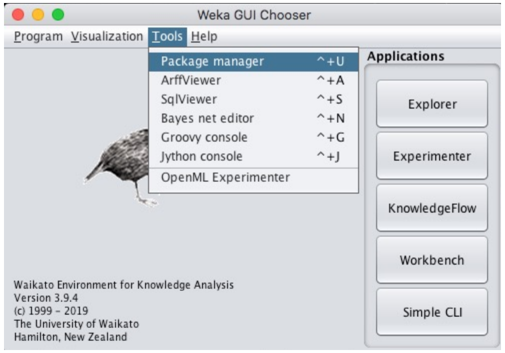
Installing Weka

Step1: Download and install the latest stable Weka release from:
https://waikato.github.io/weka-wiki/downloading_weka/

Step 2: Install packages via Package Manager

- Start Weka
- Go to Tools → Package Manager



Step 3: Select all available packages and click install (takes approx. 10 min)

A brief overview, comparison and practical applications of machine learning models

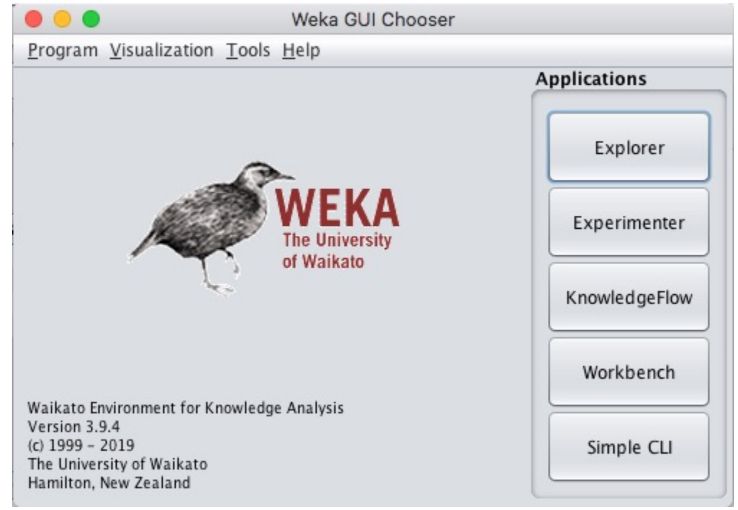
HANDS-ON WORKSHOP - LIVE

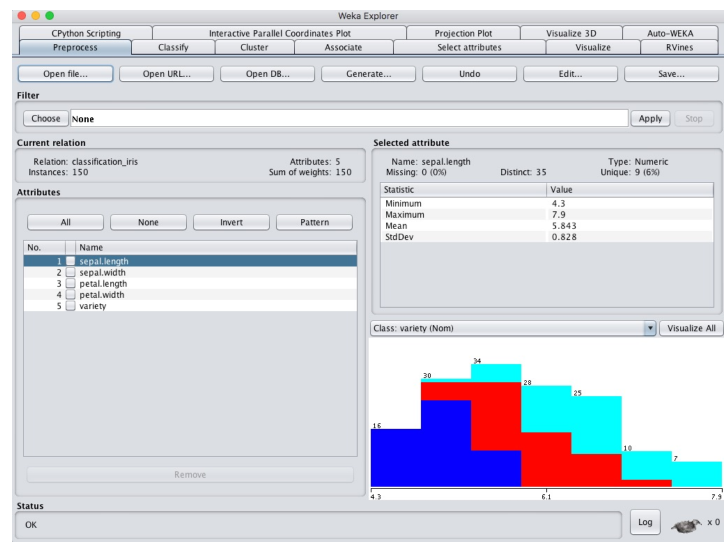
Dan Tulpan, Assistant Professor
Centre for Genetic Improvement of Livestock
Department of Animal Biosciences
Ontario Agricultural College
University of Guelph
dtulpan@uoguelph.ca

ASAS-NANP Pre-Conference Symposium
ASAS-CSAS-WSASAS Virtual Annual Meeting & Trade Show,
July 2021

Summary

- The Weka Explorer
- Classification problems
- Regression problems
- Data-related artifacts





Classification Problem 1

Data set: `classification_iris.csv`

- Number of data points: 150
- Number of classes: 3
- Number of attributes: 4
    - SL: Sepal length (cm)
    - SW: Sepal width (cm)
    - PL: Petal length (cm)
    - PW: Petal width (cm)

sepal.length	sepal.width	petal.length	petal.width	variety
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3	5.9	2.1	Virginica

Iris setosa

Iris versicolor

Iris virginica

Classification Problem 2

Data set: `classification_random_binary.csv`

- Number of data points: 100
- Number of classes: 2
- Number of attributes: 5
  - Cow_id
  - MS1
  - MS2
  - MS3
  - MS4

Cow_id	Ms1	Ms2	Ms3	Ms4	Pred_phen
1000	0.82670136	3.57660749	11.1282203	11.9935777	no
1004	0.75360992	7.04655697	10.3097353	9.39415003	no
1008	0.05777438	5.72303611	14.2557181	11.7663005	no
1012	0.23848243	5.35097367	13.0912955	12.5702272	yes
1016	0.16725548	4.16588693	9.76702334	12.1613706	yes
1020	0.69281354	6.45123742	12.519231	9.56385006	yes
1024	0.52219519	5.44130996	9.0774877	11.1876432	no
1028	0.636972	3.2613616	10.4232098	11.7134081	yes
1032	0.25784654	3.3175609	15.3697456	11.9128922	yes

Classification Problem 3

Data set: `classification_zoo_dataset.csv`

- Number of data points: 101
- Number of classes: 7
- Number of attributes: 17
    - animal_name
    - Hair
    - Feathers
    - …

animal_name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	class_type
aardvark	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	Mammal
antelope	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	Mammal
bass	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	Fish
bear	1	0	0	1	0	0	1	1	1	1	0	0	4	0	0	1	Mammal
boar	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	Mammal
buffalo	1	0	0	1	0	0	0	1	1	1	0	0	4	1	0	1	Mammal
calf	1	0	0	1	0	0	0	1	1	1	0	0	4	1	1	1	Mammal
carp	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	Fish
catfish	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	Fish
cavy	1	0	0	1	0	0	0	1	1	1	0	0	4	0	1	0	Mammal
cheetah	1	0	0	1	0	0	1	1	1	1	0	0	4	1	0	1	Mammal
chicken	0	1	1	0	1	0	0	0	1	1	0	0	2	1	1	0	Bird
chub	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	Fish
clam	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	Invertebrate

Regression Problem 1

Data set: `regression_linear_x_0_99_err0.csv`

- Number of data points: 100
- Number of attributes: 1
  - x

x	f(x)=2*x+3
0	3
1	5
2	7
3	9
4	11
5	13
6	15
7	17
8	19

Regression Problem 2

Data set: `regression_quadratic.csv`

- Number of data points: 100
- Number of attributes: 1
    - x

x	f(x)=2*x+3
0	3
1	5
2	7
3	9
4	11
5	13
6	15
7	17
8	19

Regression Problem 3

Data set: `regression_complex_func.csv`

- Number of data points: 100
- Number of attributes: 4
  - x
  - y
  - z
  - t

x	y	z	t	f(x,y,z,t) =5x-2cos(y)+3z^2/sqrt(t)
1	1	1	1	6.9194
2	2	2	2	27.8029
3	3	3	3	63.7454
4	4	4	4	117.3073
5	5	5	5	192.1378
6	6	6	6	292.6246
7	7	7	7	422.4176
8	8	8	8	583.349
9	9	9	9	775.8223
10	10	10	10	1000.3614
11	11	11	11	1258.9259

Regression Problem 4

Data set: `regression_random.csv`

- Number of data points: 100
- Number of attributes: 5
    - Cow_id
    - Ms1
    - Ms2
    - Ms3
    - Ms4

Cow_id	Ms1	Ms2	Ms3	Ms4	Pred_phen
1000	0.9470999	5.95797774	13.915448	12.9347224	0
1004	0.61073964	5.51292552	10.6797067	11.6490622	0
1008	0.79457971	6.14149345	11.5871536	10.3897927	0
1012	0.26283342	6.98005889	9.47012415	11.9177109	1
1016	0.73682092	5.88477566	9.54904708	12.1499646	1
1020	0.34022992	6.22943088	10.1080563	9.20260174	0
1024	0.31375375	7.23901781	11.2240887	11.5455358	0
1028	0.97597542	5.58866531	11.5971538	10.7777561	0
1032	0.70038486	6.92601426	10.50353	12.2210478	0
1036	0.98200496	5.25294086	13.4670354	9.02896927	1

Data-Related Artifacts

- Formatting errors
  - Special symbols
  - Extra columns
  - Duplicate header labels

- Hidden correlations
  - Unnecessary columns directly correlated with the predictor variable

Formatting Errors

data_error1_column_50_special_symbol.csv

47	1180	0.56906494	5.49645174	12.8512741	10.1804572	yes
48	1184	0.04040781	5.54765022	13.3681585	9.2797879	yes
49	1188	0.12406309	5.77318337	10.8367315	11.6485751	yes
50	1192	0.16223907	5.49''	11.3893505	10.9806007	no
51	1196	0.26181768	4.95858857	10.8131897	9.72609505	yes
52	1200	0.01718791	4.48834486	10.3239951	11.9754723	yes
53	1204	0.54080607	5.08725736	9.90381763	12.3617106	no

data_error2_extra_columns.csv

93	1364	0.9791347	4.5439163	9.817484	9.2959242	yes		
94	1368	0.7994233	7.4402879	10.065688	9.135617	yes		maybe good
95	1372	0.8082822	4.9122384	11.339509	9.072897	no		
96	1376	0.5848974	6.9507931	10.75717	11.55033	no		throw away
97	1380	0.6387242	5.8203253	9.4522329	12.018035	yes		
98	1384	0.5163807	5.8681784	10.936714	9.6650375	no		

Formatting Errors

header_error1_single_quote.csv

1	Cow_id	Ms1	Ms2'	Ms3	Ms4	Pred_phen
2	1000	0.54307371	7.09702619	10.2047803	12.1258179	no
3	1004	0.12472931	4.12297535	11.0922358	9.0506221	no
4	1008	0.61846058	5.73718235	11.7349934	9.97657243	no
5	1012	0.6453044	5.0846034	9.76661321	9.6397819	yes
6	1016	0.42973124	6.48659263	11.9666711	10.1245367	yes

header_error2_duplicate_column_labels.csv

1	Cow_id	Ms1	Ms4	Ms3	Ms4	Pred_phen
2	1000	0.26211615	6.99414146	12.0456984	9.51521005	no
3	1004	0.27628907	4.80414009	14.1992321	10.8114346	no
4	1008	0.39226337	4.88113945	14.5932792	9.947972	no
5	1012	0.57436216	4.63288931	13.1472392	12.1626214	yes
6	1016	0.97438431	6.60621846	10.9354242	11.5387297	yes
7	1020	0.33248328	3.68439992	11.3911645	10.2929808	yes

Formatting Errors

`header_error3_double_quotes.csv`

1	"Cow_id"	"Ms1"	"Ms2"	"Ms3"	"Ms4"	"Pred_phen"
2	1000	0.9082203	4.6205842	12.304296	10.26865	no
3	1004	0.4561411	4.1029248	10.416078	9.7241627	no
4	1008	0.6286037	5.283989	13.20705	9.7251261	no
5	1012	0.7591403	3.8496747	12.942408	9.5352086	yes
6	1016	0.7561974	4.2597873	12.683271	12.354336	yes
7	1020	0.7930203	7.0657708	10.398488	9.118181	yes
8	1024	0.7171487	6.9522685	10.689412	12.498773	no
9	1028	0.5653197	6.9458774	9.8353356	12.712097	yes

Correlated features

- Lead to over-inflated predictions

overinflated_acc_ded_column_E_is_dependent_on_A.csv

animal_id	gen1	gen2	gen3	pred_gen
12	2.291	2.294	103.375	0
12	2.688	1.425	81.123	0
12	2.71	0.395	68.144	0
12	2.02	2.174	102.877	0
17	2.001	2.684	70.788	7
17	2.941	0.047	82.091	7
17	2.289	1.525	58.188	7
17	2.346	2.439	76.152	7
22	2.199	2.478	97.014	14
22	2.583	2.188	74.66	14
22	2.974	2.178	104.732	14
22	2.41	2.801	73.528	14
27	2.952	0.806	110.689	21
27	2.43	0.411	102.791	21
27	2.382	2.03	77.434	21
27	2.447	0.294	62.878	21
32	2.011	2.977	94.383	28

Last Year’s Presentation

- Theoretical aspects of Machine Learning (ML)

A brief overview, comparison and practical applications of machine learning models

Dan Tulpan, Assistant Professor
Centre for Genetic Improvement of Livestock
Department of Animal Biosciences
Ontario Agricultural College
University of Guelph
dtulpan@uoguelph.ca

UNIVERSITY of GUELPH
IMPROVE LIFE.

ASAS-NANP Pre-Conference Symposium
ASAS-CSAS-WSASAS Virtual Annual Meeting & Trade Show,
July 19-23, 2020

ONTARIO AGRICULTURAL COLLEGE
DEPARTMENT OF ANIMAL BIOSCIENCES

Summary

- Machine Learning – general notions

- Types of problems
  - Classification
    - Decision trees
    - Artificial neural networks

  - Regression

  - Clustering
    - K-Nearest Neighbour

  - Dimensionality reduction

- Developing ML models (practical considerations)

- Follow-up Hands-on/Demo Workshop

Learning

“Learning is any process by which a system improves performance from experience.”

[Herbert Simon (1916-2001), American economist, political scientist and cognitive psychologist]

- Types of learning
  - **Supervised** (inductive) learning
    - Training data includes desired outputs
    - E.g.: classification problems
  - **Unsupervised** learning
    - Training data does not include desired outputs
    - E.g.: clustering problems
  - Semi-supervised learning (hybrid)
    - Training data includes a few desired outputs
  - Reinforcement learning
    - Rewards from sequence of actions
    - E.g.: intelligent robots

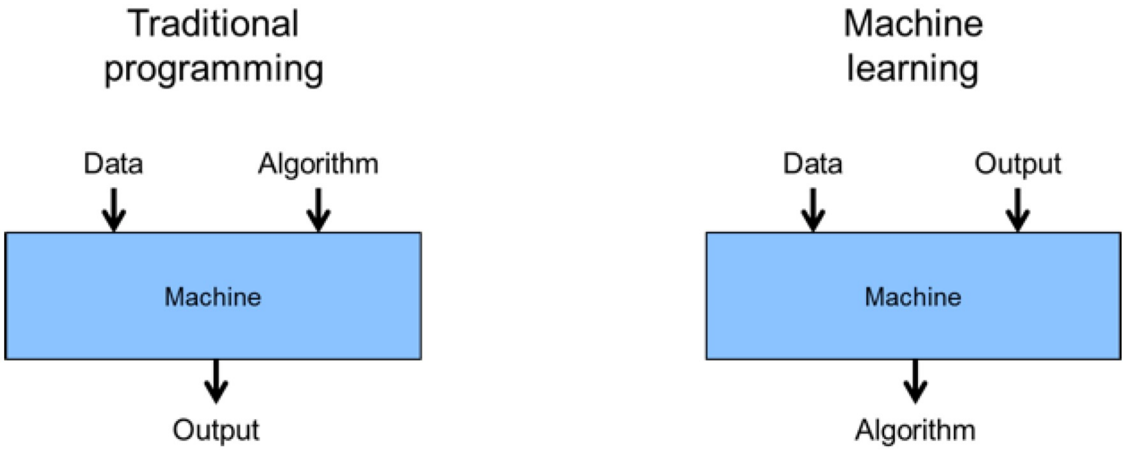
Machine learning https://me.me/

Machine learning (ML)

- The field of machine learning in concerned with the question of how to construct computer programs that automatically improve with experience.

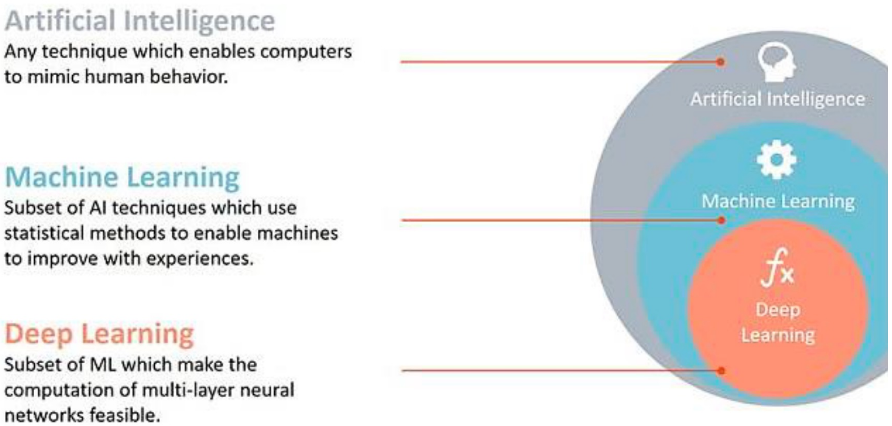
Tom Mitchell, Machine Learning (1997)

- Get computers to program themselves

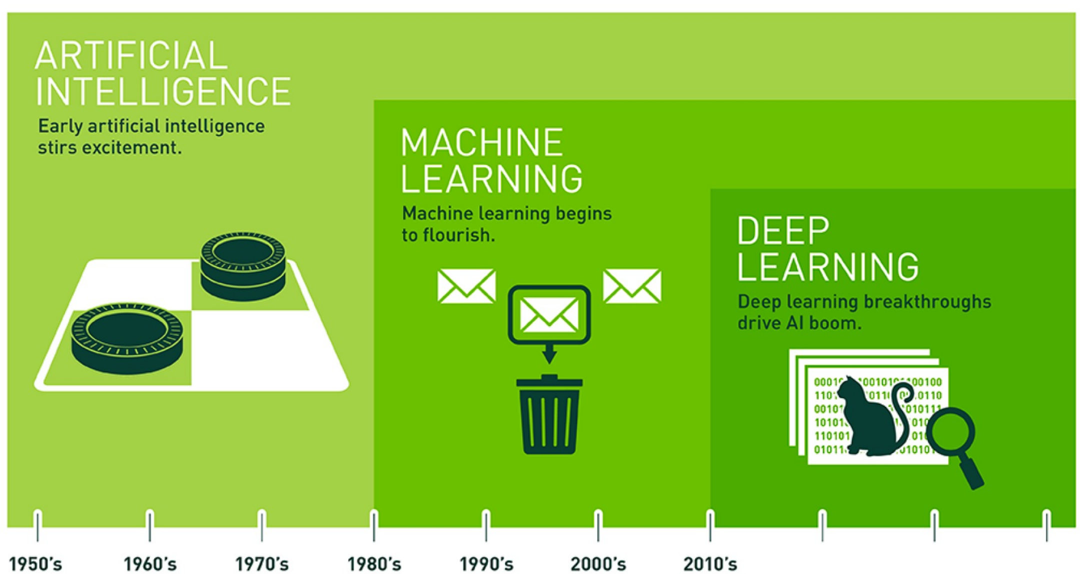


https://dzone.com/articles/what-everyone-should-know-about-machine-learning

Context

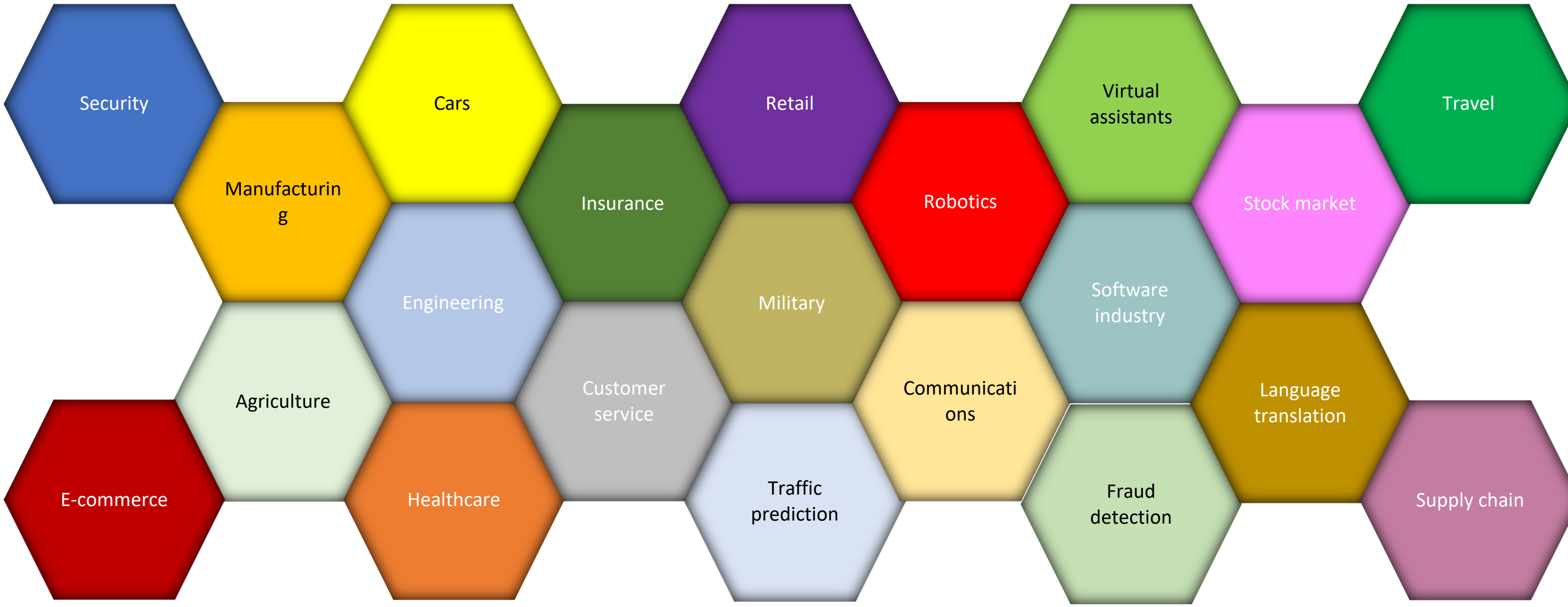


https://www.kdnuggets.com/2017/07/rapidminer-ai-machine-learning-deep-learning.html



https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Applications of ML



How does it work?

- **Step 1:** The user provides the learning system with **examples** of the concept to be learned or plain **data**.

- **Step 2:** The learning system algorithm **infers/builds** a characteristic **model** from these examples.

- **Step 3:** The **model is used to predict** quickly and with high accuracy whether or not future **novel instances** follow the model.

When to use machine learning?

- When there are patterns in the data
- When we can not figure out the functional relationships mathematically
- When we have a lot of (unlabeled) data
  - Labeled training sets are harder to find or generate
  - Data is in high-dimension
    - High dimension "features"
    - Example: sensor data
  - Want to "discover" lower-dimension representations
    - Dimensionality reduction

Inspired from: https://datatracker.ietf.org/meeting/92/materials/slides-92-sdnrg-0

The Ultimate Goal of ML

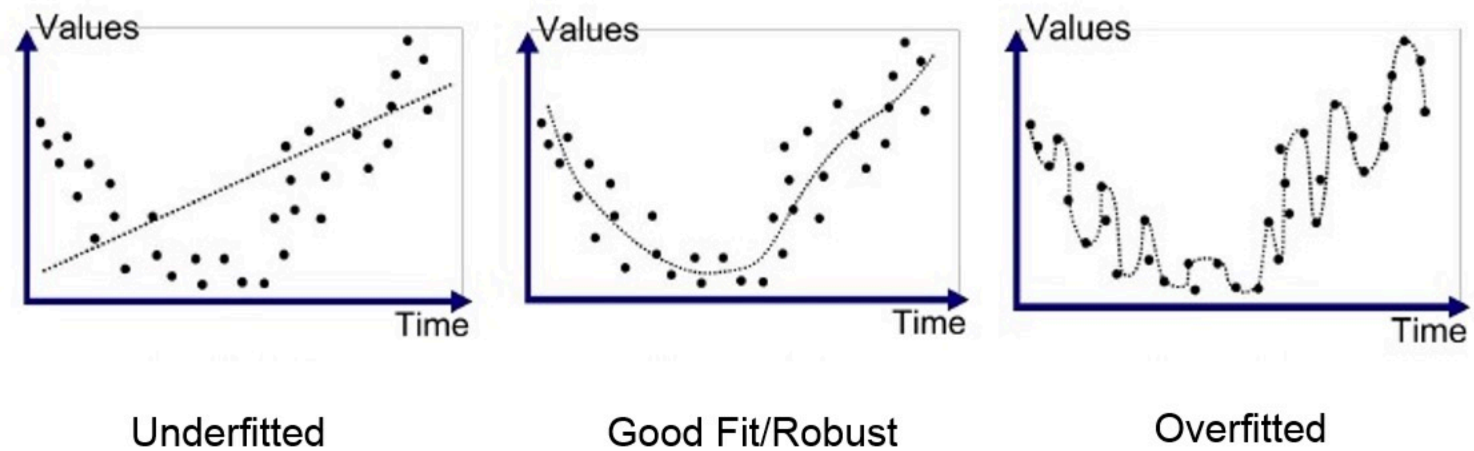
- **Generalization:** the ability of a trained model to fit unseen instances

Training set (labels known)

Test set (labels unknown)

cs.brown.edu/courses/cs143/2013/lectures/17.ppt

Generalization

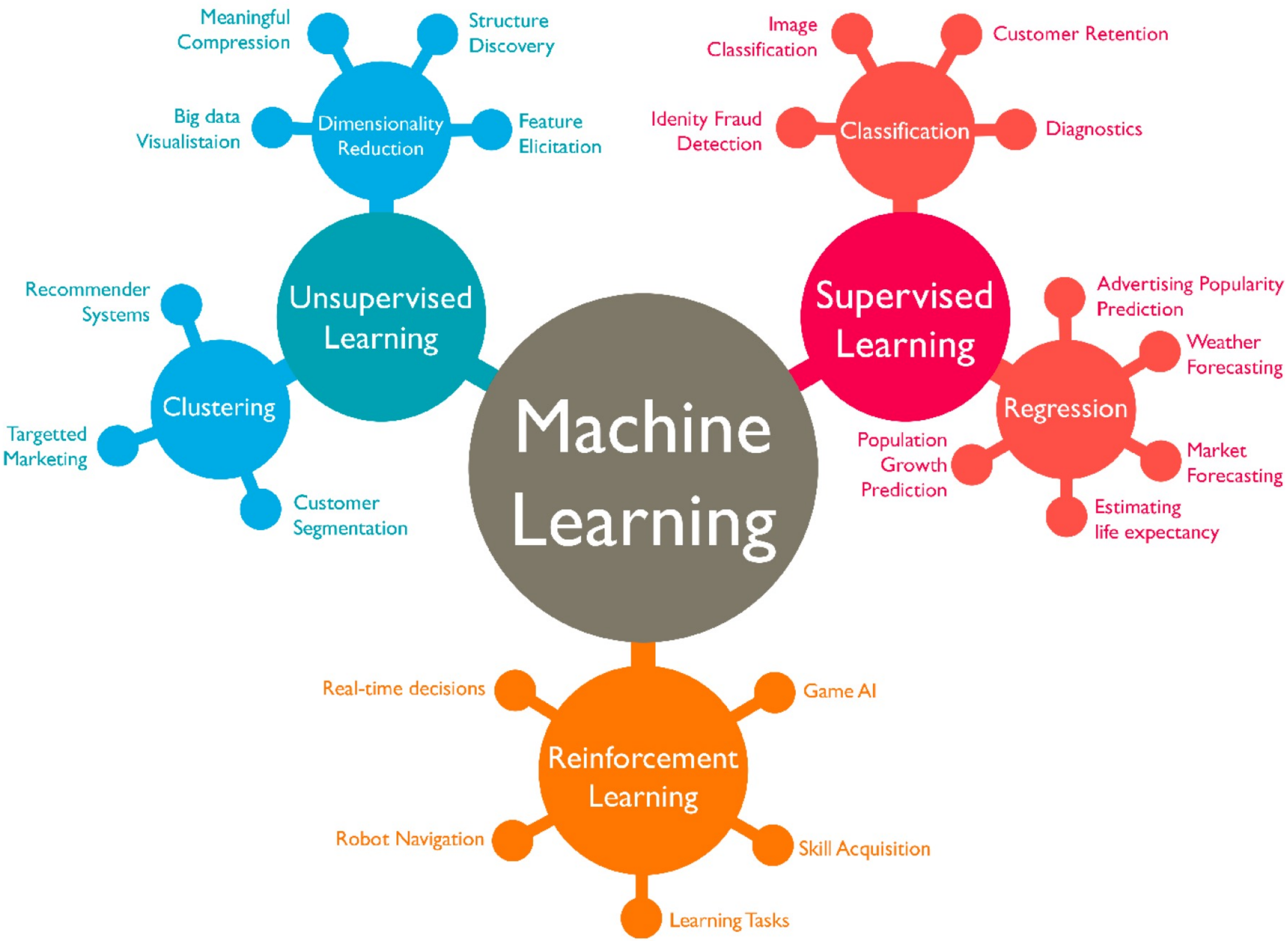


https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

- **Underfitting:** model is too "simple" to represent all the relevant class characteristics
  - High bias and low variance
  - High training error and high test error

- **Overfitting:** model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Types of problems solved by ML



https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Types of problems solved by ML

LEARNING TYPES

DATA TYPES		Supervised Learning	Unsupervised Learning
	Discrete	classification or categorization	clustering
	Continuous	regression	dimensionality reduction

Classification

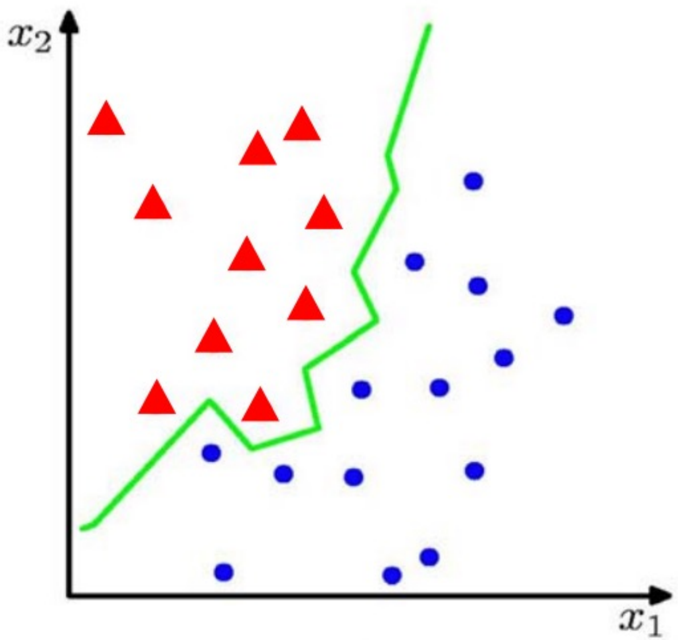
LEARNING TYPES

DATA TYPES	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Classification

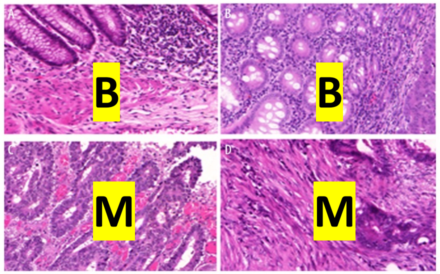
- Given a set of observations:
  - *($input_i$, $output_i$)* pairs, where $output_i \in \{c_1, c_2, \ldots\}$

- Find a function *f*, such that: $f(input_i) = output_i$



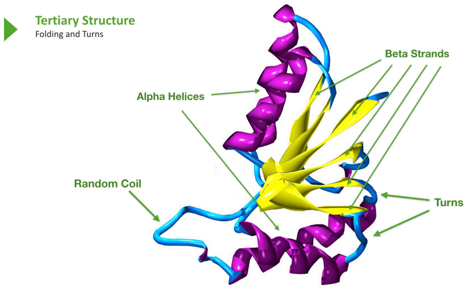
Classification Problem Examples

- Classify tumors as malign or benign



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6154116/

- Classify protein secondary structures as α-helices, β-sheets, coils or turns



http://oregonstate.edu/instruct/bb450/450material/OutlineMaterials/4_5Proteins.html

- Classify mushrooms as edible or poisonous

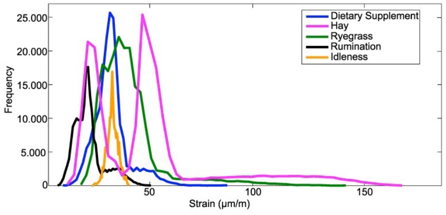
“Destroying Angel” Mushrooms

Edible Puffball Mushrooms

https://www.ck12.org/book/Biology-%252528CA-DTI3%2525292/r3/section/14.5/

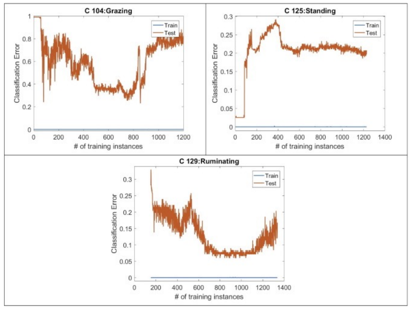
Classification Problem Examples

- Classify ruminants chewing patterns



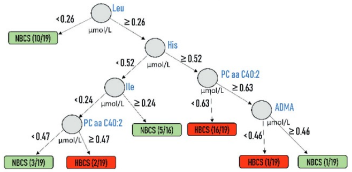
[Pegorini et al., 2015: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4701289/](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4701289/)

- Classify cattle behaviour based on ear tag, collar and halter sensors



Rahman et al., 2018:
https://www.sciencedirect.com/science/article/pii/S2214317317301099

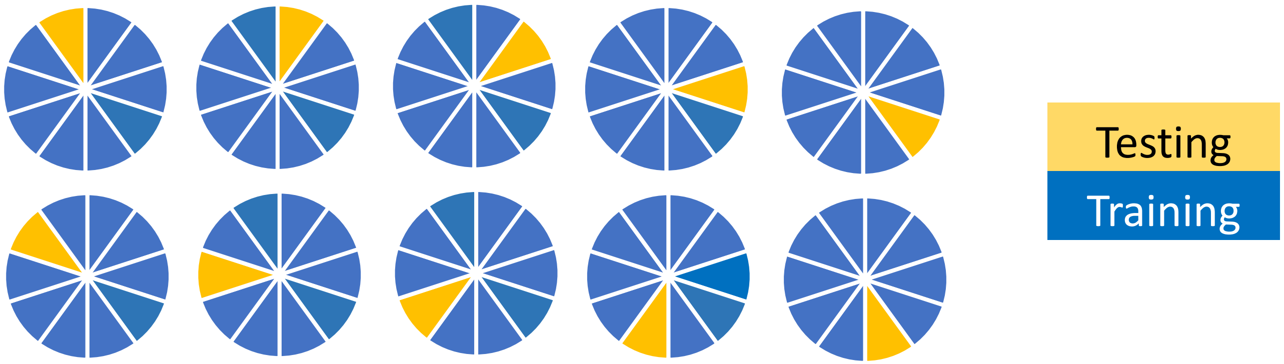
- Classify cattle BCS based on metabolite profiles



[Ghaffari et al., 2019: https://www.journalofdairyscience.org/article/S0022-0302(19)30838-0/abstract#](https://www.journalofdairyscience.org/article/S0022-0302(19)30838-0/abstract#)

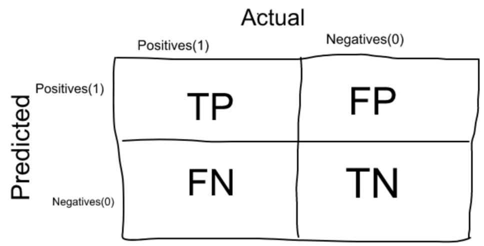
Performance evaluation protocol

- Split the data into: training, testing (and validation)
  - Fixed split approach
    - E.g. 70% training, 30% testing
  - Cross-validation approach (k-fold)

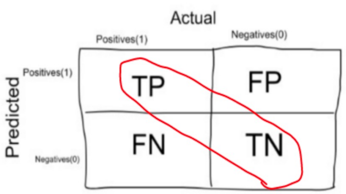


- Choose evaluation measures
- Perform measurements over n runs (n>= 1)

Performance evaluation measures

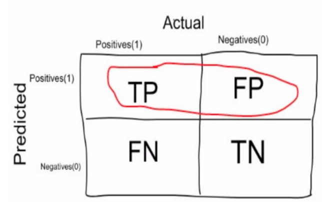


Confusion Matrix



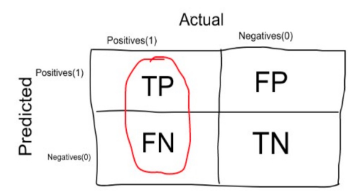
$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Use when data is balanced.



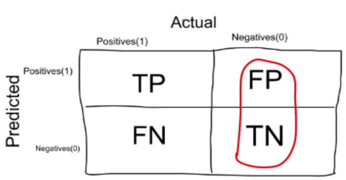
$$\text{Precision} = \frac{TP}{TP + FP}$$

Use to minimize FP.



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

Use to minimize FN.



$$\text{Specificity} = \frac{TN}{TN + FP}$$

The opposite of Recall

$$F1_{score} = \frac{2 \times precission \times recall}{precision + recall}$$

Great if it equals 1 and not good if it is 0.

+ 100 or more:

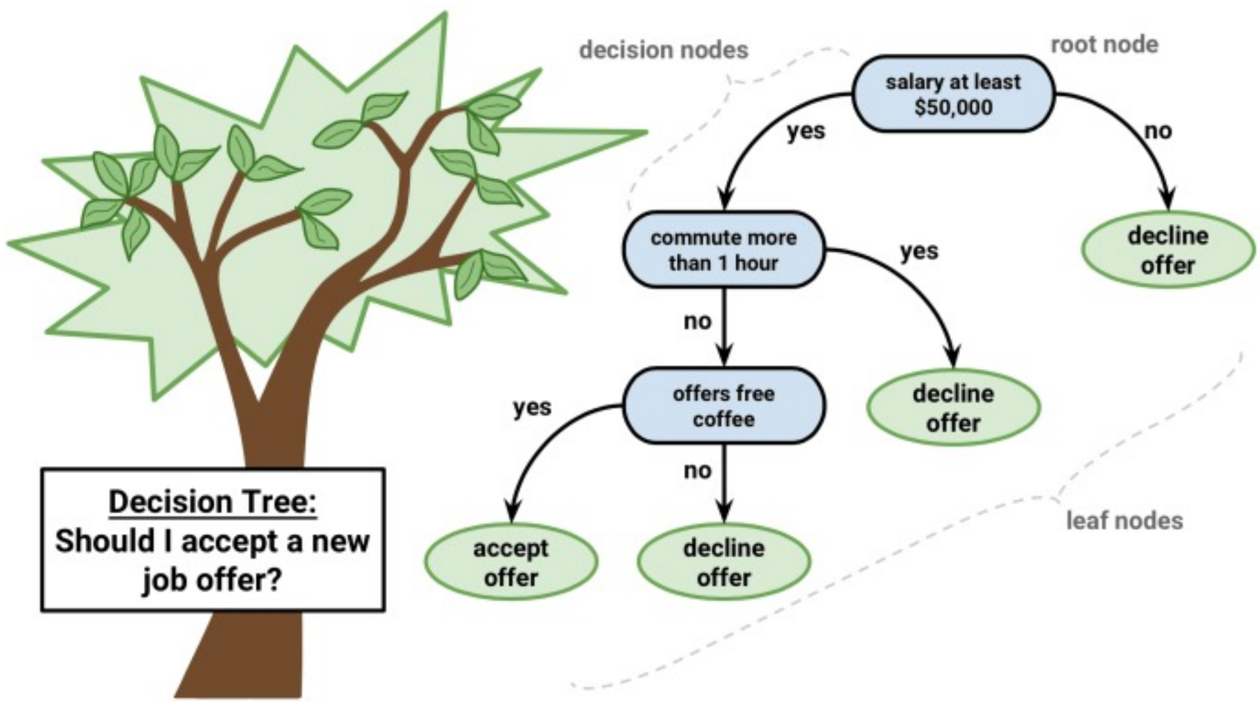
- **TPR, TDR**
- **FPR, FDR**
- **PPV, NPV**
- **MCC**
- **...**

Classification methods

- **Tree-based:**
  - **Random Forest** [Breiman 2001],
  - **J48** [Quinlan 1993]
- **Bayesian:**
  - **Naïve Bayes** [Clark & Niblett, 1989]
- **Boosting:**
  - **AdaBoost** [Freund & Schapire, 1996]
- **Kernel-based:**
  - **SVM** [Ben-Hur et al., 2001]
- **Rule-based:**
  - **Decision Table** [Kohavi 1995]
- **Artificial neural networks**
  - **Multi-layer Perceptron** [Rosenblatt 1961]
  - **RNN** [Rumelhart 1986],
- **Deep learning:**
  - **CNN** [Fukushima 1980;LeCun 1998]
- Etc.

Decision Trees

- A flow chart-like topology (a tree)
- Each internal node represents a test on an attribute
- Each branch represents an outcome of a test
- Leaf nodes represent class labels (if used for classification)

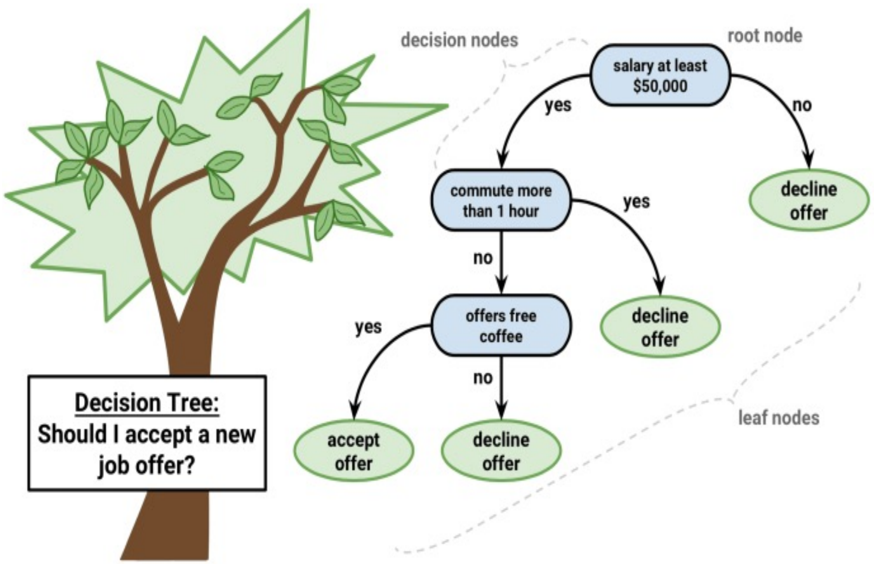


Decision Tree classifier, **Image credit:** http://www.packtpub.com

Decision Trees vs. Linear Models

- **Choose linear models if** the relationship between dependent & independent variables is well approximated by a linear model.
- **Choose a decision tree model if** there is a high non-linearity & complex relationship between dependent & independent variables.
- **Choose a decision tree model if** you need to build a model which is easy to explain to people. Decision tree models are even simpler to interpret than linear regression!

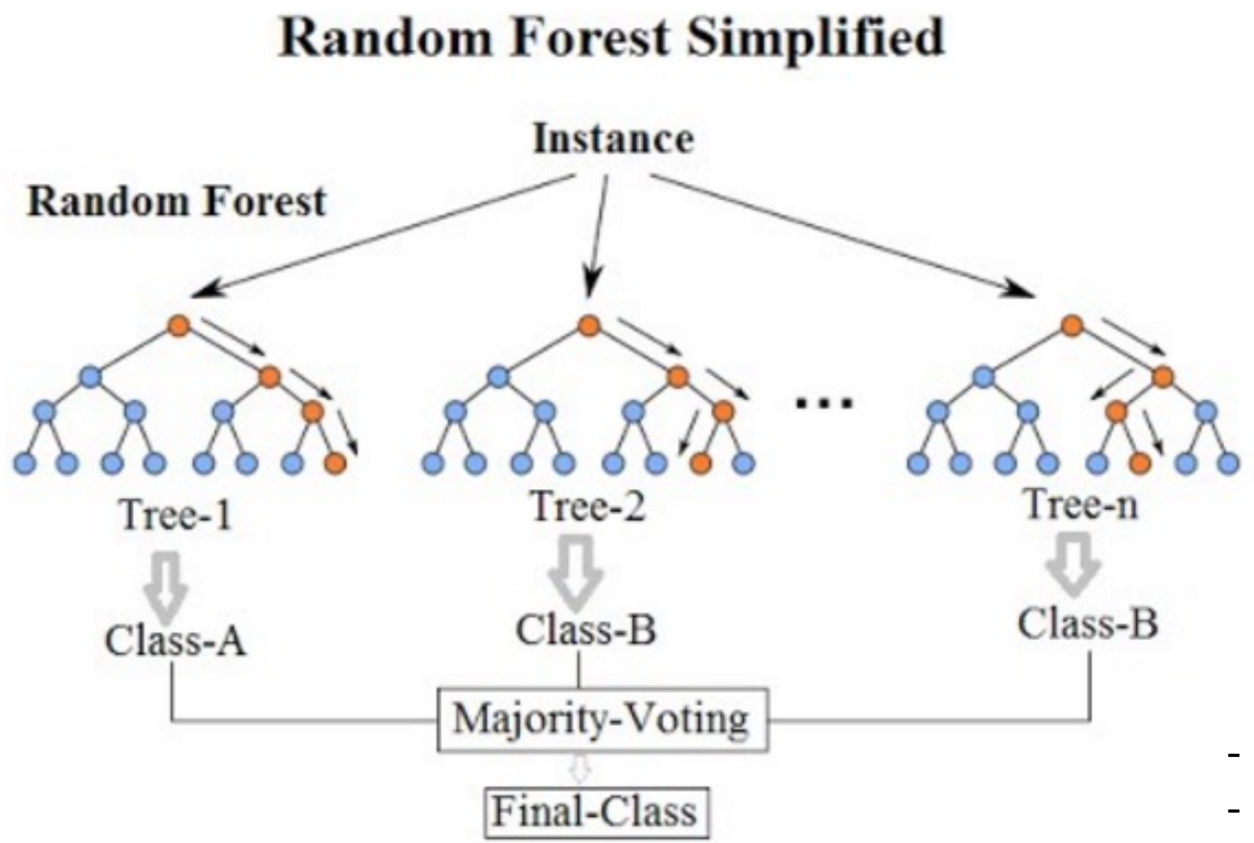
Decision Trees vs. Linear Models



Decision Tree classifier, **Image credit:** http://www.packtpub.com

$$\textbf{F(s,d,c)}= \begin{cases} 0 & s < 50000. \\ 0 & s \geq 50000 \; and \; d > 1 \\ 0 & s \geq 50000 \; and \; d < 1 \; and \; c = 0 \\ 1 & otherwise \end{cases}$$

Random Forests



https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png

Bagging / bootstrap aggregation
Random sub-samples of the data

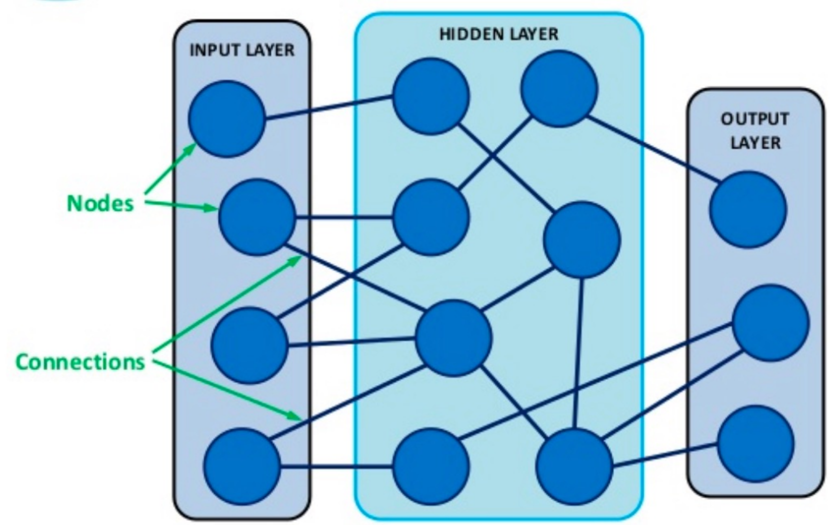
Option: **feature bagging**

↓

- Decorrelate the data
- Reduce the impact of strong predictor variables

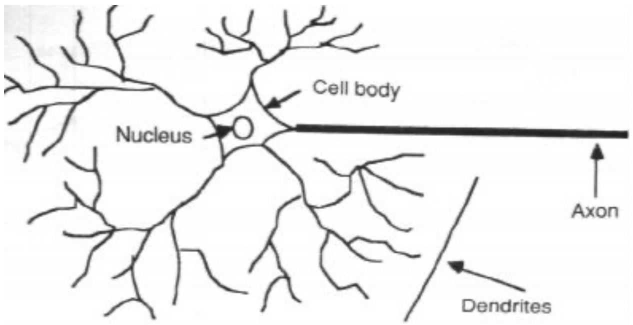
Artificial Neural Networks (ANN)

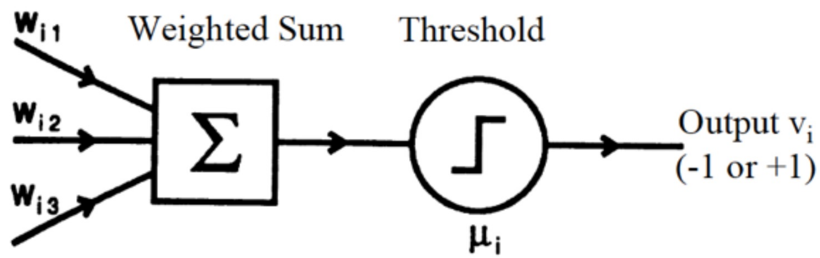
- An ANN is a biologically inspired computational model.
- ANNs attempt to mimic the functionality of the human brain.
- An ANN contains:
    - Processing elements (neurons)
    - Connections (between neurons)
    - Training & recall algorithms
- Important feature: network layout



https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

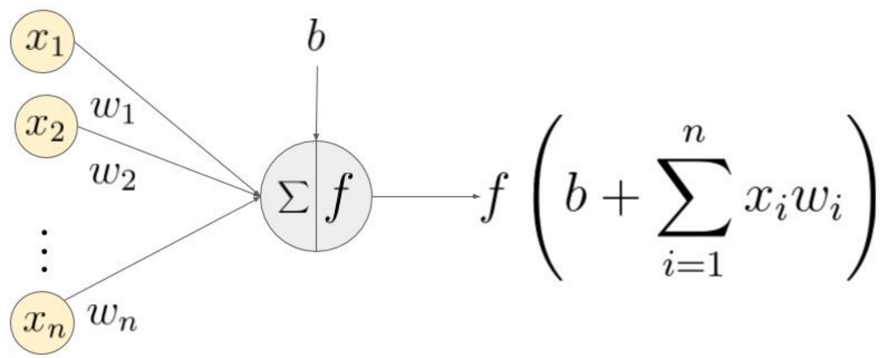
Artificial Neuron





- First model (**the perceptron**) was developed by Rosenblatt in 1957.
- **Idea**:

input data

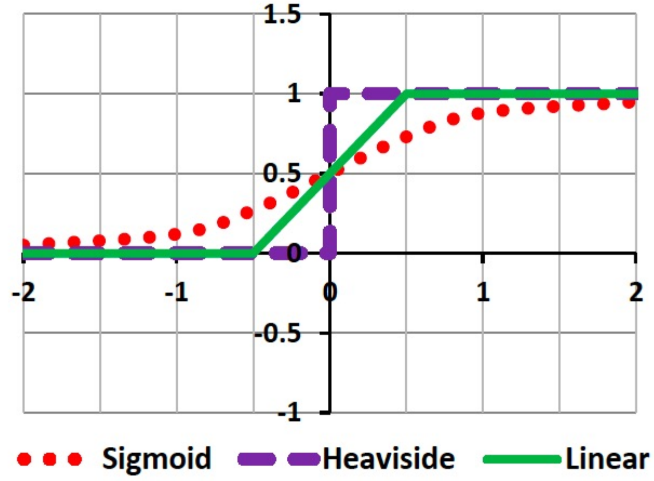


An example of a neuron showing the input (x_1 - x_n), their corresponding weights (w_1 - w_n), a bias (b) and the activation function f applied to the weighted sum of the inputs.

https://www.learnopencv.com/understanding-feedforward-neural-networks/

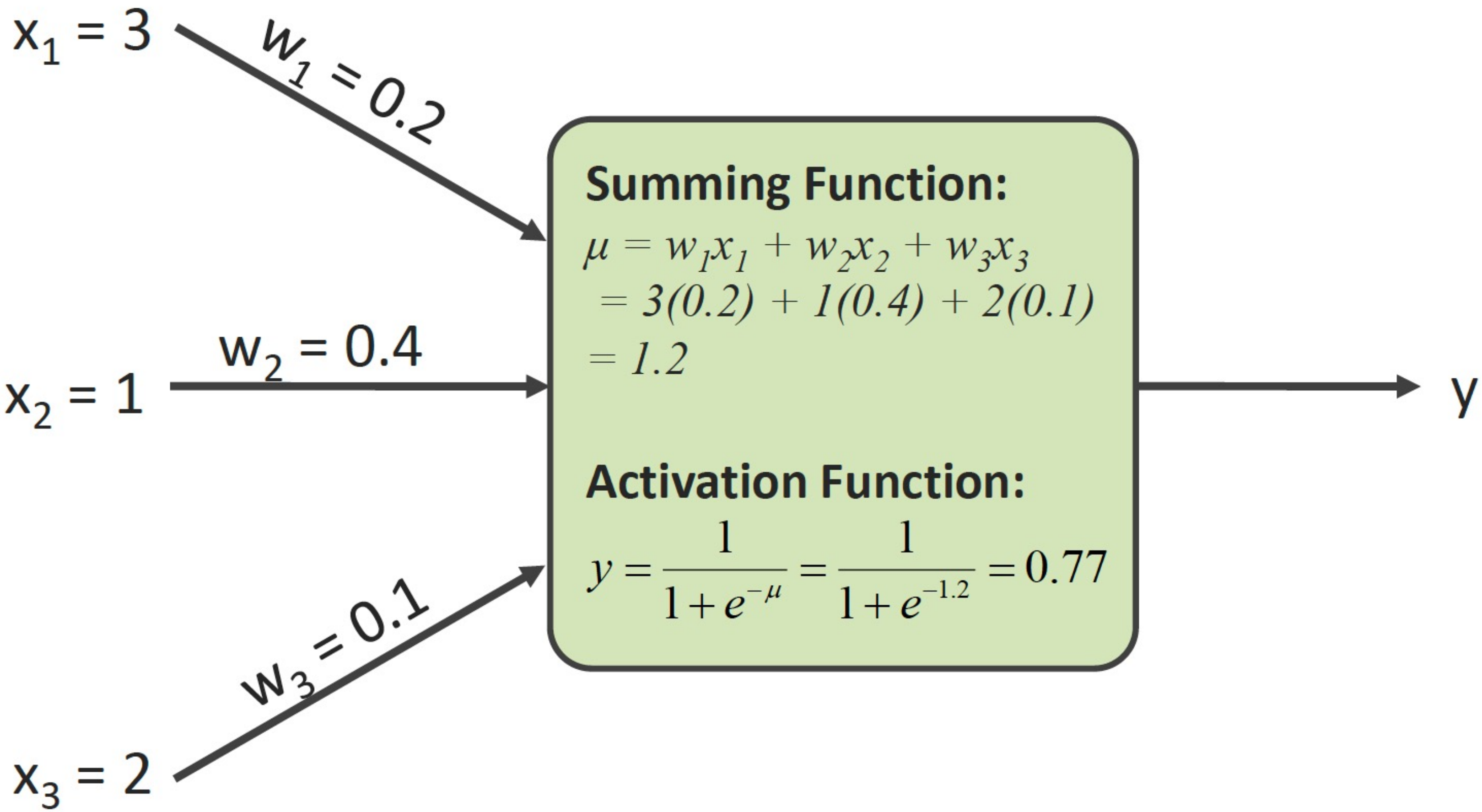
The Perceptron

- **Input signals**:
  - Continuous or discrete values fed from previous neurons
  - Each input associated with a Weight
- **Integration Function**:
  - Usually a weighted summation function
  - Threshold/Bias regulates result of Integration Function
  - Output is called *neuron net input*
- **Activation/Transfer Function**:
  - Usually a non linear function
  - Output interval [0,1] or [-1,1]
  - Output values continuous or discrete



https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

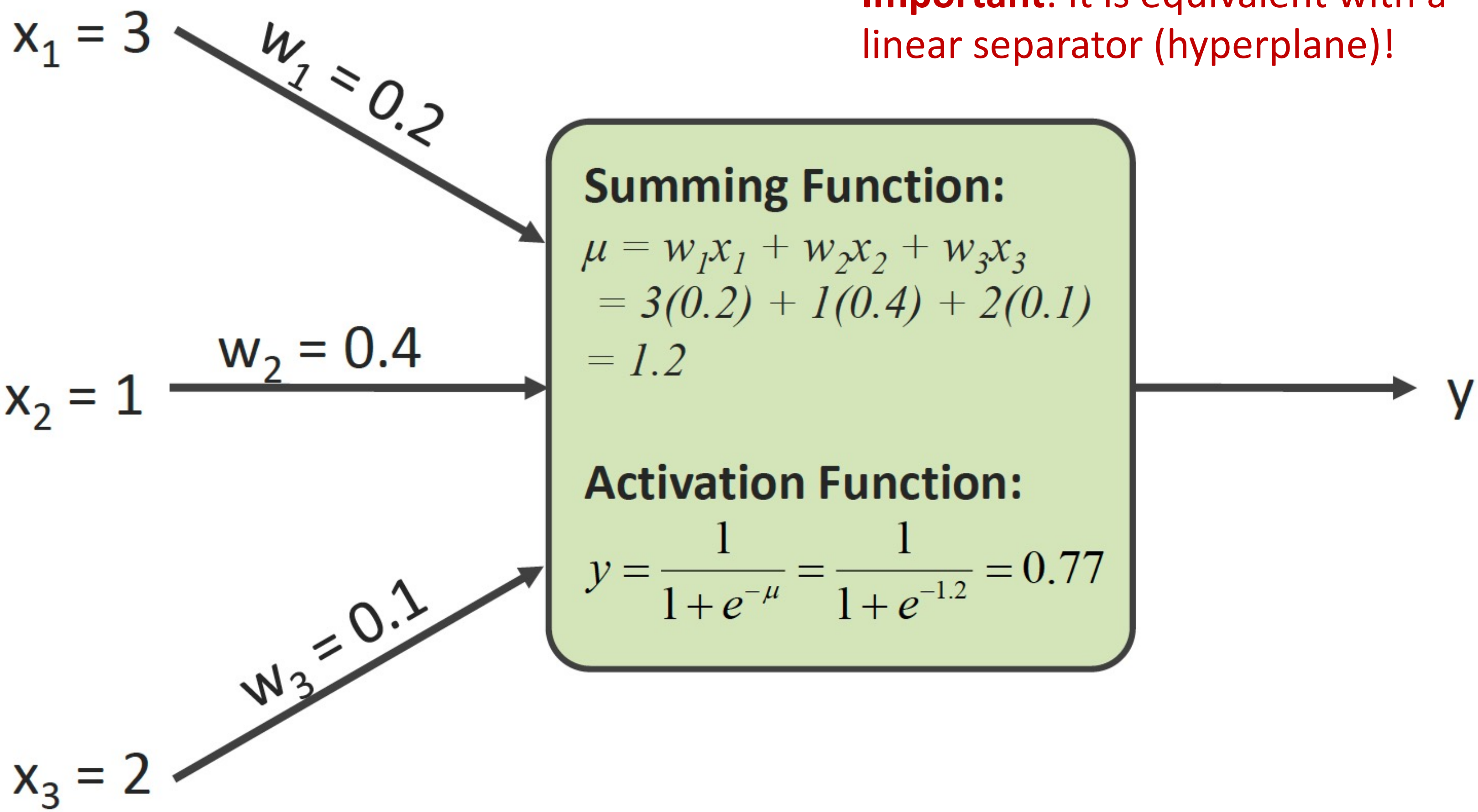
Perceptron Example



https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

Perceptron Example

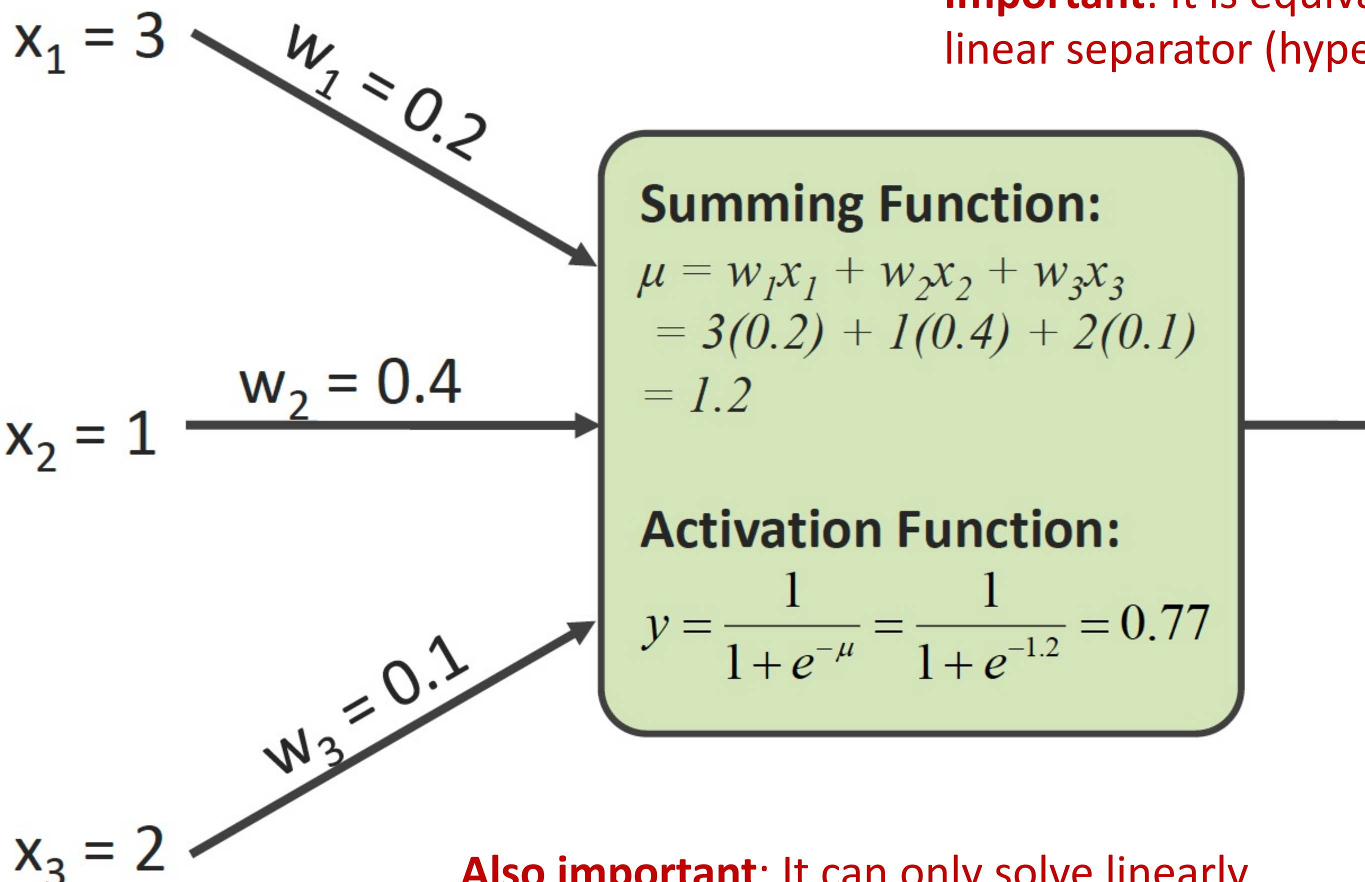
Important: It is equivalent with a linear separator (hyperplane)!

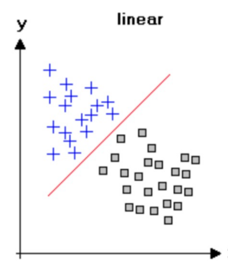


https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

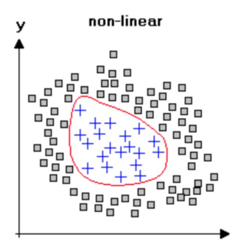
Perceptron Example

Important: It is equivalent with a linear separator (hyperplane)!





OK



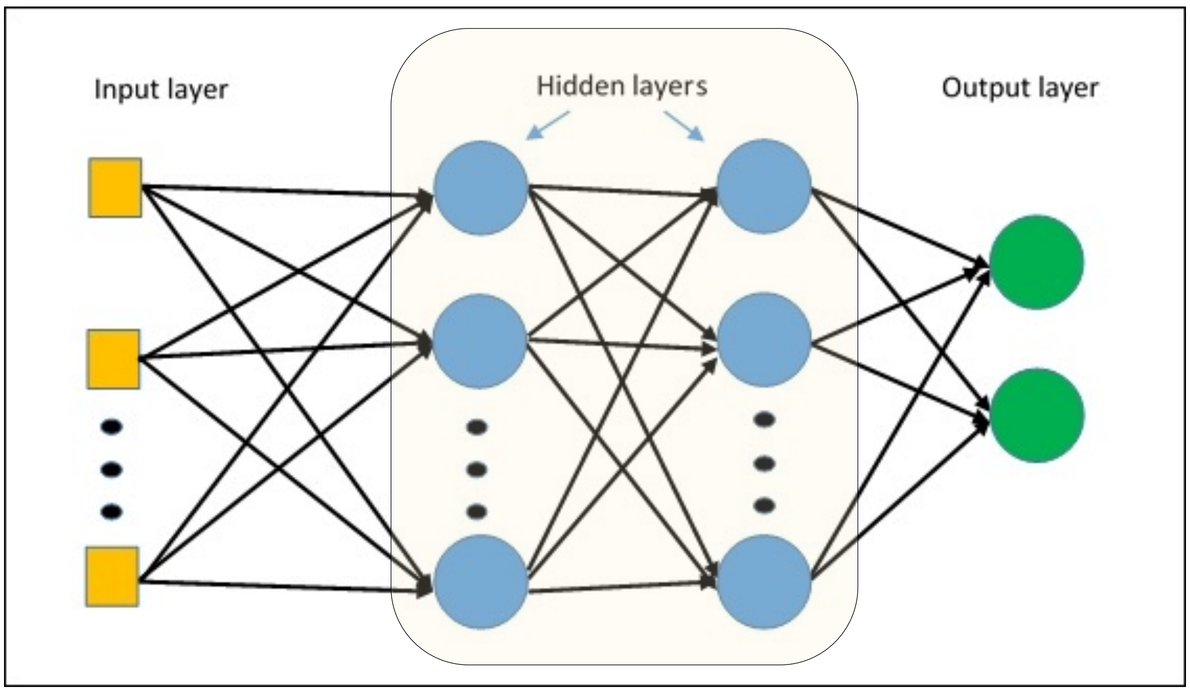
NOT OK

Also important: It can only solve linearly separable problems.

https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

Multilayer Perceptron (MLP)

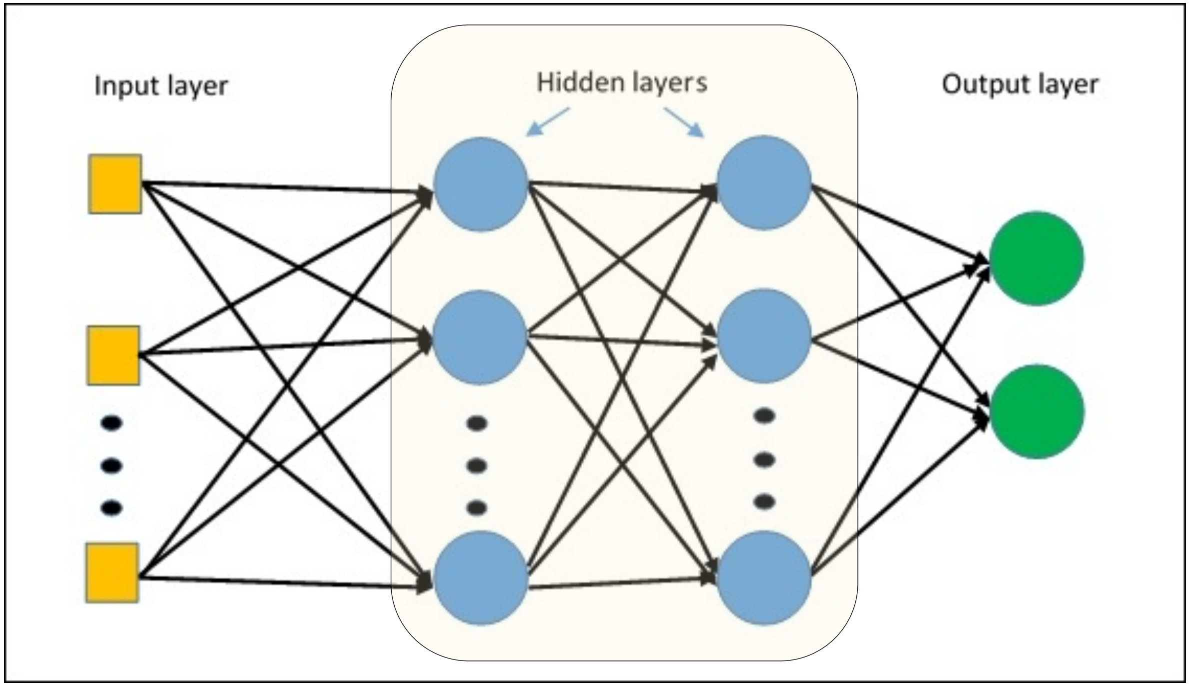
- Solution: 1980's – Multilayer perceptrons can solve non-linear separable problems

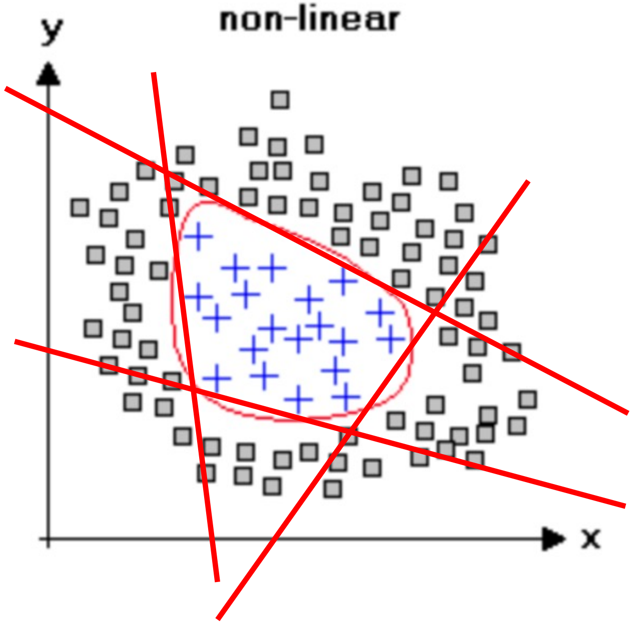


https://www.oreilly.com/library/view/getting-started-with/9781786468574/ch04s04.html

Multilayer Perceptron (MLP)

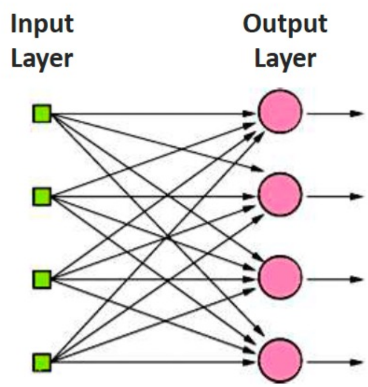
- Solution: 1980's – Multilayer perceptrons can solve non-linear separable problems



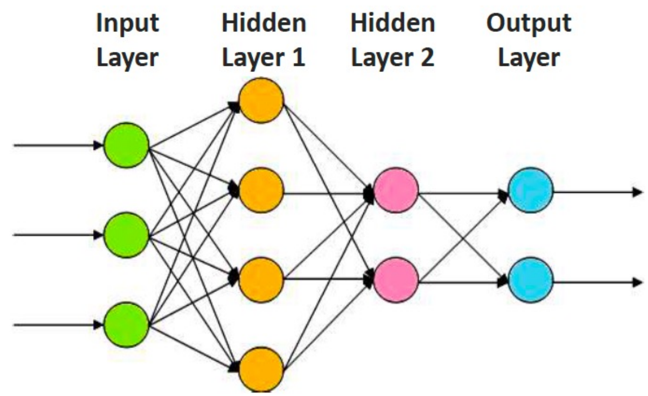


https://www.oreilly.com/library/view/getting-started-with/9781786468574/ch04s04.html

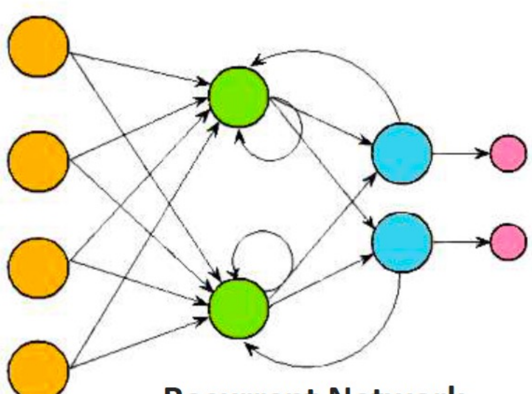
Examples of ANNs Topologies



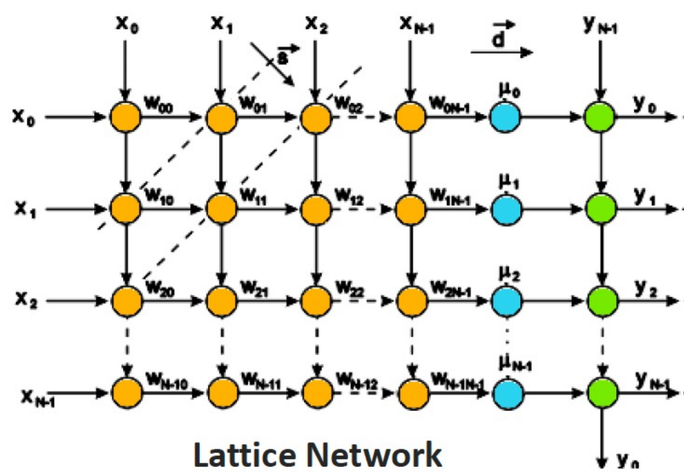
Single Layer Feedforward Network



Multi- Layer Feedforward Network



Recurrent Network



Lattice Network

https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

How do ANNs “Learn”?

- **Initialize the weights** (w_0, w_1, …, w_k)
  - Typically with random values

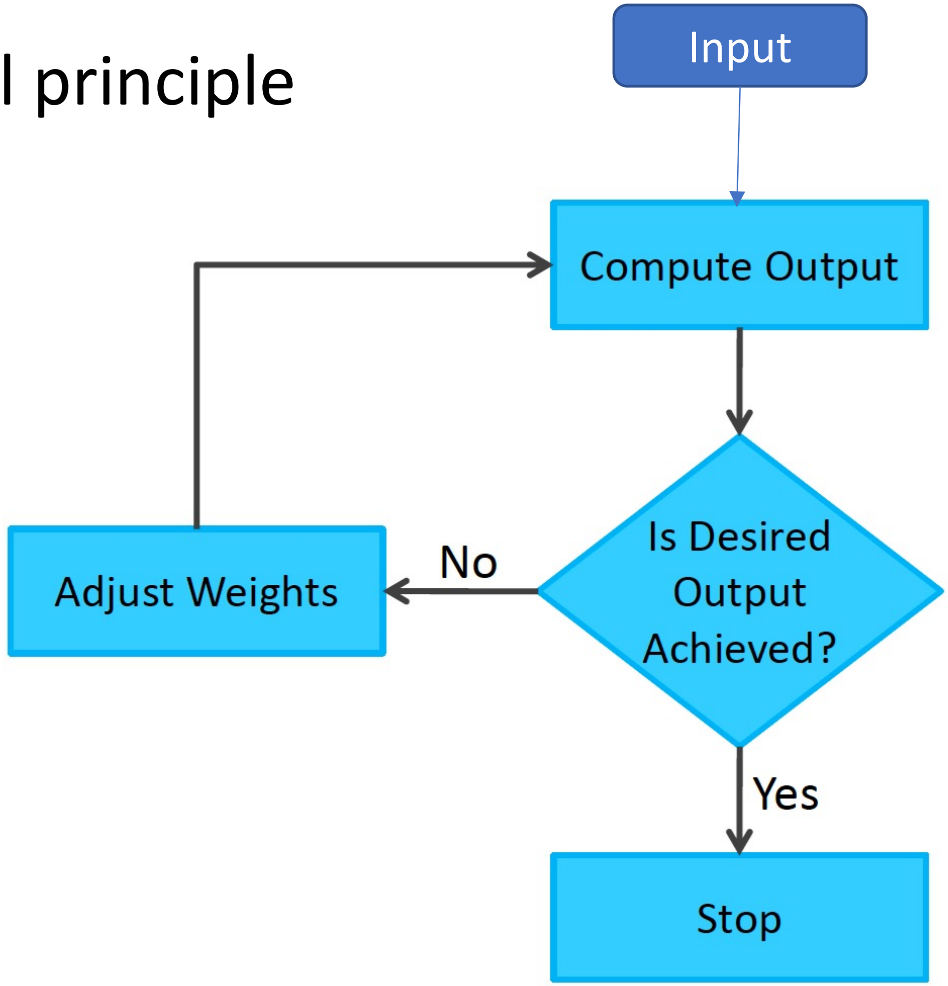
- **Adjust the weights** in such a way that the output of ANN is consistent with class labels of training examples
  - Error function:

$$E = \sum_i \left[Y_i - f(w_i, X_i) \right]^2$$

  - **Find the weights** w_i’s that **minimize** the above **error function** and adjust them proportionally with the error
    - e.g., gradient descent, backpropagation algorithm

ANN Supervised Learning

• General principle



Small error

https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

What Are ANNs Used For?

- **Classification**: Assigning each object to a known specific class
- **Clustering**: Grouping together objects similar to each other
- **Pattern Association**: Presenting of an input sample triggers the generation of specific output pattern
- **Function approximation**: Constructing a function generating almost the same outputs from input data as the modeled process
- **Optimization**: Optimizing function values subject to constraints
- **Forecasting**: Predicting future events on the basis of past history
- **Control**: Determining values for input variables to achieve desired values for output variables

https://www.slideshare.net/purneshaloni5/14-mohsin-dalvi-artificial-neural-networks-presentation-46777890

When to Use ANN?

- Input is high-dimensional discrete or raw-valued
- Output is discrete or real-valued
- Output is a vector of values
- Possibly noisy data
- The form of the target function is unknown
- Human readability of the result is not important

Regression

LEARNING TYPES

DATA TYPES

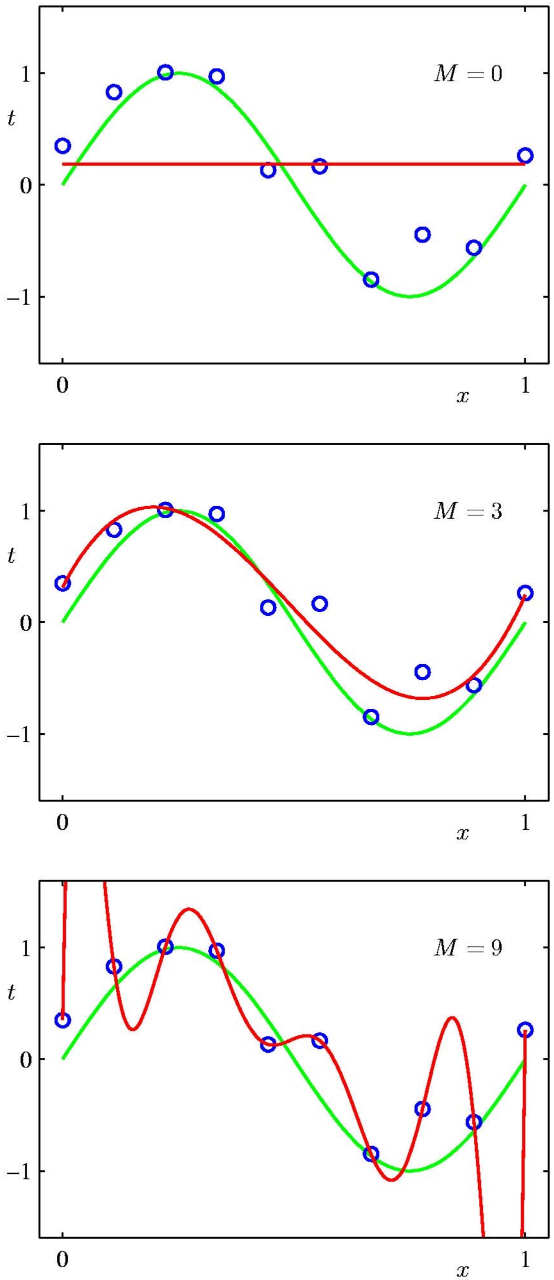
	Supervised Learning	***Unsupervised Learning***
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Regression

- **Regression** is a technique that is used to predict values of a desired target quantity when the target quantity is **continuous**.
  - **Note**: In classification, the target quantity is discrete.

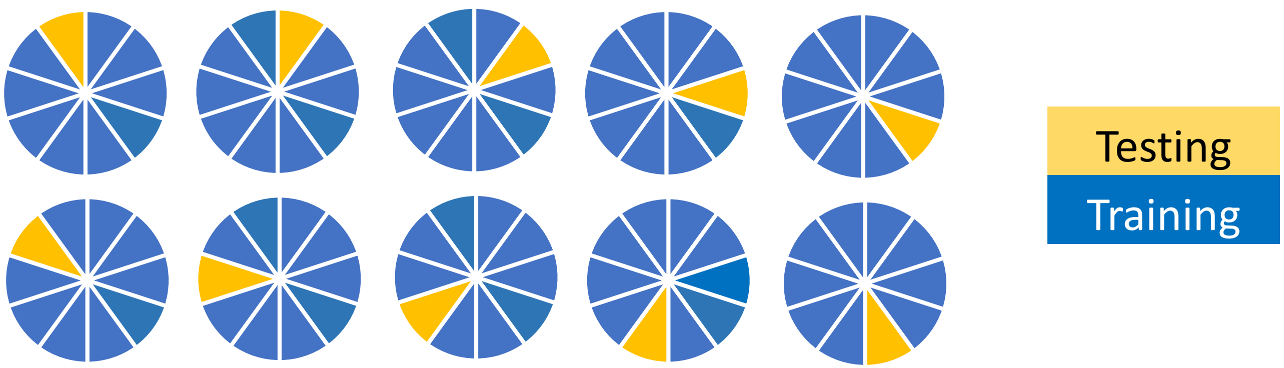
- **Multiple methods**: linear, higher-order (quadratic, polynomial), least-squares, Bayesian, non-linear, logistic, ANN, DT, Generalized Linear Models (GLMs), …

- **Note**: Most methods for classification work for regression, too, with some modifications.



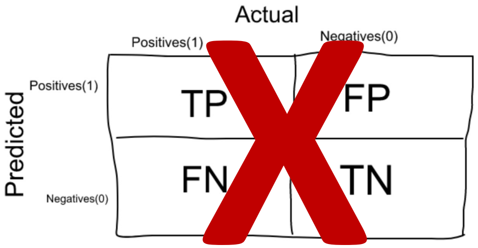
Performance evaluation protocol

- Split the data into: training, testing (and validation)
  - Fixed split approach
    - E.g. 70% training, 30% testing
  - Cross-validation approach (k-fold)



- Choose evaluation measures
- Perform measurements over n runs (n>= 1)

Performance evaluation

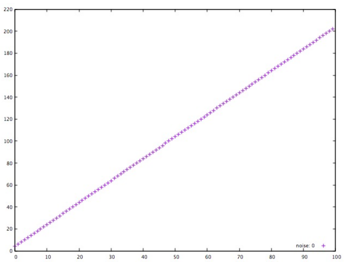


Confusion Matrix

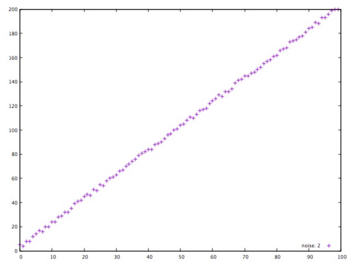
- Correlation coefficients (Pearson, Spearman, Kendall, …)
- Error functions
  - Mean absolute error
  - Mean absolute log error
  - Mean absolute perc. error
  - Root mean squared error
  - Root mean square log error
  - Root mean square perc. error
  - Root relative squared error
  - Relative absolute error
- …

Regression via multiple methods

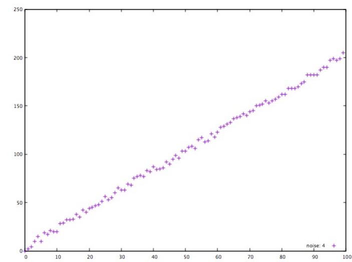
f(x) = 2x + 3 + Ɛ



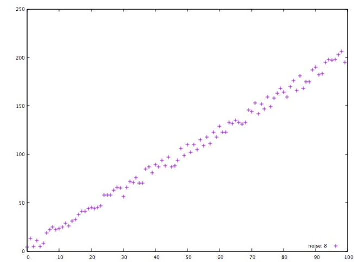
Ɛ = 0



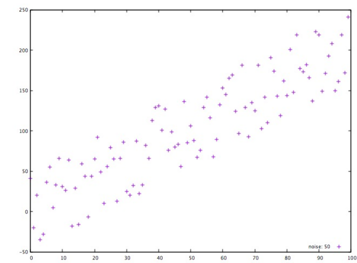
Ɛ ∈ [-2 .. +2]



Ɛ ∈ [-4 .. +4]



Ɛ ∈ [-8 .. 8]



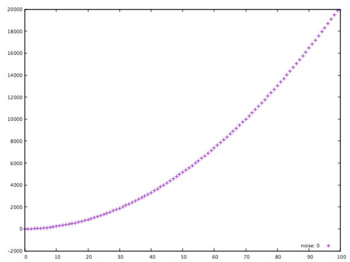
Ɛ ∈ [-50 .. 50]

Method	Ɛ = 0	Ɛ ∈ [-2 .. 2]	Ɛ ∈ [-4 .. 4]	Ɛ ∈ [-8 .. 8]	Ɛ ∈ [-50 .. 50]
Linear regression	**1.0000** 0.0000	**0.9998** 1.0805	**0.9992** 2.3775	**0.9968** 4.6496	**0.8803** 30.6469
Random Forest	**0.9998** 1.7155	**0.9996** 1.9921	**0.9986** 3.1983	**0.9953** 5.6982	**0.8173** 38.2657
ANN - MLP	**0.9995** 1.7715	**0.9993** 2.1454	**0.9985** 3.2222	**0.9967** 4.6653	**0.8740** 31.4028
Decision Table	0.9940 6.3388	0.9938 6.4299	0.9926 7.0318	0.9900 8.1663	0.8542 33.7570

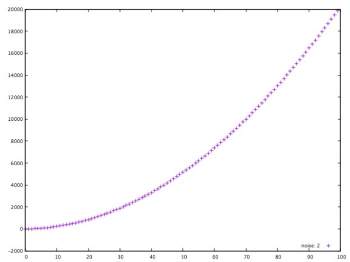
Pearson CCs & Root Mean Squared Errors (RMSE), n=100

Regression via multiple methods

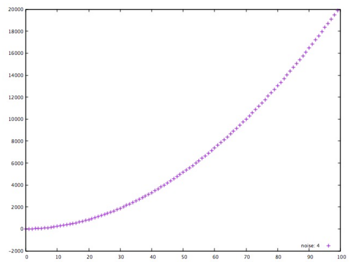
$f(x) = 2x^2 + 3x - 4 + \mathcal{E}$



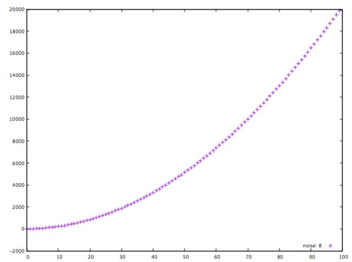
$\mathcal{E} = 0$



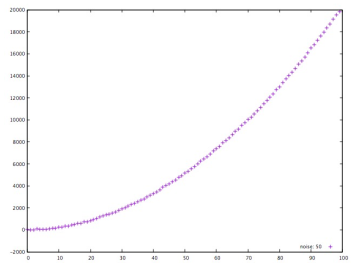
$\mathcal{E} \in [-2 .. +2]$



$\mathcal{E} \in [-4 .. +4]$



$\mathcal{E} \in [-8 .. 8]$



$\mathcal{E} \in [-50 .. 50]$

Method	$\mathcal{E} = 0$	$\mathcal{E} \in [-2 .. 2]$	$\mathcal{E} \in [-4 .. 4]$	$\mathcal{E} \in [-8 .. 8]$	$\mathcal{E} \in [-50 .. 50]$
Linear regression	0.9673 1520.0160	0.9673 1519.9055	0.9673 1519.8631	0.9673 1520.1075	0.9672 1521.9613
Random Forest	**0.9997** **190.2147**	**0.9997** **189.5386**	**0.9997** **188.0553**	**0.9997** **188.6072**	**0.9997** **185.6985**
ANN - MLP	**0.9997** **140.1311**	**0.9997** **142.1857**	**0.9998** **132.8442**	**0.9996** **161.0289**	**0.9998** **132.2034**
Decision Table	**0.9924** **737.0877**	**0.9924** **737.2094**	**0.9924** **736.5606**	**0.9924** **737.2194**	**0.9924** **734.9757**

Pearson CCs & Root Mean Squared Errors (RMSE)), n=100

Regression via multiple methods

$f(x,y,z,t) = 5x - 2\cos(y) + 3z^2/\mathrm{sqrt}(t) + \varepsilon$

Method	**$\varepsilon = 0$**	**$\varepsilon \in [-2 .. 2]$**	**$\varepsilon \in [-4 .. 4]$**	**$\varepsilon \in [-8 .. 8]$**	**$\varepsilon \in [-50 .. 50]$**
Linear regression	0.9415 29906.8719	0.9415 29906.6893	0.9415 29906.9306	0.9415 29906.2115	0.9415 29902.6757
Random Forest	**0.9997 2935.4949**	**0.9997 2935.3471**	**0.9997 2934.9531**	**0.9997 2935.7517**	**0.9997 2921.3079**
ANN - MLP	**0.9999 1399.9571**	**0.9998 1558.516**	**0.9999 1445.3883**	**0.9999 1392.7396**	**0.9999 1374.299**
Decision Table	**0.9907 12059.6908**	**0.9907 12059.7739**	**0.9907 12059.4813**	**0.9907 12059.8852**	**0.9907 12062.0892**

Pearson CCs & Root Mean Squared Errors (RMSE)), n=100

Clustering

LEARNING TYPES

DATA TYPES		*Supervised Learning*	*Unsupervised Learning*
	Discrete	classification or categorization	clustering
	Continuous	regression	dimensionality reduction

Clustering

- The most common **unsupervised learning** method

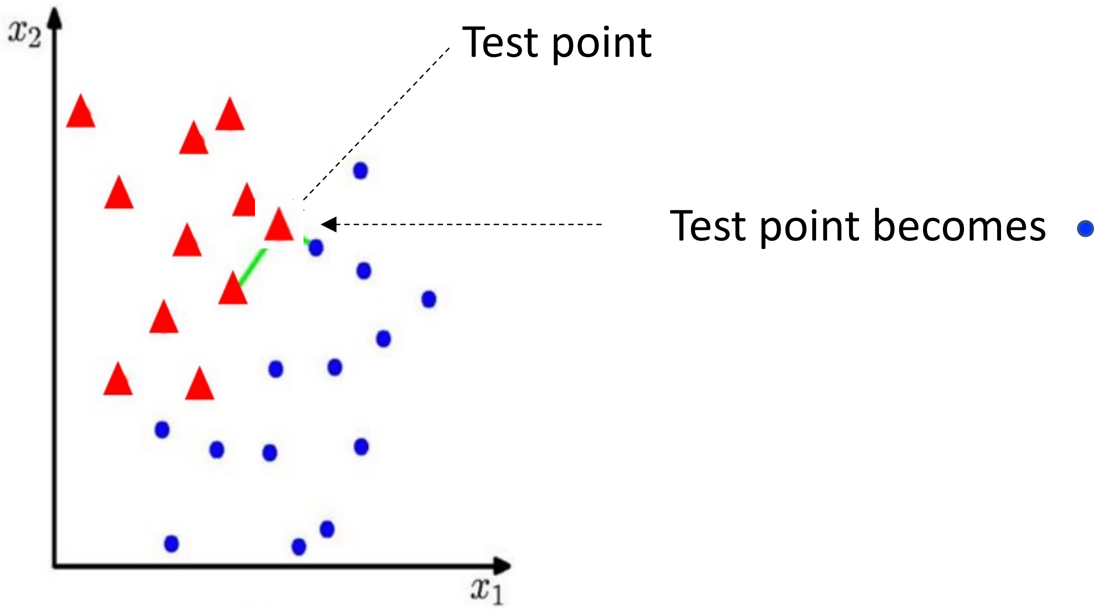
- Used for exploratory data analysis to:
  - Find hidden patterns in data
  - Find groupings in data

- Plethora of methods: KNN, K-means, hierarchical (e.g. neighbour joining), Gaussian mixture models, HMMs, self-organizing neural network maps (SOMs), …

K-Nearest Neighbour (KNN)

- For each test data point (that needs to be assigned a class), find the k-nearest labeled points in the data

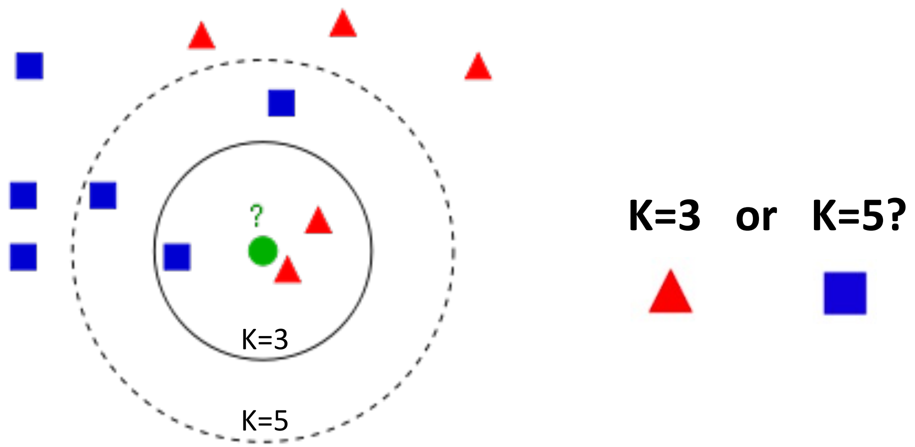
- The test point gets the class label of the majority



K-Nearest Neighbour (KNN)

- **Advantages:**
  - Simple and effective
  - Works on multi-class classification problems, too
  - Only a single parameter to tune (K)

- **Disadvantages:**
  - Accuracy depends on the distance metric
  - Sensitive to: the local structure of the data (skewed distributions), outliers, missing data



Dimensionality Reduction

LEARNING TYPES

DATA TYPES	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

Dimensionality Reduction

- Reduce the number of variables to be considered in future analysis

- Why?
  - Quicker and more accurate results from ML methods
  - Easier to visualize the data
  - Sometimes real relationships in the data are described by only a few dimensions (the rest is noise)

- A plethora of methods is available
  - **Types**: local, global and ensemble-like
  - Most of them rely on nearest-neighbour relations
  - **Examples**: PCA, manifold learning, ANN, ISOMAP, Diffusion mapping, Maximum variance unfolding, Locally Linear Embedding (LLE), Laplacian eigenmaps, Hessian LLE, Local Tangent Space Analysis, Ensemble trees, Random Forests, …

ML in Livestock

sensors MDPI

Review

Machine Learning in Agriculture: A Review

Konstantinos G. Liakos [1], Patrizia Busato [2], Dimitrios Moshou [1,3], Simon Pearson [4] and Dionysis Bochtis [1,*]

1 Institute for Bio-Economy and Agri-Technology (IBO), Centre of Research and Technology—Hellas (CERTH), 6th km Charilaou-Thermi Rd, GR 57001 Thessaloniki, Greece; k.liakos@certh.gr (K.G.L.); dmoshou@auth.gr (D.M.)

Animal Health Research Reviews

cambridge.org/ahr

Review

Cite this article: Nayeri S, Sargolzaei M, Tulpan D (2019). A review of traditional and machine learning methods applied to animal breeding. *Animal Health Research Reviews* **20**, 31–46. https://doi.org/10.1017/S1466252319000148

A review of traditional and machine learning methods applied to animal breeding

Shadi Nayeri[1], Mehdi Sargolzaei[2,3] and Dan Tulpan[1]

[1]Department of Animal Biosciences, Centre for Genetic Improvement of Livestock, University of Guelph, Guelph, Ontario, N1G 2W1, Canada; [2]Select Sires Inc., Plain City, Ohio, 43064, USA and [3]Department of Pathobiology, University of Guelph, Guelph, Ontario, N1G 2W1, Canada

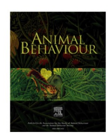
Abstract

The current livestock management landscape is transitioning to a high-throughput digital era where large amounts of information captured by systems of electro-optical, acoustical, mechanical, and biosensors is stored and analyzed on a daily and hourly basis, and actionable decisions are made based on quantitative and qualitative analytic results. While traditional animal breed-

Contents lists available at ScienceDirect

Animal Behaviour

journal homepage: www.elsevier.com/locate/anbehav



Review

Applications of machine learning in animal behaviour studies

CrossMark

John Joseph Valletta [a,*], Colin Torney [a], Michael Kings [b], Alex Thornton [b], Joah Madden [c]

[a] Centre for Mathematics and the Environment, University of Exeter, Penryn Campus, Penryn, U.K.
[b] Centre for Ecology and Conservation, University of Exeter, Penryn Campus, Penryn, U.K.
[c] Centre for Research in Animal Behaviour, University of Exeter, Exeter, U.K.

Computers and Electronics in Agriculture 162 (2019) 531–542

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Livestock vocalisation classification in farm soundscapes

James C. Bishop[a,b,*], Greg Falzon[a,b], Mark Trotter[c], Paul Kwan[b], Paul D. Meek[d,e]

[a] University of New England – Precision Agriculture Research Group (PARG), Armidale, NSW, Australia
[b] University of New England, School of Science and Technology, Armidale, NSW, Australia
[c] Institute for Future Farming Systems, CQUniversity, Rockhampton, QLD, Australia
[d] NSW Department of Primary Industries, PO Box 530, Coffs Harbour, NSW, Australia
[e] University of New England, School of Environmental and Rural Science, Armidale, NSW, Australia

Developing an ML model from Scratch - Process

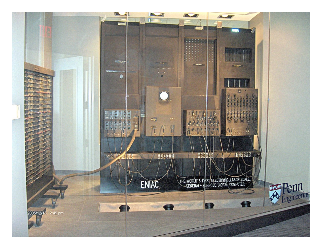
- **Clearly define the problem**
    - Objective, desired inputs and outputs
    - Is ML appropriate (good/best choice) for the problem?
- **Gather the data**
- **Prepare the data**
    - CLEAN THE DATA (if possible)
    - Re-format the data (image, txt, audio, etc. → tabular)
    - Deal with missing values, categorical vs. numerical values (encoding, scaling), …
    - Feature selection (use meaningful features) → dimensionality reduction (e.g. PCA, …)
    - Shuffle data if needed and if it makes sense (not temporal data)
    - Data splitting: training, testing, validation
- **Choose the evaluation measures**
    - Dependent on the type of problem (classification, regression, …)
    - Note: You can only improve what you can measure!!!
- **Choose an evaluation protocol**
    - fixed split, k-fold cross validation, …
- **Think about over- and under-fitting your data and how to avoid it**
- **Explore models before selecting one or more → Hands-on workshop using Weka**
    - Ideally with minimum programming effort
- **Choose one or more promising models**
- **Tune the chosen models (hyper-parameter optim.) to optimize performance**
    - Grid search, random search, etc.

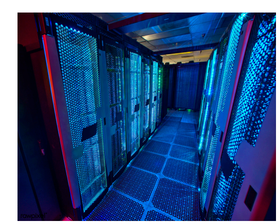
Developing an ML model from Scratch – Practical Considerations

- **Feature selection**
  - Future model use vs. “theoretical beauty” (publication worthiness)

- **Model implementation**
  - Results variability depending on implementations
  - Saving a model is sometimes problematic

- **Model training, testing & deployment**
  - Dependency on hardware, OS and software





Hands-on/Demo Workshop

- Need to install Weka: https://www.cs.waikato.ac.nz/ml/weka/

- Use `Tools`→`Package Manager` to install all models
  - Takes approximately ~15 min and requires a lot of mouse clicks
  - *Note: Do not worry about warning messages. Some external packages are not compatible with the latest version of Weka.*

- Detailed instructions and materials:

http://animalbiosciences.uoguelph.ca/~dtulpan/conferences/asas2020_mlworkshop/

Or **https://tinyurl.com/yyx8p473**

References

- Theoretical ML
  - Books (Free):
    - Kubat (2017): An Introduction to Machine Learning
    - James et al. (2017): An Introduction to Statistical Learning
    - Hastie et al. (2017): The Elements of Statistical Learning
    - Leskovec et al. (2017): Mining of Massive Datasets
  - Books (Not Free)
    - Murphy (2012): Machine Learning: a Probabilistic Perspective
    - Marsland (2009): Machine Learning: an Algorithmic Perspective

- Practical ML
  - https://towardsdatascience.com/machine-learning-general-process-8f1b510bd8af

- Online tutorials
  - https://www.tutorialandexample.com/machine-learning-tutorial/

- Libraries and software tools
  - Weka
  - KNIME,
  - Python: scikit-learn, PyTorch, Keras
  - JavaScript: TensorFlow
  - R: caret
  - Apache: Mahout
  - RapidMiner

Thank you

UNIVERSITY of GUELPH

FOOD FROM THOUGHT

UNIVERSITY of GUELPH

ONTARIO AGRICULTURAL COLLEGE

DEPARTMENT OF ANIMAL BIOSCIENCES

CGIL

Centre for Genetic Improvement of Livestock

AMERICAN SOCIETY OF ANIMAL SCIENCE

CDN Canadian Dairy Network

DAIRY at GUELPH

CANADA'S DAIRY UNIVERSITY