What is learning?

"The activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something."

Merriam Webster dictionary

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Tom Mitchell

Machine learning as meta-programming

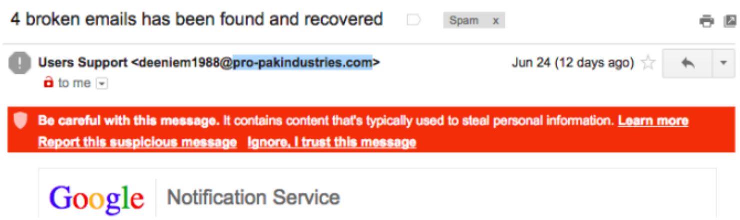
- For many problems, it's difficult to program the correct behavior by hand
  - recognizing people and objects
  - understanding human speech
- Machine learning approach: program an algorithm to automatically learn from data, or from experience, and output a program, typically to solve a prediction problem:
  - Given an **input** x,
  - **Predict** an **output** y.

- Why might you want to use a learning algorithm?
  - hard to code up a solution by hand (e.g. vision, speech)
  - system needs to adapt to a changing environment (e.g. spam detection)

Example: Spam Detection

Let's start with a few canonical examples.

- **Input x:** Incoming email



- **Output y:** "SPAM" or "NOT SPAM"
- This is a **binary classification** problem: there are two possible outputs.

Example: Medical Diagnosis

- **Input x:** Symptoms (fever, cough, fast breathing, shaking, nausea, ...)
- **Output y:** Diagnosis (pneumonia, flu, common cold, bronchitis, ...)
- A **multiclass classification** problem: choosing an output out of a *discrete* set of possible outputs.

How do we express uncertainty about the output?

- **Probabilistic classification** or **soft classification**:

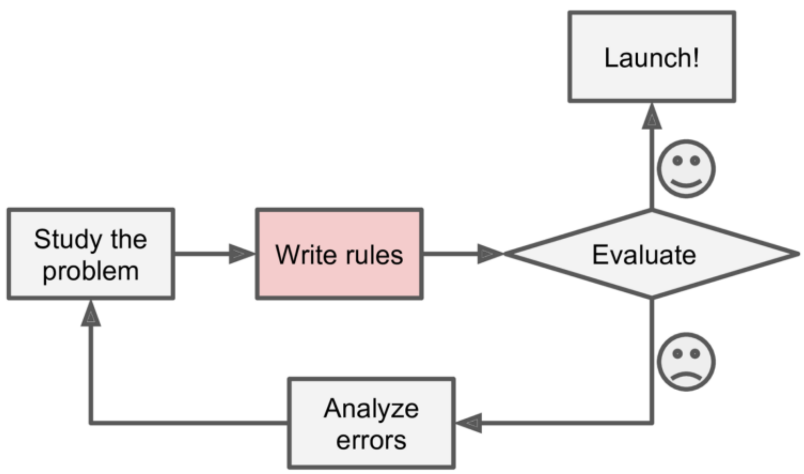
$$
\begin{aligned}
\mathbb{P}(\text{pneumonia}) &= 0.7 \\
\mathbb{P}(\text{flu}) &= 0.2 \\
\vdots \quad &\quad \vdots
\end{aligned}
$$

Comparison to Rule-Based Approaches (Expert Systems)

Consider the problem of medical diagnosis.

1. Talk to experts (in this case, medical doctors).
2. Understand how the experts come up with a diagnosis.
3. Implement this process as an algorithm (a **rule-based system**): e.g., a set of symptoms $\rightarrow$ a particular diagnosis.
4. Use logical deduction to infer new rules from the rules that are stored in the knowledge base.

Rule-Based Approach



Advantages of Rule-Based Approaches

- Leverage existing domain expertise.
- Generally **interpretable**: We can describe the rule to another human
- Produce reliable answers for the scenarios that are included in the knowledge bases.

Limitations of Rule-Based Systems

- Labor intensive to build: experts' time is expensive.
- Rules work very well for areas they cover, but often do not **generalize** to unanticipated input combinations.
- Don't naturally handle uncertainty.

The Machine Learning Approach

- Instead of explicitly engineering the process that a human expert would use to make the decision...
- We have the machine **learn** on its own from inputs and outputs (decisions).
- We provide **training data**: many examples of (input x , output y) pairs, e.g.
  - A set of videos, and whether or not each has a cat in it.
  - A set of emails, and whether or not each one should go to the spam folder.
- Learning from training data of this form (inputs and outputs) is called **supervised learning**.

Manuela Veloso, PhD

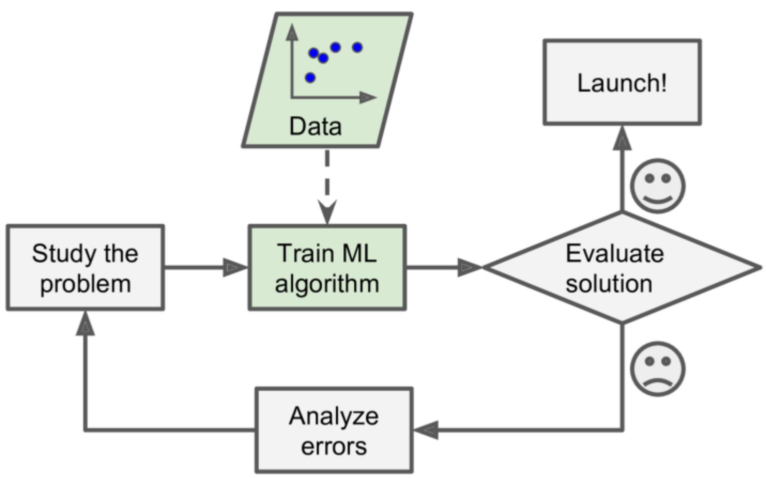
Head of AI Research, JPMorgan Chase & Co.

Dr. Manuela Veloso is Head of J.P. Morgan Chase AI Research and Herbert A. Simon University Professor Emerita at Carnegie Mellon University, where she was previously Faculty in the Computer Science Department and Head of the Machine Learning Department. Her recent interests are in Artificial Intelligence (AI), Symbiotic Human-Robot Autonomy, Continuous Learning Systems, and AI in Finance.

Machine Learning Algorithm

- A **machine learning algorithm** learns from the training data:
  - **Input**: Training Data (e.g., emails x and their labels y)
  - **Output**: A prediction function that produces output y given input x.
- The goal of machine learning is to find the "best" (to be defined) prediction function **automatically, based on the training data**
- The success of ML depends on
  - The availability of large amounts of data;
  - **Generalization** to unseen samples (the test set): just memorizing the training set will not be useful.

Machine Learning Approach



Key concepts

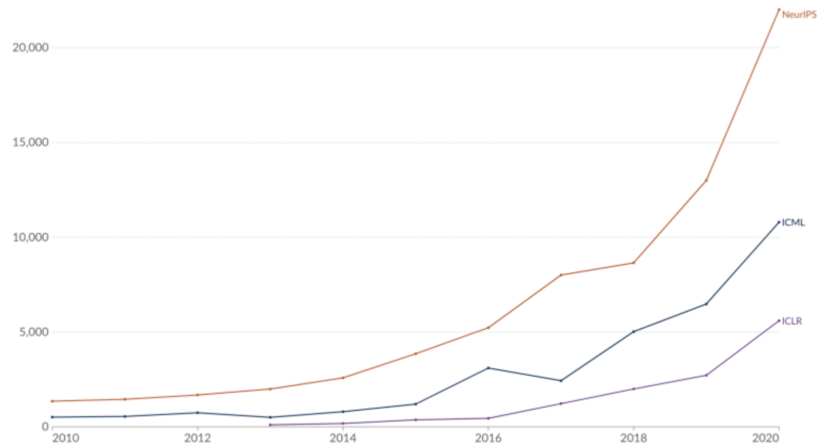
- The most common **ML problem types**:
  - Classification (binary and multiclass)
  - Regression
- **Prediction function**: predicts output y (e.g. spam or not?) given input x (e.g. email)
- **Training data**: a set of (input x, output y) pairs
- **Supervised learning algorithm**: takes training data and produces a prediction function
- Beyond prediction
  - **Unsupervised learning**: finding structures in data, e.g. clustering

Relations to human learning

- It is tempting to imagine machine learning as a component in AI just like human learning in ourselves.
- Human learning is:
  - Very data efficient
  - An entire multitasking system (vision, language, motor control, etc.)
  - Flexible to adapt new skills
  - Takes at least a few years :)
- For serving specific purposes, machine learning doesn't have to look like human learning in the end.
- Machines may borrow ideas from biological systems (e.g. neural networks).

History of machine learning

Top ML conferences attendance over year:



Regression Problem Motivation

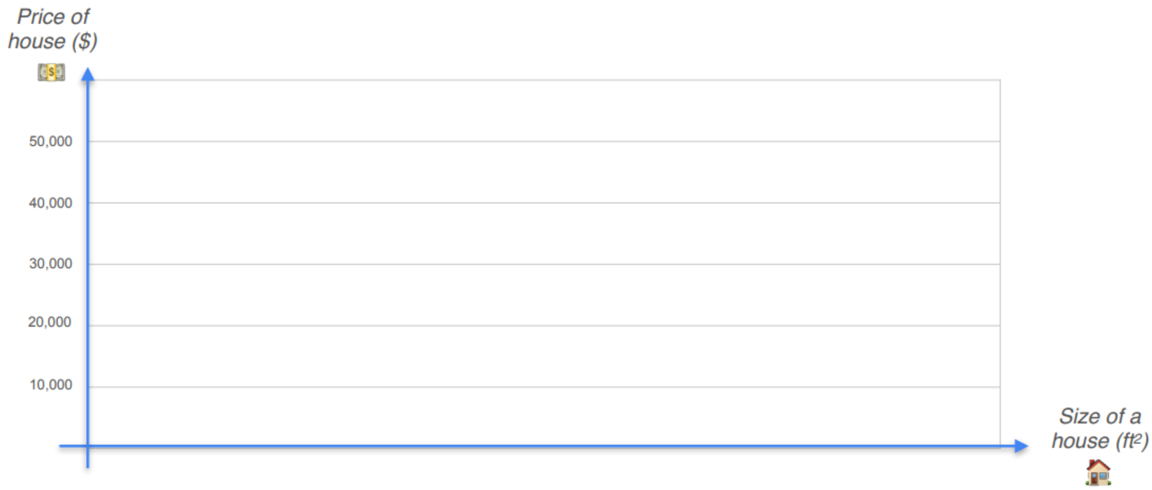
Predicting

the price of a house

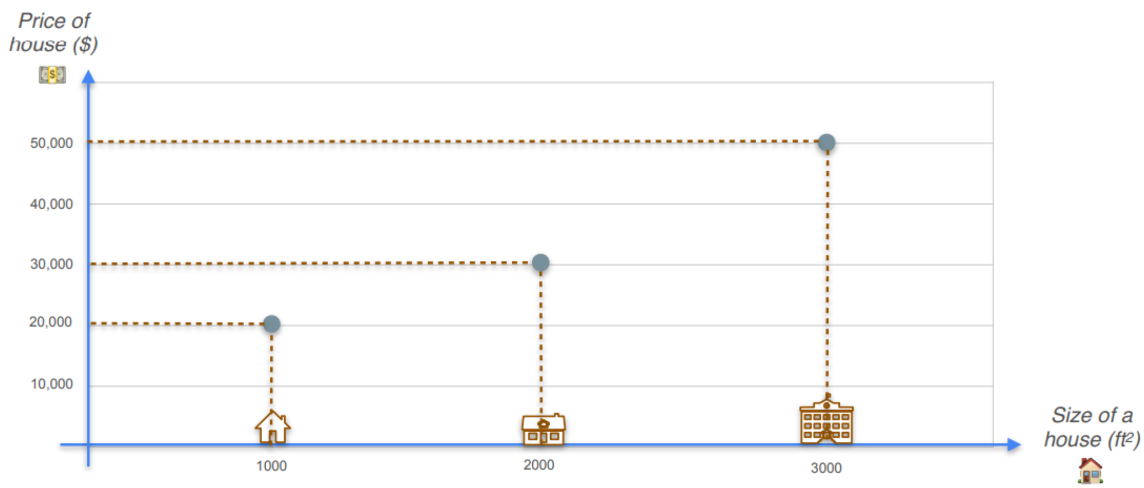
from

the size of the house

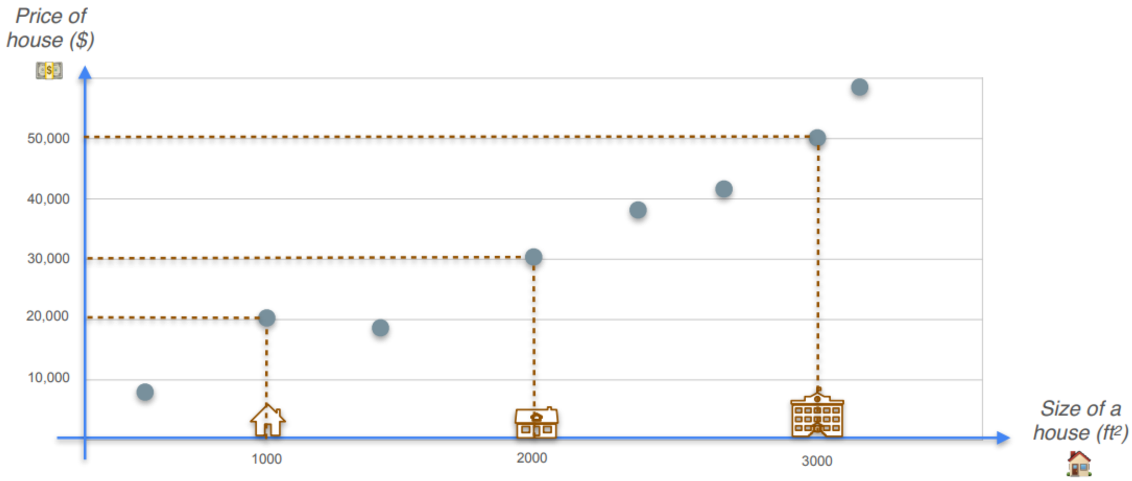
Regression Problem Motivation



Regression Problem Motivation



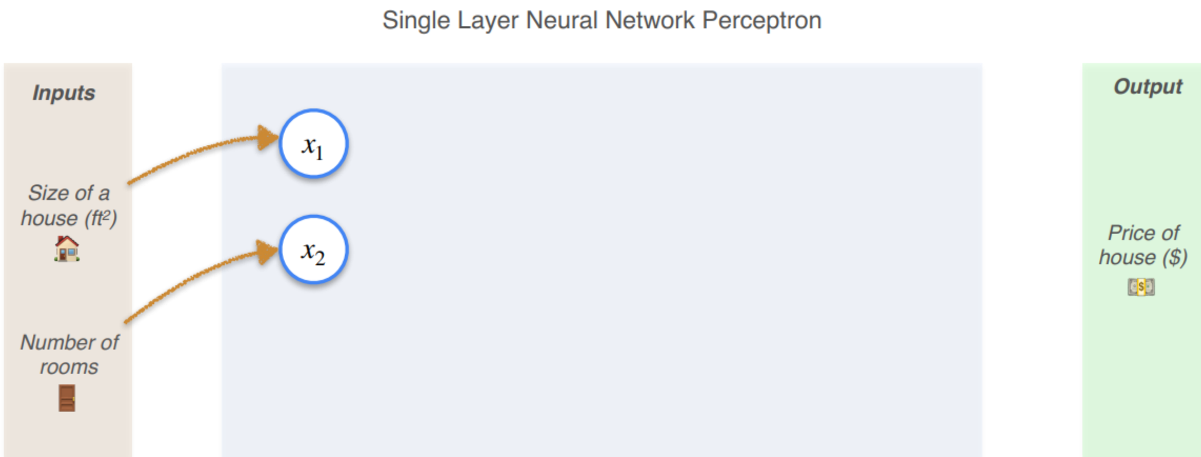
Regression Problem Motivation



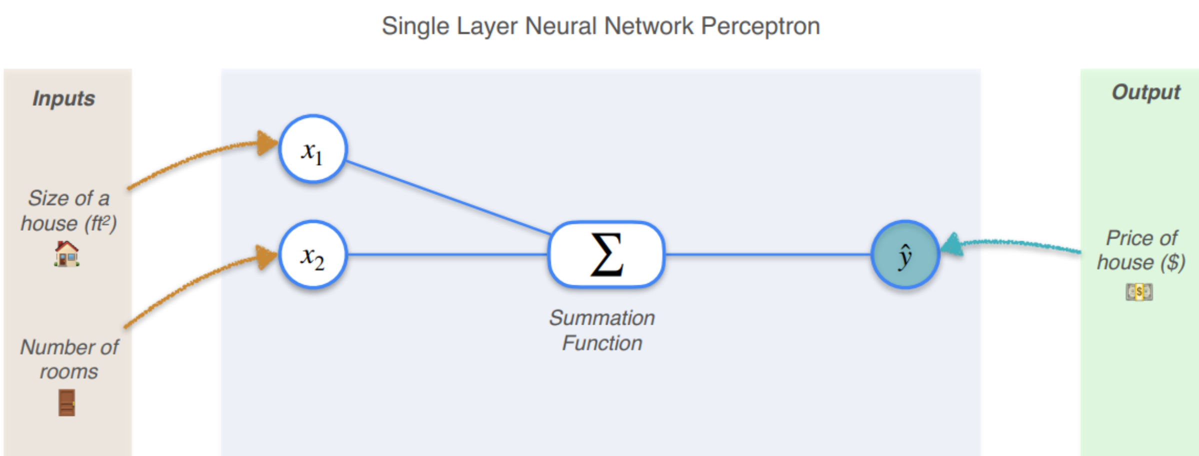
Regression With a Perceptron

	Size of a house (ft²)	Number of rooms	Price of house ($)
	1000ft²	*2*	*$20,000*
	2000ft²	*4*	*$30,000*
	3000ft²	*7*	*$50,000*

Regression With a Perceptron

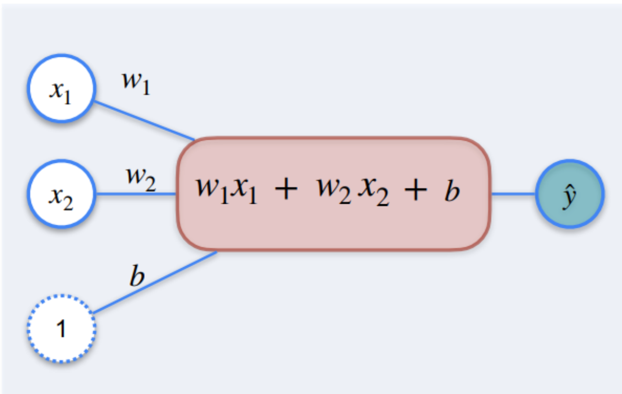


Regression With a Perceptron



Regression With a Perceptron

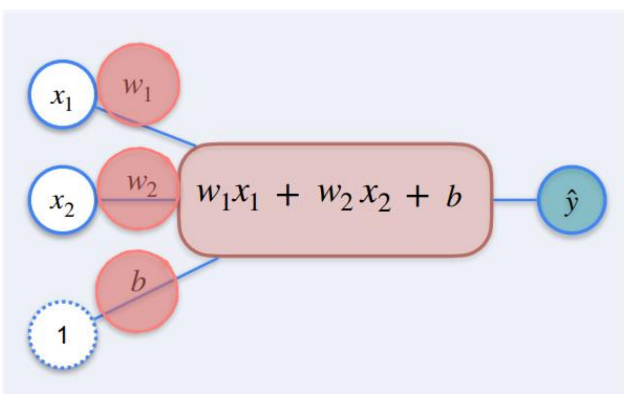
Single Layer Neural Network Perceptron



$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Regression With a Perceptron

Single Layer Neural Network Perceptron



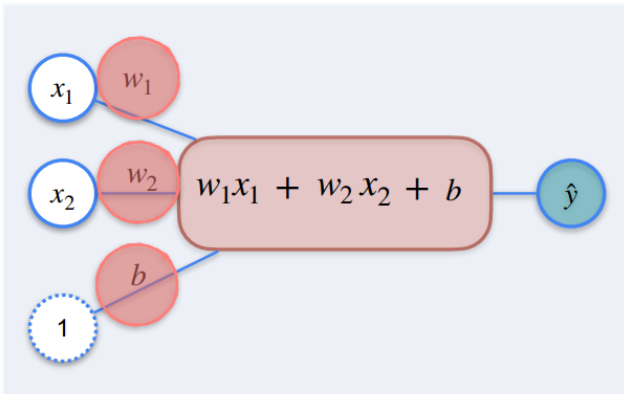
$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Main Goal:

Find weights and bias that will optimise the predictions.

Regression With a Perceptron

Single Layer Neural Network Perceptron



$$\hat{y} = w_1x_1 + w_2x_2 + b$$

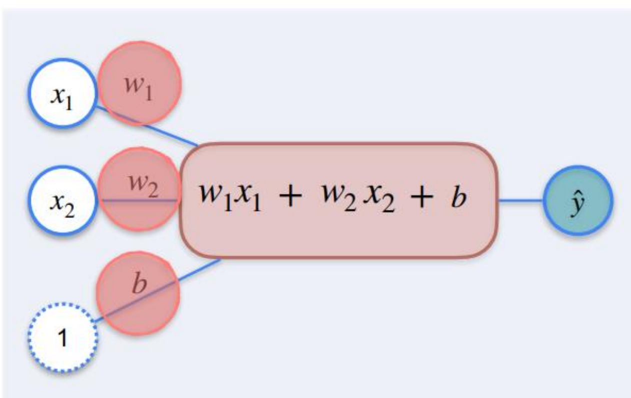
Main Goal:

Find weights and bias that will optimise the predictions.

ie. Reduce the errors in the predictions

Regression With a Perceptron

Single Layer Neural Network Perceptron



$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Main Goal:

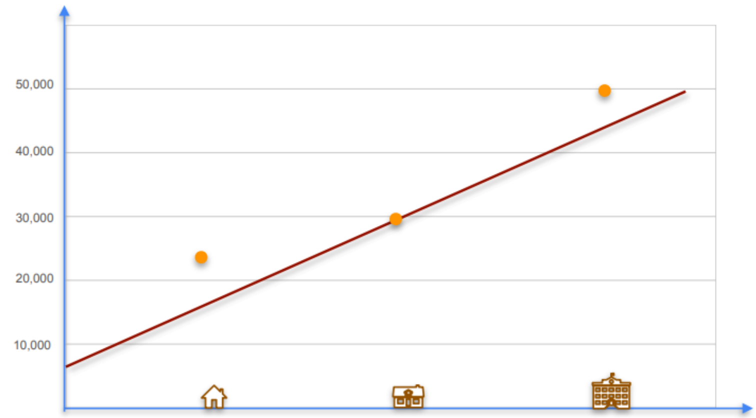
Find weights and bias that will optimise the predictions.

ie. Reduce the errors in the predictions

The Loss Function

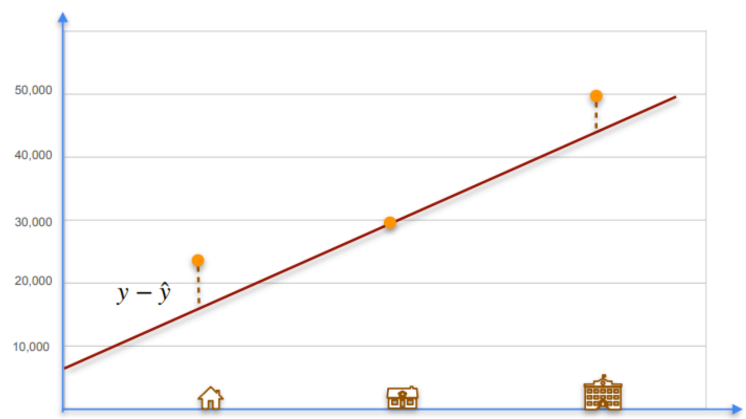
Mean Squared Error

	y	$\hat{y}$	
	\$20,000	*\$15,000*	
	\$30,000	*\$30,000*	
	\$50,000	*\$45,000*	



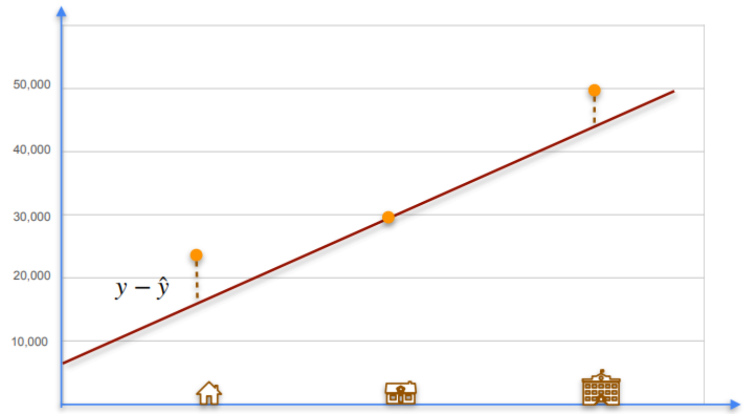
Mean Squared Error

	y	$\hat{y}$	$y - \hat{y}$
	\$20,000	*\$15,000*	***Error***
	\$30,000	*\$30,000*	***Error***
	\$50,000	*\$45,000*	***Error***



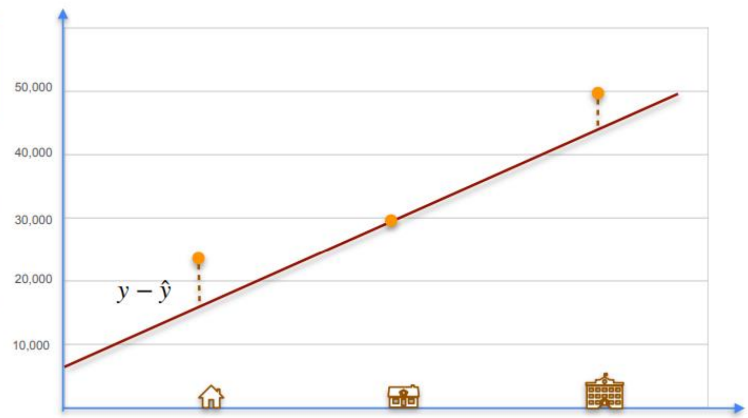
Mean Squared Error

	y	$\hat{y}$	$(y-\hat{y})^2$
	\$20,000	*\$15,000*	***Error***
	\$30,000	*\$30,000*	***Error***
	\$50,000	*\$45,000*	***Error***



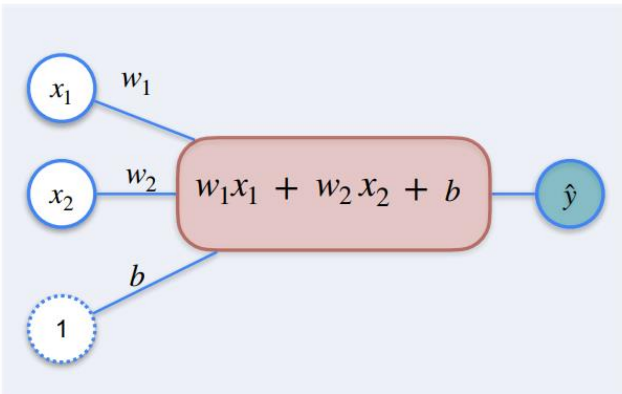
Mean Squared Error

	y	$\hat{y}$	$\frac{1}{2}(y-\hat{y})^2$
	\$20,000	*\$15,000*	***Error***
	\$30,000	*\$30,000*	***Error***
	\$50,000	*\$45,000*	***Error***



Regression With a Perceptron

Single Layer Neural Network Perceptron



Prediction Function:

$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Loss Function:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

Main Goal:

Find w_1 , w_2 , b that give $\hat{y}$ with the least error

Regression With a Perceptron

Prediction Function:

$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Loss Function:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

Main Goal:

Find w_1 , w_2 , b that give $\hat{y}$ with the least error

To find optimal values for:
w_1 , w_2 , b

You need gradient descent

$$w_1 \rightarrow w_1 - \alpha\frac{\partial L}{\partial w_1}$$

$$w_2 \rightarrow w_2 - \alpha\frac{\partial L}{\partial w_2}$$

Regression With a Perceptron

Prediction Function:

$$\hat{y} = w_1x_1 + w_2x_2 + b$$

Loss Function:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

Main Goal:

Find w_1 , w_2 , b that give $\hat{y}$ with the least error

To find optimal values for:
w_1 , w_2 , b

You need gradient descent

Some initial starting values

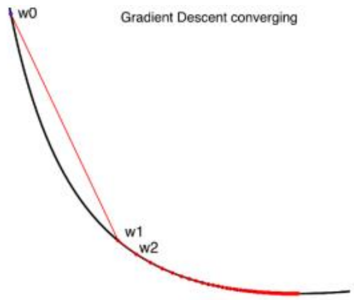
$$w_1 \rightarrow w_1 - \alpha\frac{\partial L}{\partial w_1}$$

$$w_2 \rightarrow w_2 - \alpha\frac{\partial L}{\partial w_2}$$

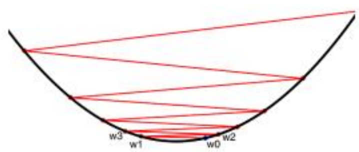
$$b \rightarrow b - \alpha\frac{\partial L}{\partial b}$$

SUB-TASK

Find the following partial derivatives



Gradient Descent diverging (stepsize too large)

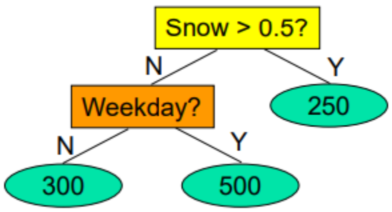


Lasso Regression (Tikhonov Form, soft penalty)

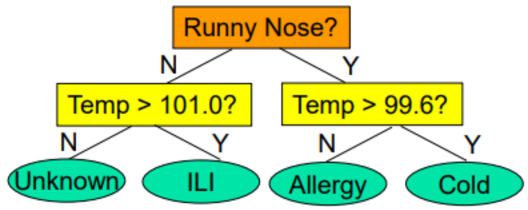
$$\hat{w} = \underset{w \in \mathrm{R}^d}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda \|w\|_1,$$

where $\|w\|_1 = |w_1| + \cdots + |w_d|$ is the ℓ_1-norm.

- A <u>decision tree</u> is a set of rules that can be learned from data and used to predict an unknown value.
  - Unknown real value (regression): What is the expected incidence of car thefts in NYC on a given day?
  - Unknown category value (classification): What type of illness does patient X have, given their symptoms and demographic data?



How many thefts on Tuesday, January 3 (0.2 inches of snow)?



What do we predict for a patient with Temp = 100 and a runny nose?

Learning binary decision trees

<u>Example dataset</u>:
Predicting whether a car is fuel-efficient, given its number of cylinders (4, 6, or 8), weight (light, medium, or heavy), and horsepower (real-valued).

<u>MPG, cylinders, HP, weight</u>

good, 4, 75, light
bad, 6, 90, medium
bad, 4, 110, medium
bad, 8, 175, weighty
bad, 6, 95, medium
bad, 4, 94, light
bad, 4, 95, light
bad, 8, 139, weighty
bad, 8, 190, weighty
bad, 8, 145, weighty
bad, 6, 100, medium
good, 4, 92, medium
bad, 6, 100, weighty
bad, 8, 170, weighty
good, 4, 89, medium
good, 4, 65, light
bad, 6, 85, medium
bad, 4, 81, light
bad, 6, 95, medium
good, 4, 93, light

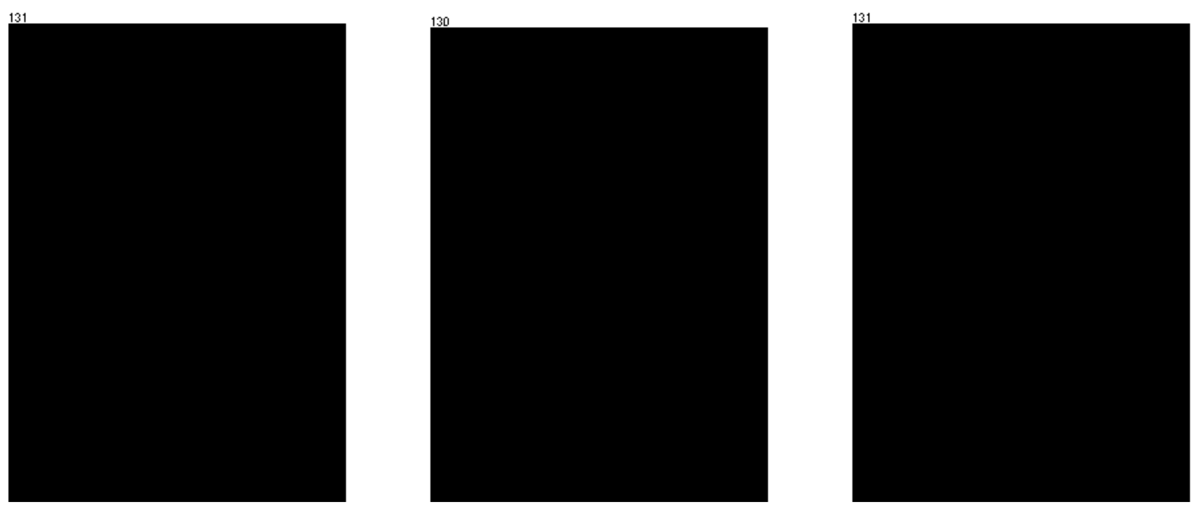
1. Discretize mpg

Name: mpg
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-18.8]'	131	131
2	'(18.8-26.9]'	130	130
3	'(26.9-inf)'	131	131

Class: modelyear (Num)

Visualize All



2. Change Numeric to Nominal

Selected attribute

Name: cylinders   Type: Nominal

Missing: 0 (0%)   Distinct: 5   Unique: 0 (0%)

No.	Label	Count	Weight
1	3	4	4
2	4	199	199
3	5	3	3
4	6	83	83
5	8	103	103

Class: modelyear (Num)   Visualize All

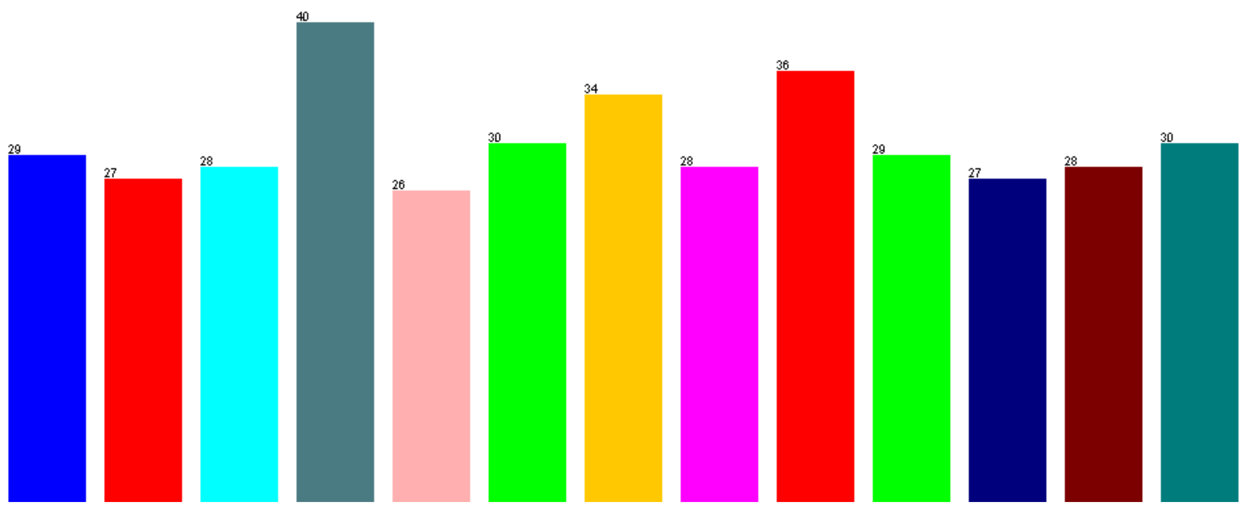
Selected attribute

Name: modelyear — Type: Nominal

Missing: 0 (0%) — Distinct: 13 — Unique: 0 (0%)

No.	Label	Count	Weight
1	70	29	29
2	71	27	27
3	72	28	28
4	73	40	40
5	74	26	26
6	75	30	30
7	76	34	34
8	77	28	28
9	78	36	36
10	79	29	29
11	80	27	27
12	81	28	28
13	82	30	30

Class: modelyear (Nom) — Visualize All



3. Classify: ZeroR

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     mpg_all-weka.filters.unsupervised.attribute.Discretize-F-B3-M-1.0-Rfirst-precision6-unset-class-temporarily-weka.filters.unsupervised.attribute.NumericToNominal-R2,7
Instances:    392
Attributes:   7
              mpg
              cylinders
              displacement
              horsepower
              weight
              acceleration
              modelyear
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

ZeroR predicts class value: '(-inf-18.8]'

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         130               33.1633 %
Incorrectly Classified Instances       262               66.8367 %
Kappa statistic                         -0.0038
Mean absolute error                      0.4444
Root mean squared error                  0.4714
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              392

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.893    0.900    0.332      0.893    0.484      -0.011   0.495     0.332     '(-inf-18.8]'
                 0.000    0.000    ?          0.000    ?          ?        0.497     0.330     '(18.8-26.9]'
                 0.099    0.103    0.325      0.099    0.152      -0.007   0.495     0.332     '(26.9-inf)'
Weighted Avg.    0.332    0.335    ?          0.332    ?          ?        0.496     0.331

=== Confusion Matrix ===

   a   b   c   <-- classified as
 117   0  14 |   a = '(-inf-18.8]'
 117   0  13 |   b = '(18.8-26.9]'
 118   0  13 |   c = '(26.9-inf)'
```

4. Classify: Decision Tree (J48): binarySpilts, reducedErrorPruning, SaveInstanceData

```
displacement <= 200.0
|   horsepower <= 84.0: '(26.9-inf)' (86.0/18.0)
|   horsepower > 84.0
|   |   cylinders = 3: '(18.8-26.9]' (4.0/1.0)
|   |   cylinders != 3
|   |   |   modelyear = 80: '(26.9-inf)' (5.0/1.0)
|   |   |   modelyear != 80
|   |   |   |   modelyear = 82: '(26.9-inf)' (7.0/2.0)
|   |   |   |   modelyear != 82: '(18.8-26.9]' (53.0/11.0)
displacement > 200.0
|   displacement <= 232.0: '(18.8-26.9]' (21.0/7.0)
|   displacement > 232.0: '(-inf-18.8]' (86.0/7.0)

Number of Leaves  :     7

Size of the tree :      13


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         322               82.1429 %
Incorrectly Classified Instances        70               17.8571 %
Kappa statistic                          0.7321
Mean absolute error                      0.1669
Root mean squared error                  0.3272
Relative absolute error                 37.549  %
Root relative squared error             69.4009 %
Total Number of Instances              392

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.885    0.077    0.853      0.885    0.869      0.802    0.938     0.842     '(-inf-18.8]'
                 0.692    0.111    0.756      0.692    0.723      0.596    0.792     0.639     '(18.8-26.9]'
                 0.885    0.080    0.847      0.885    0.866      0.796    0.920     0.807     '(26.9-inf)'
Weighted Avg.    0.821    0.089    0.819      0.821    0.819      0.732    0.883     0.763

=== Confusion Matrix ===

   a   b   c   <-- classified as
 116  15   0 |   a = '(-inf-18.8]'
  19  90  21 |   b = '(18.8-26.9]'
   1  14 116 |   c = '(26.9-inf)'
```

5. Visualizer tree