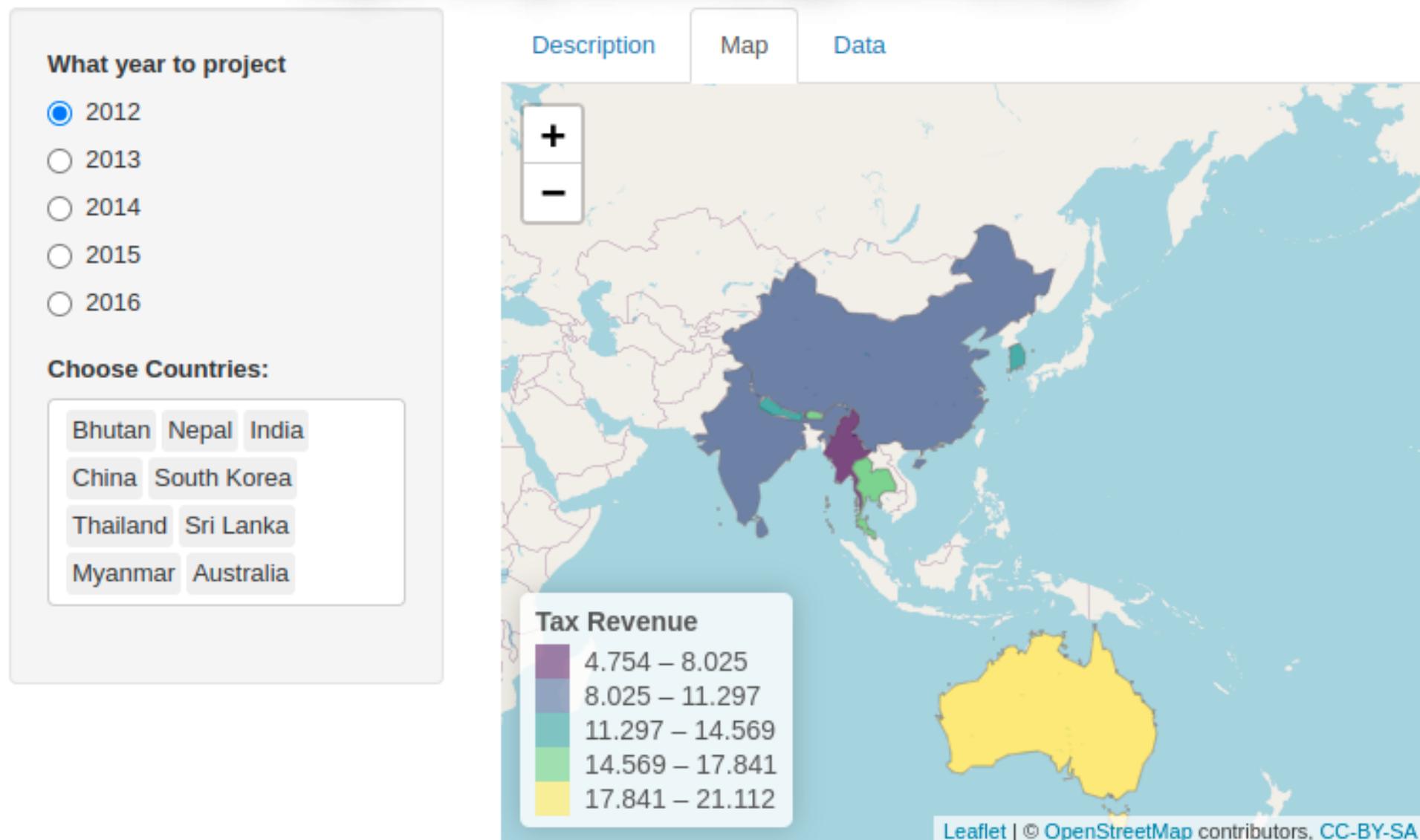


Introduction to Classification

Fall 2022

November 02 2022

Mini project 3 https://deepbas.shinyapps.io/my_app/



Classification

Predicting what category a (future) observation falls into

Examples:

- Astronomy: Whether an exoplanet is habitable (or not)
- Filtering: Identify spam emails
- Medicine: Use lab results to determine who has a disease (or not)
- Product preference: make product recommendations based on past purchases

A photograph of a forest fire in a dense woodland. Tall, thin trees stand in a grid-like pattern, many with white, peeling bark. A large, intense fire is visible on the left, with bright orange flames and thick, billowing smoke. Another smaller fire is visible on the right side of the frame. The ground is covered in green moss and small plants. A red hose is partially visible on the far left.

Fire can be deadly, destroying homes, wildlife habitat and timber, and polluting air with harmful emission

Predicting the next forest fire ..

Dataset

- contains a culmination of forest fire observations
- based on two regions of Algeria: the Bejaia region and the Sidi Bel-Abbes region.
- from June 2012 to September 2012

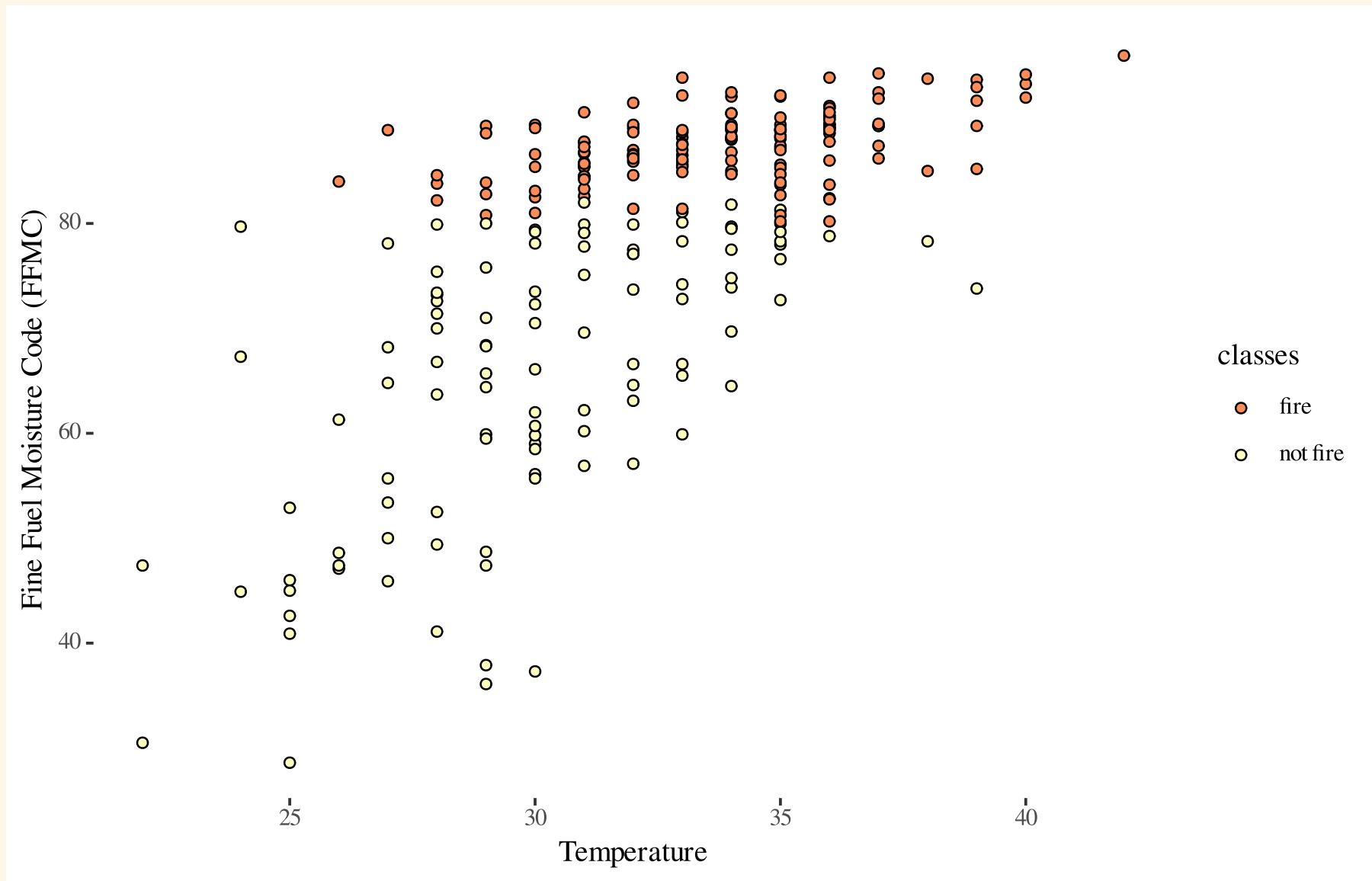
Data Description

Variable	Description
Date	(DD-MM-YYYY) Day, month, year
Temp	Noon temperature in Celsius degrees: 22 to 42
RH	Relative Humidity in percentage: 21 to 90
Ws	Wind speed in km/h: 6 to 29
Rain	Daily total rain in mm: 0 to 16.8
Fine Fuel Moisture Code (FFMC) index	28.6 to 92.5
Duff Moisture Code (DMC) index	1.1 to 65.9
Drought Code (DC) index	7 to 220.4
Initial Spread Index (ISI) index	0 to 18.5
Buildup Index (BUI) index	1.1 to 68
Fire Weather Index (FWI) index	0 to 31.1
Classes	Two classes, namely fire and not fire

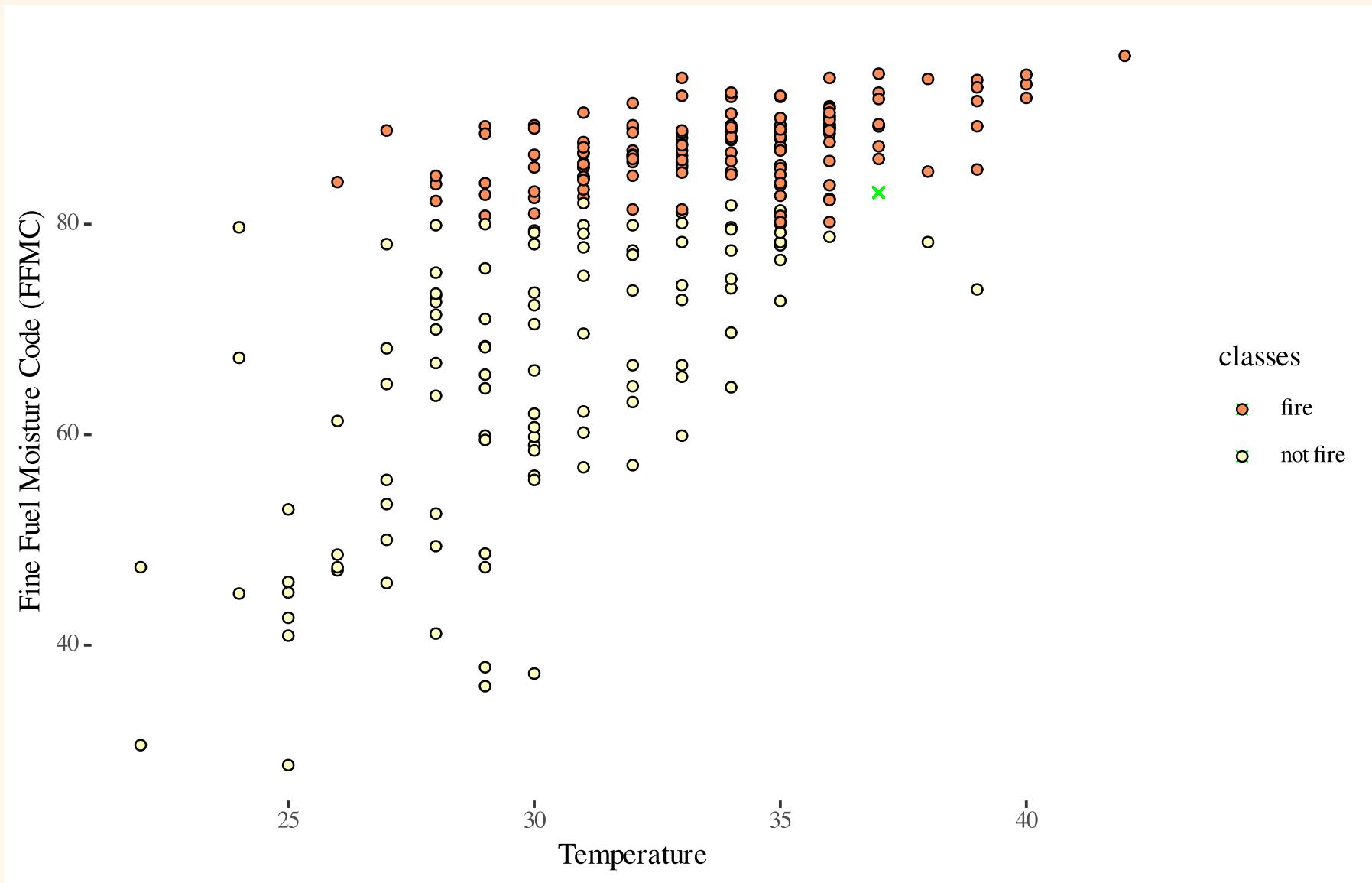
Glimpse of the data

```
glimpse(fire)
Rows: 243
Columns: 12
$ date      <dttm> 2012-06-01, 2012-06-02, 2012-06-03, 2012-06-04, 2012-06-0...
$ temperature <dbl> 29, 29, 26, 25, 27, 31, 33, 30, 25, 28, 31, 26, 27, 30, 28...
$ rh          <dbl> 57, 61, 82, 89, 77, 67, 54, 73, 88, 79, 65, 81, 84, 78, 80...
$ ws          <dbl> 18, 13, 22, 13, 16, 14, 13, 15, 13, 12, 14, 19, 21, 20, 17...
$ rain         <dbl> 0.0, 1.3, 13.1, 2.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2, 0.0, 0.0, 0.0, 0.0...
$ ffmc        <dbl> 65.7, 64.4, 47.1, 28.6, 64.8, 82.6, 88.2, 86.6, 52.9, 73.2...
$ dmc          <dbl> 3.4, 4.1, 2.5, 1.3, 3.0, 5.8, 9.9, 12.1, 7.9, 9.5, 12.5, 1...
$ dc           <dbl> 7.6, 7.6, 7.1, 6.9, 14.2, 22.2, 30.5, 38.3, 38.8, 46.3, 54...
$ isi          <dbl> 1.3, 1.0, 0.3, 0.0, 1.2, 3.1, 6.4, 5.6, 0.4, 1.3, 4.0, 4.8...
$ bui          <dbl> 3.4, 3.9, 2.7, 1.7, 3.9, 7.0, 10.9, 13.5, 10.5, 12.6, 15.8...
$ fwi          <dbl> 0.5, 0.4, 0.1, 0.0, 0.5, 2.5, 7.2, 7.1, 0.3, 0.9, 5.6, 7.1...
$ classes     <chr> "not fire", "not fire", "not fire", "not fire", "not fire"...
```

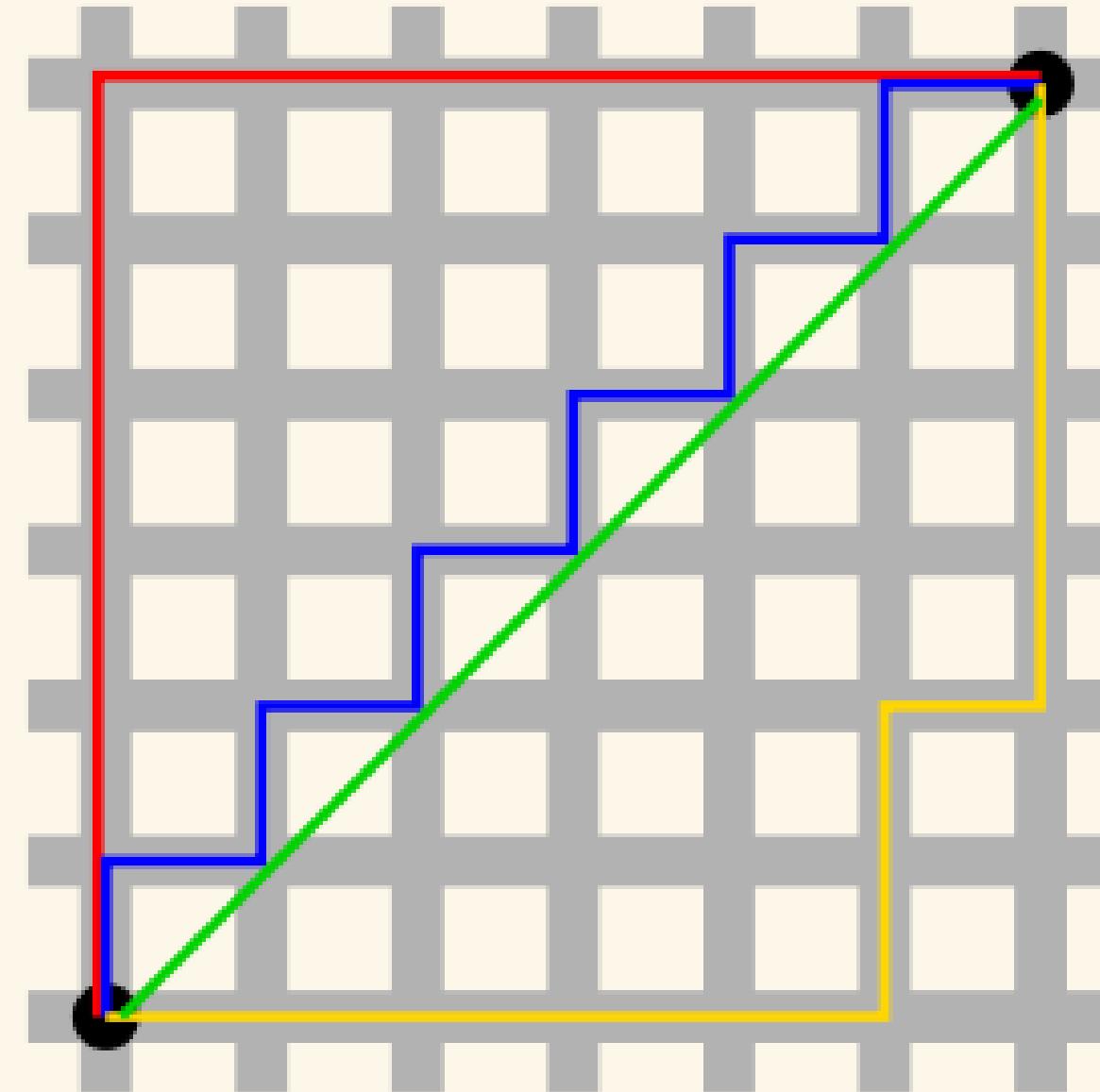
Scatterplot



How can we classify a new observation?



Calculating distance



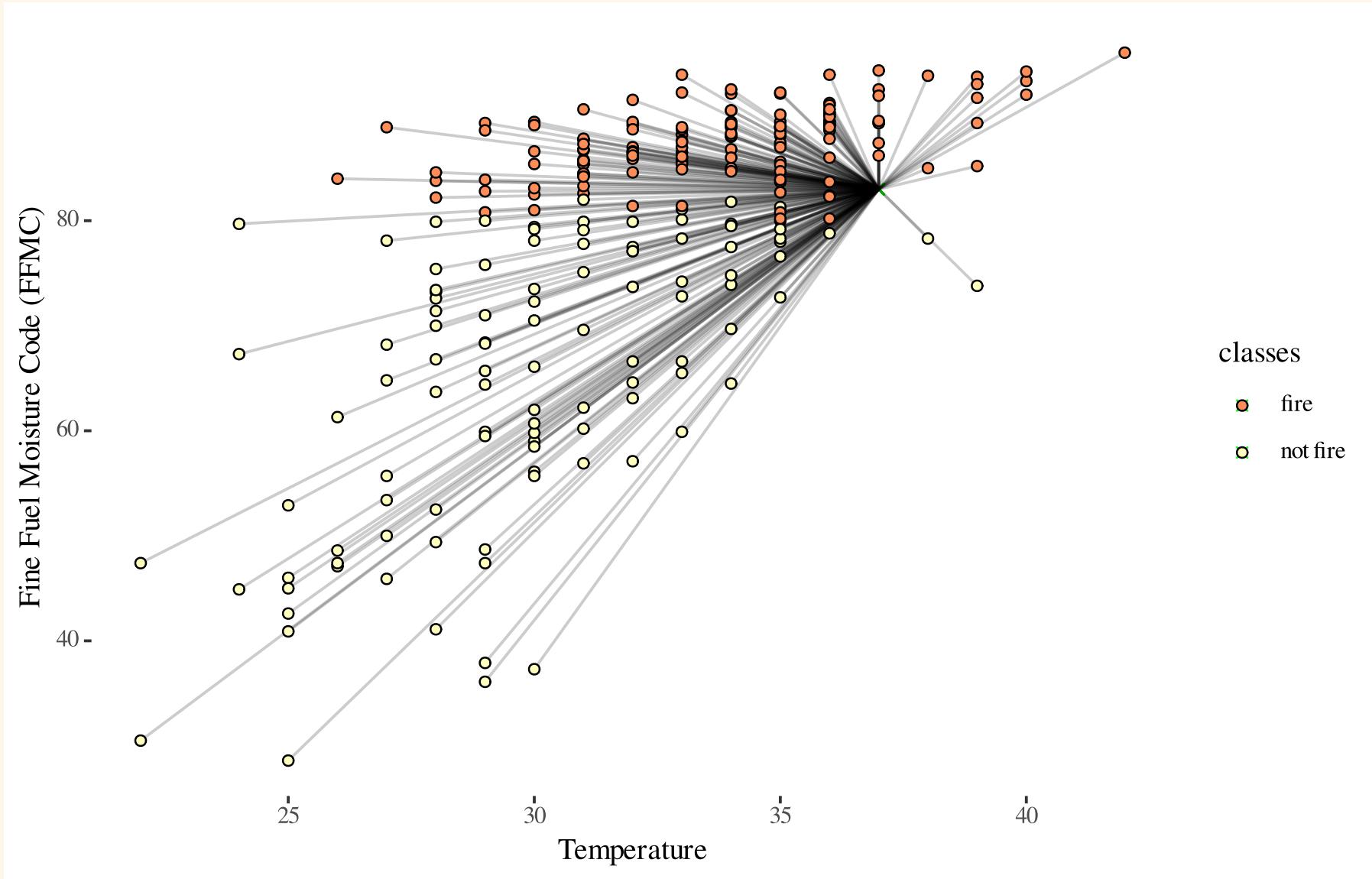
Euclidean distance: the straight line distance between two points on the x-y plane with coordinates $((x_a, y_a))$ and $((x_b, y_b))$

$$\text{Distance} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

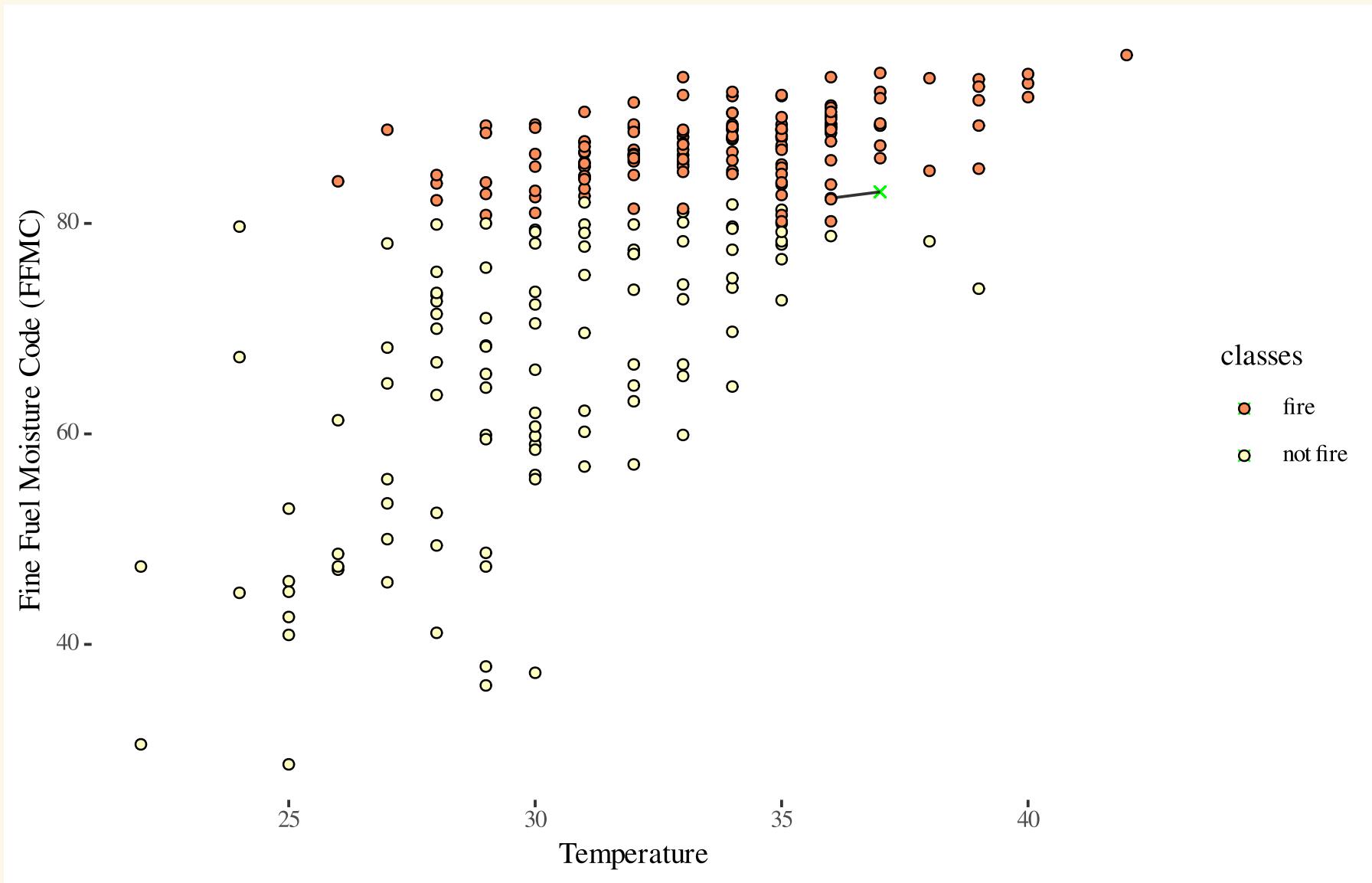
Manhattan distance: the "taxi-cab" distance between two points on the x-y plane

$$\text{Distance} = |x_a - x_b| + |y_a - y_b|$$

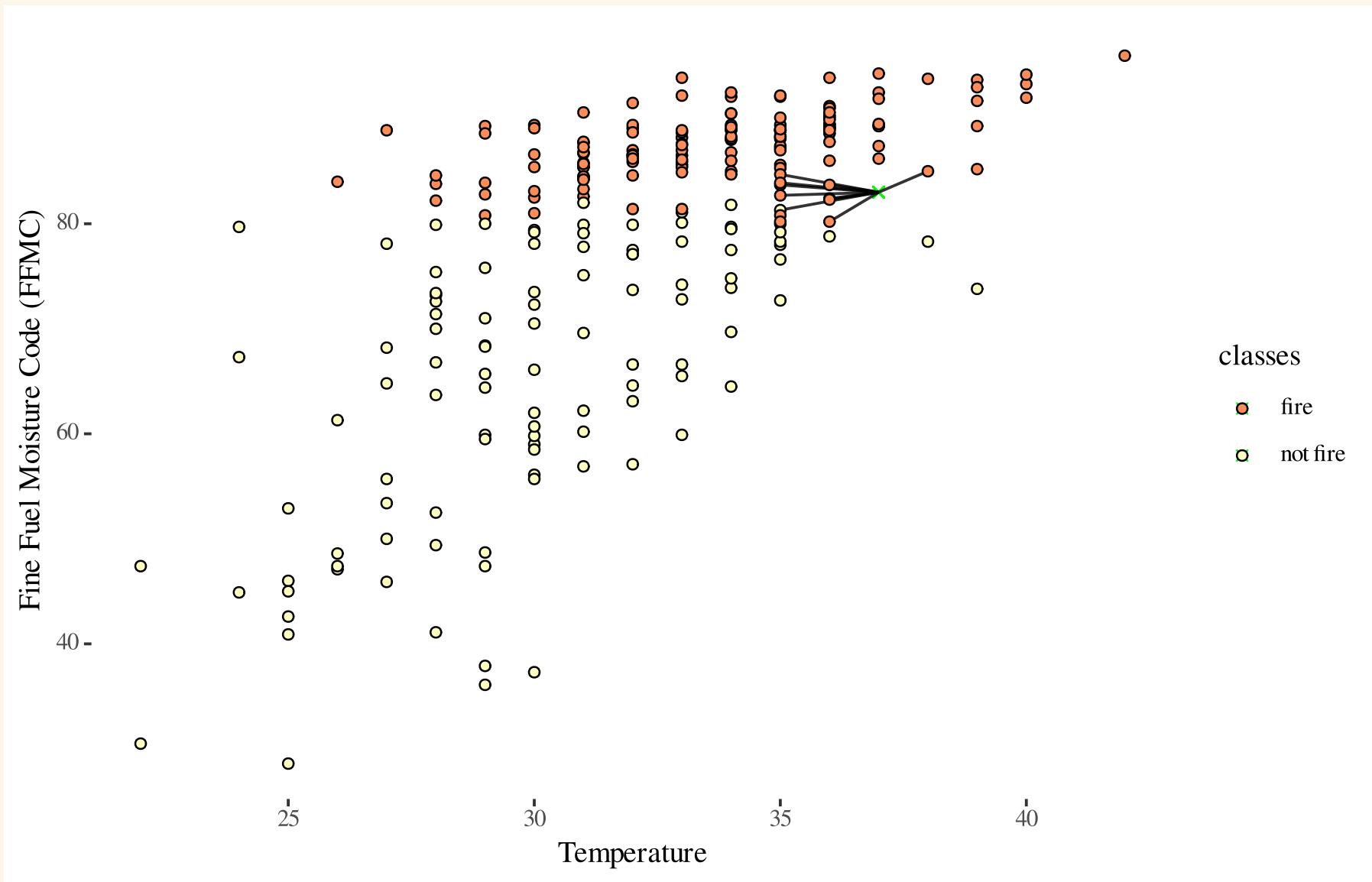
Looking at Euclidean distance



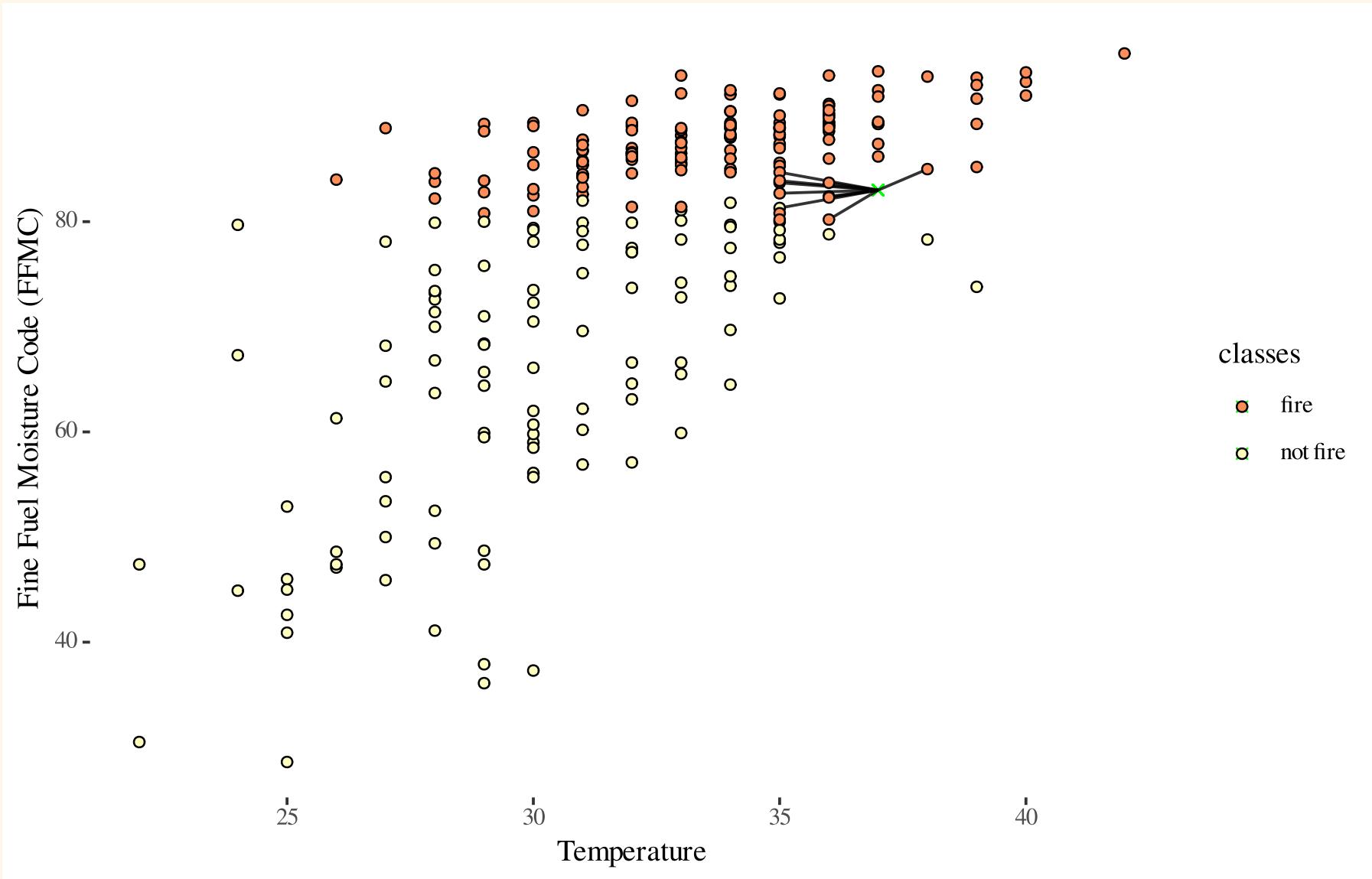
1-Nearest Neighbor (NN)



10-NN



Wait, something is not quite right..



Need to standardize data

```
standardize <- function(x, na.rm = FALSE) {  
  (x - mean(x, na.rm = na.rm)) / sd(x, na.rm = na.rm)  
}
```

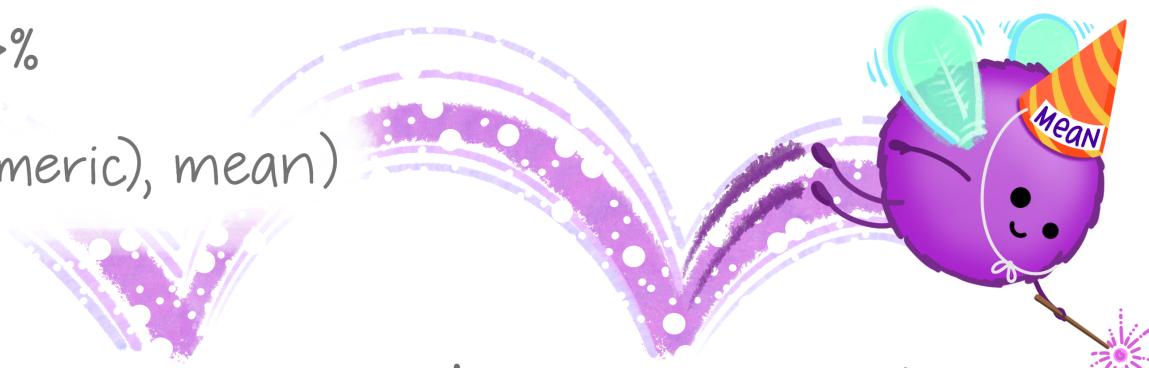
- Predictors with larger variation will have larger influence on which cases are “nearest” neighbors
- Methods relying on distance can be sensitive (i.e. not invariant) to the scale of the predictors
- Standardizing only shifts and rescales the variable, it doesn't change the shape of the distribution

dplyr::across()

use within `mutate()` or `summarize()` to apply function(s) to a **selection of columns!**

EXAMPLE:

```
df %>%  
  group_by(species) %>%  
  summarise(  
    across(where(is.numeric), mean)  
)
```



species	mass_g	age_yr	range_sqmi
pika	163	2.4	0.46
marmot	1509	3.0	0.87
marmot	2417	5.6	0.62

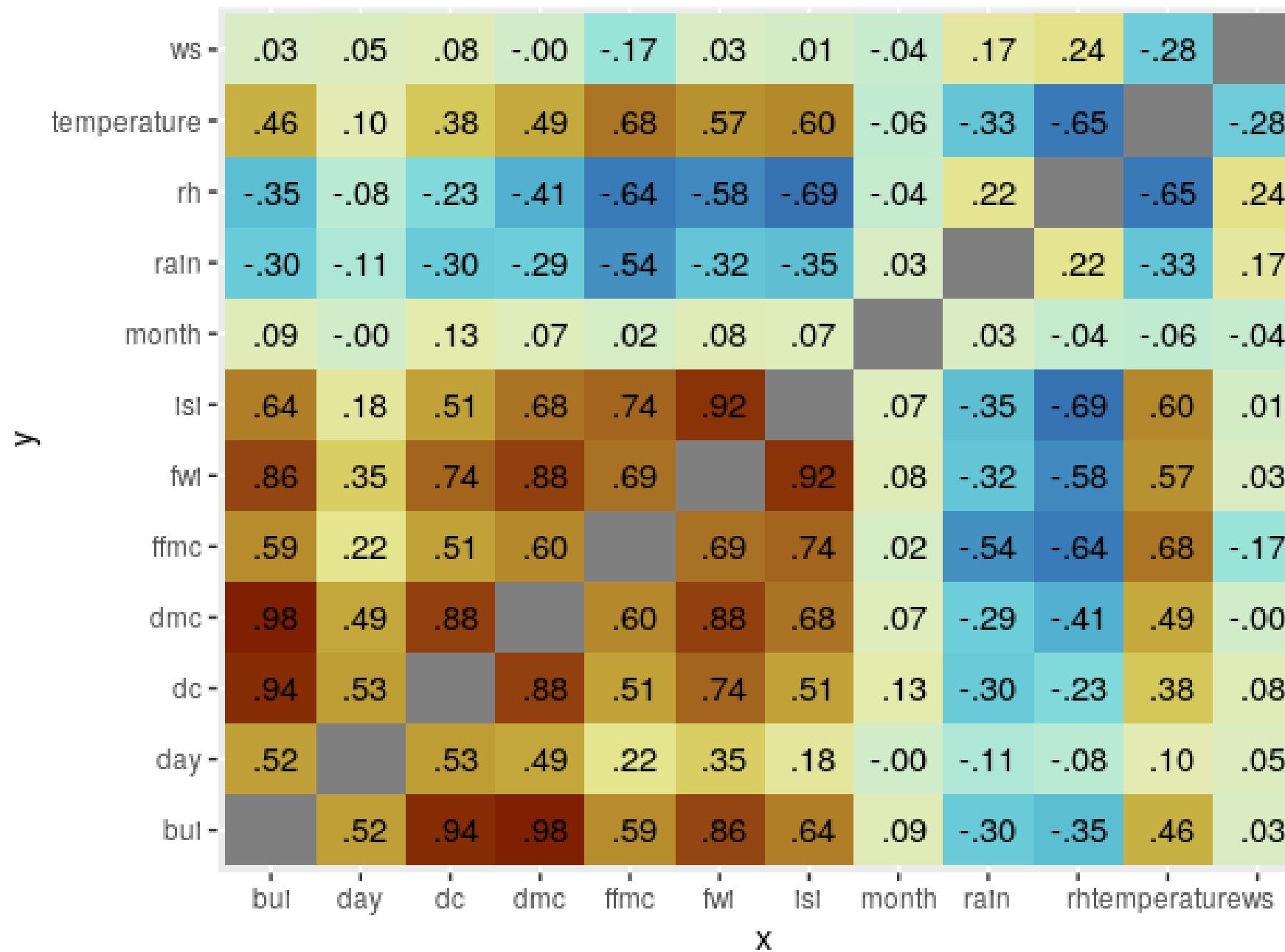
Standardized data

```
fire1 <- fire %>% mutate(across(where(is.numeric), standardize))
fire1 %>% summary()
  date              temperature          rh
  Min.   :2012-06-01 00:00:00.00  Min.   :-2.79828  Min.   :-2.76778
  1st Qu.:2012-07-01 00:00:00.00  1st Qu.:-0.59323  1st Qu.:-0.64345
  Median :2012-08-01 00:00:00.00  Median :-0.04197  Median : 0.06466
  Mean   :2012-07-31 13:43:42.22  Mean   : 0.00000  Mean   : 0.00000
  3rd Qu.:2012-08-31 00:00:00.00  3rd Qu.: 0.78492  3rd Qu.: 0.77278
  Max.   :2012-09-30 00:00:00.00  Max.   : 2.71434  Max.   : 1.88552

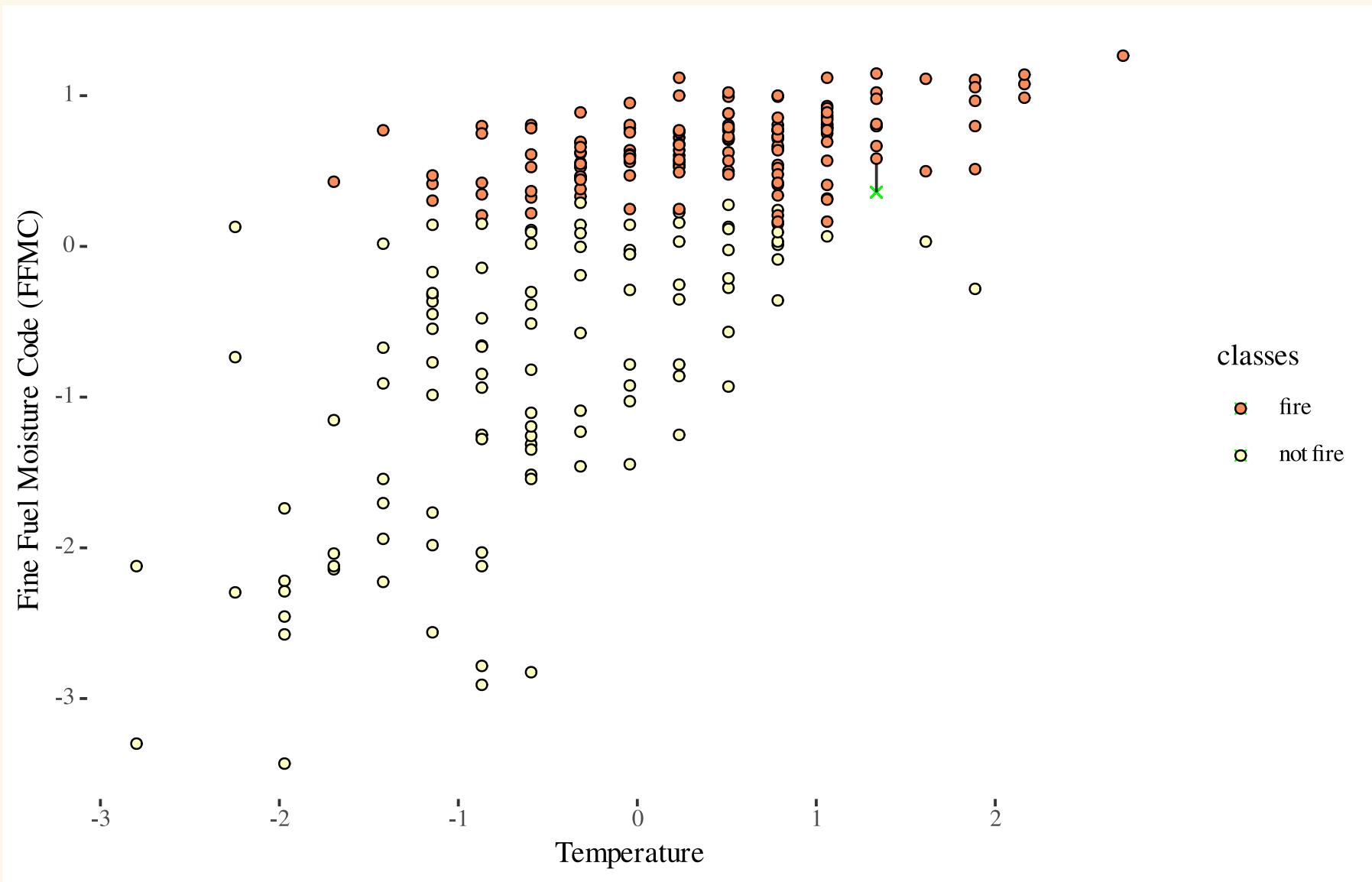
  ws              rain              ffmc          dmc
  Min.   :-3.3769  Min.   :-0.3809  Min.   :-3.4316  Min.   :-1.1281
  1st Qu.:-0.5313 1st Qu.:-0.3809  1st Qu.:-0.4176  1st Qu.:-0.7166
  Median :-0.1757  Median :-0.3809  Median : 0.3803  Median :-0.2728
  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000
  3rd Qu.: 0.5357 3rd Qu.:-0.1313  3rd Qu.: 0.7288  3rd Qu.: 0.4938
  Max.   : 4.8041  Max.   : 8.0057  Max.   : 1.2654  Max.   : 4.1329

  dc              isi              bui           fwi
  Min.   :-0.8923  Min.   :-1.1416  Min.   :-1.0957  Min.   :-0.9455
  1st Qu.:-0.7779 1st Qu.:-0.8046  1st Qu.:-0.7514  1st Qu.:-0.8515
  Median :-0.3426  Median :-0.2991  Median :-0.3015  Median :-0.3811
  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000
  3rd Qu.: 0.4126 3rd Qu.: 0.6036  3rd Qu.: 0.4188  3rd Qu.: 0.5933
  Max.   : 3.5868  Max.   : 3.4321  Max.   : 3.6061  Max.   : 3.2342

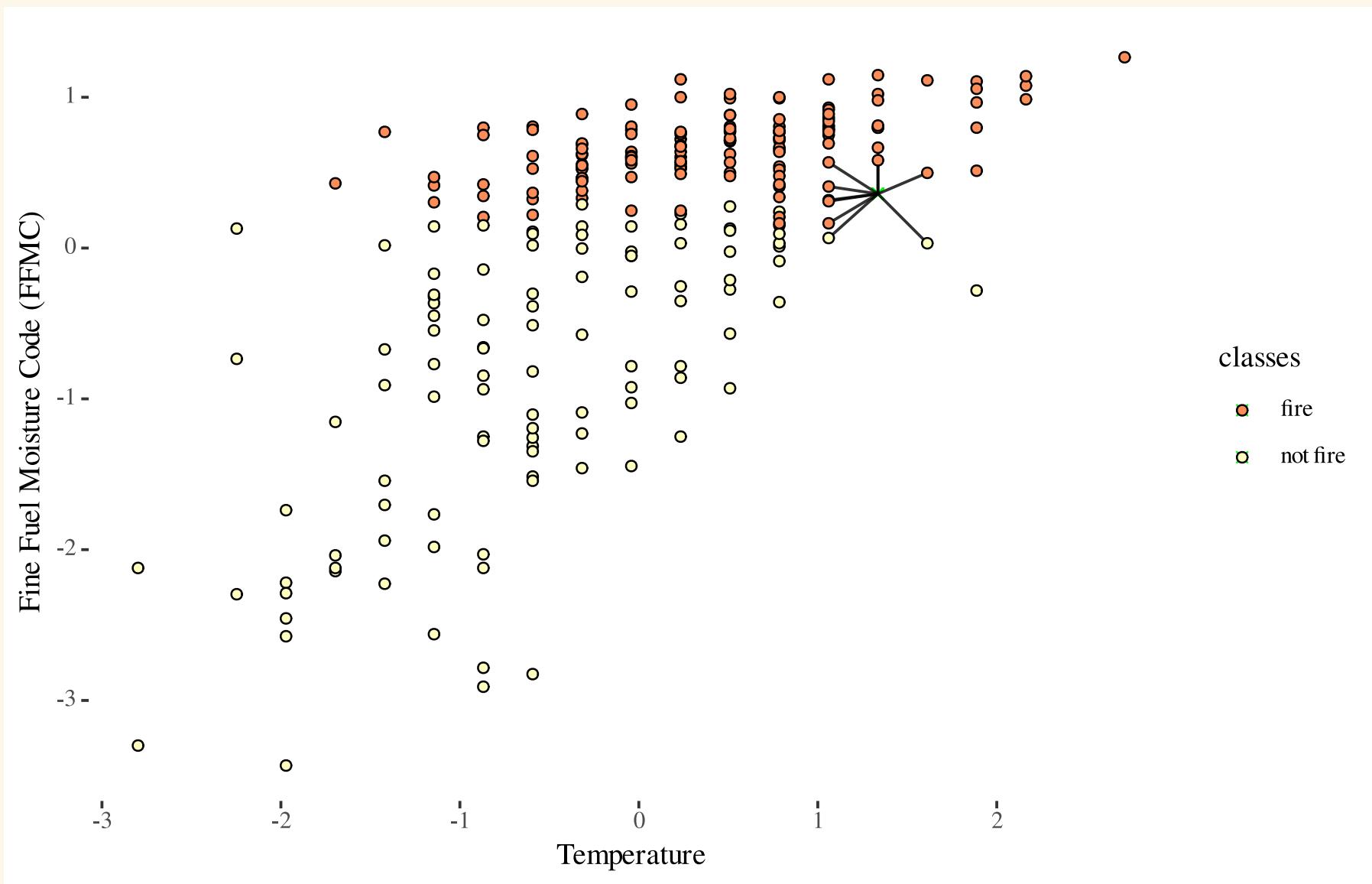
  classes
Length:243
Class :character
Mode  :character
```



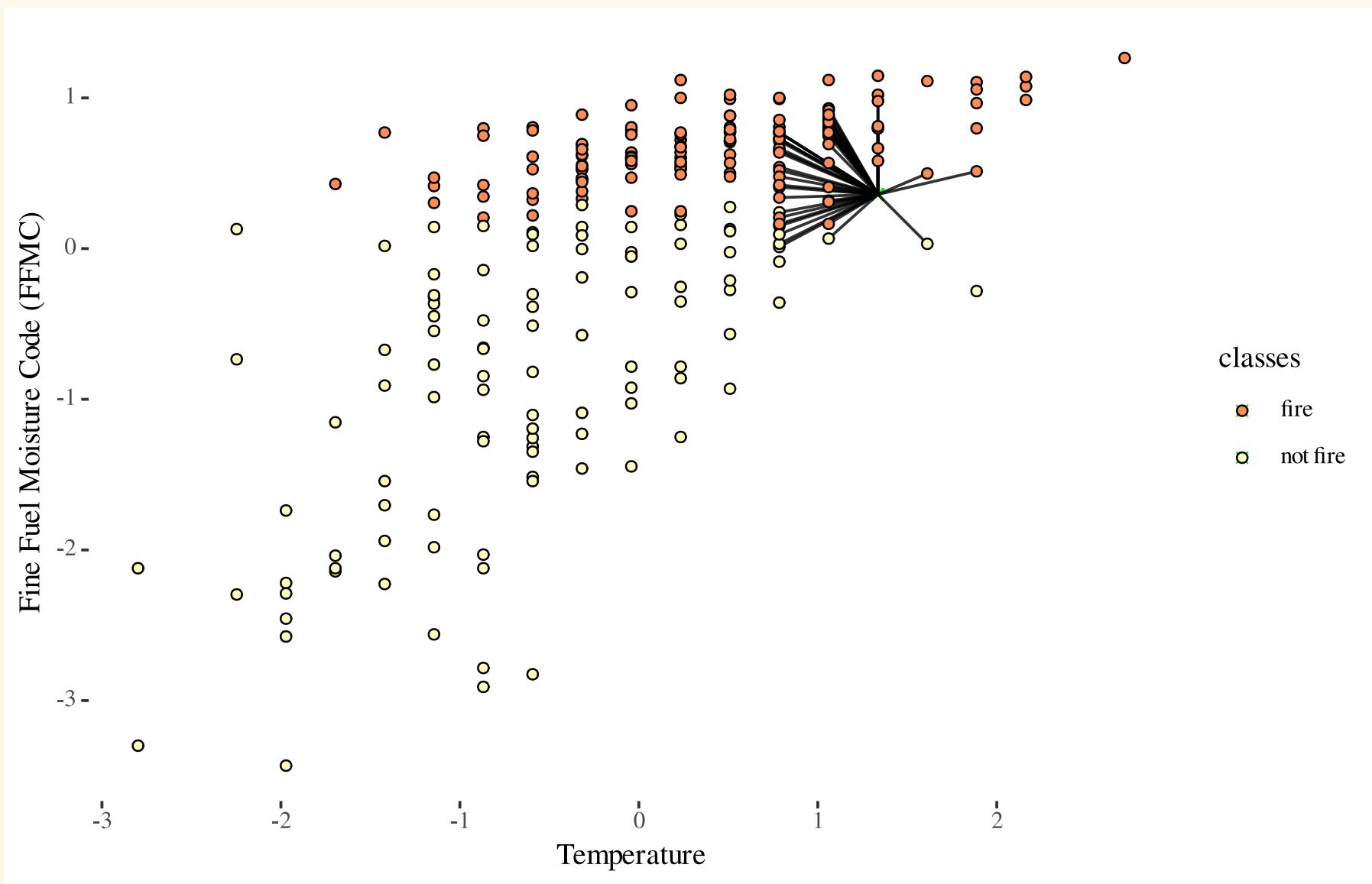
1-NN again



10-NN again



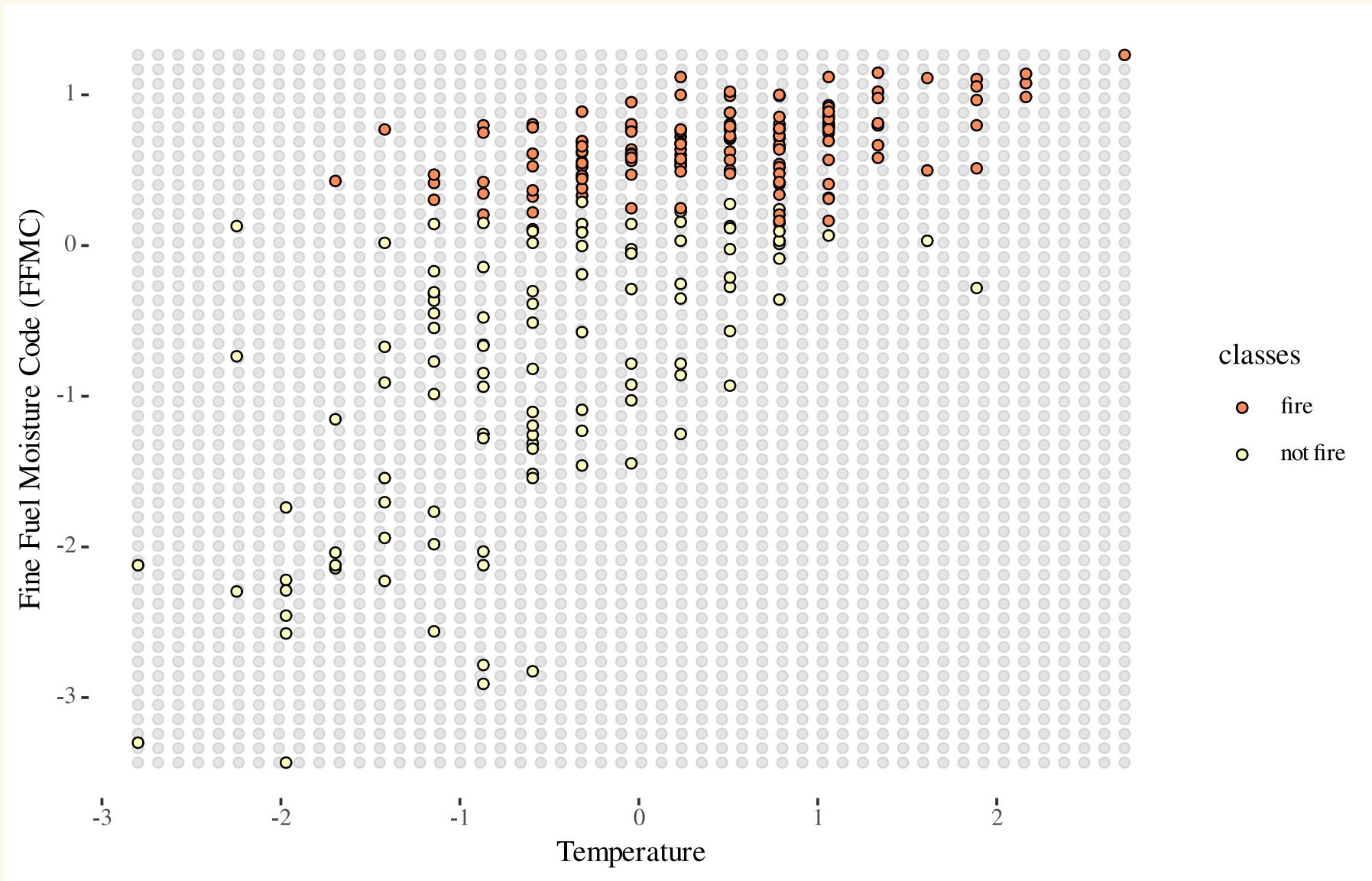
50-NN again



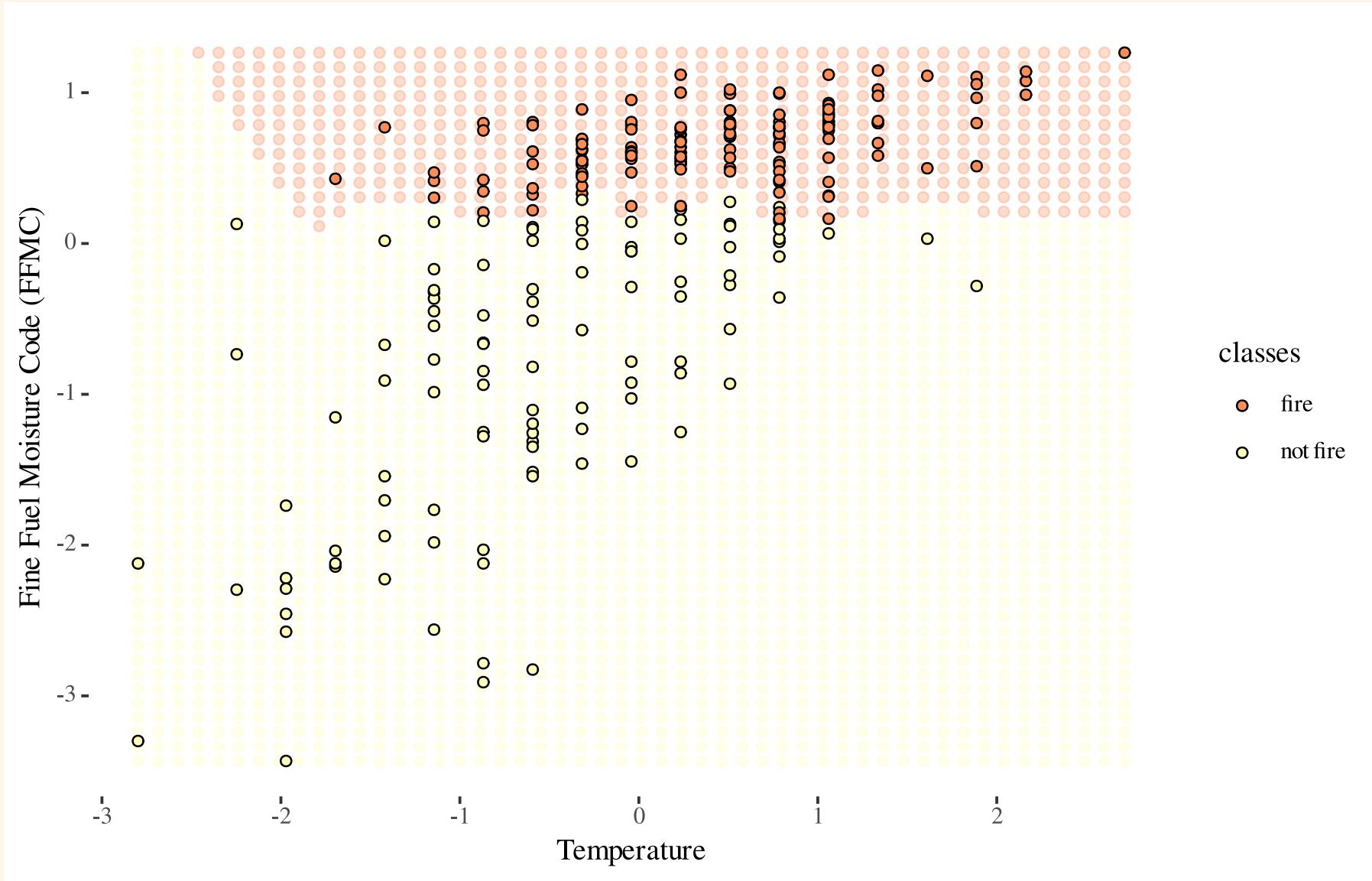
Visualizing the decision boundary

- We can map out the region in feature-space where the classifier would predict 'fire', and the kinds where it would predict 'not fire'
- There is some boundary between the two, where points on one side of the boundary will be classified 'fire' and points on the other side will be classified 'not fire'
- This boundary is called decision boundary

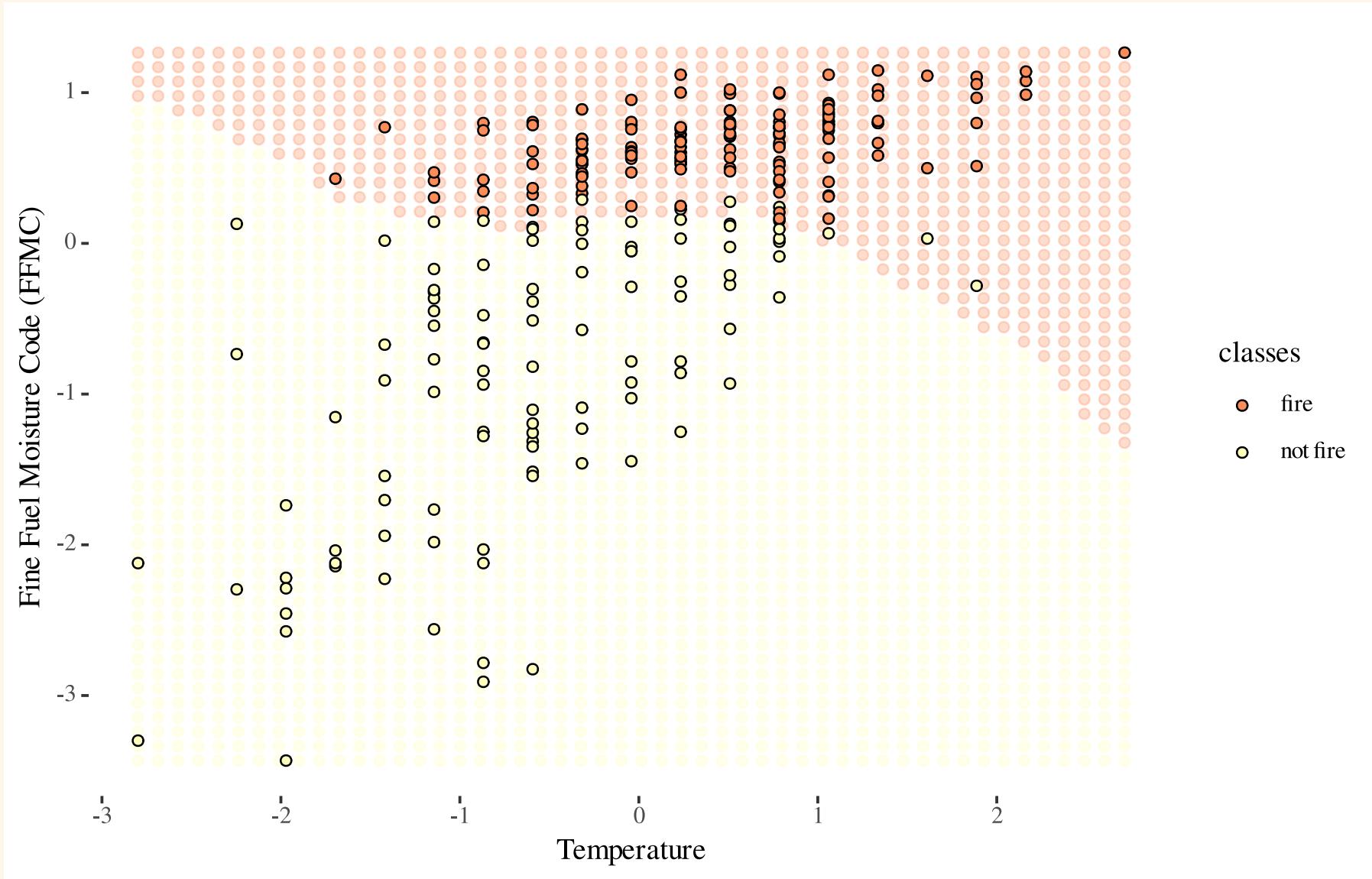
Visualizing the decision boundary

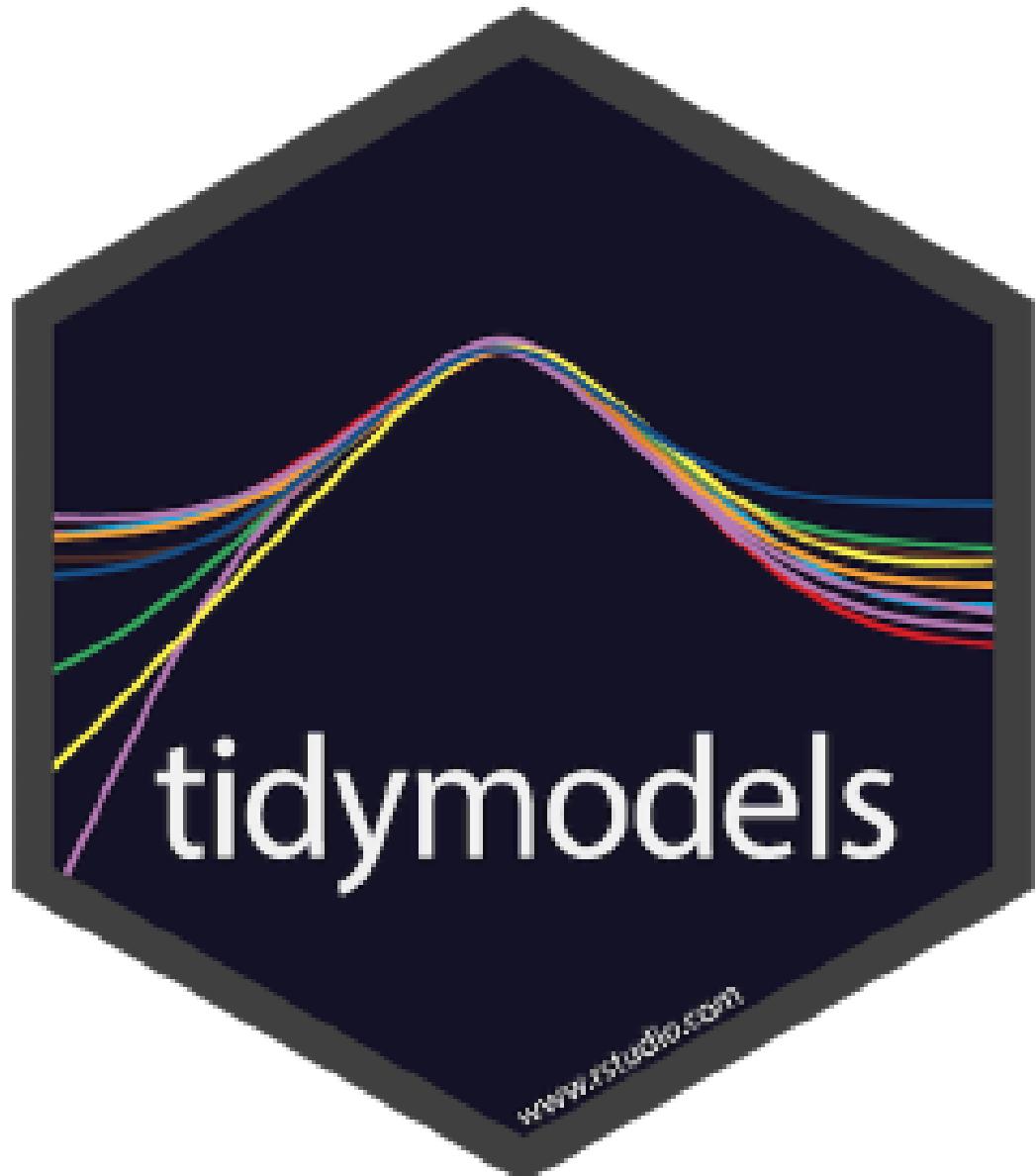


1-NN decision boundary



25-NN decision boundary





a collection of packages for
modeling and machine learning using
tidyverse principles

1. Load data and convert types

```
fire_raw <- read_csv("https://raw.githubusercontent.com/deepbas/statc  
clean_names() %>% na.omit() %>%  
mutate(classes = as_factor(classes)) %>%  
mutate_at(c(10,13), as.numeric) %>%  
select(temperature, ffmc, classes)
```

```
head(fire_raw)  
# A tibble: 6 × 3  
  temperature ffmc classes  
        <dbl> <dbl> <fct>  
1          29   65.7 not fire  
2          29   64.4 not fire  
3          26   47.1 not fire  
4          25   28.6 not fire  
5          27   64.8 not fire  
6          31   82.6 fire
```



2. Create a recipe for data preprocessing

```
fire_recipe <- recipe(classes ~ ., data = fire_raw) %>%  
  step_scale(all_predictors()) %>%  
  step_center(all_predictors()) %>%  
  prep()
```

3. Apply the recipe to the data set

```
fire_scaled <- bake(fire_recipe, fire_raw)
```

```
# A tibble: 243 × 3
  temperature    ffmc classes
  <dbl>     <dbl> <fct>
1 -0.869   -0.846 not fire
2 -0.869   -0.937 not fire
3 -1.70    -2.14  not fire
4 -1.97    -3.43  not fire
5 -1.42    -0.909 not fire
6 -0.318    0.332 fire
7  0.234    0.722 fire
8 -0.593    0.610 fire
9 -1.97    -1.74  not fire
10 -1.14   -0.324 not fire
# ... with 233 more rows
```

4. Create a model specification

```
knn_spec <- nearest_neighbor(mode = "classification",
                                engine = "kknn",
                                weight_func = "rectangular",
                                neighbors = 5)
```

5. Fit the model on the preprocessed data

```
knn_fit <- knn_spec %>%
  fit(classes ~ ., data = fire_scaled)
```

6. Classify

Suppose we get two new observations, use predict to classify the observations

```
# Data frame/tibble of new observations  
new_observations <- tibble(temperature = c(1, 2), ffmc = c(-1, 1))
```

```
# Making classifications (i.e. predictions)  
predict(knn_fit, new_data = new_observations)  
# A tibble: 2 × 1  
  .pred_class  
  <fct>  
1 not fire  
2 fire
```

Further Practice: Pima Indians Diabetes

- Owned by the National Institute of Diabetes and Digestive and Kidney Diseases
- A data frame with 768 observations on 9 variables.
- We have the lab results of 158 patients, including whether they have CKD
- Response variable: `diabetes` = pos, neg
- Predictor variables: *pregnant, glucose, pressure, triceps, insulin, mass, pedigree, age*

Variables

Variable	Description
pregnant	Number of times pregnant
glucose	Plasma glucose concentration (glucose tolerance test)
pressure	Diastolic blood pressure (mm Hg)
triceps	Triceps skinfold thickness (mm)
insulin	2-Hour serum insulin (mu U/ml)
mass	Body mass index (weight in kg/(height in m) ²)
pedigree	Diabetes pedigree function
age	Age (years)
diabetes	diabetes case (pos/neg)



Group Activity 1

10:00



- Let's go over to maize server/ local Rstudio and our class [moodle](#)
- Get the class activity 22.Rmd file
- Work on activity 1
- Ask me questions