Applied Machine Learning

Course number: W207

Prof. Alexander I. Iliev, Ph.D.

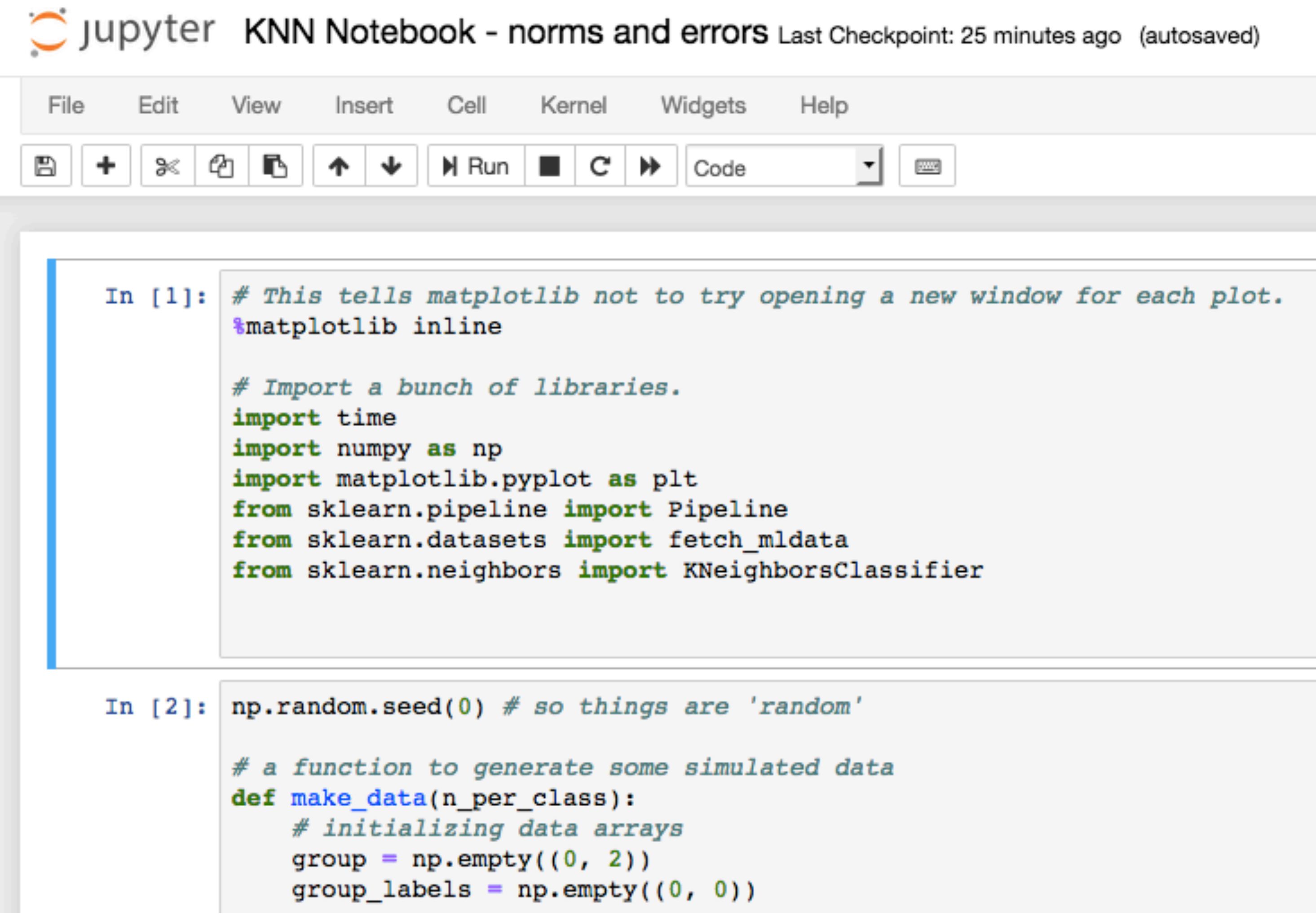
Applied Machine Learning

Lecture 2 …

- *Jupyter Notebook example*
- *Supervised vs Unsupervised*
- *How to split your data into Train / Validation / Test sets*

Jupyter Notebook

- Example:



Applied Machine Learning

Lecture 2 …

- *Lets define some concepts first …*

The weather problem

- Simple Examples:
  - The Weather Problem:

Outlook	Temperature	Humidity	Windy	Play
Sunny	hot	high	false	no
Sunny	hot	high	true	no
Overcast	hot	high	false	yes
Rainy	mild	high	false	yes
Rainy	cool	normal	false	yes
Rainy	cool	normal	true	no
Overcast	cool	normal	true	yes
Sunny	mild	high	false	no
Sunny	cool	normal	false	yes
Rainy	mild	normal	false	yes
Sunny	mild	normal	true	yes
Overcast	mild	high	true	yes
Overcast	hot	normal	false	yes
Rainy	mild	high	true	no

The weather problem

- Conditions for playing a certain game

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
…	…	…	…	…

A set of learned rules might look like this:

```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```

The weather problem

- Conditions for playing a certain game

attributes

instances

	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Machine Learning

- Simple Examples:
  - The Weather Problem:
    - *instances* in a dataset are characterized by the values of *features*, or *attributes* , that measure different aspects of the instance
    - In our case there are four attributes:
      - Outlook,
      - Temperature,
      - Humidity,
      - Windy.
    - The outcome is whether to play or not

Machine Learning

- Simple Examples:
  - The Weather Problem:
    - there are 36 possible combinations (3 x 3 x 2 x 2 = 36)
    - 14 of them are present in the set of input examples
    - These rules are meant to be interpreted in order:
      - If outlook = sunny and humidity = high then play = no
      - If outlook = rainy and windy = true then play = no
      - If outlook = overcast then play = yes
      - If humidity = normal then play = yes
      - If none of the above then play = yes

The weather problem

- Conditions for playing a certain game

attributes

instances

	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Rule:

if humidity = normal then play = yes

Correct or incorrect?

Machine Learning

- Simple Examples:
  - The Weather Problem:
    - It is possible to just look for any rules that strongly associate different attribute values
    - These are called *association rules*
    - Some of them are:

```
If temperature = cool                 then humidity = normal
If humidity = normal and windy = false then play = yes
If outlook = sunny and play = no      then humidity = high
If windy = false and play = no        then outlook = sunny and
                                      humidity = high
```

Machine Learning

- Simple Examples:
  - The Weather Problem: *Weather Data with Some Numeric Attributes*

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	false	no
Sunny	80	90	true	no
Overcast	83	86	false	yes
Rainy	70	96	false	yes
Rainy	68	80	false	yes
Rainy	65	70	true	no
Overcast	64	65	true	yes
Sunny	72	95	false	no
Sunny	69	70	false	yes
Rainy	75	80	false	yes
Sunny	75	70	true	yes
Overcast	72	90	true	yes
Overcast	81	75	false	yes
Rainy	71	91	true	no

Weather data with mixed attributes

- Some attributes have numeric values

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

```
If outlook = sunny and humidity > 83 then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity < 85 then play = yes
If none of the above then play = yes
```

Machine Learning

- Simple Examples:
  - The Weather Problem:
    - This scheme must create inequalities involving these new numeric attributes rather than simple equality tests as in the former case.
    - This is called a *numeric-attribute problem*
    - It is a mixed-attribute problem, because not all attributes are numeric
    - To compare, the first rule given earlier might take the form:
      - If outlook = sunny and humidity > 83 then play = no

compare

```
If outlook = sunny and play = no        then humidity = high
```

- These rules are *classification rules, they predict the classification*

Classification vs. association rules

- Classification rule:
  predicts value of a given attribute (the classification of an example)

```
If outlook = sunny and humidity = high
   then play = no
```

- Association rule:
  predicts value of arbitrary attribute (or combination)

```
If temperature = cool then humidity = normal
If humidity = normal and windy = false
   then play = yes
If outlook = sunny and play = no
   then humidity = high
If windy = false and play = no
   then outlook = sunny and humidity = high
```

Classification vs. association rules

attributes

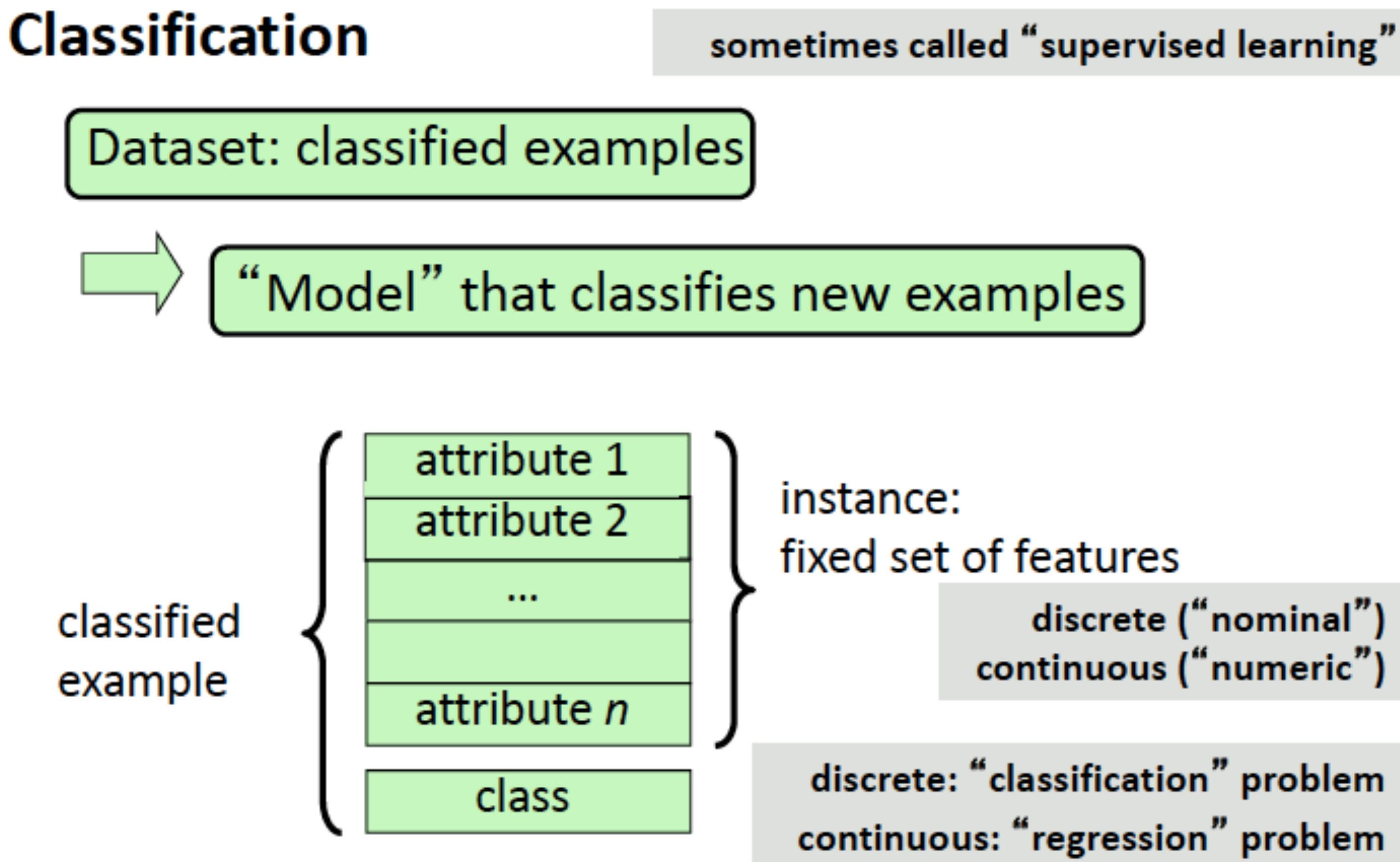
instances

Classification problem: predict the “class” value

	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	False	No
2	Sunny	Hot	[illegible]	True	No
3	Overcast	[illegible]	[illegible]	False	Yes
4	Rainy	[illegible]	[illegible]	False	Yes
5	[illegible]	[illegible]	Normal	False	Yes
[illegible]	[illegible]	[illegible]	Normal	True	No
[illegible]	[illegible]	[illegible]	Normal	True	Yes
[illegible]	[illegible]	Mild	[illegible]	[illegible]	No
[illegible]	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Weather data with mixed attributes

- Some attributes have numeric values



Applied Machine Learning

Lecture 2 …

- *Jupyter Notebook example*
- *Supervised vs Unsupervised*
- *How to split your data into Train / Validation / Test sets*

Machine Learning

- Supervised Machine Learning:
  - The majority of practical machine learning uses supervised learning

    Supervised vs. **Unsupervised Machine Learning**. ... At a high level, these **different** algorithms can be classified into two groups based on the way they "**learn**" about data to make predictions: **supervised** and **unsupervised learning**. **Supervised machine learning** is the more commonly used **between** the **two**. Jul 13, 2017

    Supervised vs. Unsupervised Machine Learning - DataScience.com
    https://www.datascience.com/.../supervised-and-unsupervised-machine-learning-algorith...

  - Supervised learning is to have input variables (x) and an output variable (Y) and we use an algorithm to learn the mapping function from the input to the output, hence finding: $Y = f(X)$
  - The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data
  - It is supervised learning because the process of learning from the training dataset can be thought of as a teacher supervising the learning process

Machine Learning

- Supervised Machine Learning:
  - The two groups of supervised learning are:
    - Classification: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”
    - Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”
  - Some popular examples of supervised machine learning algorithms are:
    - Linear regression for regression problems.
    - Random forest for classification and regression problems.
    - Support vector machines (SVM) for classification problems.

Machine Learning

- Unsupervised Machine Learning:
  - Unsupervised learning is where you only have input data (X) and no corresponding output variables.
  - The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data.
  - These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data.

Machine Learning

- Unsupervised Machine Learning:
  - Unsupervised learning problems can be further grouped into clustering and association problems:
    - Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
    - Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.
  - Some popular examples of unsupervised learning algorithms are:
    - k-means for clustering problems.
    - A priori algorithm for association rule learning problems.

Machine Learning

- Semi-Supervised Machine Learning:
  - Problems where we have a large amount of input data (X) and only some of it is labeled (Y) are called semi-supervised learning problems
  - These problems sit in between both supervised and unsupervised learning.
  - A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.

Machine Learning

- Semi-Supervised Machine Learning:
  - Many real world machine learning problems fall into this area
  - It can be expensive or time-consuming to label data as it may require access to domain experts
  - Hence unlabeled data is cheap and easy to collect and store

Machine Learning

- Semi-Supervised Machine Learning:
  - You can use unsupervised learning techniques to discover and learn the structure in the input variables
  - You can also use supervised learning techniques to make best guess predictions for the unlabeled data, feed that data back into the supervised algorithm as training data and use the model to make predictions on new unseen data

Applied Machine Learning

Lecture 2 …

- *Jupyter Notebook example*
- *Supervised vs Unsupervised*
- *How to split your data into Train / Validation / Test sets*

Holdout estimation

- What should we do if we only have a single dataset?
- The *holdout* method reserves a certain amount for testing and uses the remainder for training, after shuffling
  - Usually: one third for testing, the rest for training
- Problem: the samples might not be representative
  - Example: class might be missing in the test data
- Advanced version uses *stratification*
  - Ensures that each class is represented with approximately equal proportions in both subsets

Repeated holdout method

- Holdout estimate can be made more reliable by repeating the process with different subsamples

  - In each iteration, a certain proportion is randomly selected for training (possibly with *stratificiation*)
  - The error rates on the different iterations are averaged to yield an overall error rate

- This is called the *repeated holdout* method

- Still not optimum: the different test sets overlap
  - Can we prevent overlapping?

Cross-validation

- *K-fold cross-validation* avoids overlapping test sets
  - First step: split data into *k* subsets of equal size
  - Second step: use each subset for testing, the remainder for training
  - This means the learning algorithm is applied to *k* different training sets
- Often the subsets are *stratified* before the cross-validation is performed to yield stratified *k*-fold cross-validation
- The error estimates are averaged to yield an overall error estimate; also, standard deviation is often computed
- Alternatively, predictions and actual target values from the *k* folds are pooled to compute one estimate
  - Does not yield an estimate of standard deviation

More on cross-validation

- Standard method for evaluation is:

  stratified ten-fold cross-validation

- Why ten?
  - Extensive experiments have shown that this is the best choice to get an accurate estimate
  - There is also some theoretical evidence for this

- Stratification reduces the estimate's variance

- Even better: repeated stratified cross-validation
  - E.g., ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

Leave-one-out Cross-Validation

- Leave-one-out:
  is a particular form of *k*-fold cross-validation (CV):
  - Set number of folds to = number of training instances
  - I.e., for *n* training instances, build classifier *n* times
- Makes best use of the data (especially when small set)
- Involves no random subsampling
- Very computationally expensive (exception: using lazy classifiers such as the nearest-neighbor classifier)

Leave-one-out CV and stratification

- Disadvantage of Leave-one-out CV:

  stratification is not possible

  - In fact, it *guarantees* a non-stratified sample because there is only one instance in the test set!

- Extreme example:

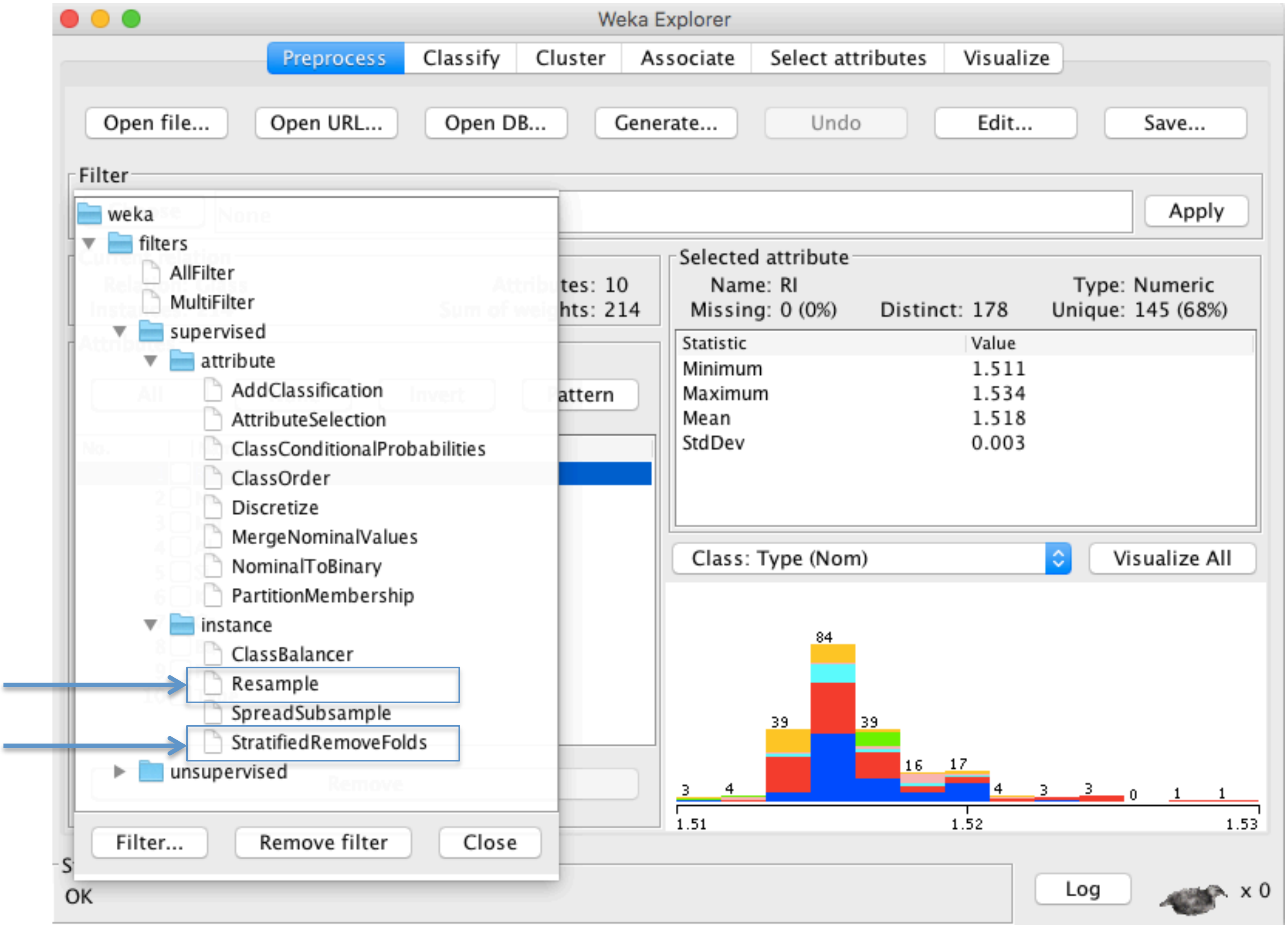
  random dataset split equally into two classes

  - Best inducer predicts majority class
  - 50% accuracy on fresh data
  - Leave-one-out CV estimate gives 100% error!

The bootstrap

- CV uses sampling *without replacement*
    - The same instance, once selected, can not be selected again for a particular training/test set
- The *bootstrap* uses sampling *with replacement* to form the training set, also known as *bagging*
    - Sample a dataset of n instances n times *with replacement* to form a new dataset of n instances
    - Use this data as the training set
    - Use the instances from the original dataset that do not occur in the new training set for testing

Weka examples



Weka examples