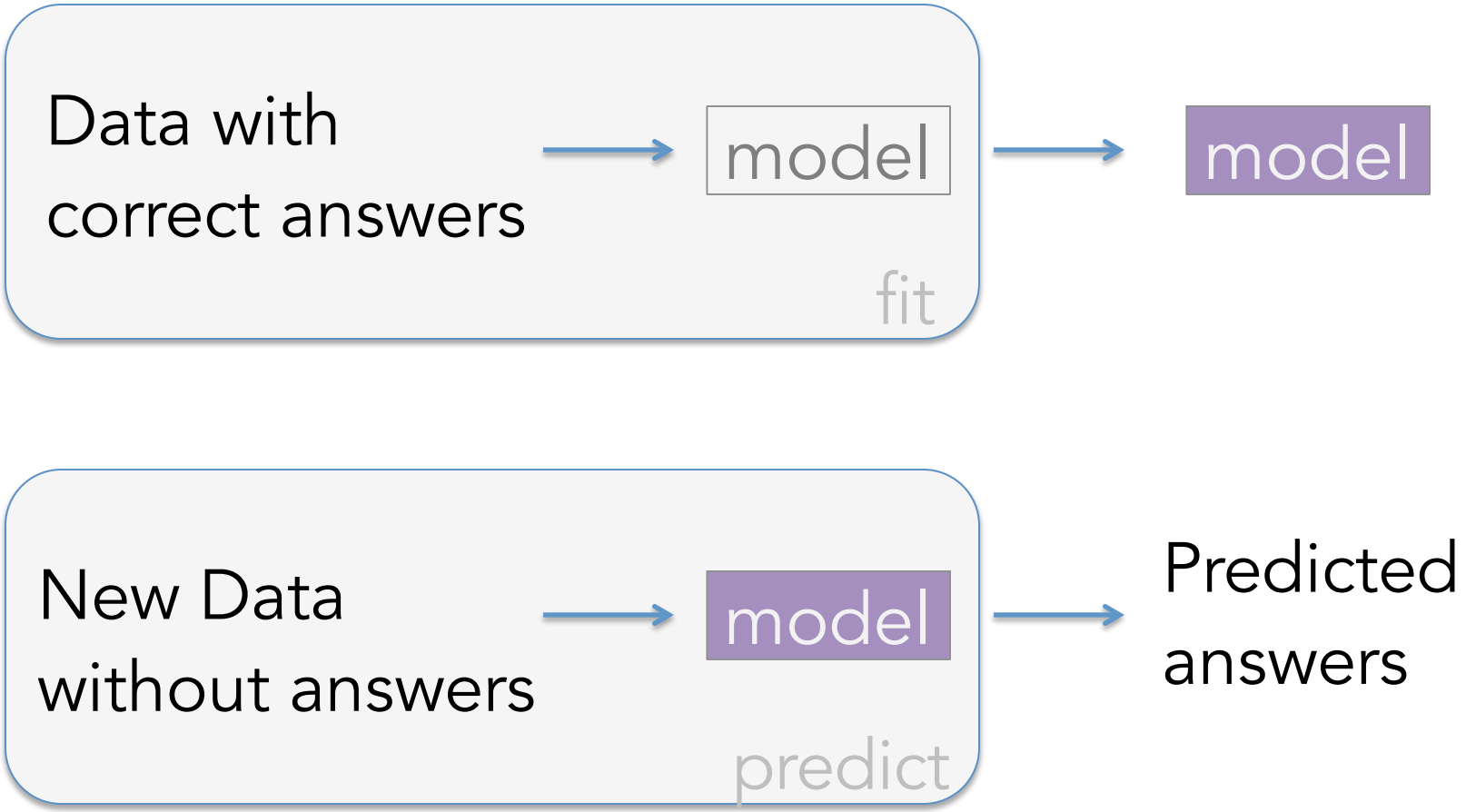
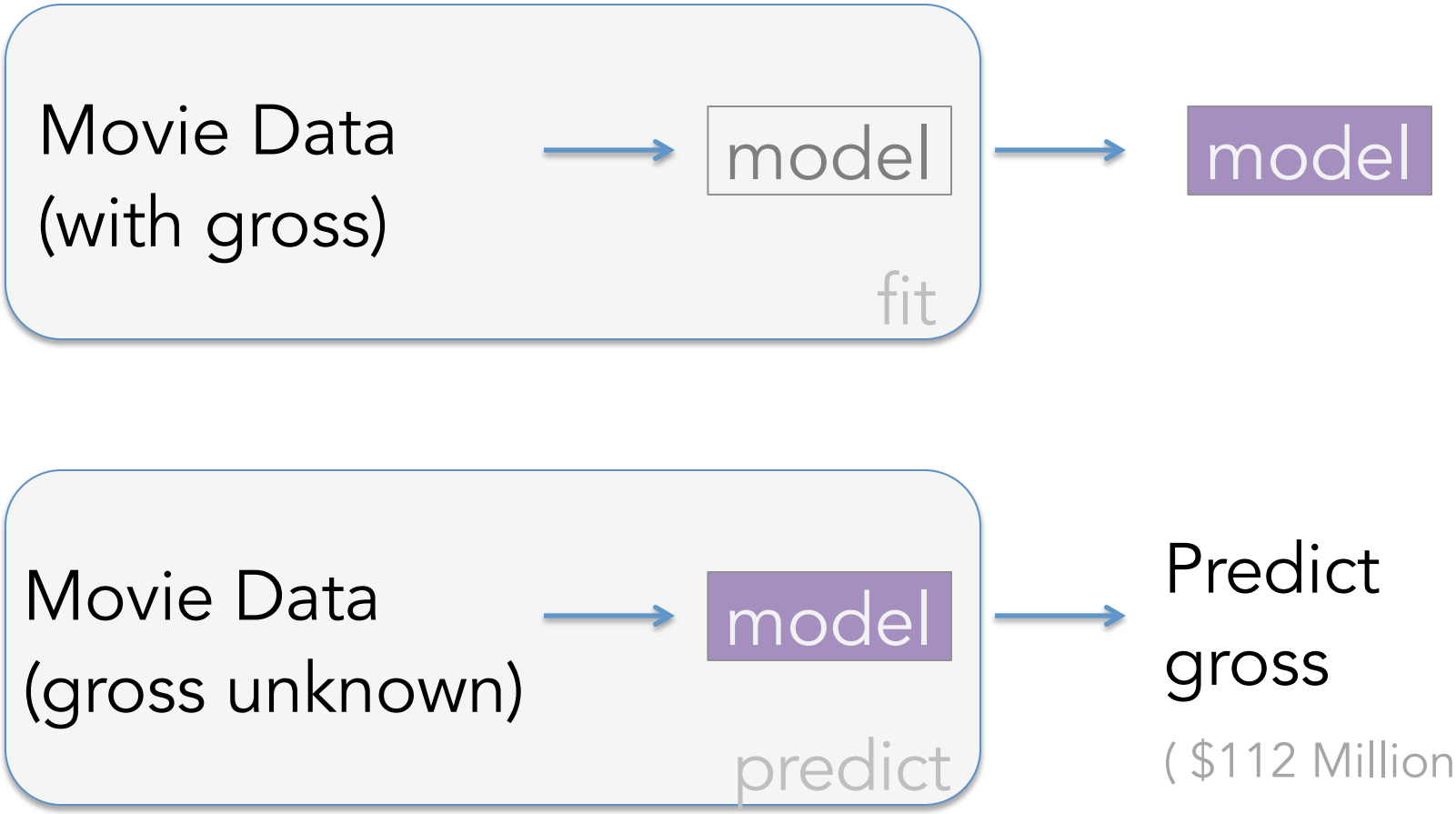
Supervised Learning: Regression and Classification

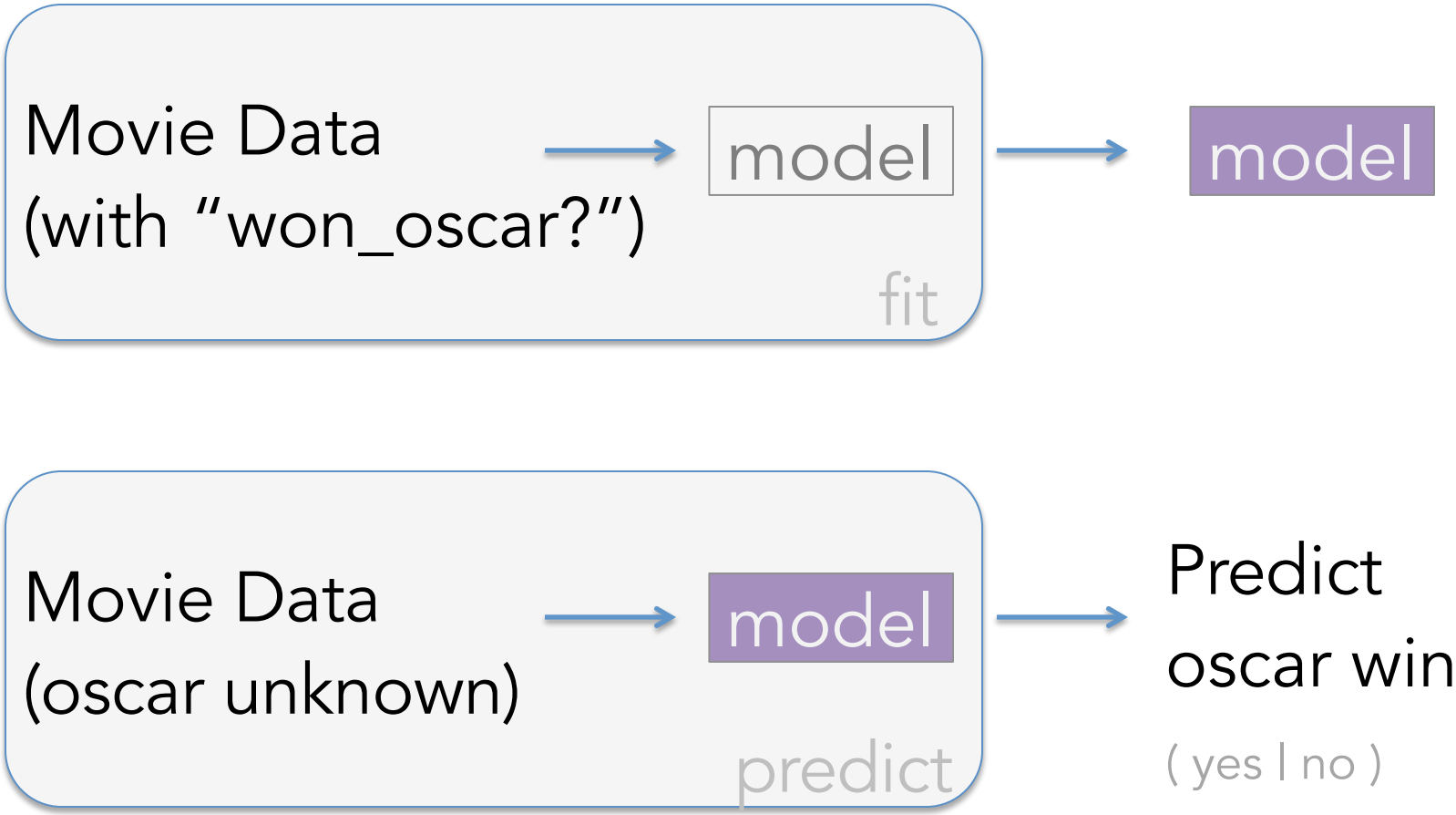
What is Supervised Learning?



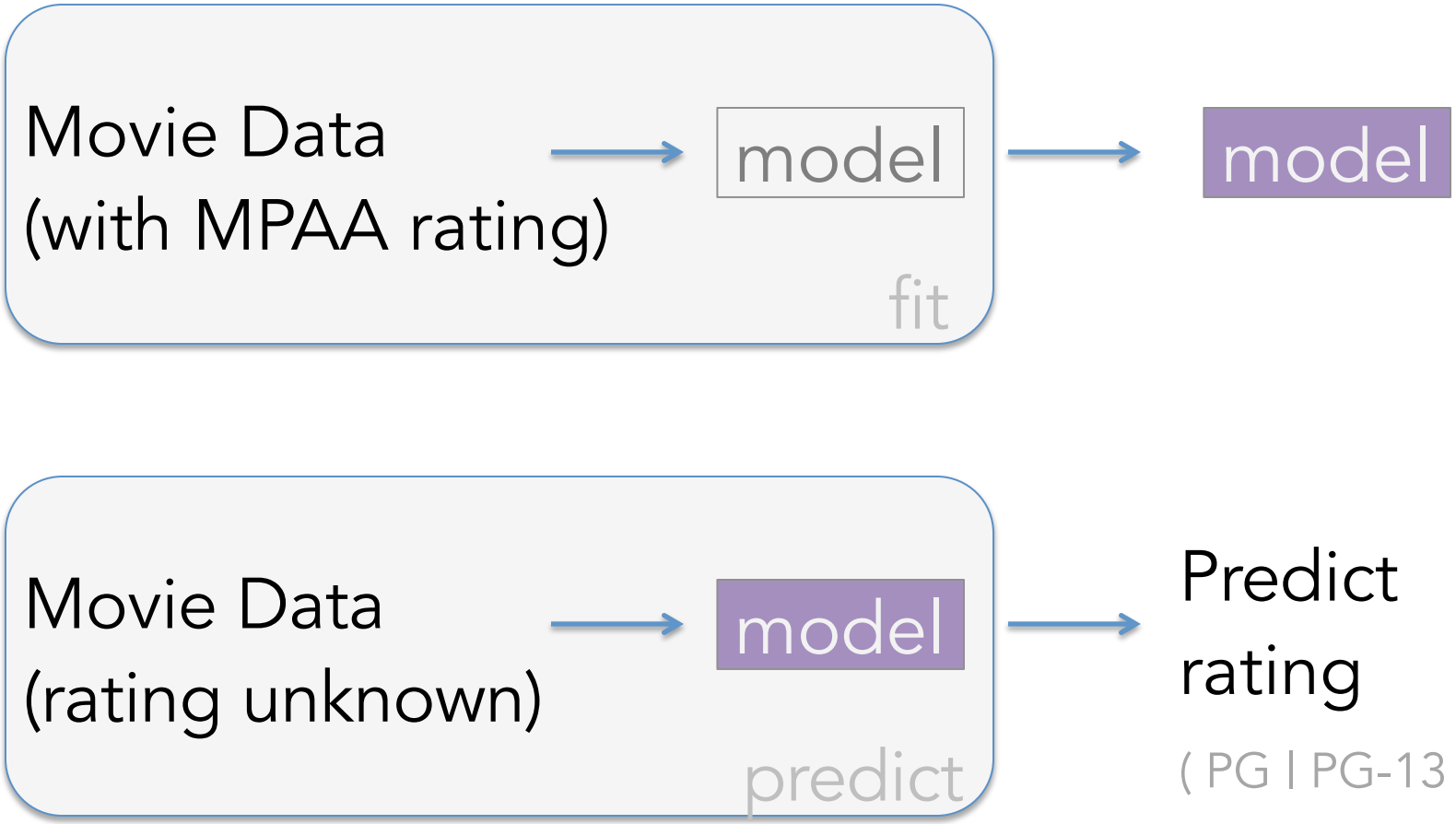
Regression:
"answers" are numeric



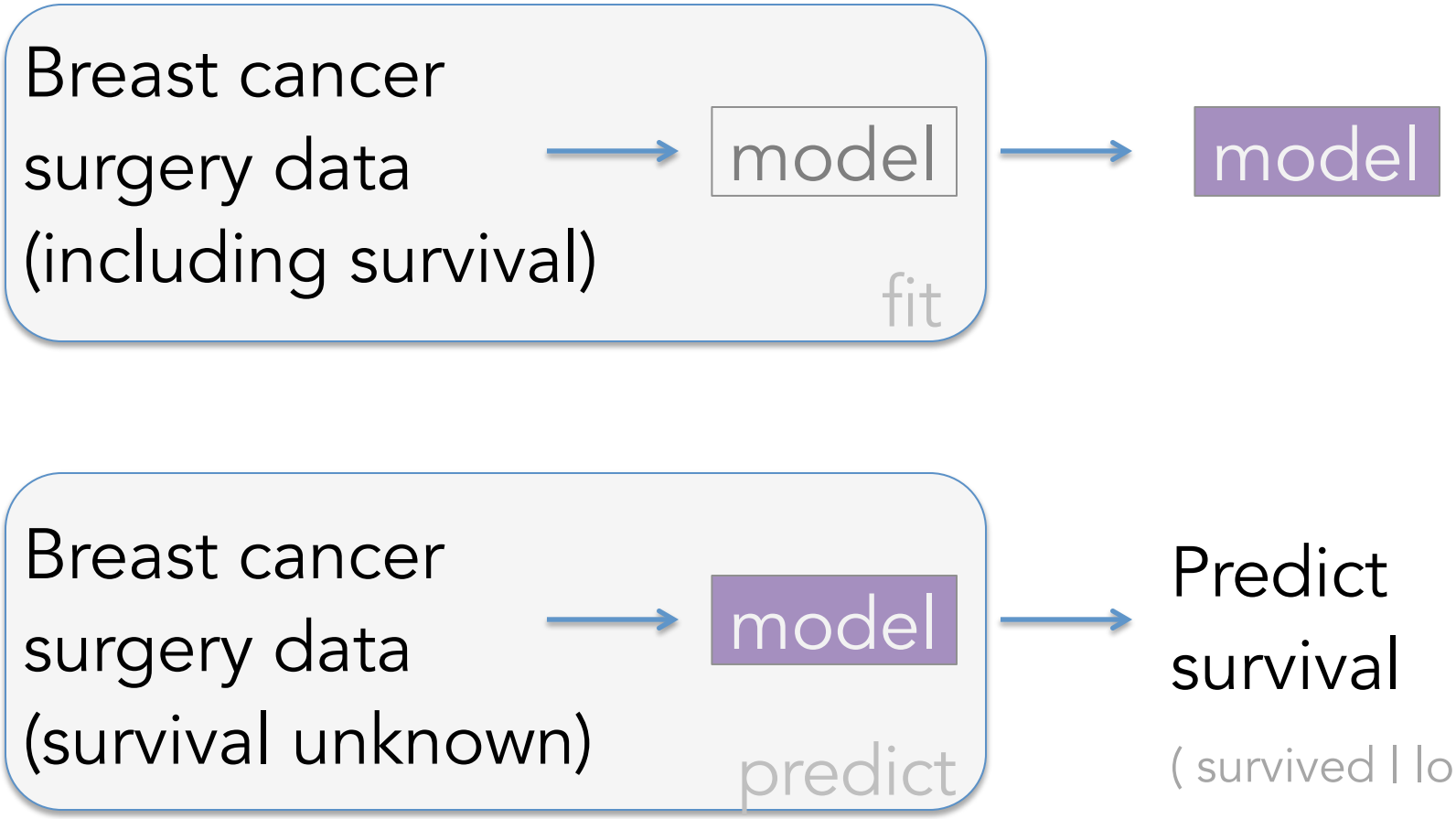
Classification:
"answers" are categories



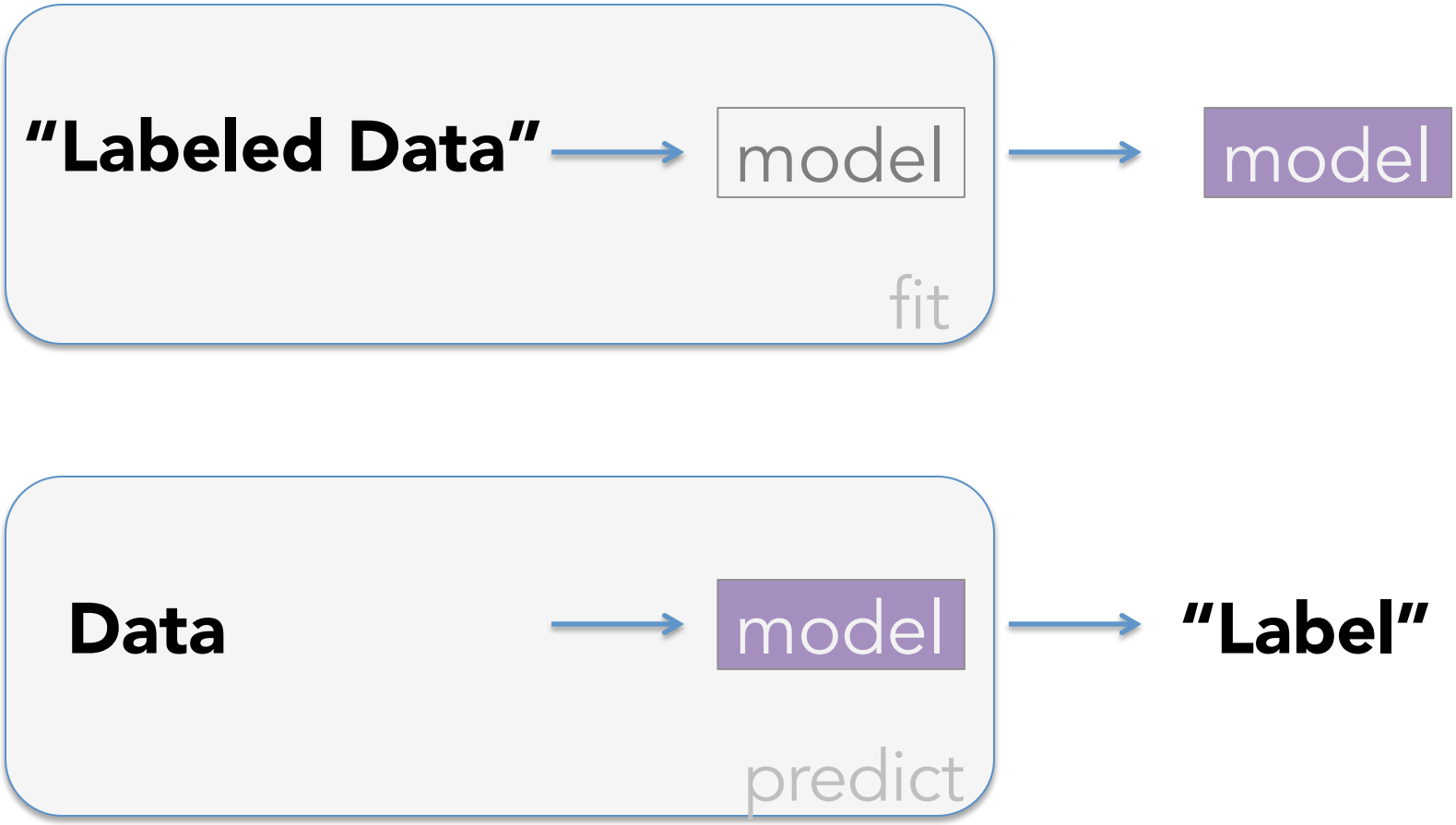
Classification:
"answers" are categories



Classification:
"answers" are categories



Classification:
"answers" are categories



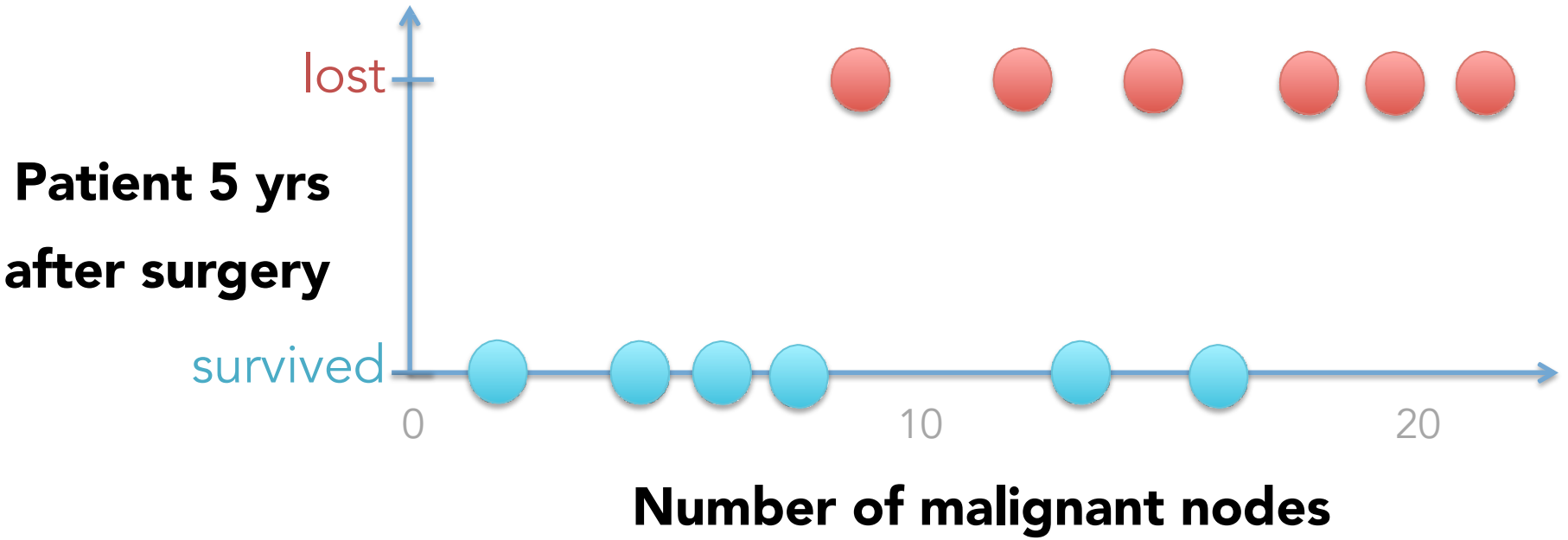
Example each data point
(one row)

Target predicted property
(column to predict)

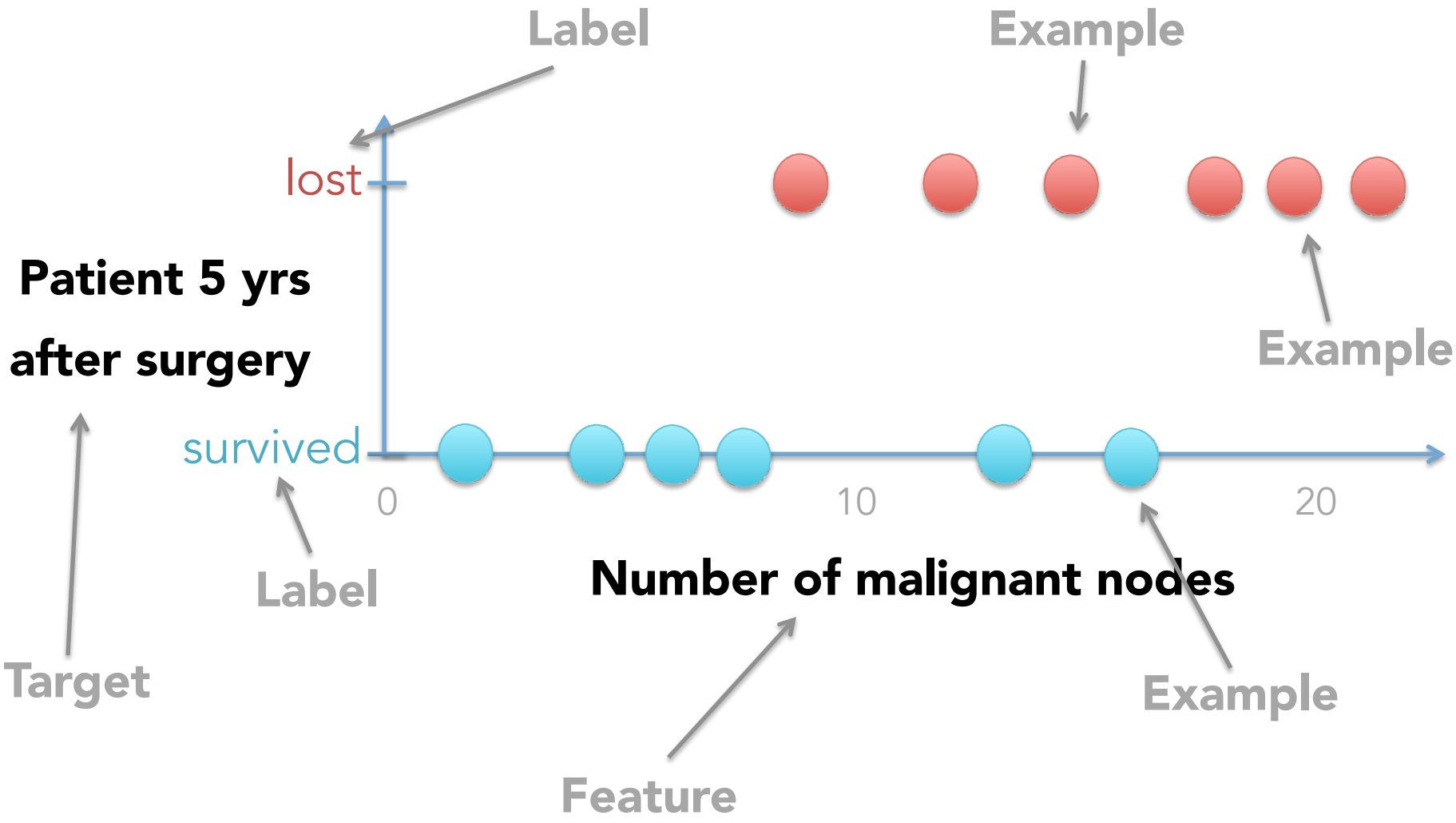
Label Target / category of the point
(value of target column)

Feature a property of the point
used for prediction
(non-target columns in the model)

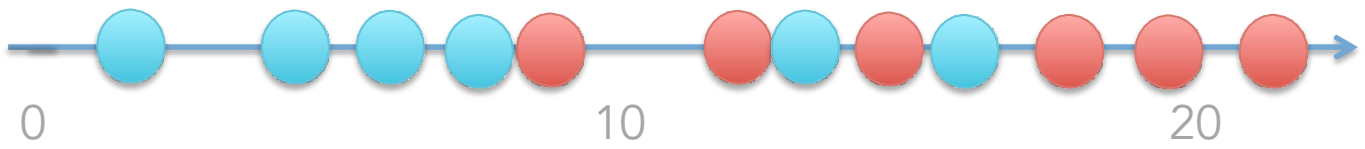
1 Feature. Number of malignant nodes
2 Labels. Survived / Lost



1 Feature. Number of malignant nodes
2 Labels. Survived / Lost

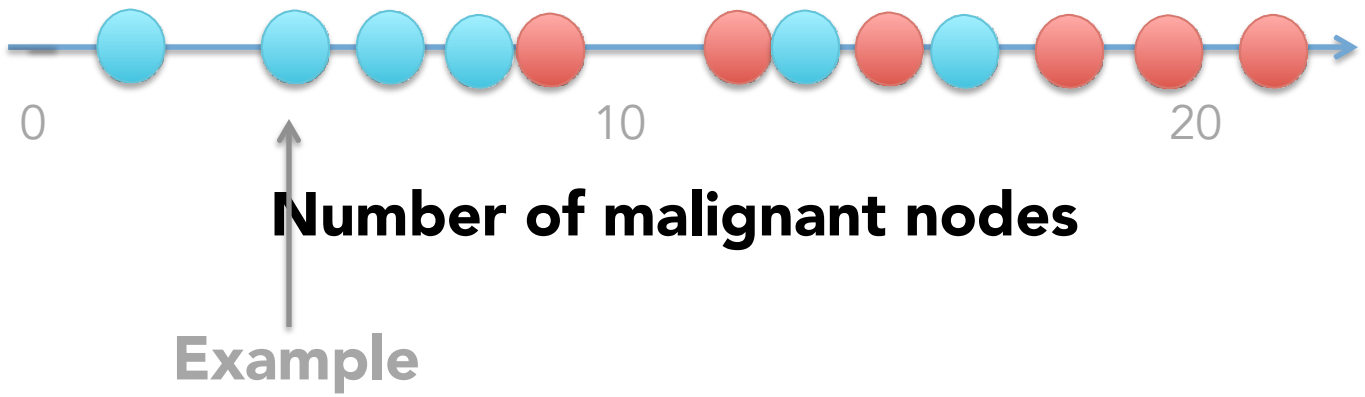


1 Feature. Number of malignant nodes
2 Labels. Survived / Lost

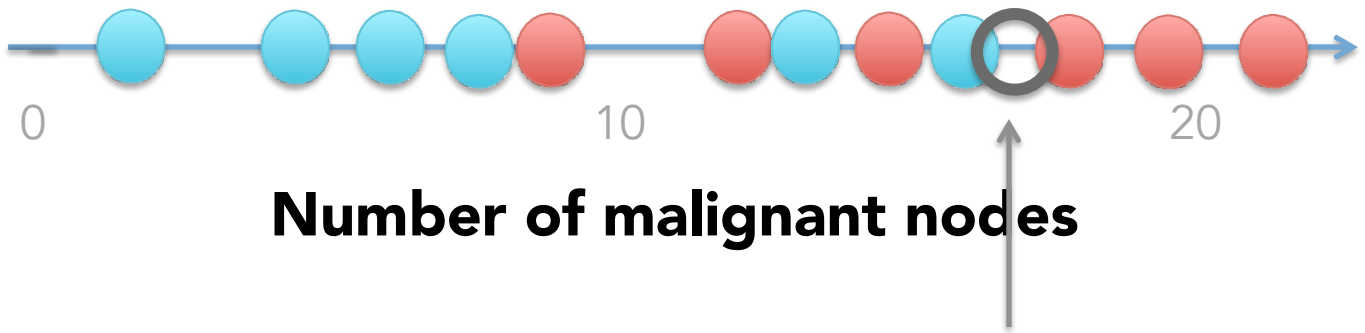


Number of malignant nodes

1 Feature. Number of malignant nodes
2 Labels. Survived / Lost

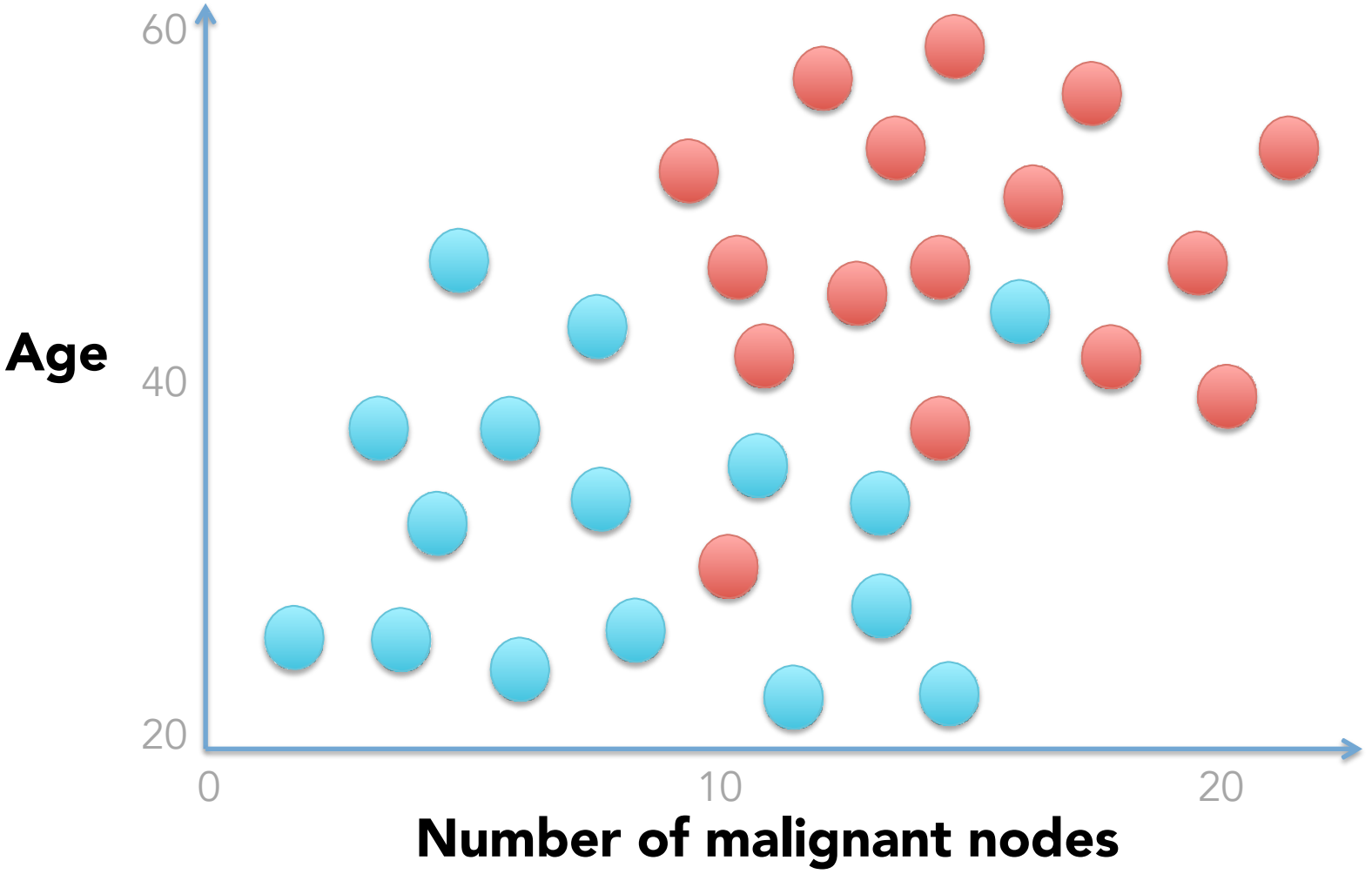


1 Feature. Number of malignant nodes
2 Labels. Survived / Lost

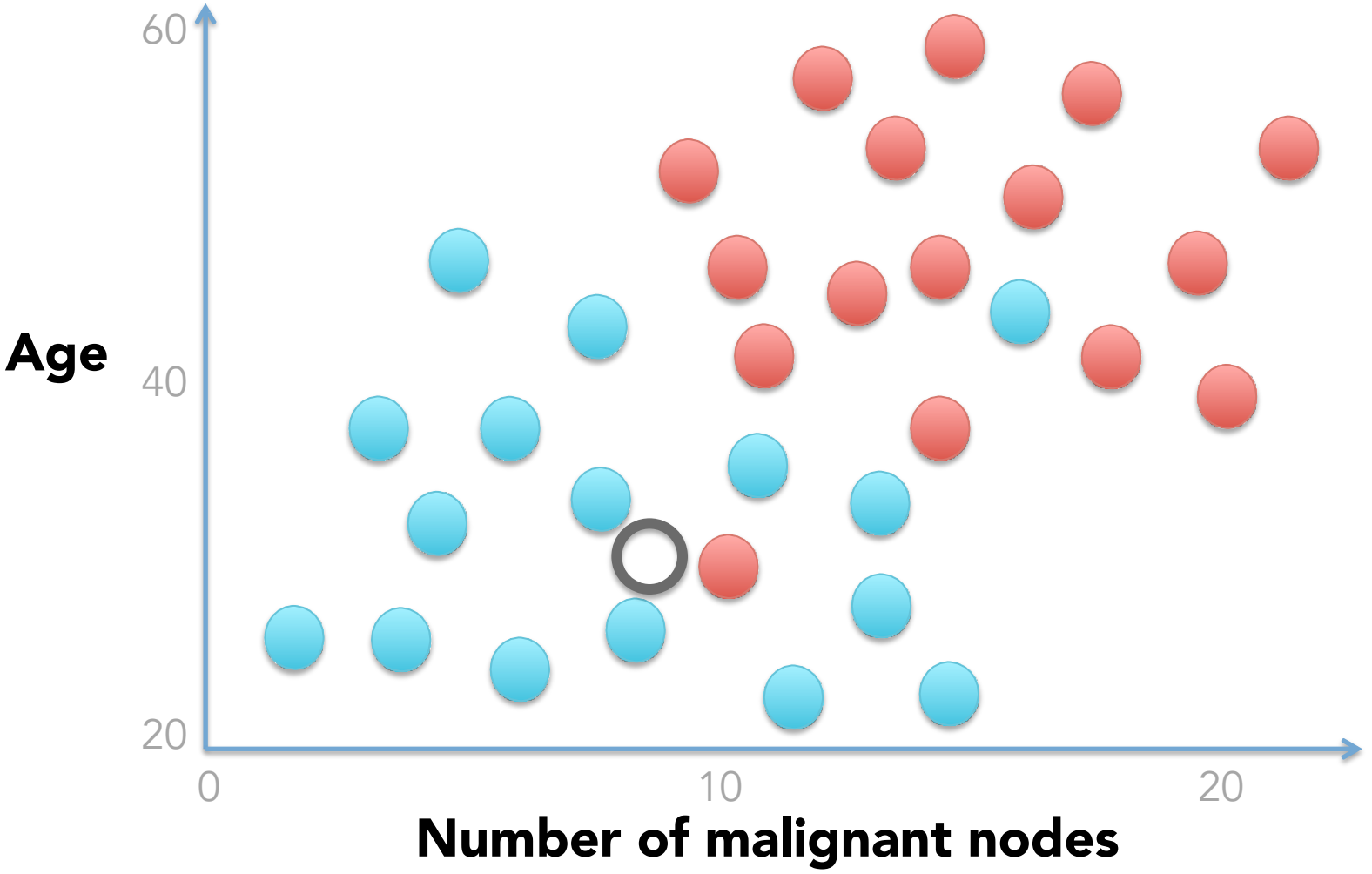


2 Features. No of malignant nodes / **Age**

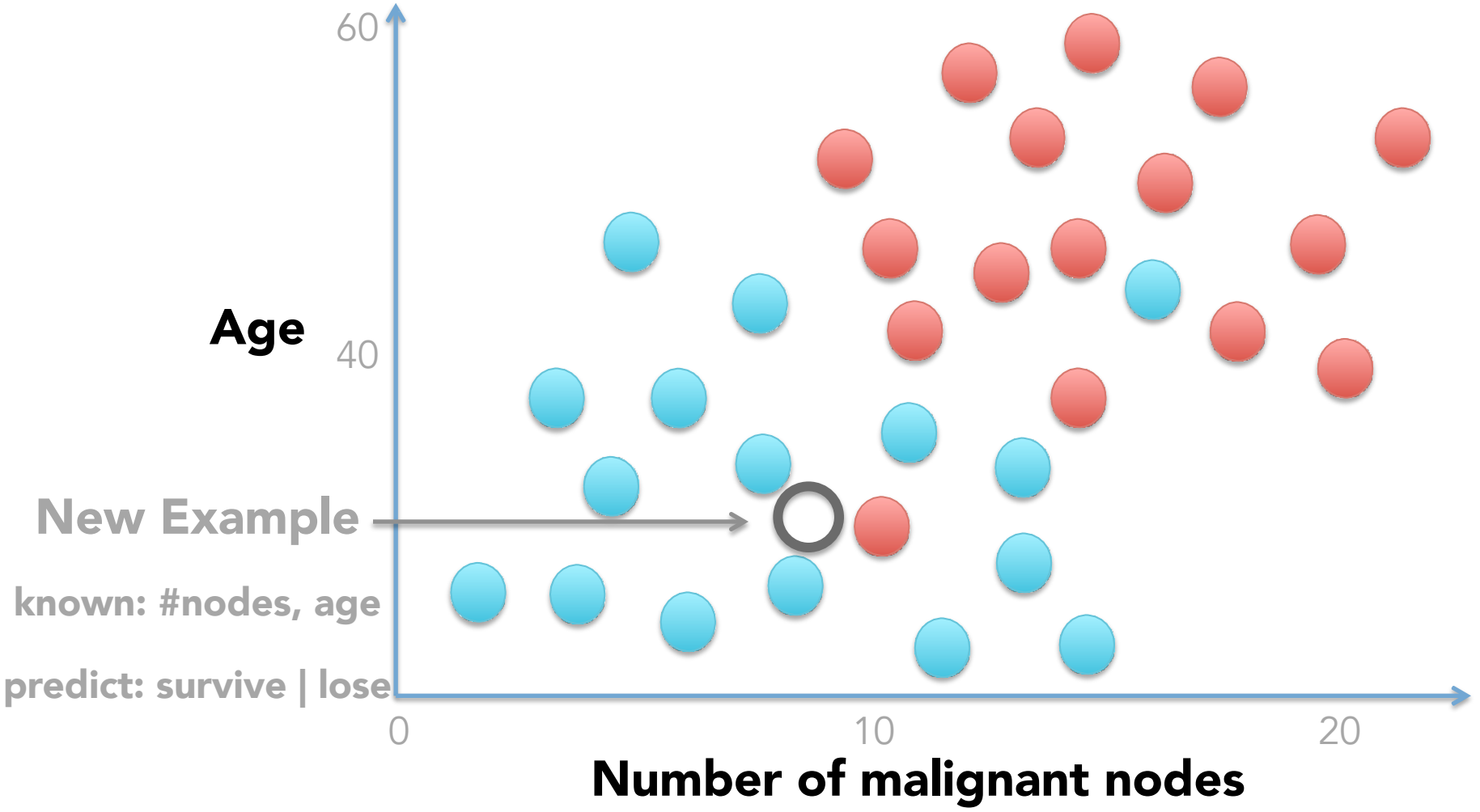
2 Labels. Survived / Lost



2 Features. No of malignant nodes / Age
2 Labels. Survived / Lost

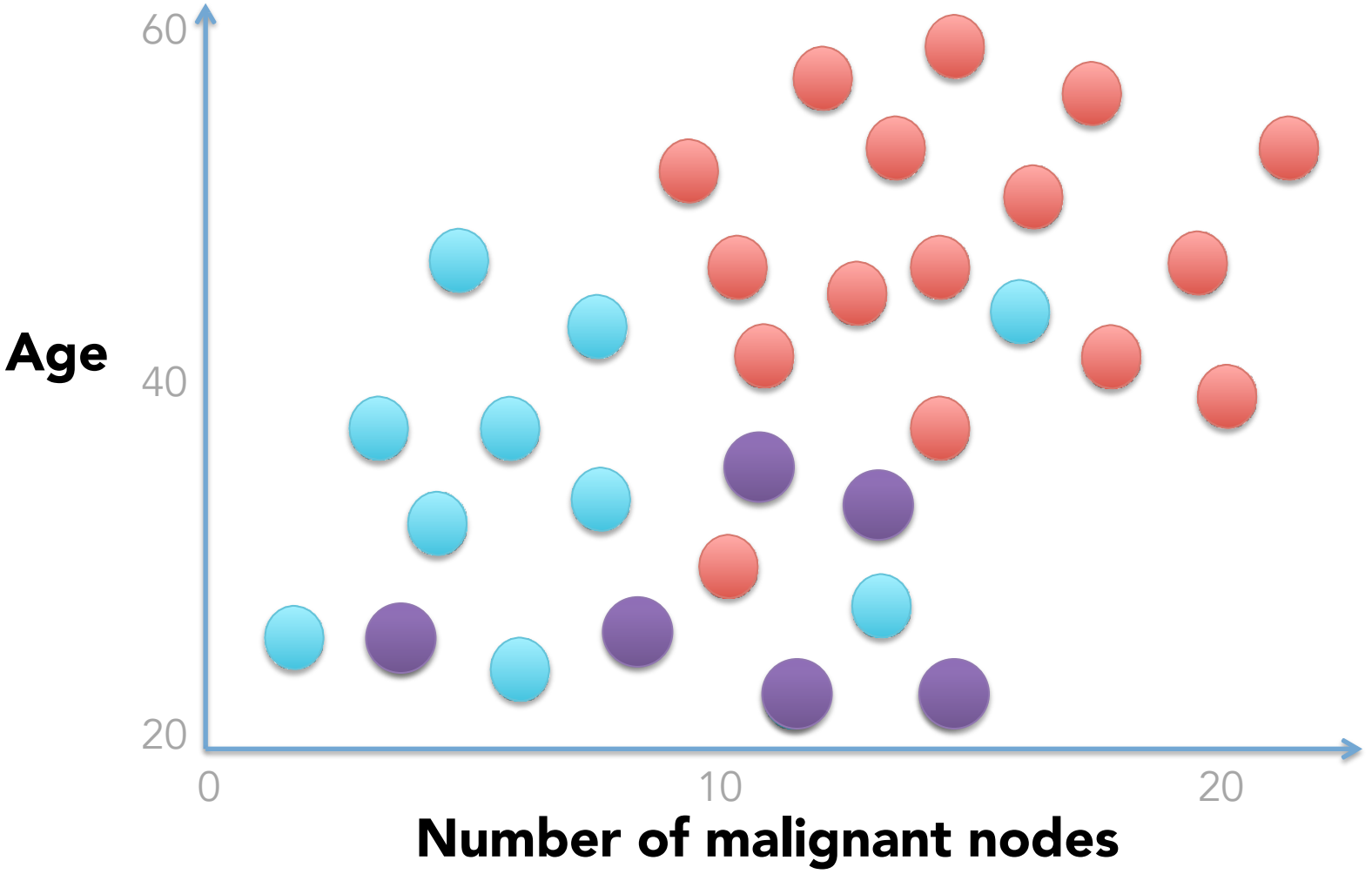


2 Features. No of malignant nodes / Age
2 Labels. Survived / Lost

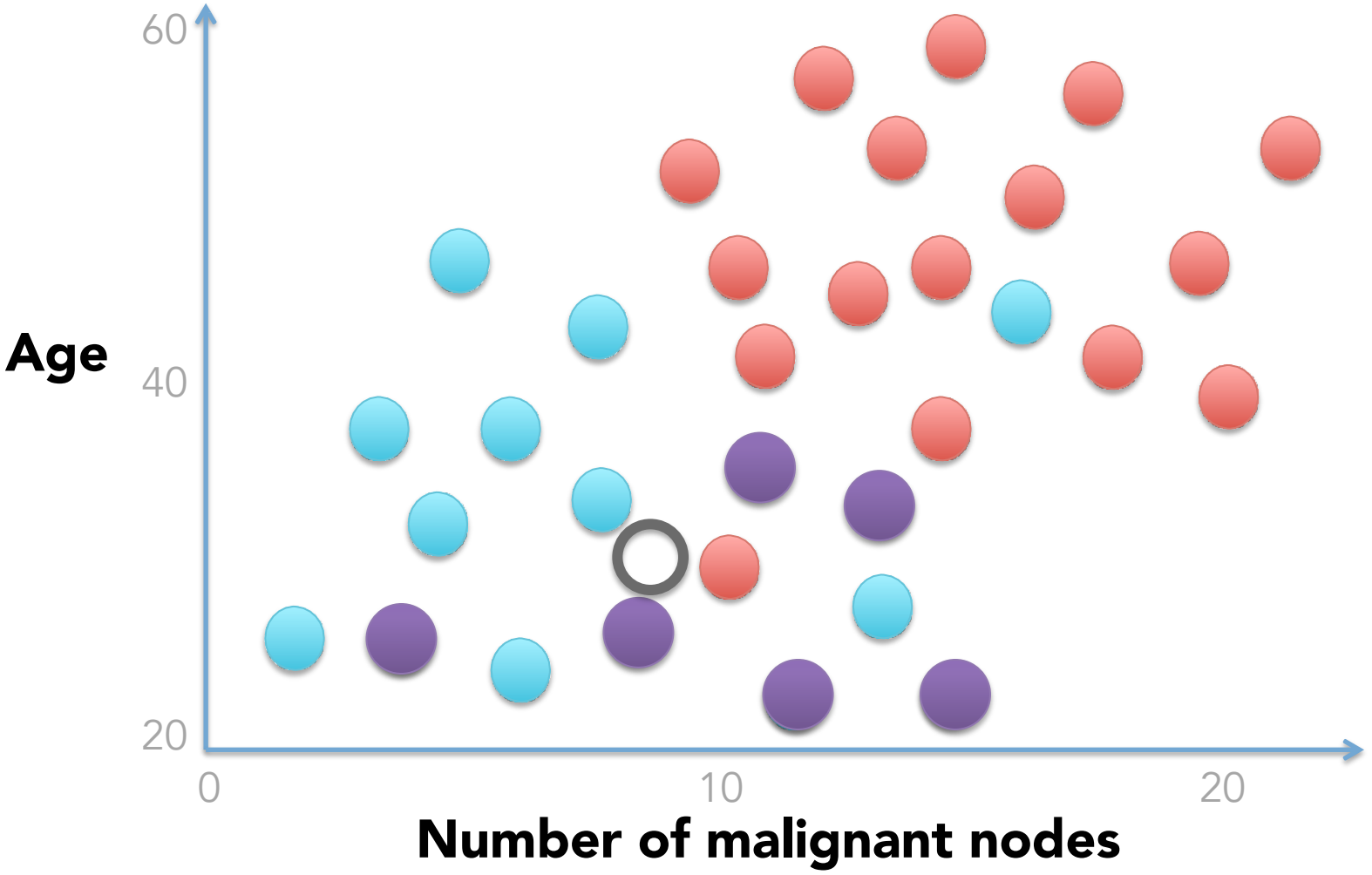


2 Features. No of malignant nodes / Age

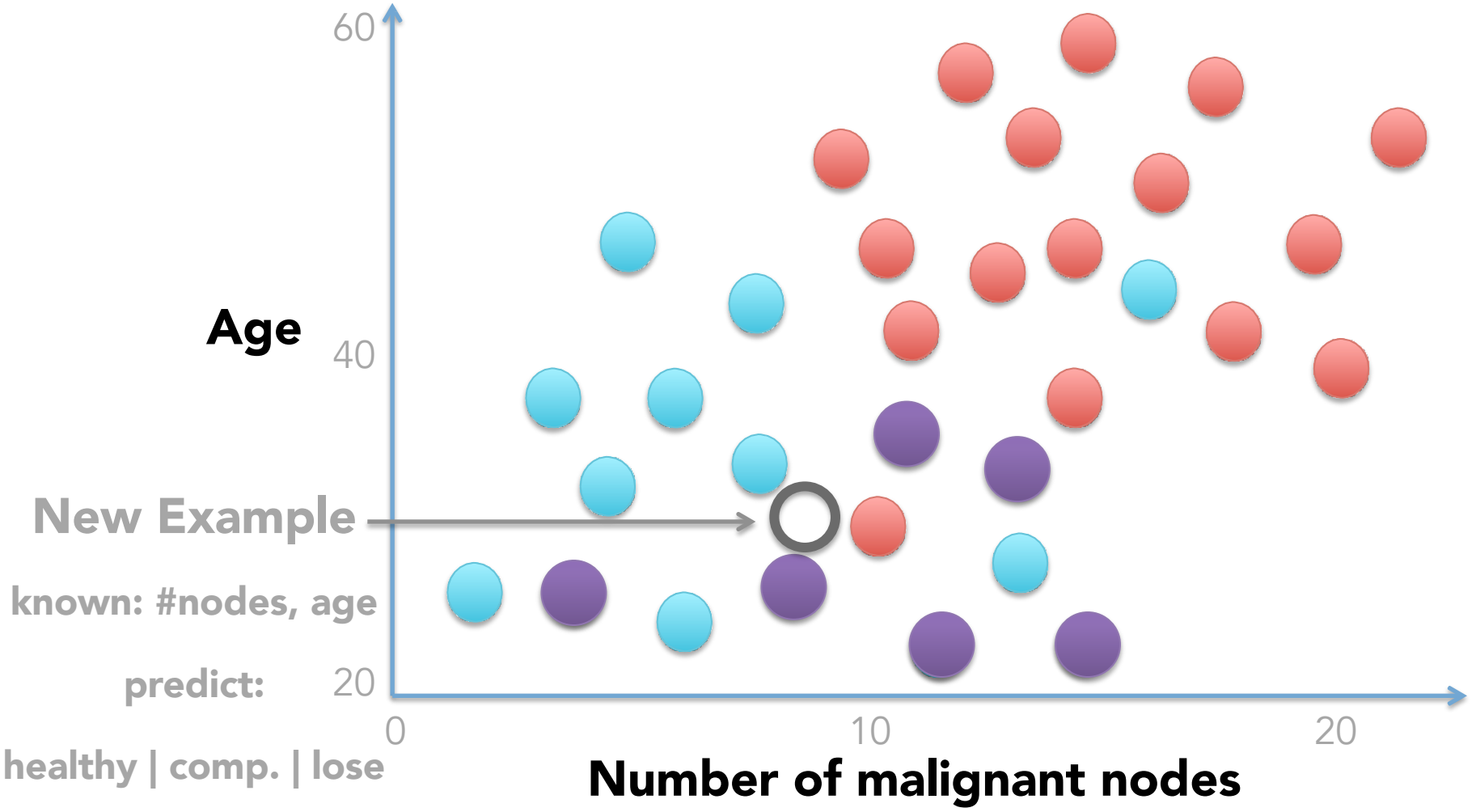
3 Labels. Healthy / Complications / Lost



2 Features. No of malignant nodes / Age
3 Labels. Healthy / Complications / Lost

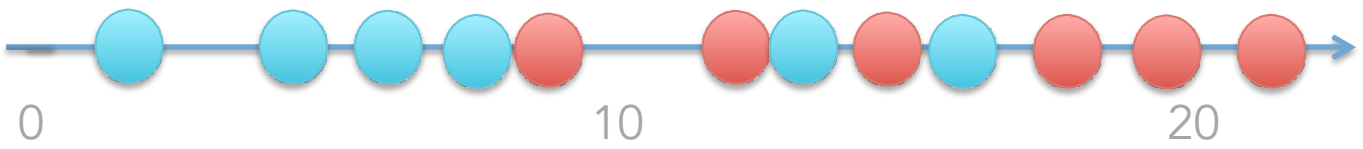


2 Features. No of malignant nodes / Age
3 Labels. Healthy / Complications / Lost



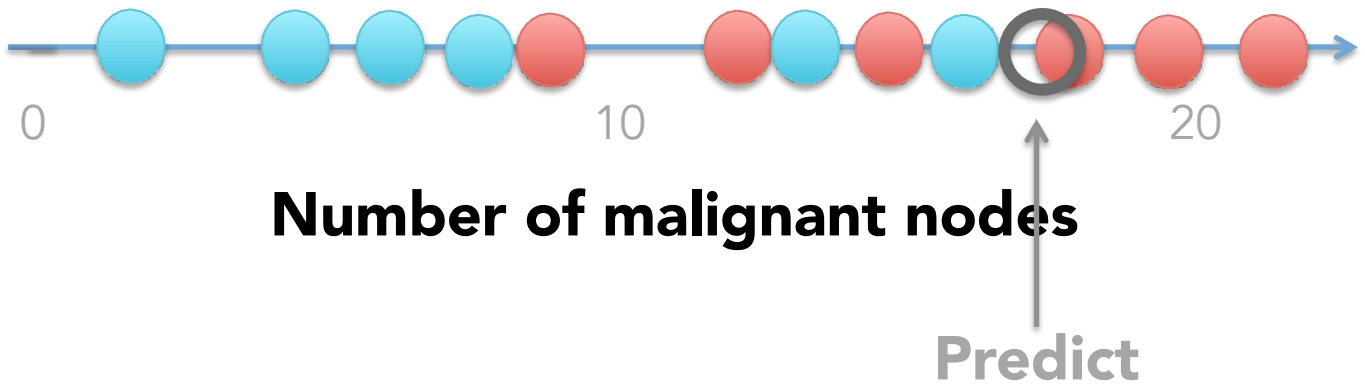
K NEAREST NEIGHBORS

K Nearest Neighbors



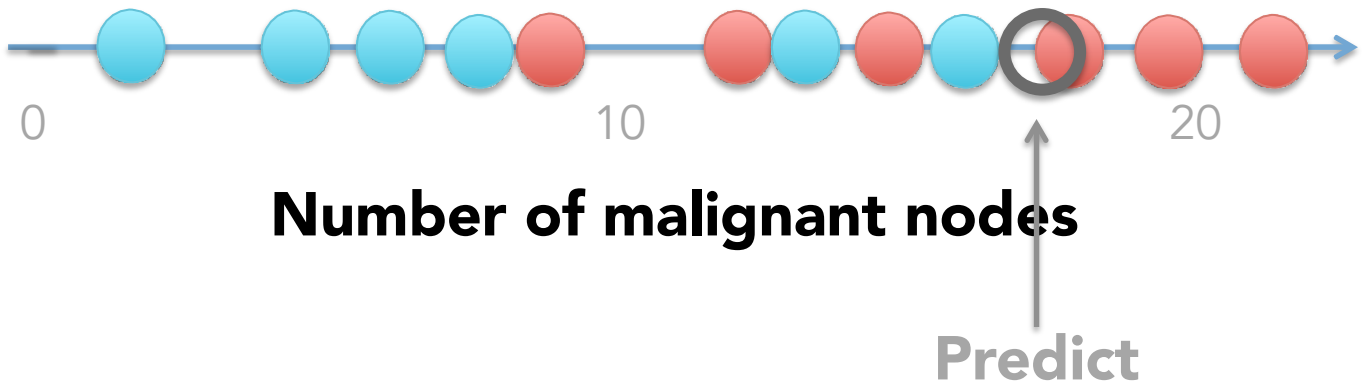
Number of malignant nodes

K Nearest Neighbors



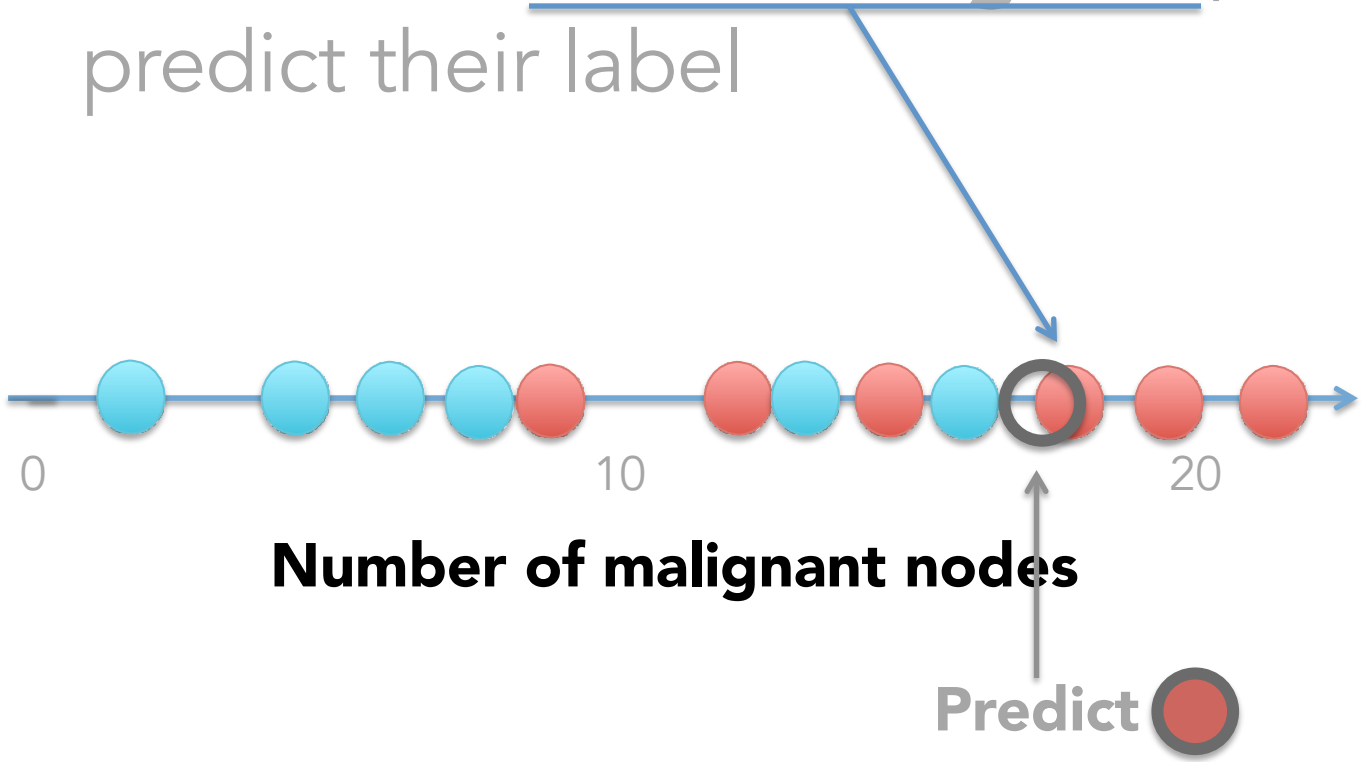
K Nearest Neighbors

K=1 Look at the nearest neighbor, predict their label



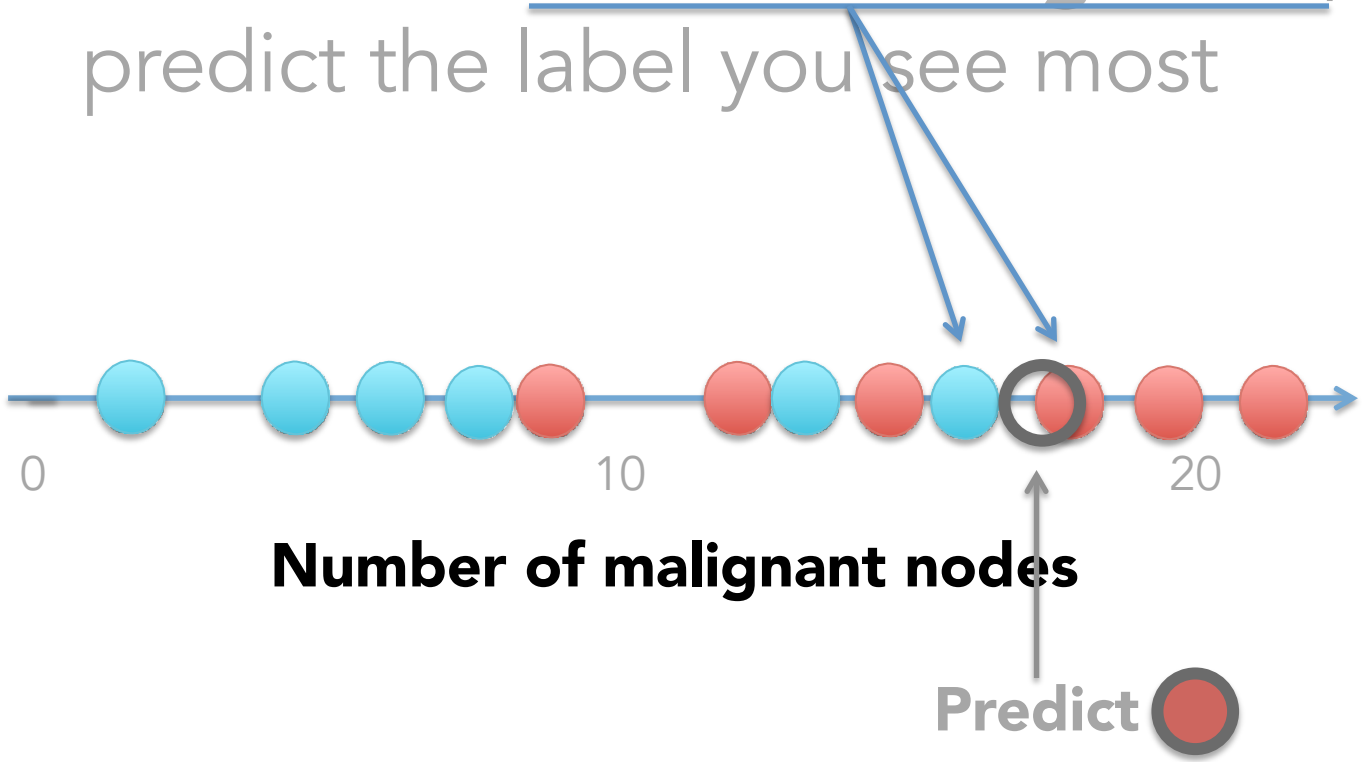
K Nearest Neighbors

K=1 Look at the **nearest neighbor**, predict their label



K Nearest Neighbors

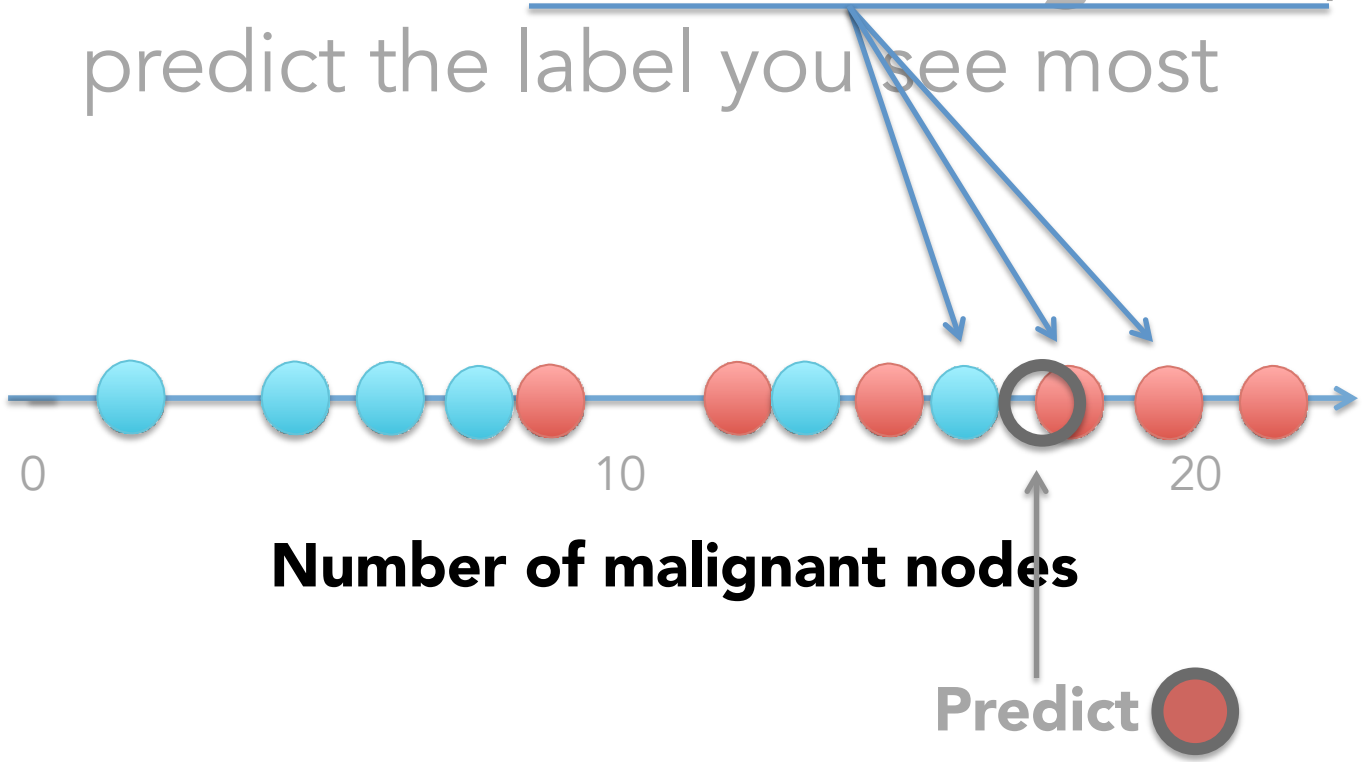
K=2 Look at the **2 nearest neighbors**, predict the label you see most



K Nearest Neighbors

K=3

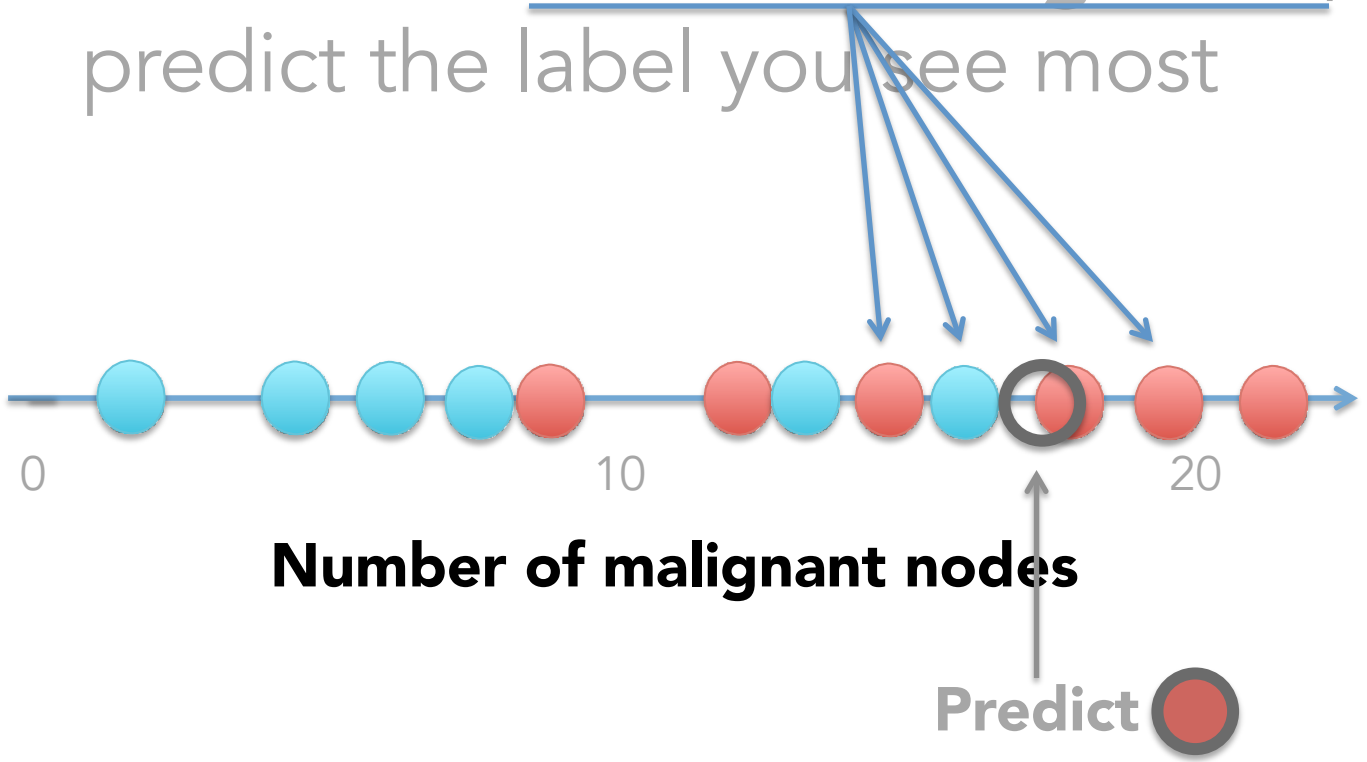
Look at the **3 nearest neighbors**, predict the label you see most



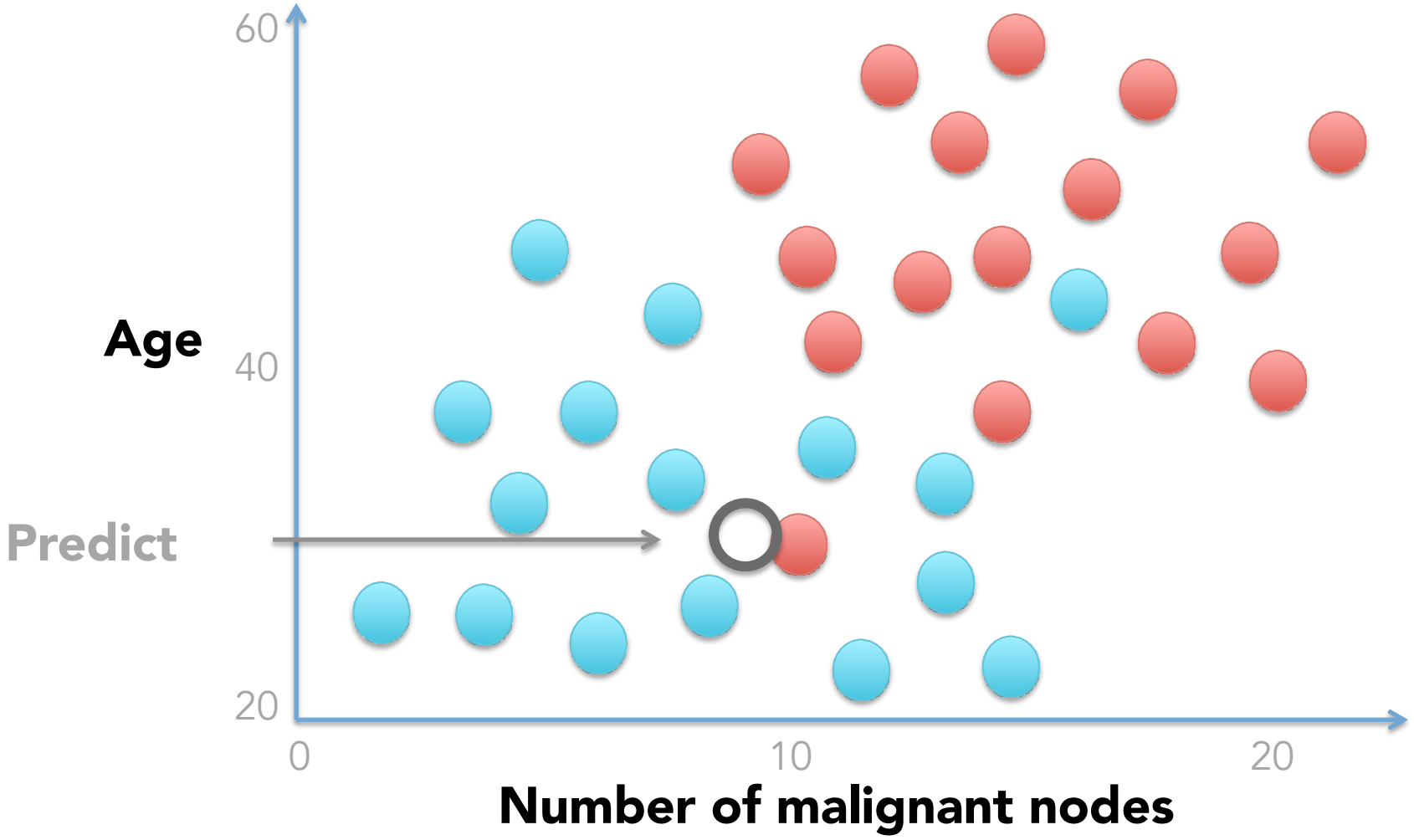
K Nearest Neighbors

K=4

Look at the **4 nearest neighbors**, predict the label you see most

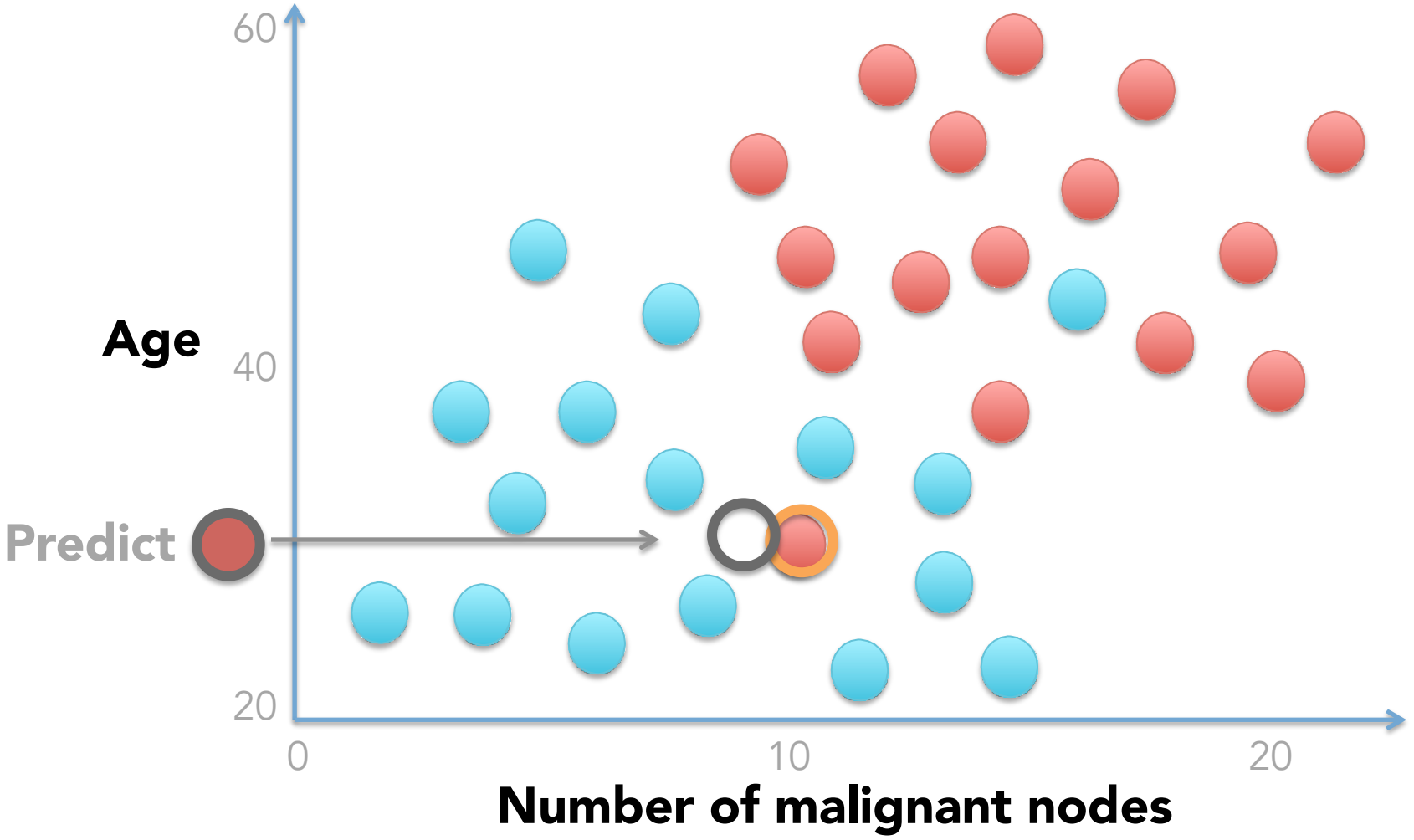


K Nearest Neighbors



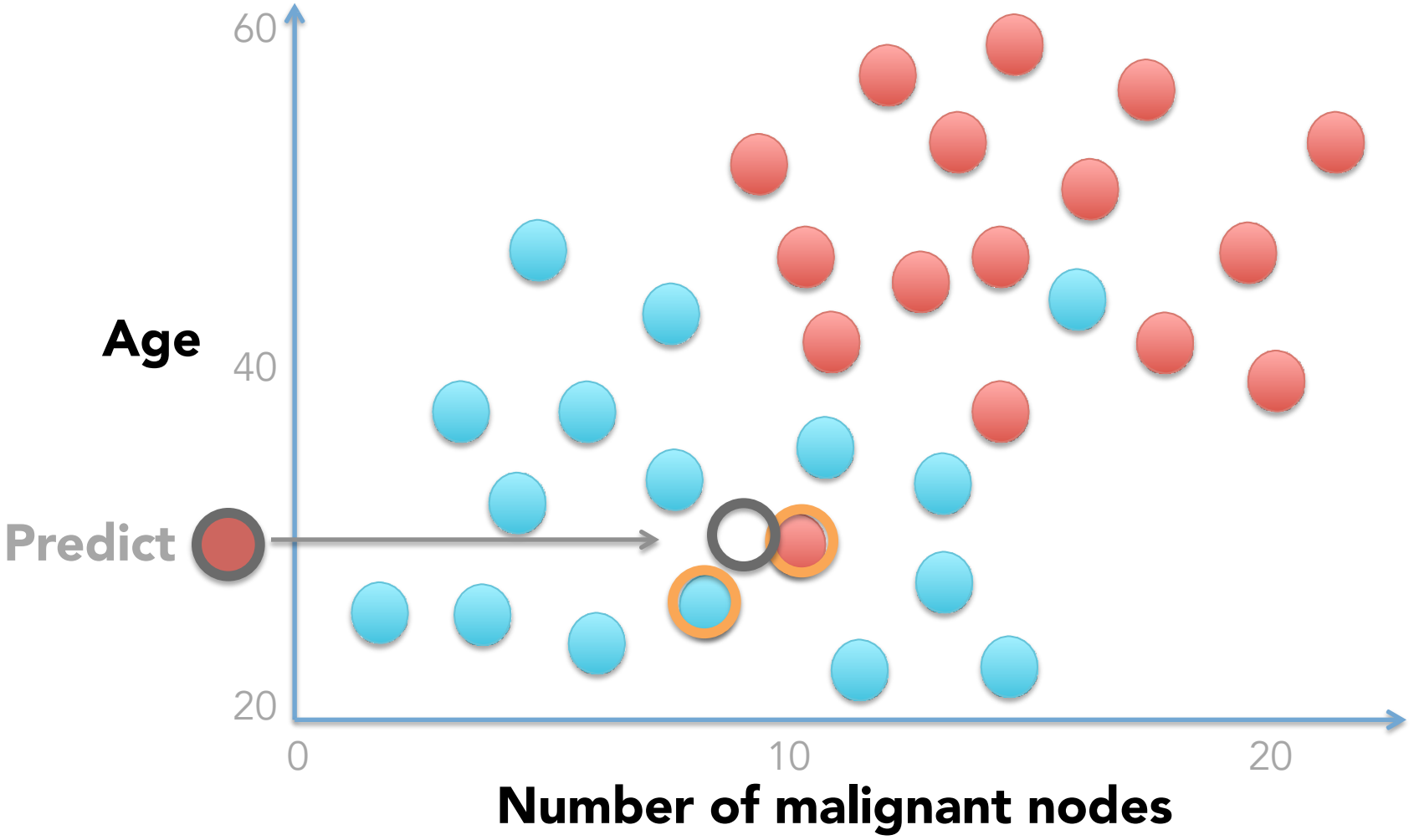
K Nearest Neighbors K=1

Neighbor count: 0 1



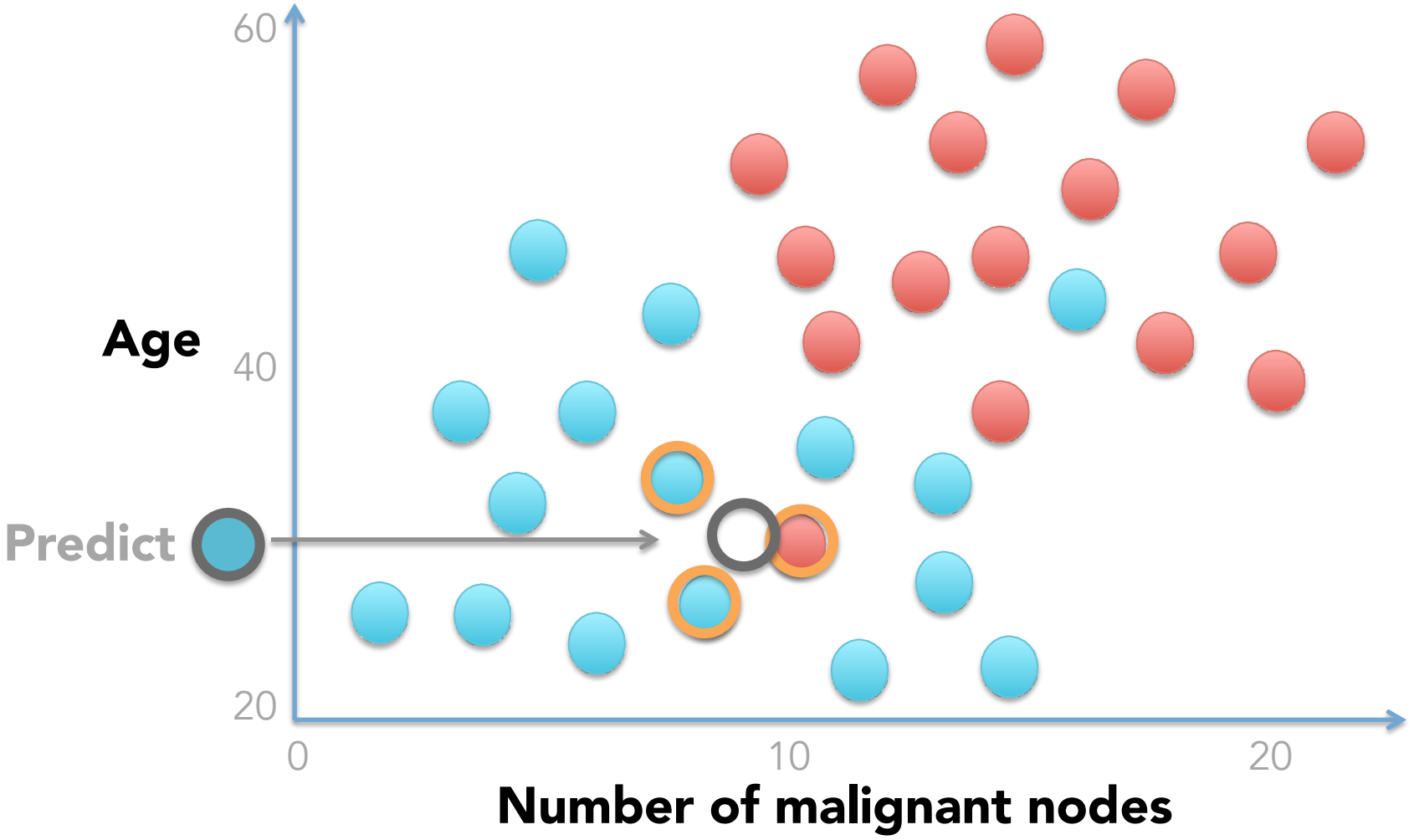
K Nearest Neighbors K=2

Neighbor count: 1 1



K Nearest Neighbors K=3

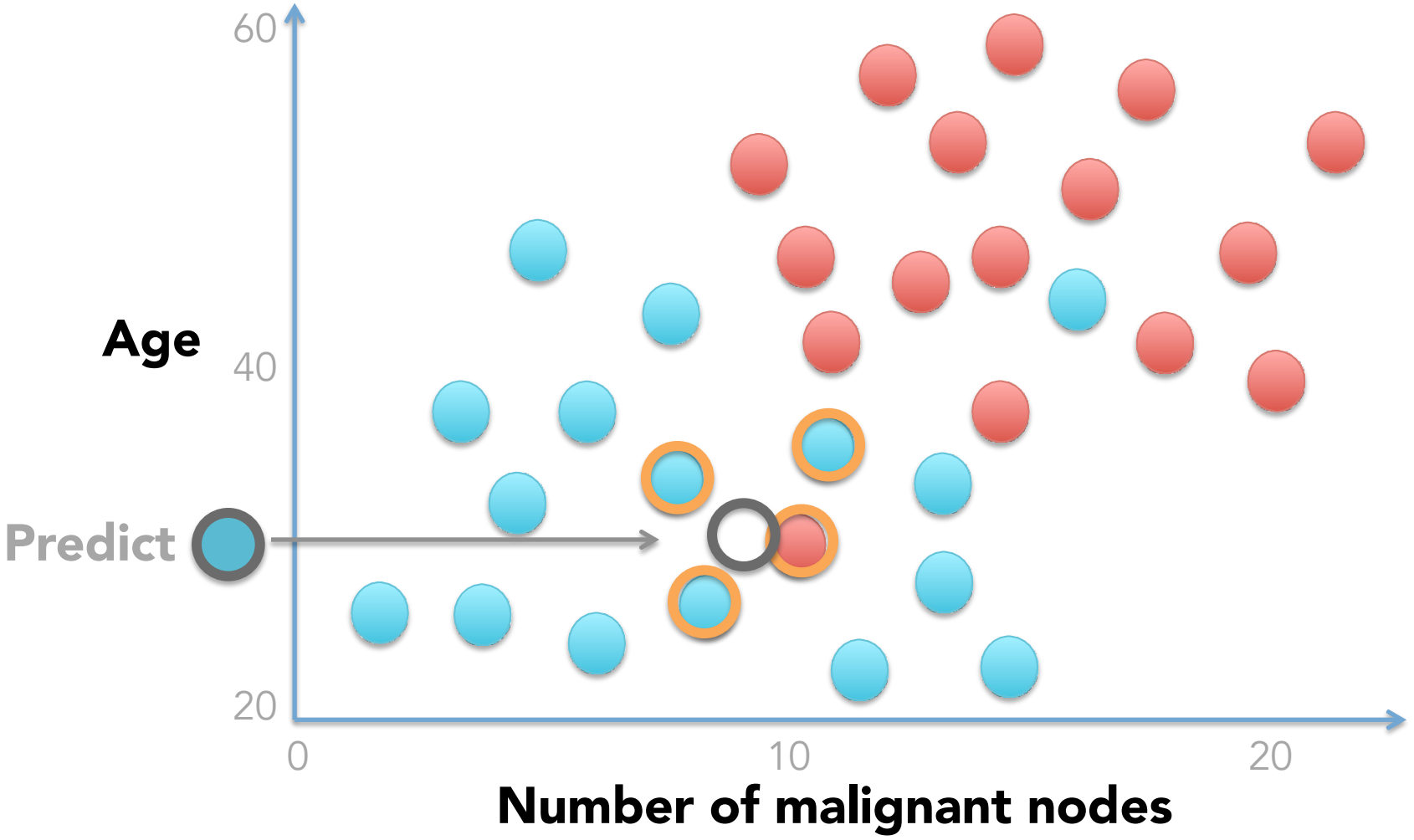
Neighbor count: 2 1



K Nearest Neighbors

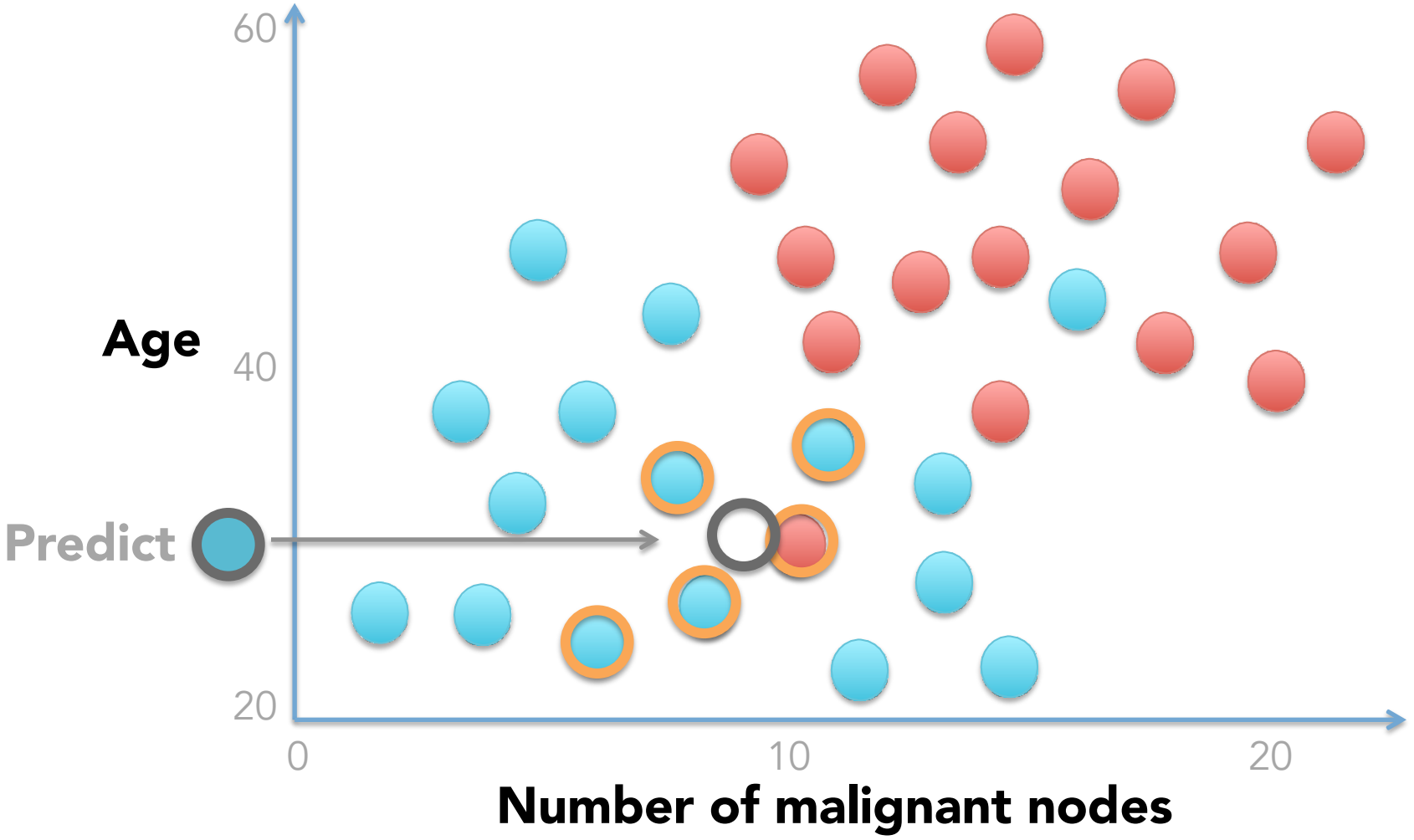
K=4

Neighbor count: 3 1



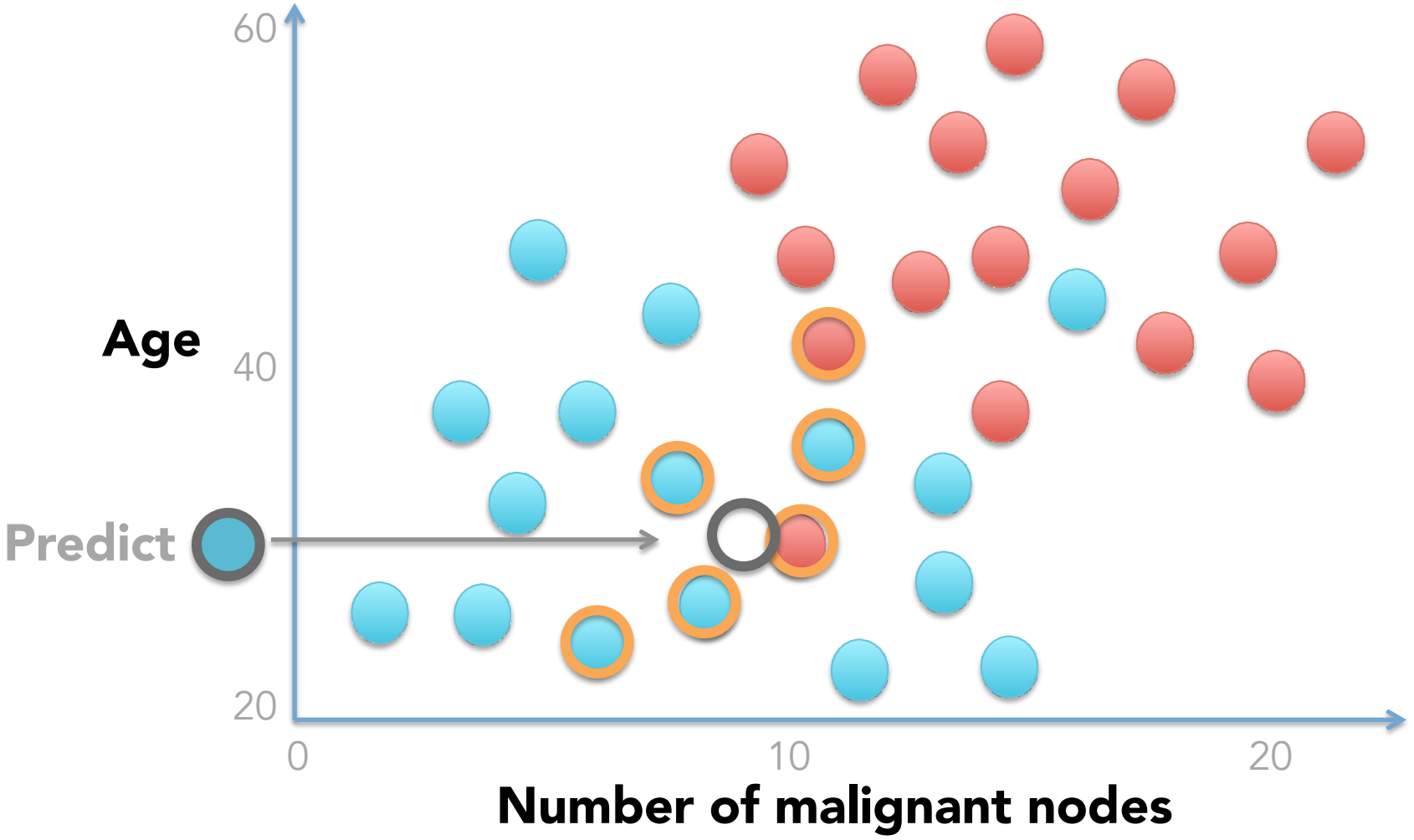
K Nearest Neighbors K=5

Neighbor count: 4 1

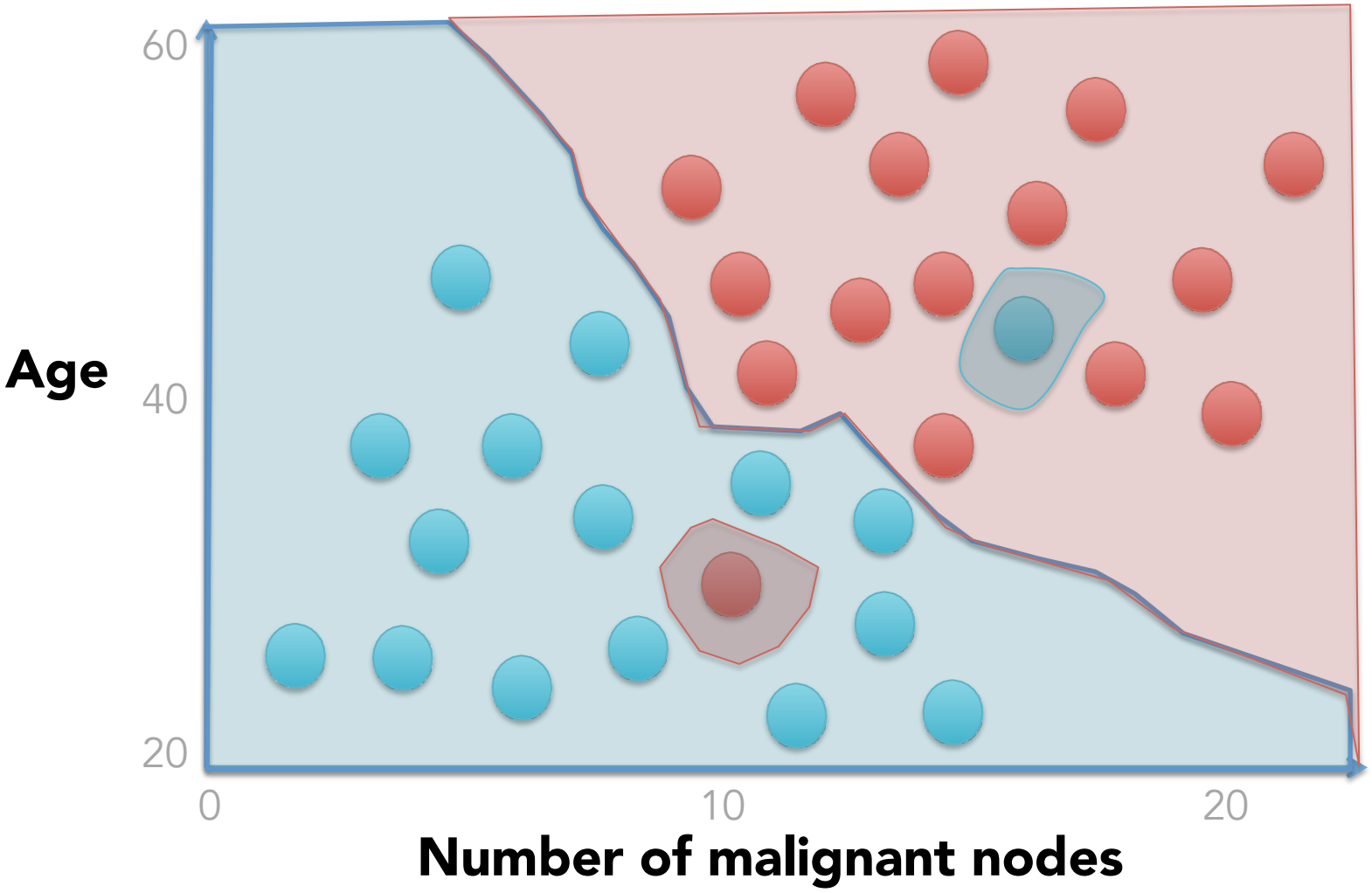


K Nearest Neighbors K=6

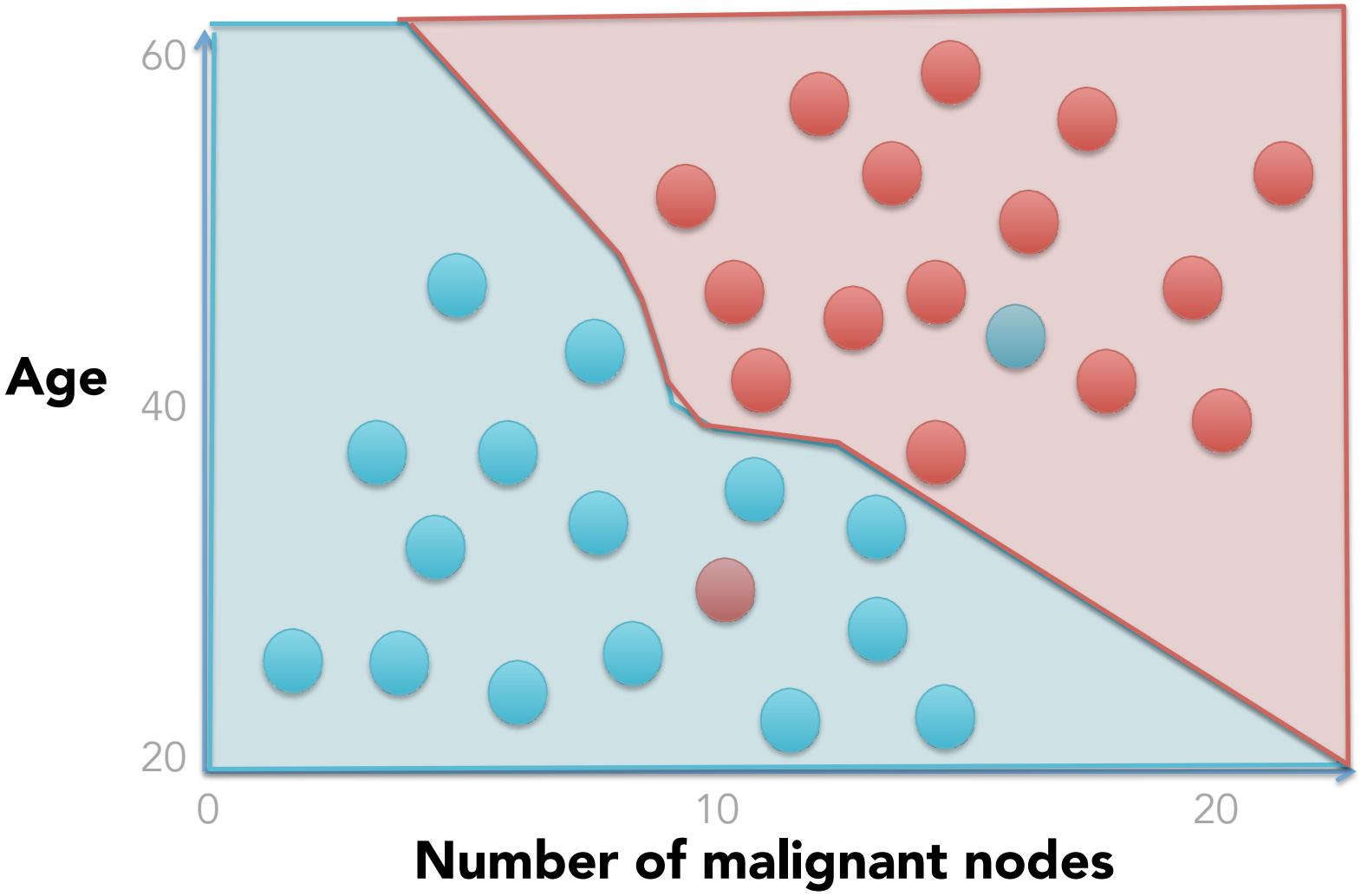
Neighbor count: 4 2



KNN Decision Boundary K=1

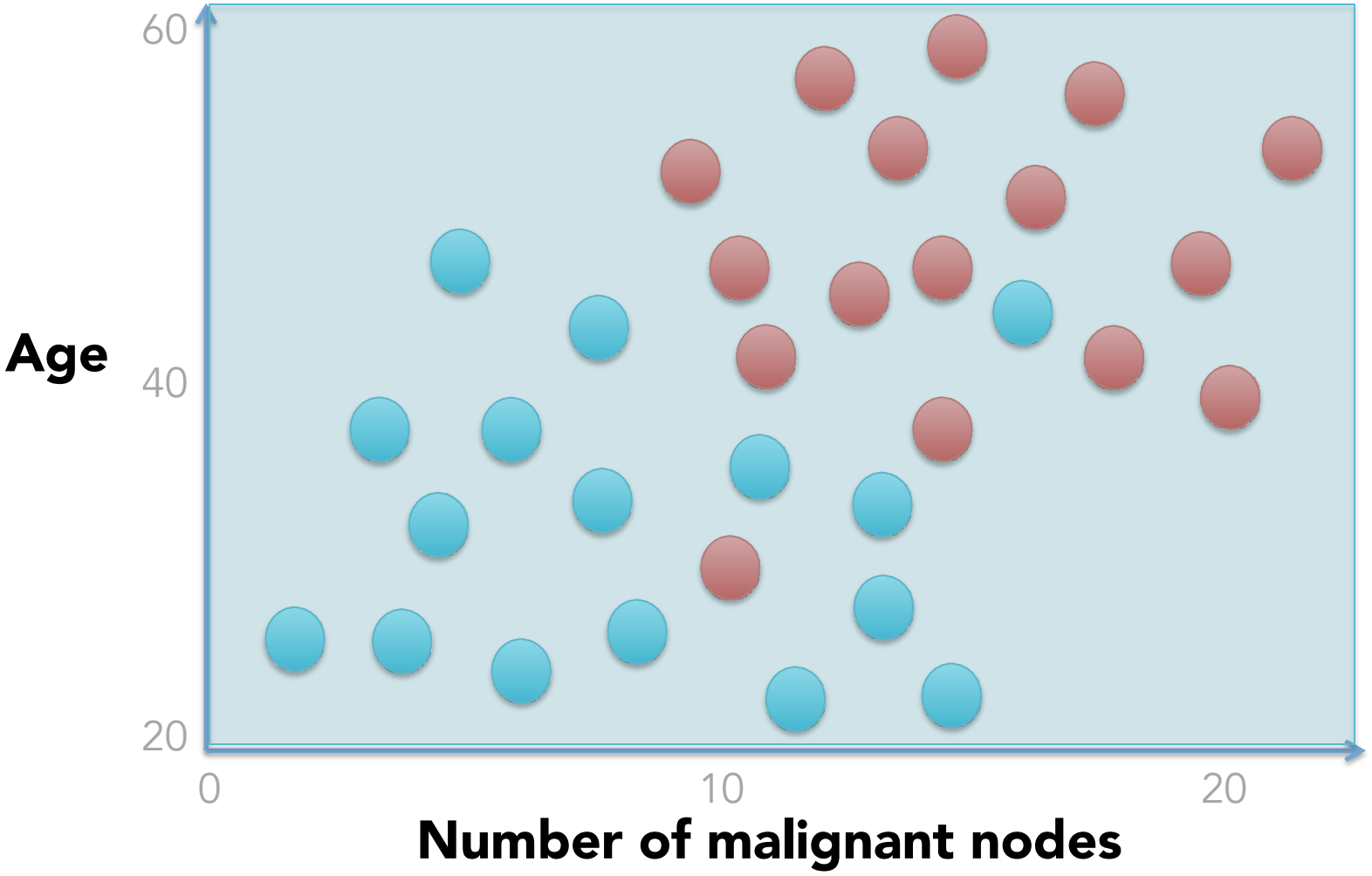


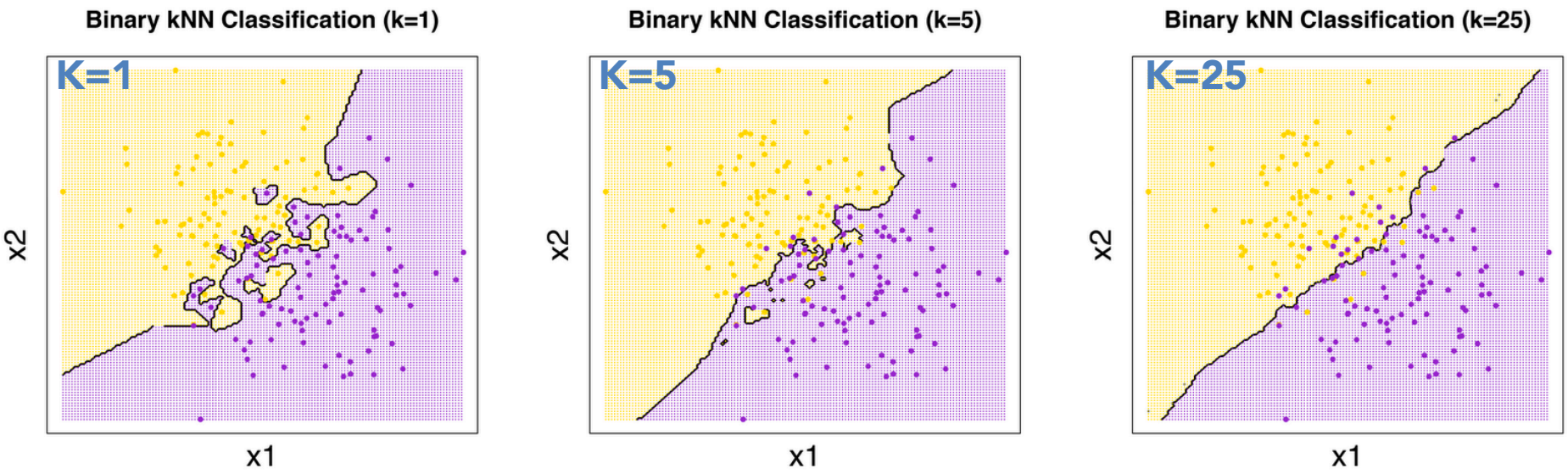
KNN Decision Boundary K=5

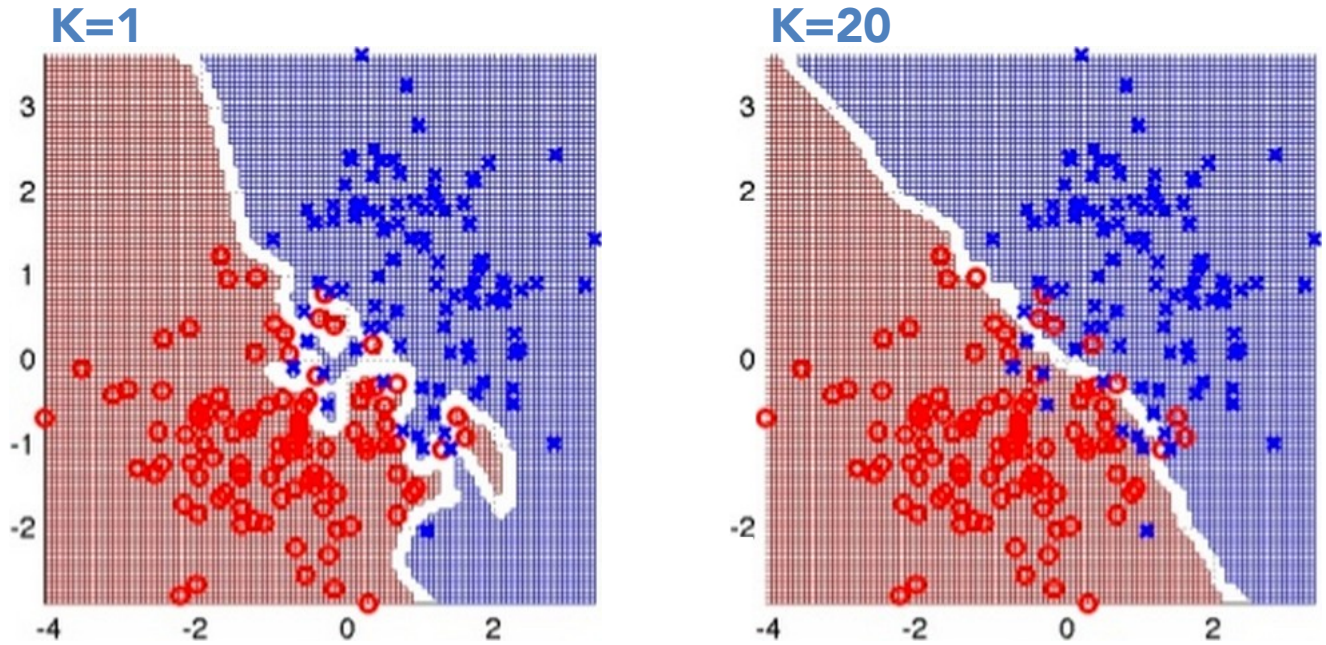


KNN Decision Boundary K=34

Total counts: 18 16



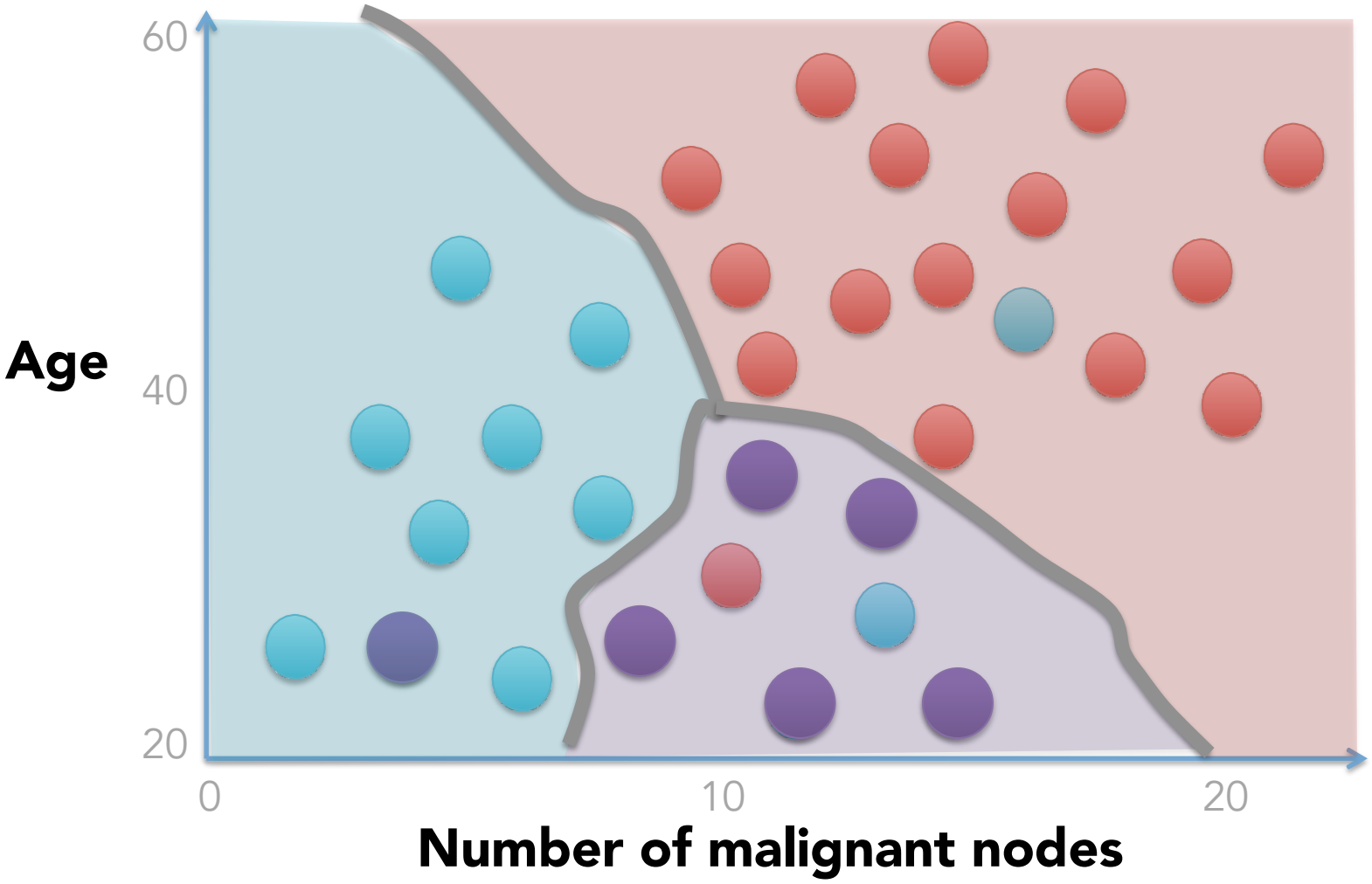




http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/

Multiclass KNN Decision Boundary

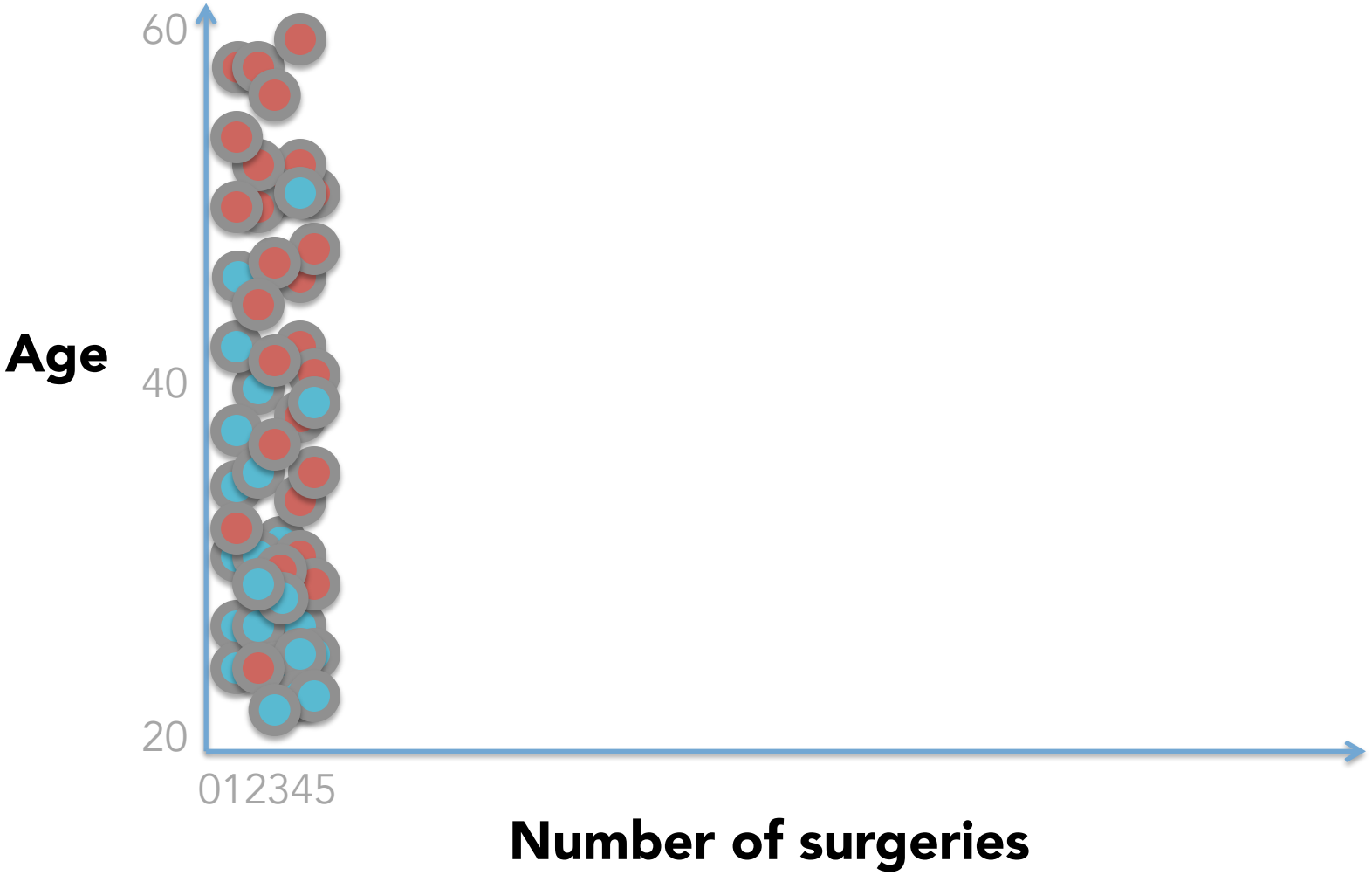
K=5



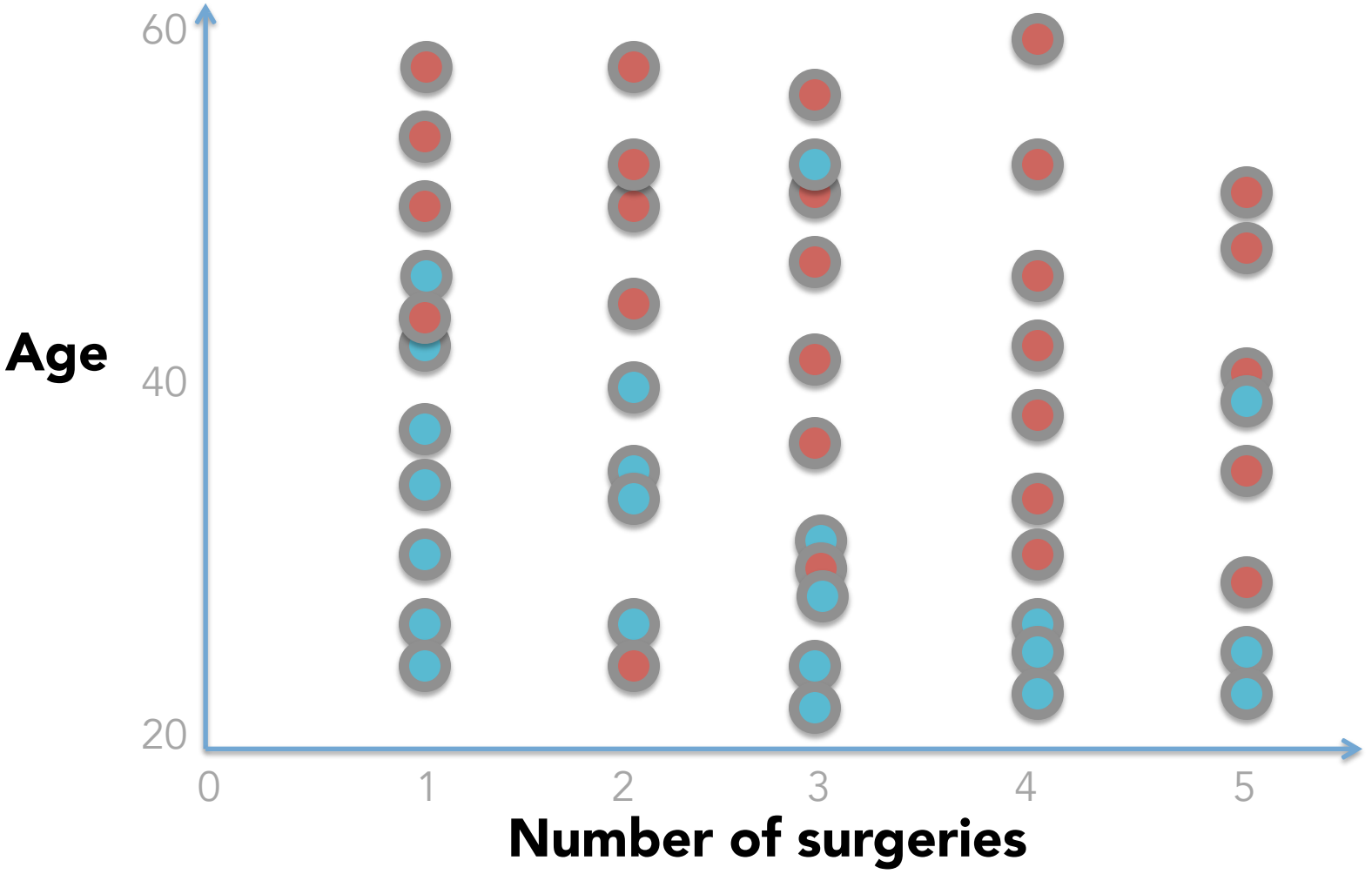
```
from sklearn.neighbors import KNeighborsClassifier


# Interface not different from LinearRegression at all
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, Y_train)
Y_pred = model.predict(X_test)
```

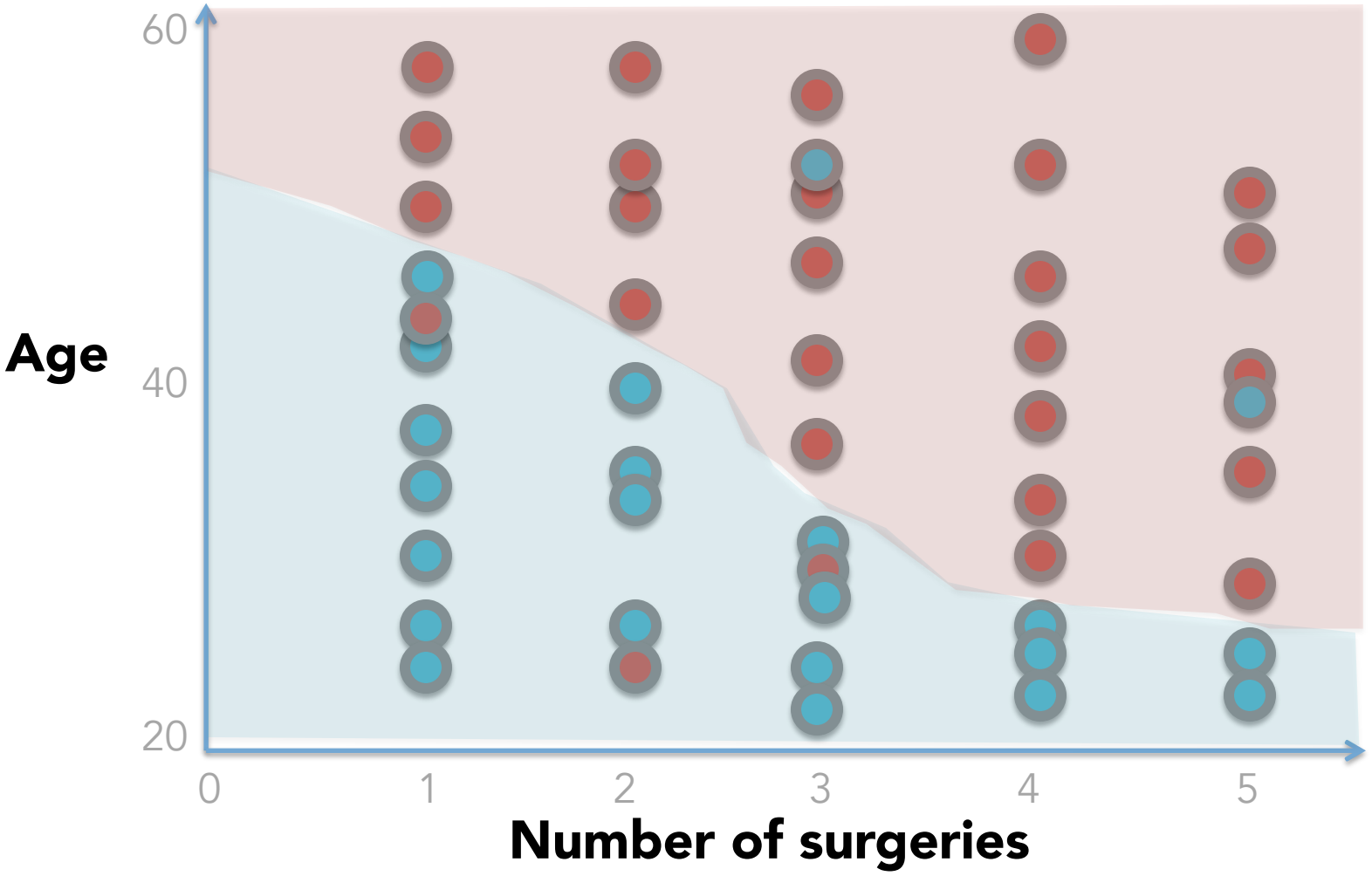
Scaling is crucial!



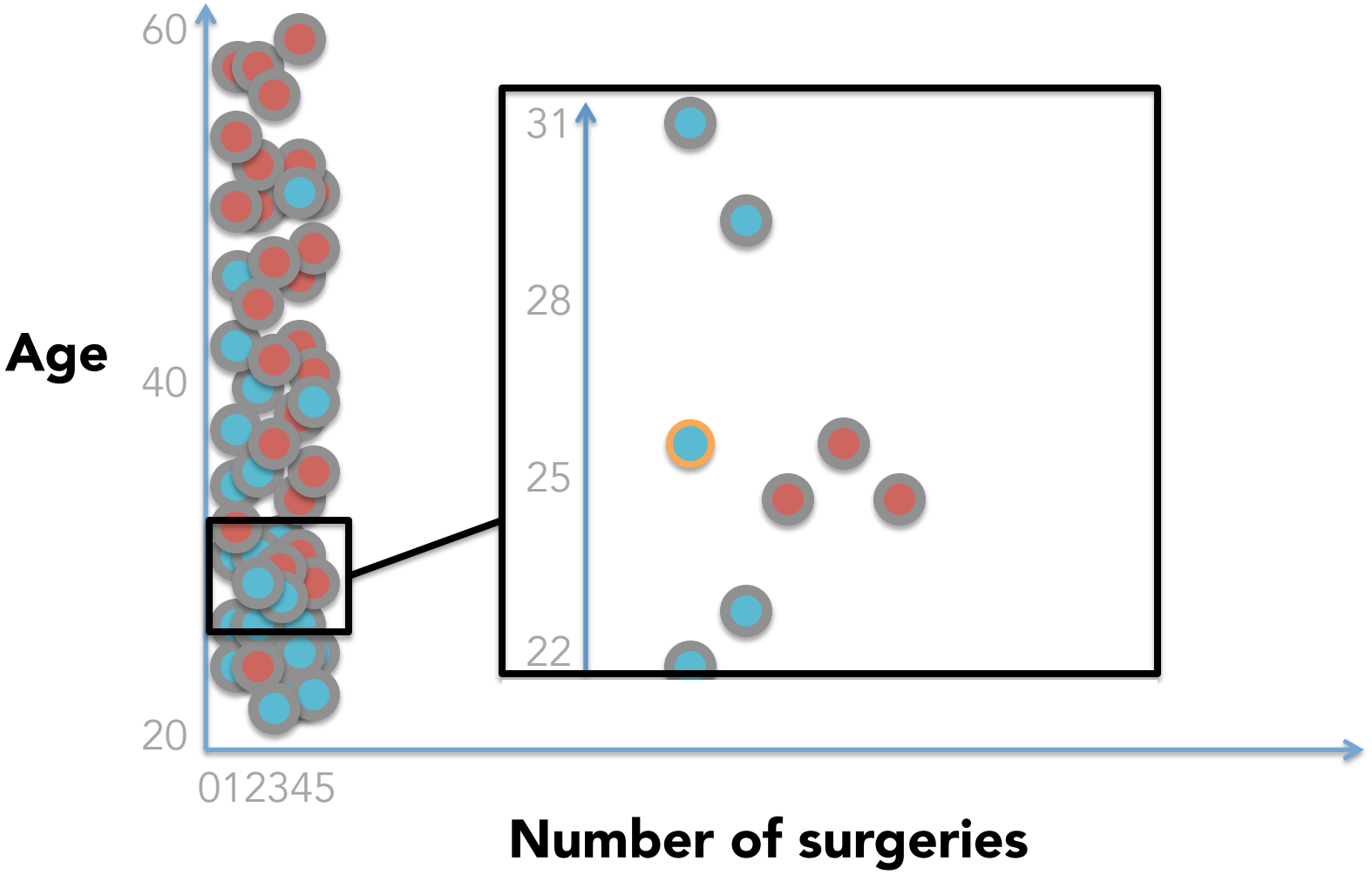
Scaling is crucial!



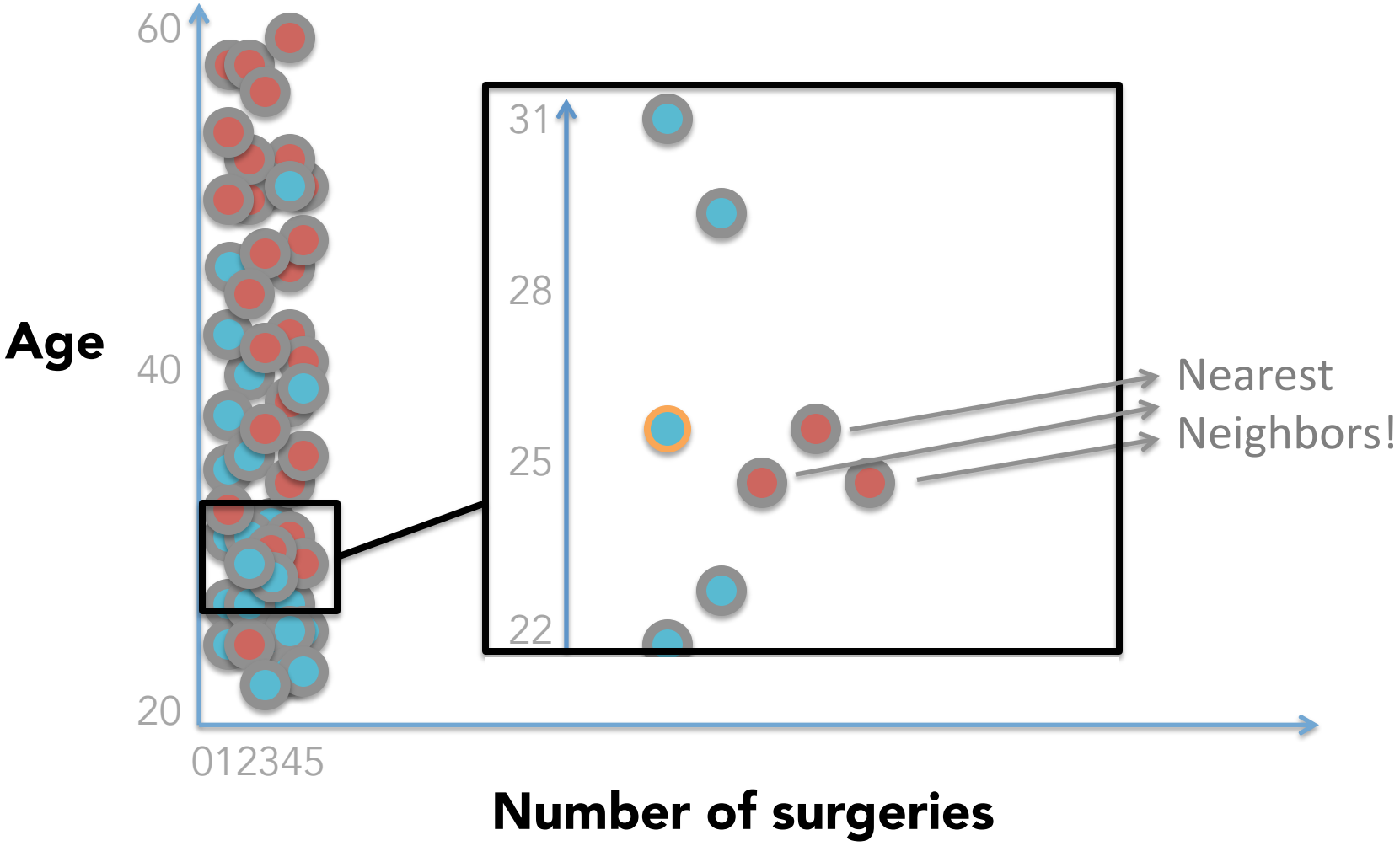
Scaling is crucial!



Scaling is crucial!



Scaling is crucial!



```
from sklearn.preprocessing import scale

X = scale( X )
```

K Nearest Neighbors

- Lazy
- “fits” fast, predicts slow
- Assuming that our data is d-dimensional, then the straight forward implementation is O(dn) time.
- Higher memory (saves entire training set)
- Various implementations, including weighted KNN

scikit.learn.

algorithms galore.