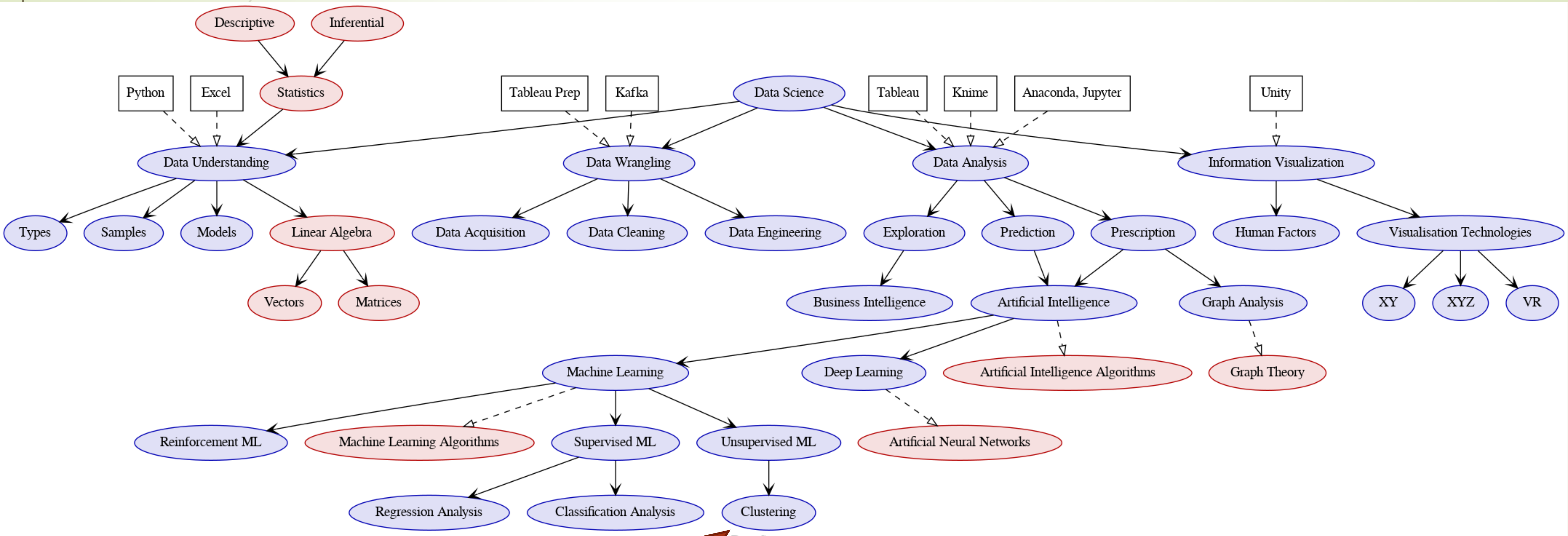
Data Science

Martin Vestergaard (mrv@cphbusiness.dk)
Todorka Dimitrova (tdi@cphbusiness.dk)

Intended Learning Outcomes

- Classification
  - What, why?
  - The iris dataset
  - Classification Models:
    - KNN
    - Naïve Bayes
  - How to choose a model?
  - How to tune a model?
  - How to measure the performance of a model?



Data Science topics

Classification

- Terminology:
  - **Rows:** samples / observations / examples / instances / records
  - **Columns:** features / predictors / attributes / independent variable / input / regressors / covariates
  - **Values to be predicted:** responses / targets / outcomes / labels / dependent variables

Feature A	Feature B	Feature C	Feature D	…	Label
0.431	0.98	43	Blue	…	Foo
-1.34	0.99	77	Green	…	Bar
…	…	…	…	…	…

Classification

- Supervised learning
  - We know the labels beforehand
  - This means we can check (supervise) the performance!

Feature A	Feature B	Feature C	Feature D	…	Label
0.431	0.98	43	Blue	…	Foo
-1.34	0.99	77	Green	…	Bar
…	…	…	…	…	…

Classification

- In *classification*, the **response** is
- In *regression*, the **response** is

Classification

- Used in
  - Finance
  - Healthcare
  - Political science
  - Handwriting detection
  - Image recognition
  - Credit rating
  - Loan safety prediction
  - Spam detection
  - Fraud detection
  - ...

The FAMOUS …….

- iris dataset
- you will see this again and again in machine learning litterature

The iris dataset

- Statistician, geneticist, eugenicist
- Collected iris flower measurements in 1936

Ronald Fisher
(1890 - 1962)

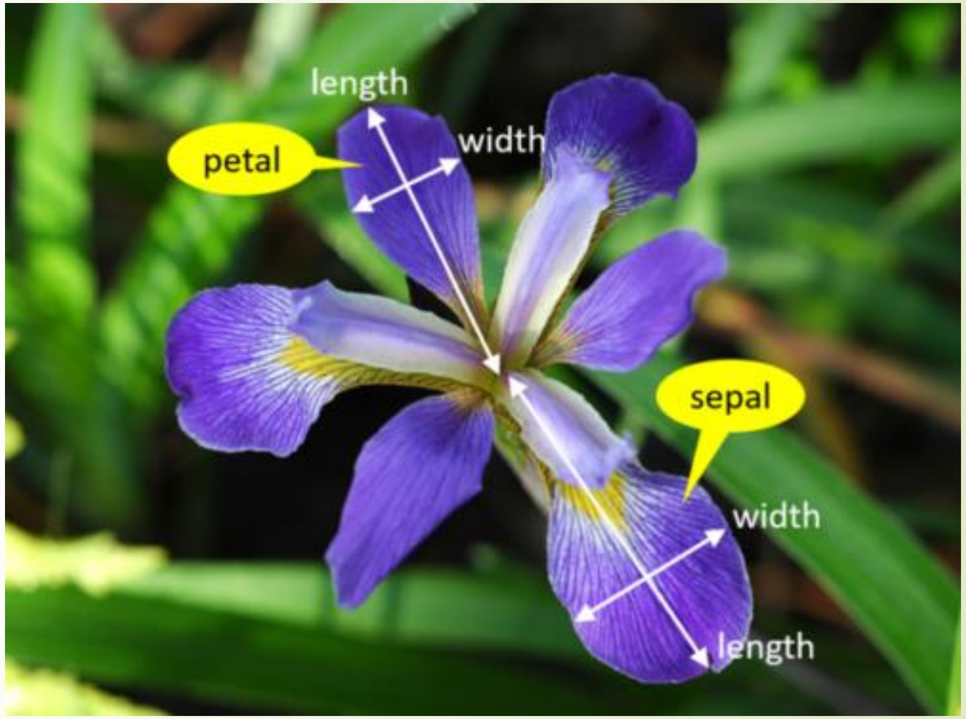
Iris setosa

Iris versicolor

Iris virginica

The iris dataset

- 150 samples (50 pr. class)
- 4 features:
  - Petal length (cm)
  - Petal width (cm)
  - Sepal length (cm)
  - Sepal width (cm)
- 1 label, one of:
  - Iris setosa
  - Iris versicolor
  - Iris virginica



Sepal: bægerblad
Petal: kronblad

The iris dataset

features (lengths)

label / class (species)

```
5.1,  3.8,  1.6,  0.2,  Iris-setosa
4.6,  3.2,  1.4,  0.2,  Iris-setosa
5.3,  3.7,  1.5,  0.2,  Iris-setosa
5.0,  3.3,  1.4,  0.2,  Iris-setosa
7.0,  3.2,  4.7,  1.4,  Iris-versicolor
6.4,  3.2,  4.5,  1.5,  Iris-versicolor
6.9,  3.1,  4.9,  1.5,  Iris-versicolor
5.5,  2.3,  4.0,  1.3,  Iris-versicolor
6.5,  2.8,  4.6,  1.5,  Iris-versicolor
5.7,  2.8,  4.5,  1.3,  Iris-versicolor
...
```

The iris dataset

features (lengths)

label / class (species)

5.1,	3.8,	1.6,	0.2,	Iris-setosa
4.6,	3.2,	1.4,	0.2,	Iris-setosa
5.3,	3.7,	1.5,	0.2,	Iris-setosa
[illegible]	3.3,	1.4,	0.2,	Iris-setosa
[illegible]	3.2,	4.7,	1.4,	Iris-versicolor
6.4,	3.2,	4.5,	1.5,	Iris-versicolor
6.9,	3.1,	4.9,	1.5,	Iris-versic[illegible]
[illegible]5,	2.3,	4.0,	1.3,	Iris-versic[illegible]
6[illegible]	2.8,	4.6,	1.5,	Iris-versicolo[illegible]
5.[illegible]	2.8,	4.5,	1.3,	Iris-vers[illegible]

5.8,	3.7,	1.4,	0.3,	????????

Given a new of **these**...

...can we predict **this?**

The iris dataset

- Obtaining the data
  - UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php
  - Sklearn:

```
from sklearn import datasets
irisBunch = datasets.load_iris()

X = irisBunch.data
y = irisBunch.target
```

a "bunch" is the data and its attributes

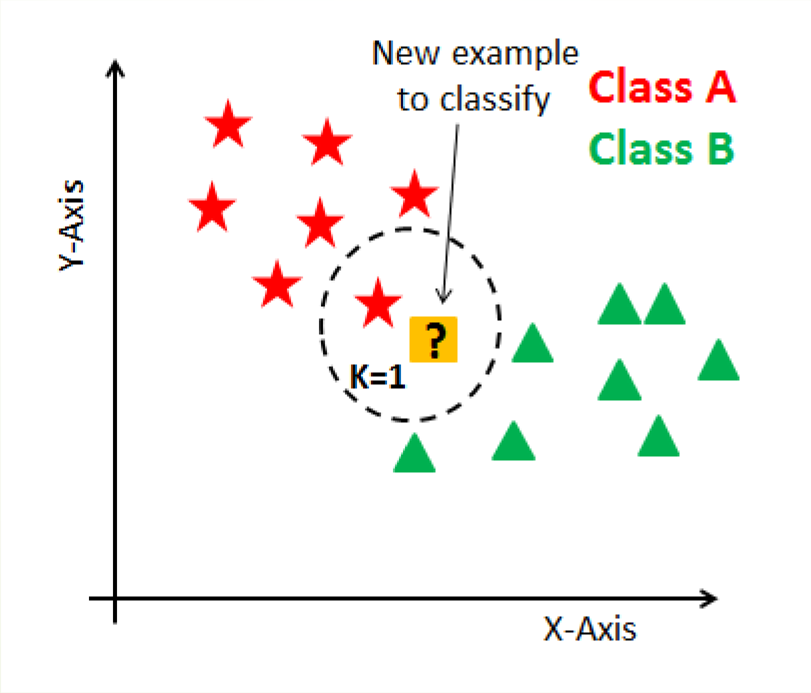
Classification

- Classification Models:
  - KNN
  - Naïve Bayes
  - Logistic Regression
  - Support Vector Machines

KNN

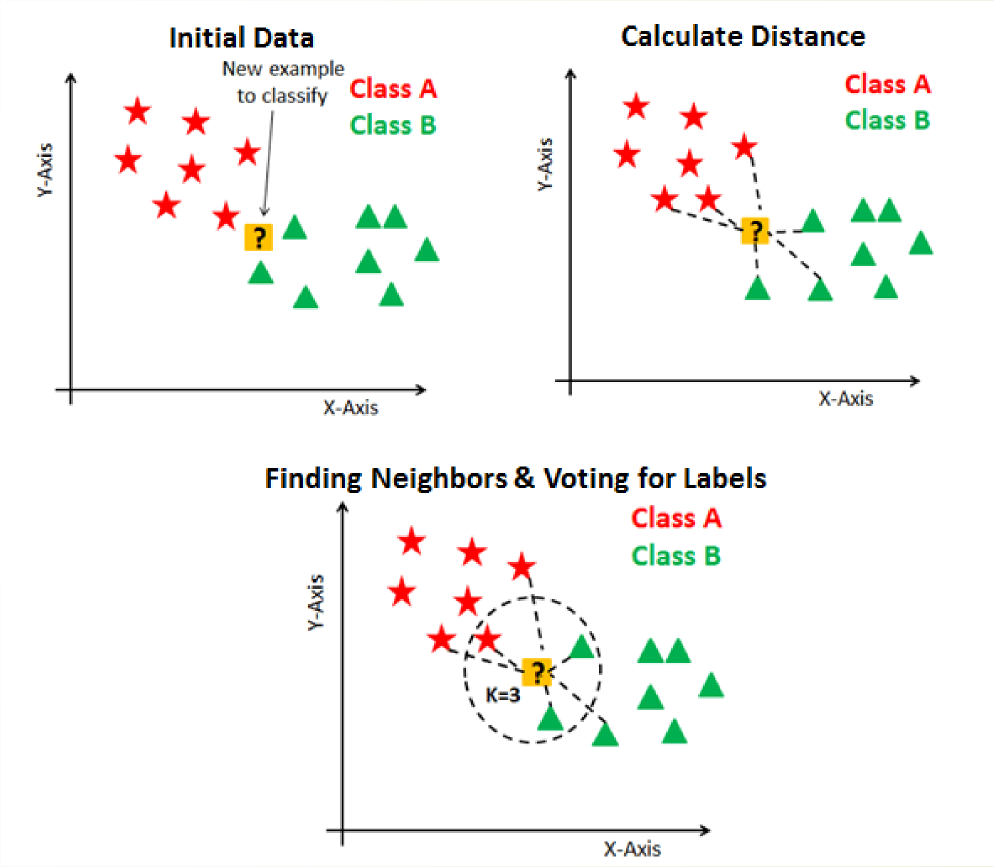
- Is a lazy algorithm (NB: not the same as lazily evaluated datastructure!)
  - It means: It doesn't require *training* before usage
  - It works directly on the given dataset, and builds a model that can be used to classify *new* data

KNN



KNN

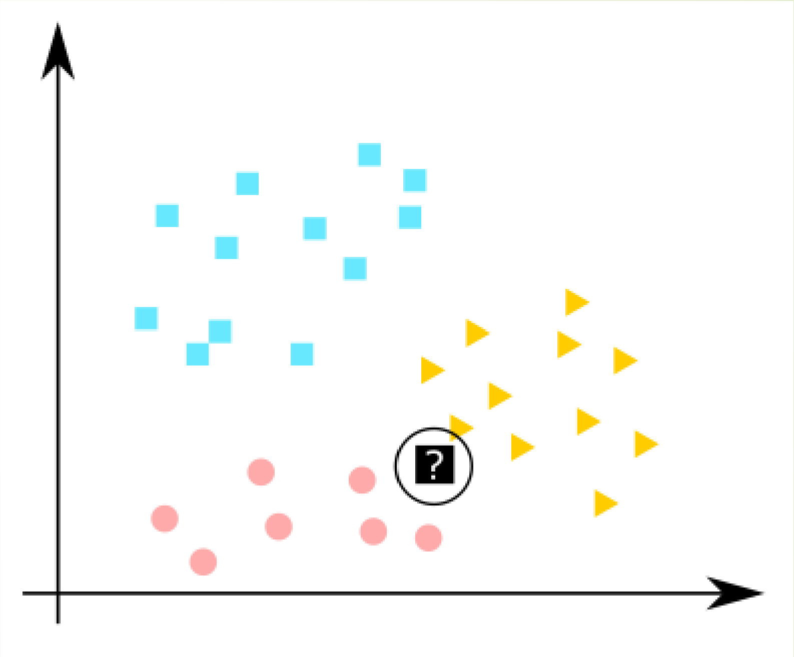
- For a new sample s:
- 1. Find its *k* nearest neighbors
- 2. Count how many belongs to each cluster
- 3. Assign the majority-cluster to s.



https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

KNN

- We only have to find out what to choose for *k*.
- Lower k: sharper boundaries.
- Higher k: smoother boundaries.



Let's try it!

- Our standard of work:

Pre processing

→ Training

→ Testing

→ Validation

KNN

```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


# KNN classification
iris = datasets.load_iris()


# Make the KNN classifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(iris.data, iris.target)

# Check how well it scores
knn.score(X_test, y_test)
```

KNN

```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


# KNN classification
iris = datasets.load_iris()


# Make the KNN classifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(iris.data, iris.target)

# Check how well it scores
knn.score(iris.data, iris.target)
```

What's the problem here?

KNN

```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


# KNN classification
iris = datasets.load_iris()


# Make the KNN classifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(iris.data, iris.target)

# Check how well it scores
knn.score(iris.data, iris.target)
```

What's the problem here?

We are basing the score on the trained data. We can do better!

KNN

```
from sklearn import datasets
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


# KNN classification
iris = datasets.load_iris()


# Split
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target)


# Make the KNN classifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Check how well it scores
knn.score(X_test, y_test)
```

KNN

- Curse of dimensionality:
  - KNN is heavily reliant on distance measure
  - The higher the dimensionality, the higher average distance between points

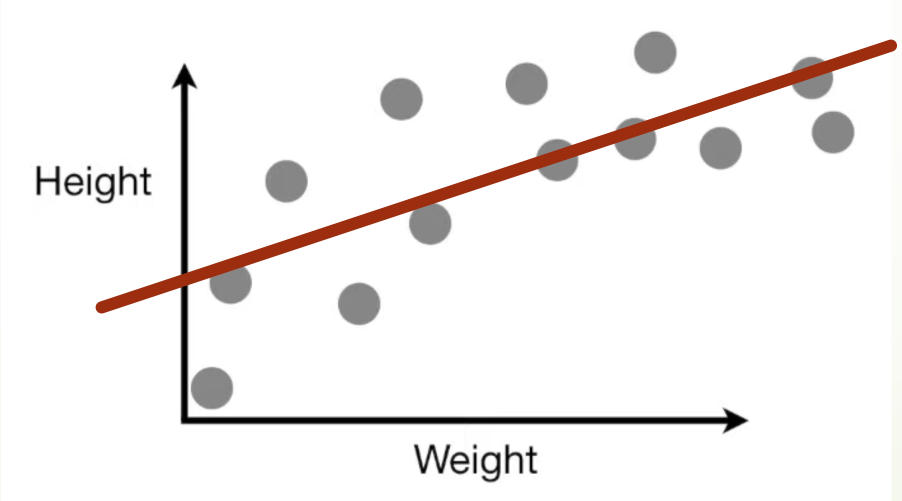
Bias and variance

- Bias: how well a model *can* solve a problem (high bias → poorly models the problem)
- Variance: How much the score varies if changing the dataset.

Bias and variance

- https://www.youtube.com/watch?v=EuBBz3bI-aA:

BIAS

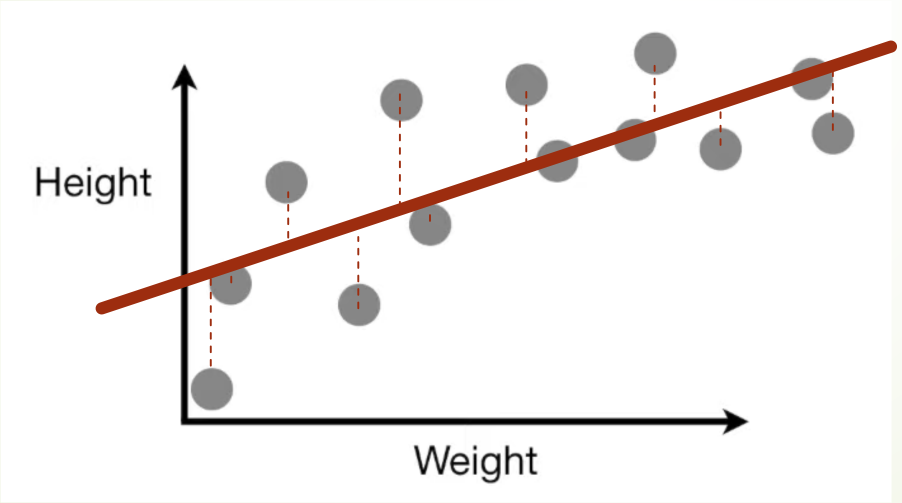


High bias!
(biased towards seeing the problem as a straight line)

Bias and variance

- https://www.youtube.com/watch?v=EuBBz3bI-aA:

BIAS

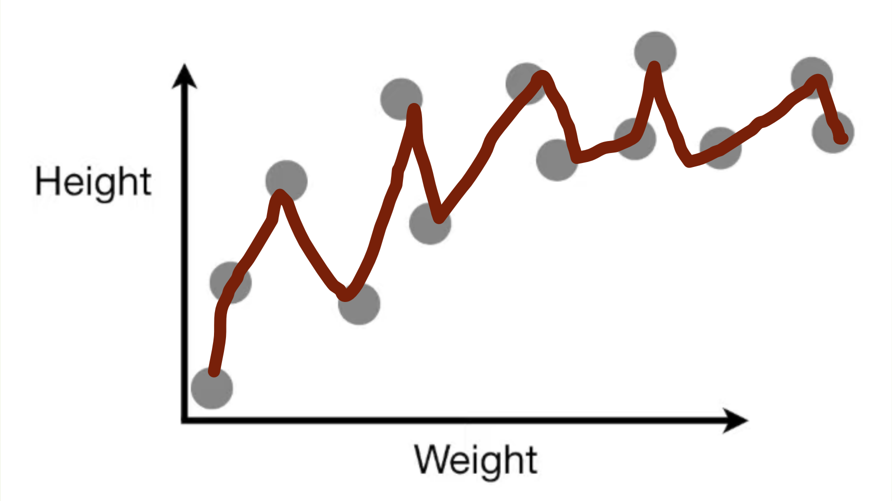


High bias!
(biased towards seeing the problem as a straight line)

Bias and variance

- https://www.youtube.com/watch?v=EuBBz3bI-aA:

BIAS

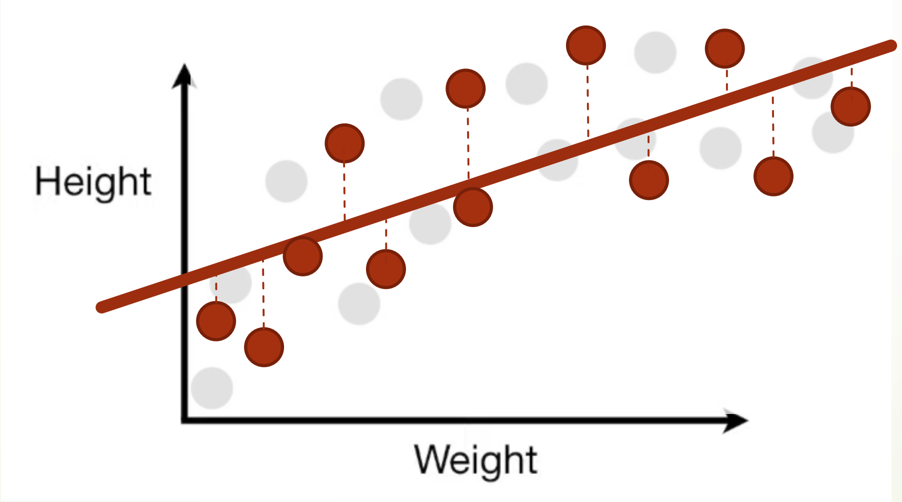


Low bias!
(This high-order polynomial can fit almost any dataset!)

Bias and variance

- https://www.youtube.com/watch?v=EuBBz3bI-aA:

VARIANCE

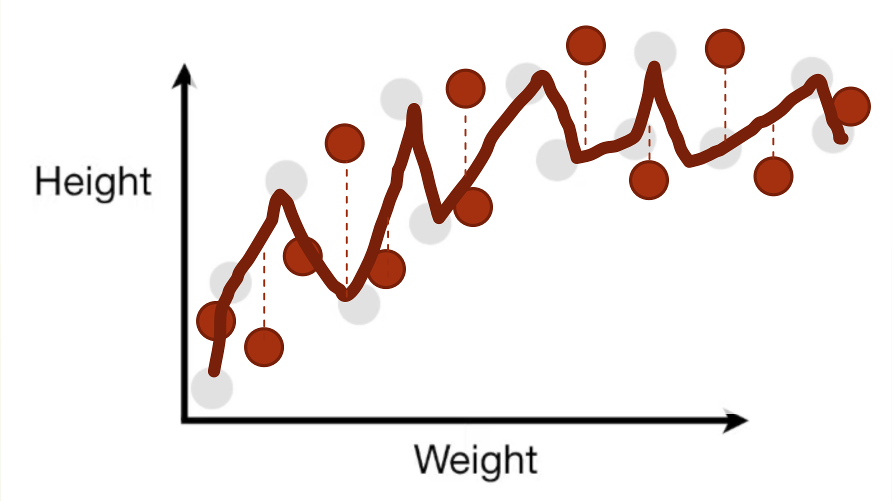


Low variance!
(A new dataset fits as well as the training set)

Bias and variance

- https://www.youtube.com/watch?v=EuBBz3bI-aA:

VARIANCE

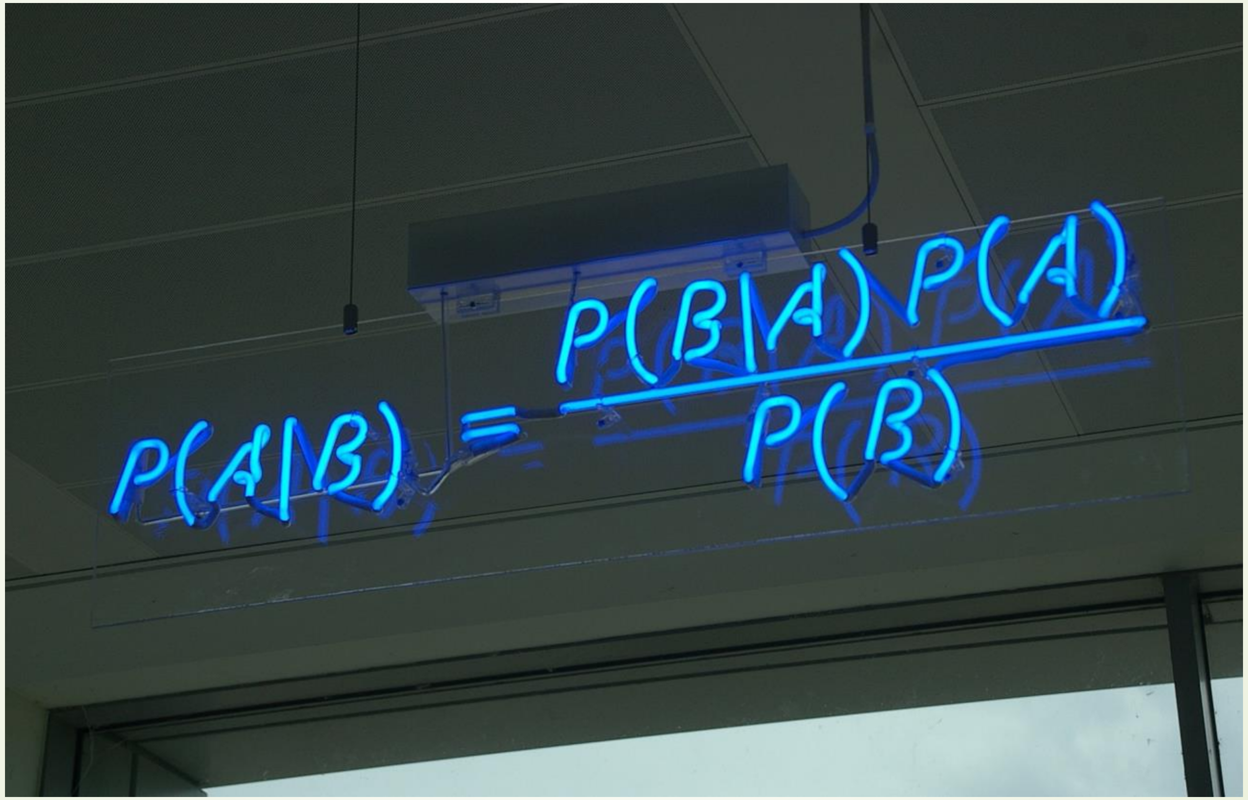


High variance!
(A new dataset fits terribly -- score varies a lot between datasets!)

Naive Bayes classifier

- Bayes?!?!?!? Bayes!

Bayes' rule



Bayes' theorem

- Thomas Bayes: statistician, philosopher, priest

He is known to have published two works in his lifetime, one theological and one mathematical:

1. *Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures* (1731)
2. *An Introduction to the Doctrine of Fluxions, and a Defence of the Mathematicians Against the Objections of the Author of The Analyst* (published anonymously in 1736), in which he defended the logical foundation of Isaac Newton's calculus ("fluxions") against the criticism by George Berkeley, a bishop and noted philosopher, the author of *The Analyst*

Thomas Bayes
(1701 - 1761)

Bayes' theorem

- " ... Bayes' theorem ... describes the probability of an event, based on prior knowledge of conditions that might be related to the event." (wikipedia)

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- $P(A \mid B)$: probability of A, given B is true
- $P(B \mid A)$: probability of B, given A is true
- $P(A)$: probability that A is true (prior probability)
- $P(B)$: probability that B is true (prior probability)
- A and B must be different events
- $P(B) \neq 0$

Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- " ... Bayes' theorem ... describes the probability of an event, based on prior knowledge of conditions that might be related to the event." (wikipedia)
- Suppose there is a school having 60% boys and 40% girls as students. The girls wear trousers or skirts in equal numbers; all boys wear trousers. **An observer sees a (random) student from a distance; all the observer can see is that this student is wearing trousers. What is the probability this student is a girl?** The correct answer can be computed using Bayes' theorem.

Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- " ... Bayes' theorem ... describes the probability of an event, based on prior knowledge of conditions that might be related to the event." (wikipedia)
- Suppose there is a school having 60% boys and 40% girls as students. The girls wear trousers or skirts in equal numbers; all boys wear trousers. **An observer sees a (random) student from a distance; all the observer can see is that this student is wearing trousers. What is the probability this student is a girl?** The correct answer can be computed using Bayes' theorem.

$$P(G \mid T) = \frac{P(T \mid G)\, P(G)}{P(T)}$$

Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- " ... Bayes' theorem ... describes the probability of an event, based on prior knowledge of conditions that might be related to the event." (wikipedia)
- Suppose there is a school having 60% boys and 40% girls as students. The girls wear trousers or skirts in equal numbers; all boys wear trousers. **An observer sees a (random) student from a distance; all the observer can see is that this student is wearing trousers. What is the probability this student is a girl?** The correct answer can be computed using Bayes' theorem.

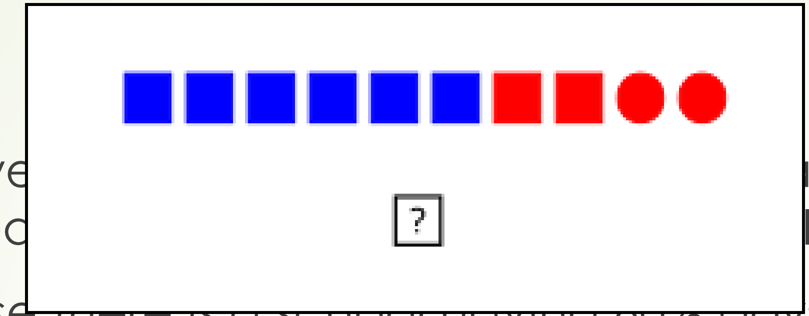
$$P(G \mid T) = \frac{P(T \mid G)\, P(G)}{P(T)}$$

$$\begin{aligned} P(T \mid G) &= 0.5 \\ P(G) &= 0.4 \\ P(T) &= 0.8 \end{aligned}$$

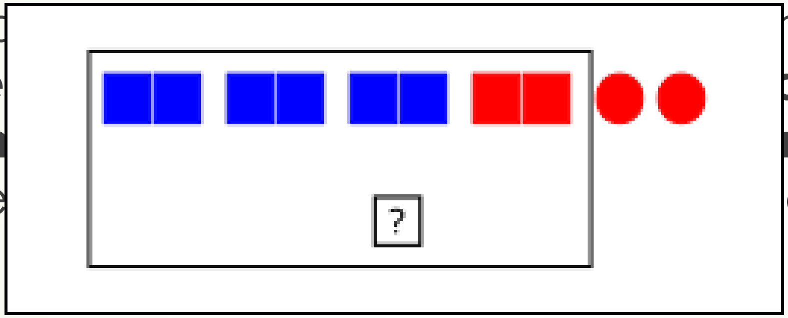
$$P(G \mid T) = \frac{0.5 \times 0.4}{0.8} = 0.25$$

Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$



- " ... Baye … bility of an event, based on prior knowled … lated to the event." (wikipedia)
- Suppose … s and 40% girls as students. The girls wea … rs; all boys wear trousers. **An observe … distance; all the observer can see is that th … t is the probability this student is a girl?** The … d using Bayes' theorem.



$$P(G \mid T) = \frac{P(T \mid G)\, P(G)}{P(T)}$$

P(T | G) = 0.5
P(G) = 0.4
P(T) = 0.8

$$P(G \mid T) = \frac{0.5 \times 0.4}{0.8} = 0.25$$

Bayes' theorem, example

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- **dangerous fires** are rare **(1%)**
- but **smoke** is fairly common **(10%)** due to barbecues,
- and **90%** of dangerous **fires make smoke**

Probability of dangerous Fire when there is Smoke:

- **P(Fire|Smoke) = ?**

Bayes' theorem, example

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)}$$

- **dangerous fires** are rare **(1%)**
- but **smoke** is fairly common **(10%)** due to barbecues,
- and **90%** of dangerous **fires make smoke**

Probability of dangerous Fire when there is Smoke:

- **P(Fire|Smoke) = P(Smoke|Fire) × P(Fire) / P(Smoke)**
  = 0.9 * 0.01 / 0.1
  = 0.09 = 9%

https://www.mathsisfun.com/data/bayes-theorem.html

Bayes' theorem, example

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

You are planning a picnic today, but the **morning is cloudy**

- Oh no! **50%** of all **rainy days start off cloudy**!
- But cloudy mornings are common (about **40%** of **days start cloudy**)
- And this is usually a dry month (only 3 of 30 days tend to be **rainy**, or **10%**)
- **What is the chance of rain during the day?**
- P(rain | cloudyMorning) = ?

https://www.mathsisfun.com/data/bayes-theorem.html

Bayes' theorem, example

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

You are planning a picnic today, but the **morning is cloudy**

- Oh no! **50%** of all **rainy days start off cloudy**!
- But cloudy mornings are common (about **40%** of **days start cloudy**)
- And this is usually a dry month (only 3 of 30 days tend to be **rainy**, or **10%**)
- **What is the chance of rain during the day?**
- P(rain | cloudyMorning) = P(cloudyMorning | rain) × P(rain) / P(cloudyMorning)

  = 0.5 × 0.1 / 0.4

  = 0.125 = 12.5% rain

https://www.mathsisfun.com/data/bayes-theorem.html

Naïve Bayes Classifier

- A classifier based on Bayes' theorem.
- It's naïve, because it assumes that all parameters are *independent*.
- Example: https://da.wikipedia.org/wiki/Naiv_Bayes_klassifikator

Logistic Regression classifier

Choosing between two models

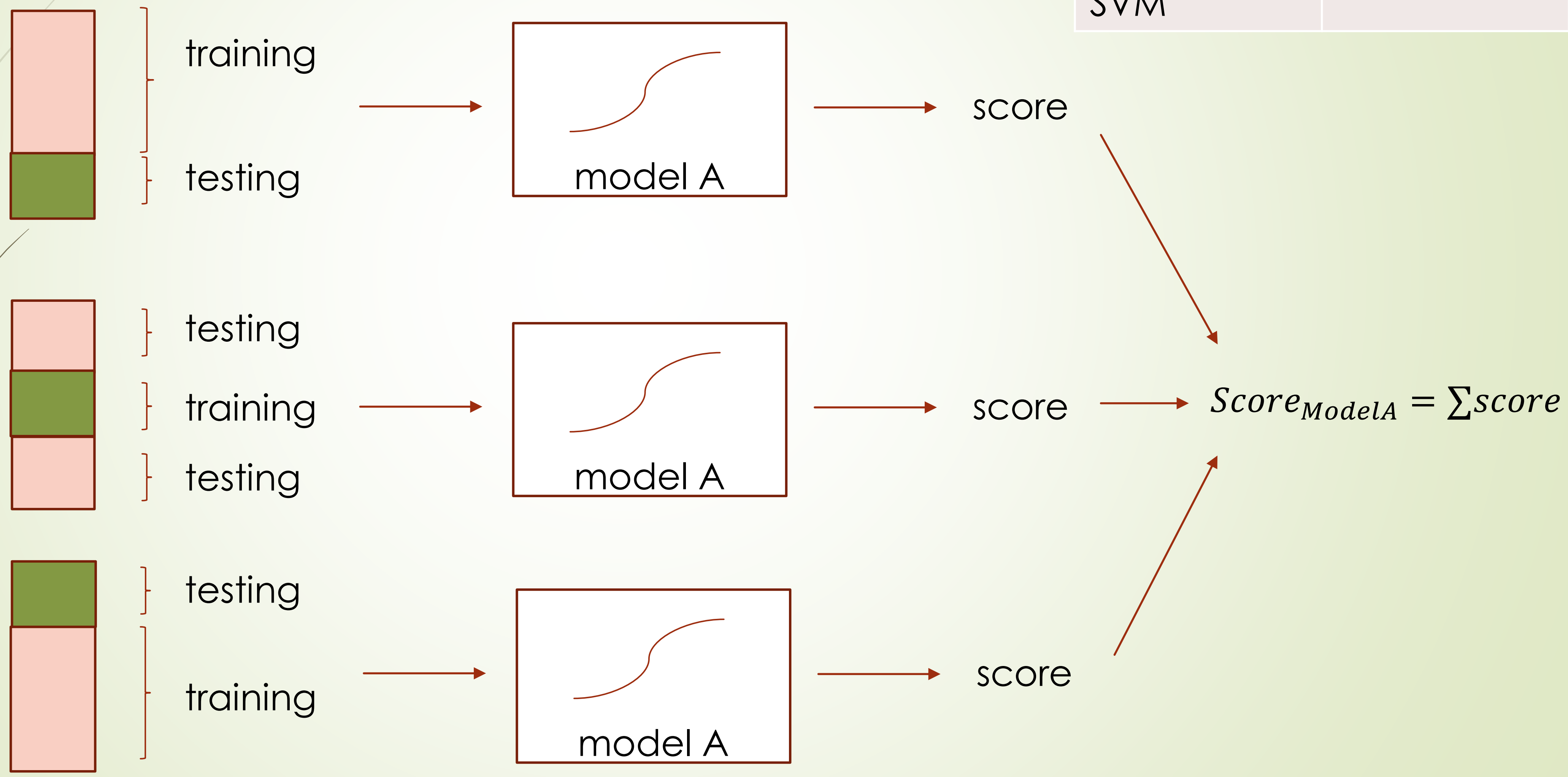
- Cross validation to the rescue!

Let's say we need to predict heart disease based on symptoms: chest pain, blood circulation, blocked arteries, weight, ...

We could use Logistic Regression, KNN, SVM, ... Which one do we choose?

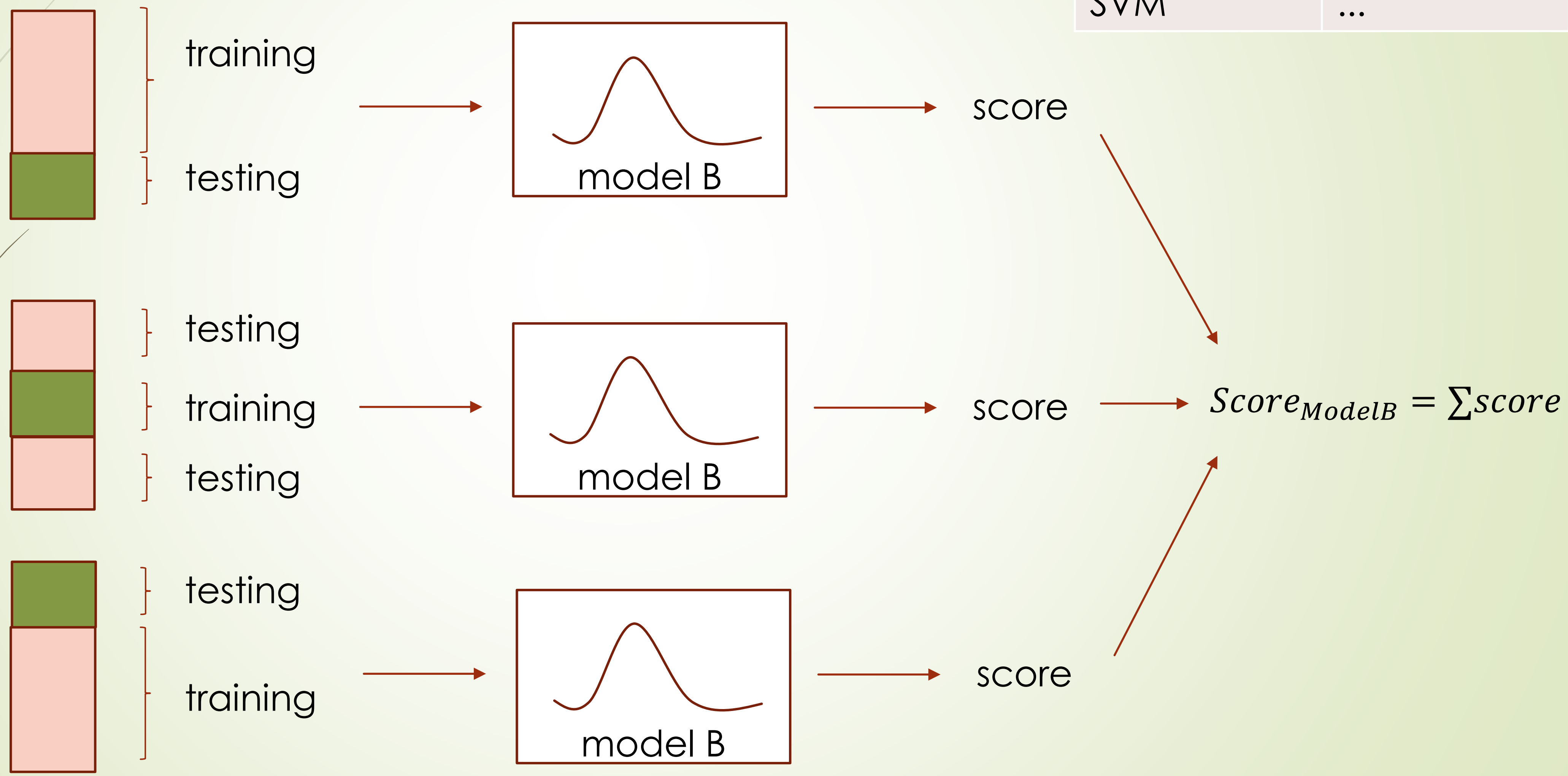
Cross validation

Model	$\sum$score
Logistic Reg	2.6
Naive Bayes	
KNN	
SVM	



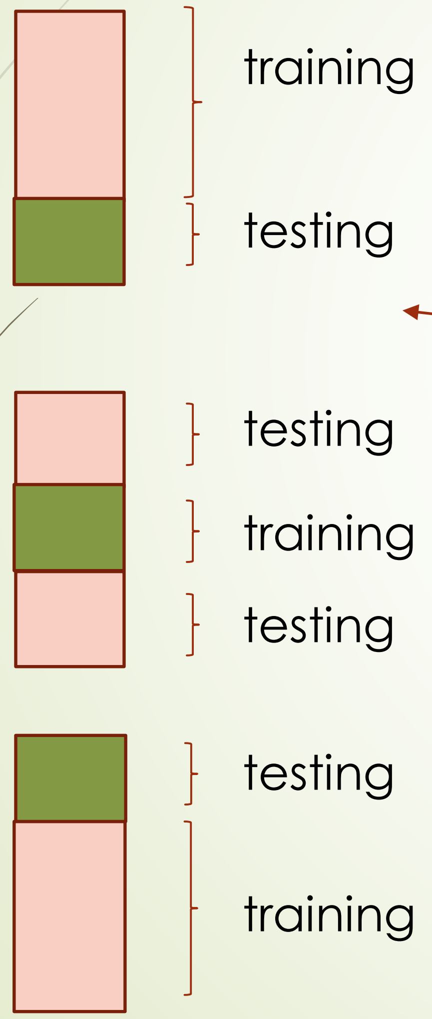
Cross validation

Model	∑score
Logistic Reg	2.6
Naive Bayes	2.4
KNN	…
SVM	…



Cross validation

Model	∑score
Logistic Reg	2.6
Naive Bayes	2.4
KNN	...
SVM	



This is a 3-fold cross validation. Data is divided into 3 blocks. Each block is tried as a test.

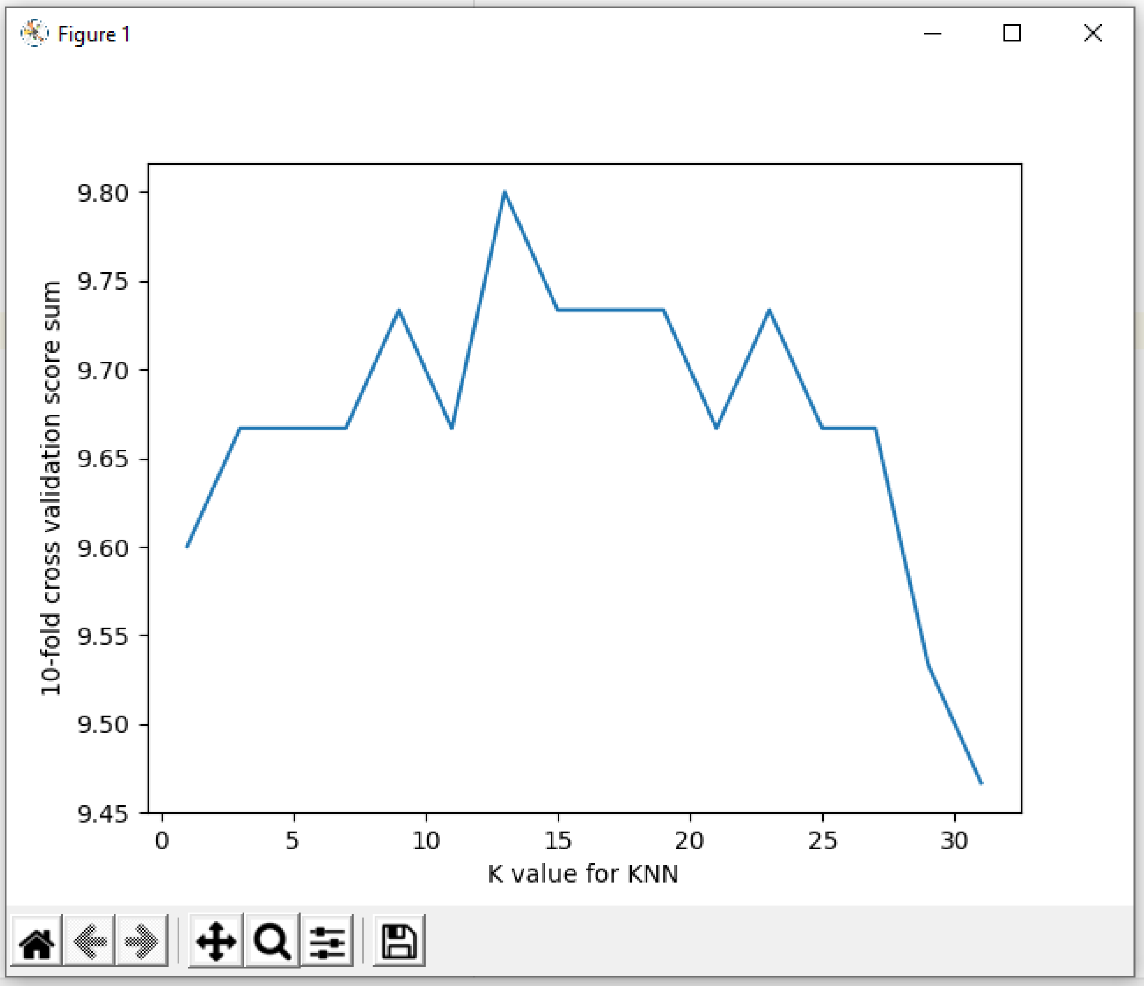
10-fold is common.

Another is called "leave-one-out cross validation", where one sample is left out as the single test sample.

Cross validation

```python

iris = load_iris()

allscores = []
allks = []
bestk = 1
bestsumcvscore = 0
for i in range(1,32,2):
    knn = KNeighborsClassifier(n_neighbors=i)
    cvscore = cross_val_score(knn, iris.data, iris.target, cv=10, scoring='accuracy')

    sumcvscore = sum(cvscore)

    allscores.append(sumcvscore)
    allks.append(i)

    if sumcvscore > bestsumcvscore:
        bestsumcvscore = sumcvscore
        bestk = i
    print('k (neighbors): ', i, ', Sum score:', round(sumcvscore,2))

print("best k value for knn: ", bestk,", with the score ", bestsumcvscore)

plt.plot(allks, allscores)
plt.xlabel("K value for KNN")
plt.ylabel("10-fold cross validation score sum")

```

for i in range(1,32,2)



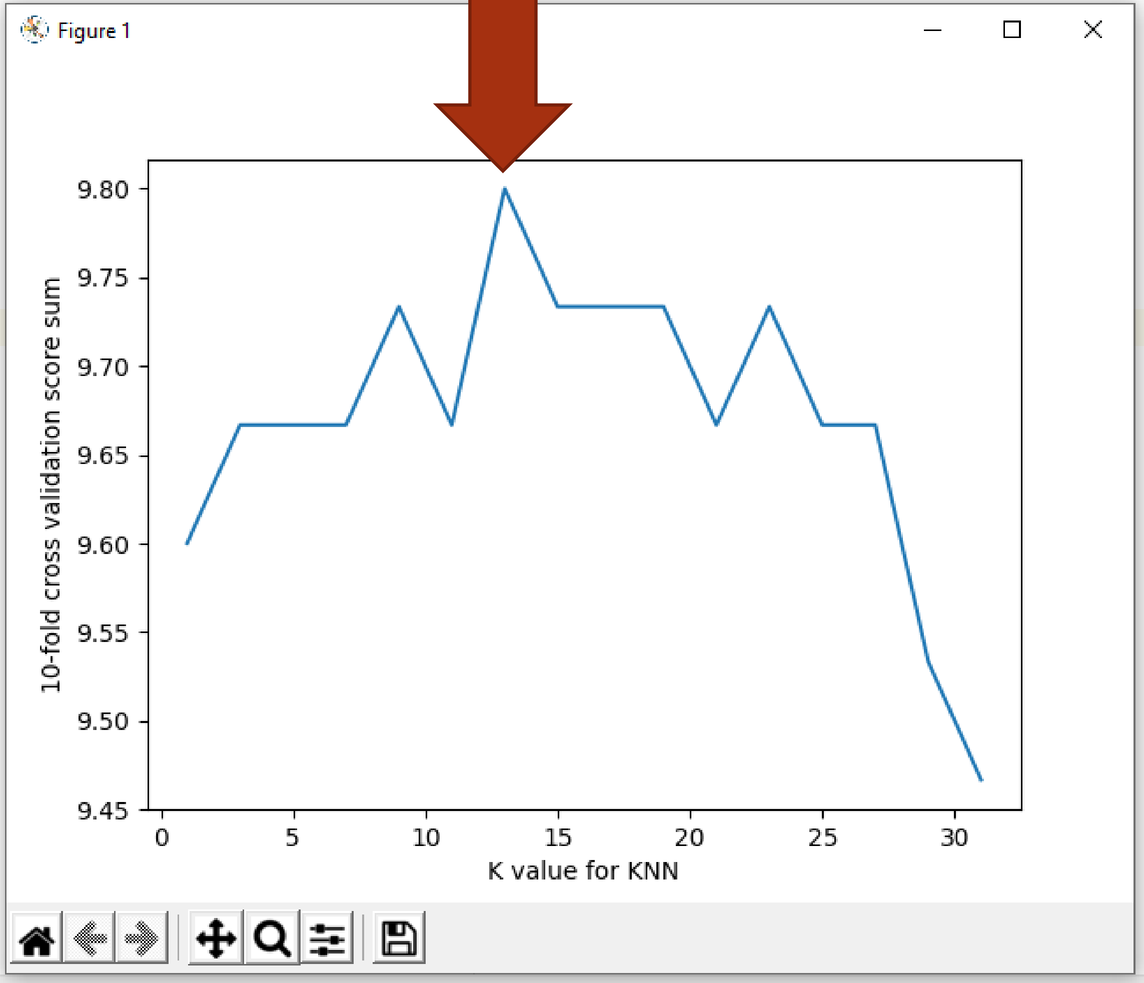
Cross validation

Tradeoff between bias and variance

(in this dataset!)

```python
iris = load_iris()

allscores = []
allks = []
bestk = 1
bestsumcvscore = 0
for i in range(1,32,2):
    knn = KNeighborsClassifier(n_neighbors=i)
    cvscore = cross_val_score(knn, iris.data, iris.target, cv=10, scoring='accuracy')

    sumcvscore = sum(cvscore)

    allscores.append(sumcvscore)
    allks.append(i)

    if sumcvscore > bestsumcvscore:
        bestsumcvscore = sumcvscore
        bestk = i
    print('k (neighbors): ', i, ', Sum score:', round(sumcvscore,2))

print("best k value for knn: ", bestk,", with the score ", bestsumcvscore)

plt.plot(allks, allscores)
plt.xlabel("K value for KNN")
plt.ylabel("10-fold cross validation score sum")
```

for i in range(1,32,2)



KNIME