https://harvard-iacs.github.io/2019-CS109A/

CAP5768/IDC4140:

Introduction to Data Science

##### Guang Wang Department of Computer Science

Florida State University

Data, Models, and Code

* Be able to soundly prepare machine learning experiments, in terms of data splits and formatting
* Know common strategies in debugging code
* Feel prepared 6543tgrepcheck2&&\*\*(11 tackling the remaining course content / gain confidence!

##### Review of Models Review of Data Debugging

Review of Models

##### Review of Data Debugging

What is the most difficult concept for you right now?

row corresponds to a distinct

**Age**

**Play**

**Rainy**

**Temp**

i.i.d. (Independent and identically distributed) observation

|  |  |  |  |
| --- | --- | --- | --- |
| 22 | N | Y | 91 |
| 29 | Y | N | 89 |
| 31 | N | N | 56 |
| 23 | Y | N | 71 |
| 37 | N | Y | 72 |
| 41 | Y | N | 83 |
| 29 | Y | Y | 97 |
| 21 | N | N | 64 |
| 30 | Y | N | 68 |

* You may be interested in a

particular column

row corresponds to a distinct

**Play**

**Rainy**

**Temp**

**Age**

i.i.d. observation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * You may be interested in a particular column 6543tgrepcheck2&&\*\*(11 | 22  29 | N  Y | Y  N | 91  89 |
|  | 31 | N | N | 56 |
|  | 23 | Y | N | 71 |
|  | 37 | N | Y | 72 |
|  | 41 | Y | N | 83 |
|  | 29 | Y | Y | 97 |
|  | 21 | N | N | 64 |
|  | 30 | Y | N | 68 |

row corresponds to a distinct

**Age**

**Rainy**

**Temp**

**Play**

i.i.d. observation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * You may be interested in a particular column | 22  29 | N  Y | Y  N | 91  89 |
|  | 31 | N | N | 56 |
|  | 23 | Y | N | 71 |
|  | 37 | N | Y | 72 |
|  | 41 | Y | N | 83 |
|  | 29 | Y | Y | 97 |
|  | 21 | N | N | 64 |
|  | 30 | Y | N | 68 |

row corresponds to a distinct

**Age**

**Play**

**Rainy**

**Temp**

i.i.d. observation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * You may be interested in a particular column | 22  29 | N  Y | Y  N | 91  89 |
|  | 31 | N | N | 56 |
|  | 23 | Y | N | 71 |
|  | 37 | N | Y | 72 |
|  | 41 | Y | N | 83 |
|  | 29 | Y | Y | 97 |
|  | 21 | N | N | 64 |
|  | 30 | Y | N | 68 |

row corresponds to a distinct

**Age**

**Play**

**Rainy**

**Temp**

i.i.d. observation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * You may be interested in a particular column (e.g. **Temp**) | 22  29 | N  Y | Y  N | 91  89 |
|  | 31 | N | N | 56 |
|  | 23 | Y | N | 71 |
|  | 37 | N | Y | 72 |
|  | 41 | Y | N | 83 |
|  | 29 | Y | Y | 97 |
|  | 21 | N | N | 64 |
|  | 30 | Y | N | 68 |
|  |  |  |  |  |

row corresponds to a distinct

i.i.d. observation

* You may be interested in a particular column (e.g. **Temp**)
* Let’s divide our data and learn how data **X** is related to data **Y**

𝑿

* Assert that:

**Age**

22

29

31

23

37

41

29

21

30

**Play**

N Y N Y N Y Y N

Y

**Rainy**

Y N N N Y N Y N

N

**Temp**

91

89

56

71

72

83

97

64

68

##### 𝒀 = 𝑓

+ 𝜀

row corresponds to a distinct

i.i.d. observation

* You may be interested in a particular column (e.g. **Temp**)
* Let’s divide our data and learn how data **X** is related to data **Y**

𝑿

* Assert that:

##### 𝒀 = 𝑓

+ 𝜀

**Age**

22

29

31

23

37

41

29

21

30

**Play**

N Y N Y N Y Y N

Y

**Rainy**

Y N N N Y N Y N

N

**Temp**

91

89

56

71

72

83

97

64

68



row corresponds to a distinct

i.i.d. observation

* You may be interested in a particular column (e.g. **Temp**)
* Let’s divide our data and learn how data **X** is related to data **Y**

𝑿

* Assert that:

##### 𝒀 = 𝑓

+ 𝜀

* Want a model 𝑓 that is:

**Age**

22

29

31

23

37

41

29

21

30

**Play**

N Y N Y N Y Y N

Y

**Rainy**

Y N N N Y N Y N

N

**Temp**

91

89

56

71

72

83

97

64

68

* + Supervised

**Age**

22

29

31

23

37

41

29

21

30

**Play**

N Y N Y N Y Y N

Y

**Rainy**

Y N N N Y N Y N

N

**Temp**

91

89

56

71

72

83

97

64

68

•

• Given some data such that each row corresponds to a distinct

i.i.d. observation

• You may be interested in a particular column (e.g. **Temp**)

• Let’s divide our data and learn how data **X** is related to data **Y**

• Assert that: 𝒀 = 𝑓 𝑿 + 𝜀

• Want a model 𝑓 that is:

• Supervised

Supervised

#### Def:

• Given some data such that each row corresponds to a distinct

i.i.d. observation

• You may be interested in a particular column (e.g. **Temp**)

• Let’s divide our data and learn how data **X** is related to data **Y**

• Assert that: 𝒀 = 𝑓 𝑿 + 𝜀

• Want a model that is:

• Supervised

**Play Rainy**

#### Temp

**Supervised** models use target N Y 91

**Age**

22

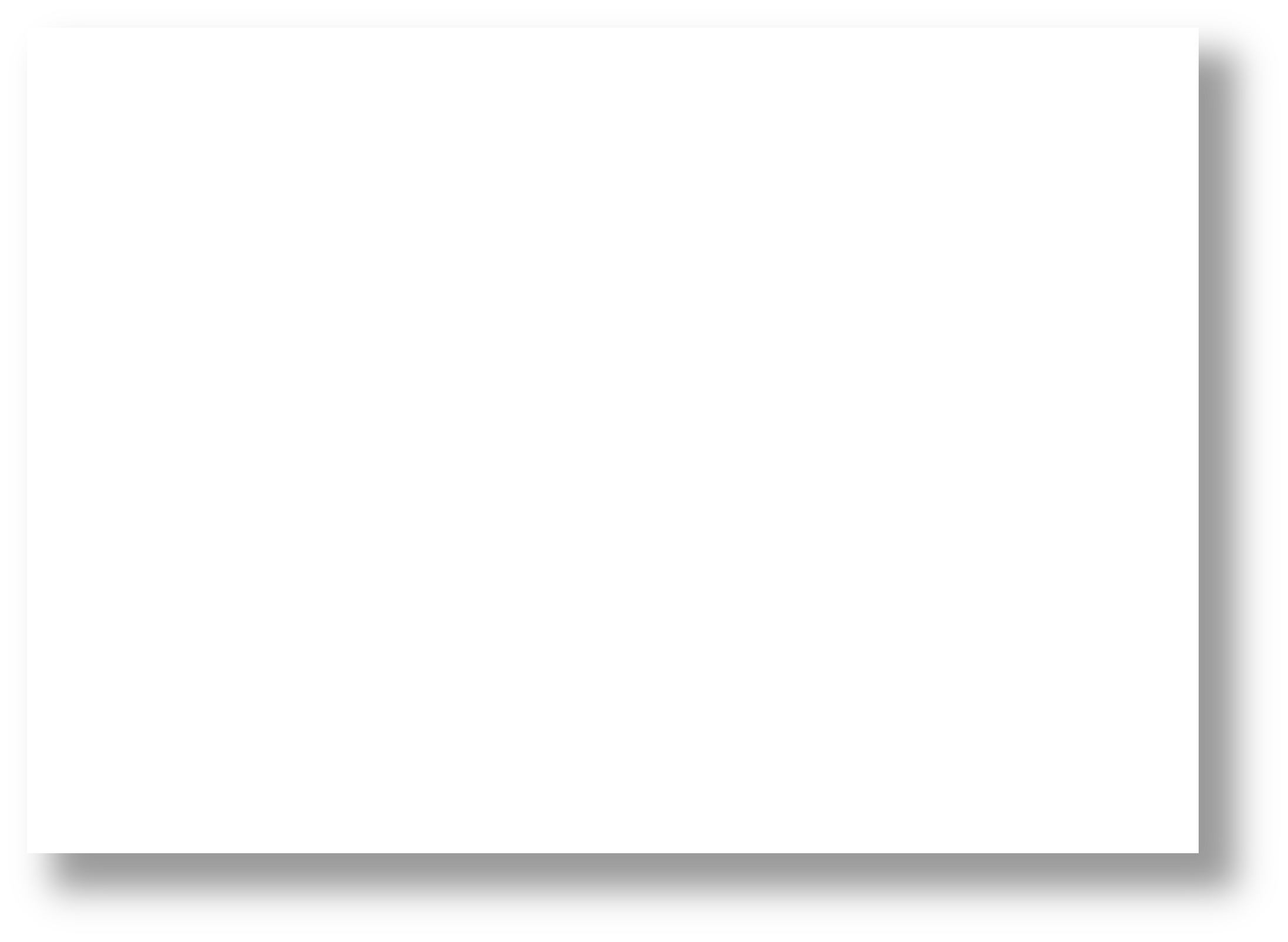
29

31

23

37

41



##### data, **Y**, to provide feedback so Y N 89

##### that your model can learn the N N 56

|  |  |  |  |
| --- | --- | --- | --- |
|  | Y N  N Y | | 71  72 |
| Y | N | 83 |
| 29 | Y | Y | 97 |
| 21 | N | N | 64 |
| 30 | Y | N | 68 |

##### relationship between **X** and **Y**.

Ŷ = 𝑓 𝑿

•

Supervised

* Given some data such that each row corresponds to a distinct

i.i.d. observation

* You may be interested in a particular column (e.g. **Temp**)
* Let’s divide our data and learn how data **X** is related to data **Y**

𝑿

* Assert that:

##### 𝒀 = 𝑓

+ 𝜀

**Age**

22

29

31

23

37

41

29

21

30

**Play**

N Y N Y N Y Y N

Y

**Rainy**

Y N N N Y N Y N

N

**Temp**

91

89

56

71

72

83

97

64

68

* Want a model 𝑓 that is:
  + Supervised
  + Predicts real numbers

(regression model)

• Given some data such that each row corresponds to a distinct

i.i.d. observation

• You may be interested in a particular column (e.g. **Temp**)

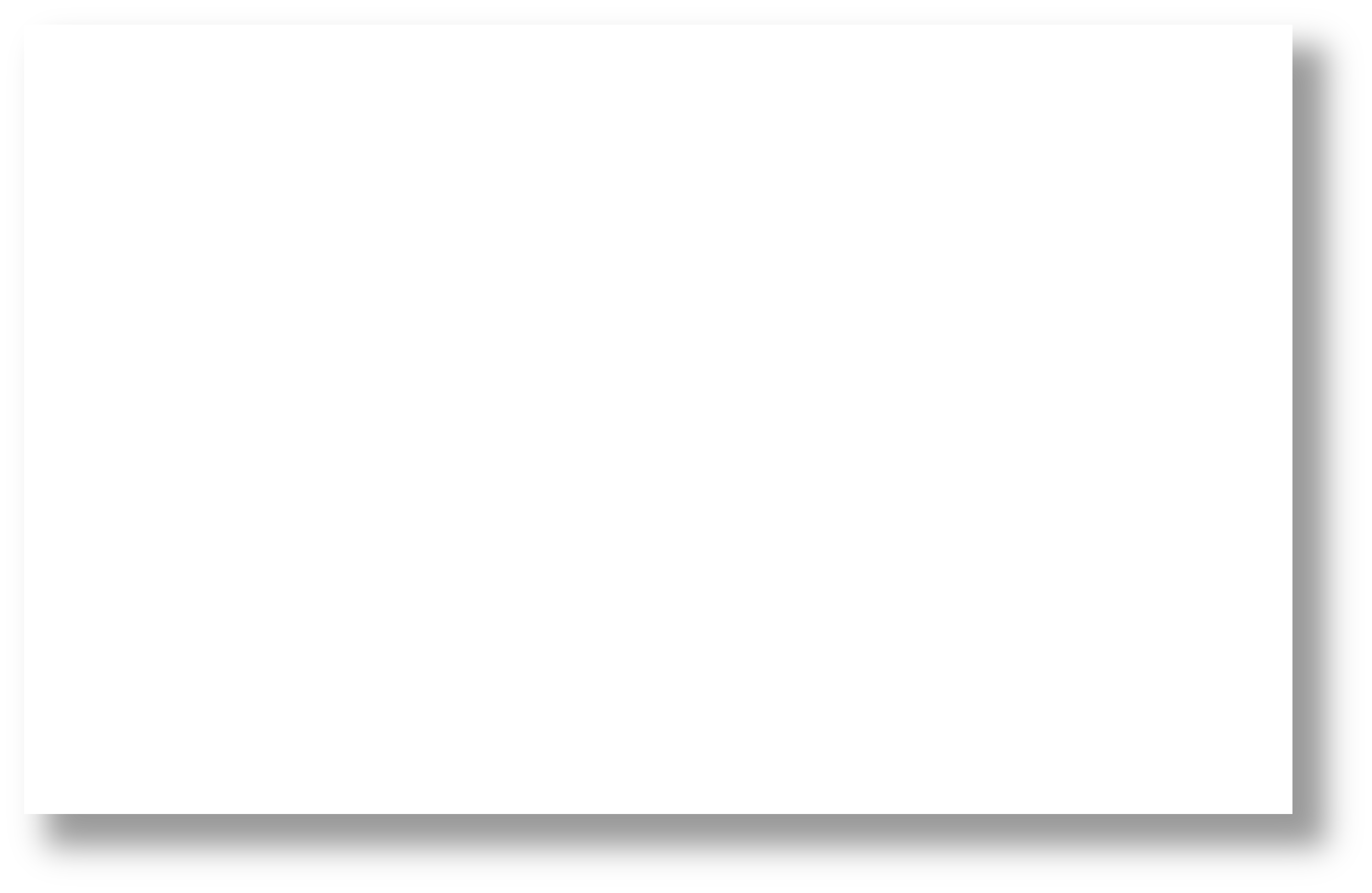
• Let’s divide our data and learn how data **X** is related to data **Y**

• Assert that: 𝒀 = 𝑓 𝑿 + 𝜀

• Want a model that is:

• Supervised

• Predicts real numbers (regression model)



**Temp**

91

89

56

71

72

83

97

64

68

##### 21 N

30 Y

regression model

#### Rainy

Y N N N Y N Y N N

**Age Play**

22 N

29 Y

31 N

23 Y

37 N

41 Y

29 Y

**Def:**

**Regression** models are **supervised**

models, whereby **Y** are *continuous* values.

#### Def:

**Regression** models are **supervised**

models, whereby **Y** are *continuous* values.

**Classification** models are **supervised**

models, whereby **Y** are *categorical* values.

#### Rainy

##### Y N N N Y N Y

**Age Play**

22 N

29 Y

31 N

23 Y

37 N

41 Y

29 Y

21 N N

• Given some data such that each row corresponds to a distinct

i.i.d. observation

• You may be interested in a particular column (e.g. **Temp**)

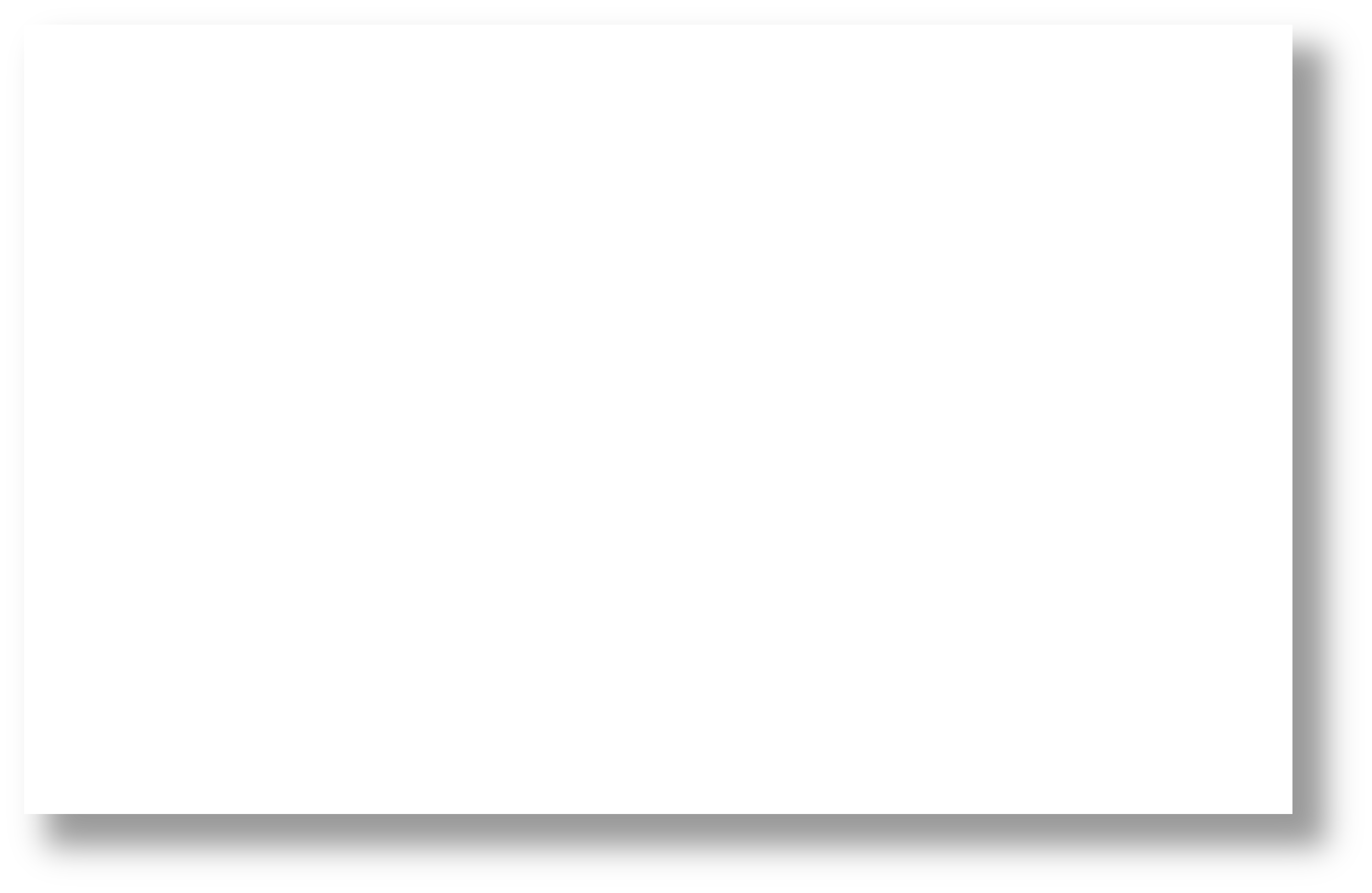
• Let’s divide our data and learn how data **X** is related to data **Y**

• Assert that: 𝒀 = 𝑓 𝑿 + 𝜀

• Want a model that is:

• Supervised

• Predicts real numbers (regression model)



**Temp**

91

89

56

71

72

83

97

64

68

30 Y N

regression model

## Y

𝑓

𝑿

## X

**Y**

91

𝑦

𝑓

𝑿

𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$

## X

22

𝑥1

N

𝑥2

Y

𝑥3

**Y**

91

𝑦

𝑦

𝑓

𝑿

𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$

𝑏$

𝑏1

𝑥1

𝑏2

𝑏

𝑥2

3

𝑥3

𝚺

## X

High-level

Mathematically

22

𝑥1

N

𝑥2

Y

𝑥3

Graphically (NN format)

**Linear Regression**

**Y** 91

𝑦

𝑦 𝚺

𝑓 𝑿 𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$

𝑏$

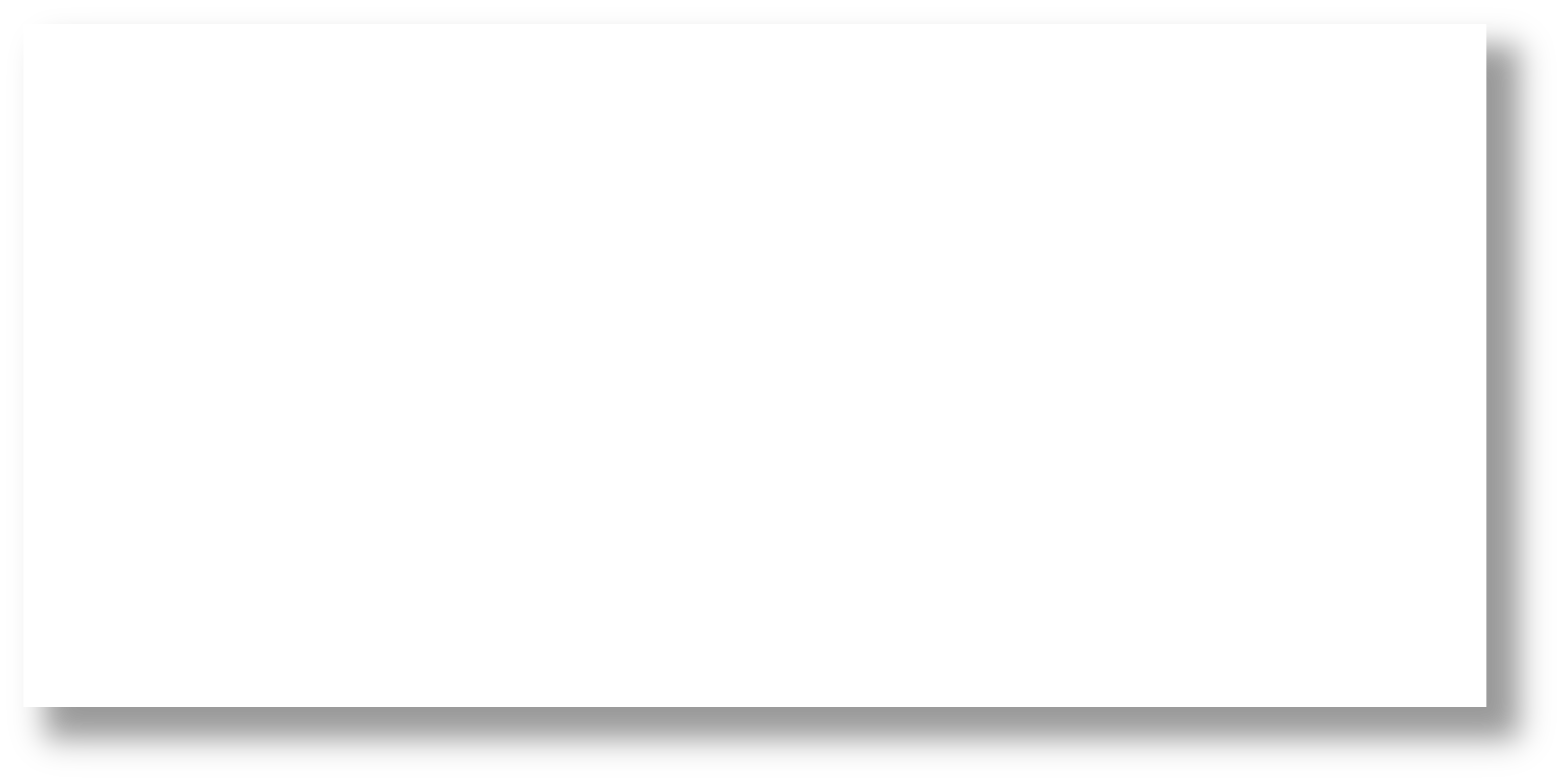
𝑏1 𝑏2 𝑏3

22 N Y 𝑥1 𝑥2 𝑥3

**X** 𝑥1 𝑥2 𝑥3

Graphically

High-level Mathematically (NN format)



**NOTE:**

For convenience, in machine learning we tend to let 𝜽 represent all of our model’s parameters (e.g., 𝜽 = {𝑏%, 𝑏&, 𝑏2, 𝑏(} )

# IMPORTANT

When **training** any supervised 6543TGREPCHECK2&&\*\*(11 model, be mindful of what you select for:

1. Our **loss function** (aka cost function)

Measures how bad our current parameters 𝜽 are

# IMPORTANT

When **training** any supervised model, be mindful of what you select for:

1. Our **loss function** (aka cost function)

Measures how bad our current parameters 𝜽 are

When **testing** our model’s predictions, be mindful of what you select for:

3. Our **evaluation metric**

Determines our model’s performance (e.g., Mean Squared Error (MSE), 𝑅2,

𝐹1 score, etc.)

###### Linear Regression

**Q1** When training our model, how do we

91

𝑦

measure its 𝑚 predictions 𝒚/ ?

𝑚

**A1** Cost function

1

J 𝜽 = 2 +

𝑦,

− 𝑦 2

&'1

𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$

22

𝑥1

N

𝑥2

Y

𝑥3

“Least Squares”

###### Linear Regression

**Q1** When training our model, how do we

91

𝑦

measure its 𝑚 predictions 𝒚/ ?

𝑚

**A1** Cost function

1

J 𝜽 = 2 +

𝑦,

− 𝑦 2

&'1

𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$



22

𝑥1

N

𝑥2

Y

𝑥3

“Least Squares”

How do we find the optimal 𝜽 so that we yield the best predictions?

**Q2**

###### Linear Regression

**Q1** When training our model, how do we

91

𝑦

measure its 𝑚 predictions 𝒚/ ?

𝑚

**A1** Cost function

1

J 𝜽 = 2 +

𝑦,

− 𝑦 2

&'1

𝑦 = 𝑏0 + 𝑏1𝑥1 + 𝑏2𝑥2+ 𝑏$𝑥$

“Least Squares”

How do we find the optimal 𝜽 so that we yield the best predictions?

**Q2**

**A2** Two optimization algorithm options:



22

𝑥1

N

𝑥2

Y

𝑥3

* Gradient Descent (iteratively search)
* Directly (closed-form solution)

𝜽 = 𝑋𝑇𝑋 \*1𝑋𝑇𝑌

Mathematically

###### Linear Regression

**Fitted model example**

The plane is chosen to minimize the sum of the squared vertical distances (per our loss function, least squares) between each observation (red dots) and the plane.

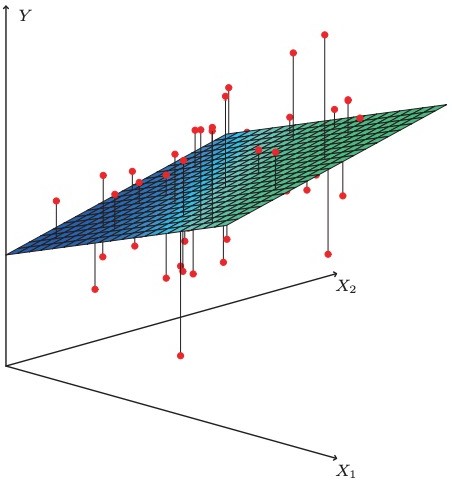


Photo from ”*An Introduction to Statistical Learning*” (James, et al. 2017)

# PROS

**Linear Regression**

# CONS

* **Simple** and **fast** approach to model linear relationships
* Interpretable results via 𝜽

(𝖰 coefficients)

* Can’t model **non-linear**

relationships

* Vulnerable to **outliers**
* Vulnerable to **collinearity**
* Assumes error terms are

###### uncorrelated\*

\* otherwise, we have false feedback during training

**Supervised** vs

**Unsupervised**

**Regression** vs

**Classification**

Linear Regression

**Supervised Regression**

* Parametric models assume our data **X** and **Y** can be represented by

an underlying model 𝒇 (i.e., 𝒀 = 𝒇 + 𝜀) that has a particular form

𝑿

(e.g., a linear relationship, hence our using a linear model)

* Next, we aimed to fit the model 𝒇 by 6543TGREPCHECK2&&\*\*(11 estimating its parameters 𝜽

(we did so in a **supervised** manner)

* Parametric models assume our data **X** and **Y** can be represented by

an underlying model 𝒇 (i.e., 𝒀 = 𝒇 + 𝜀) that has a particular form

𝑿

(e.g., a linear relationship, hence our using a linear model)

* Next, we aimed to fit the model 𝒇 by estimating its parameters 𝜽

(we did so in a **supervised** manner)

* Parametric models make the above assumptions. Namely, that there exists an underlying model 𝒇 that has a fixed number of parameters.

**Supervised** vs

**Unsupervised**

**Regression** vs

**Classification**

**Parametric** vs

**Non-Parametric**

Linear Regression

**Supervised Regression**

**Parametric**

Alternatively, what if we make no assumptions about the underlying model 𝒇 ? Specifically, let’s **not assume** 𝒇 :

* has any particular distribution/shape

(e.g., Gaussian, linear relationship, etc.)

* can be represented by a finite number of parameters.

Alternatively, what if we make no assumptions about the underlying model 𝒇 ? Specifically, let’s **not assume** 𝒇 :

* has any particular distribution/shape

(e.g., Gaussian, linear relationship, etc.)

* can be represented by a finite number of parameters.

This would constitute a non-parametric model.

* Non-parametric models are **allowed to have parameters**; in fact, oftentimes the # of parameters grows as our amount of training data increases
* Since they make no strong assumptions about the form of the function/model, they are free to learn **any functional form** from the training data -- *infinitely complex.*

**k-NN**

**Refresher:**

N

𝑦

Mathematically

22

𝑥1

91

𝑥2

Y

𝑥3

* k-NN doesn’t train a model

𝑦 = 𝜎(𝛽𝑋)

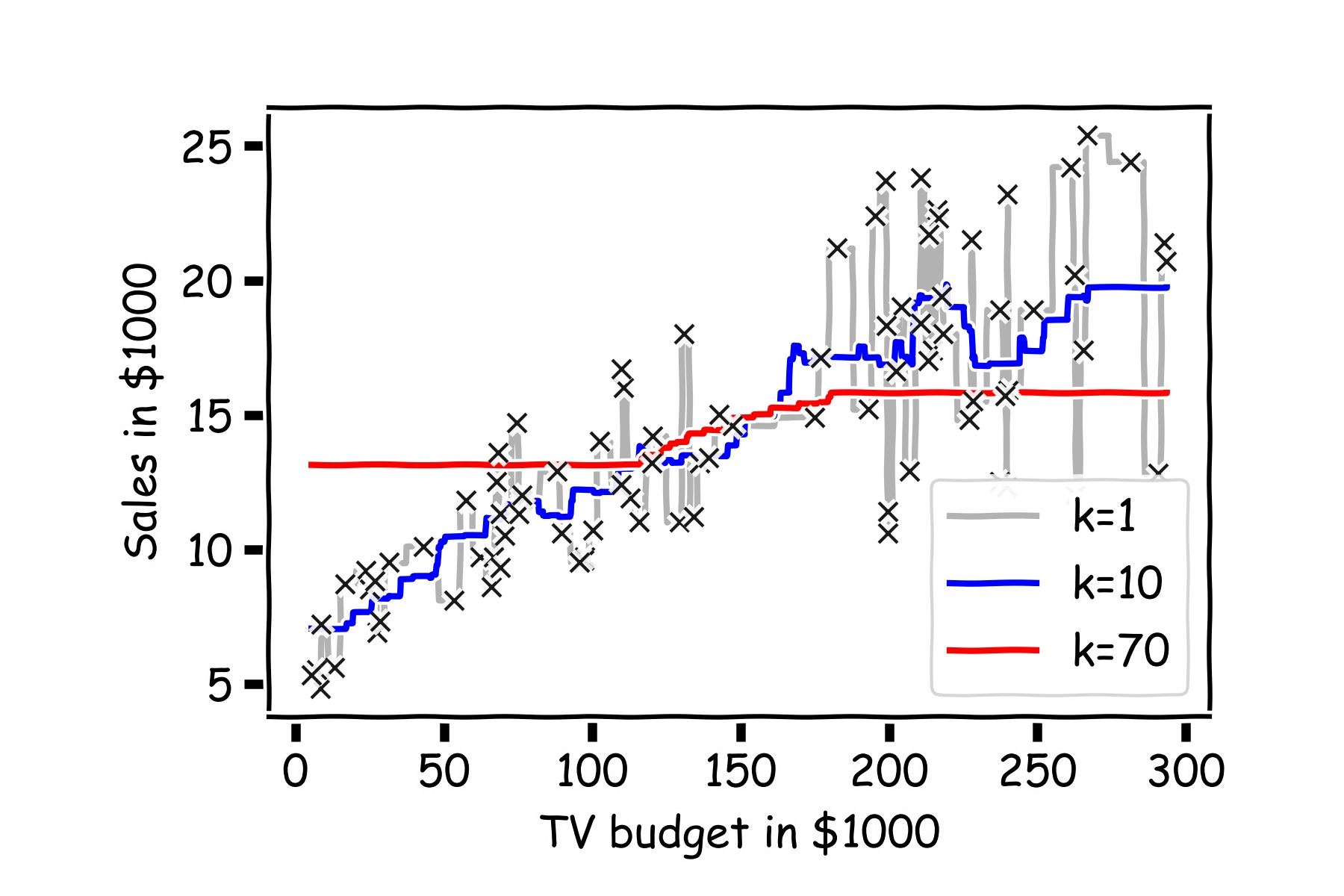
* One merely specifies a 𝒌 value
* At test time, a new piece of data 𝒂:
  + must be compared to all other training data 𝒃, to determine its k-nearest neighbors, per some 6543tgrepcheck2&&\*\*(11 distance metric 𝒅

𝒂, 𝒃

* + is classified as being the majority class (if categorical) or average (if quantitative) of its k-neighbors

**k-NN**

#### Conclusion:

* k-NN makes no assumptions about the data 𝑿 or the form of 𝒇(𝑿)
* k-NN is a **non-parametric model**

# PROS

* **Intuitive** and **simple** approach
* Can model any type of data / places no assumptions on the data
* Fairly robust to missing data
* Good for highly sparse data

(e.g., user data, where the columns are thousands of potential items of interest)

1. **NN**

# CONS

* + Can be very **computationally expensive** if the data is large or high-dimensional
  + Should carefully think about features, including scaling them
  + Mixing quantitative and categorical data can be tricky
  + Interpretation isn’t meaningful
  + Often, regression models are better, especially with little data

**Supervised** vs

**Unsupervised**

**Regression** vs

**Classification**

**Parametric** vs

**Non-Parametric**

Linear Regression

k-NN

**Supervised Regression Parametric Supervised either Non-Parametric**

# IMPORTANT

When **training** any supervised model, be careful of **overfitting** your model A good model should generalize well to unseen (i.e., testing) data

Consider adding **regularization** term 𝑅 to your cost function

𝜃

**Imposes a penalty based on your parameter values** 𝜽

**L1** regularization: 𝑅

𝜃

= ∑𝑛 |𝜃&|

Prefers sparse weights (many 0’s)

**L2** regularization: 𝑅

&'1

= ∑𝑛 𝜃2 Prefers many small-weight values

&'1 &

𝜃

# IMPORTANT

When **training** any supervised model, wisely use your training data A good model should generalize well to unseen (i.e., testing) data

1. Shuffle your training data and optionally bootstrap samples
2. Perform cross-validation

**Q1.** Say our dataset has just 4 features (each row corresponds to a distinct day):

* + - Day of the week, where each value is one of {”M”, “T”, “W”, “Th”, “F”, “Sa”, “Su”}
    - Average temperature outside today (range is from 21F to 102F)
    - Average weather today, where each value is one of {“Sunny”, “Cloudy”, “Rainy”, “Overcast”, “Snowy”}
    - # of people who paid to rent/ride a public city bike (i.e., Boston’s BlueBike) for at least 10 minutes on the given day

We are interested in predicting the last feature by using the previous three. How would you go about modelling this? Data and modelling choices? Discuss w/ your group.

44

**Q2.** Say we want to estimate the quality of our Q1 model. In addition to using MSE, we want to use look at the 𝑅2 value.

Imagine our code includes 2 variables:

* + - target\_y which is a list of numbers that represent the gold truth numbers from the dataset
    - pred\_y which is a list of the numbers that represent our predictions
    - Write 1 line of code to uses sklearn’s built-in function to calculate the 𝑅2 value

45

##### Review of Models Review of Data Debugging

Review of Models

Review of Data

##### Debugging

**Age**

**Play**

**Rainy**

**Temp**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* Let’s say our data has 1,000 rows

1,000

**Age**

**Play**

**Rainy**

**Temp**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**

1,000

**Age**

**Play**

**Rainy**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**

1,000

**Age**

**Play**

**Rainy**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**

###### Which data should we use? All of it?

**TRAINING**

**Age**

**Play**

**Rainy**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**
* **Which data should we use? All of it?**

**TRAINING**

m1 m2 **…** m𝑁

* + Could train 𝑛 unique models on the **training set**

(e.g., adjust the hyperparameters)

* + Evaluate on **training set**

**Age**

**Play**

**Rainy**

22

29

31

23

37

41

29

21

30

45

21

19

44

36

13

22

28

35

52

17

N Y N Y N Y Y N Y N N N Y Y Y N Y N Y

Y

Y N N N Y N Y N N N Y Y Y N Y Y N N Y

Y

**X**

**Y**

* Let’s say our data has 1,000 rows

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

80

83

68

71

55

92

78

* And that we want to predict **Temp**

###### Which data should we use? All of it?

**TRAINING**

* + Models always have a risk of just memorizing/overfitting on the training data
  + Many will likely yield 100% accuracy
  + We need some indication as to how well we’ll do on **unseen, future data**

80

83

68

71

55

92

78

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

N Y Y N N Y

Y

Y Y N Y N Y

Y

36

13

22

28

35

52

17

Y N N N Y N Y N N N Y Y

Y

N Y N Y N Y Y N Y N N N

Y

22

29

31

23

37

41

29

21

30

45

21

19

44

**Rainy**

**Play**

**Age**

**TESTING**

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**

**TRAINING**

###### Which data should we use? Split it in two?

80

83

68

71

55

92

78

**Temp**

91

89

56

71

72

83

97

64

68

61

56

36

53

N Y Y N N Y

Y

Y Y N Y N Y

Y

36

13

22

28

35

52

17

Y N N N Y N Y N N N Y Y

Y

N Y N Y N Y Y N Y N N N

Y

22

29

31

23

37

41

29

21

30

45

21

19

44

**Rainy**

**Play**

**Age**

* Let’s say our data has 1,000 rows
* And that we want to predict **Temp**

**TRAINING**

* **Which data should we use? Split it in two?**

### m1 m2 **…** m𝑁

* + Could train models on the **training set**

**TESTING**

* + Evaluate on the **testing set**
  + Select the model **m**𝐢 that performs best on the **testing set**?

**X**

**Y**

* Let’s say our data has 1,000 rows

80

83

68

71

55

92

78

91

89

56

71

72

83

97

64

68

61

56

36

53

**Temp**

N Y Y N N Y

Y

Y Y N Y N Y

Y

36

13

22

28

35

52

17

Y N N N Y N Y N N N Y Y

Y

N Y N Y N Y Y N Y N N N

Y

22

29

31

23

37

41

29

21

30

45

21

19

44

**Rainy**

**Play**

**Age**

* And that we want to predict **Temp**

**TRAINING**

###### Which data should we use? Split it in two?

* + Selecting our model based on its performance on the **testing set** could yield unrealistic expectations and it’s like cheating, if such performance mattered (e.g., developing new models); it’s like gaming the system

**TESTING**

83

68

71

55

92

78

64

68

61

56

36

53

80

**Temp**

91

89

56

71

72

83

97

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

**TESTING**

* Let’s say our data has 1,000 rows

**TRAINING**

* And that we want to predict **Temp**

###### Which data should we use? Split it into three?

**DEV**

83

68

71

55

92

78

64

68

61

56

36

53

80

**Temp**

91

89

56

71

72

83

97

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

* Let’s say our data has 1,000 rows

**TRAINING**

* And that we want to predict **Temp**
* **Which data should we use? Split it into three?**

**Validati on**

### m1 m2 **…** m𝑁

* + Could train models on the **training set**

**TESTING**

* + Evaluate on the **validation set**
  + Select the model **m**𝐢 that performs best on the **validation set**?

83

68

71

55

92

78

64

68

61

56

36

53

80

**Temp**

91

89

56

71

72

83

97

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

* Let’s say our data has 1,000 rows

**TRAINING**

* And that we want to predict **Temp**
* **Which data should we use? Split it into three?**

**Validati on**

### m1 m2 **…** m𝑁

* + Could train models on the **training set**

**TESTING**

* + Evaluate on the **validation set**
  + Select the model **m**𝐢 that performs best on the **validation set**?



**X**

**Y**

* Let’s say our data has 1,000 rows

83

68

71

55

92

78

64

68

61

56

36

53

80

91

89

56

71

72

83

97

**Temp**

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

**TRAINING**

* And that we want to predict **Temp**

###### Which data should we use? Split it into three?

**Validati on**

* Perfect! This allows us to tweak our models and optimize for the **val** set, with no cheating, for we truly ignore the **test** set until we want a fair, final, one-time evaluation of our best model. **It’s like blindfolding ourselves so we have a fair impression of the goodness of our model.**

**TESTING**



**X**

**Y**

* Let’s say our data has 1,000 rows

64

68

61

56

36

53

80

83

68

71

55

92

78

91

89

56

71

72

83

97

**Temp**

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

**TRAINING**

* And that we want to predict **Temp**

###### Which data should we use? Split it into three?

**Validati on**

* This allows us to fairly evaluate our model.
* Moreover, if we develop novel models, we can fairly compare against others’ systems that run on the same data.

**TESTING**

* The more data we have, the better **X Y**

83

68

71

55

92

78

64

68

61

56

36

53

80

**Temp**

91

89

56

71

72

83

97

Y Y N N Y

Y

Y N Y N Y

Y

13

22

28

35

52

17

N N N Y Y Y

N

N Y N N N Y

Y

21

30

45

21

19

44

36

Y N N N Y N

Y

N Y N Y N Y

Y

22

29

31

23

37

41

29

**Rainy**

**Play**

**Age**

**TRAINING**

* We should always use these splits
* **Training** is typically ~70% of your entire data

**Validati on**

* While this approach is sound, the **val** set is often fairly small. Gives limited view into performance expectations. **Cross-validation** to the rescue!

**TESTING**

* Popular datasets often come with these 3 splits already designated

A machine learning model aims to use **data** to perform a particular **task**

well. Often, the task is to predict or classify new data.



CHOMBOSAN / ALAMY STOCK PHOTO

Self-driving cars Voice assistants Face recognition

If models for the following tasks 6543tgrepcheck2&&\*\*(11 only used a training set, no dev set or test, then all systems would be horrible!



CHOMBOSAN / ALAMY STOCK PHOTO

Self-driving cars Voice assistants Face recognition

Woops, never seen this road before. CRASH!

“*Sorry, I’ve never heard words pronounced like this before*.”

Never seen your face before. Phone locked.

When thinking of dataset splits, I like to think about a real student as our machine learning model.

When thinking of dataset splits, I like to think about a real student as our machine learning model.

Say María is taking **organic chemistry**.

##### Her task is to learn the material, and her evaluation of such will be her final exam.

María’s **training** data is all of the course homework problems. It prepares her. It’s how she *learns*.

##### That data is her best indication of what’s to come for the real- deal final exam (i.e., **testing** set).

If she simply **memorized** (i.e., overfit to) her homework problems over the semester, she would likely be able to achieve 100% on those exact problems (the **training** set).

##### But, this gives no indication as to how she’ll fare on the final exam (the **testing** set).

We always assume the **testing** set is from the same distribution that generated the **training** data. If it’s *identical* to the training (homework), though, that would be bizarre, trivial, and lame.

**ANOTHER ANALOGY X Y**

ans500

ans451 ans452

.

.

.

ans450

ans401 ans402

.

.

.

ans400

ans1 ans2 ans3

.

.

.

Problem451 Problem452

.

Problem500

Problem401 Problem402

.

.

Problem450

Problem1 Problem2 Problem3

**.**

**.**

**.**

Problem400

**TRAINING**

##### It would be way better if she learned the homework (**training**), then used the held-out problems (**val** set) to fine-tune her learning.

**Validati on**

This would help her learn best for the unseen problems that are to come! (**test** set)

**TESTING**

##### Review of Models Review of Data Debugging

Review of Models

##### Review of Data

Debugging

We have recently used two highly popular, useful libraries, **statsmodels** and **sklearn**.

**statsmodels** is mostly focused on the *inference* task. It aims to make good estimates for 𝒇() (via solving for our 𝛽's), and it provides expansive details about its certainty. It provides

lots of tools to discuss confidence, but isn't great at dealing with test sets.

**sklearn** is mostly focused on

the *prediction* task. It aims to make a well-fit line to our input data 𝑿, so as to make good 𝒀 predictions for some unseen inputs 𝑿. It provides a shallower analysis of our variables. In other words, **sklearn** is great at test sets and dev sets, but it can't really reveal much about the uncertainty in the parameters or predictions.

#### REMINDER

**Inference**: estimates the function 𝒇, but the goal isn't to make predictions for 𝒀; rather, it is more concerned with understanding the relationship between 𝑿 and 𝒀.

**Prediction**: estimates the function 𝒇 with the goal of making accurate 𝒀 predictions for some unseen 𝑿.

* + Programming takes practice. No way around it. Practice, practice, practice.
  + We teach the **fundamentals** of data science and machine learning. Library packages, by themselves, are not fundamentals; they are accessible, high-level tools to accomplish certain tasks. They also change from time-to-time. Thus, it would be horrible if the course’s main focus was on functions provided by **pandas**, **NumPy**, **scikits-learn**, etc.
  + Via examples, we aim to show the benefits and power of them.
  + They are incredibly powerful tools, and we make heavy usage of them, but we rely on you to explore how to make them work for you.
* **Debugging** is a large topic**,** and yet it’s only a small aspect of the larger field of **software testing**

##### Here are some of my tips:

* Pay close attention to the error message (read the output)

##### When coding, start simple! Stop trying to do too much at once.

* **print**() things out. Often. Check the **type**, **size**, and **contents** of the data structure at hand.

##### Be suspicious and doubtful of everything you type. The default is for things to not work. By default, your code will not work.

* + Pay attention to scope and leaky variables
  + In Jupyter Notebooks, pay careful attention to the order in which you run cells
  + Writing on paper to draft ideas and debug can be highly useful.
  + Get a proper debugger (e.g., with Visual Studio code editor).
  + Everything can be distilled down to data structures and algorithms..
  + Code can have:
    - Correct logic, but incorrect implementation
    - Incorrect logic, but correct implementation of your bad logic
    - Incorrect logic, and incorrect implementation
    - Correct logic, correct implementation.
  + If there’s an error, it’s either in your **logic** or **implementation**, or both

**Exercise time!**

80