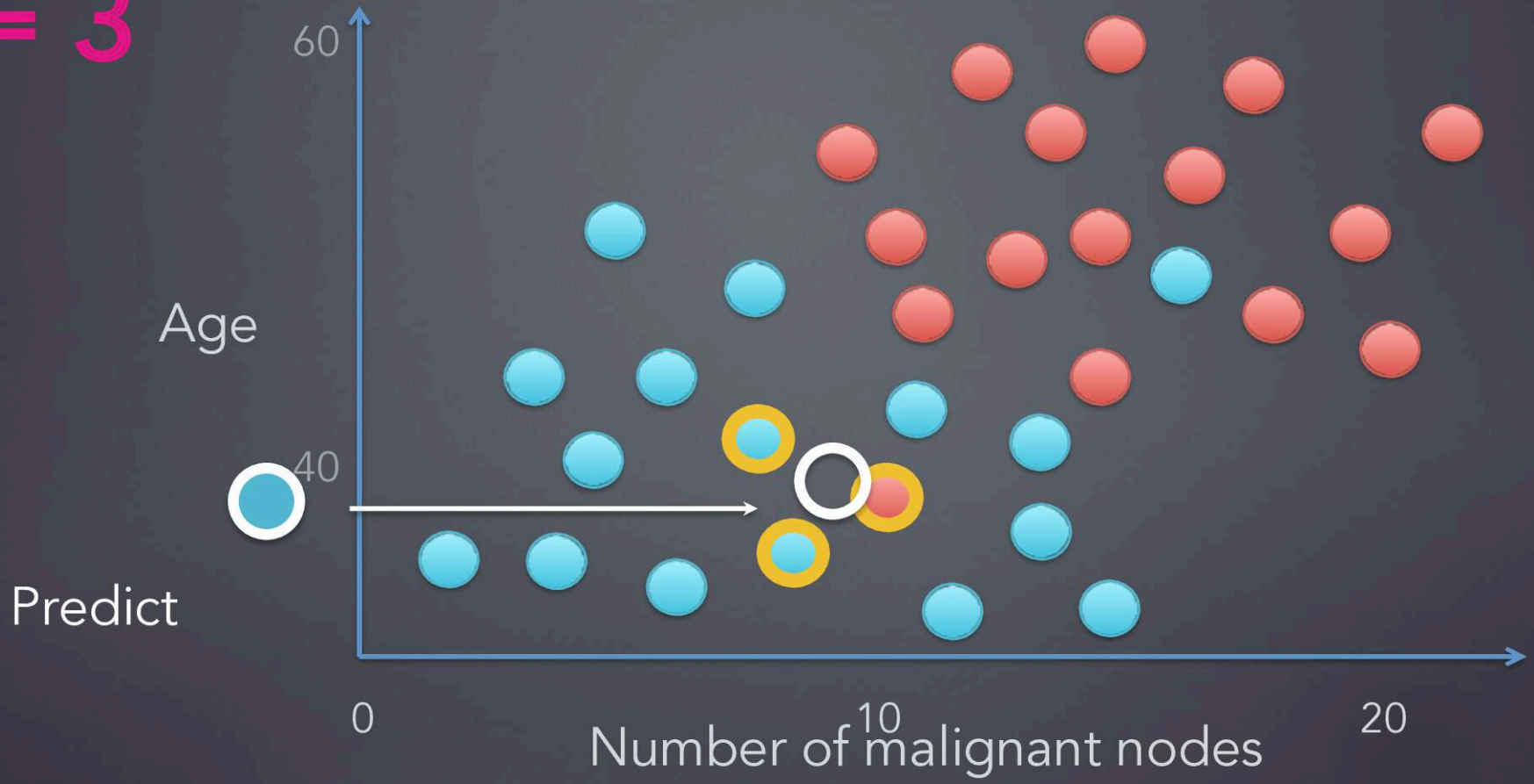
K = 2



Supervised Learning - Example

Look at the three nearest neighbors, predict the label you see most

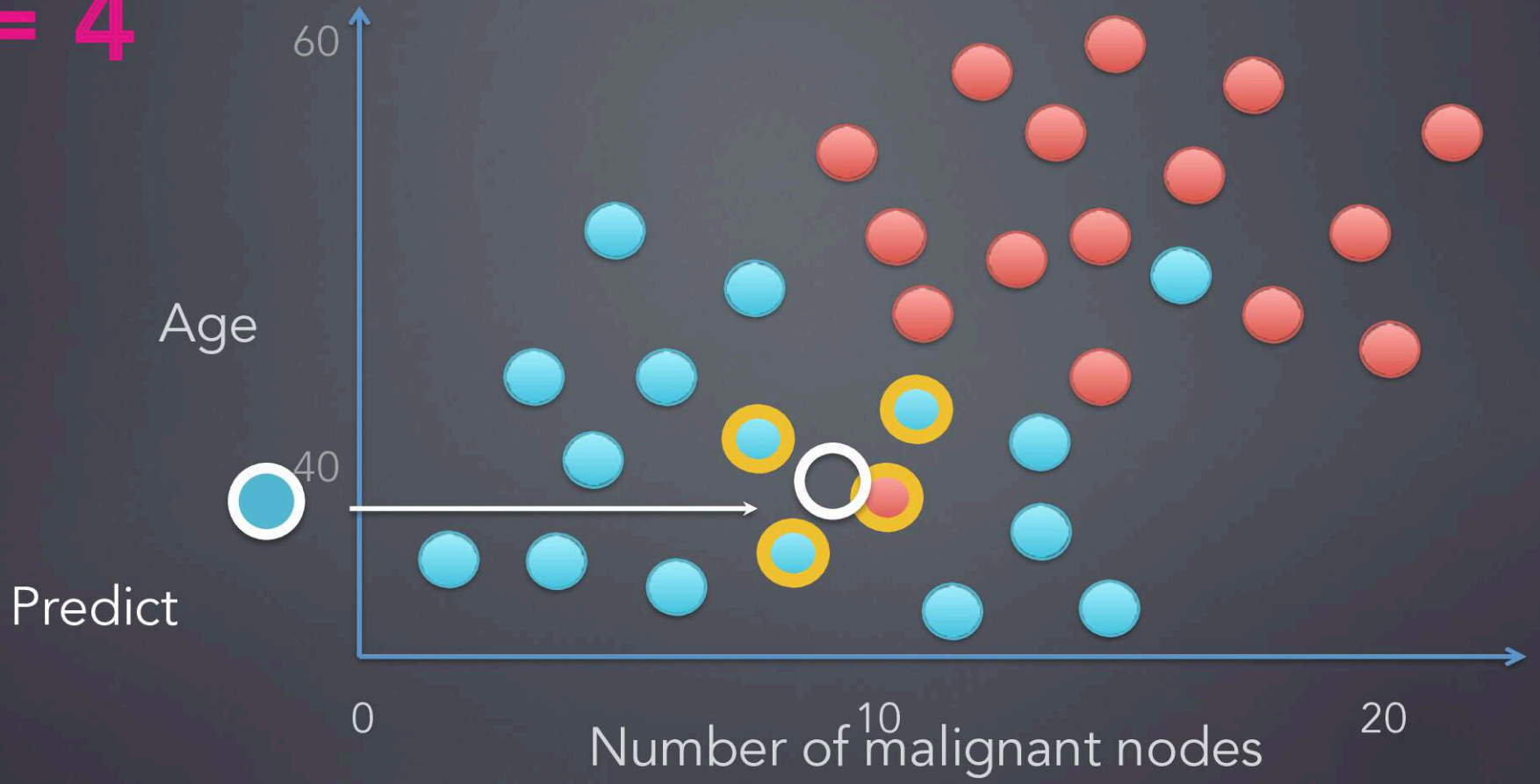
K = 3



Supervised Learning - Example

Look at the four nearest neighbors, predict the label you see most

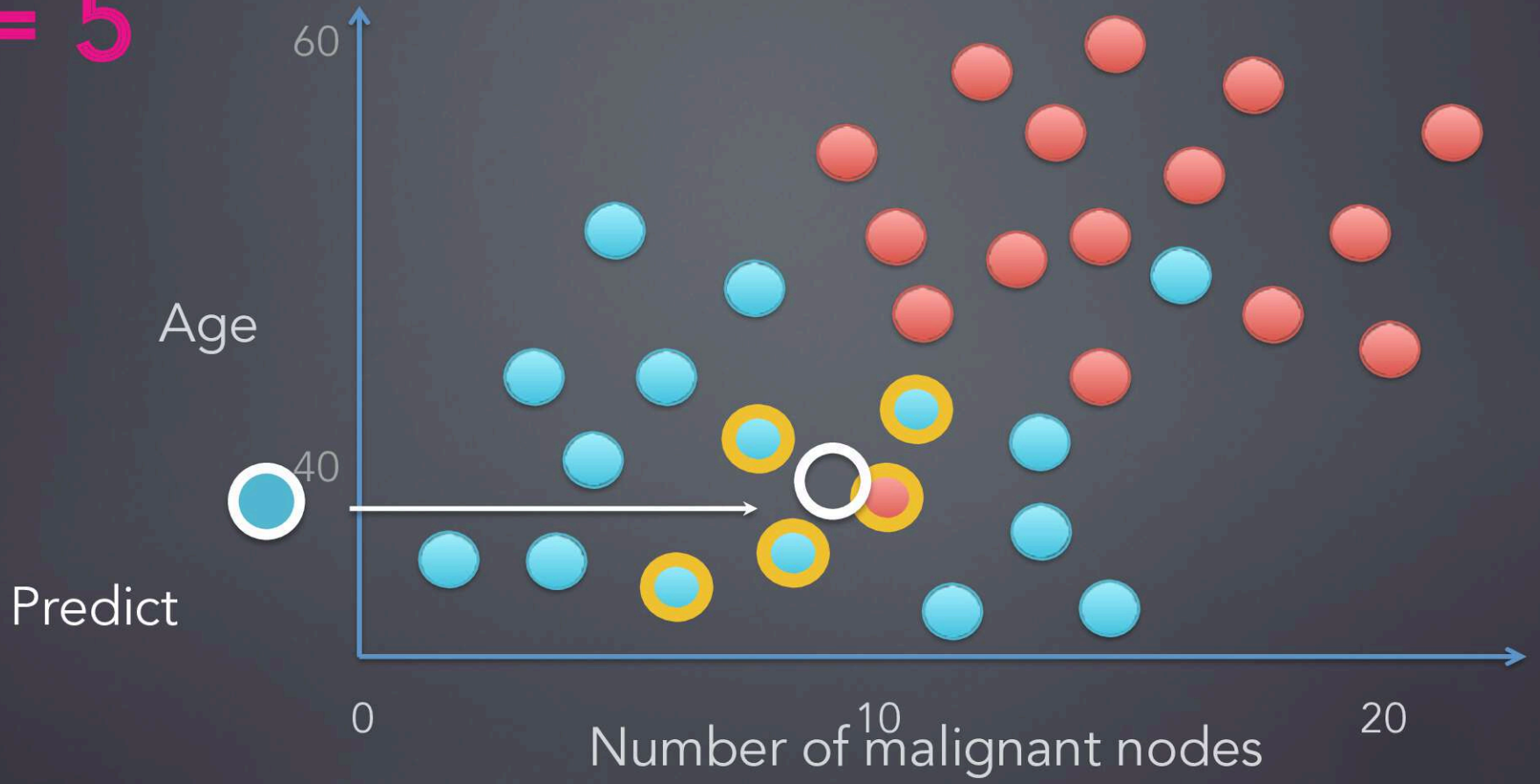
K = 4



Supervised Learning - Example

Look at the five nearest neighbors, predict the label you see most

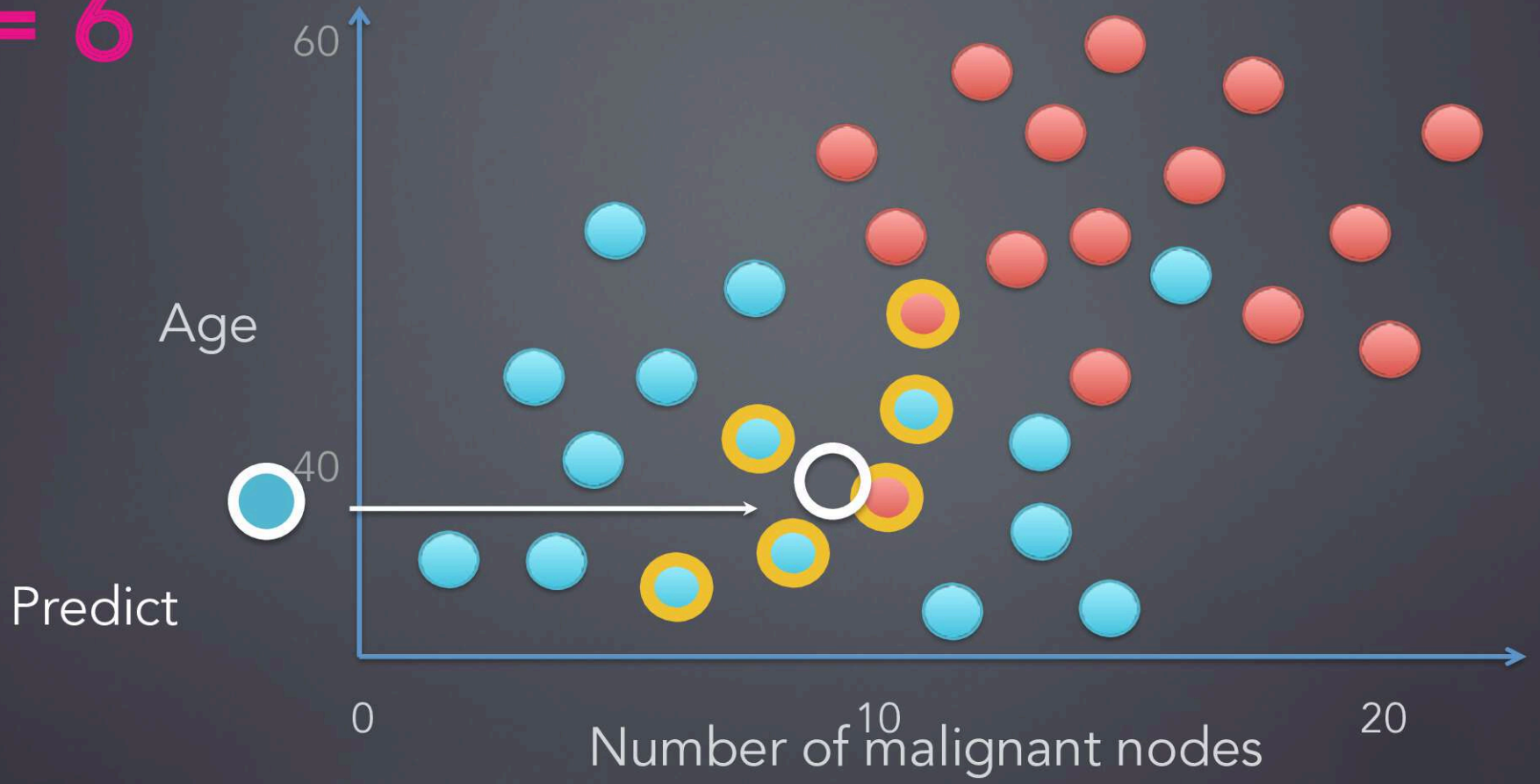
K = 5



Supervised Learning - Example

Look at the five nearest neighbors, predict the label you see most

K = 6



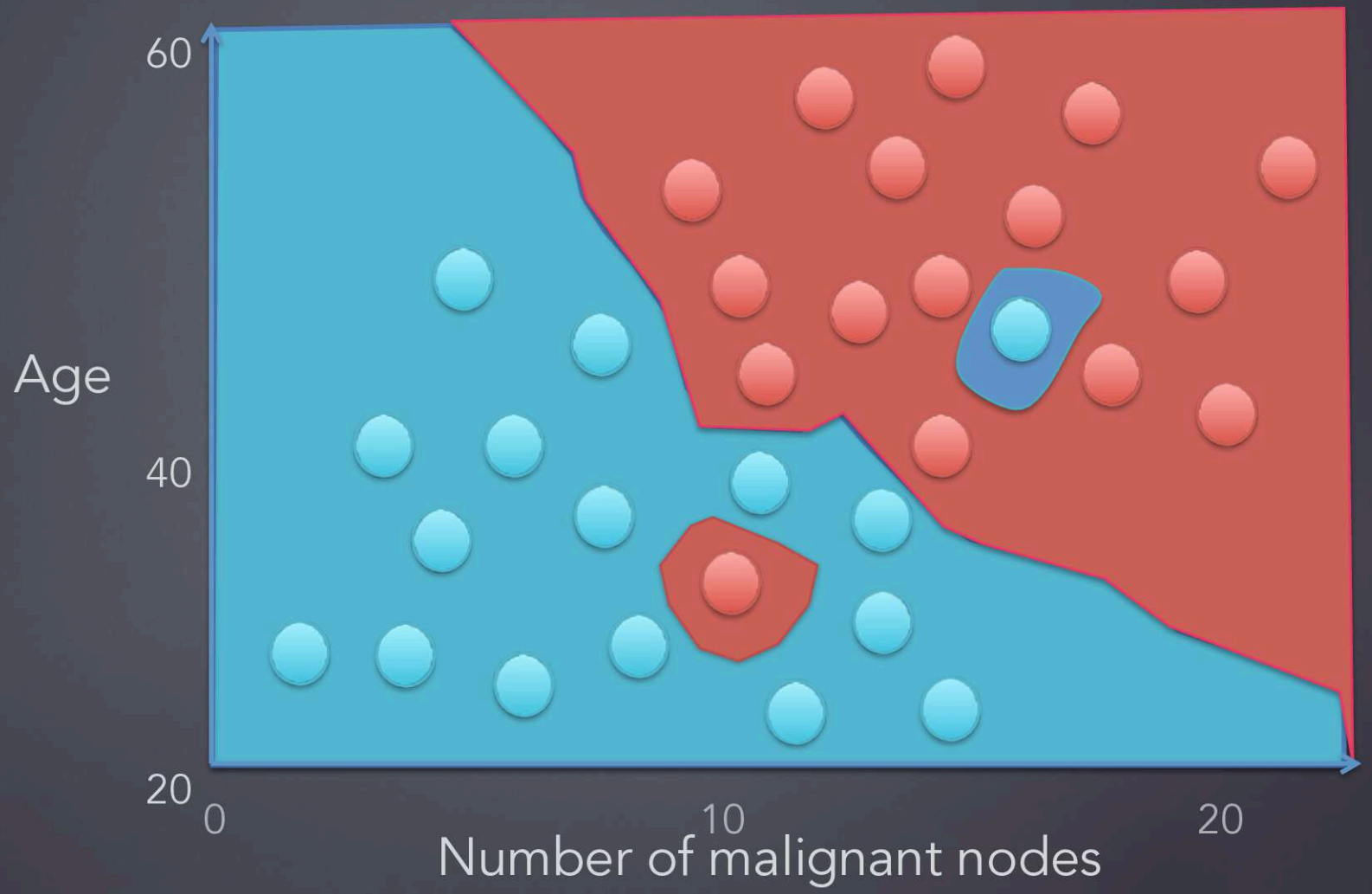
Decision Regions

Decision Regions - What are they?

- In general, a machine learning classifier partitions the feature space into volumes called **decision regions**.
- All observations **inside** a decision region are assigned to the same category
- The decision regions are separated by surfaces called **decision boundaries**. These boundaries represent points where there are ties between two or more categories

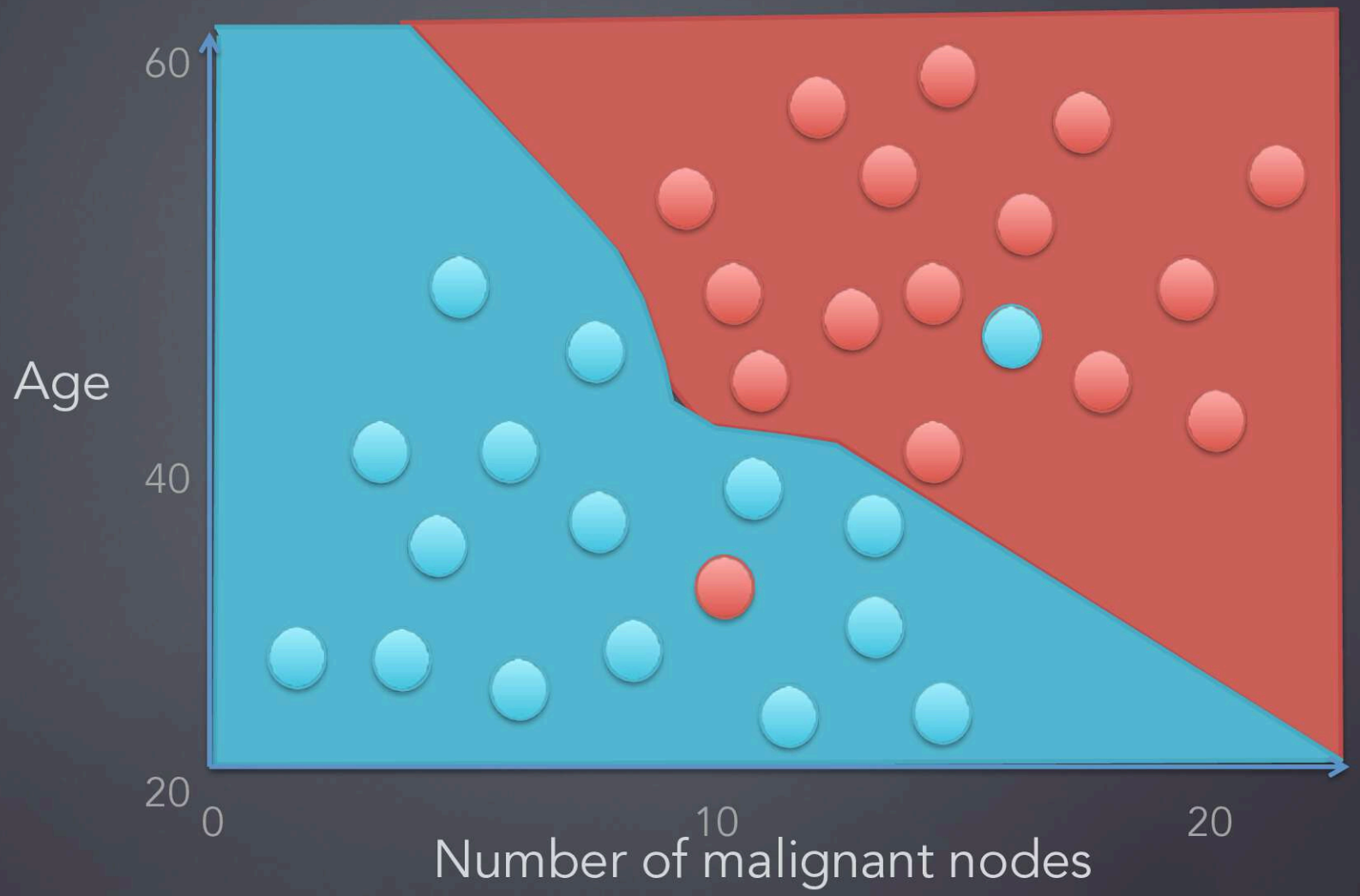
KNN Decision Boundary

K = 1



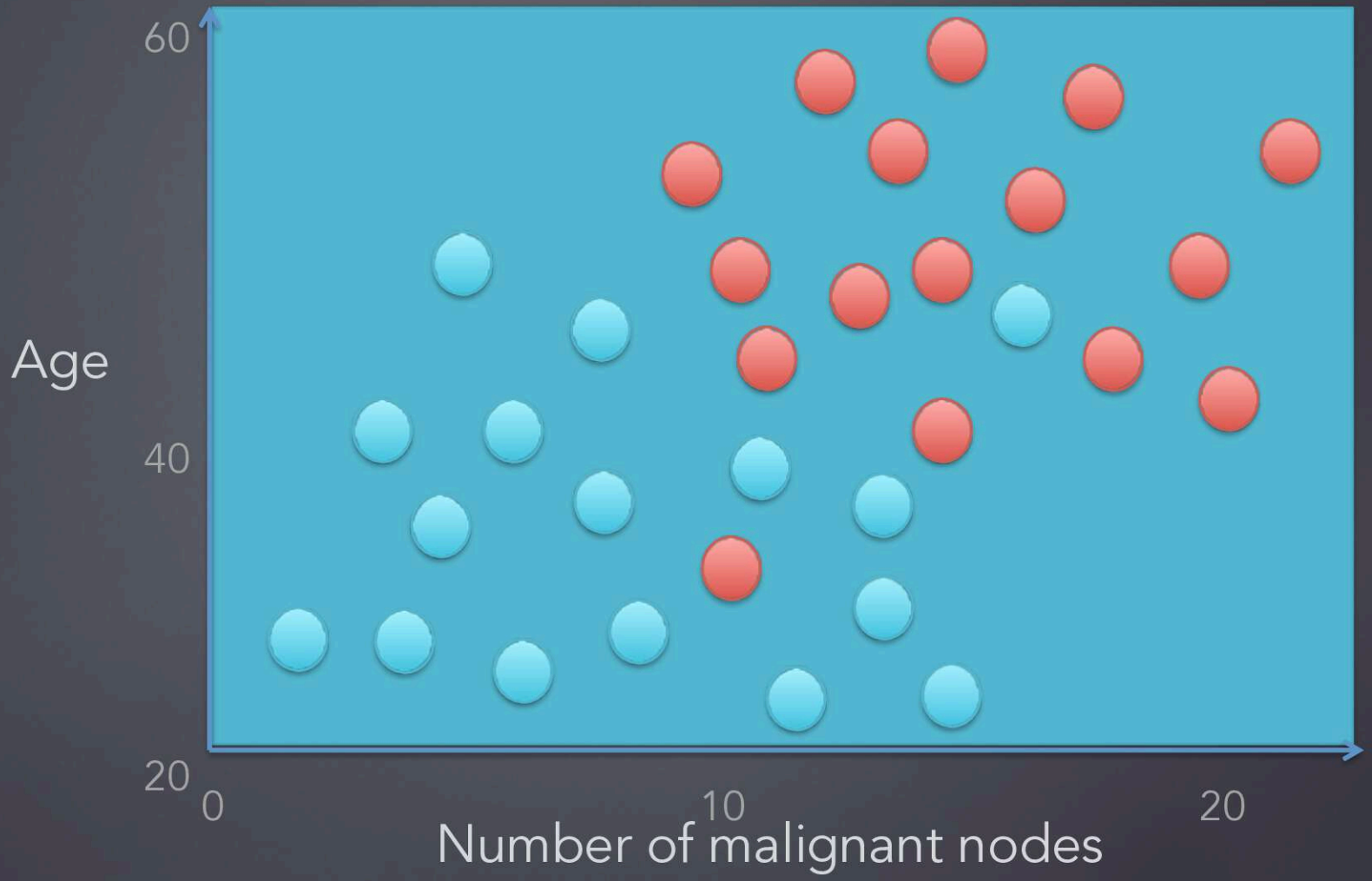
KNN Decision Boundary

K = 5

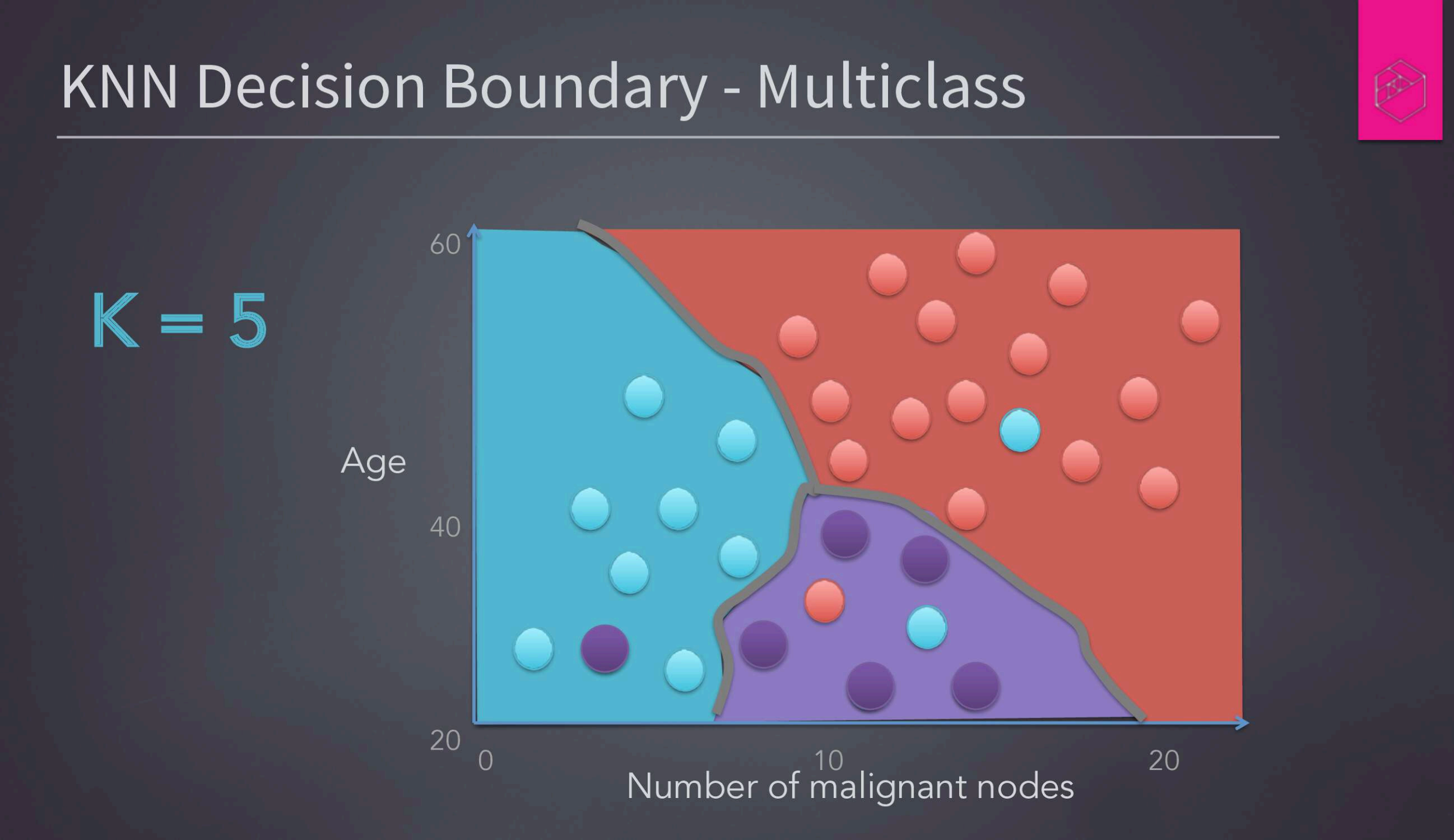


KNN Decision Boundary

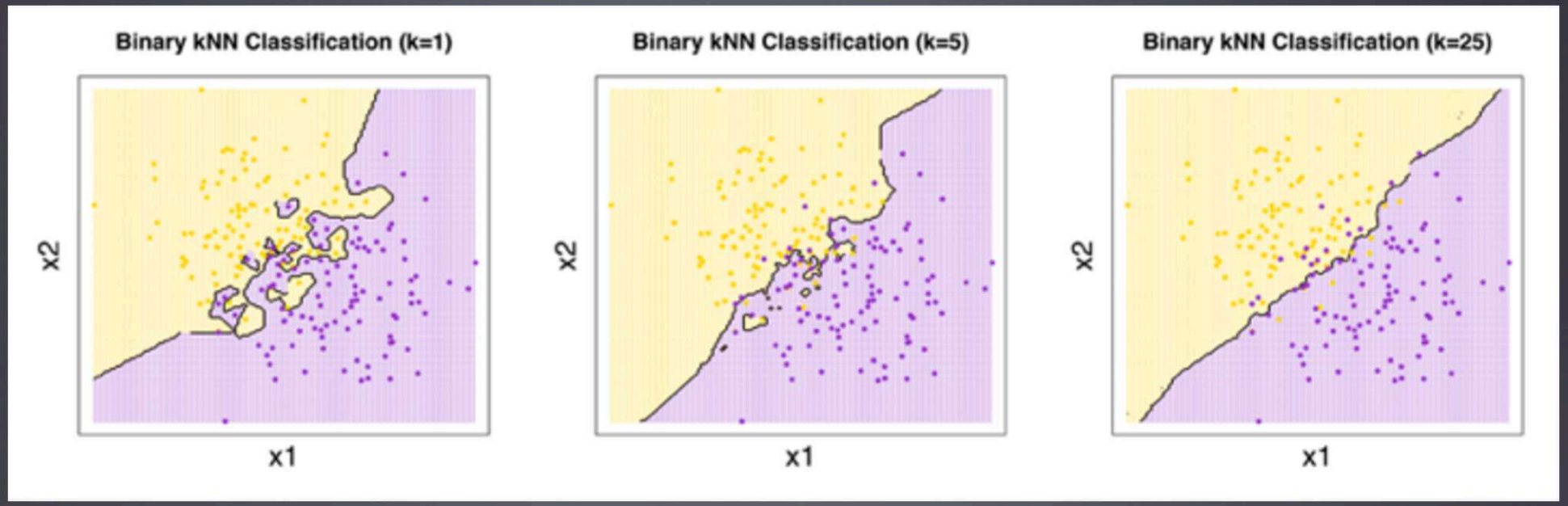
K = 34



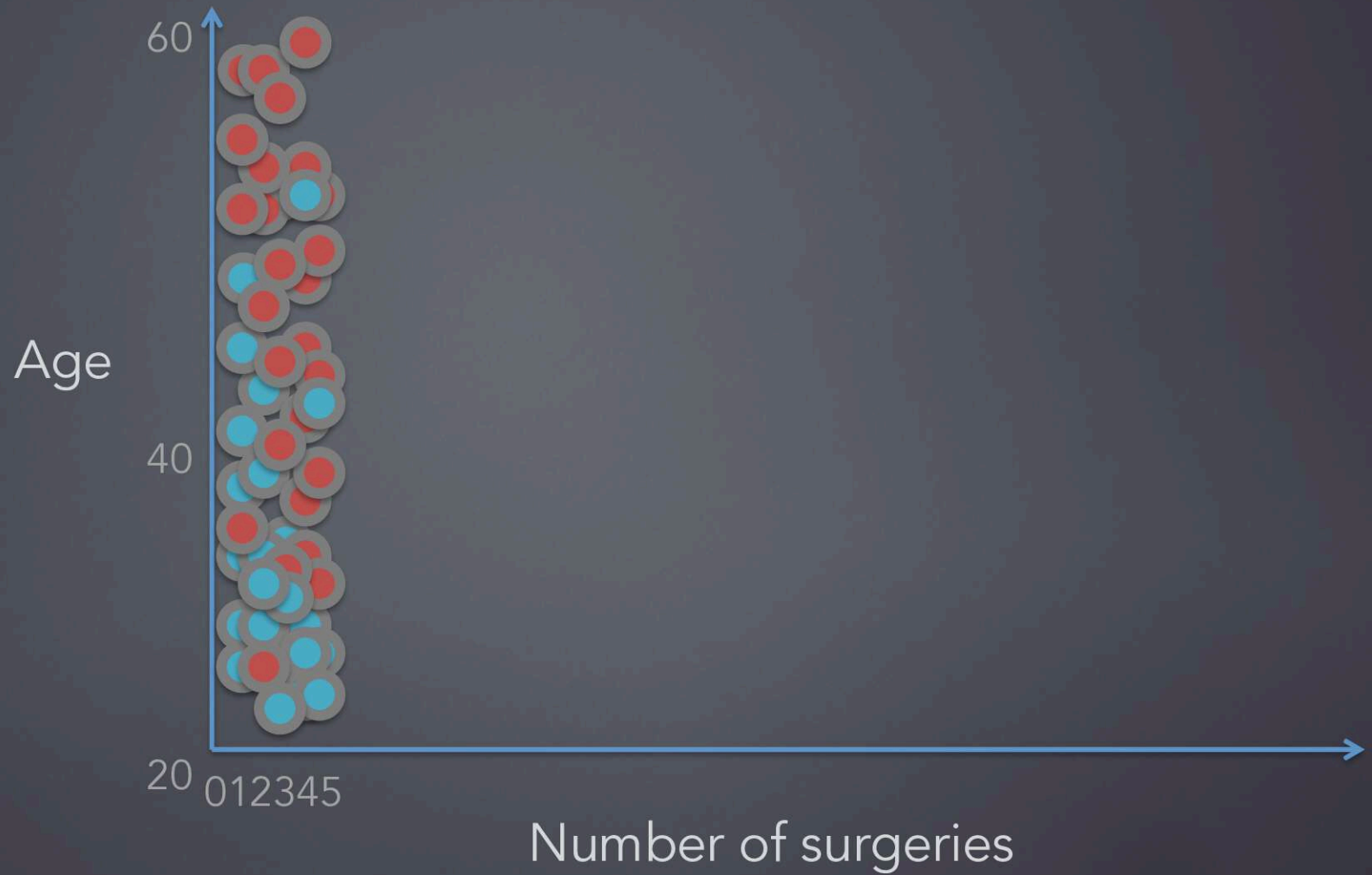
KNN Decision Boundary - Multiclass



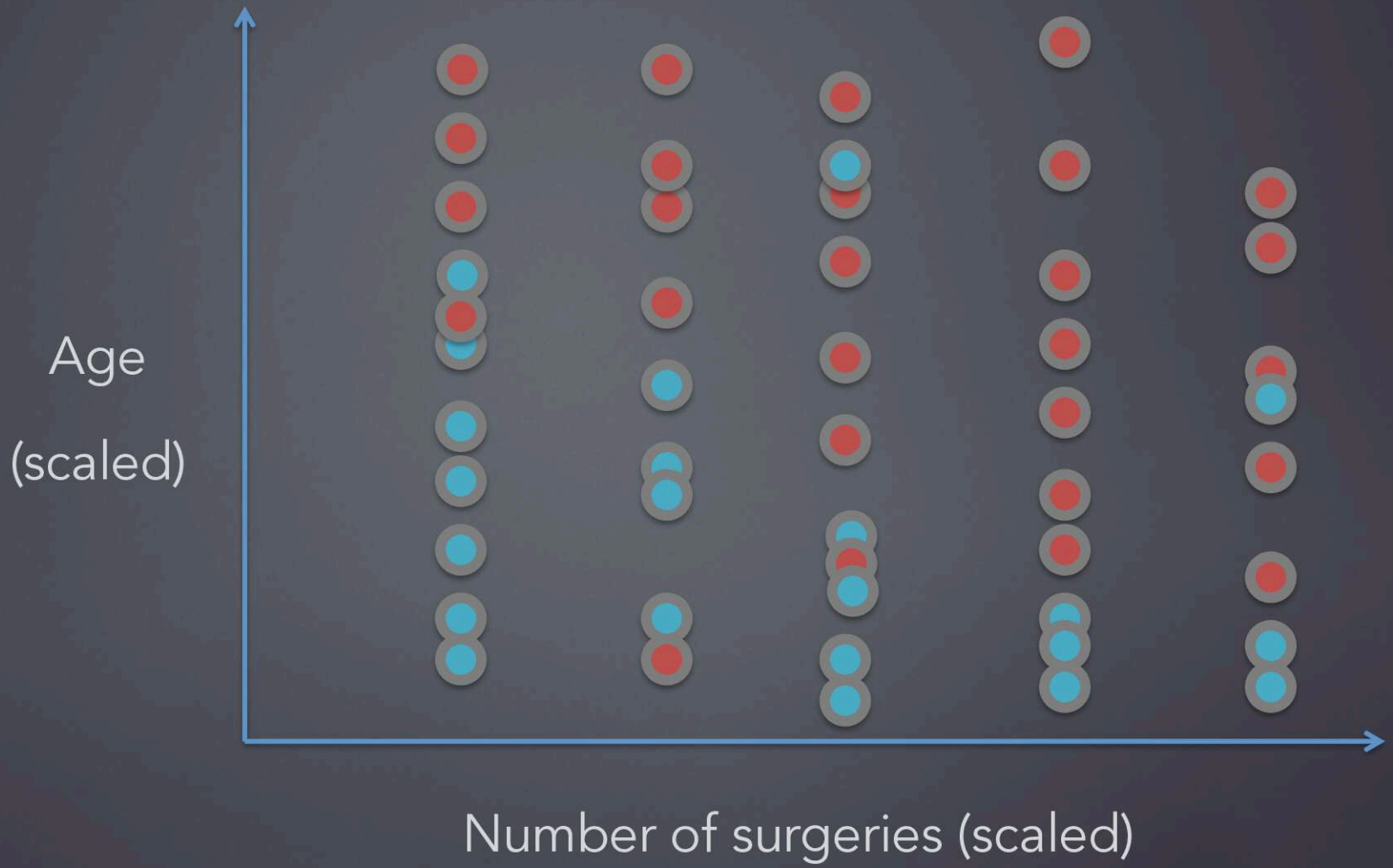
KNN Decision Boundary



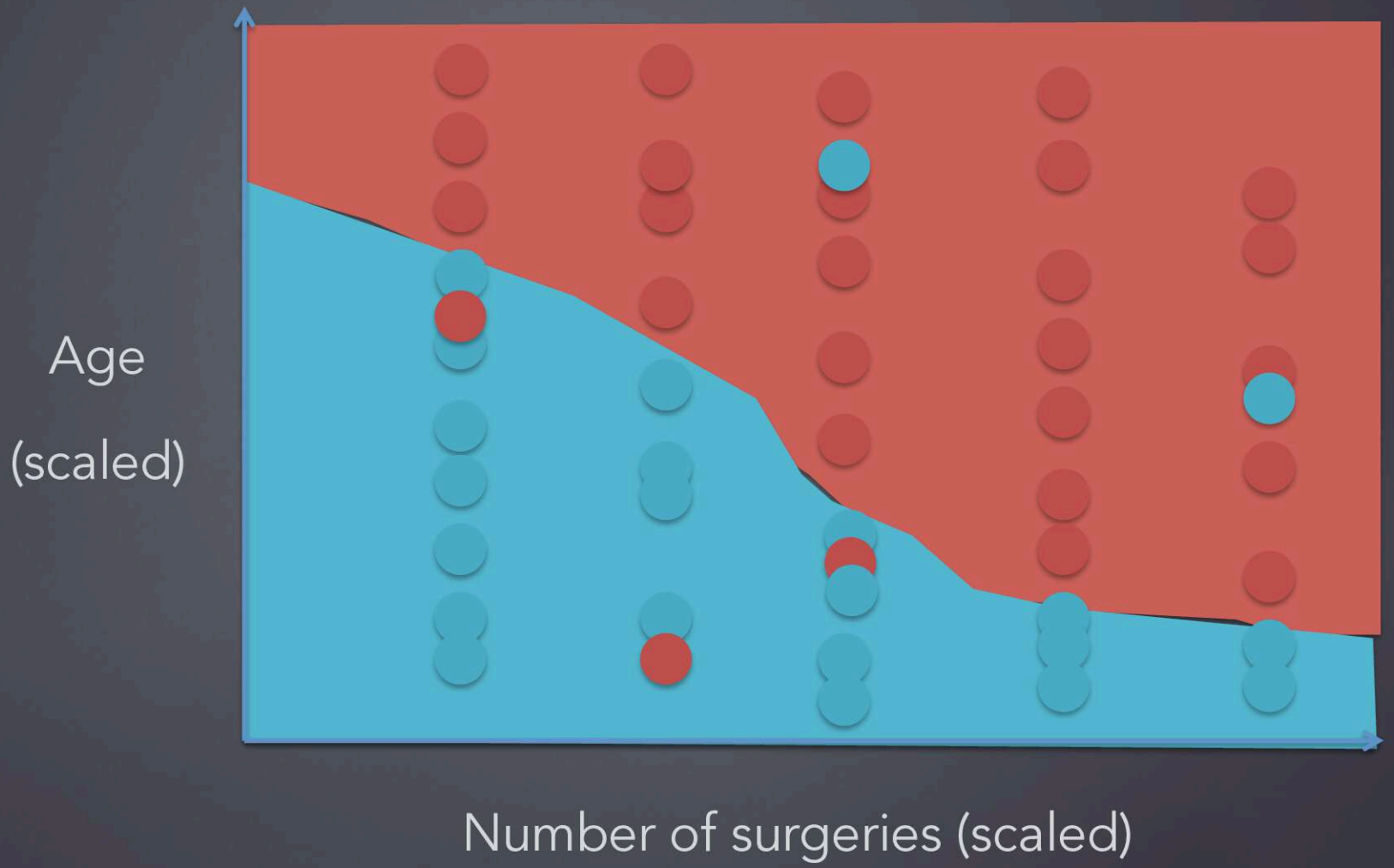
Remember to Scale!



Remember to Scale!



Remember to Scale!



KNN - Overview

Pros

- Classifier adapts as new training data is collected
- Easy to implement
- Interpretable

Cons

- “Lazy” - KNN doesn’t have any “training time”, but instead memorizes the training data.
- Memory intensive
- Prediction time scales linearly as more training data is introduced

Thank You!