## Deadlines

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Date of assignment</th>
<th>Deadline (midnight 23:59)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW1</td>
<td>Sep 6</td>
<td>Sep 19</td>
</tr>
<tr>
<td>HW2</td>
<td>Sep 20</td>
<td>Oct 3</td>
</tr>
<tr>
<td>HW3</td>
<td>Oct 4</td>
<td>Oct 17</td>
</tr>
<tr>
<td>Paper summary</td>
<td>Oct 11</td>
<td>Oct 31</td>
</tr>
<tr>
<td>HW4</td>
<td>Oct 18</td>
<td>Oct 31</td>
</tr>
<tr>
<td>HW5</td>
<td>Nov 1</td>
<td>Nov 14</td>
</tr>
<tr>
<td>HW6</td>
<td>Nov 22</td>
<td>Dec 5</td>
</tr>
<tr>
<td>Project</td>
<td>Oct 12</td>
<td>Nov 15 - 17</td>
</tr>
</tbody>
</table>

*All deadlines are subject to change, check out CampusWire and website for updates*
“In previous episodes...”
There are two main ways of looking at this (A and B):

**way A:** something wrong with a model

**way B:** something wrong with a data

**Epoch #5**

![Graph showing loss over epochs with overfitting indicated](image)
Regularisation methods

Way A methods

Way B methods
Way A methods
Explicit regularisation methods
(Dropout, L1/L2 regularisation)
Despite some differences, these methods belong to so called **explicit regularisation methods**.

They **explicitly limit** predictive power of the model by introducing additional constraints.

**L2/L1** regularisation  **VS**  **Dropout**
Way B methods
Implicit regularisation methods
These are the most basic transforms
More advanced things are available via github/aleju/imgaug
Explicit regularisation methods
(Dropout, L1/L2 regularisation)

Implicit regularisation methods
There are plenty of other cool ways to get **better performance** and **fight overfitting**!
There are plenty of other cool ways to get better performance and fight overfitting!

Weak learners, ENSEMBLE!
Ensemble Learning
In 1907, 787 villagers tried to guess the weight of an ox. None of them guessed it correctly, but the average guess (542.9 kg) was very close to the actual weight of the ox (543.4 kg).

James Surowiecki, *The Wisdom of Crowds*
Multiple models are built on training data

Linear Regression

\[
y = 1.6 + 0.79x
\]
Multiple models are built on training data

Ridge Regression

Linear Regression

\[ y = 1.6 + 0.79x \]

\[ y = 1.94 + 0.64x \]

Test

Multiple models are built on training data
Multiple models are built on training data.
Multiple models are built on training data
Multiple models are built on training data
Multiple models are built on training data.
Multiple models are built on training data
Multiple models are built on training data

- **Ridge Regression**
- **Linear Regression**
- **Lasso Regression**

Multiple models are built on training data.
Multiple models are built on training data.

- **Ridge Regression**
  - Equation: $1.6 + 0.79x$
  - Graph: Red line

- **Linear Regression**
  - Equation: $3.28 + 0.14x$
  - Graph: Blue line

- **Lasso Regression**
  - Equation: $1.94 + 0.64x$
  - Graph: Green line
Multiple models are built on training data

\[
\text{Ridge Regression: } 1.6 + 0.79^*x + 1.94 + 0.64^*x + 3.28 + 0.14^*x
\]

Linear Regression

Lasso Regression
Multiple models are built on training data

\[ y = 1.6 + 0.79x \]

\[ y = 3.28 + 0.14x \]

\[ y = 1.94 + 0.64x \]

\[ y = 3.28 + 0.14x \]

Multiple models include:
- Linear Regression
- Ridge Regression
- Lasso Regression

The diagrams show the relationships between the input \( x \) and output \( y \) for different regression models.
Multiple models are built on training data

\[ y = 1.6 + 0.79x \]

\[ y = 1.94 + 0.64x \]

\[ y = 3.28 + 0.14x \]

Multiple models are built on training data:
- Linear Regression
- Ridge Regression
- Lasso Regression

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ X_1 = 2 \quad X_2 = 5 \]
Multiple models are built on training data

Linear Regression

Ridge Regression

Lasso Regression

\[ 1.6 + 0.79^\times \]

\[ 3.28 + 0.14^\times \]

\[ 1.94 + 0.64^\times \]

\[ X_1 = 2 \quad X_2 = 5 \]

LR

\[ \begin{array}{c|c}
1 & 2 \\
\hline
\end{array} \]
Multiple models are built on training data.

For $X_1 = 2$ and $X_2 = 5$:

- Linear Regression: $1.6 + 0.79 \times 2 + 0.14 \times 2$
- Ridge Regression: $1.94 + 0.64 \times 2$
- Lasso Regression: $3.28 + 0.14 \times 2$

The table shows:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multiple models are built on training data

\[ y = 1.6 + 0.79x + 3.28 + 0.14x \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
</tbody>
</table>

\[ X_1 = 2 \quad X_2 = 5 \]
Multiple models are built on training data

\[
\text{Linear Regression: } 1.6 + 0.79^*x \\
\text{Ridge Regression: } 1.94 + 0.64^*x \\
\text{Lasso Regression: } 3.28 + 0.14^*x
\]

\[
\begin{array}{c|c|c}
\text{Model} & 1 & 2 \\
\hline
\text{LR} & 3.18 & 5.55 \\
\text{Ridge} & 3.23 & 5.17 \\
\text{Lasso} & 3.57 & 4 \\
\end{array}
\]

\[X_1 = 2 \quad X_2 = 5\]
Multiple models are built on training data

- **Ridge Regression**
- **Linear Regression**
- **Lasso Regression**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average the predictions

$$X_1 = 2 \quad X_2 = 5$$
Multiple models are built on training data

$$y = 1.6 + 0.79 \times x + 1.94 + 0.64 \times x + 3.28 + 0.14 \times x$$

X₁ = 2  X₂ = 5

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

mean(3.18, 3.23, 3.57)

Average the predictions
Multiple models are built on training data.

$y = 3.28 + 0.14x$

$y = 1.94 + 0.64x$

$y = 1.6 + 0.79x$

$X_1 = 2$  $X_2 = 5$

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td></td>
</tr>
</tbody>
</table>

$\text{mean}(3.18, 3.23, 3.57)$

Average the predictions
Multiple models are built on training data.

- **Ridge Regression**
- **Linear Regression**
- **Lasso Regression**

An example of predictions:

- **LR**
  - Predictions: 3.18, 5.55
- **Ridge**
  - Predictions: 3.23, 5.17
- **Lasso**
  - Predictions: 3.57, 4

Mean predictions:

$$\text{mean}(5.55, 5.17, 4)$$

Average the predictions.
Multiple models are built on training data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
</tbody>
</table>

$\text{mean}(5.55, 5.17, 4)$

**Average the predictions**
Multiple models are built on training data

- **Ridge Regression**
  - Equation: \( 1.94 + 0.64x \)
  - Prediction for \( X_1 = 2, X_2 = 5 \):
    - LR: 3.18
    - Ridge: 3.23
    - Lasso: 3.57
    - Average: 3.32

- **Linear Regression**
  - Equation: \( 1.6 + 0.79x \)
  - Prediction for \( X_1 = 2, X_2 = 5 \):
    - LR: 5.55
    - Ridge: 5.17
    - Lasso: 4
    - Average: 4.89

- **Lasso Regression**
  - Equation: \( 3.28 + 0.14x \)
  - Prediction for \( X_1 = 2, X_2 = 5 \):
    - LR: 3.18
    - Ridge: 3.23
    - Lasso: 3.57
    - Average: 3.32

Average the predictions
Multiple models are built on training data

\[ y = 1.6 + 0.79x \]
\[ y = 1.94 + 0.64x \]
\[ y = 3.28 + 0.14x \]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
<tr>
<td>TRUE</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
Multiple models are built on training data

**Ridge Regression**

- Equation: \( 1.94 + 0.64x \)
- Data points:
  - [1, 2]
  - [2, 3]

**Linear Regression**

- Equation: \( 3.28 + 0.14x \)
- Data points:
  - [1, 2]
  - [2, 3]

**Lasso Regression**

- Equation: \( 1.6 + 0.79x \)
- Data points:
  - [1, 2]
  - [2, 3]

**Basic ensemble**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
<tr>
<td>TRUE</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

\( X_1 = 2 \quad X_2 = 5 \)
Multiple models are built on training data.

Similar idea can be applied for classification.

Basic ensemble

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
<tr>
<td>TRUE</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
How do we know who of these people will be successful in running the country?
Multiple models are built on training data

Training data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4.00</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
<tr>
<td>TRUE</td>
<td>4.00</td>
<td>5.00</td>
</tr>
</tbody>
</table>

Basic ensemble

Equations:
- $1.94 + 0.64x$
- $1.6 + 0.79x$
- $3.28 + 0.14x$
Multiple models are built on training data

Training data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.18</td>
<td>5.55</td>
</tr>
<tr>
<td>Ridge</td>
<td>3.23</td>
<td>5.17</td>
</tr>
<tr>
<td>Lasso</td>
<td>3.57</td>
<td>4.00</td>
</tr>
<tr>
<td>Average</td>
<td>3.32</td>
<td>4.89</td>
</tr>
<tr>
<td>TRUE</td>
<td>4.00</td>
<td>5.00</td>
</tr>
</tbody>
</table>

Basic ensemble
Multiple models are built on training data

Training data

Model 1
Model 2
Model 3
Ensemble
TRUE  Dog

Basic ensemble
Multiple models are built on training data

Training data

Test

<table>
<thead>
<tr>
<th>Model 1</th>
<th>Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 2</td>
<td></td>
</tr>
<tr>
<td>Model 3</td>
<td></td>
</tr>
<tr>
<td>Ensemble</td>
<td></td>
</tr>
<tr>
<td>TRUE</td>
<td>Dog</td>
</tr>
</tbody>
</table>

Basic ensemble
Multiple models are built on training data.

Training data

<table>
<thead>
<tr>
<th>Test</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Ensemble</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dog</td>
<td>Dog</td>
<td>Cat</td>
<td>Dog</td>
</tr>
</tbody>
</table>

Dog - 2, Cat - 1

Majority voting
Multiple models are built on training data

Training data

<table>
<thead>
<tr>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
</tr>
<tr>
<td>Model 2</td>
</tr>
<tr>
<td>Model 3</td>
</tr>
<tr>
<td>Ensemble</td>
</tr>
</tbody>
</table>

Dog wins!

Majority voting
Training data

<table>
<thead>
<tr>
<th>Model 1</th>
<th>Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 2</td>
<td>Dog</td>
</tr>
<tr>
<td>Model 3</td>
<td>Cat</td>
</tr>
<tr>
<td>Ensemble</td>
<td>Dog</td>
</tr>
<tr>
<td><strong>TRUE</strong></td>
<td><strong>Dog</strong></td>
</tr>
</tbody>
</table>

**Majority voting**
Training data
Majority voting implies equal weight of each model’s vote
Majority voting implies equal weight of each model’s vote.
Are these models actually **equally useful**?

Training data

Majority voting implies **equal weight** of each model’s vote
Are these models actually equally useful?

Let’s assume that we trust one model more than the other model.
Are these models actually equally useful?

Let’s assume that we trust **one model** more than the **other model**.

What does it mean?
Training data
Training data

Representatives of each model
Training data

Ensemble parlement

Representatives of each model
Training data

Ensemble parlement

Representatives of each model

33 sits
Training data

Ensemble parliament

Representatives of each model

0.33

0.37

0.3

37 sits
Training data

Ensemble parlement

Representatives of each model

30 sits
Training data

Ensemble parlement

Total (100 sits)

Representatives of each model
Training data

Representatives of each model

Ensemble parlement

M1
M2
M3
M1
M2
M3
Total (100 sits)
Training data

Ensemble parlement

Total (100 sits)
Each party (model) votes for the class as one team.
Previously each model was given only one vote.

Now each model has different number of votes.
In this case, the outcome (dog class wins) remains the same.
Things change if there would be 3 classes (dog, cat, rabbit) instead of 2.

Previously each model was given only one vote.

Now each model has different number of votes.
Things change if there would be **3 classes** (dog, cat, rabbit) instead of **2**

- **Basic Ensemble**
  - M1 for Rabbit
  - M2 for Dog
  - M3 for Cat

- **Weighted Ensemble**
  - 33 for Rabbit
  - 37 for Dog
  - 30 for Cat

Previously each model was given only **one** vote.
Now each model has different number of votes.
Previously each model was given only one vote.

Now each model has different number of votes.

M1 for Rabbit
M2 for Dog
M3 for Cat

Training data
Relative weight

Ensemble parlement
Dog
Cat
Total (100 sits)

Things change if there would be 3 classes (dog, cat, rabbit) instead of 2.

Basic Ensemble

Weighted Ensemble

Draw (random choice)

Dog class

33 for Rabbit
37 for Dog
30 for Cat

(37 > 33 & 37 > 30)
If you know that your models are not equally useful, **how do you choose weights?**
We can estimate the weight of each model based on \textbf{CV} on training data.

If you know that your models are not equally useful, \textbf{how do you choose weights?}
We can estimate the weight of each model based on CV on training data.
We can estimate the weight of each model based on CV on training data
We can estimate the weight of each model based on CV on training data.
We can estimate the weight of each model based on CV on training data.

Let's choose some realistic values.
We can estimate the weight of each model based on CV on training data.

Let's choose some realistic values:

- 0.75
- 0.70
- 0.65
We can estimate the weight of each model based on CV on training data.
We can estimate the weight of each model based on CV on training data.

The weight of the model is proportional to its average score.
We can estimate the weight of each model based on CV on training data.

The weight of the model is proportional to its **average score**.

Average score: \( \frac{0.75 + 0.7 + 0.65}{3} \)
We can estimate the weight of each model based on CV on training data.

The weight of the model is proportional to its **average score**.
We can estimate the weight of each model based on CV on training data.
We can estimate the weight of each model based on CV on training data.
We can estimate the weight of each model based on CV on training data.
Ensembles tend to yield **better results** when there is a **higher diversity** among the models

(similar reasoning is applied to enabling diversity in human teams)
How to achieve **diversity** of models with **only one type** of algorithm (e.g. decision tree)?
How to achieve **diversity** of models with **only one type** of algorithm (e.g. decision tree)?

Training data

We can **play around** with the **data**!
Training data
Training data

Random
70%
Training data

Random 70%
Training data

Random 70%
Training data

Bootstrapped data

Random 70%
Bootstrapping

Bootstrapped data

Training data

Random 70%
Bootstrapping

Training data → Bootstrapped data

Random 70%
Training data

Bootstrapped data

Training on different parts of data produces diverse models
Bootstrapping

Training data

Bootstrapped data

Random 70%
Bootstrapping

Training data

Bootstrapped data

Random 70%

Validation data
Bootstrapping

Training data → Bootstrapped data

Random 70%

Validation data
Bootstrapping

Training data

Validation data

Bootstrapped data

Random 70%

Good old majority vote

Dog

Cat
Bootstrapping

Training data

Random 70%

Bootstrapped data

Validation data

Dog

Cat

Dog
Bootstrapping

Training data

Validation data

Bootstrapped data

Random 70%

Aggregation (majority vote or averaging)

Dog

Cat

Dog
Bootstrapping + Aggregation = Bagging
Bootstrapping + Aggregation = Bagging

Training data

Random 70%

Validation data

Bootstrapped data

Validation data

Bootstrapped data

Bootstrapped data

B1

B2

B3
Decision Tree Algorithm

By asking a simple **question** about value of **independent variable** it tries to predict a value of **dependent variable**

Is distance > X

- False
  - fare amount = Y
- True
  - fare amount = Z
Decision Tree Algorithm

By asking a simple **question** about value of $X_1$ and $X_2$ it tries to predict a class $(1, 0)$.
Decision Tree Algorithm

By asking a simple **question** about value of $X_1$ and $X_2$ it tries to predict a **class** $(1, 0)$
Decision Tree Algorithm

By asking a simple question about value of $X_1$ and $X_2$ it tries to predict a class $(1, 0)$
Decision Tree Algorithm

By asking a simple **question** about value of $X_1$ and $X_2$ it tries to predict a class $(1, 0)$
Let's apply **bagging** to learn **three different decision trees**.
Building the **first** decision tree based on data in the **bag #1**
Building the **first** decision tree based on data in the **bag #1**

What we need to **do** in order to build a decision tree?
Building the **first** decision tree based on data in the **bag #1**

1. Need to **evaluate** all possible splits
Building the **first** decision tree based on data in the **bag #1**

1. Need to **evaluate** all possible splits

What are the **all possible splits**
Building the **first** decision tree based on data in the **bag #1**

1. Need to **evaluate** all possible splits

What are the **all possible splits**
Building the **first** decision tree based on data in the **bag #1**

1. Need to **evaluate** all possible splits

What are the **all possible splits**

Are these **are only possible splits**?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

Are these \textit{are only possible splits}?

\(X_1\) and \(X_2\) are both features that we can use to divide data
Building the first decision tree based on data in the bag #1

$X_1$ and $X_2$ are both features that we can use to divide data.

Are these are only possible splits?

1. Need to evaluate all possible splits.

What are the all possible splits?
Building the **first** decision tree based on data in the bag #1

1. Need to evaluate all possible splits

What are the all possible splits

Are **all possible splits** are also **reasonable** splits?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

What are the all possible splits

These splits are not reasonable (data is not divided)

Are all possible splits are also reasonable splits?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

What are the all possible splits
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits.

What are the **all possible splits** (but also **reasonable**)?
Building the **first** decision tree based on data in the bag #1

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits

What are the **all possible splits** (but also **reasonable**)?
Building the **first** decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits
   - What are the all possible splits (but also reasonable)?

We can compute **accuracy** of the split
Building the **first** decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits

What are the all possible splits (but also reasonable)?

We can compute accuracy of the split
Building the first decision tree based on data in the bag #1

**Accuracy** = (# of guessed correctly)/(# of all)

We can compute **accuracy** of the split

---

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits

What are the **all possible splits** (but also **reasonable**)?
Building the **first** decision tree based on data in the bag #1

**Accuracy** = \((\text{# of guessed correctly})/(\text{# of all})\)

How do we evaluate splits?

1. Need to evaluate all possible splits

What are the all possible splits (but also reasonable)?

We can compute **accuracy** of the split
Building the **first** decision tree based on data in the bag #1.

Accuracy = \( \frac{3}{\text{(# of all)}} \)

1. Need to evaluate all possible splits.

How do we evaluate splits?

What are the all possible splits (but also reasonable)?

We can compute **accuracy** of the split.
Building the first decision tree based on data in the bag #1

We can compute accuracy of the split

Accuracy = \( \frac{3}{4} \)

How do we evaluate splits?

1. Need to evaluate all possible splits

What are the all possible splits (but also reasonable)?
Building the first decision tree based on data in the bag #1.

We can compute accuracy of the split

\[ \text{Accuracy} = 75\% \]

1. Need to evaluate all possible splits

How do we evaluate splits?

What are the all possible splits (but also reasonable)?

We can compute accuracy of the split.
Building the first decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits (but also reasonable)?

We can compute accuracy of the split
This split is double sided
This split is **double sided**
This split is **double sided**

What is the **accuracy** of the mirrored split?
This split is **double sided**

What is the **accuracy** of the mirrored split?
This split is **double sided**

Let’s differentiate them by the **arrow**
Building the first decision tree based on data in the bag #1

We can compute **accuracy** of the split

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits (but also **reasonable**)?

What are the all possible splits (but also **reasonable**)?

We can compute **accuracy** of the split
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits
   (but also **reasonable**)?

How do we **evaluate** splits?

We can compute **accuracy** of the split.
Building the first decision tree based on data in the bag #1

We can compute accuracy of the split

It seems a lot more splits than we thought at first…

1. Need to evaluate all possible splits (but also reasonable)?

How do we evaluate splits?

We can compute accuracy of the split
1. Need to evaluate all possible splits

How do we evaluate splits?

What are the all possible splits (but also reasonable)?

Building the first decision tree based on data in the bag #1

We can compute accuracy of the split

It seems a lot more splits than we thought at first…

Worry not, accuracy of the split defines the accuracy of the reverse split

25% 2
25% 3
25% 4
25% 5

75% 1
75% 2
75% 3
75% 4
75% 5

X1

X2
Building the first decision tree based on data in the bag #1

We can compute accuracy of the split

1. Need to evaluate all possible splits (but also reasonable)

How do we evaluate splits?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits (but also reasonable)?

How do we evaluate splits?

We can compute accuracy of the split.
Building the first decision tree based on data in the bag #1.

How do we evaluate splits?

1. Need to evaluate all possible splits (but also reasonable)?

We can compute accuracy of the split.
Building the first decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits (but also reasonable)?

We can compute accuracy of the split
Building the **first** decision tree based on data in the bag #1

We can compute **accuracy** of the split.

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits.

What are the **all possible splits** (but also **reasonable**)?
Building the first decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits (but also reasonable)?

We can compute accuracy of the split
Building the **first** decision tree based on data in the bag #1

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits (but also reasonable)?

We can compute **accuracy** of the split
Building the **first** decision tree based on data in the bag #1

We can compute **accuracy** of the split

1. Need to evaluate all possible splits (but also reasonable)

How do we **evaluate** splits?
Building the **first** decision tree based on data in the
bag #1

We can compute **accuracy** of the split

1. Need to **evaluate** all possible splits (but also **reasonable**)?

How do we **evaluate** splits?
Building the first decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits
   
   What are the all possible splits (but also reasonable)?

We can compute accuracy of the split
Building the **first** decision tree based on data in the bag #1

We can compute **accuracy** of the split

1. Need to **evaluate** all possible splits (but also **reasonable**)

How do we **evaluate** splits?
Building the first decision tree based on data in the bag #1

How do we evaluate splits?

1. Need to evaluate all possible splits

What are the all possible splits (but also reasonable)?

We can compute accuracy of the split.
Building the **first** decision tree based on data in the bag #1

How do we **evaluate** splits?

1. Need to **evaluate** all possible splits (but also **reasonable**)?

We can compute **accuracy** of the split
1. Need to evaluate all possible splits.

How do we evaluate splits?

Building the first decision tree based on data in the bag #1.

What are the all possible splits (but also reasonable)?

How do we evaluate splits?
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

2. Choose the best split
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits

2. Choose the **best** split
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits

2. Choose the **best** split

**Which splits** are we interested in?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

2. Choose the best split

Which splits are we interested in?
Building the first decision tree based on data in the bag #1

1. Need to evaluate all possible splits

2. Choose the best split

Which splits are we interested in?
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits

2. Choose the **best** split

Which splits are we interested in?
Building the **first** decision tree based on data in the bag #1

1. Need to **evaluate** all possible splits
2. Choose the **best** split
Building the first decision tree based on data in the bag #1

Here is our tree so far:

1. $X_1 > 4.5$
   - False
     - Class = 1
   - True
     - Class = 0
Are we **done** with a tree for **bag #1**?

Here is our tree so far:

- **$X_1 > 4.5$**
  - False: **Class = 1**
  - True: **Class = 0**
Are we **done** with a tree for **bag #1**?

Here is our tree so far:

- **X₁ > 4.5**
  - False
    - **Class = 1**
    - 66.6% correct
  - True
    - **Class = 0**
    - 100% correct
Are we **done** with a tree for bag #1?

Here is our tree so far:

- **X₁ > 4.5**
  - False
  - Class = 1 (66.6% correct)
  - True
  - Class = 0 (100% correct)

Can we **improve** accuracy of this leaf?
Let’s add another layer
Let’s add another layer

1. Need to **evaluate** all possible splits

2. Choose the **best** split
Let’s add another layer

1. Need to **evaluate** all possible splits

2. Choose the **best** split
Let's add another layer

1. Need to **evaluate** all possible splits

2. Choose the **best** split
Let’s add another layer

1. Need to evaluate all possible splits

2. Choose the best split
Let's add another layer

1. Need to evaluate all possible splits

2. Choose the **best** split
Let’s add another layer

1. Need to evaluate all possible splits

2. Choose the best split
The final decision tree for bag #1
The final decision tree for **bag #1**

- **Class = 0**
- **Class = 1**

**Decision Tree Diagram:**
- **Root Node:** $X_1 > 4.5$
  - **False**
    - **Node:** $X_1 > 2.5$
      - **False**
        - **Node:** Class = 0
      - **True**
        - **Node:** Class = 0
  - **True**
    - **Node:** Class = 1

**Data Points:**
- (1, 1)  →  Class = 0
- (2, 2)  →  Class = 0
- (3, 4)  →  Class = 0
- (4, 5)  →  Class = 0
- (5, 6)  →  Class = 1

**Class Distribution:**
- **Class = 0**  66%
- **Class = 1**  34%
As we want to build three trees we need three independent bootstraps.
As we want to build three trees we need three independent bootstraps.
As we want to build three trees we need three independent bootstrapsts.
As we want to build three trees we need three independent bootstrapts.
As we want to build three trees we need three independent bootstrapts.
As we want to build three trees we need three independent bootstrapts.
As we want to build three trees we need three independent *bootstraps*.
As we want to build three trees we need three independent bootstraps.
As we want to build three trees we need three independent bootstraps.
As we want to build three trees we need three independent bootstrapst.
We can **build** the ensemble on the original data by **combining** trees.
We can build the ensemble on the original data by combining trees.
We can **build** the **ensemble** on the original data by **combining** trees.
We can **build** the **ensemble** on the original data by **combining** trees.
We can build the ensemble on the original data by combining trees.
We can **build** the **ensemble** on the original data by **combining** trees.
Perform the **majority vote** per segment to get the final decision boundaries.
Perform the **majority vote** per segment to get the final decision boundaries.
3 red vs 0 blue
3 red vs 0 blue
Perform the **majority vote** per segment to get the final decision boundaries.
Perform the **majority vote** per segment to get the final decision boundaries.
Bag of decision trees

B1

B2

B3
Let’s add a very simple **twist** to this algorithm.
As last time we shall start with generating bootstraps.
As last time we shall start with generating bootstraps
Something will change with respect to **how we build trees**
Familiar algorithm of building tree had **2 steps**:
Familiar algorithm of building tree had **2 steps**:

1. Need to evaluate all possible splits

2. Choose the **best split**
Here we add **a new step**:

1. Need to evaluate *all possible splits*

2. Choose the **best** split
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split

Here we have only 2 features ($X_1$ and $X_2$), so we will choose one, but normally you would keep about 80% of the original features.
Here we add a new step:

0. Choose a **random set** of features

1. Need to evaluate all possible splits

2. Choose the **best** split

---

![Chart](chart.png)

**Bag #1**

- **Heads**, means we keep $X_2$
- **Tail**, means we keep $X_1$
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split

Bag #1

We keep $X_2$
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split

We keep $X_2$ and get rid of $X_1$
Here we add a new step:

0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split

We keep $X_2$ and get rid of $X_1$
0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
0. Choose a **random set** of features

1. Need to evaluate all possible splits

2. Choose the **best** split

Thanks to **step 0** we have only one option left…
0. Choose a random set of features

1. Need to evaluate all possible splits

2. Choose the best split
0. Choose a **random set** of features

1. Need to evaluate **all possible splits**

2. Choose the **best split**

Both *original* and *reverse* are equally bad, so we *toss a coin* again…
0. Choose a **random set** of features

1. Need to evaluate all possible splits

2. Choose the **best split**
This is what we have built so far:

- $X_2 > 4.5$
  - False
    - Class = 1
  - True
    - Class = 0

Bag #1
This is what we have built so far:

In principle you may go deeper into each one of the leaves.

Every time you would need to toss a coin to select a new random set of features from initial features.
We kept $X_2$.
We kept $X_2$.
We kept $X_2$.
We kept $X_2$.
We kept $X_2$

We keep $X_1$
We kept $X_2$.

We keep $X_1$. 

80%
We kept $X_1$
We can build an **ensemble**

We kept $X_2$

B1

We kept $X_1$

B2

We kept $X_1$

B3

We can build an ensemble

80%
We can build an ensemble.
We can build an ensemble.

We kept $X_2$

We kept $X_1$

We kept $X_1$

We kept $X_1$
This model is **imperfect**

We used only *1 layer* trees

Selecting features randomly for each node, has been shown to produce great results in practice (acts as **regularisation**).
The Random Forest algorithm

Form several bags (using bootstrapping)

For each bag build a tree

For each node in the tree choose random set of features

Merge predictions using majority vote or averaging
That's all Folks!