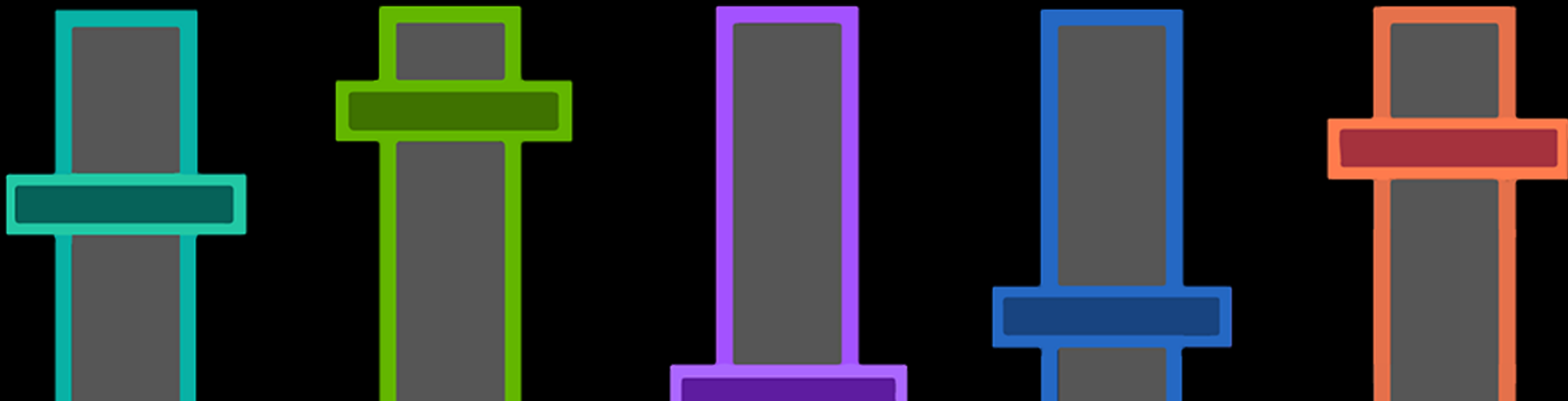
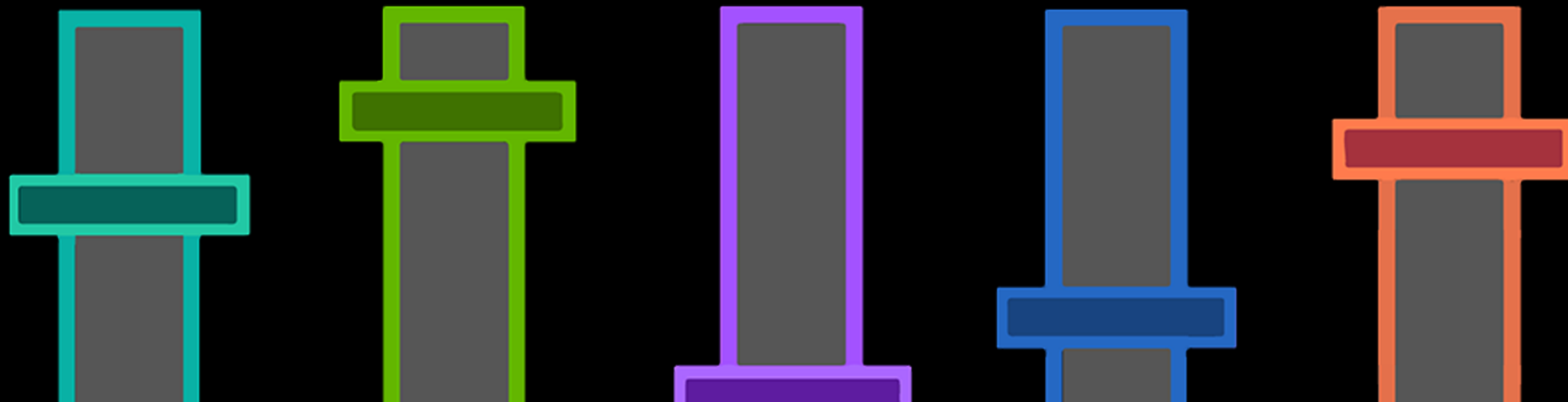


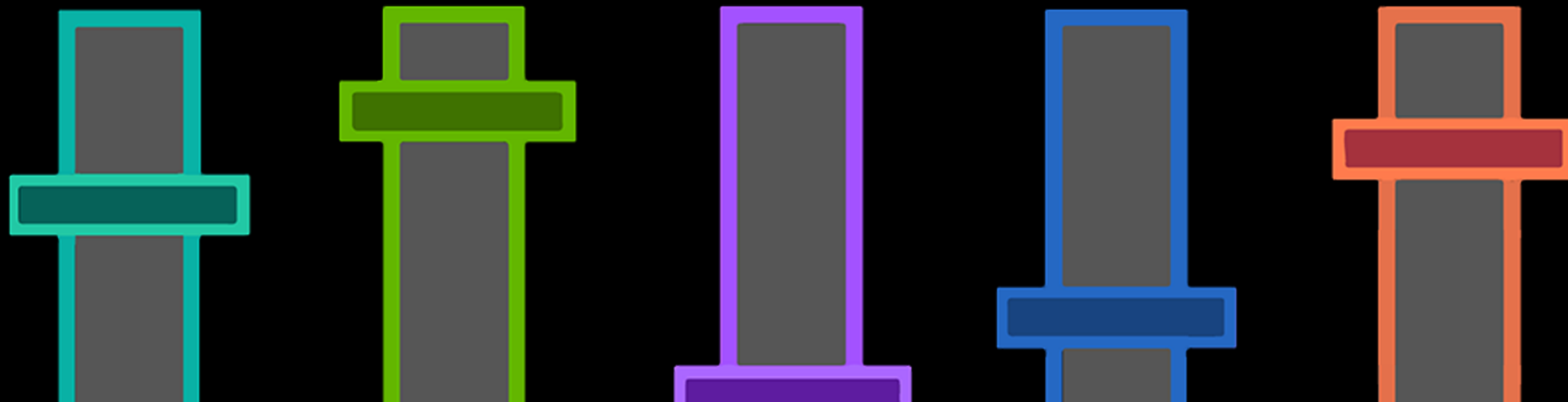
TUNE



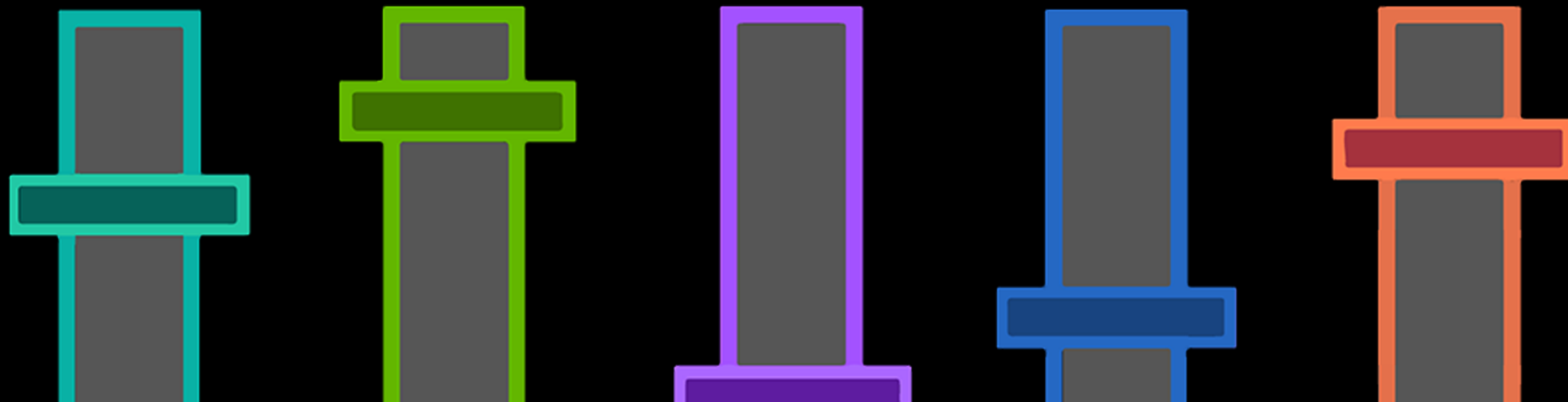
TUNING *HYPERPARAMETERS,* *IN 5 MINUTES*



~~Hyperparameter tuning~~ failing to tune *HYPERPARAMETERS, IN 5 MINUTES*

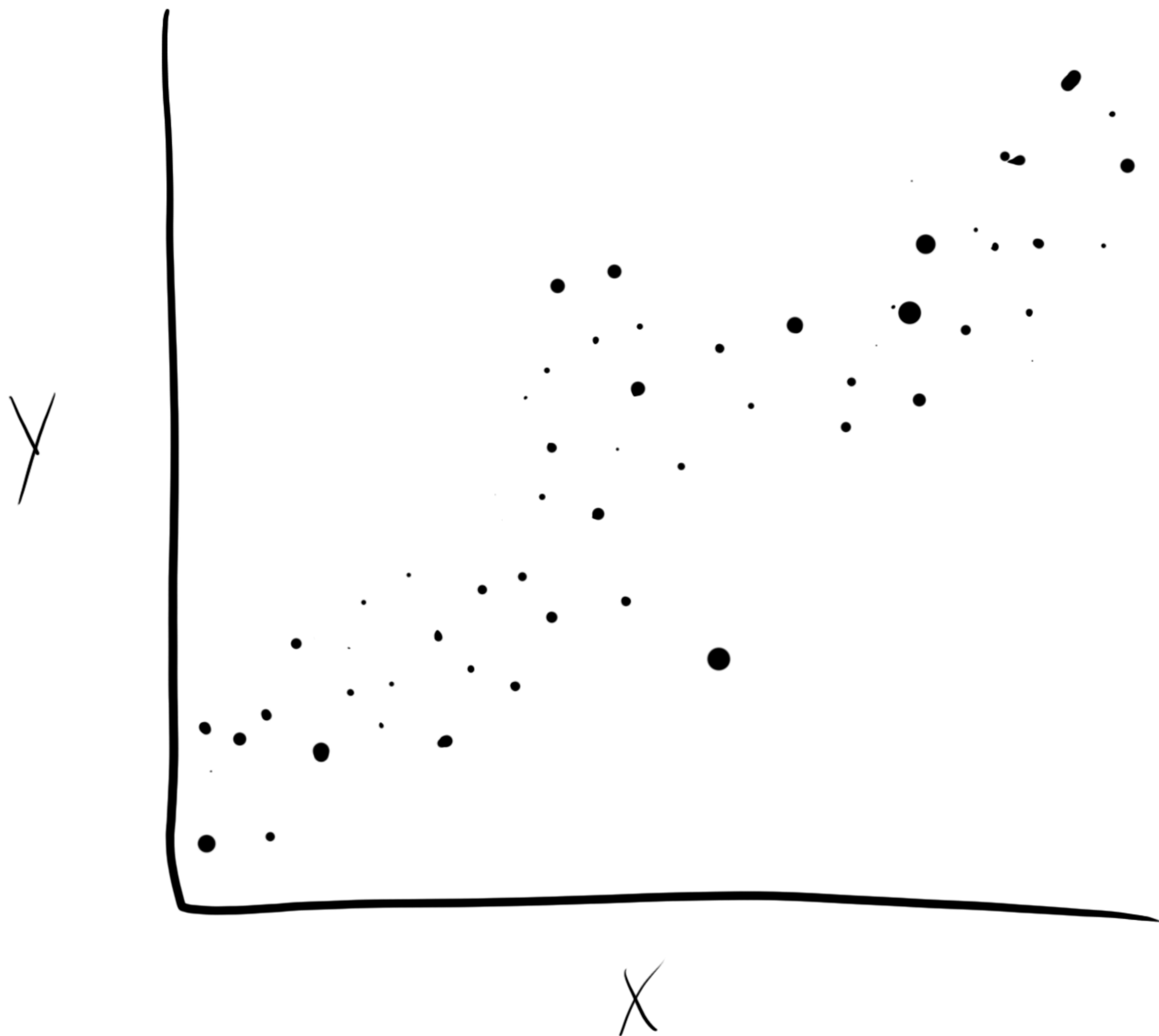


TUNING *HYPERPARAMETERS,* *IN 5 MINUTES*



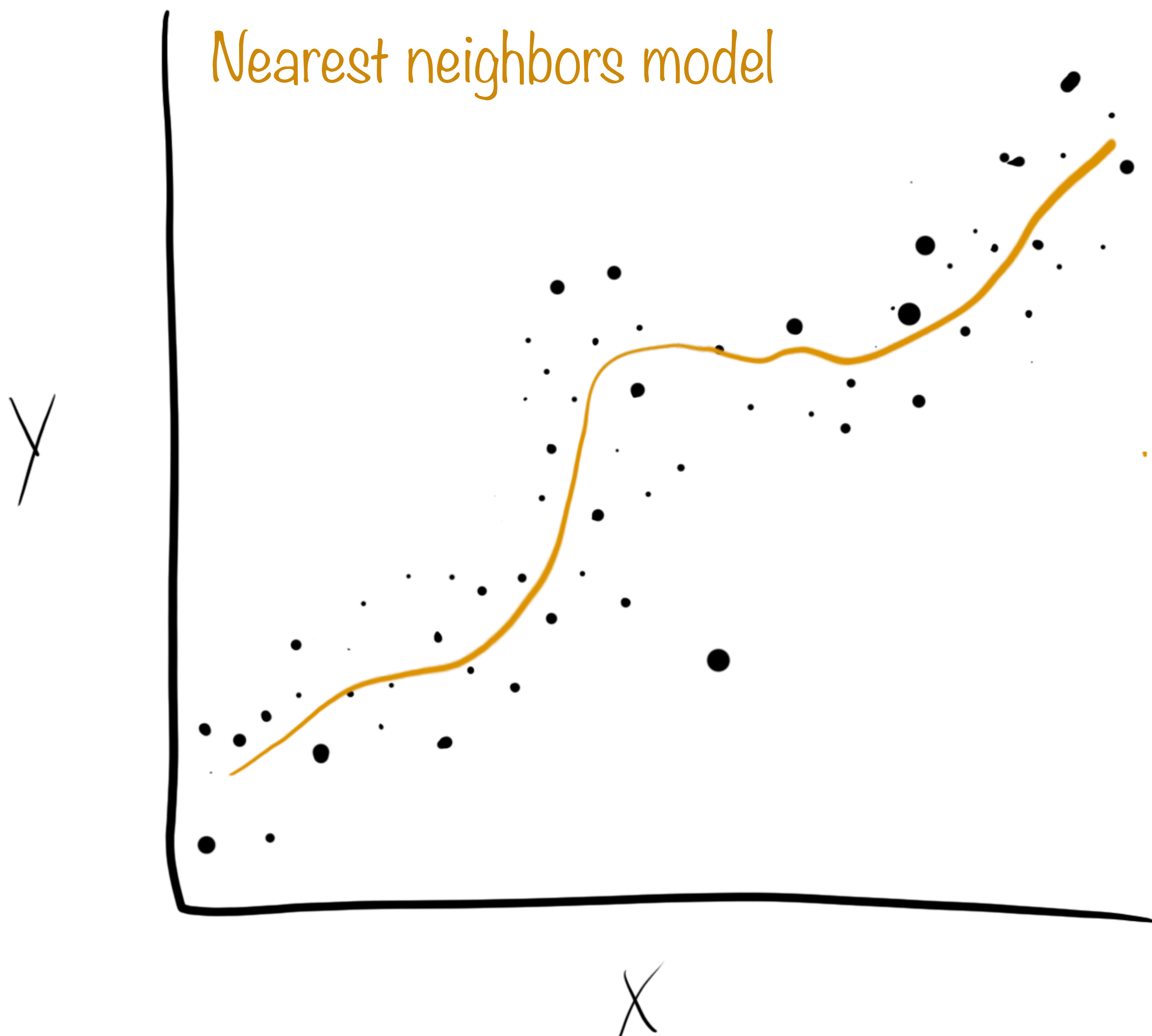


simonpcouch/oss_demo_day_tune



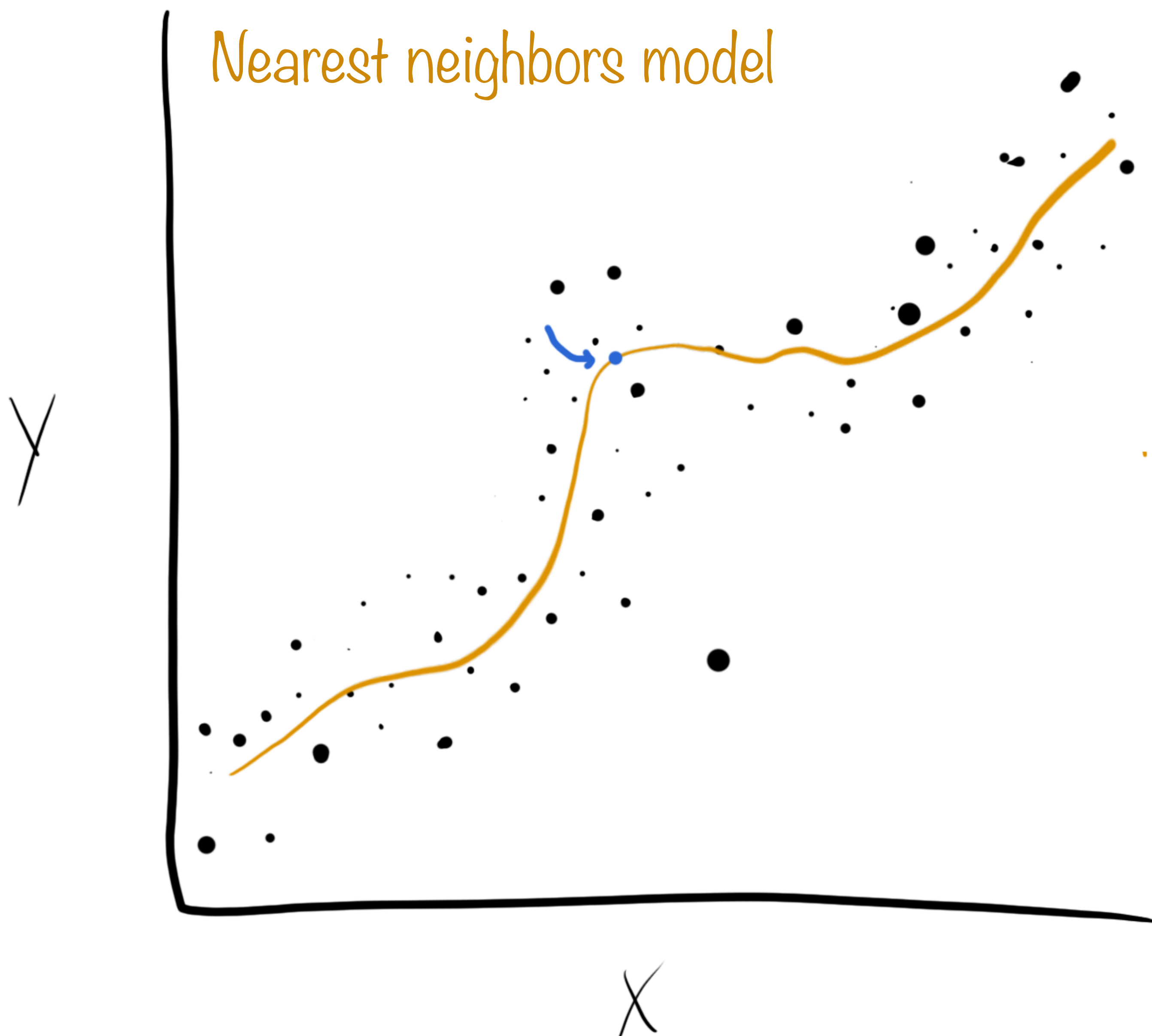


Nearest neighbors model





Nearest neighbors model



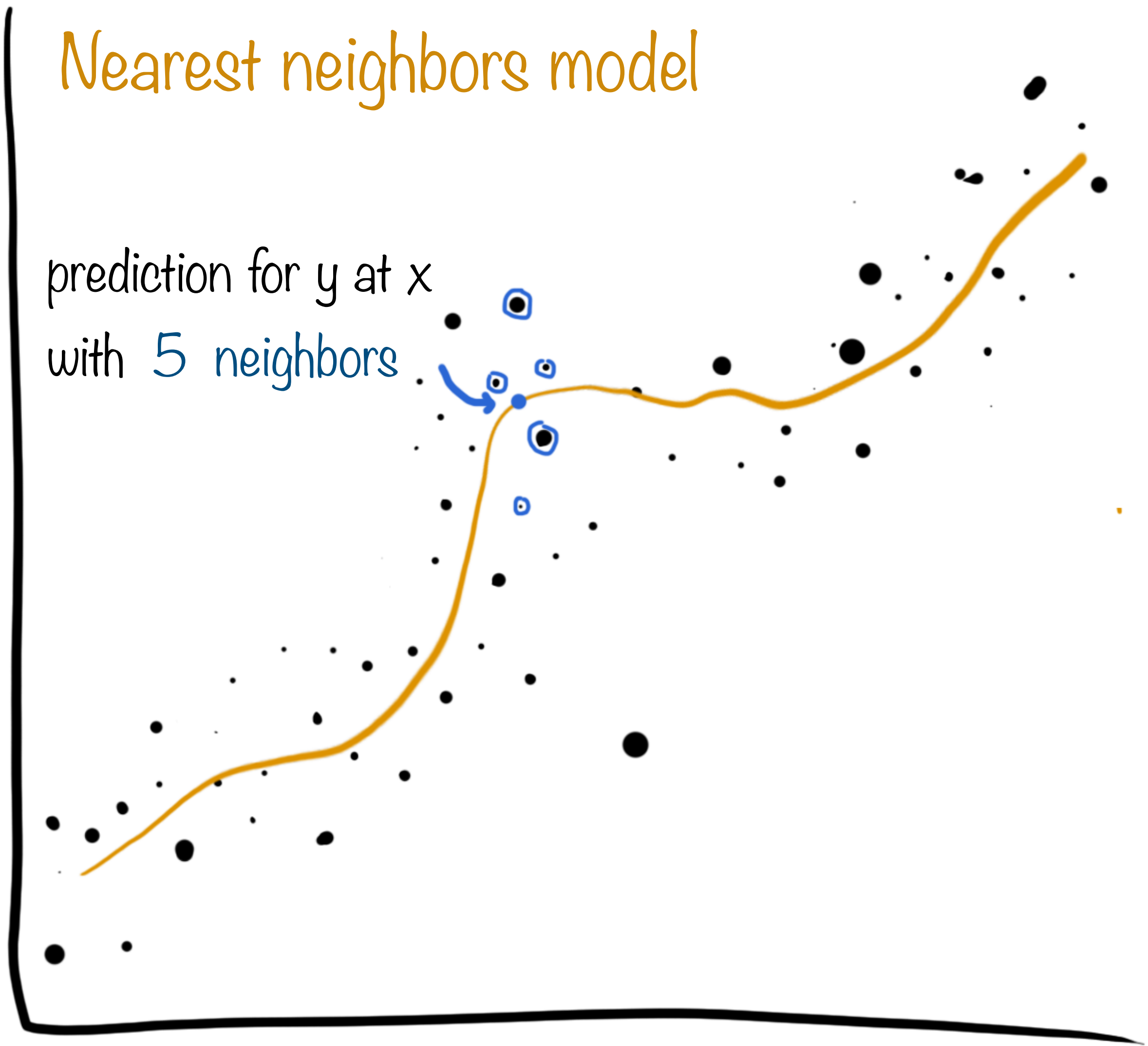


Nearest neighbors model

prediction for y at x
with 5 neighbors

Y

X



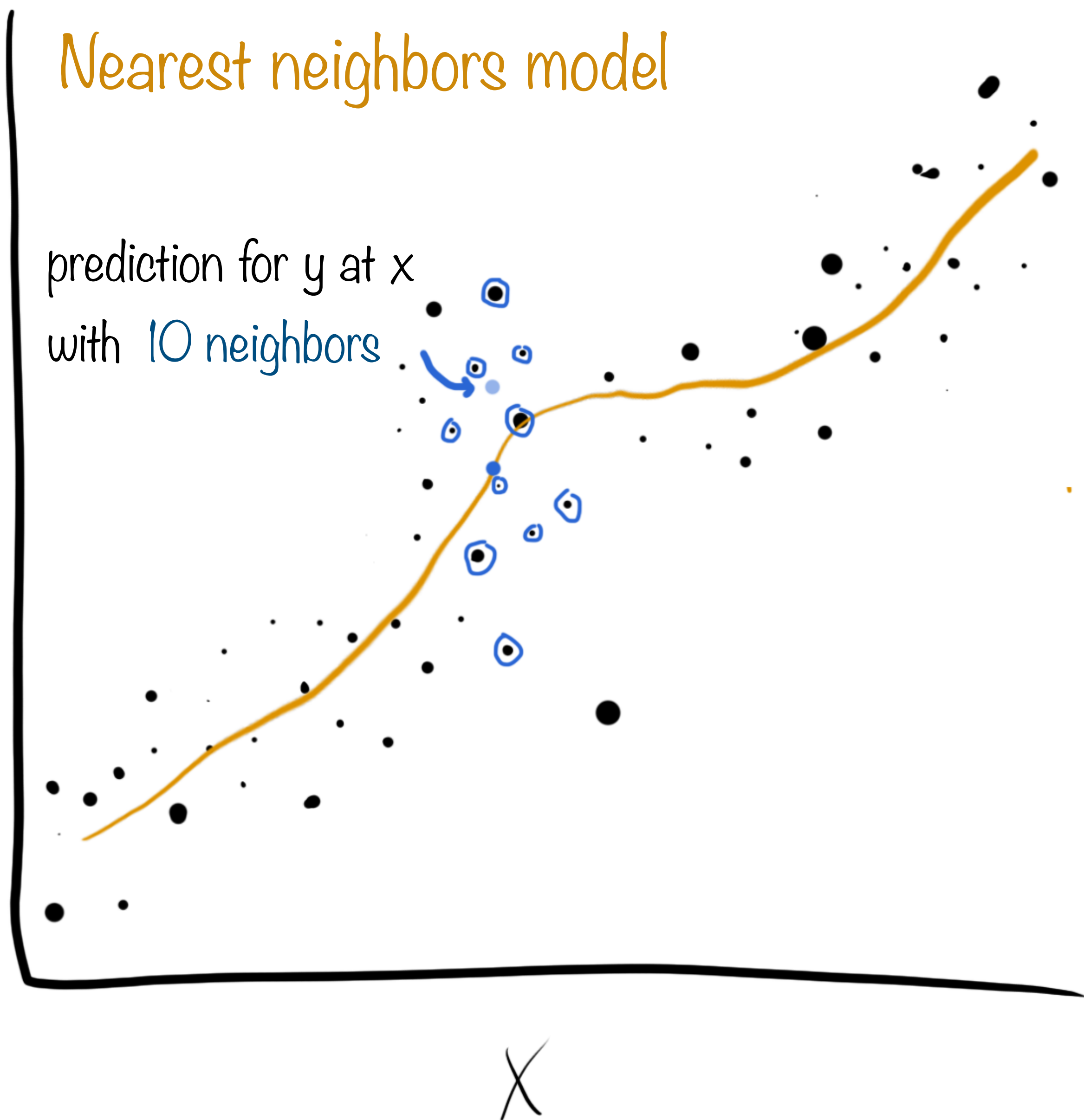


Nearest neighbors model

prediction for y at x
with 10 neighbors

Y

X

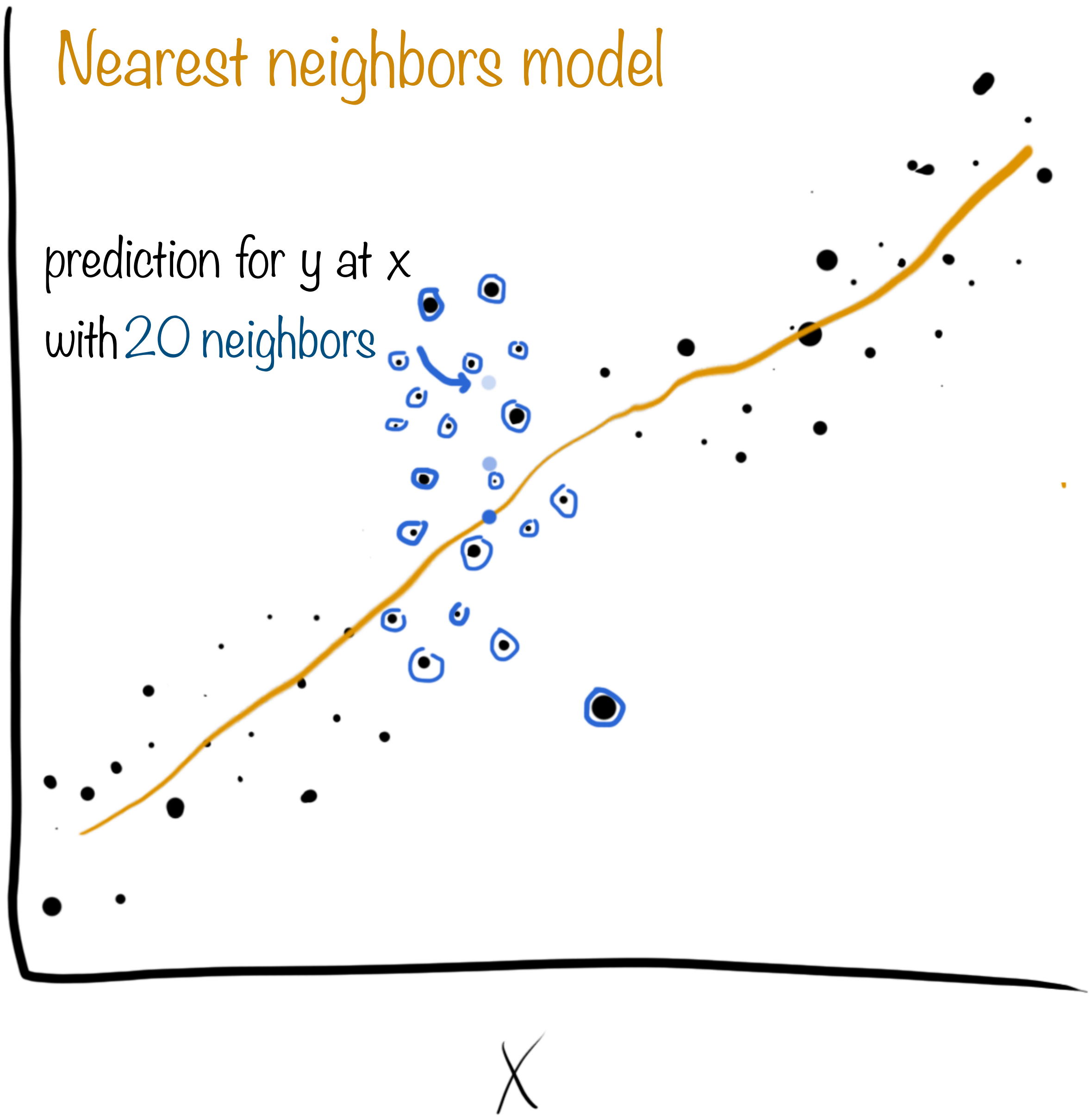




Nearest neighbors model

prediction for y at x
with 20 neighbors

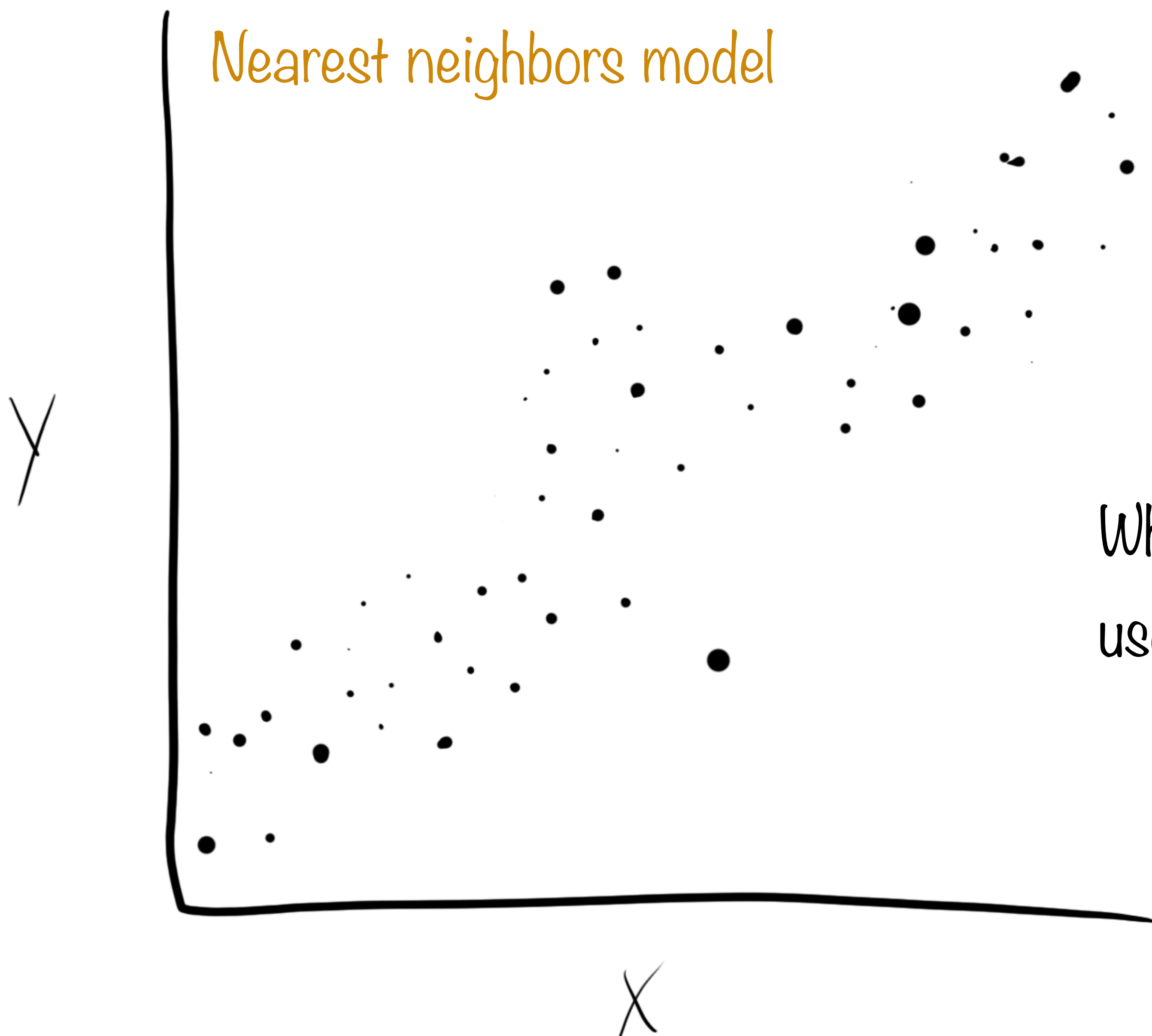
Y



X



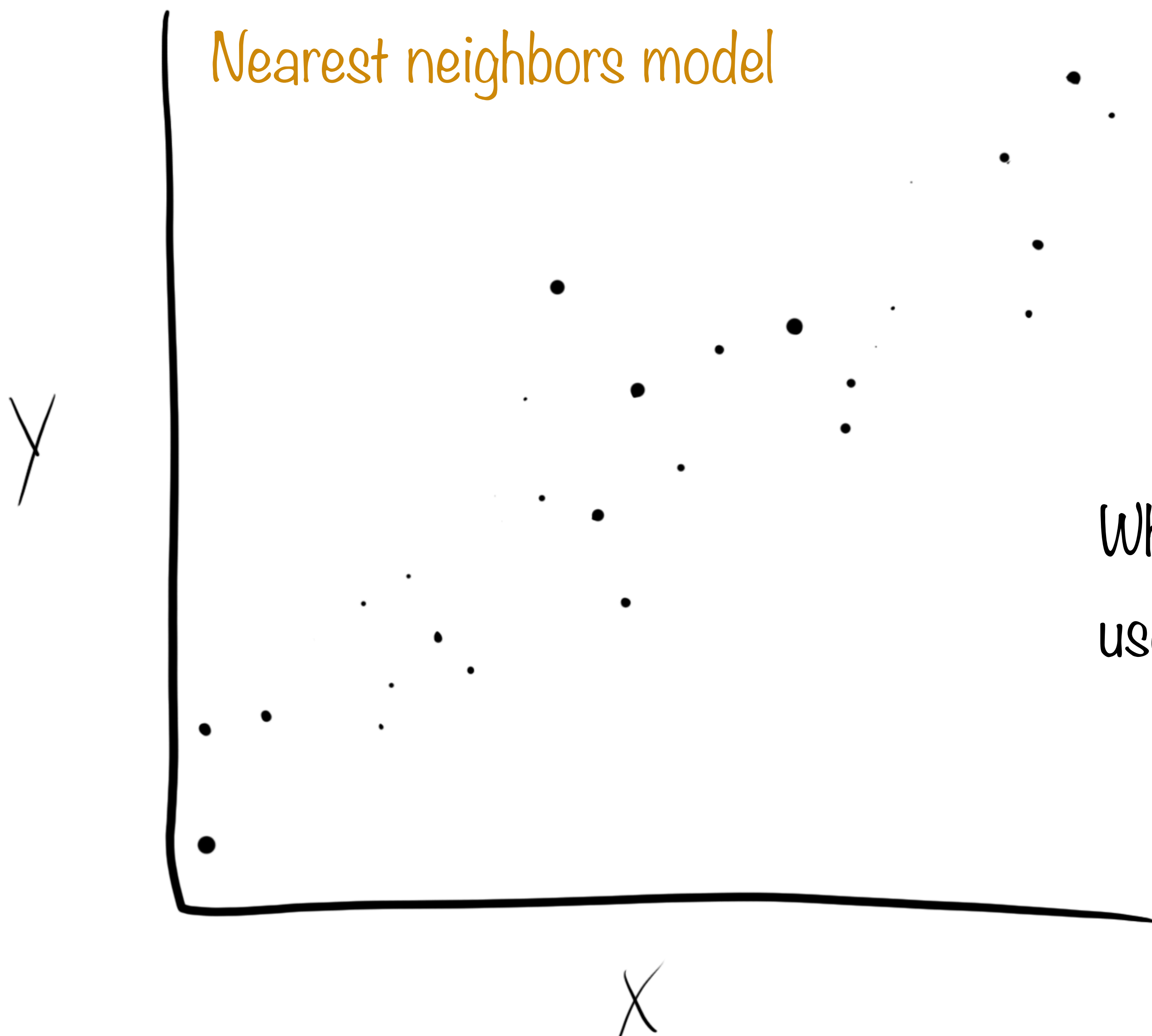
Nearest neighbors model



Why not always
use 10 neighbors?



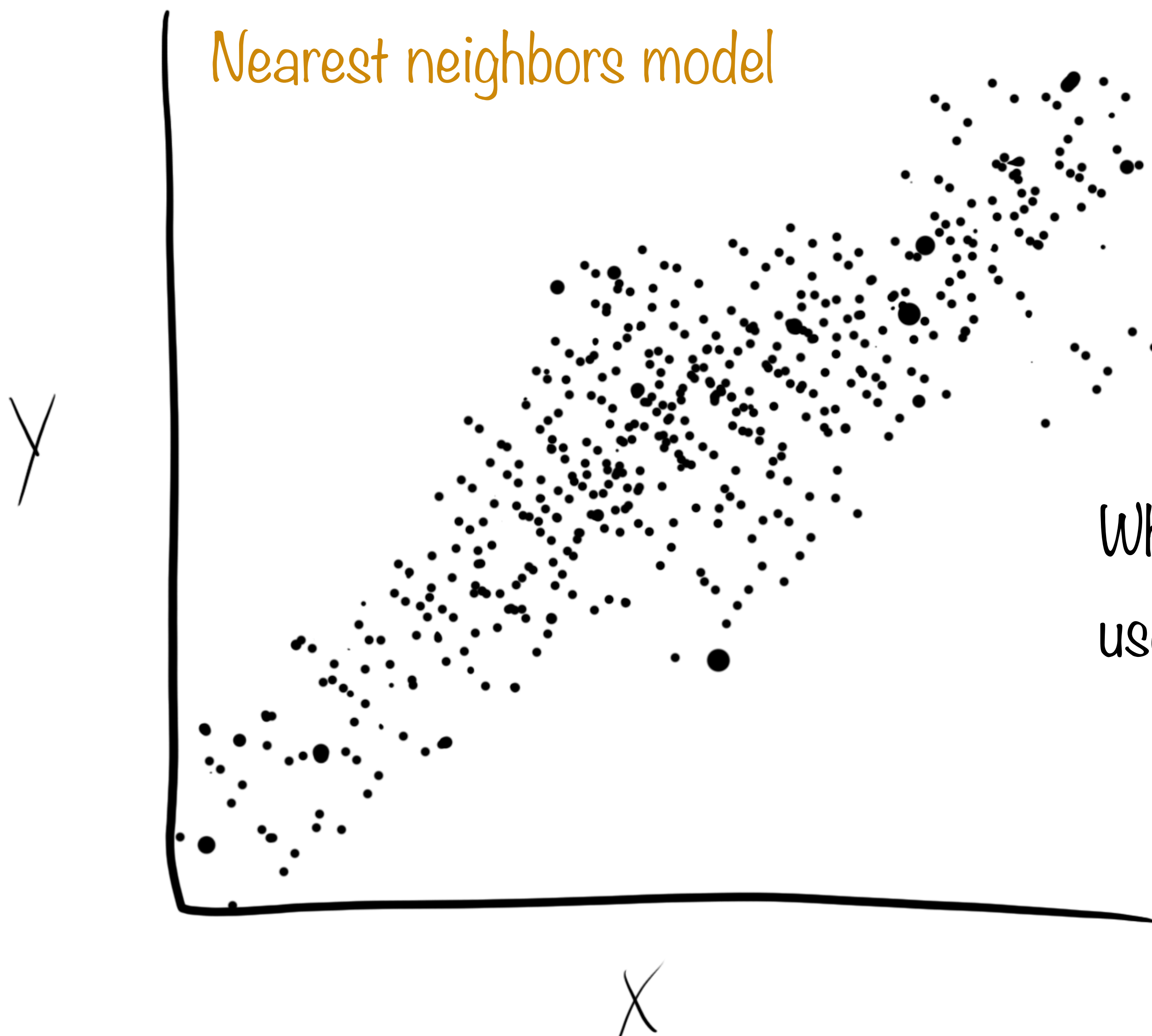
Nearest neighbors model



Why not always
use 10 neighbors?



Nearest neighbors model

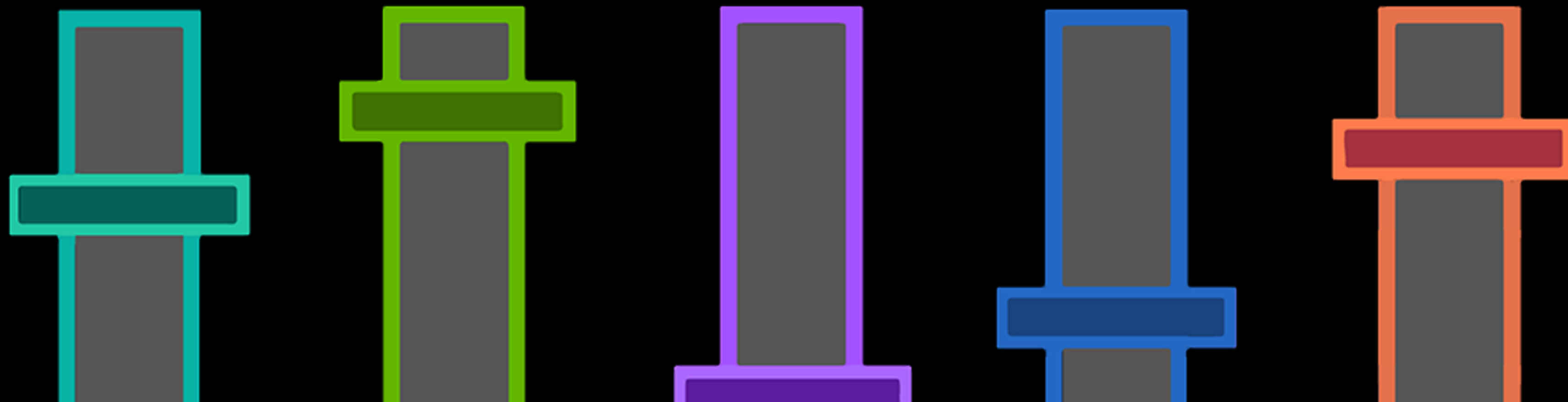


Why not always
use 10 neighbors?



simonpcouch/oss_demo_day_tune

~~Hyperparameter tuning~~ failing to tune *HYPERPARAMETERS, IN 5 MINUTES*





1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier

demo.R x

Source on Save

Run

Source

```
14
15 # tune a nearest neighbors regression
16 res <-
17   tune_grid(
18     nearest_neighbor("regression", neighbors = tune()),
19     y ~ x,
20     bootstraps(dat)
21   )
22
23 autoplot(res)
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
```

16:7 (Top Level) R Script

Console Terminal x Background Jobs x

R 4.2.2 · ~/Documents/rrr/oss_demo_day_tune/

> |



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier



simonpcouch/oss_demo_day_tune

```
data.frame(x = 1:1e6)
```

```
15 # tune a nearest neighbors regression|
16 res <-
17   tune_grid(
18     nearest_neighbor("regression", neighbors = tune()),
19     y ~ x,
20     bootstraps(dat)
21   )
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

15:38 (Top Level) ↕ R Script ↕

Environment History Connections Build Git Tutorial

R 4.2.2 · ~/Documents/rrr/oss_demo_day_tune/ ↗

> |

R 4.2.2 · ~/Documents/rrr/oss_demo_day_tune/ ↗

> |



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*()`
4. Return on interrupt
5. Making tune's job easier


```

15 # tune a nearest neighbors regression,
16 # but be a silly goose
17 raise_concerns <- function(x) {
18   warning("eep!")
19   stop("ope!")
20 }
21
22 ctrl <- control_grid(extract = raise_concerns)
23
24 res <-
25   tune_grid(
26     nearest_neighbor("regression", neighbors = tune()),
27     y ~ x,
28     bootstraps(dat),
29     control = ctrl
30   )
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```

```

> res <-
+   tune_grid(
+     nearest_neighbor("regression", neighbors = tune()),
+     y ~ x,
+     bootstraps(dat),
+     control = ctrl
+   )
→ A | warning: eep!
→ B | error: ope!
There were issues with some computations  A: x25  B: x25
> |

```



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier

Source on Save

RunSource

```
14
15 # tune a nearest neighbors regression,
16 # but be a silly goose
17 raise_concerns <- function(x) {
18   warning("eep!")
19   stop("ope!")
20 }
21
22 ctrl <- control_grid(extract = raise_concerns)
23
24 res <-
25   tune_grid(
26     nearest_neighbor("regression", neighbors = tune()),
27     y ~ x,
28     bootstraps(dat),
29     control = ctrl
30 )
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

30:4 (Top Level) R Script

R 4.2.2 · ~/Documents/rrr/oss_demo_day_tune/

```
> res <-
+   tune_grid(
+     nearest_neighbor("regression", neighbors = tune()),
+     y ~ x,
+     bootstraps(dat),
+     control = ctrl
+   )
→ A | warning: eep!
→ B | error: ope!
There were issues with some computations  A: x25  B: x25
> |
```



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier



1. `.Last.tune.result`
2. The "issue cataloger"
3. `collect_*`
4. Return on interrupt
5. Making tune's job easier

```
15 # tune a nearest neighbors regression
16 res <-
17   tune_grid(
18     nearest_neighbor("regression", neighbors = tune()),
19     y ~ x,
20     bootstraps(dat)
21   )
```

>



Slides and source code:

`github.com/simonpcouch/oss_demo_day_tune`

Install dev tune with:

```
pak::pak("tidymodels/tune")
```