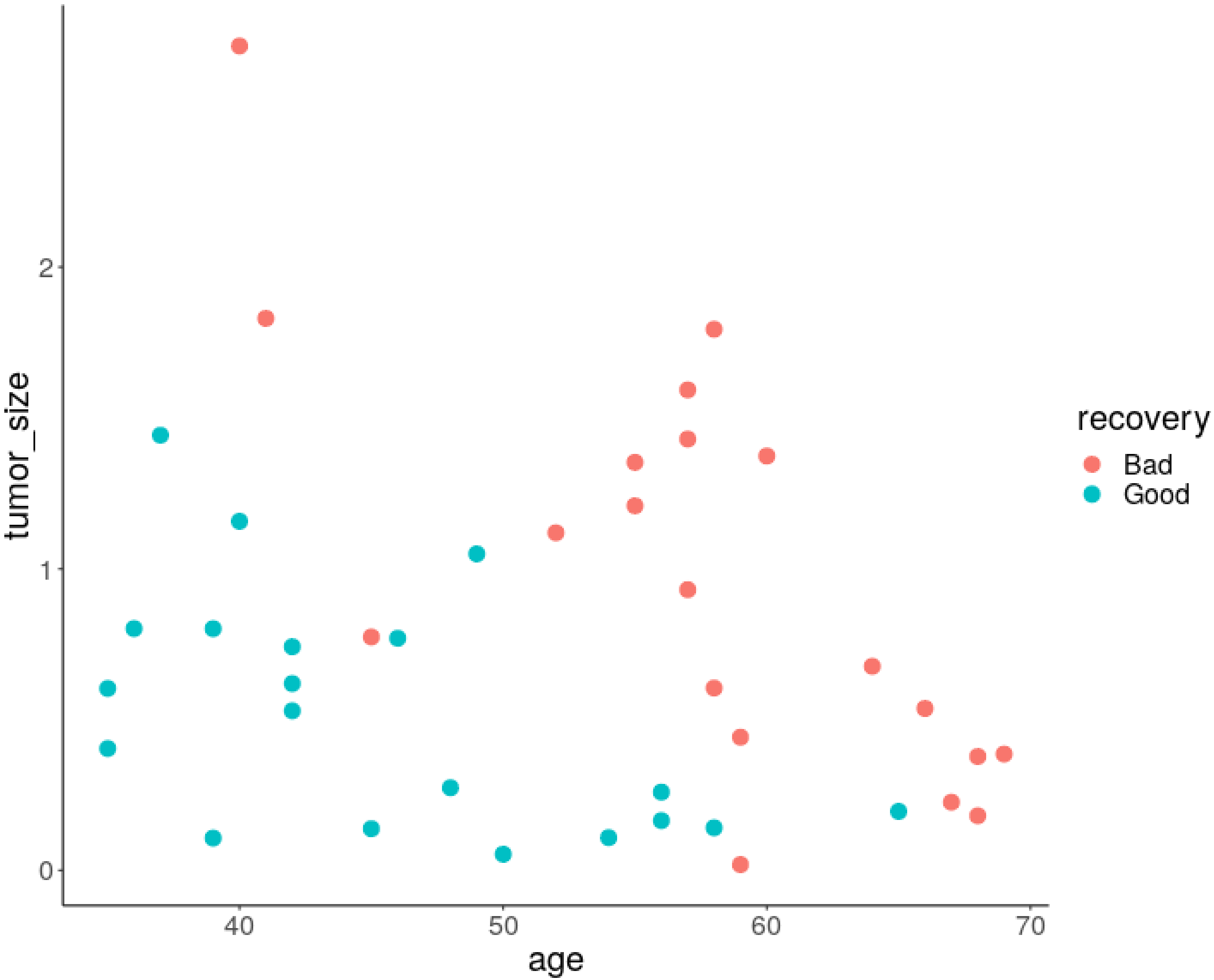
Statistical Learning

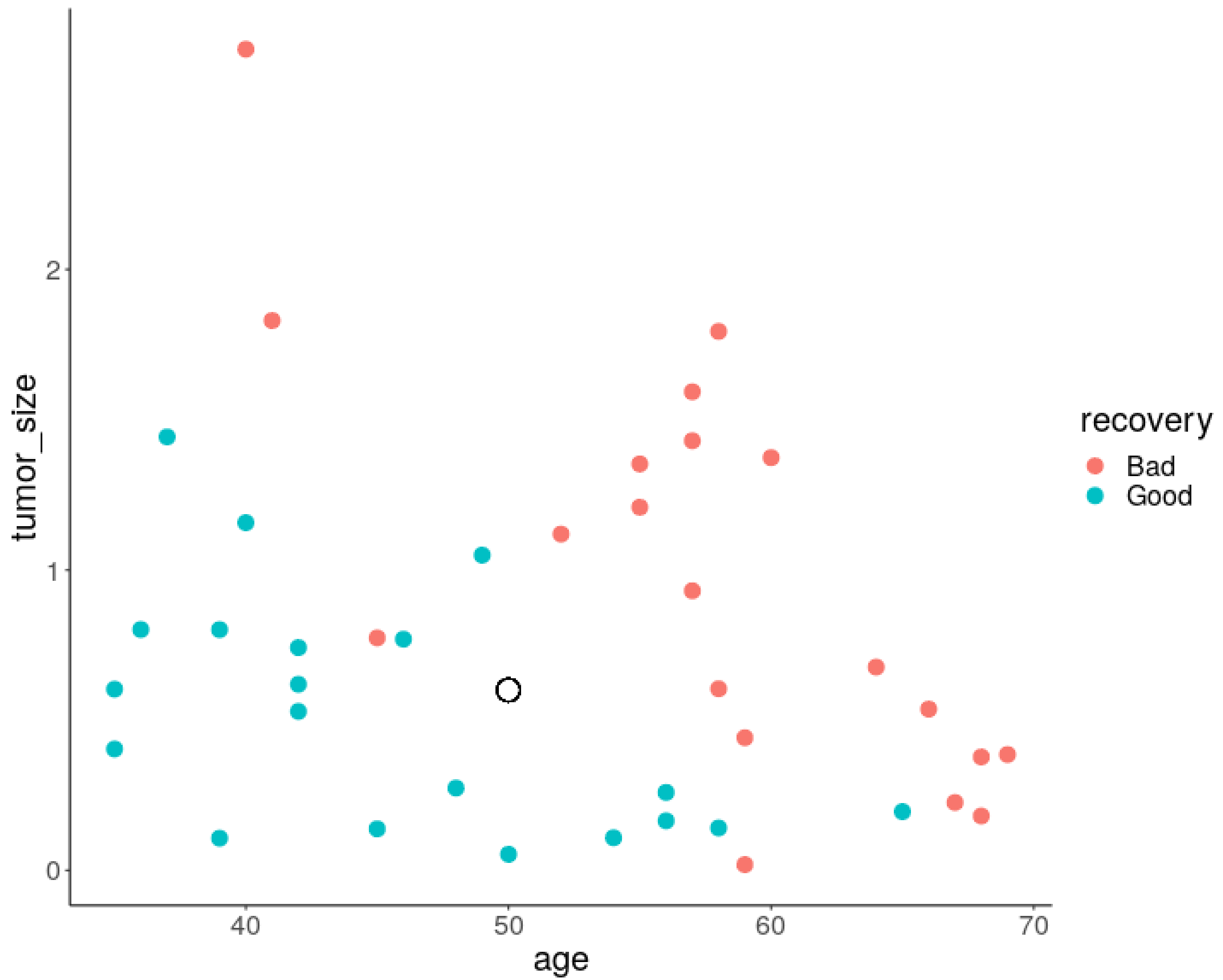
Juan C. Laria

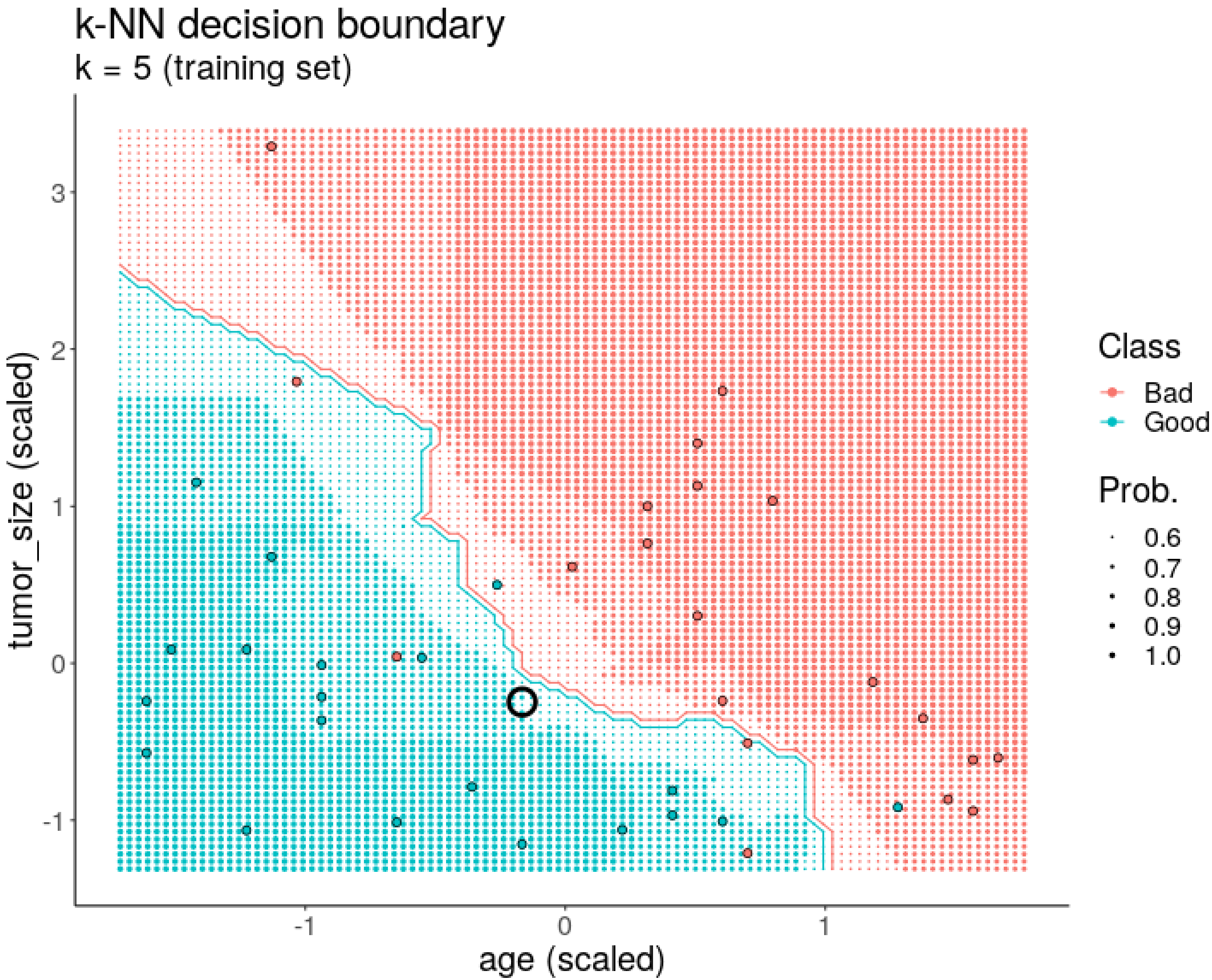
2018/11/14

... with R

Getting Started with k Nearest Neighbors


tumor_size
2
1
0
40
50
60
70
age
recovery
Bad
Good


tumor_size
age
recovery
Bad
Good
0
1
2
40
50
60
70


k-NN decision boundary
k = 5 (training set)
tumor_size (scaled)
age (scaled)
3
2
1
0
-1
-1
0
1
Class
Bad
Good
Prob.
0.6
0.7
0.8
0.9
1.0

How does it work?

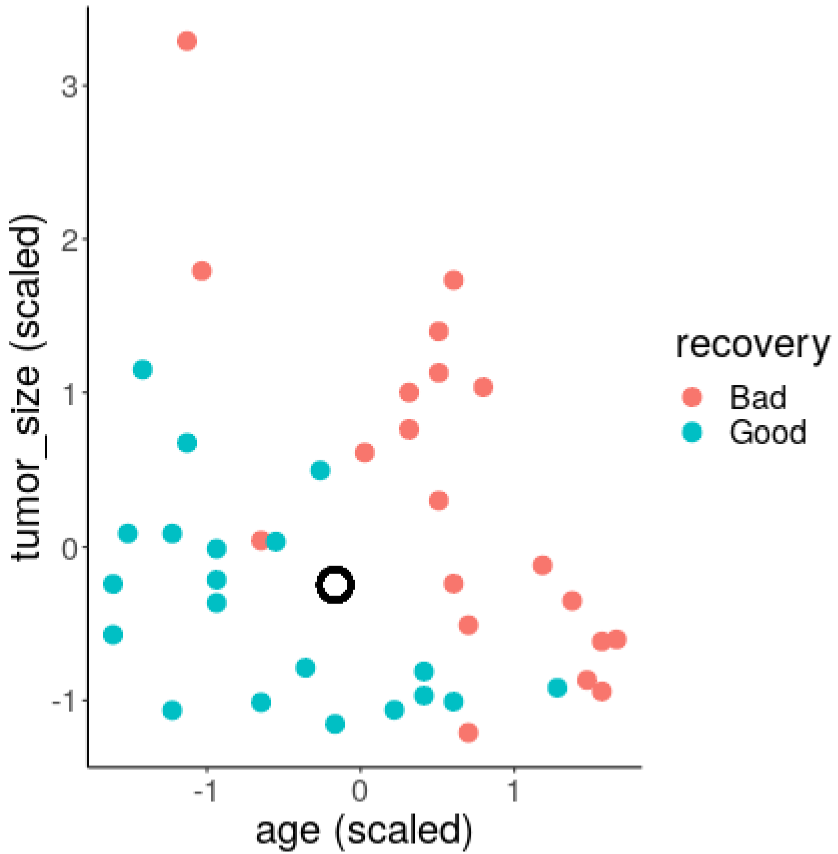
Euclidean distance

$$d(\mathbf{x}_1, \mathbf{x}_2) = ((\mathbf{x}_1 - \mathbf{x}_2)'(\mathbf{x}_1 - \mathbf{x}_2))^{1/2} = \sqrt{\sum_{j=1}^{p}(x_{1j} - x_{2j})^2}.$$

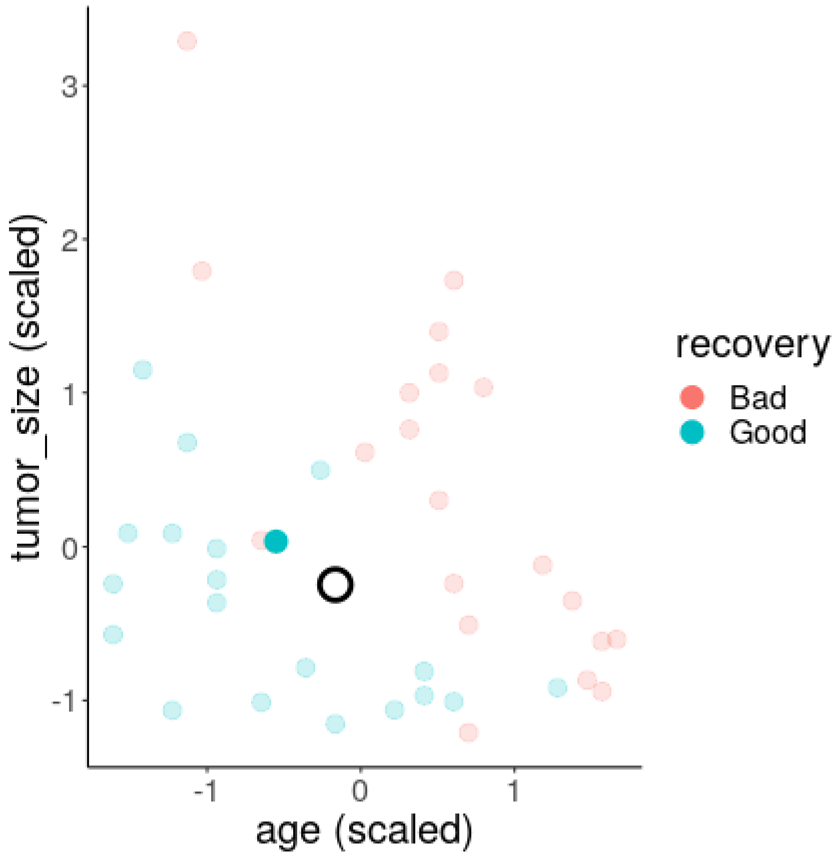
What does it do?

$$k = 5$$

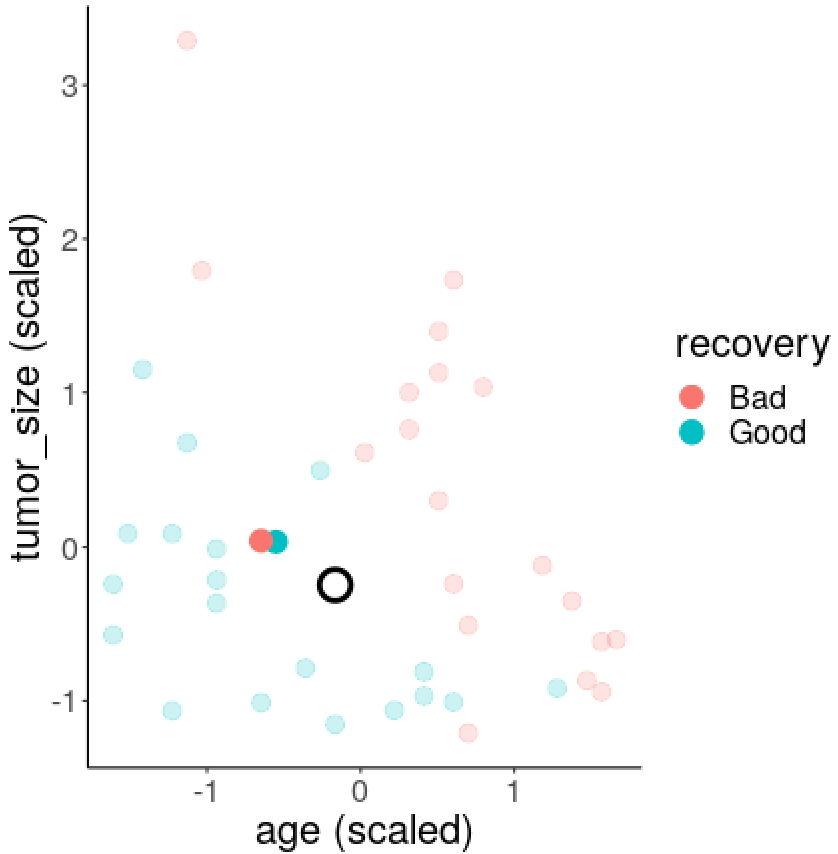
What does it do?



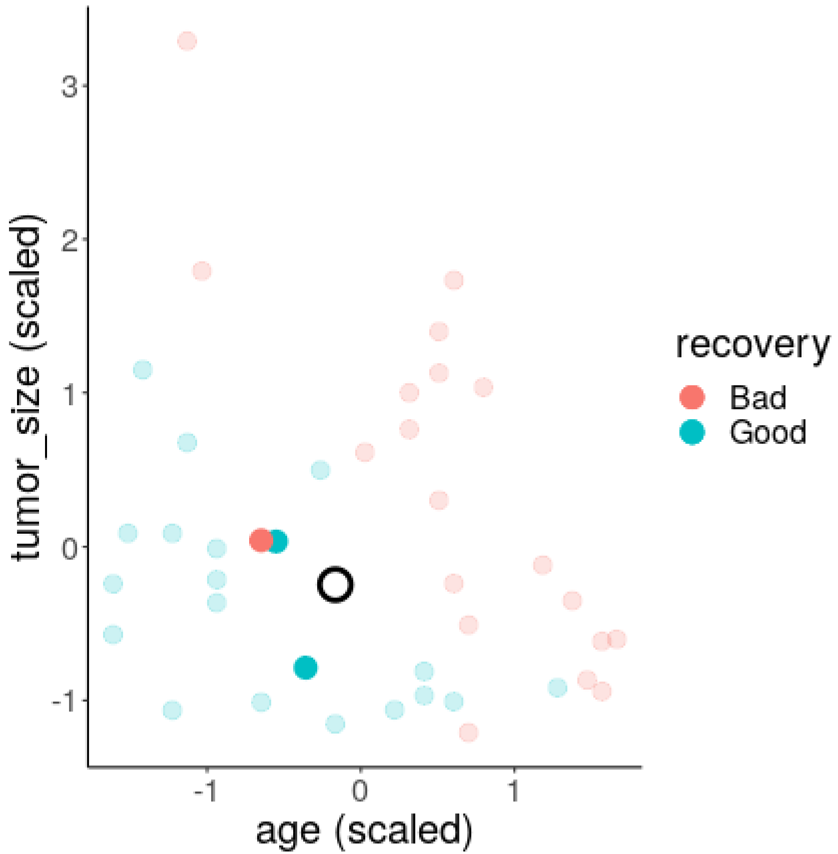
What does it do?



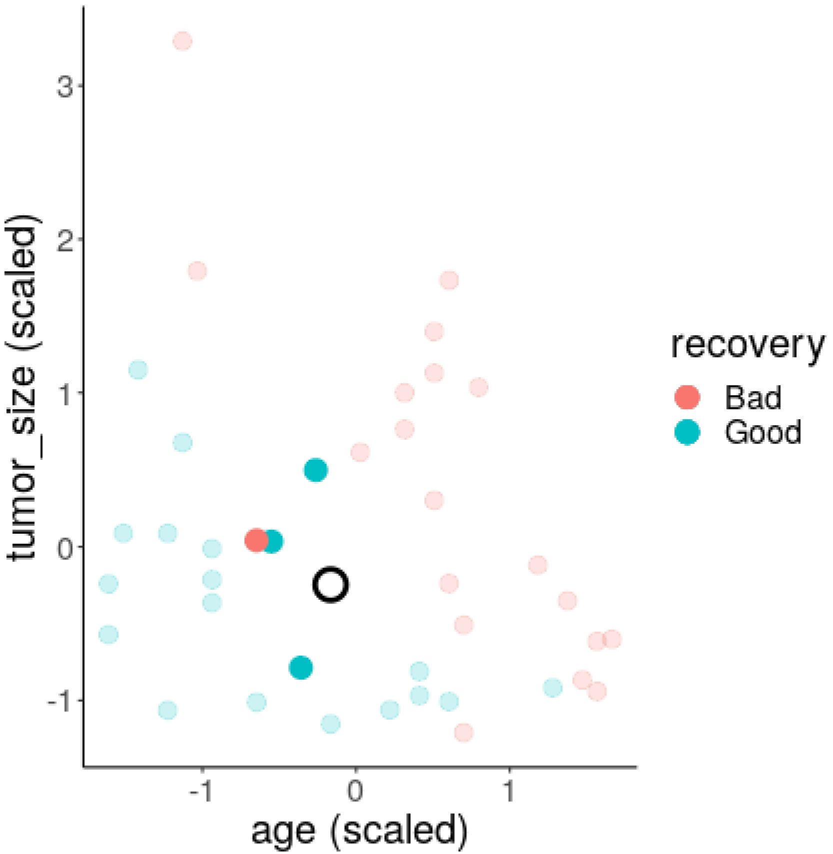
What does it do?



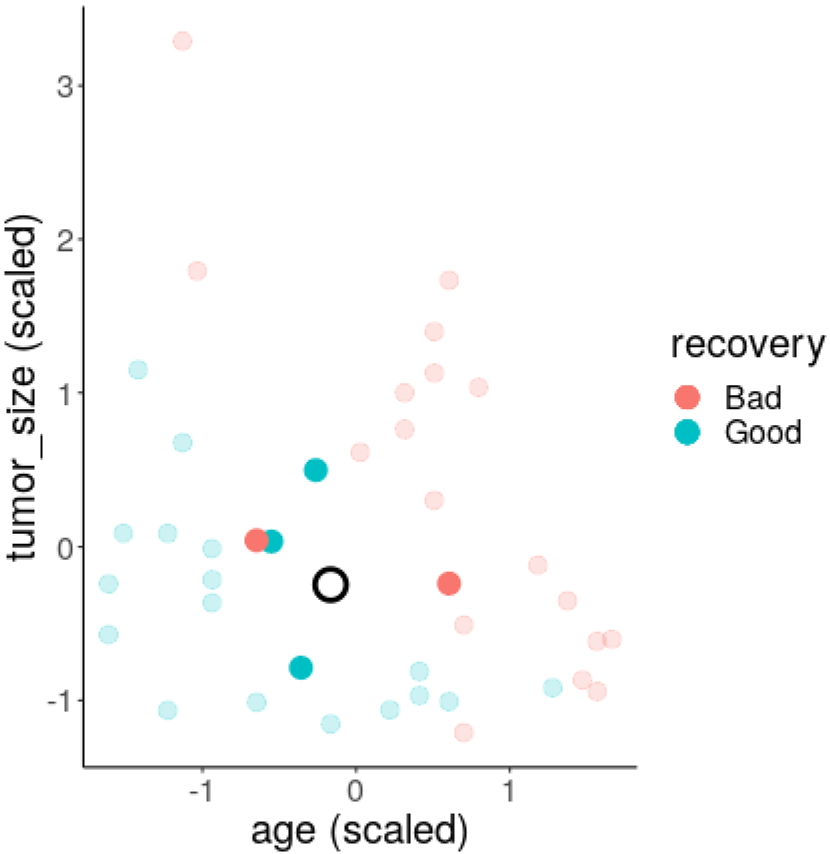
What does it do?



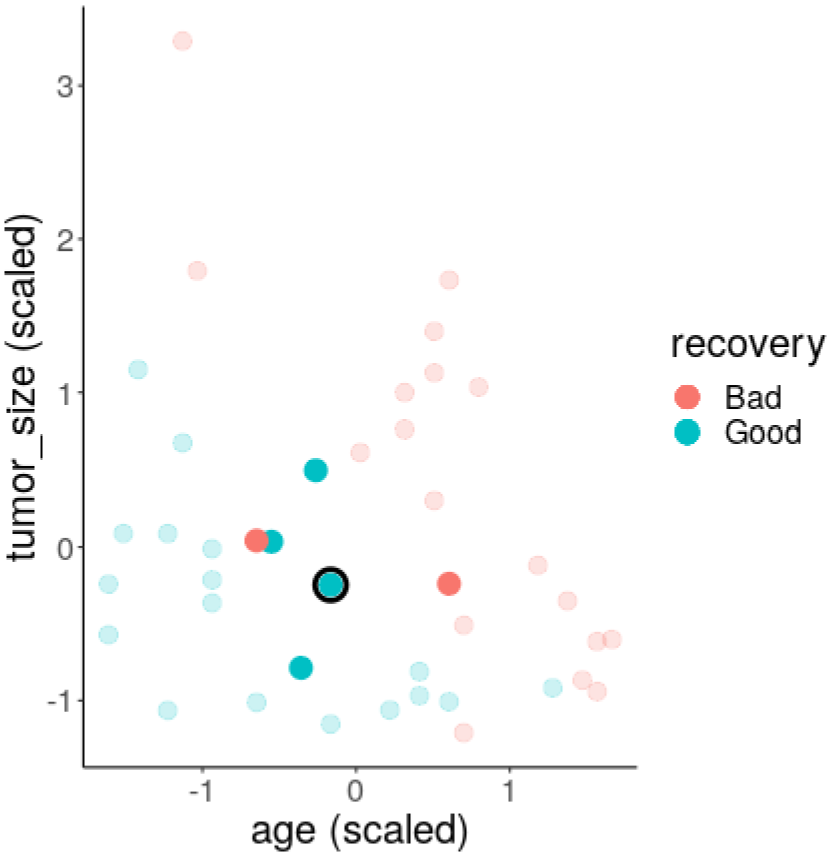
What does it do?



What does it do?



What does it do?



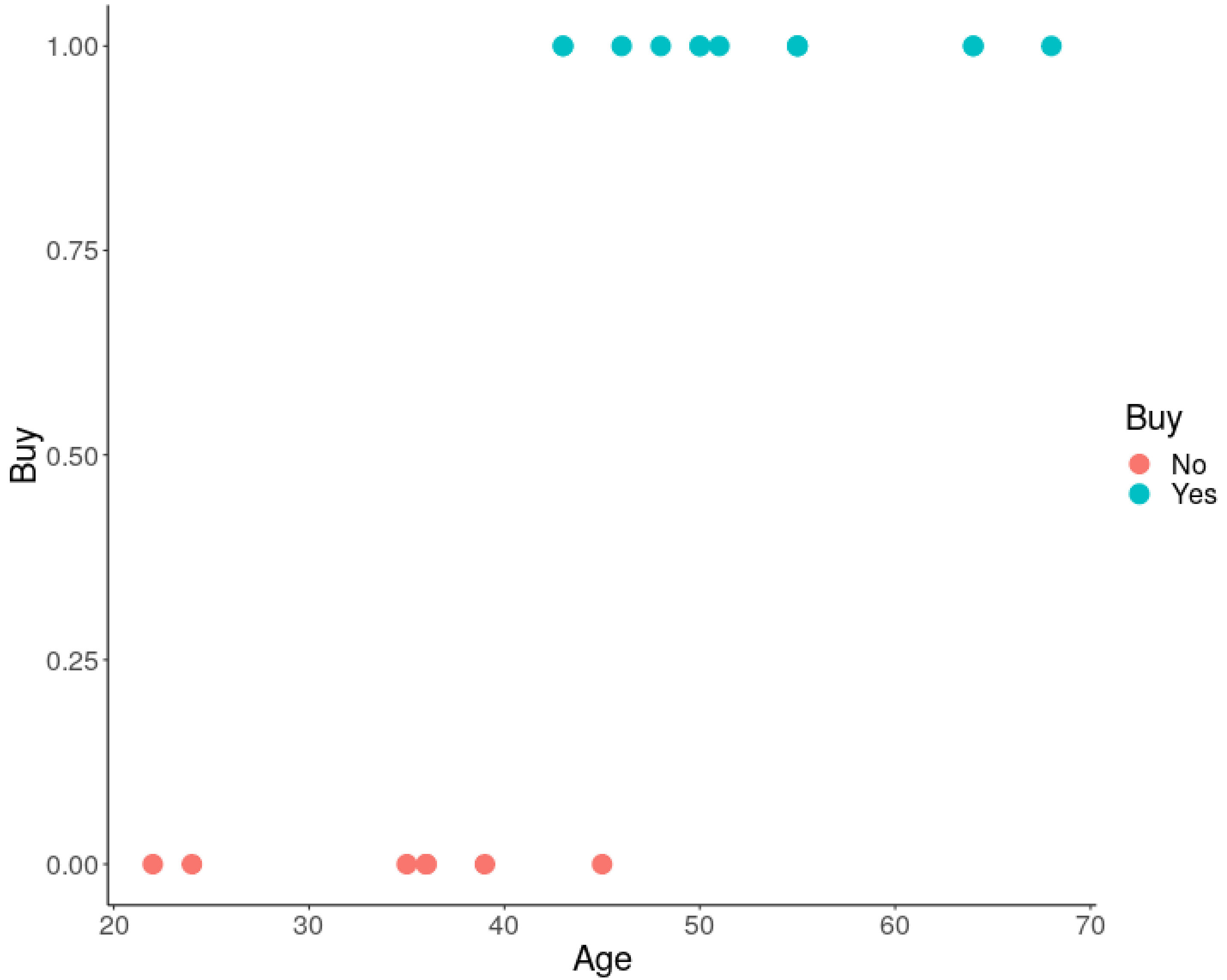
Practice time!

https://jlaria.github.io/SUsl/knn

https://raw.githubusercontent.com/jlaria/SUsl/master/source/knn_script.R

Lunch time!

Logistic Regression



1.00
0.75
0.50
0.25
0.00
Buy
20
30
40
50
60
70
Age
Buy
No
Yes

The intuition

We can think of the action of buying the product as a random variable defined as,

$$Y = \begin{cases} 0, & \text{Not buying} \\ 1, & \text{buying} \end{cases}$$

The intuition

We can think of the action of buying the product as a random variable defined as,

$$Y = \begin{cases} 0, & \text{Not buying} \\ 1, & \text{buying} \end{cases}$$

Conditioning Y on the value of X, we can model the following probabilities.

$$P(Y = 1|X = x) = p(x),$$

$$P(Y = 0|X = x) = 1 - p(x).$$

The intuition

We can think of the action of buying the product as a random variable defined as,

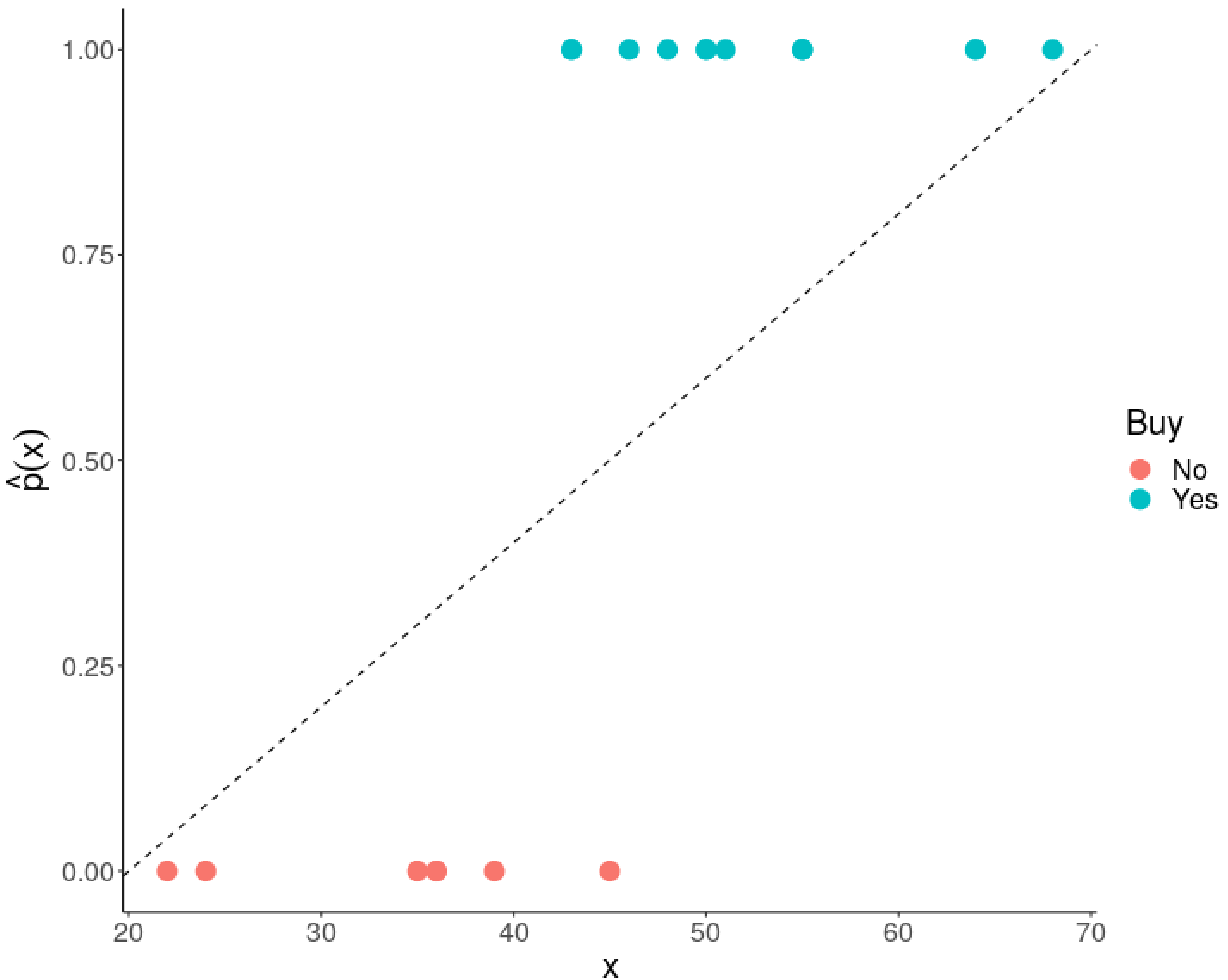
$$Y = \begin{cases} 0, & \text{Not buying} \\ 1, & \text{buying} \end{cases}$$

Conditioning Y on the value of X, we can model the following probabilities.

$$P(Y = 1|X = x) = p(x),$$

$$P(Y = 0|X = x) = 1 - p(x).$$

This means that $Y|X = x$ follows a Bernoulli distribution with probability of success $p(x)$. Instead of modelling the response Y, we model $p(x)$.



1.00
0.75
0.50
0.25
0.00
$\hat{p}(x)$
20
30
40
50
60
70
x
Buy
No
Yes

The intuition

- If we wanted to work with linear functions, say $\beta_0 + \beta X$, it shouldn't be $p(x)$ the function to be approximated. Notice that,
  - $0 \leq p \leq 1$

The intuition

- If we wanted to work with linear functions, say $\beta_0 + \beta X$, it shouldn't be $p(x)$ the function to be approximated. Notice that,
  - $0 \leq p \leq 1$
- Then, the *odds*, defined as $p/(1-p)$ are such that,
  - $0 \leq p/(1-p) \leq +\infty$

The intuition

- If we wanted to work with linear functions, say $\beta_0 + \beta X$, it shouldn't be $p(x)$ the function to be approximated. Notice that,
  - $0 \leq p \leq 1$
- Then, the *odds*, defined as $p/(1-p)$ are such that,
  - $0 \leq p/(1-p) \leq +\infty$
- The odds can be approximated better by a linear function, but notice that, for $p = 0.5$, $p/(1-p) = 1$. This means that for x such that $p(x) < 0.5$, the odds are between 0 and 1, and for x such that $p(x) > 0.5$, the odds are between 1 and ∞.

The intuition

- If we wanted to work with linear functions, say $\beta_0 + \beta X$, it shouldn't be $p(x)$ the function to be approximated. Notice that,
  - $0 \leq p \leq 1$
- Then, the *odds*, defined as $p/(1-p)$ are such that,
  - $0 \leq p/(1-p) \leq +\infty$
- The odds can be approximated better by a linear function, but notice that, for $p = 0.5$, $p/(1-p) = 1$. This means that for x such that $p(x) < 0.5$, the odds are between 0 and 1, and for x such that $p(x) > 0.5$, the odds are between 1 and ∞.
- To tackle this unbalance, we can take logarithms.
  - $-\infty \leq \log(p/(1-p)) \leq +\infty$

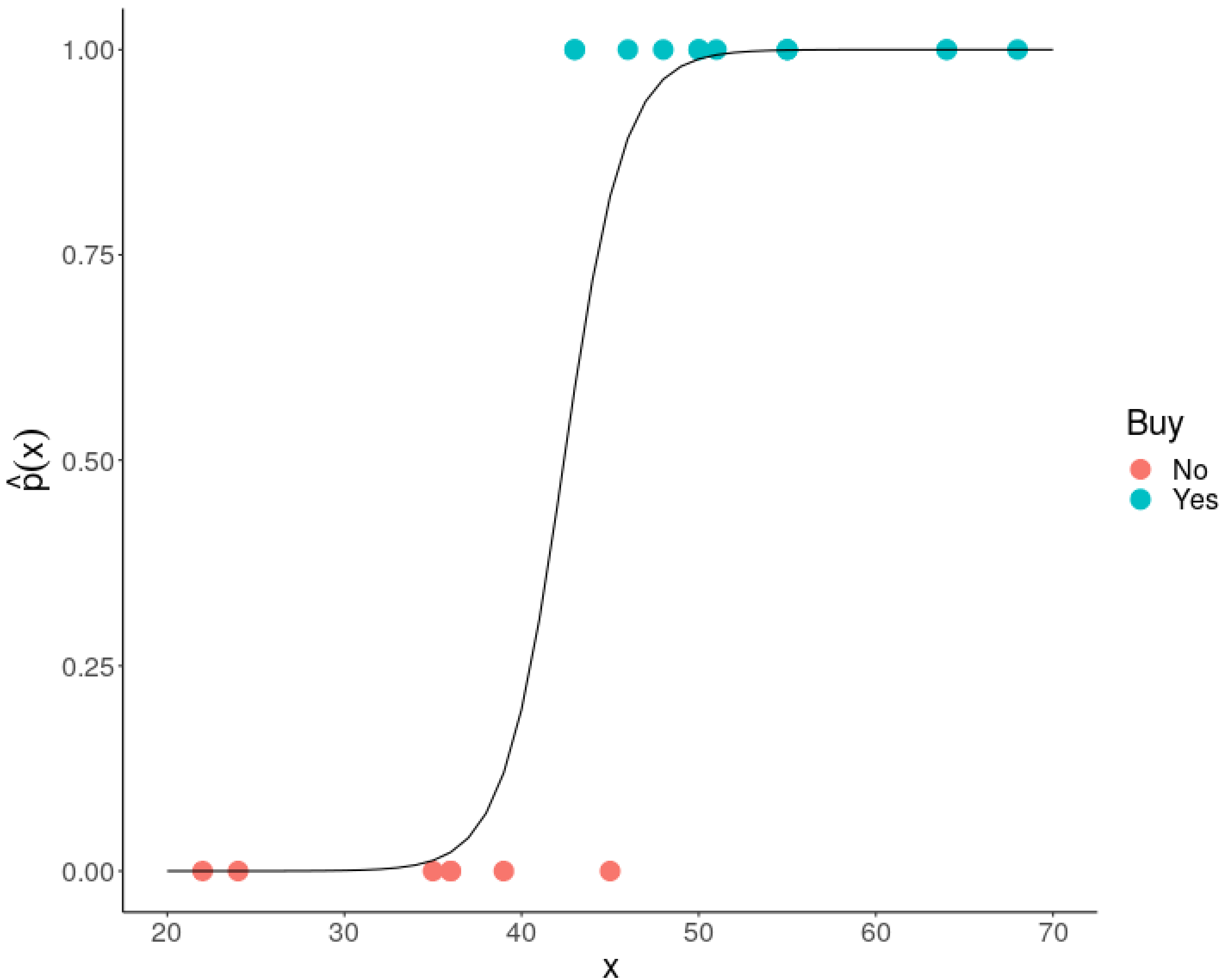
The intuition

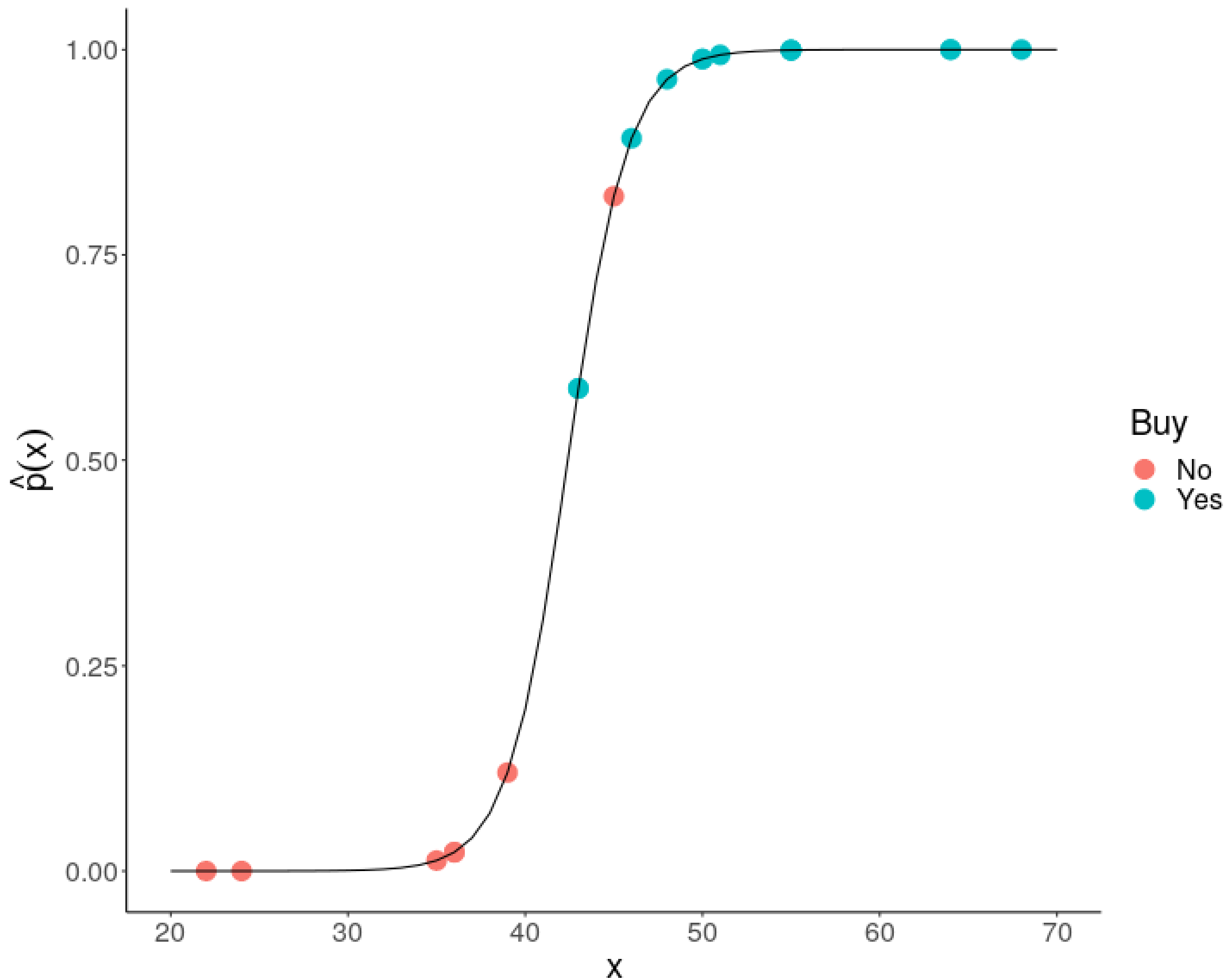
- The *log-odds* may be the perfect candidates for a linear approximation.

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta X$$

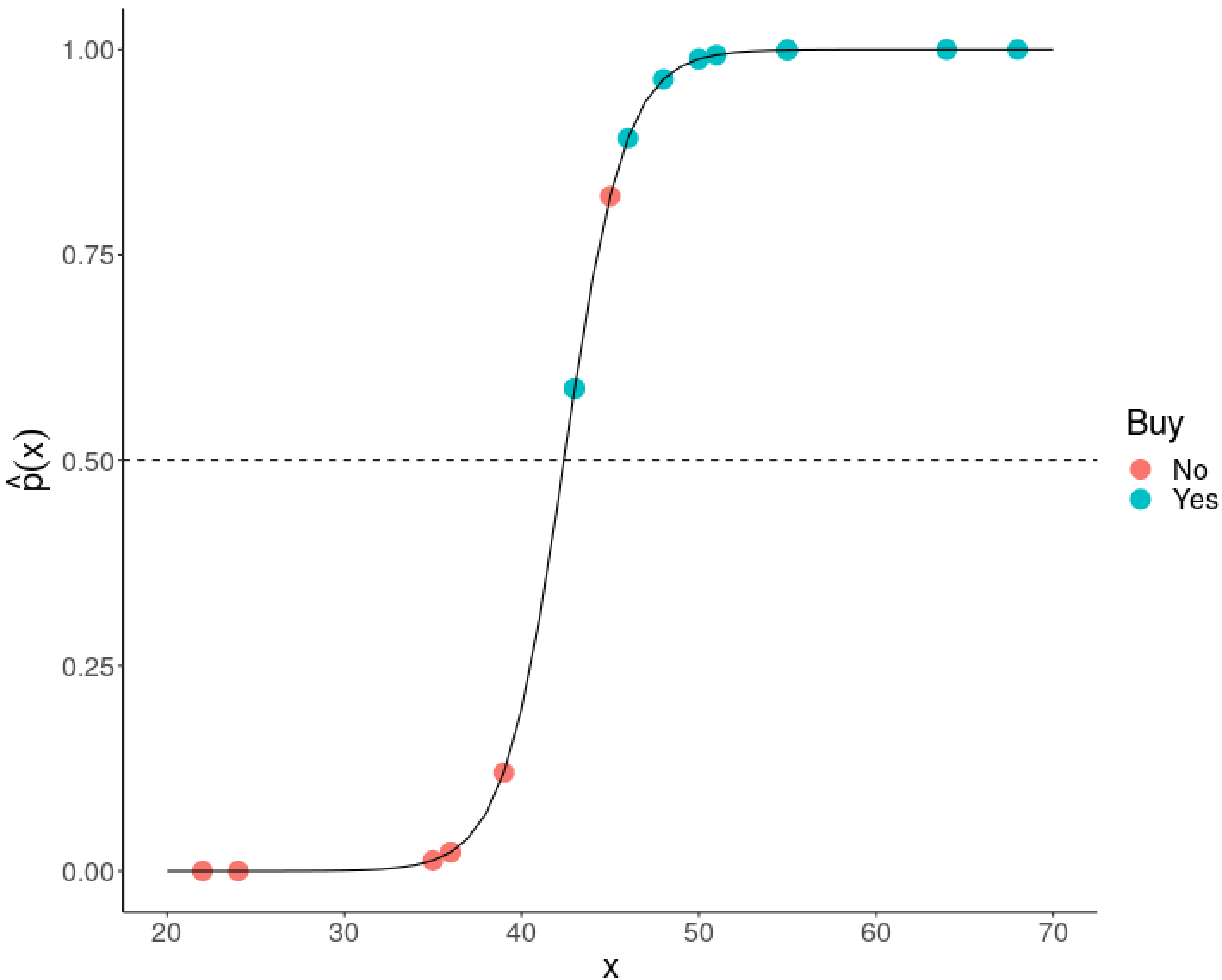
- Taking p out,

$$p = \left[1 + \exp(-\beta_0 - \beta X)\right]^{-1}$$


1.00
0.75
0.50
0.25
0.00
$\hat{p}(x)$
20
30
40
50
60
70
x
Buy
No
Yes



1.00
0.75
0.50
0.25
0.00
p̂(x)
20
30
40
50
60
70
x
Buy
No
Yes


1.00
0.75
0.50
0.25
0.00
$\hat{p}(x)$
20
30
40
50
60
70
x
Buy
No
Yes

How does it work?

Let $y_1, y_2, \ldots y_N$ be the responses, and $p_1, p_2, \ldots p_N$ the underlying probabilities that generated them from the Bernoulli model. Then, the likelihood is,

$$L(y_1 \ldots y_N | p_1 \ldots p_N) = \prod_{i=1}^{N} p_i^{y_i} (1 - p_i)^{(1-y_i)} = \prod_{i=1}^{N} \left(\frac{p_i}{1 - p_i} \right)^{y_i} (1 - p_i)$$

Then, the log-likelihood becomes,

$$l(y_1 \ldots y_N | p_1 \ldots p_N) = \sum_{i=1}^{N} \left\{ y_i \log\left(\frac{p_i}{1 - p_i} \right) + \log(1 - p_i) \right\}$$

Let $\eta_i = \mathbf{x}_i^T \beta$, and notice that the odds can be written in terms of η_i as

$$\frac{p_i}{1 - p_i} = \exp(\eta_i)$$

How does it work?

Then, the log-likelihood can be written

$$l(y_1 \dots y_N | p_1 \dots p_N) = \sum_{i=1}^{N} \{ y_i \eta_i + \log([1 + \exp(\eta_i)]^{-1}) \}$$

$$l(y_1 \dots y_N | p_1 \dots p_N) = \sum_{i=1}^{N} \{ y_i \eta_i - \log(1 + e^{\eta_i}) \}$$

Therefore, the objective of logistic regression is to minimize with respect to β the function

$$R(\beta) = \sum_{i=1}^{N} \log\big(1 + \exp(\mathbf{x}_i^T \beta)\big) - \sum_{i=1}^{N} y_i \mathbf{x}_i^T \beta$$

Practice time!

https://jlaria.github.io/SUsl/logit

https://raw.githubusercontent.com/jlaria/SUsl/master/source/logit_script.R

Unsupervised statistical learning

k-means clustering

Hierarchical clustering

The intuition

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

The ~~magic~~ math behind k-means

Within-cluster scatter

- Let K be the number of clusters (fixed). A clustering of points $x_1, x_2 \dots x_n$ is a function C that assigns each observation x_i to a group $k \in \{1 \dots K\}$.

> Notation

- $C(i) = k$ means that x_i is assigned to group k.
- n_k is the number of points in group k.

> Definition

- The within-cluster scatter is defined as

$$W = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{C(i)=k, \\ C(i')=k}} D(x_i, x_{i'}).$$

Finding the best group assignments

- Smaller W the better assignments.
- Why don't we just find the clustering C that minimizes W?

Finding the best group assignments

- Smaller W the better assignments.
- Why don't we just find the clustering C that minimizes W?
- Trying all possible assignments of n points into K groups requires a number of operations of order

$$A(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^n \approx K^n.$$

- Notice that $A(10, 4) = 34105$ and $A(25, 4) \approx 5 \cdot 10^{13}$.
- BigData problems are clearly bigger than $n = 25, K = 4$.
- We will have to look for an approximate optimal solution.

k means

K-means algorithm is intended for situations in which,

- all variables are of *quantitative* type,
- **squared Euclidean** distance

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\|_2^2$$

is chosen as the dissimilarity measure.

K-means minimization problem

- In K-means algorithm, we want to minimize over clusterings C the within-cluster scatter

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{C(i)=k,\\ C(i')=k}} \|x_i - x_{i'}\|_2^2.$$

- It is equivalent to minimizing over C the within-cluster variation

$$W = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2,$$

where

$$\bar{x}_k = \frac{1}{n_k} \sum_{C(i)=k} x_i.$$

Rewriting the minimization

- We want to choose C to minimize,

$$\sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|_2^2.$$

- For any $z_1, \ldots z_m \in R^p$, the quantity $\sum_{i=1}^{m} \|z_i - c\|_2^2$ is minimized by taking $c = \bar{z}$.

- So our problem is the same as minimizing the enlarged criterion

$$\sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - c_k\|_2^2,$$

  over both clusterings C and $c_1, \ldots c_K \in R^p$.

- The K-means clustering algorithm approximately minimizes the enlarged criterion by alternately minimizing over C and $c_1, \ldots c_K$.

Practice time!

https://jlaria.github.io/SUsl/kmeans

https://jlaria.github.io/SUsl/hclust

k-means properties

- Within-cluster variation *decreases* with each iteration of the algorithm.
- It always *converges*, no matter the initial cluster centers (it takes less than K^n iterations)
- The solution depends on the initial (random) cluster assignments.
- K-means algorithm finds a *local* rather than a global optimum. For this reason, it is important to run the algorithm multiple times from different initial configurations. Then, one selects the *best* solution (in terms of within-cluster variation).

Thanks!

Slides created via the R package **xaringan**.

remark.js, **knitr**, and R Markdown.