Big Bio-Data Analysis (Artificial Intelligence and Machine Learning)

03 March 2023

Introduction to Machine Learning Algorithms

By

Richard Sserunjogi
Department of Computer Science,
Makerere University, Uganda
sserurich@gmail.com

AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS & DATA INTENSIVE SCIENCE

Some Housekeeping

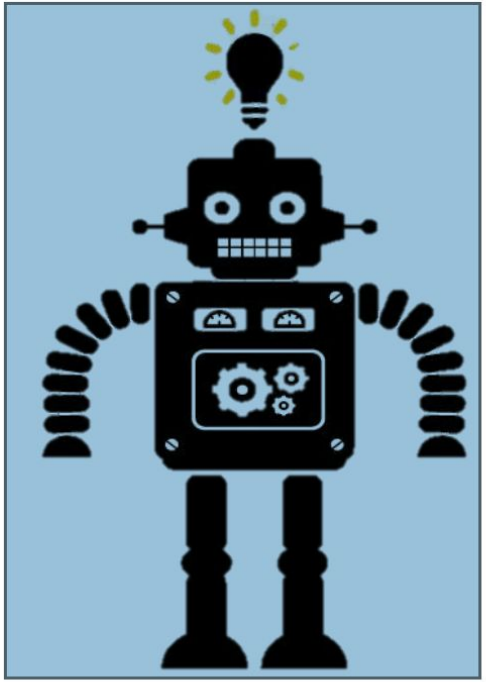
- Today we will be running examples using Weka
- If you are using a classroom computer, please make sure you can open the programs
- If you are using your own laptop, please install Weka if you do not have it already:
- Weka (requires Java): https://www.cs.waikato.ac.nz/ml/weka/

Slides, Code and Datasets

- Slides, Code & Datasets shall be made available on GitHub
  https://github.com/sserurich/ace-big-bio-data-analysis
- Each dataset is available in CSV (Weka) and ARFF (Weka-native) file format

What is Machine Learning (ML)?

- Machine learning allows computers to learn and infer from data
- The field of ML stems from research on Artificial Intelligence (AI) in the 1950’s



What is Machine Learning?

- Back then the goal was to replicate tedious human tasks using explicit rules (algorithms):

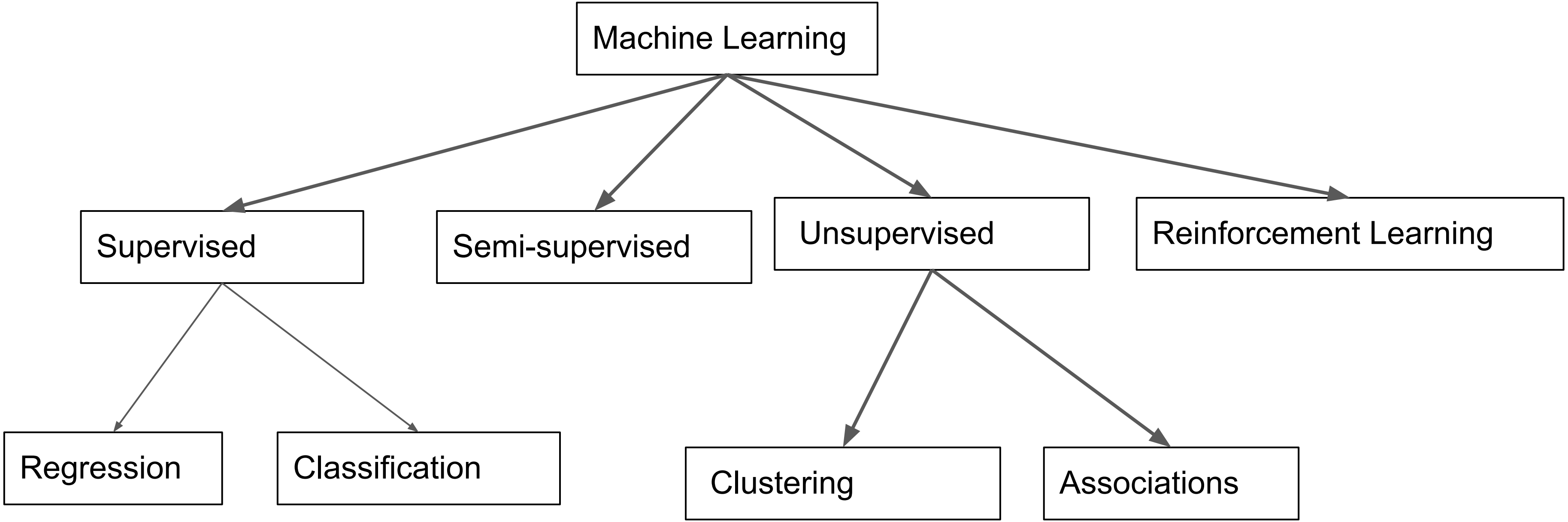
  [Data + Rules => Answers],

- ML’s goal is to have a computer learn to *teach itself* an algorithm that produces a useful *approximation* of data:

  [Data + Answers => Rules]

Buzzwords like “Deep Learning,” “Decision Trees,” ”Support Vector Machines,” etc. are *subfields* of ML.

Machine Learning Taxonomy



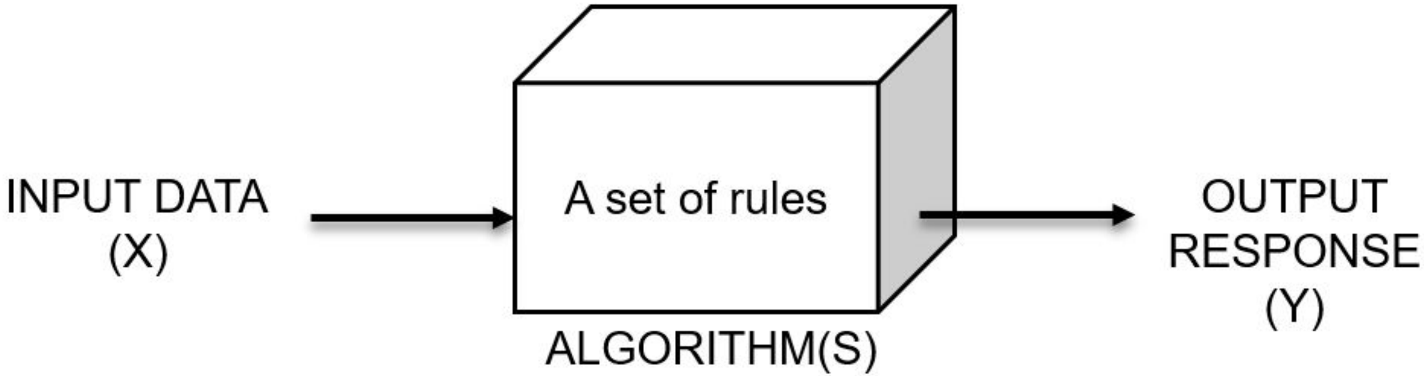
Machine Learning in Our Daily Lives

- SPAM FILTERING
- WEB SEARCH
- POSTAL MAIL ROUTING
- FRAUD DETECTION
- MOVIE RECOMMENDATIONS
- VEHICLE DRIVER ASSISTANCE
- WEB ADVERTISEMENTS
- SOCIAL NETWORKS
- SPEECH RECOGNITION

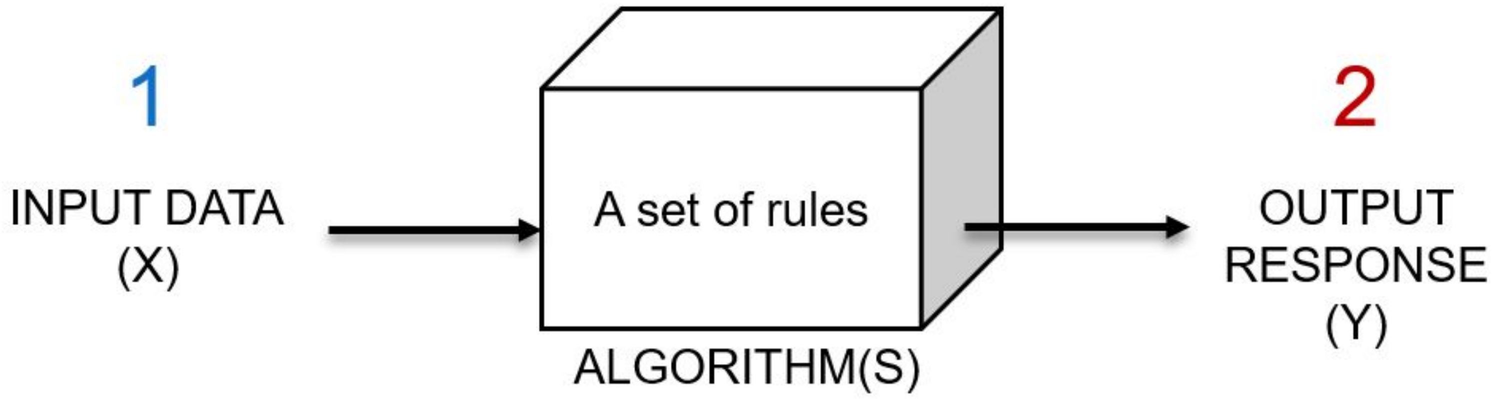
Some Examples of ML in Bioinformatics

- Cluster patients with similar phenotypes
- Predict the function of a newly discovered protein or virus
- Identify promising targets for new drug development from a huge database of compounds
- Predict whether someone has diabetes or not
- Any other examples?

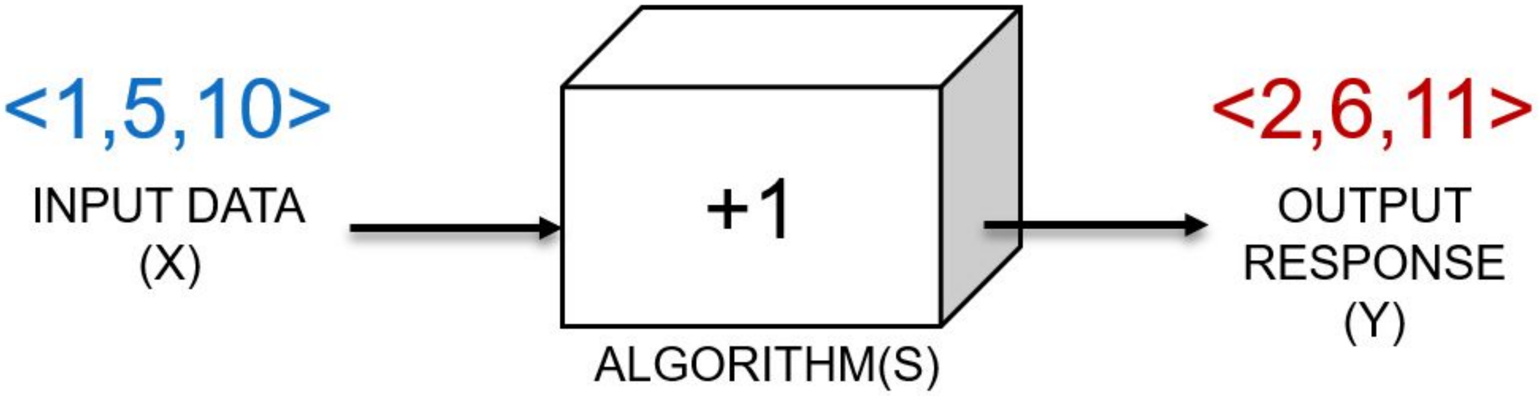
Basic Idea of an Algorithm



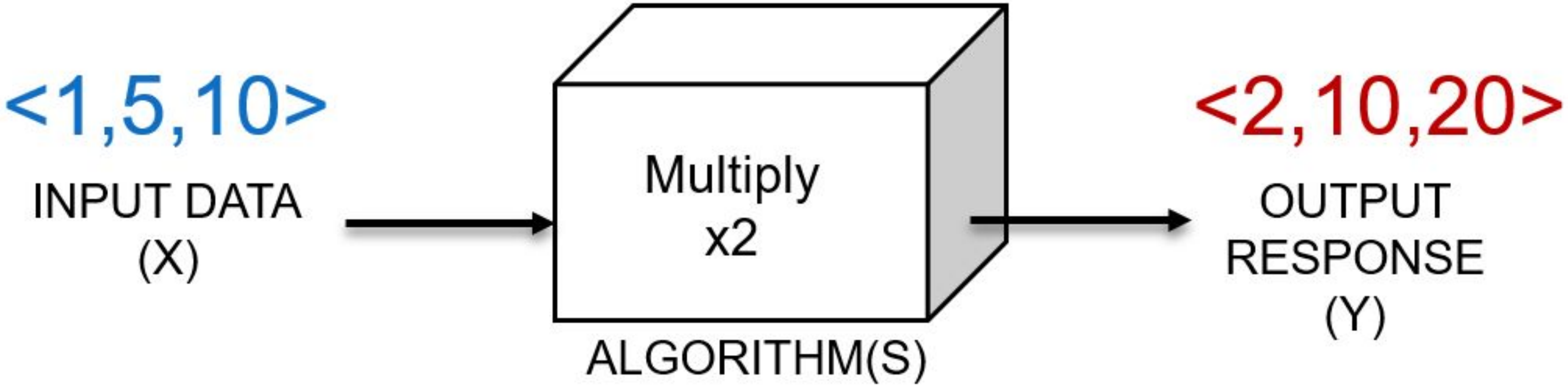
Basic Idea of an Algorithm



Basic Idea of an Algorithm



Basic Idea of *Learning* an Algorithm

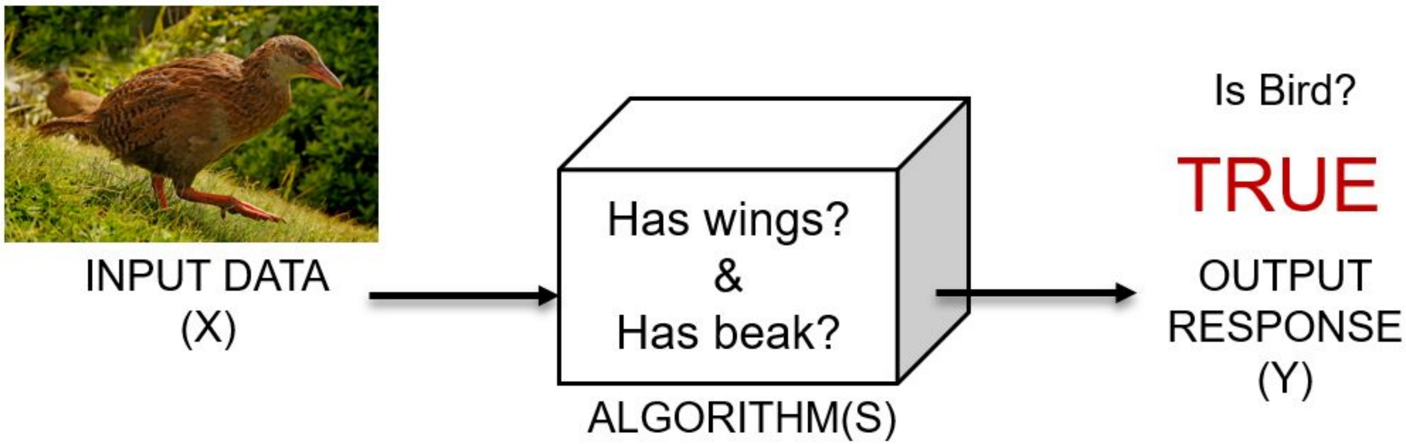


This is a basic regression problem!

We're fitting a function that maps INPUT to OUTPUT

$$y = 2x + 0$$

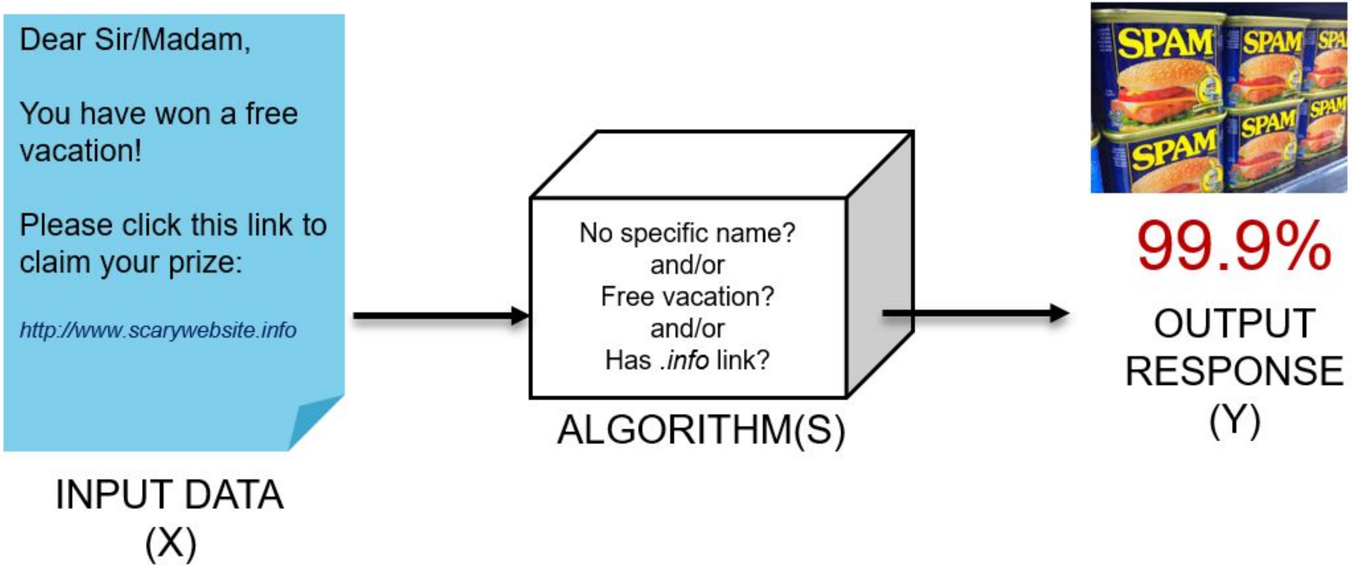
Basic Idea of *Learning* an Algorithm



This is a classification problem!

Typically these produce binary (limited to 2 topics) or categorical (multi-topic) outputs

Basic Idea of *Learning* an Algorithm



This is another classification problem with numerical (prediction probabilities) output

Types of Machine Learning

SUPERVISED	Data points have known outcome
UNSUPERVISED	Data points have unknown outcome

Supervised Learning

- Some of the previous examples of ML algorithms were “**supervised learning**” – where we train our approach using both example INPUT and example OUTPUT (label or class).
- You can now take your trained approach (a “model”) and use it to make predictions on new (unseen) data!
- Examples: Classifying

Types of Supervised Learning

REGRESSION	Outcome is continuous (numerical)
CLASSIFICATION	Outcome is a category

Unsupervised Learning

- What if we do not know the OUTPUT labels for our data?

- In “unsupervised learning” – we apply a specific set of rules to the INPUT to identify trends/patterns in the data.

- Examples: Clustering, trend/topic detection, outlier identification, dimensionality reduction

Types of Unsupervised Machine Learning

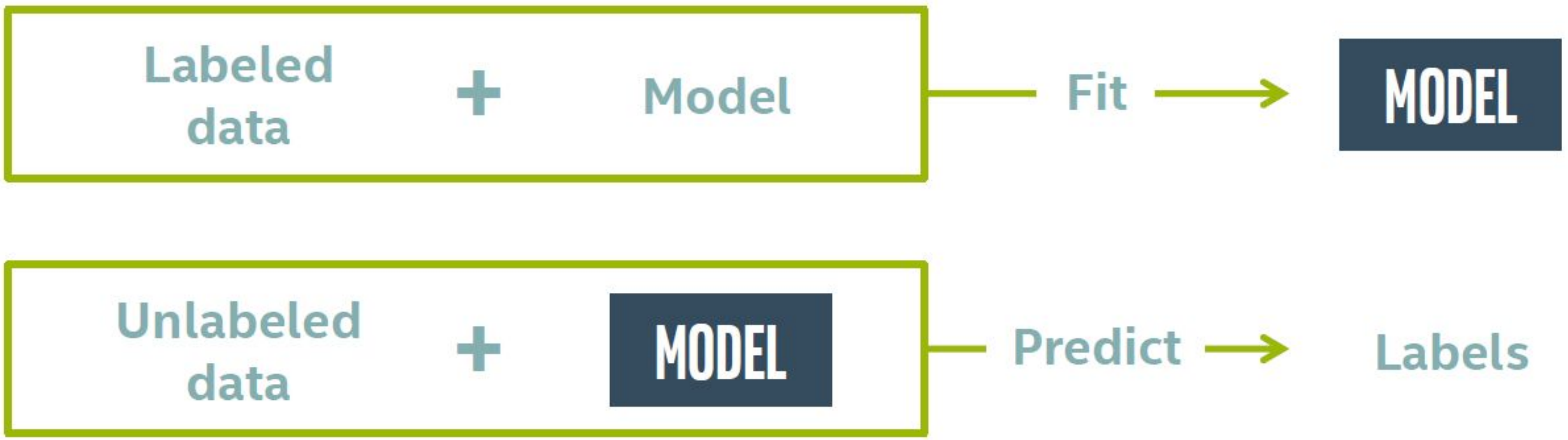
CLUSTERING	Identify unknown structure in data
DIMENSIONALITY REDUCTION	Use structural characteristics to simplify data

Classification in Machine Learning

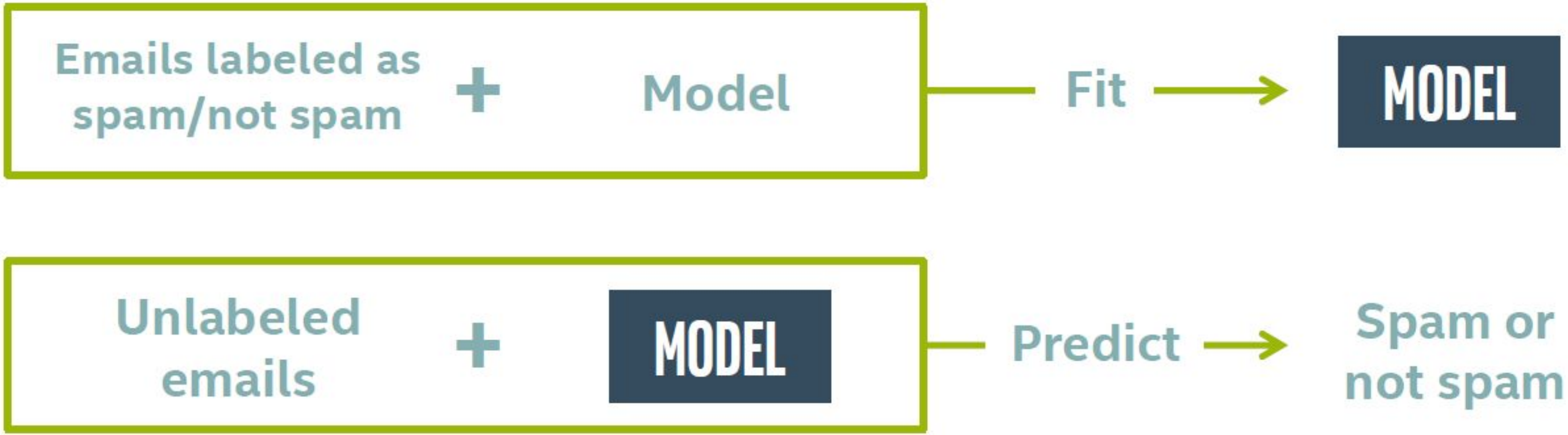
- Classification is a predictive modeling problem where a class label is predicted for a given example of input data

- Assign input vector to one of two or more classes

Classification in Machine Learning



Classification in Machine Learning



Classification : Terminology

- **Target:** predicted category the data (column to predict)
- **Features:** properties of the data used for prediction (non-target columns)
- **Example:** a single data point within the data (one row)
- **Label:** the target value for a single data point

Classification : Terminology

Example using the Iris Dataset

	Sepal length	Sepal width	Petal length	Petal width	Species
	6.7	3.0	5.2	2.3	Virginica
	6.4	2.8	5.6	2.1	Virginica
Examples →	4.6	3.4	1.4	0.3	Setosa
	6.9	3.1	4.9	1.5	Versicolor
	4.4	2.9	1.4	0.2	Setosa
	4.8	3.0	1.4	0.1	Setosa
	5.9	3.0	5.1	1.8	Virginica
	5.4	3.9	1.3	0.4	Setosa
	4.9	3.0	1.4	0.2	Setosa
	5.4	3.4	1.7	0.2	Setosa

Types of Classification in ML

- Binary Classification
  - Classification tasks that have two class labels
  - *Examples: spam or not, cancer detected/not detected*
- Multi-Class Classification
  - Classification tasks that have more than two class labels.
  - *Examples: Plant species classification, Optical character recognition*

Types of Classification in ML

- Multi-Label Classification
  - Classification tasks that have two or more class labels, where one or more class labels may be predicted.
  - *Example: photo classification*
- Imbalanced Classification
  - Classification tasks where the number of examples in each class is unequally distributed.
  - *Examples: Fraud detection, Medical diagnostic test*

Classification Algorithms

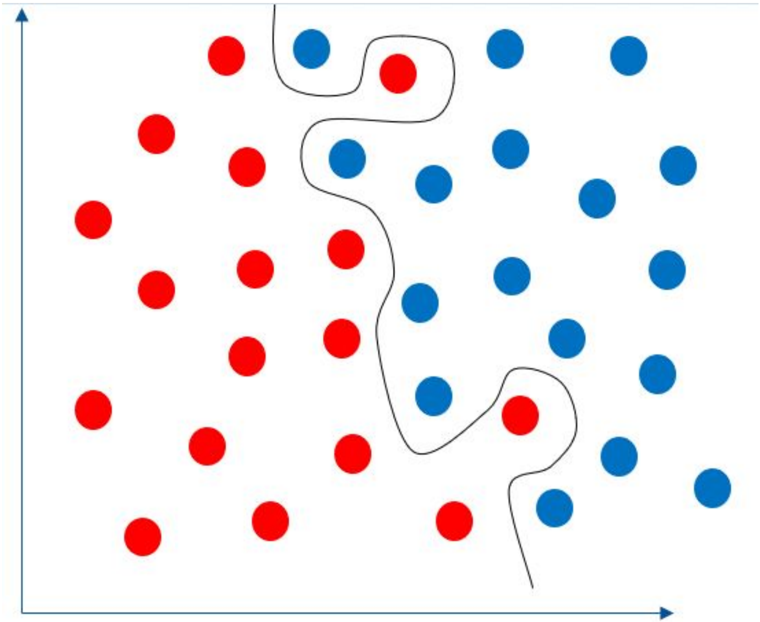
- Logistic Regression
- k-Nearest Neighbors
- Decision Trees
- Support Vector Machine
- Naive Bayes
- Gradient Boosting
- Neural networks
- Others

Setting Up a Supervised ML Experiment

- Gather representative data
- Split into Training, Tuning and Testing partitions
- Train a model and use Tuning partition like “Testing” while adjusting model parameters
- Merge Training + Tuning and Train your final model
- Run final model on Testing partition and evaluate

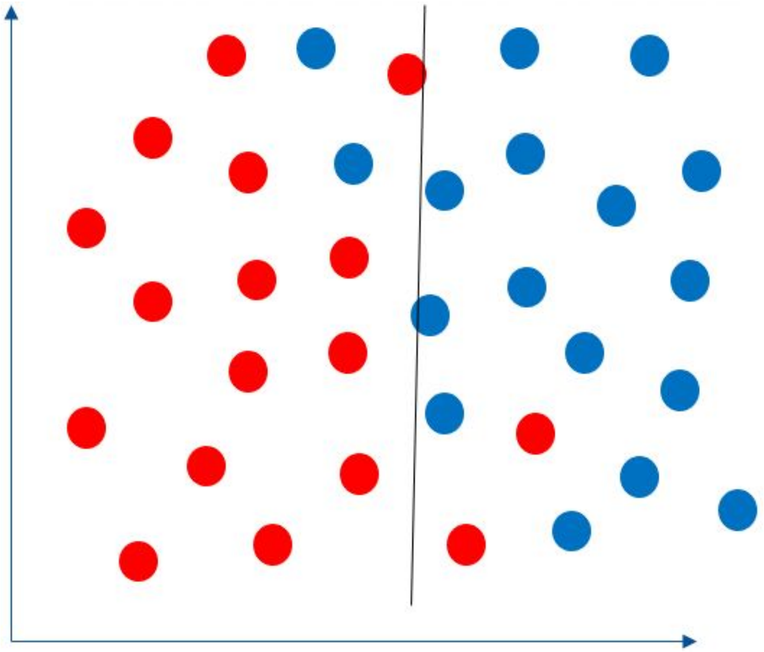
How Much Training is Enough?

- **Over-fitting:** Training our model too much!
  We are making it fit the training data so well that it does not *generalize* well to new (unseen) points



How Much Training is Enough?

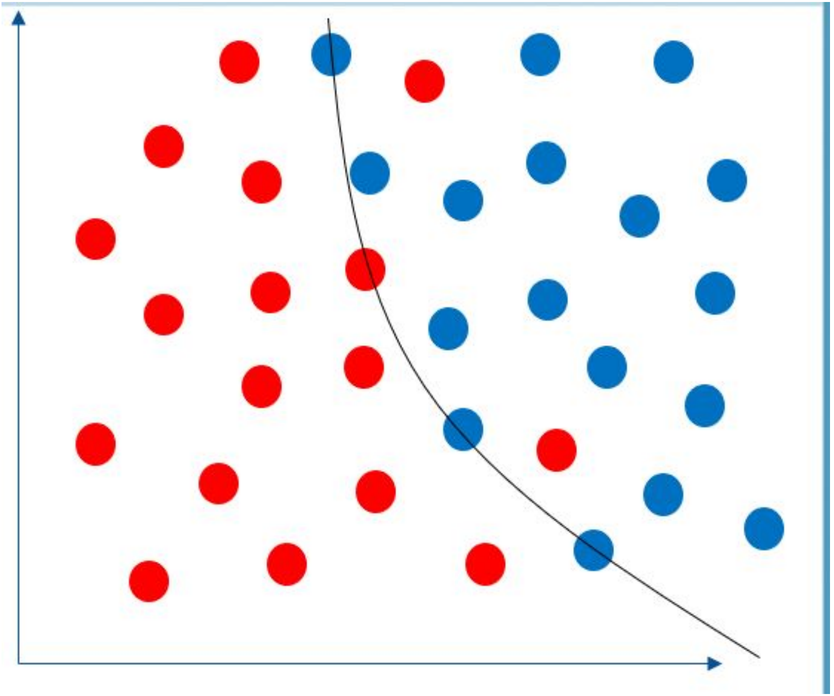
Under-fitting: Our model is missing key parameters needed to model the data- typically this means you need to perform more training



How Much Training is Enough?

Generalization: How well does our model perform on new (unseen) data? Often more training data helps!

Regularization: Adding information (or a penalty) to a score/ loss function to make a model perform better.
For example, if we allow a few misses in training set, we might actually generalize better to the testing set!



Evaluating Model Performance

Choosing the Right Error Measurement is key for machine learning problems

For algorithms that produce ***real-valued numerical predictions*** we can calculate the difference between predictions and the actual ans $\sum_{i=1}^{n}(y_i - \bar{y})^2$ ple, sum of squares:

Common performance metrics for ***binary predictions*** (TRUE/FALSE, 0/1, A/B, etc.) are based on the number of true positives (**TP**), true negatives (**TN**), false positives (**FP**) and false negatives (**FN**):

Evaluating Model Performance

```
=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         220               76.9231 %
Incorrectly Classified Instances        66               23.0769 %
Kappa statistic                          0.3506
Mean absolute error                      0.3536
Root mean squared error                  0.4205
Relative absolute error                 84.5297 %
Root relative squared error             92.0024 %
Total Number of Instances              286

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.945    0.647    0.776      0.945    0.852      0.389    0.650     0.773     no-recurrence-events
                 0.353    0.055    0.732      0.353    0.476      0.389    0.650     0.460     recurrence-events
Weighted Avg.    0.769    0.471    0.762      0.769    0.740      0.389    0.650     0.680

=== Confusion Matrix ===

   a   b   <-- classified as
 190  11 |   a = no-recurrence-events
  55  30 |   b = recurrence-events
```

WEKA

RESULTS

Evaluating Model Performance

- You are asked to build a classifier for leukemia
- Training data: 1% patients with leukemia, 99% healthy
- Measure accuracy: total % of predictions that are correct
- Build a simple model that always predicts “healthy”
- Accuracy will be 99%…

Evaluation Metrics

Sensitivity: $^{TP}/_{TP+FN}$

Specificity: $^{TN}/_{TN+FP}$

Specificity: $^{TP}/_{TP+FP}$

Accuracy: $^{TP+TN}/_{TP+TN+FP+FN}$

F-Measure: $^{2TP}/_{2TP+FP+FN}$

Matthews Correlation Coefficient (MCC): $\frac{TP * TN - FP * FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Classification : Error Metrics

Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Classification : Error Metrics

Accuracy : Predicting Correctly

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

Classification : Error Metrics

Recall /Sensitivity: Identifying All Positive Instances

What proportion of actual positives was identified correctly?

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Recall or Sensitivity} = \frac{TP}{TP + FN}$$

Classification : Error Metrics

Others:

- *Classification Report*
- Receiver Operator Curve (ROC)

KNeighborsClassifier : The Syntax

Import the class containing the classification method

```
from sklearn.neighbors import KNeighborsClassifier
```

Create an instance of the class

```
KNN= KNeighborsClassifier(n_neighbors=3)
```

Fit the instance on the data and then predict the expected value

```
KNN= KNN.fit(X_data, y_data)
y_predict = KNN.predict(X_data)
```

Logistic Regression : The Syntax

Import the class containing the regression method

```
from sklearn.linear_model import LogisticRegression
```

Create an instance of the class

```
LR= LogisticRegression(random_state=0)
```

Fit the instance on the data and then predict the expected value

```
LR= LR.fit(X_train, y_train)
y_predict= LR.predict(X_test)
```

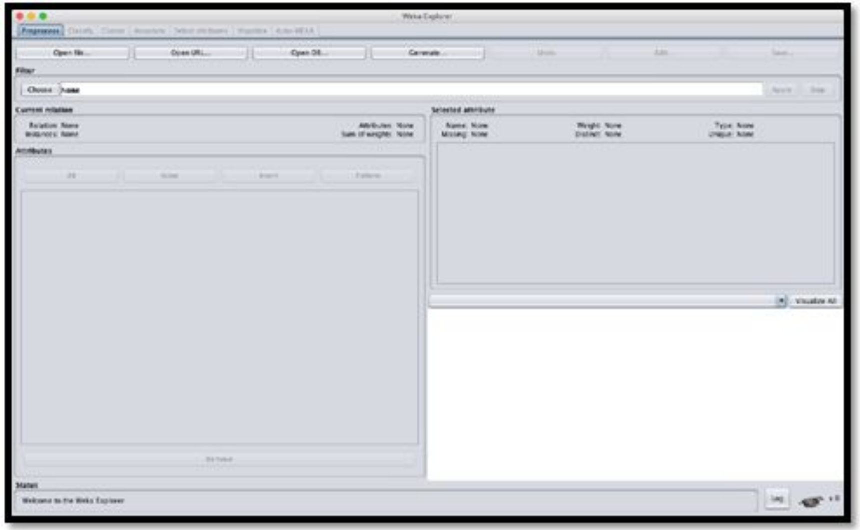
How to Use WEKA

The Weka is a bird native to New Zealand

Some WEKA Features

- Ideal starting point to explore data – figure out basic trends, try out different algorithms, etc.
- Use this after you have selected and an approach and you need to exhaustively check parameters and consistency of the method
- Helpful visual representation of your workflow – can be saved to enable reproducible pipelines
- Combined interface that has Explorer and Experimenter options together in one place
- Command line interface

WEKA Explorer Mode - Properties



- Both Attributes (X) and Classes (Y) can be of the following Types:
- **Numerical** (numbers)
- **Nominal** (1+ categories)
- **Date** (considered Nominal)
- **Binary** (2 categories *only*)

Lets all open up the Iris data set!

What to remember about ML Algorithms

- No free lunch: machine learning algorithms are tools, not dogmas
- Try simple algorithms first

- Better to have smart features and simple algorithms than simple features and smart algorithms

References

- https://machinelearningmastery.com/types-of-classification-in-machine-learning/
- Lecture notes by Dan Veltri
- Intel Academy Machine Learning Course
- https://developers.google.com/machine-learning/crash-course/
- https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages
- https://www.cs.waikato.ac.nz/ml/weka/
- https://www.seas.upenn.edu/~cis519/fall2017/lectures/01_intro

Thank you

If you have any questions feel free to email me:

sserurich@gmail.com