

# Logistic Regression Models

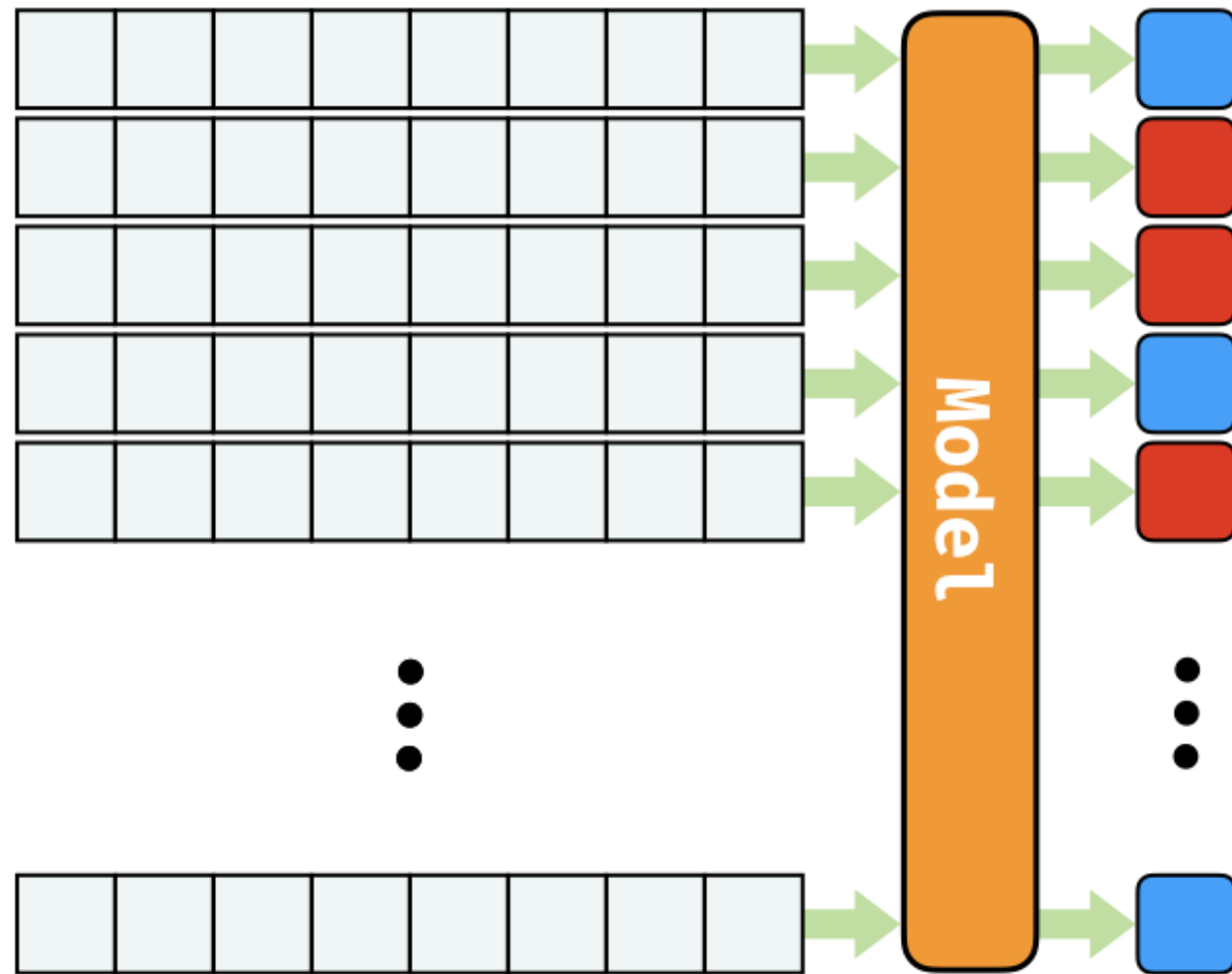
MACHINE LEARNING IN THE TIDYVERSE



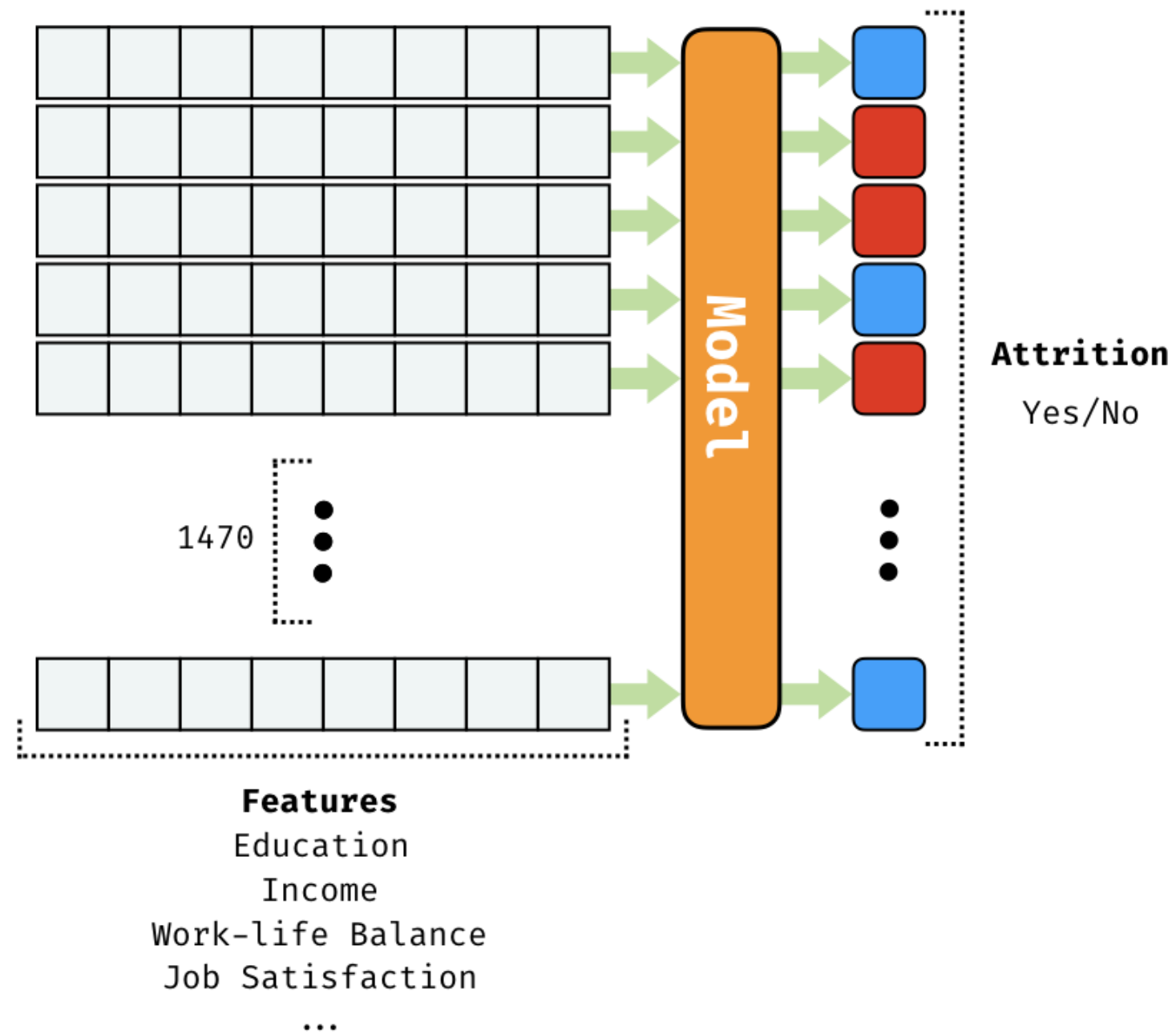
**Dmitriy (Dima) Gorenshteyn**

Lead Data Scientist, Memorial Sloan  
Kettering Cancer Center

# Binary Classification



# The attrition Dataset



# Logistic Regression

```
glm(formula = ___, data = ___, family = "binomial")
```

# glm()

```
head(cv_data)
```

```
# A tibble: 5 x 4
  splits      id  train          validate
* <list>    <chr> <list>          <list>
1 <S3: rsplit> Fold1 <data.frame [882 x 31]> <data.frame [221 x 31]>
2 <S3: rsplit> Fold2 <data.frame [882 x 31]> <data.frame [221 x 31]>
3 <S3: rsplit> Fold3 <data.frame [882 x 31]> <data.frame [221 x 31]>
4 <S3: rsplit> Fold4 <data.frame [883 x 31]> <data.frame [220 x 31]>
5 <S3: rsplit> Fold5 <data.frame [883 x 31]> <data.frame [220 x 31]>
```

```
cv_models_lr <- cv_data %>%
  mutate(model = map(train, ~glm(formula = Attrition~.,
                                data = .x, family = "binomial"))))
```

# Time to Practice

MACHINE LEARNING IN THE TIDYVERSE

# Evaluating Classification Models

MACHINE LEARNING IN THE TIDYVERSE



**Dmitriy (Dima) Gorenshteyn**

Lead Data Scientist, Memorial Sloan  
Kettering Cancer Center

# Ingredients for Performance Measurement

- 1) Actual attrition classes
- 2) Predicted attrition classes
- 3) A metric to compare 1) & 2)



# 1) Prepare Actual Classes

attrition	class
Yes	TRUE
No	FALSE

```
validate$Attrition
```

```
No No No No No Yes No Yes ... No No No
```

```
validate_actual <- validate$Attrition == "Yes"  
validate_actual
```

```
FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE ... FALSE FALSE FALSE
```

## 2) Prepare Predicted Classes

P(attrition)	class
$> 0.5$	TRUE
$\leq 0.5$	FALSE

```
validate_prob <- predict(model, validate, type = "response")  
validate_prob
```

```
0.324 0.012 0.077 0.001 0.104 0.940 0.116 0.811 0.261 0.027 0.065 0.060
```

```
validate_predicted <- validate_prob > 0.5  
validate_predicted
```

```
FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE
```

### 3) A metric to compare 1) & 2)

		Predicted	
		FALSE	TRUE
Actual	FALSE	181	5
	TRUE	17	18

```
table(validate_actual, validate_predicted)
```

```
      validate_predicted
validate_actual FALSE TRUE
      FALSE    181    5
      TRUE     17    18
```

### 3) Metric: Accuracy

		Predicted	
		FALSE	TRUE
Actual	FALSE	181	5
	TRUE	17	18

$$\text{Accuracy} = \frac{181 + 18}{181 + 5 + 17 + 18}$$

```
accuracy(validate_actual, validate_predicted)
```

```
0.9004525
```

### 3) Metric: Precision

		Predicted	
		FALSE	TRUE
Actual	FALSE	181	5
	TRUE	17	18

Precision =  $\frac{18}{5 + 18}$

```
precision(validate_actual, validate_predicted)
```

```
0.7826087
```

### 3) Metric: Recall

		Predicted	
		FALSE	TRUE
Actual	FALSE	181	5
	TRUE	17	18

Recall =

$$\frac{18}{17 + 18}$$

```
recall(validate_actual, validate_predicted)
```

```
0.5142857
```

# Let's practice!

MACHINE LEARNING IN THE TIDYVERSE

# Classification With Random Forests

MACHINE LEARNING IN THE TIDYVERSE



**Dmitriy (Dima) Gorenshteyn**

Lead Data Scientist, Memorial Sloan  
Kettering Cancer Center



# ranger() for Classification

```
cv_tune <- cv_data %>%  
  crossing(mtry = c(2, 4, 8, 16))  
  
cv_models_rf <- cv_tune %>%  
  mutate(model = map2(train, mtry, ~ranger(formula = Attrition~.,  
                                           data = .x, mtry = .y,  
                                           num.trees = 100, seed = 42)))
```

# 1) Prepare Actual Classes

attrition	class
Yes	TRUE
No	FALSE

```
validate$Attrition
```

```
No No No No No Yes No Yes ... No No No
```

```
validate_actual <- validate$Attrition == "Yes"  
validate_actual
```

```
FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE ... FALSE FALSE FALSE
```

## 2) Prepare Predicted Classes

P(attrition)	class
Yes	<b>TRUE</b>
No	<b>FALSE</b>

```
validate_classes <- predict(rf_model, rf_validate)$predictions  
validate_classes
```

```
No No No No No Yes No No ... No No No
```

```
validate_predicted <- validate_classes == "Yes"  
validate_predicted
```

```
FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE ... FALSE FALSE FALSE
```

# Build the Best Attrition Model

MACHINE LEARNING IN THE TIDYVERSE

# Recap: Machine Learning in the Tidyverse

MACHINE LEARNING IN THE TIDYVERSE



**Dmitriy (Dima) Gorenshteyn**

Lead Data Scientist, Memorial Sloan  
Kettering Cancer Center

# Chapter 1 - The List Column Workflow

1

Make a  
list column

`nest()`

2

Work with  
list columns

`map()`

3

Simplify the  
list columns

`unnest()`

`map_*()`

# Chapter 2 - Explore Multiple Models With broom

1

Make a  
`list` column

`nest()`

2

Work with  
`list` columns

`map()`  
`tidy()`  
`glance()`  
`augment()`

3

Simplify the  
`list` columns

`unnest()`

# Chapter 3 - Build, Tune & Evaluate Regression Models

1 Make a  
list column

`nest()`  
`initial_split()`  
`vfold_cv()`  
`crossing()`

2 Work with  
list columns

`map()`  
`training()`  
`testing()`  
`lm()`  
`ranger()`  
`mae()`

3 Simplify the  
list columns

`unnest()`  
`map_dbl()`



# Chapter 4 - Build, Tune & Evaluate Classification Models

1

Make a  
list column

`nest()`  
`initial_split()`  
`vfold_cv()`  
`crossing()`

2

Work with  
list columns

`map()`  
`training()`  
`testing()`  
`glm()`  
`ranger()`  
`recall()`

3

Simplify the  
list columns

`unnest()`  
`map_dbl()`

# Congratulations!

MACHINE LEARNING IN THE TIDYVERSE