An Introduction to Supervised Machine Learning

Mohamed Siala
https://siala.github.io/

INSA-Toulouse & LAAS-CNRS

March 8, 2022

Context

Context

- An exponential increase of real-life applications of machine learning

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity
- I ask questions very often, so please be interactive

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity
- I ask questions very often, so please be interactive
- Feel free to stop me anytime. There is no stupid question!

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity
- I ask questions very often, so please be interactive
- Feel free to stop me anytime. There is no stupid question!
- Speak up if you spot a typo or an error. You might get bonus points!!

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity
- I ask questions very often, so please be interactive
- Feel free to stop me anytime. There is no stupid question!
- Speak up if you spot a typo or an error. You might get bonus points!!
- The evaluation will be decided later. Most likely it will be based on lecture questions and practical sessions

Context

- An exponential increase of real-life applications of machine learning
- This is an introduction course. Next year you will follow more advanced courses (depending on your orientation)
- You need some basic knowledge regarding linear algebra, algorithms and complexity
- I ask questions very often, so please be interactive
- Feel free to stop me anytime. There is no stupid question!
- Speak up if you spot a typo or an error. You might get bonus points!!
- The evaluation will be decided later. Most likely it will be based on lecture questions and practical sessions
- The course is articulated around three parts: introduction, interpretable machine learning (myself), and neural networks (Arthur Bit Monnot)

References

- T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition.*
  Springer Series in Statistics, Springer, 2009.
- S. J. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach, Third International Edition.*
  Pearson Education, 2010.
- C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *CoRR*, vol. abs/2103.11251, 2021.

Part 1: Introduction

Chapter 1: Context

[1] Image from `https://en.wikipedia.org/wiki/Cycling`

Figure 1: How to cycle? [1]

[1] Image from https://en.wikipedia.org/wiki/Cycling

[2] Image from `https://en.wikipedia.org/wiki/Global_biodiversity`

Figure 2: How to teach a child animal recognition? [2]

[2] Image from https://en.wikipedia.org/wiki/Global_biodiversity

[3] Image from `https://en.wikipedia.org/wiki`

Figure 3: How to predict a player's performance? [3]

[3] Image from `https://en.wikipedia.org/wiki`

[4] Image from https://en.wikipedia.org/wiki/DNA

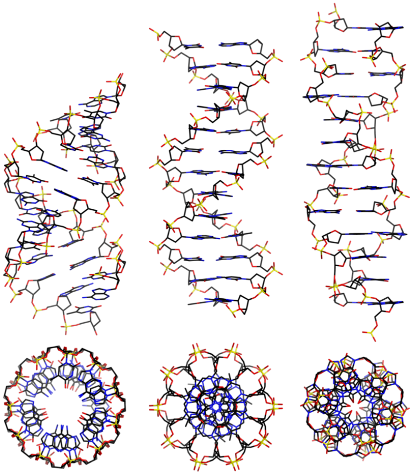


Figure 4: Analysis of evolutionary biology based on DNA patterns [4]

[4] Image from `https://en.wikipedia.org/wiki/DNA`

- Cycling: It needs a sequence of successful/unsuccessful trials!

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly
- DNA clustering: Let the machine discover by itself the different patterns.

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly
- DNA clustering: Let the machine discover by itself the different patterns.
- That's pretty much the philosophy of machine learning: feed the computer some date, and let it learn by itself

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly
- DNA clustering: Let the machine discover by itself the different patterns.
- That's pretty much the philosophy of machine learning: feed the computer some date, and let it learn by itself
- Note 1: Some computational problems are simply not solvable in a traditionally way and need machine learning

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly
- DNA clustering: Let the machine discover by itself the different patterns.
- That's pretty much the philosophy of machine learning: feed the computer some date, and let it learn by itself
- Note 1: Some computational problems are simply not solvable in a traditionally way and need machine learning
- Note 2: Machine learning is not always the solution! Consider for instance basic arithmetic operations

- Cycling: It needs a sequence of successful/unsuccessful trials!
- Animal recognition: It does not make sense to show the picture of every animal!

  $\Longrightarrow$ Show some pictures per animal and let the child learn
- Player performance: No straightforward formulae. Keep track of its latest performances and predict accordingly
- DNA clustering: Let the machine discover by itself the different patterns.
- That's pretty much the philosophy of machine learning: feed the computer some date, and let it learn by itself
- Note 1: Some computational problems are simply not solvable in a traditionally way and need machine learning
- Note 2: Machine learning is not always the solution! Consider for instance basic arithmetic operations

Examples of Machine Learning Applications [1]

- Autonomous cars
- Flying drones
- Face recognition
- Computer vision
- Natural language processing
- Music/movie recommendation
- Dating apps
- Friends recommendation
- Weather prediction
- Trading
- ...

The Big Picture

It depends on the problem at hand and on the nature of data!

The Big Picture

It depends on the problem at hand and on the nature of data!

1. **Labelled data**:
   - The task of animal recognition is to predict the animal in a given picture. The data (used to "train" the computer) is a set of pictures and each picture is associated to an animal. In this case, the **label** is a **category**

The Big Picture

It depends on the problem at hand and on the nature of data!

1. **Labelled data**:
    - The task of animal recognition is to predict the animal in a given picture. The data (used to "train" the computer) is a set of pictures and each picture is associated to an animal. In this case, the **label** is a **category**
    - In trading, the task is to predict the price evolution of a given share and the data is simply historical evolution of the share. In this case the data is **labelled** with **real numbers**.

The Big Picture

It depends on the problem at hand and on the nature of data!

1. **Labelled data**:
   - The task of animal recognition is to predict the animal in a given picture. The data (used to "train" the computer) is a set of pictures and each picture is associated to an animal. In this case, the **label** is a **category**
   - In trading, the task is to predict the price evolution of a given share and the data is simply historical evolution of the share. In this case the data is **labelled** with **real numbers**.
2. **Unlabelled data**: For instance, when using clustering to evaluate the density of a population. The data can simply be a set of coordinates with no labels.

The Big Picture

It depends on the problem at hand and on the nature of data!

1. **Labelled data**:
   - The task of animal recognition is to predict the animal in a given picture. The data (used to “train” the computer) is a set of pictures and each picture is associated to an animal. In this case, the **label** is a **category**
   - In trading, the task is to predict the price evolution of a given share and the data is simply historical evolution of the share. In this case the data is **labelled** with **real numbers**.
2. **Unlabelled data**: For instance, when using clustering to evaluate the density of a population. The data can simply be a set of coordinates with no labels.
3. **Continuously updated data**: The data is continuously updated according to previous experiences: For instance, a robot that tries to ride a bicycle learns how to bike by a sequence of trials

Supervised/Unsupervised/Reinforcement Learning

Supervised/Unsupervised/Reinforcement Learning

- Supervised Learning (Labelled data): Predict a function that associates inputs to outputs based on historical data

Supervised/Unsupervised/Reinforcement Learning

- Supervised Learning (Labelled data): Predict a function that associates inputs to outputs based on historical data
  - Categorical labels (discrete values): Classification

Supervised/Unsupervised/Reinforcement Learning

- Supervised Learning (Labelled data): Predict a function that associates inputs to outputs based on historical data
  - Categorical labels (discrete values): Classification
  - Non-categorical labels (real numbers): Regression

Supervised/Unsupervised/Reinforcement Learning

- Supervised Learning (Labelled data): Predict a function that associates inputs to outputs based on historical data
  - Categorical labels (discrete values): Classification
  - Non-categorical labels (real numbers): Regression
- Unsupervised Learning: The task is to figure out patterns presented in the data (unlabelled data)

Supervised/Unsupervised/Reinforcement Learning

- Supervised Learning (Labelled data): Predict a function that associates inputs to outputs based on historical data
  - Categorical labels (discrete values): Classification
  - Non-categorical labels (real numbers): Regression
- Unsupervised Learning: The task is to figure out patterns presented in the data (unlabelled data)
- Reinforcement learning: learning from a series of rewards /punishments
- But also, depending on the problem, data could be both labelled/non labelled, etc.. (semi-supervised learning)

[5] Image from https://en.wikipedia.org/wiki/Global_biodiversity

Figure 5: How to teach a child animal recognition? [5]

[5] Image from https://en.wikipedia.org/wiki/Global_biodiversity

Figure 5: How to teach a child animal recognition? [5]

Classification task

[5] Image from https://en.wikipedia.org/wiki/Global_biodiversity

[6] Image from https://en.wikipedia.org/wiki

Figure 6: How to predict a player's performance? [6]

[6] Image from https://en.wikipedia.org/wiki

Figure 6: How to predict a player's performance? [6]

Regression task

[6] Image from `https://en.wikipedia.org/wiki`

[7] Image from https://en.wikipedia.org/wiki/DNA

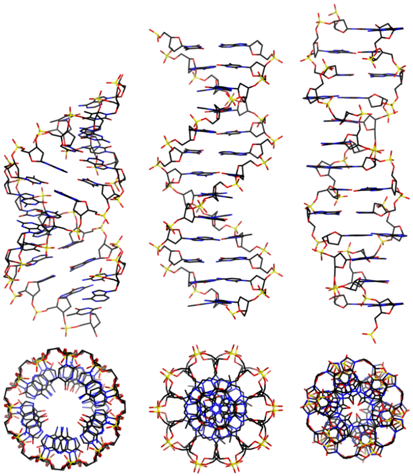


Figure 7: Analysis of evolutionary biology based on DNA patterns [7]

[7] Image from https://en.wikipedia.org/wiki/DNA

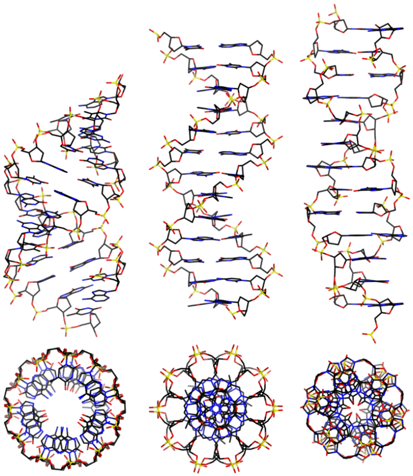


Figure 7: Analysis of evolutionary biology based on DNA patterns [7]

Unsupervised learning (clustering) task

[7] Image from `https://en.wikipedia.org/wiki/DNA`

[8] Image from https://en.wikipedia.org/wiki/Cycling

Figure 8: How to cycle? [8]

[8] Image from https://en.wikipedia.org/wiki/Cycling

Figure 8: How to cycle? [8]

Reinforcement learning

[8] Image from https://en.wikipedia.org/wiki/Cycling

Chapter 2: Supervised Machine Learning

Supervised Machine Learning: The Basics

- We focus in this course on Supervised ML
- Examples of applications
  - Tumor detection:

Supervised Machine Learning: The Basics

- We focus in this course on Supervised ML
- Examples of applications
  - Tumor detection: The data is a collection of brain scans. Each scan is associated with a label indicating the type of tumor
    $\Longrightarrow$ Classification
  - Credit score:

Supervised Machine Learning: The Basics

- We focus in this course on Supervised ML
- Examples of applications
  - Tumor detection: The data is a collection of brain scans. Each scan is associated with a label indicating the type of tumor
    $\Longrightarrow$ Classification
  - Credit score: The data is a collection of clients profiles (age, salary, genre (?), job, ...) with a positive or negative feedback
    $\Longrightarrow$ Binary classification
  - Precipitation prediction:

Supervised Machine Learning: The Basics

- We focus in this course on Supervised ML
- Examples of applications
  - Tumor detection: The data is a collection of brain scans. Each scan is associated with a label indicating the type of tumor
    $\Longrightarrow$ Classification
  - Credit score: The data is a collection of clients profiles (age, salary, genre (?), job, ...) with a positive or negative feedback
    $\Longrightarrow$ Binary classification
  - Precipitation prediction: (loosely speaking) the data is a collection of sequential weather conditions and the purpose is to predict the Precipitation chance (real value)
    $\Longrightarrow$ Regression

Problem Definition [2]

Problem Definition [2]

- Given a historical data (**training set**) in the form of input-output examples: $\{(x_1, y_1), \ldots (x_n, y_n)\}$ where x_i is an input, y_i is the output of x_i drawn from an unknown function f

Problem Definition [2]

- Given a historical data (**training set**) in the form of input-output examples: $\{(x_1, y_1), \ldots (x_n, y_n)\}$ where x_i is an input, y_i is the output of x_i drawn from an unknown function f
- Find a function f_h (called a hypothesis or model) that approximates the true function f
- The approximation criterion can be defined in different ways. We can consider it as a function minimizing some error.

Training Phase

Training Phase

- The function f_h is constructed via a **training algorithm**

Training Phase

- The function f_h is constructed via a **training algorithm**
- The training algorithm depends on the hypothesis space

Training Phase

- The function f_h is constructed via a **training algorithm**
- The training algorithm depends on the hypothesis space
- Examples of hypothesis space (family of functions) include polynomial functions, trigonometry functions, decision trees, decision lists, neural networks, ...

Testing Phase

Testing Phase

- The evaluation of the constructed hypothesis (or model) is done via a set of unseen examples called **testing set**

Testing Phase

- The evaluation of the constructed hypothesis (or model) is done via a set of unseen examples called **testing set**
- The testing set is usually taken randomly from the initial data. However, it shouldn't be part of the training set

Testing Phase

- The evaluation of the constructed hypothesis (or model) is done via a set of unseen examples called **testing set**
- The testing set is usually taken randomly from the initial data. However, it shouldn't be part of the training set
- The common way is to split the initial data into a training and testing sets randomly

Testing Phase

- The evaluation of the constructed hypothesis (or model) is done via a set of unseen examples called **testing set**
- The testing set is usually taken randomly from the initial data. However, it shouldn't be part of the training set
- The common way is to split the initial data into a training and testing sets randomly

Model Evaluation: Classification

Model Evaluation: Classification

- Training accuracy: percentage of training examples that are well predicted

Model Evaluation: Classification

- Training accuracy: percentage of training examples that are well predicted
- Testing accuracy: percentage of testing examples that are well predicted

Model Evaluation: Classification

- Training accuracy: percentage of training examples that are well predicted
- Testing accuracy: percentage of testing examples that are well predicted
- The concept of **generalisation** is precisely the quality of testing accuracy

Model Evaluation: Classification

- Training accuracy: percentage of training examples that are well predicted
- Testing accuracy: percentage of testing examples that are well predicted
- The concept of **generalisation** is precisely the quality of testing accuracy

Classification Evaluation: The Confusion Matrix

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement
- Consider binary classification (positive/negative classes) with 80% testing accuracy

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement
- Consider binary classification (positive/negative classes) with 80% testing accuracy
- 80% seems good, however, a deeper investigation shows that most of negative examples are wrongly predicted! This shows a biases in the model (biased towards positive examples)

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement
- Consider binary classification (positive/negative classes) with 80% testing accuracy
- 80% seems good, however, a deeper investigation shows that most of negative examples are wrongly predicted! This shows a biases in the model (biased towards positive examples)
- The purpose of the confusion matrix is precisely to have a refined evaluation

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement
- Consider binary classification (positive/negative classes) with 80% testing accuracy
- 80% seems good, however, a deeper investigation shows that most of negative examples are wrongly predicted! This shows a biases in the model (biased towards positive examples)
- The purpose of the confusion matrix is precisely to have a refined evaluation
- In the case on m classes, the matrix is of dimension $m \times m$ where $M[i][j]$ is the number of examples of class i that are predicted to be as the class b

Classification Evaluation: The Confusion Matrix

- Accuracy alone hinders the way predictions are made. Model evaluation needs more refinement
- Consider binary classification (positive/negative classes) with 80% testing accuracy
- 80% seems good, however, a deeper investigation shows that most of negative examples are wrongly predicted! This shows a biases in the model (biased towards positive examples)
- The purpose of the confusion matrix is precisely to have a refined evaluation
- In the case on m classes, the matrix is of dimension $m \times m$ where $M[i][j]$ is the number of examples of class i that are predicted to be as the class b
- The model bias can be easily seen: For instance, one can answer statistical queries such as: is the error evenly distributed? to which class the model is likely to predict? ...

The Confusion Matrix in the Case of Binary Classification

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class
- The confusion matrix is of dimension 2×2

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class
- The confusion matrix is of dimension 2×2
- True Positive Rate (TP rate): the likelihood that a positive example is well classified

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class
- The confusion matrix is of dimension 2×2
- True Positive Rate (TP rate): the likelihood that a positive example is well classified
- False Positive Rate (FP rate): the likelihood that a positive example is wrongly classified

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class
- The confusion matrix is of dimension 2×2
- True Positive Rate (TP rate): the likelihood that a positive example is well classified
- False Positive Rate (FP rate): the likelihood that a positive example is wrongly classified
- True Negative Rate (TN rate): the likelihood that a negative example is well classified

The Confusion Matrix in the Case of Binary Classification

- A positive class and a negative class
- The confusion matrix is of dimension 2×2
- True Positive Rate (TP rate): the likelihood that a positive example is well classified
- False Positive Rate (FP rate): the likelihood that a positive example is wrongly classified
- True Negative Rate (TN rate): the likelihood that a negative example is well classified
- False Negative Rate (FN rate): the likelihood that a negative example is wrongly classified

The Covid Example (1)

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

- TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

- TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
- TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

- TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
- TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
- Positive individuals has 93% chance to be correctly predicted

The Covid Example (1)

- Assume that we have a prediction model with 85% accuracy
- 100 individuals: 70 positives and 30 negatives
- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

- TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
- TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
- Positive individuals has 93% chance to be correctly predicted
- Negative individuals has 66% chance to be correctly predicted

The Covid Example (2)

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted
  - Negative individuals has 66% chance to be correctly predicted

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted
  - Negative individuals has 66% chance to be correctly predicted
- What is the probability for someone who tested positive to be truly positive?

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted
  - Negative individuals has 66% chance to be correctly predicted
- What is the probability for someone who tested positive to be truly positive? $= 65/(65 + 10) = 86\%$

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted
  - Negative individuals has 66% chance to be correctly predicted
- What is the probability for someone who tested positive to be truly positive? $= 65/(65+10) = 86\%$
- What is the probability for someone who tested negative to be truly negative?

The Covid Example (2)

- The confusion matrix:

	Positive	Negative
Positive	65	5
Negative	10	20

  - TP rate is $65/70 = 93\%$; FP rate is $5/70 = 7\%$
  - TN rate is $20/30 = 66\%$; FN rate is $10/30 = 33\%$
  - Positive individuals has 93% chance to be correctly predicted
  - Negative individuals has 66% chance to be correctly predicted
- What is the probability for someone who tested positive to be truly positive? $= 65/(65+10) = 86\%$
- What is the probability for someone who tested negative to be truly negative? $= 20/(5+20) = 80\%$

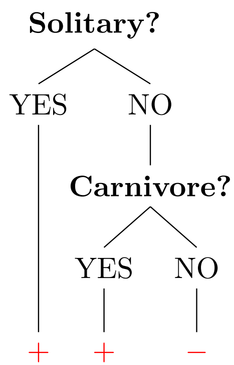
Toy Example: DTs to Predict The Likelihood of Animal Extinction

Toy Example: DTs to Predict The Likelihood of Animal Extinction

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

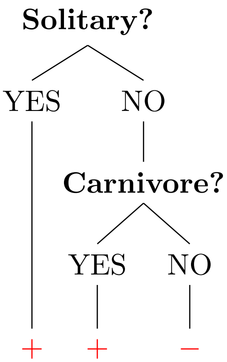
Toy Example: DTs to Predict The Likelihood of Animal Extinction

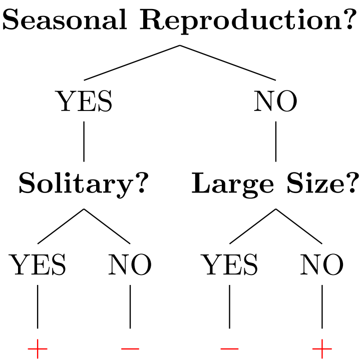
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes



Toy Example: DTs to Predict The Likelihood of Animal Extinction

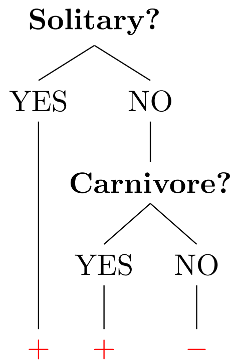
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

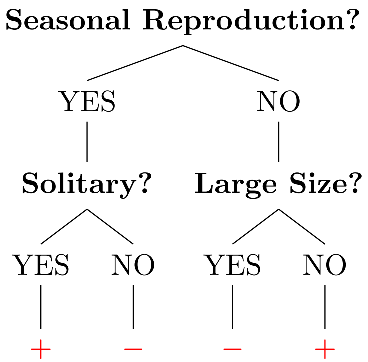




Toy Example: DTs to Predict The Likelihood of Animal Extinction

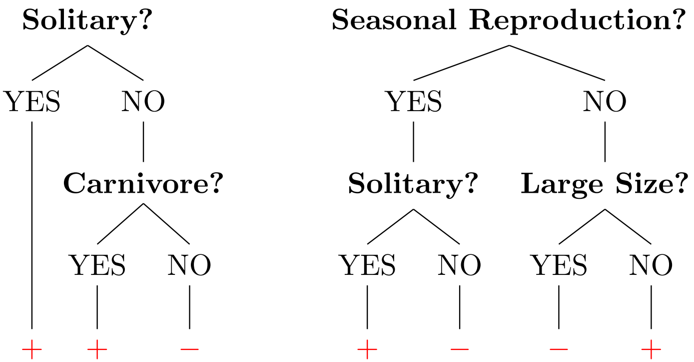
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes





Toy Example: DTs to Predict The Likelihood of Animal Extinction

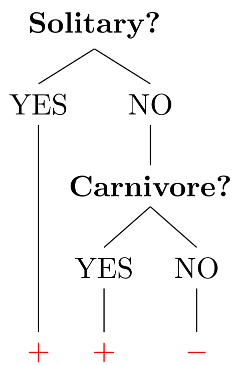
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

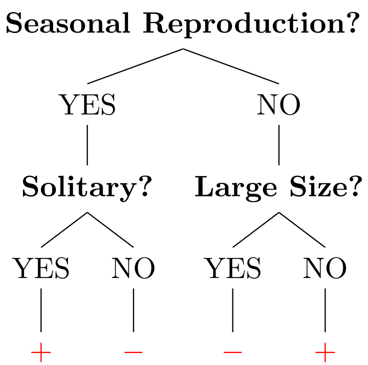


- Tabular data

Toy Example: DTs to Predict The Likelihood of Animal Extinction

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

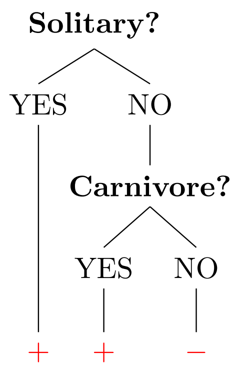


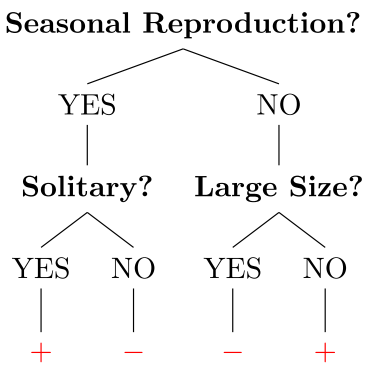


- Tabular data
- Hypothesis space: Decision trees

Toy Example: DTs to Predict The Likelihood of Animal Extinction

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

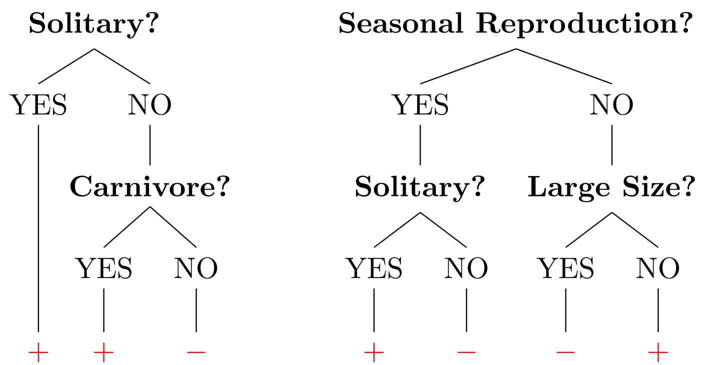




- Tabular data
- Hypothesis space: Decision trees
- Left tree: accuracy 2/5

Toy Example: DTs to Predict The Likelihood of Animal Extinction

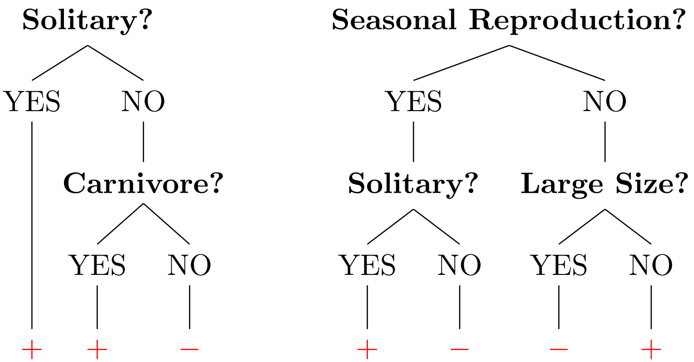
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes



- Tabular data
- Hypothesis space: Decision trees
- Left tree: accuracy 2/5
- Right tree: accuracy 2/5

Toy Example: DTs to Predict The Likelihood of Animal Extinction

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes



- Tabular data
- Hypothesis space: Decision trees
- Left tree: accuracy 2/5
- Right tree: accuracy 2/5

Model Evaluation: Regression

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted
- How about regression?

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted
- How about regression?
- Take the example of estimating the price of a house based on the surface and the distance to the city.

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted
- How about regression?
- Take the example of estimating the price of a house based on the surface and the distance to the city.
- Suppose that the real price of a house is 1 Million and the model predicted $999k$.

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted
- How about regression?
- Take the example of estimating the price of a house based on the surface and the distance to the city.
- Suppose that the real price of a house is 1 Million and the model predicted $999k$.
- Obviously it's not a correct prediction, however, it seems close enough! But how close?

Model Evaluation: Regression

- In the case of classification, a simple way to define error is to count the number of examples wrongly predicted
- How about regression?
- Take the example of estimating the price of a house based on the surface and the distance to the city.
- Suppose that the real price of a house is 1 Million and the model predicted $999k$.
- Obviously it's not a correct prediction, however, it seems close enough! But how close?
- We need an error function that take into account a notion of distance between true values and predicted values

Model Evaluation: Regression

Model Evaluation: Regression

- Two well known functions: Mean Absolute Error(MAE) and Mean Square Error(MSE) (or Root Mean Square Error($RMSE$))

Model Evaluation: Regression

- Two well known functions: Mean Absolute Error(MAE) and Mean Square Error(MSE) (or Root Mean Square Error($RMSE$))
- Consider a dataset with n examples where y is the vector of the true values and $\hat{y}$ is the vector of predicted values:

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{MSE}$$

Beyond Traditional Evaluations: Crucial Predictions

Beyond Traditional Evaluations: Crucial Predictions

- Take the example of tumor detection

Beyond Traditional Evaluations: Crucial Predictions

- Take the example of tumor detection
- Predicting the non-presence of a tumor for a patient that has a tumor is a serious problem

Beyond Traditional Evaluations: Crucial Predictions

- Take the example of tumor detection
- Predicting the non-presence of a tumor for a patient that has a tumor is a serious problem
- How to deal with such situation?

Beyond Traditional Evaluations: Crucial Predictions

- Take the example of tumor detection
- Predicting the non-presence of a tumor for a patient that has a tumor is a serious problem
- How to deal with such situation?
- For instance, one can add a weight on the mis-classification error as follows: the cost of mis-classifying a patient with a tumor is 5 times the cost of mis-classifying a patient without a tumor

Beyond Traditional Evaluations: Crucial Predictions

- Take the example of tumor detection
- Predicting the non-presence of a tumor for a patient that has a tumor is a serious problem
- How to deal with such situation?
- For instance, one can add a weight on the mis-classification error as follows: the cost of mis-classifying a patient with a tumor is 5 times the cost of mis-classifying a patient without a tumor
- In this case the objective function is slightly different from the standard accuracy (weighted sum)

Computational Hardness

Out of these three questions, which one is the hardest and which one is the easiest (computationally)?

1. Build a perfect decision tree (a tree that classifies every example correctly) 100% accuracy ?
2. Build the best decision tree within a height h ?
3. Build a perfect decision tree with a minimum height ?

Computational Hardness

Out of these three questions, which one is the hardest and which one is the easiest (computationally)?

1. Build a perfect decision tree (a tree that classifies every example correctly) 100% accuracy ?
2. Build the best decision tree within a height h ?
3. Build a perfect decision tree with a minimum height ?

- Question 1 is the easiest: we can always develop children to classify every example
- Let k be the number of features; $f(n)$ be the complexity of Question 2 and $g(n)$ be the complexity of Question 3
- $g(n) = O(k \times f(n))$: We can solve Question 3 by solving Question 2 iteratively by increasing/decreasing the height

Computational Hardness

Out of these three questions, which one is the hardest and which one is the easiest (computationally)?

1. Build a perfect decision tree (a tree that classifies every example correctly) 100% accuracy ?
2. Build the best decision tree within a height h ?
3. Build a perfect decision tree with a minimum height ?

- Question 1 is the easiest: we can always develop children to classify every example
- Let k be the number of features; $f(n)$ be the complexity of Question 2 and $g(n)$ be the complexity of Question 3
- $g(n) = O(k \times f(n))$: We can solve Question 3 by solving Question 2 iteratively by increasing/decreasing the height
- Different objective functions can be defined (i.e., the training problem itsetf can have different definitions)
- The definition of the objective function with the hypothesis space has an impact on the complexity of training

To summarize

- Supervised Learning: The data is labelled

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values
- Classification: Supervised Learning with discrete values (classes)

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values
- Classification: Supervised Learning with discrete values (classes)
- Regression evaluation: accuracy in terms of minimum error (Mean Absolute Error, Root Mean Square Error)

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values
- Classification: Supervised Learning with discrete values (classes)
- Regression evaluation: accuracy in terms of minimum error (Mean Absolute Error, Root Mean Square Error)
- Classification evaluation: accuracy in terms of examples well classified but also the confusion matrix (it can show some bias)

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values
- Classification: Supervised Learning with discrete values (classes)
- Regression evaluation: accuracy in terms of minimum error (Mean Absolute Error, Root Mean Square Error)
- Classification evaluation: accuracy in terms of examples well classified but also the confusion matrix (it can show some bias)
- Depending on the problem at hand, the objective function can have have different forms

To summarize

- Supervised Learning: The data is labelled
- Each example is defined by a sequence of values (of the different attributes/features) with an output
- Training vs. Testing sets (disjoints)
- **Generalisation:** when the model generalizes well on unseen data
- Regression: Supervised Learning with real values
- Classification: Supervised Learning with discrete values (classes)
- Regression evaluation: accuracy in terms of minimum error (Mean Absolute Error, Root Mean Square Error)
- Classification evaluation: accuracy in terms of examples well classified but also the confusion matrix (it can show some bias)
- Depending on the problem at hand, the objective function can have have different forms
- The way the optimisation problem is defined impacts its computational complexity

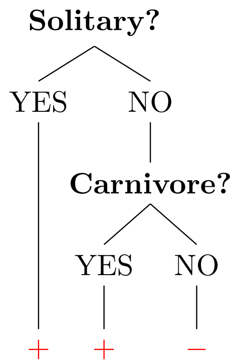
Questions & Exercises

Questions & Exercises

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

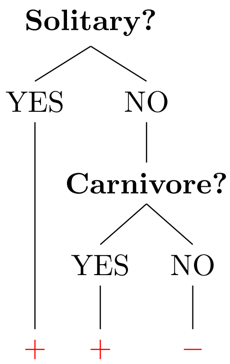
Questions & Exercises

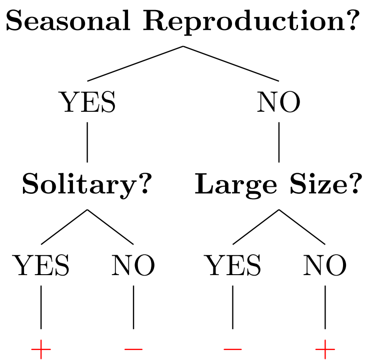
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes



Questions & Exercises

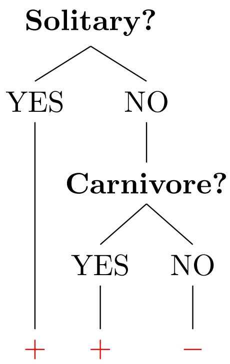
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

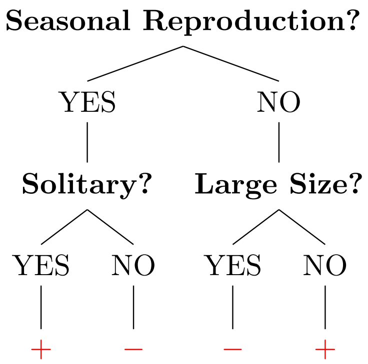




Questions & Exercises

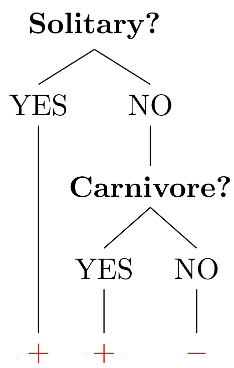
Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

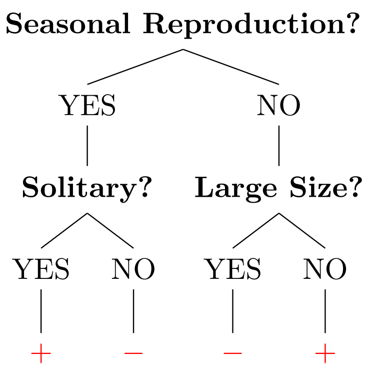




Questions & Exercises

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

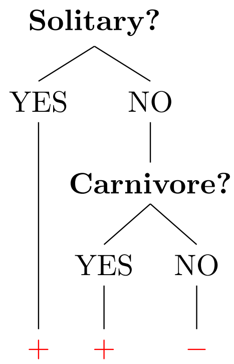


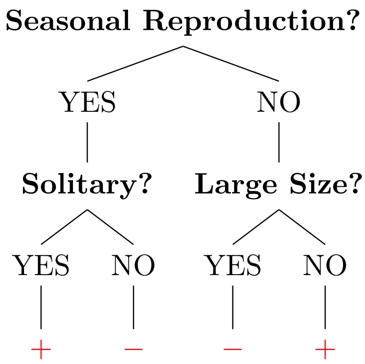


- Find a perfect tree

Questions & Exercises

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

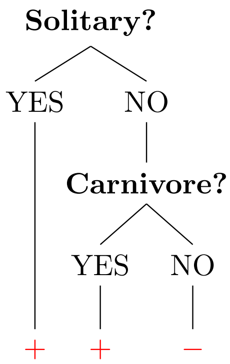


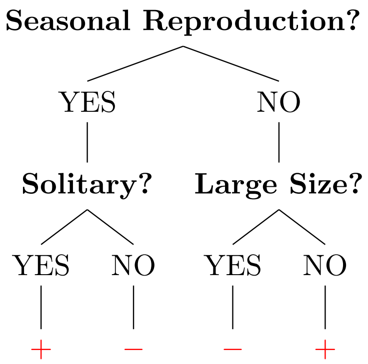


- Find a perfect tree
- Find a perfect tree with a minimum weight

Questions & Exercises

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

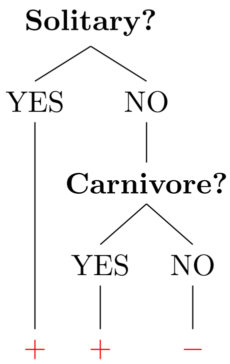


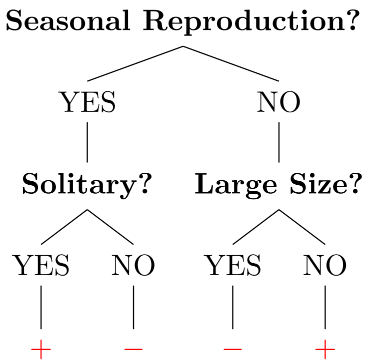


- Find a perfect tree
- Find a perfect tree with a minimum weight
- Find a best tree with height 2

Questions & Exercises

Big Size	Carnivore	Seasonal Reproduction	Solitary	Extinct
0	1	0	1	yes
1	0	0	1	yes
0	0	0	1	no
1	1	1	0	no
0	0	1	0	yes

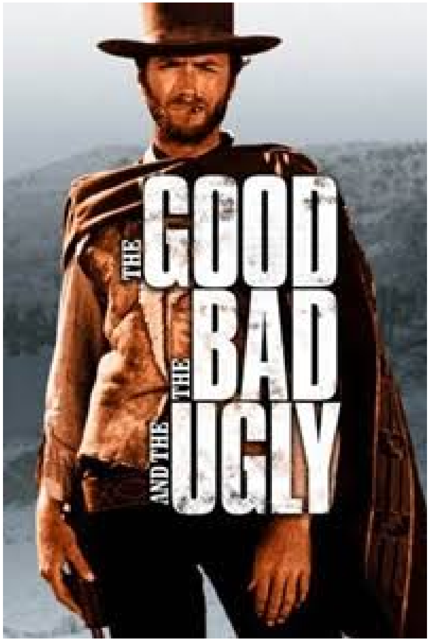




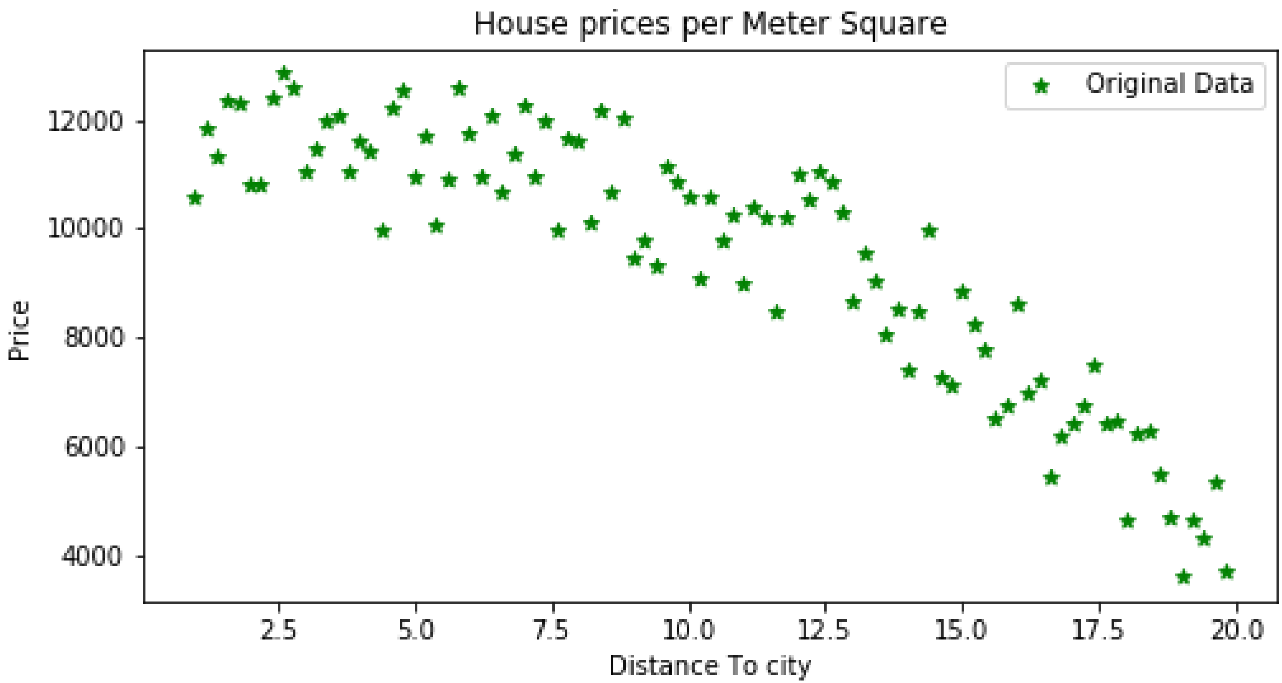
- Find a perfect tree
- Find a perfect tree with a minimum weight
- Find a best tree with height 2

Chapter 3: Deeper Evaluations

Overfitting, Underfitting, and Goodfitting

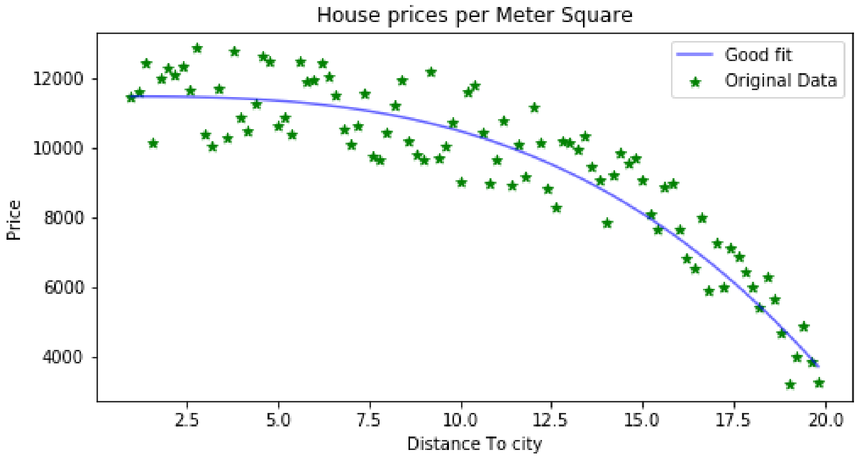


The Housing Prices Example



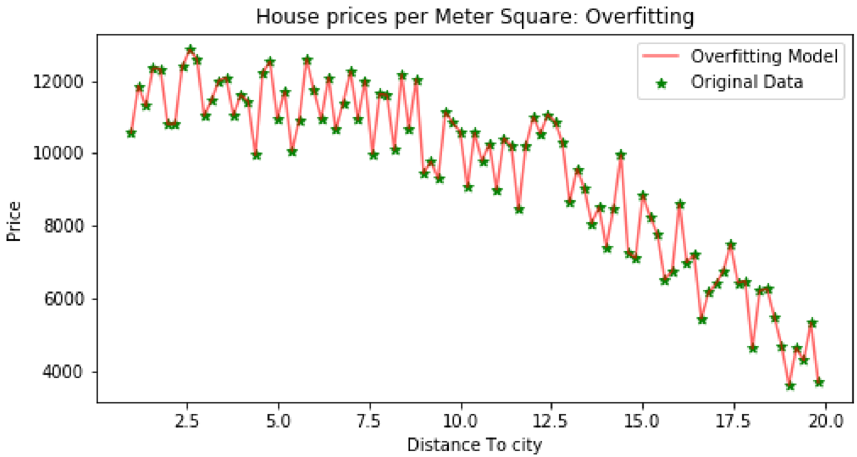
This data includes some **noise**. That is, points that are not correctly collected (which is often the case in real applications)

The Housing Prices Example: The Good

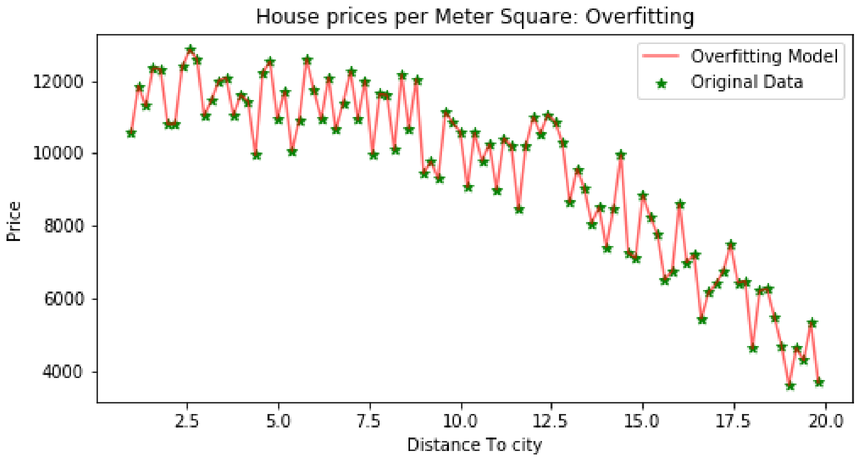


We can make an analogy to a smart student who has a good understanding of a lecture

The Housing Prices Example: The Bad (Overfitting)

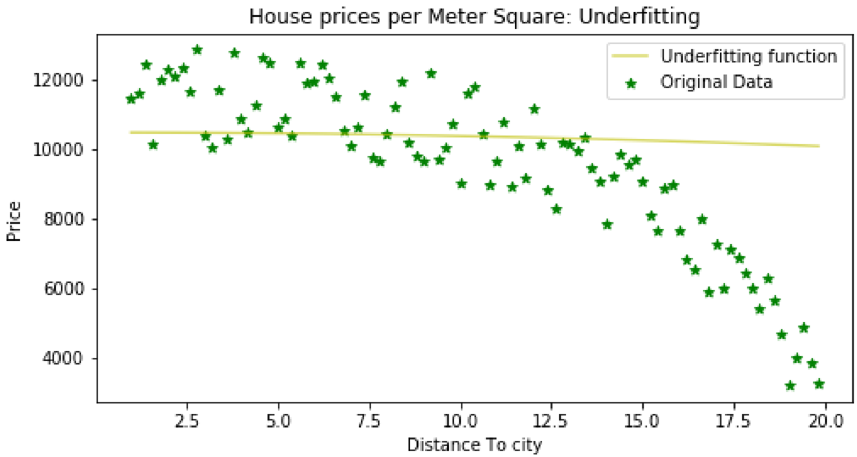


The Housing Prices Example: The Bad (Overfitting)

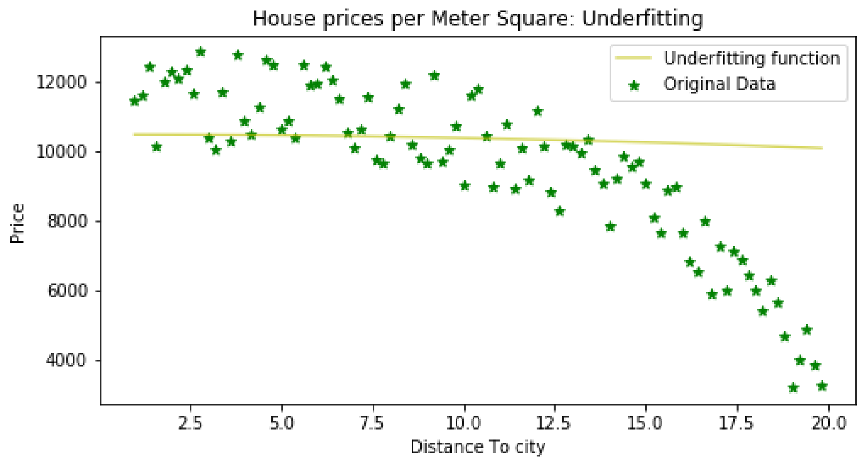


We can make an analogy to the student who "learns" the lecture mechanically without a real understanding.

The Housing Prices Example: The Ugly (Underfitting)

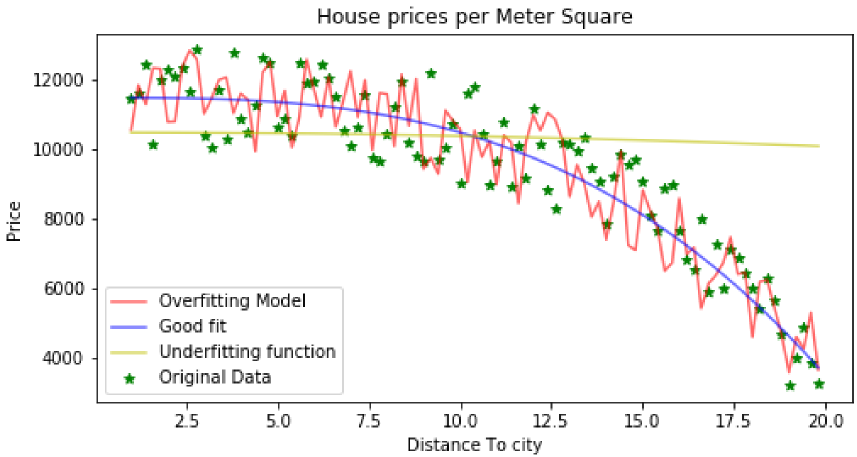


The Housing Prices Example: The Ugly (Underfitting)



We can make an analogy to a lazy student who barely remember the lecture without any understanding

The Housing Prices Example: All Together



Overfitting, Underfitting, and a Good Fit

Overfitting, Underfitting, and a Good Fit

- Overfitting happens when the model tries to squeeze everything in including noise without an "intuitive understanding of the data"

Overfitting, Underfitting, and a Good Fit

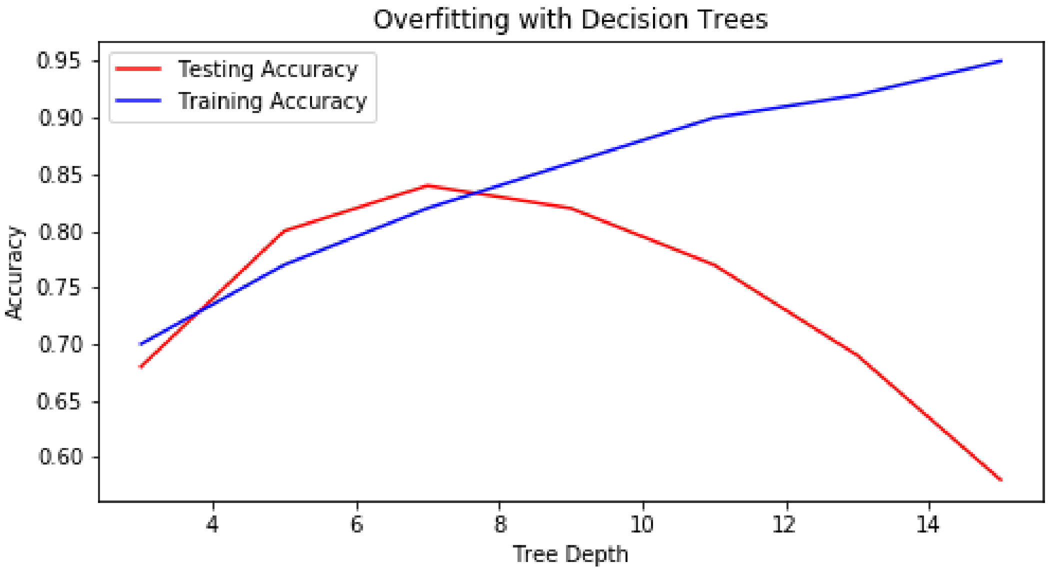
- Overfitting happens when the model tries to squeeze everything in including noise without an ”intuitive understanding of the data”
- Underfitting happens when the model performs badly on the training and testing data (no real learning).

Overfitting, Underfitting, and a Good Fit

- Overfitting happens when the model tries to squeeze everything in including noise without an "intuitive understanding of the data"
- Underfitting happens when the model performs badly on the training and testing data (no real learning).
- A good fit happens when the model approximates well the true distribution without being disturbed by noise (good generalisation)

Overfitting with Decision Trees as an Example

Overfitting with Decision Trees as an Example



Overfitting with Decision Trees as an Example

- The longer the tree, the better the training accuracy gets, however, this is not necessarily the case for the testing accuracy

Overfitting with Decision Trees as an Example

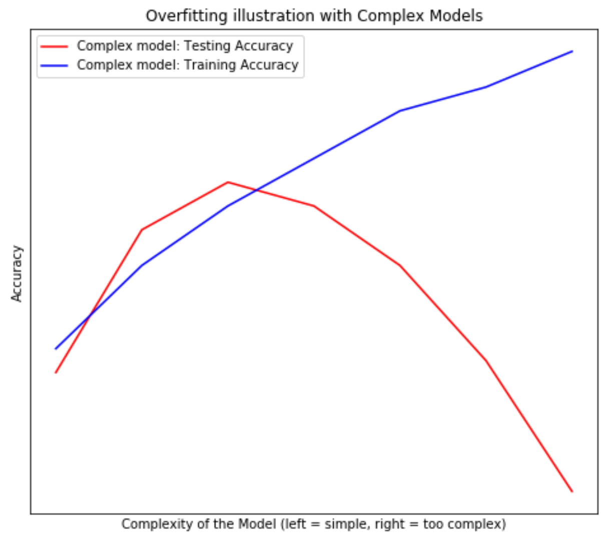
- The longer the tree, the better the training accuracy gets, however, this is not necessarily the case for the testing accuracy
- Testing accuracy increases at the beginning until a certain value (depth $= 7$), the it decreases afterwards

Overfitting with Decision Trees as an Example

- The longer the tree, the better the training accuracy gets, however, this is not necessarily the case for the testing accuracy
- Testing accuracy increases at the beginning until a certain value (depth $= 7$), the it decreases afterwards
- This happens because with longer trees, the model can classify correctly more examples in the training set, however, this includes noise.

Overfitting Based on the Complexity of the Model

Overfitting Based on the Complexity of the Model



- When the model is too simple, there is a risque of underfitting
- When the model is too complex, there is a risque of overfitting
- We need a Model that is somehow in between
- ML libraries offer parameters for regulation to avoid overfitting/underfitting

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?
- Generalisation should be evaluated a number of random splits

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?
- Generalisation should be evaluated a number of random splits
- Also the confusion matrices can be used for evaluation

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?
- Generalisation should be evaluated a number of random splits
- Also the confusion matrices can be used for evaluation
- A common way is to use the $k-$fold cross validation:

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?
- Generalisation should be evaluated a number of random splits
- Also the confusion matrices can be used for evaluation
- A common way is to use the $k-$fold cross validation:
    1. Split the data into k folds

Training Algorithms Evaluation

- Suppose that we have a number of training algorithms for a given hypothesis space
- How to choose the best?
- Generalisation should be evaluated a number of random splits
- Also the confusion matrices can be used for evaluation
- A common way is to use the $k-$fold cross validation:
  1. Split the data into k folds
  2. Perform the training k times. At each iteration, a different fold is chosen as a testing set and the rest is used for training
  3. Typical values for k are 5 and 10

Overcome Overfitting

- How to avoid overfitting?

Overcome Overfitting

- How to avoid overfitting?
- The testing set is inaccessible at the moment of training
- We can sacrifice a part of the training set as a 'validation set' to evaluate the generalisation of the model.
- Basically, the training set has a subset for training and a subset for validation (evaluation)
- A common way is to use $k-$cross validation on the training set to overcome overfitting
- Also, we can restrict the hypothesis space with simple models

Complexity/Quality Tradeoff

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)
- For instance, training quadratic functions is much harder (computationally) than training linear functions

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)
- For instance, training quadratic functions is much harder (computationally) than training linear functions
- However, may be a quadratic function is a better fit for the data at hand

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)
- For instance, training quadratic functions is much harder (computationally) than training linear functions
- However, may be a quadratic function is a better fit for the data at hand
- There is a trade-off between the quality of predictions and the model complexity

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)
- For instance, training quadratic functions is much harder (computationally) than training linear functions
- However, may be a quadratic function is a better fit for the data at hand
- There is a trade-off between the quality of predictions and the model complexity
- For example training a tree with depth 5 is much faster than training a tree of depth 9, but in terms of training quality, trees of depth 9 are better. However, trees with depth 9 might overfit

Complexity/Quality Tradeoff

- Training algorithms have resources costs: memory and runtime (we will see later how to train decision trees/linear functions)
- For instance, training quadratic functions is much harder (computationally) than training linear functions
- However, may be a quadratic function is a better fit for the data at hand
- There is a trade-off between the quality of predictions and the model complexity
- For example training a tree with depth 5 is much faster than training a tree of depth 9, but in terms of training quality, trees of depth 9 are better. However, trees with depth 9 might overfit
- Most ML libraries offer the possibility to control the complexity with a regularization parameter

Ockham's Razor Principle

[9] Philosopher https://en.wikipedia.org/wiki/William_of_Ockham

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?

[9] Philosopher https://en.wikipedia.org/wiki/William_of_Ockham

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements
- Ockham's Razor Principle[9]: pick the simplest!

[9] Philosopher https://en.wikipedia.org/wiki/William_of_Ockham

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements
- Ockham's Razor Principle[9]: pick the simplest!
- Simplicity is also hard to define

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements
- Ockham's Razor Principle[9]: pick the simplest!
- Simplicity is also hard to define
- In decision trees, simplicity could be the depth, the number of features, a combination of both, etc

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements
- Ockham's Razor Principle[9]: pick the simplest!
- Simplicity is also hard to define
- In decision trees, simplicity could be the depth, the number of features, a combination of both, etc
- When using polynomials (as a hypothesis space), lower degrees seems to be simpler

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Ockham's Razor Principle

- In the previous example, we have two different trees with the same accuracy
- Which tree to choose?
- Hard to answer without specific requirements
- Ockham's Razor Principle[9]: pick the simplest!
- Simplicity is also hard to define
- In decision trees, simplicity could be the depth, the number of features, a combination of both, etc
- When using polynomials (as a hypothesis space), lower degrees seems to be simpler
- In other cases it is very hard to define simplicity

[9] Philosopher `https://en.wikipedia.org/wiki/William_of_Ockham`

Complexity/Quality/Overfitting Tradeoff

The bottom line

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting
  - hard/easy training algorithms

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting
  - hard/easy training algorithms
  - complex/simple models
- Complex models can be computationally hard, however have better flexibility (some parameters can be turned off) and might have better quality

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting
  - hard/easy training algorithms
  - complex/simple models
- Complex models can be computationally hard, however have better flexibility (some parameters can be turned off) and might have better quality
- Complex models might overfit

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting
  - hard/easy training algorithms
  - complex/simple models
- Complex models can be computationally hard, however have better flexibility (some parameters can be turned off) and might have better quality
- Complex models might overfit
- Simple models might underfit

Complexity/Quality/Overfitting Tradeoff

The bottom line

- There are fine lines between:
  - overfitting/underfitting
  - hard/easy training algorithms
  - complex/simple models
- Complex models can be computationally hard, however have better flexibility (some parameters can be turned off) and might have better quality
- Complex models might overfit
- Simple models might underfit
- Ideally, we look for a hypothesis that is ‘easy’ to compute and simple enough to be a good fit

Chapter 4: Interpretable Models

The COMPAS Tool

MIT Technology Review

Featured Topics Newsletters Events Podcasts

Sign in Subscribe

TECH POLICY

AI is sending people to jail—and getting it wrong

Using historical data to train risk assessment tools could mean that machines are copying the mistakes of the past.

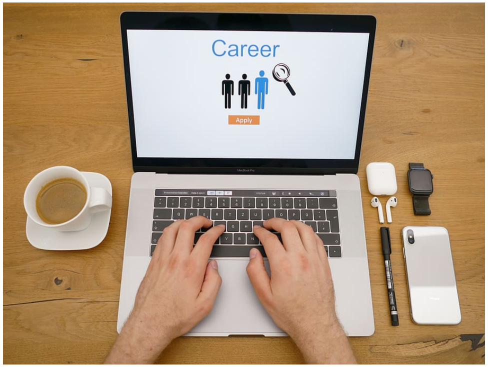
By Karen Hao

January 21, 2019

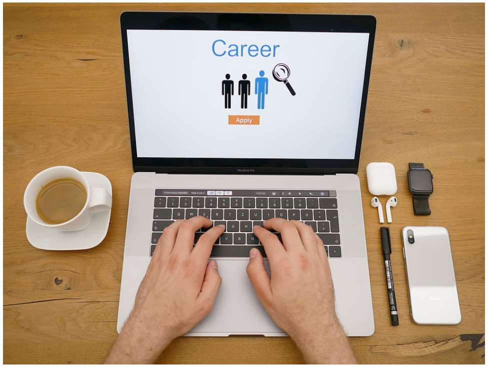
IAN WALDIE/GETTY IMAGES

Increasing Number of Real Life and Social AI Applications

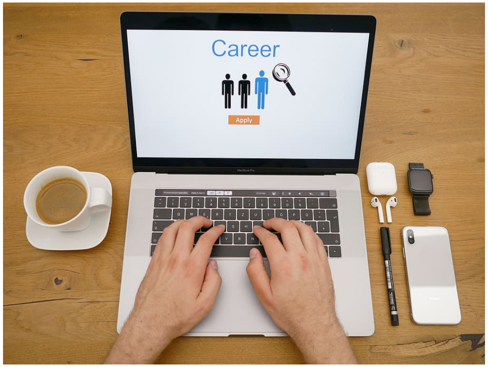
Increasing Number of Real Life and Social AI Applications



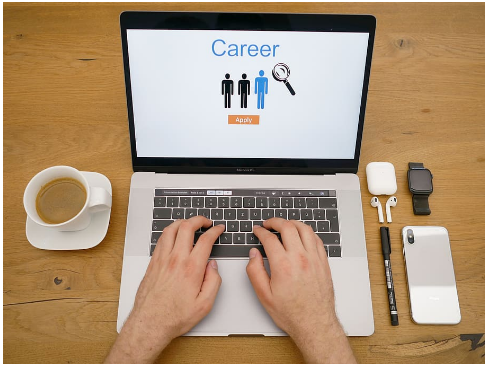
Increasing Number of Real Life and Social AI Applications



Increasing Number of Real Life and Social AI Applications



Increasing Number of Real Life and Social AI Applications



AI: Increasing Number of Real Life Applications Of Machine Learning

- The diverse applications of AI raised many ethical issues and questions

AI: Increasing Number of Real Life Applications Of Machine Learning

- The diverse applications of AI raised many ethical issues and questions
  - Job applications: AI that parses CVs for software engineers and recommends to hire mostly men

AI: Increasing Number of Real Life Applications Of Machine Learning

- The diverse applications of AI raised many ethical issues and questions
  - Job applications: AI that parses CVs for software engineers and recommends to hire mostly men
  - Credit scoring: AI that gives a credit score (for bank loans and credit applications) that recommends people from a particular geographical region, specific gender, social class, etc

AI: Increasing Number of Real Life Applications Of Machine Learning

- The diverse applications of AI raised many ethical issues and questions
  - Job applications: AI that parses CVs for software engineers and recommends to hire mostly men
  - Credit scoring: AI that gives a credit score (for bank loans and credit applications) that recommends people from a particular geographical region, specific gender, social class, etc
  - Compass tool: (2016) used by judges in the US to predict which criminals are likely to re-offend is found to be biased by the ethnicity (African-American/Caucasian).

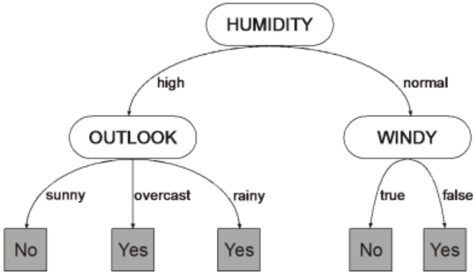
COMPASS data and Rule-based Predictions

Sex	Age	Priors	Juvenile Felonies	Juvenile Crimes	Ethnicity
Male	15	1	0	1	Caucasian
Male	15	1	0	1	African-American
Female	33	1	0	1	African-American
Female	27	0	1	0	Caucasian
Male	41	0	1	0	Caucasian
...	...	...	...	...	...

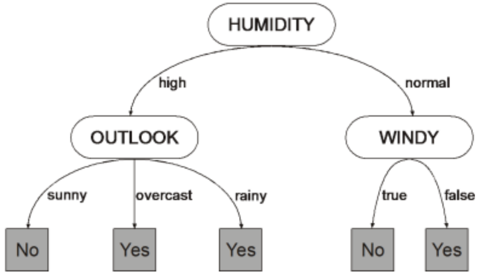
The problem is to predict recidivism. That is, the tendency of a convicted criminal to re-offend.

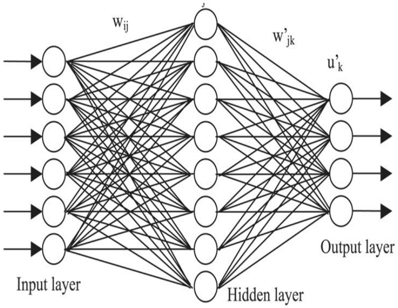
Black-Box vs Interpretable Models

Black-Box vs Interpretable Models



Black-Box vs Interpretable Models





Definitions [3]

- **Black-box model** : A formula that is either too complicated for any human to understand, or proprietary, so that one cannot understand its inner workings
- **Interpretable model** obeys a domain-specific set of constraints to allow it (or its predictions, or the data) to be more easily understood by humans. These constraints can differ dramatically depending on the domain.

Why Interpretable Models?

Why Interpretable Models?

- Transparent

Why Interpretable Models?

- Transparent
- Trustworthy

Why Interpretable Models?

- Transparent
- Trustworthy
- Inherently Explainable
- Well adapted for troubleshooting and diagnosis

Why Interpretable Models?

- Transparent
- Trustworthy
- Inherently Explainable
- Well adapted for troubleshooting and diagnosis
- **Mandatory criteria in high-stake decision making**

- We consider in this course tabular data (but extensions to other types is possible)
- Models:Decision trees, decision lists, decision rules, and linear functionns . . .

Decision rules & Decision Sets

Decision rules & Decision Sets

- They are defined as If-Condition-Then-Prediction rules

Decision rules & Decision Sets

- They are defined as If-Condition-Then-Prediction rules
- **Decision sets**: no specific order is given between the rules. Ties are broken by majority votes

Decision rules & Decision Sets

- They are defined as If-Condition-Then-Prediction rules
- **Decision sets**: no specific order is given between the rules. Ties are broken by majority votes
- **Decision rules**: rules are ordered by priority

Decision rules & Decision Sets

- They are defined as If-Condition-Then-Prediction rules
- **Decision sets**: no specific order is given between the rules. Ties are broken by majority votes
- **Decision rules**: rules are ordered by priority

Example of Rule List found by FairCORELS

- Data : `https://www.kaggle.com/danofer/compass`
- FairCORELS: `https://github.com/ferryjul/fairCORELS`

Decision rules & Decision Sets

- They are defined as If-Condition-Then-Prediction rules
- **Decision sets**: no specific order is given between the rules. Ties are broken by majority votes
- **Decision rules**: rules are ordered by priority

Example of Rule List found by FairCORELS

- Data : `https://www.kaggle.com/danofer/compass`
- FairCORELS: `https://github.com/ferryjul/fairCORELS`

```
if [priors:>3] then [recidivism]
else if [age:21-22 && gender:Male] then [recidivism]
else if [age:18-20] then [recidivism]
else if [age:23-25 && priors:2-3] then [recidivism]
else [no recidivism]
```

Rule list 5. Example of an unconstrained rule list found by `FairCORELS` on COMPAS dataset, with Accuracy $= 0.681$, $\mathrm{UNF}_{\mathrm{EOdds}} = 0.217$ and $\mathrm{UNF}_{\mathrm{CUAE}} = 0.046$