Reminders

Upcoming due dates
Mon Nov 10 Q6
Wed Nov 12 Data checkpoint
Fri Nov 14 D6

Machine Learning

Data Science in Practice

Jason G. Fleischer, PhD Dept. of Cognitive Science UC San Diego

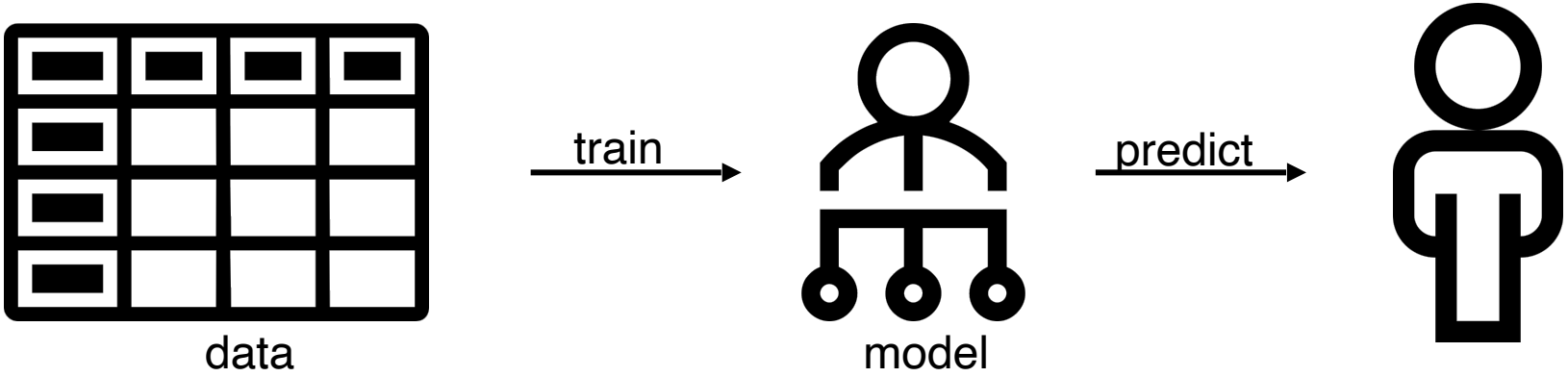
A survey!

- **Problem:** Detecting whether credit card charges are fraudulent.
- **Data science question:** Can we use the time of the charge, the location of the charge, and the price of the charge to predict whether that charge is fraudulent or not?
- **Type of analysis:** Predictive analysis

Robert Hecht-Neilsen and Zeus (and others) sold HNC for $810M in 2002
Around here lots of people see him as a major contributor to the development of neural networks and data science as we know them today

predictive analysis
uses data you have now to make predictions in the future

machine learning
approaches are used for predictive analysis!



In contrast to statistical approaches which care more about the model accurately reflecting the process than nailing the predictions

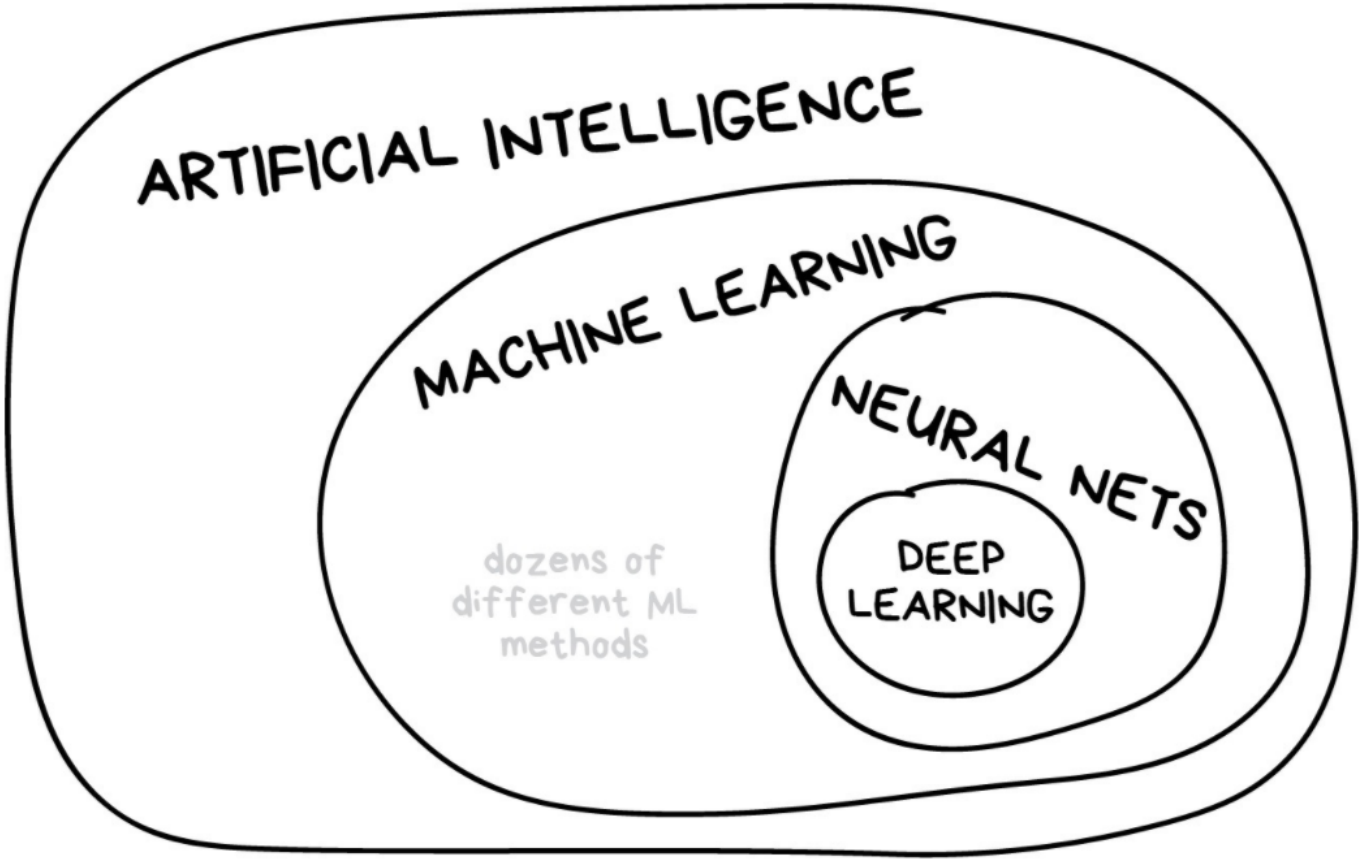


Image source: https://vas3k.com/blog/machine_learning/

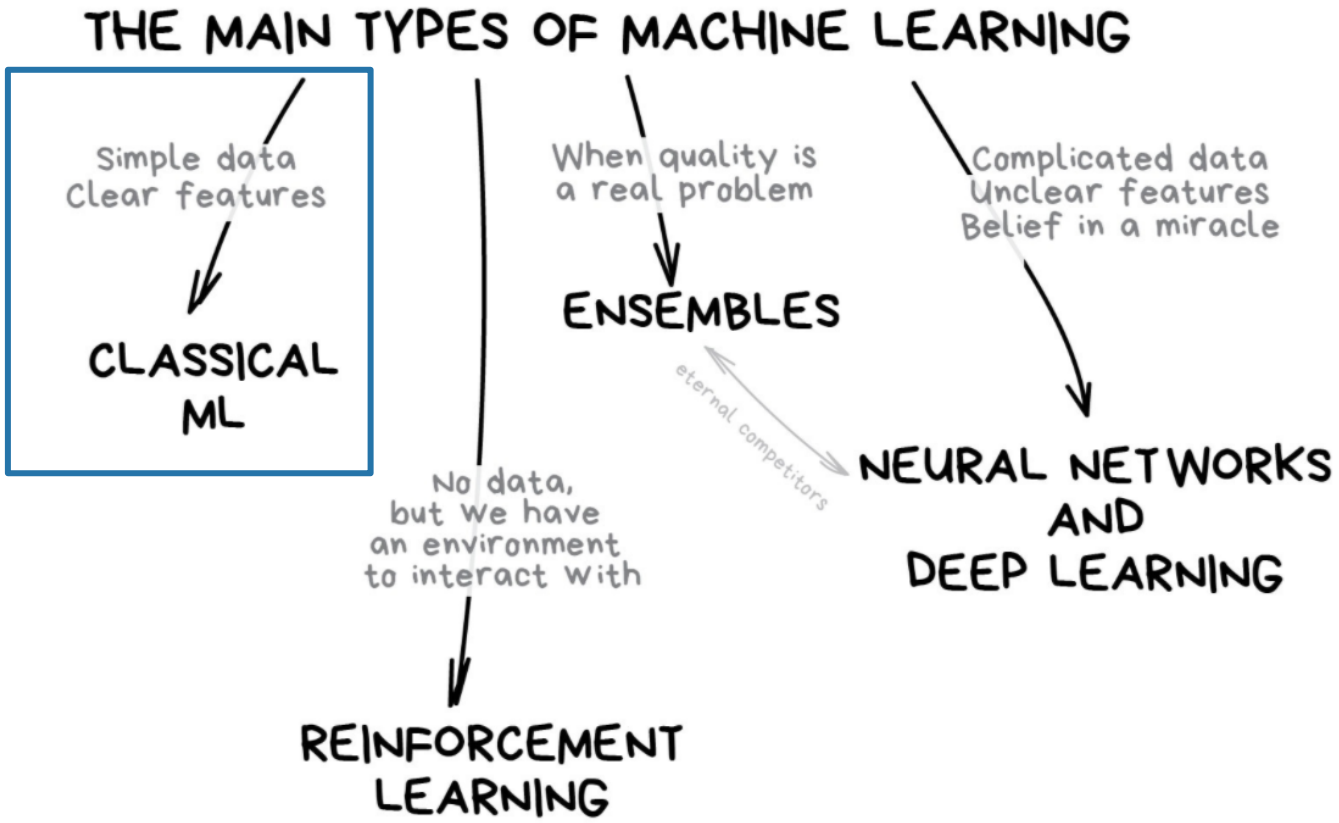
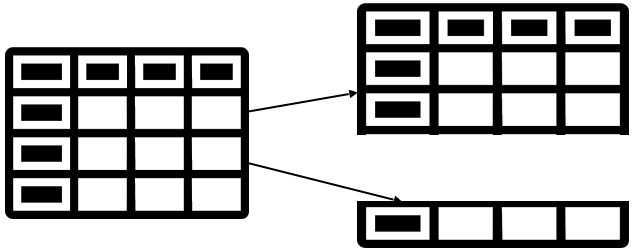
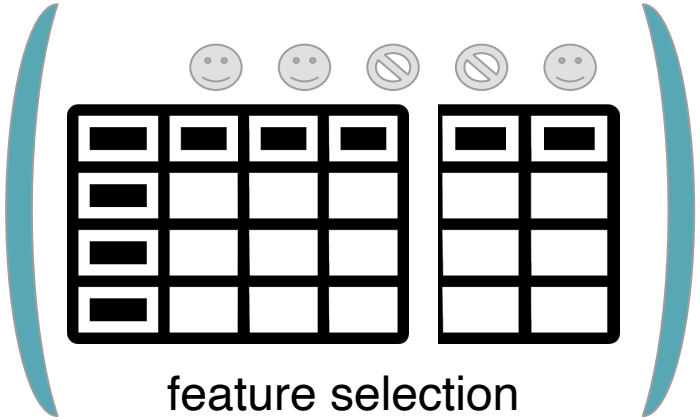


Image source: https://vas3k.com/blog/machine_learning/

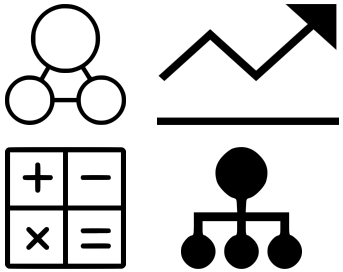
Machine Learning - The usual ingredients in the recipe



data
partitioning



feature selection



model selection

model assessment

Drowning in data?

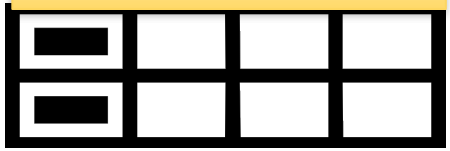
Image borrowed from
https://medium.com/@tmaoconnor/marketing-is-drowning-in-data-yet-its-starving-for-answers-f15ee890a732

Limited data?

Data partitioning:
Evaluating how well you will generalize to new data with huge amounts of data

All the data

training set

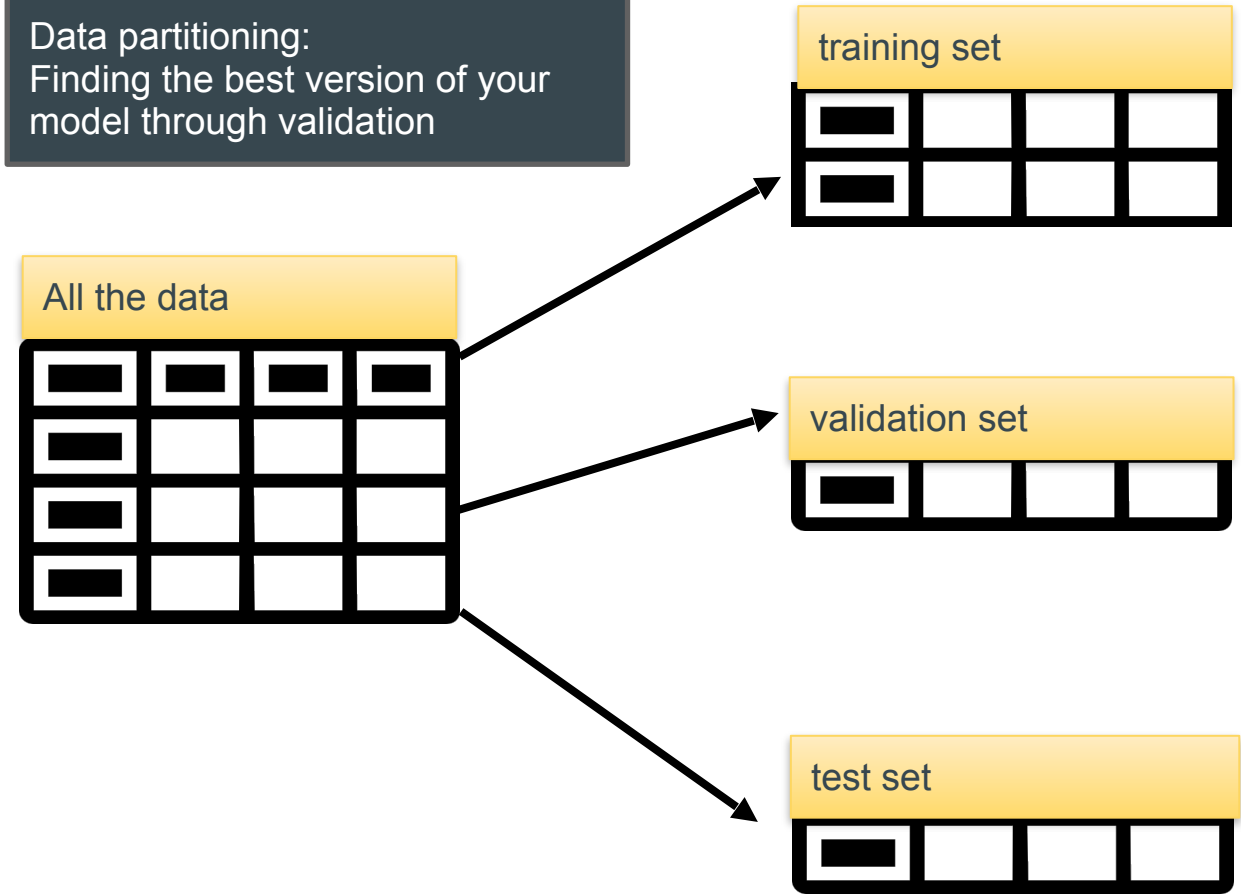


Data used to build your model

test set

Used to assess how good the final model is (Generalization)

Data partitioning:
Finding the best version of your model through validation

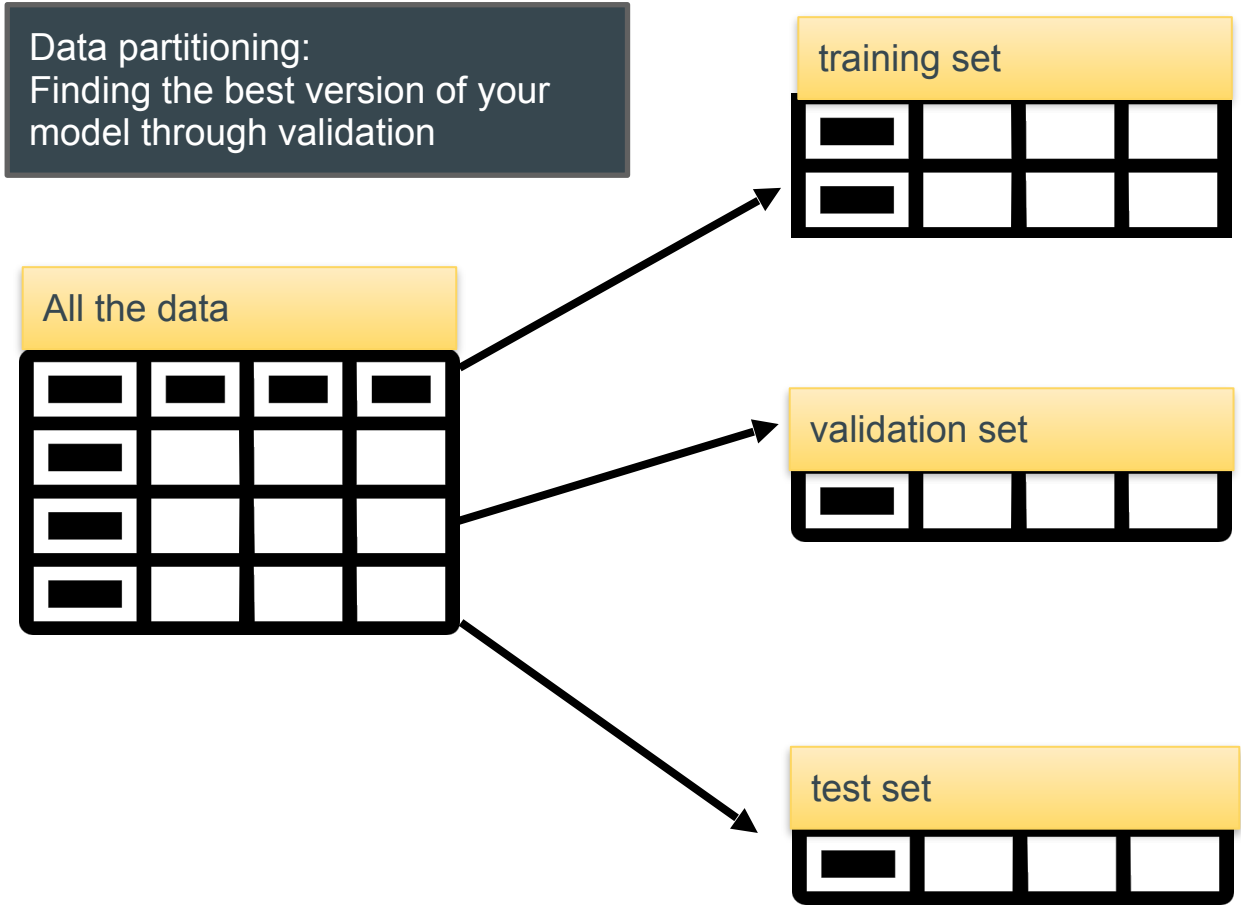


Data used to build your model

Fine-tune the model to increase prediction accuracy (Hyper-parameter tuning, feature selection, model selection)

Used to assess how good the final model is (Generalization)

Data partitioning:
Finding the best version of your model through validation



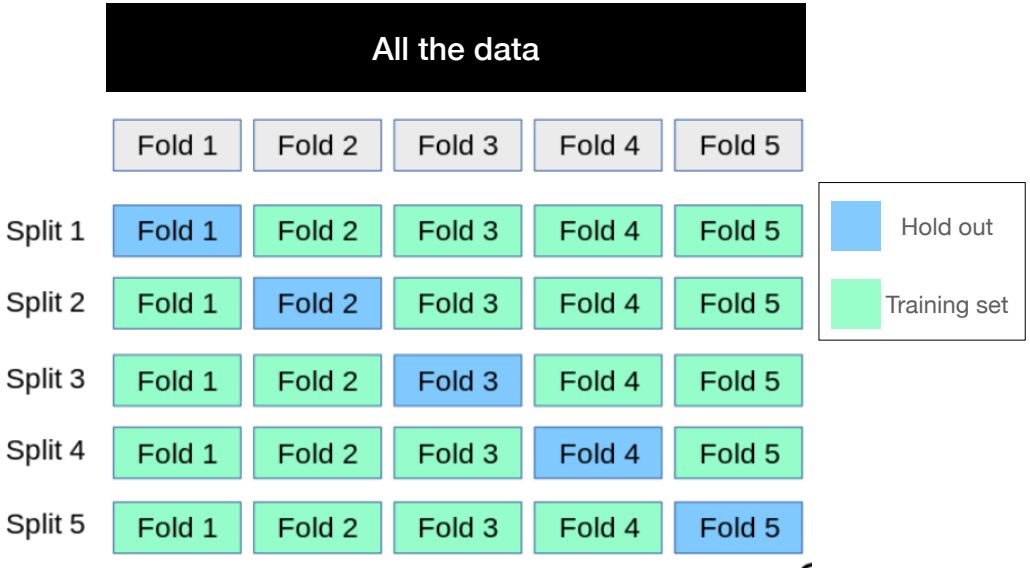
Typically biggest chunk
50 - 80%

Smaller
10 - 25%

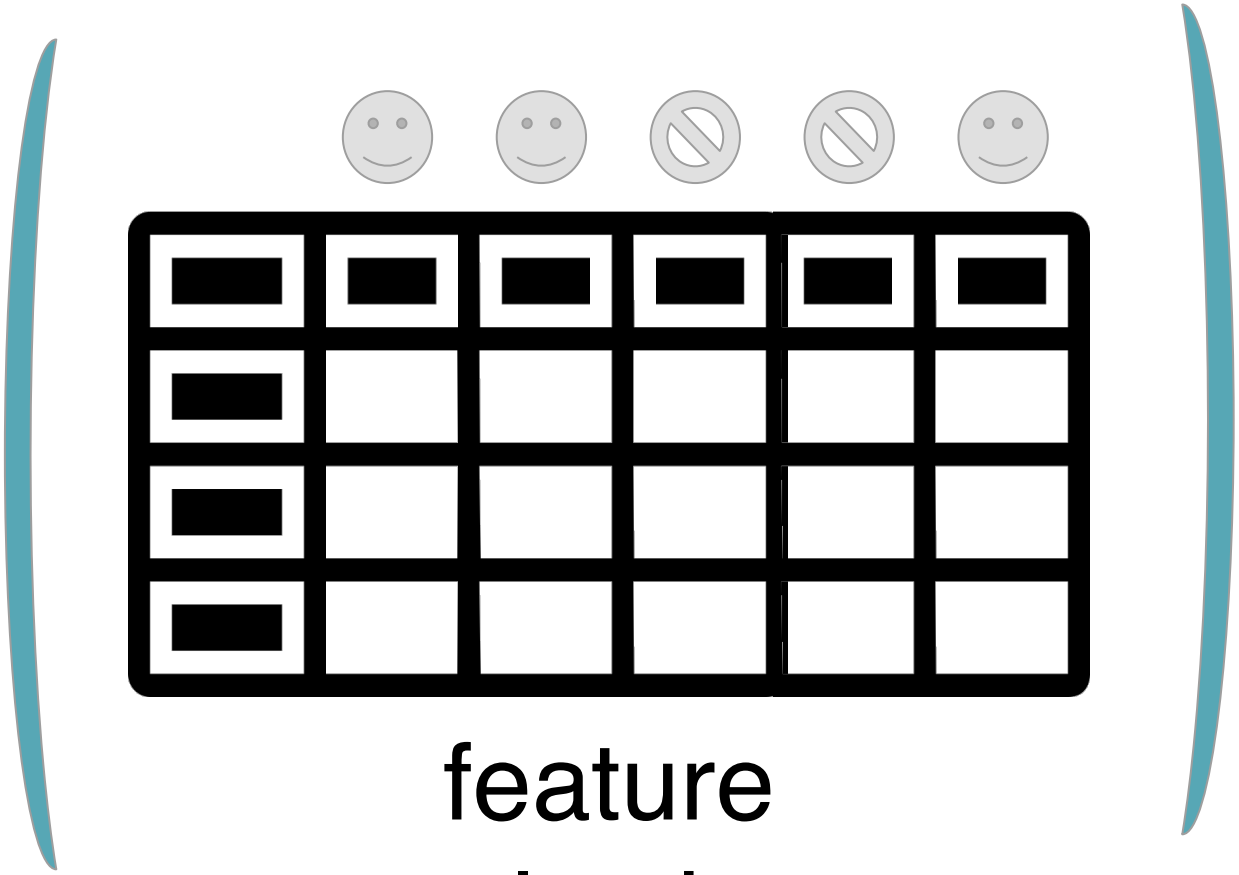
Smaller
10 - 25%

When you don’t have huge data

Cross validation

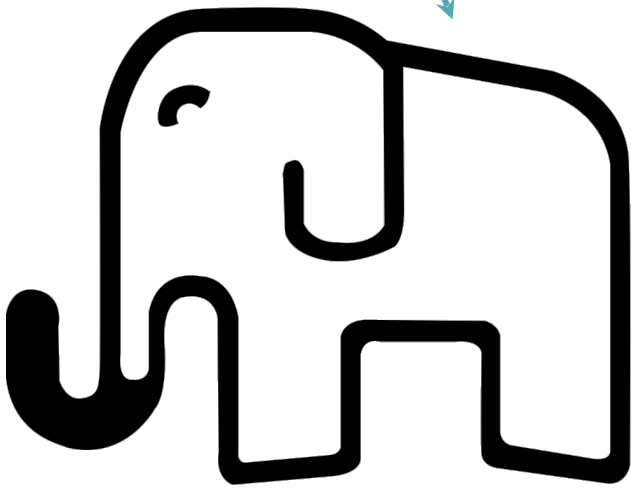


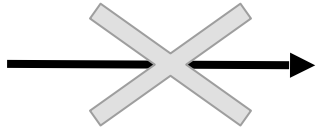
Use the mean of the hold out set performances to estimate either test or validation error

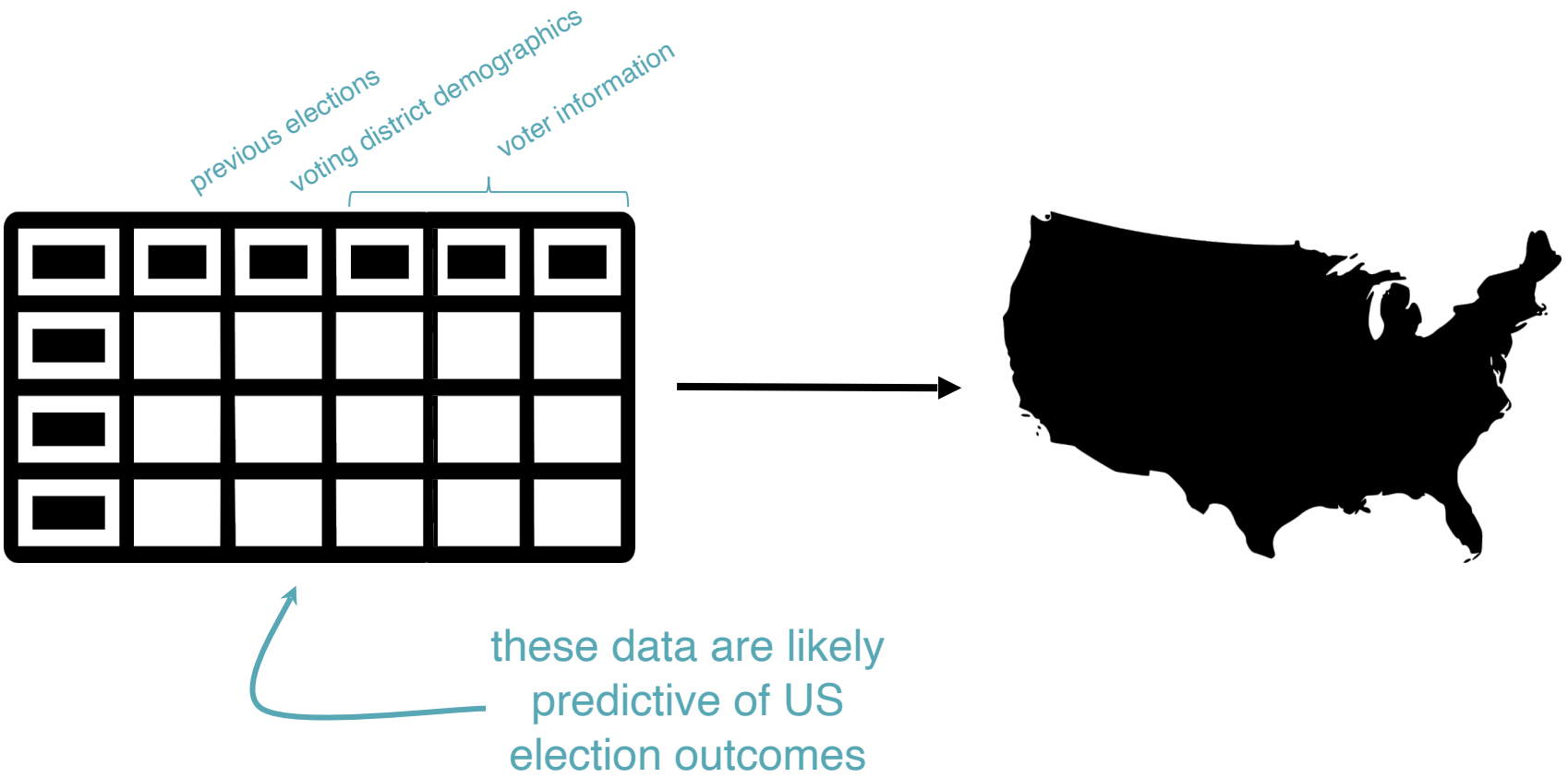


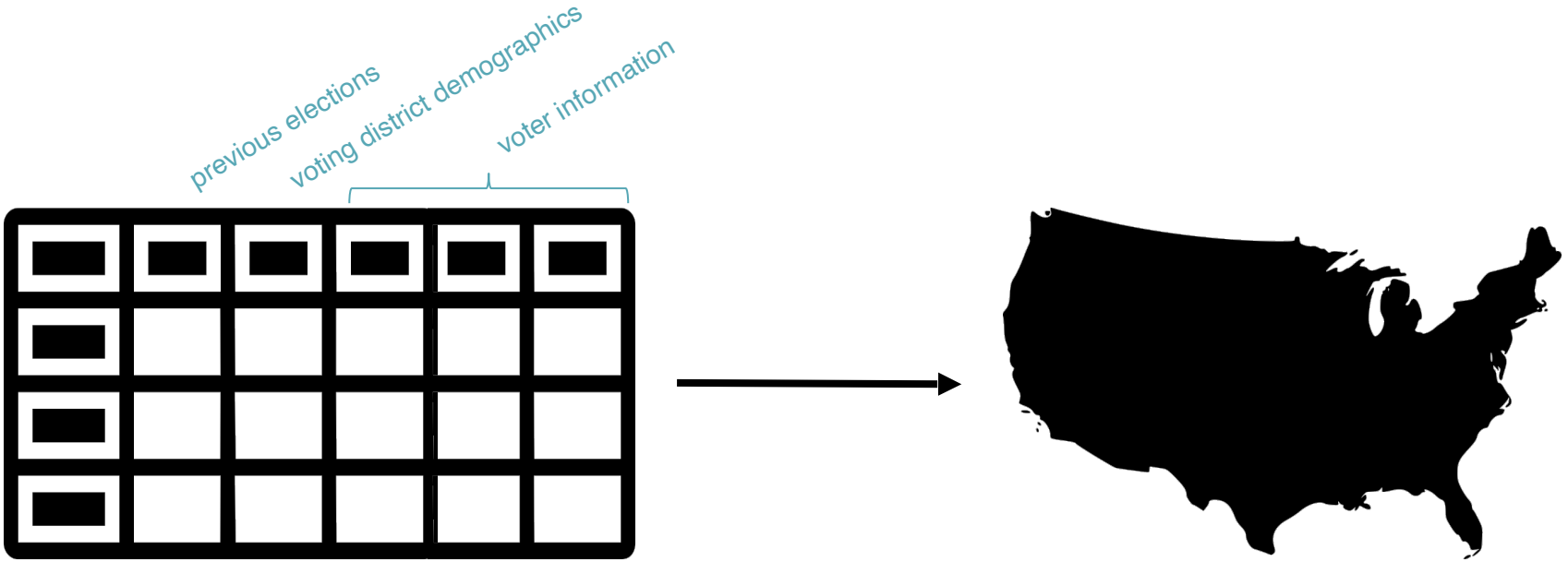
feature
selection

elephant height data
are likely not predictive
of US elections

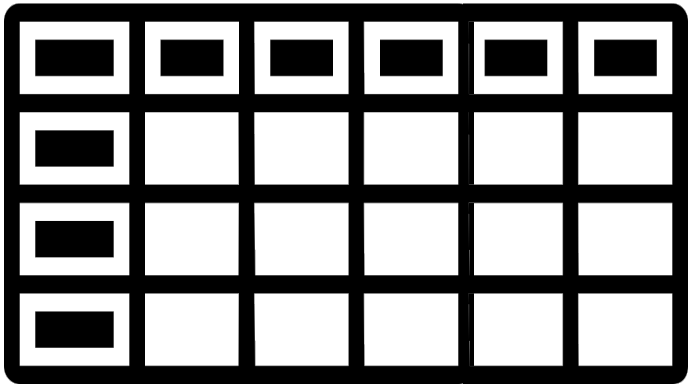





previous elections
voting district demographics
voter information
these data are likely predictive of US election outcomes

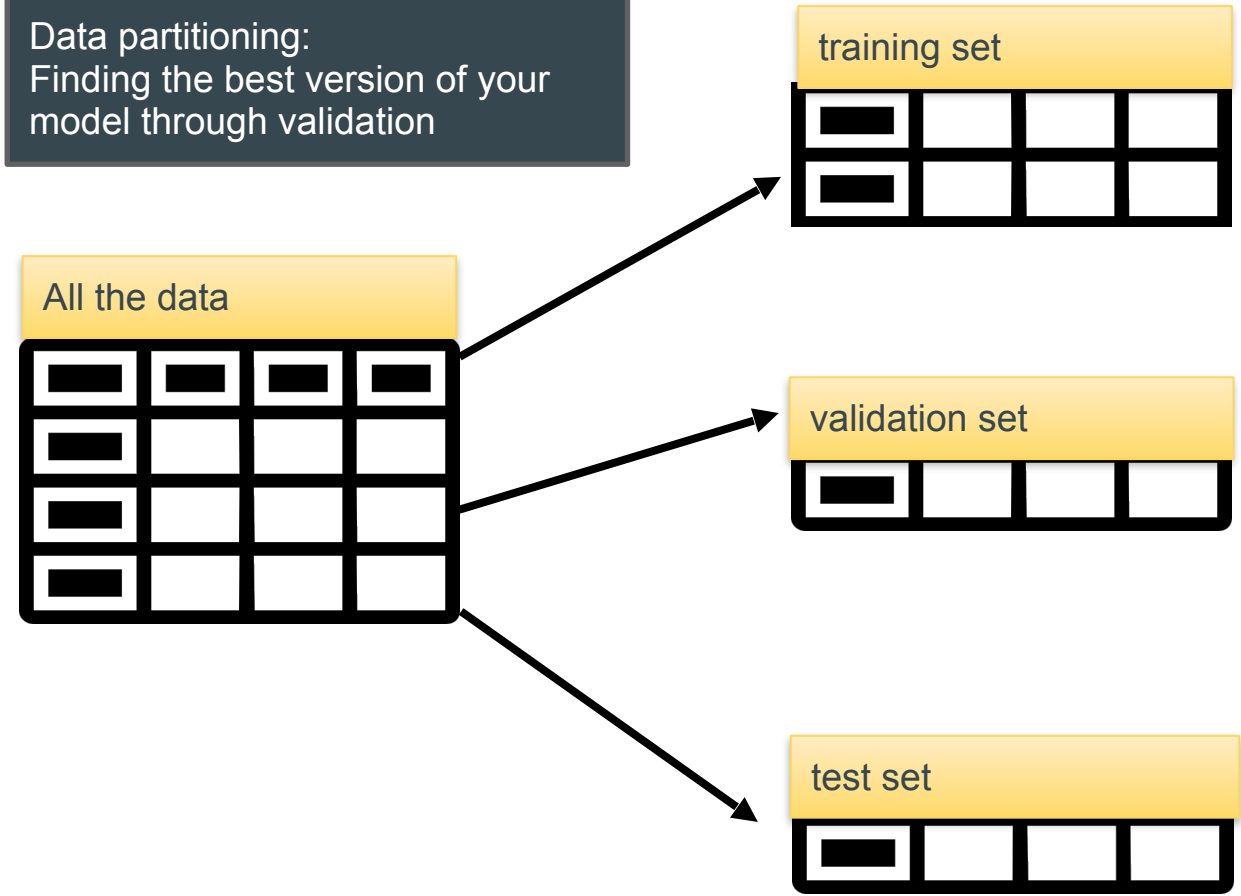


feature selection determines which variables are most predictive and includes them in the model



variables that can be used for accurate prediction exploit the relationship between the variables but do NOT mean that one causes the other

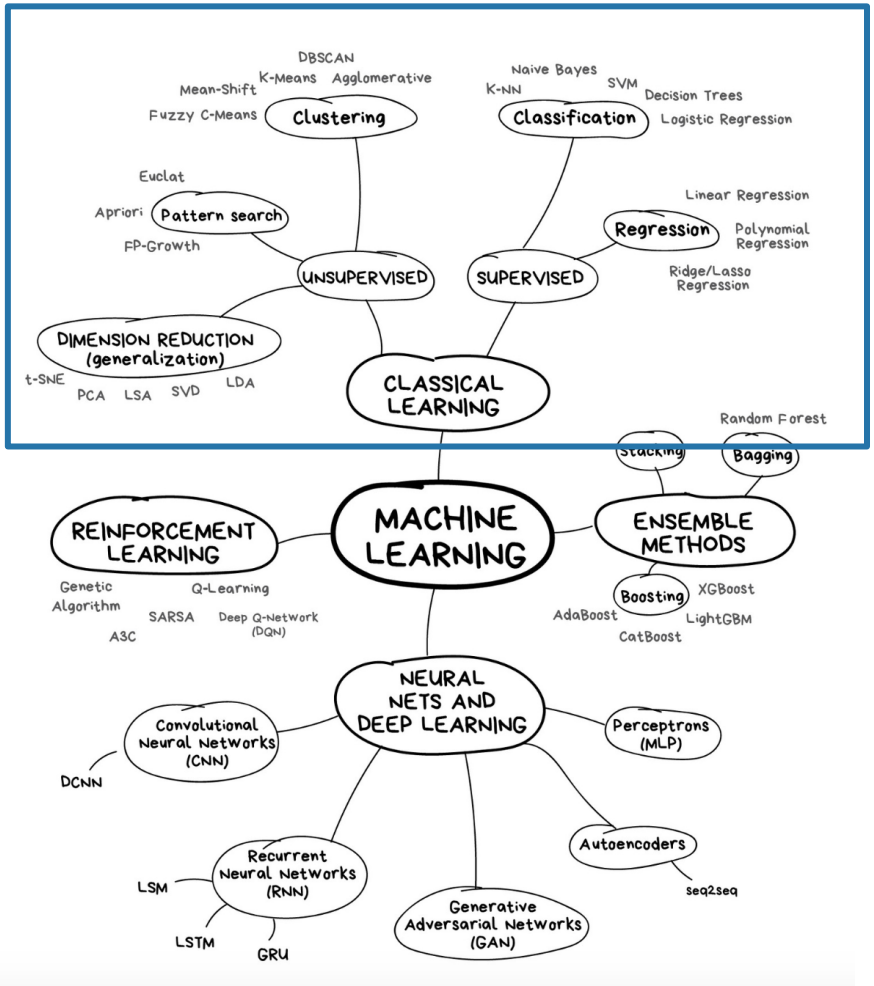
Data partitioning:
Finding the best version of your model through validation



Data used to build your model

Fine-tune the model to increase prediction accuracy (Hyper-parameter tuning, feature selection, model selection)

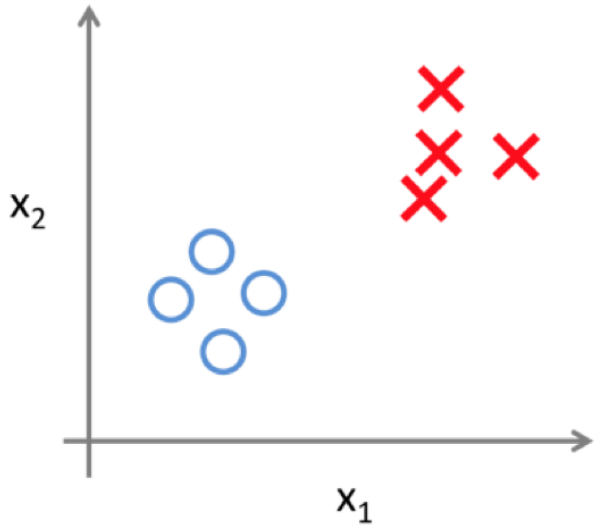
Used to assess how good the final model is (Generalization)



https://vas3k.com/blog/machine_learning/

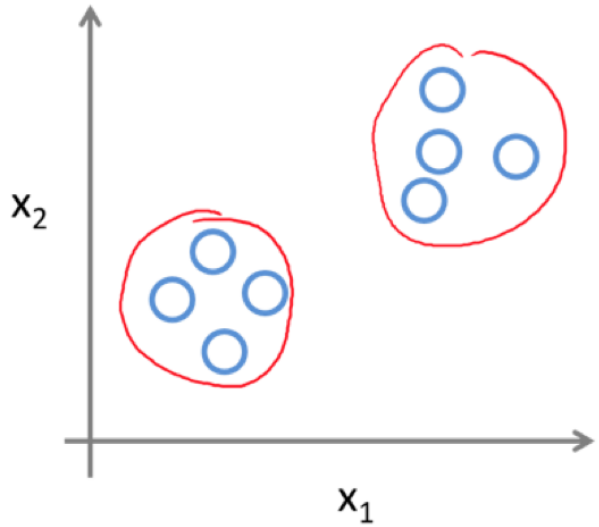
Two modes of machine learning

Supervised Learning



You give examples, the computer learns from them

Unsupervised Learning



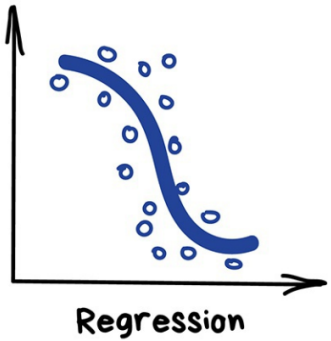
The computer determines how to classify based on properties within the data

Regression

"Draw a line through these dots. Yep, that's the machine learning"

Today this is used for:

- Stock price forecasts
- Demand and sales volume analysis
- Medical diagnosis
- Any number-time correlations



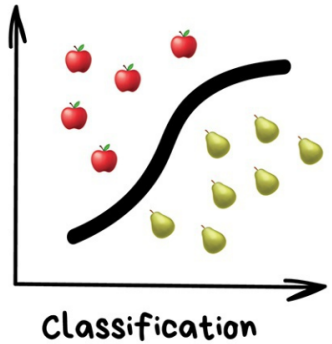
Popular algorithms are Linear and Polynomial regressions.

continuous variable prediction

Classification

"Splits objects based at one of the attributes known beforehand. Separate socks by based on color, documents based on language, music by genre"

Today used for:
– Spam filtering
– Language detection
– A search of similar documents
– Sentiment analysis
– Recognition of handwritten characters and numbers
– Fraud detection



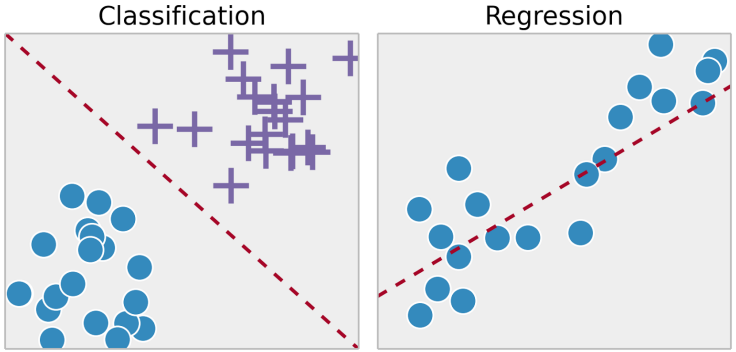
Popular algorithms: Naive Bayes, Decision Tree, Logistic Regression, K-Nearest Neighbours, Support Vector Machine

categorical variable prediction

https://vas3k.com/blog/machine_learning/

Approaches to machine learning

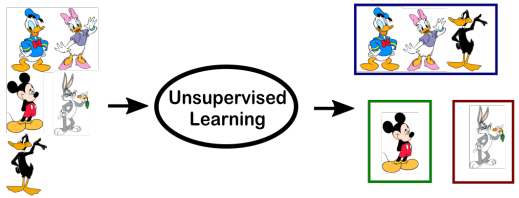
Supervised Learning



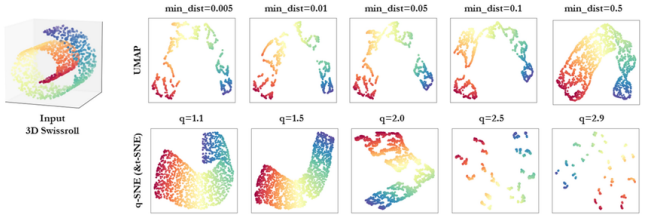
categorical variables

continuous variables

Unsupervised Learning



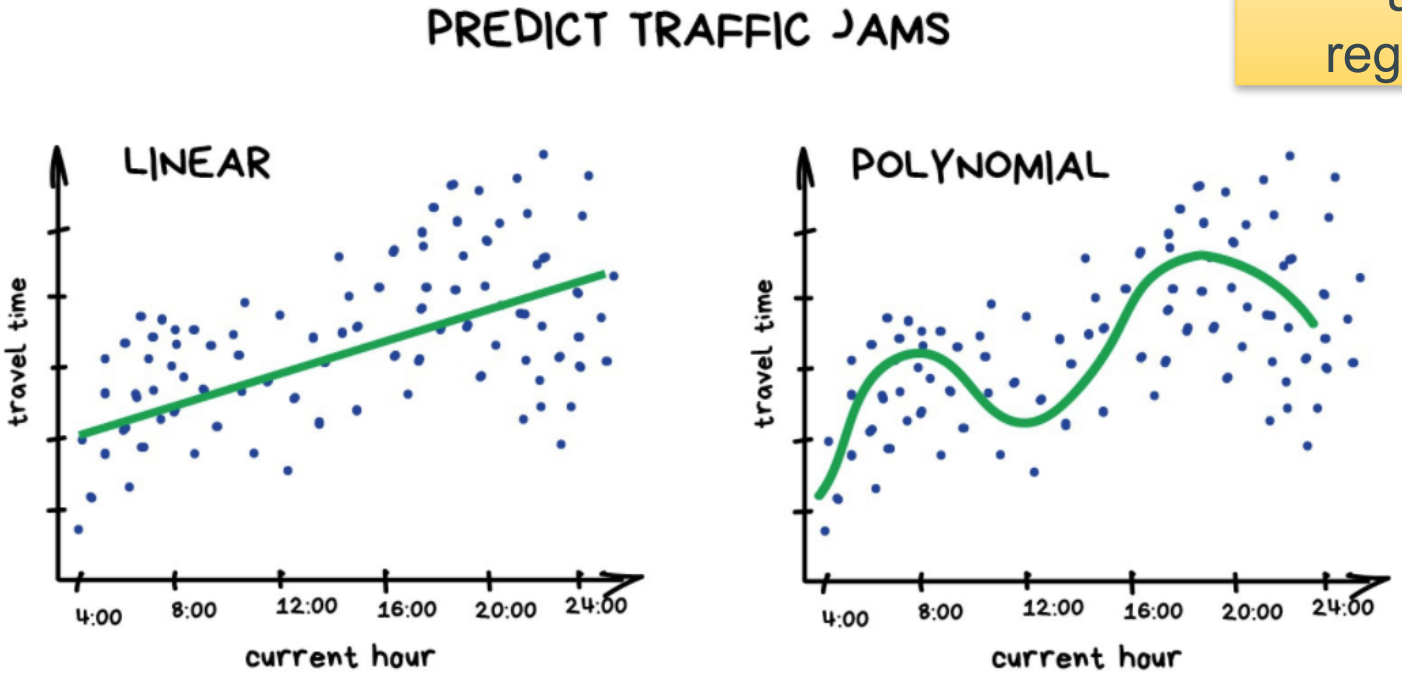
Clustering: categorical variables



Dim Red: continuous variables

Image source: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

Predicting using regression

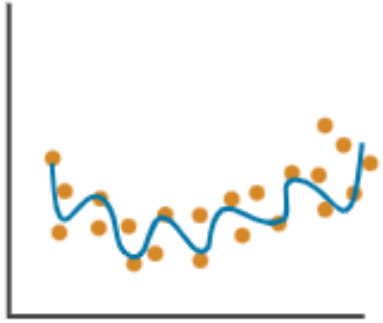


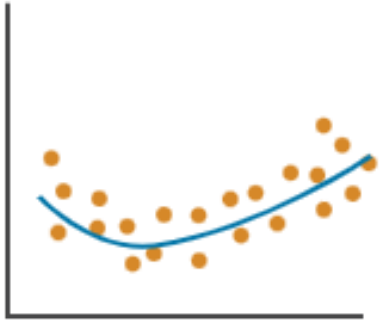
https://vas3k.com/blog/machine_learning/

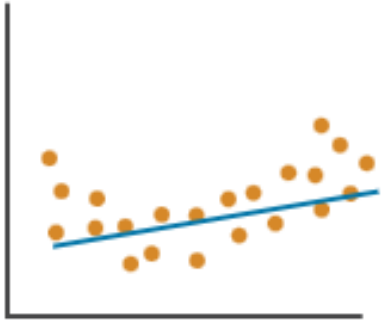
https://forms.gle/HBiDbxR2oRBYxcUh7

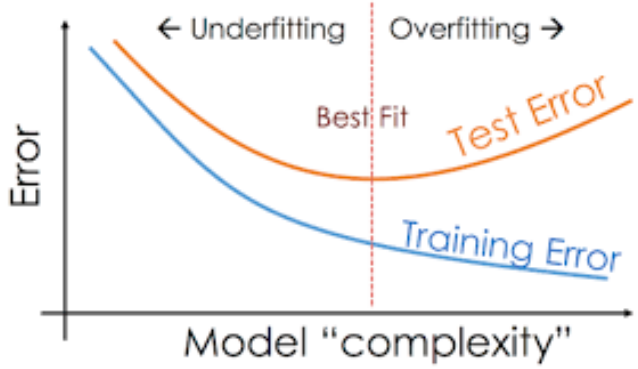
Overfit Best fit? Underfit

Regression

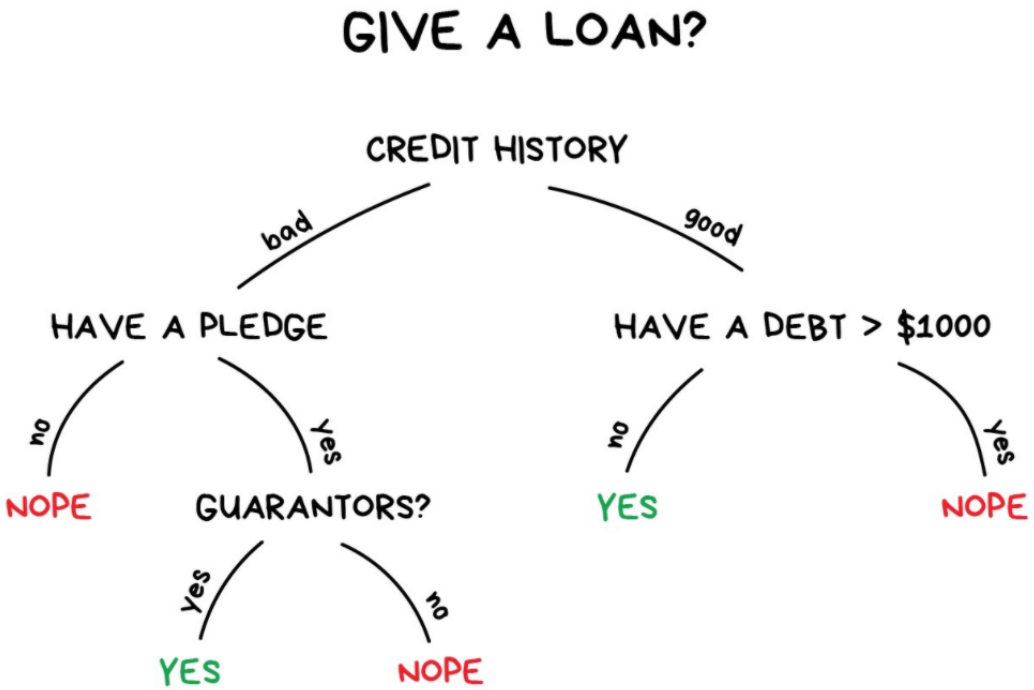








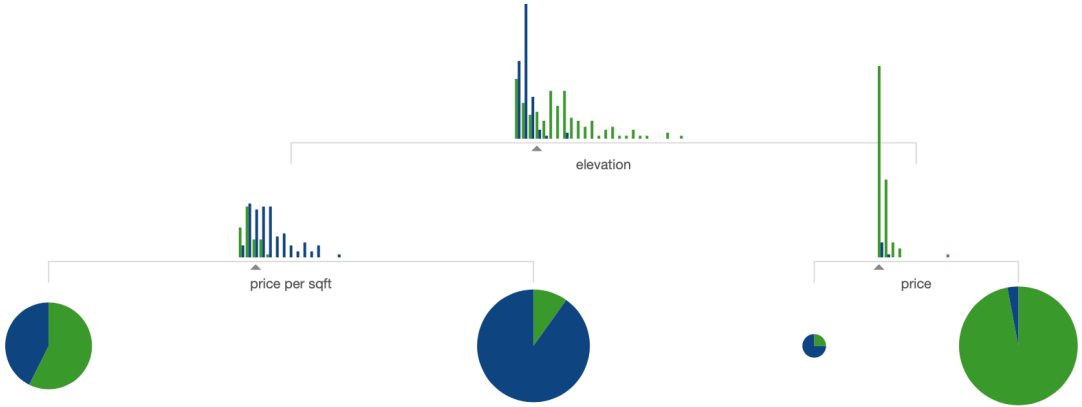
Classification using a decision tree



https://vas3k.com/blog/machine_learning/

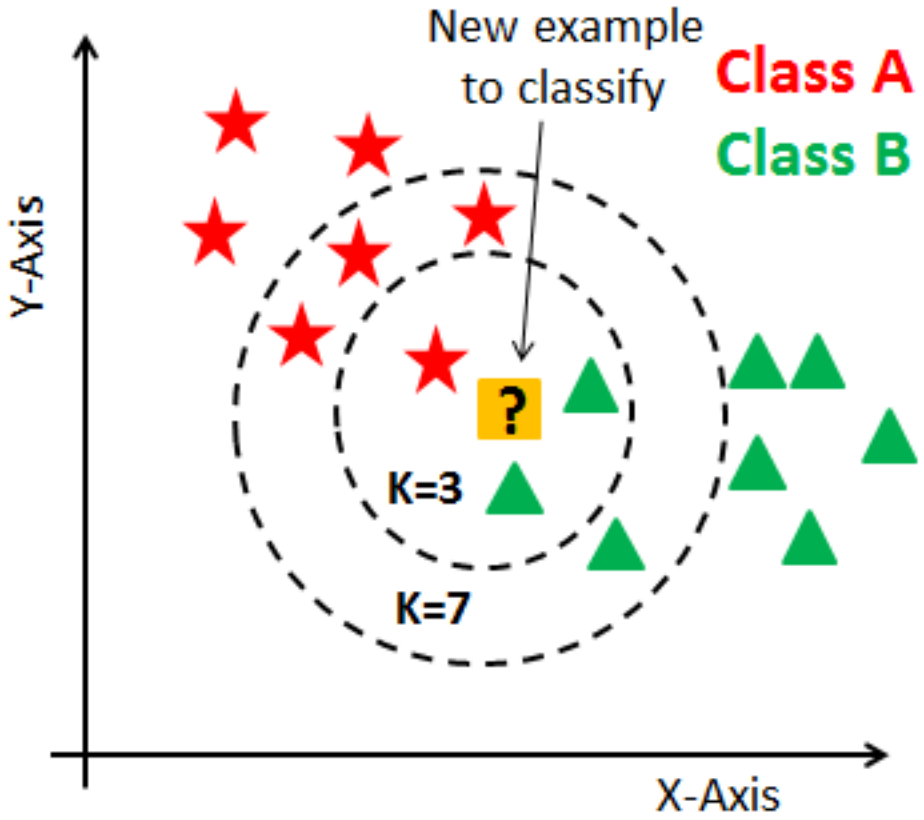
Growing a tree

Additional forks will add new information that can increase a tree's **prediction accuracy**.

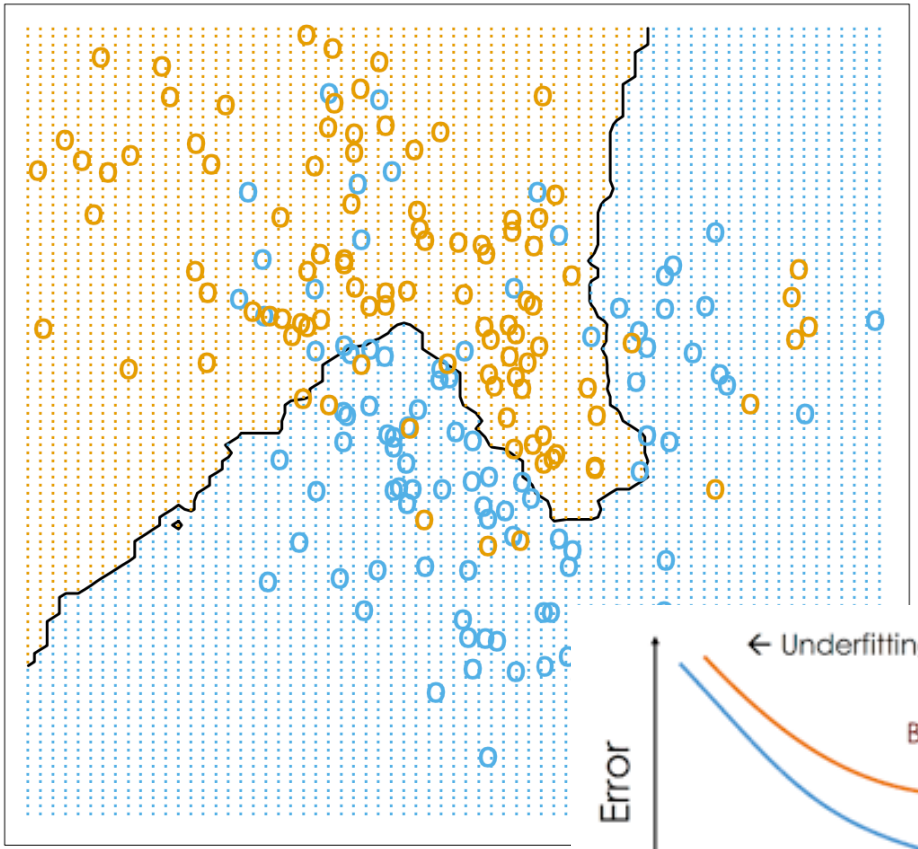


http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

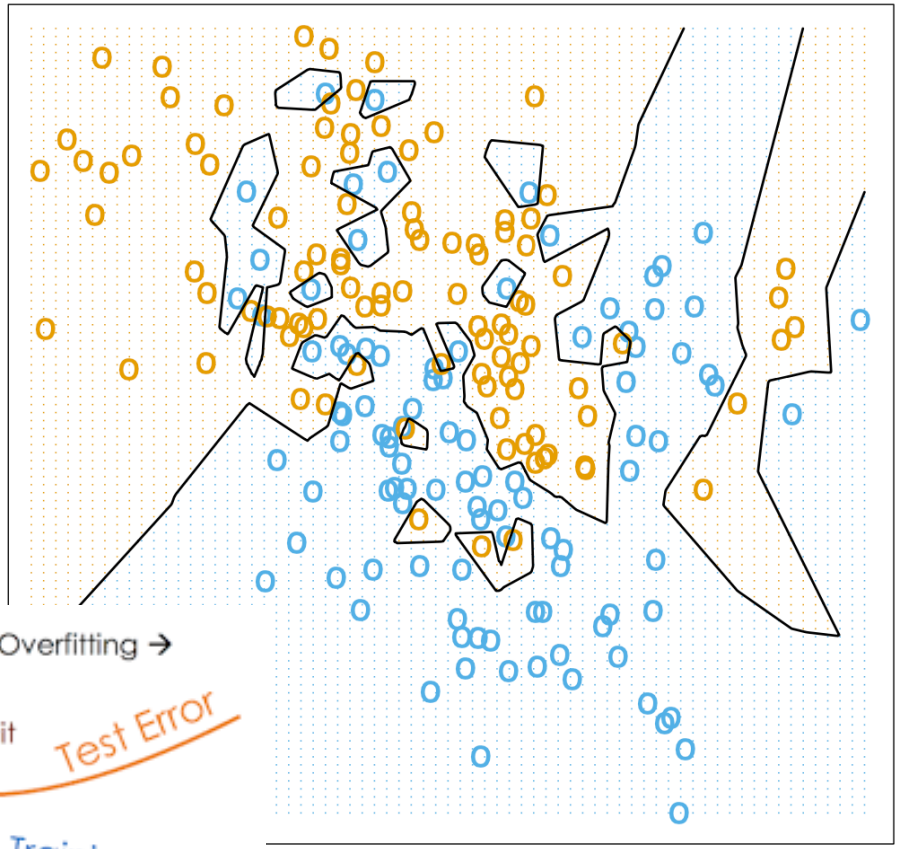
K-nearest neighbors

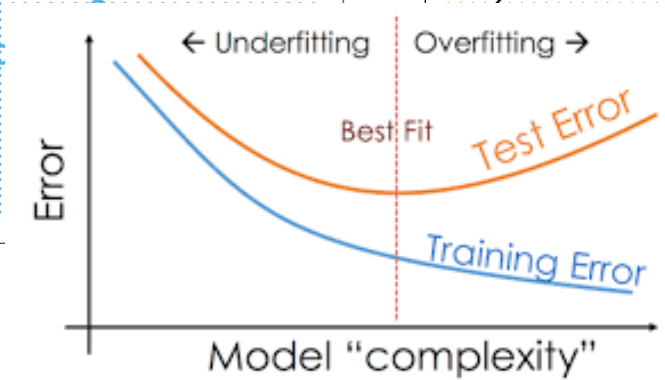


15-Nearest Neighbor Classifier



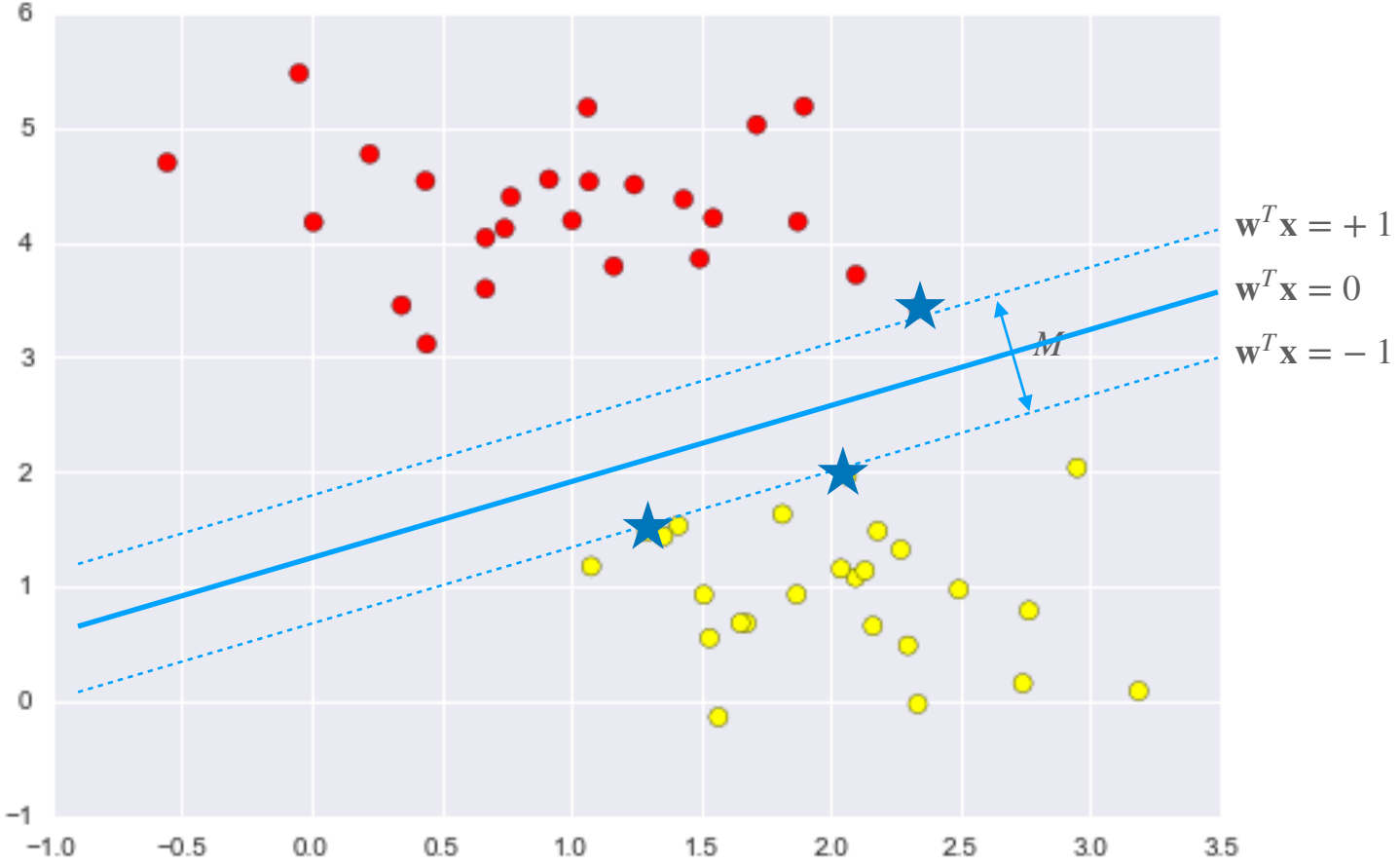
1–Nearest Neighbor Classifier



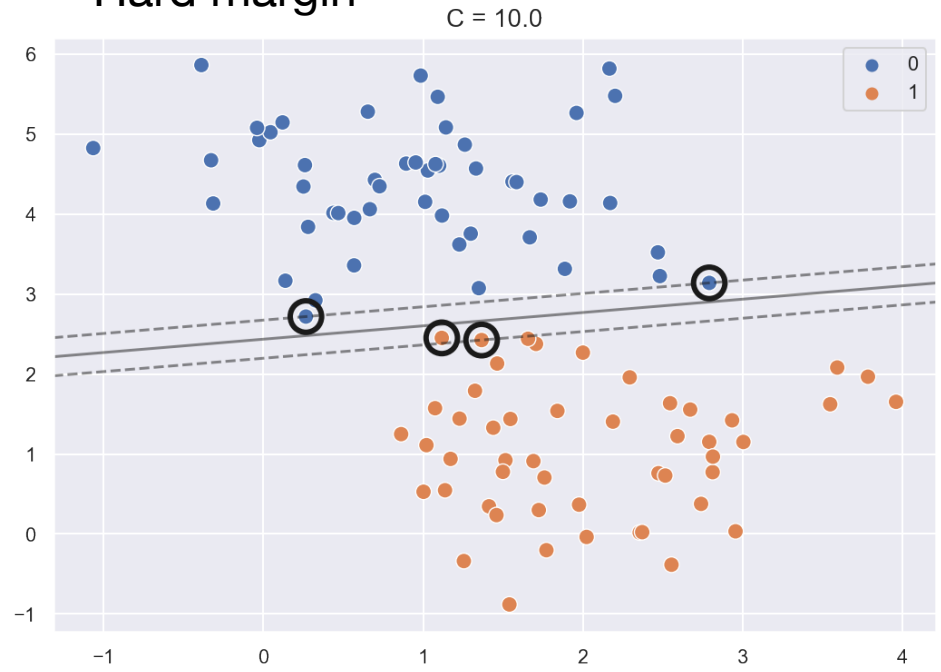


Support Vector Machines:

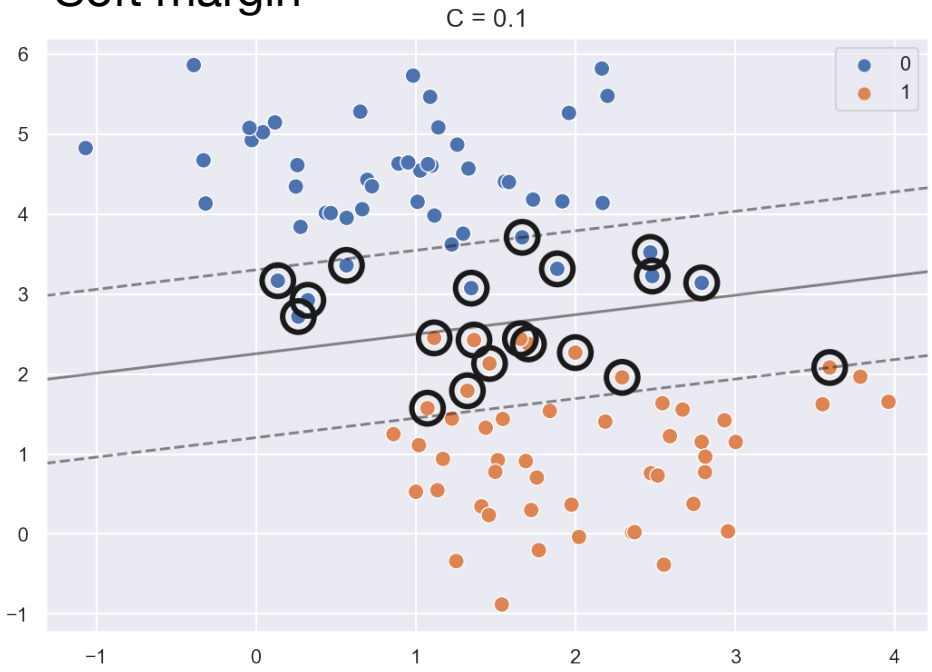
Find the support data points that maximize the margin M



Hard margin

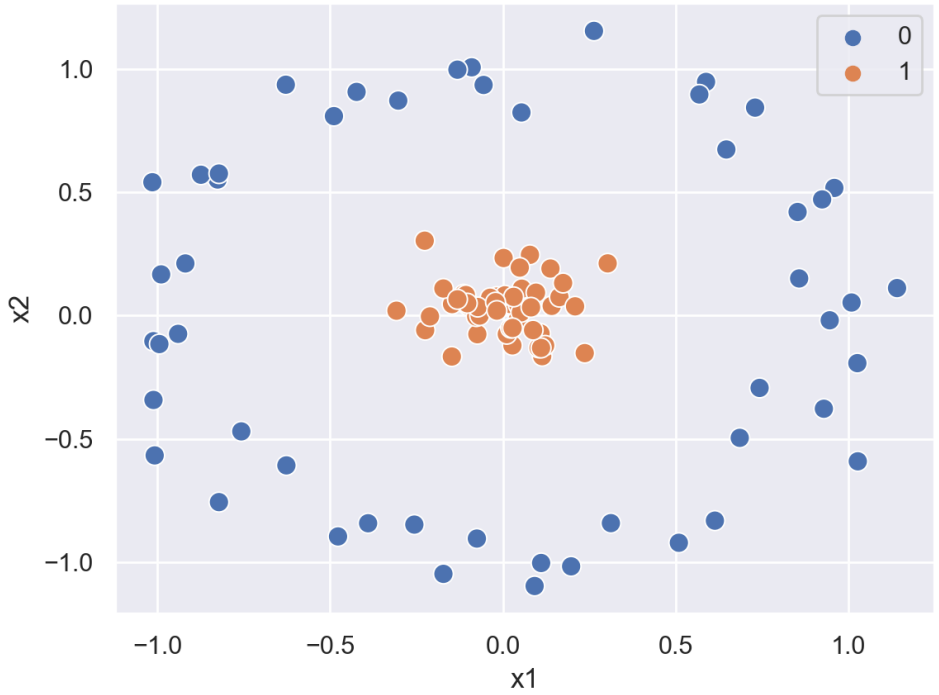


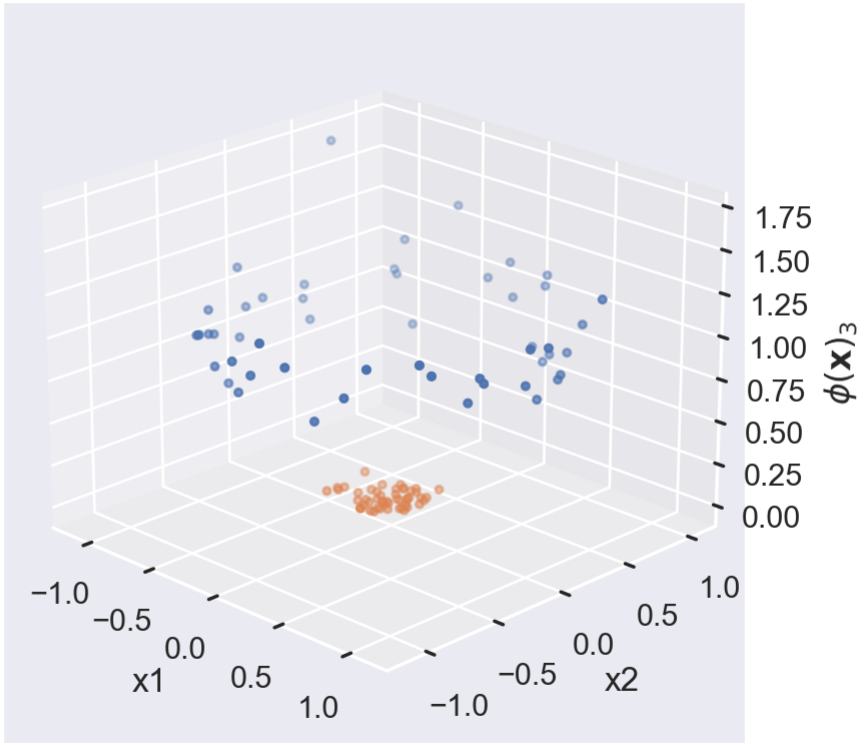
Soft margin



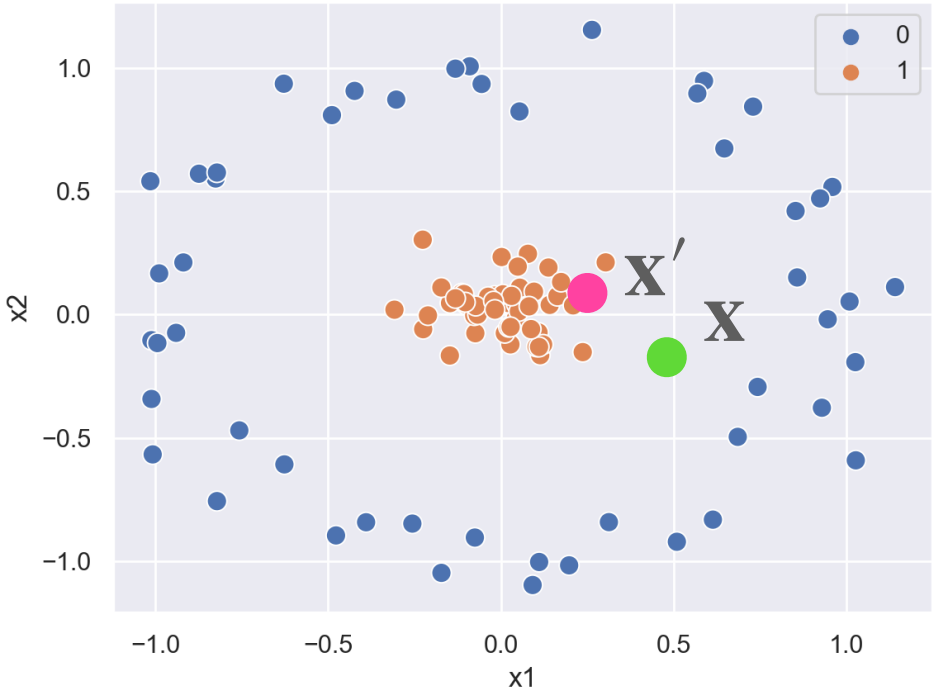
Find: $\arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C \times \sum_{i=1}^{n}(1 - y_i \times (\mathbf{w}^T\mathbf{x}_i + b))_+$

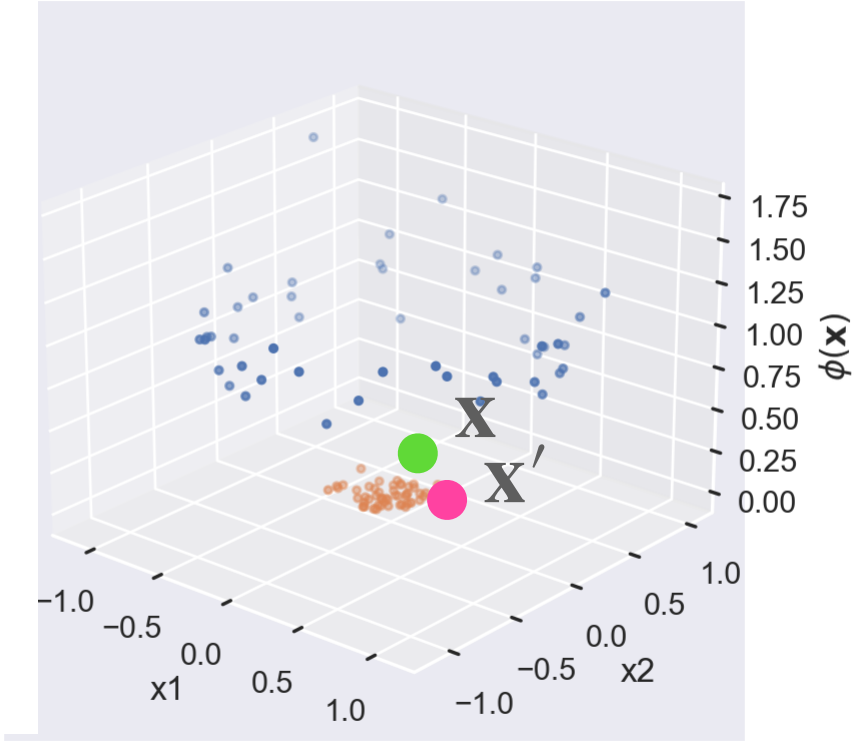
$$\phi(\mathbf{x}) = \left[x_1,\ x_2,\ x_1 \cdot x_2 + x_1^2 \cdot x_2^2\right]$$





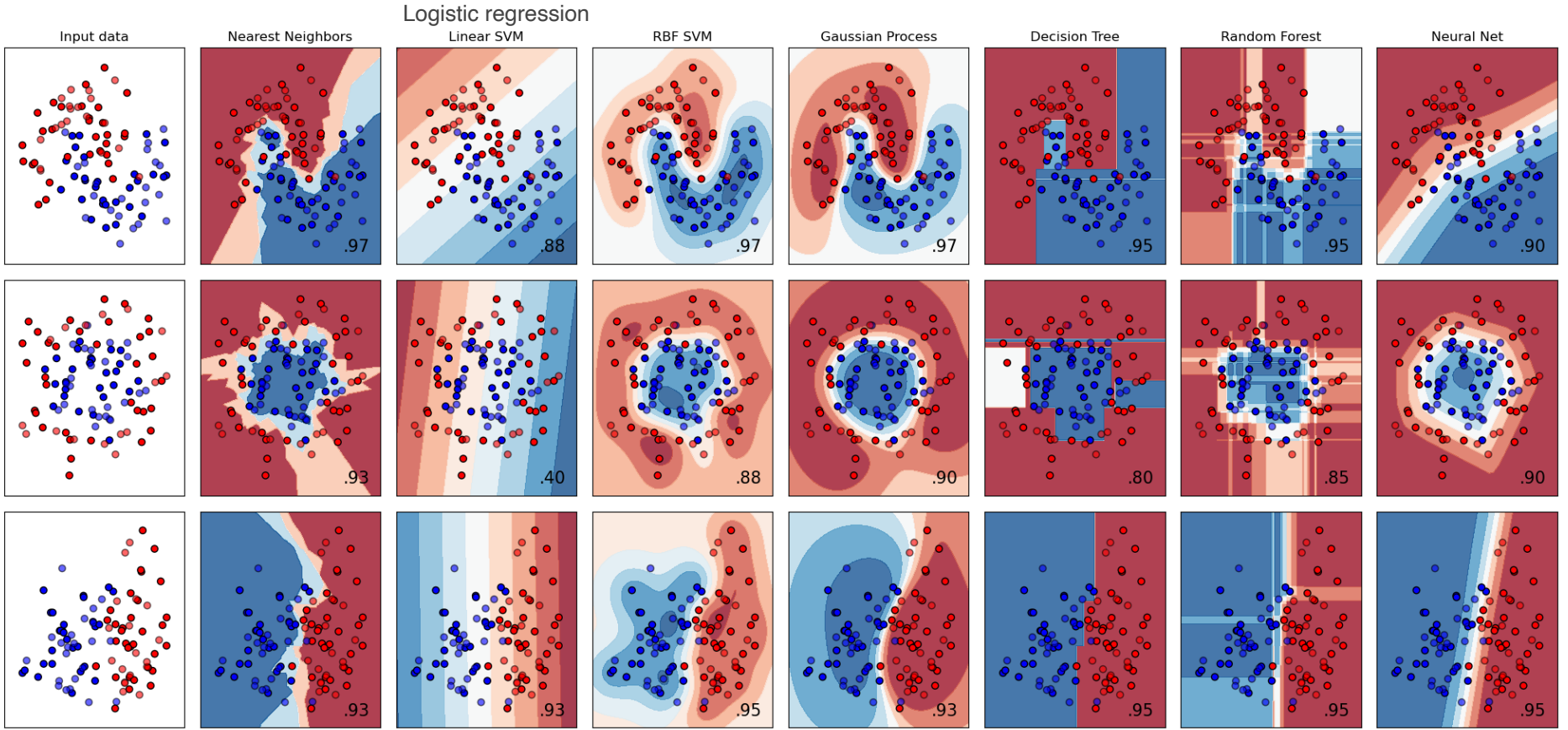
More generally this is the polynomial kernel


0
1
x2
x1
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
x′
x


x
x′
x1
x2
ϕ(x)
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−1.0
−0.5
0.0
0.5
1.0

Linear vs Non-linear classifiers

- LINEAR - Lower complexity, less overfitting, less power
    - Logistic regression
    - “Linear kernel” SVM
- NON-LINEAR - Higher complexity, more overfitting, more power
    - K-nearest Neighbors
    - “RBF kernel” SVM

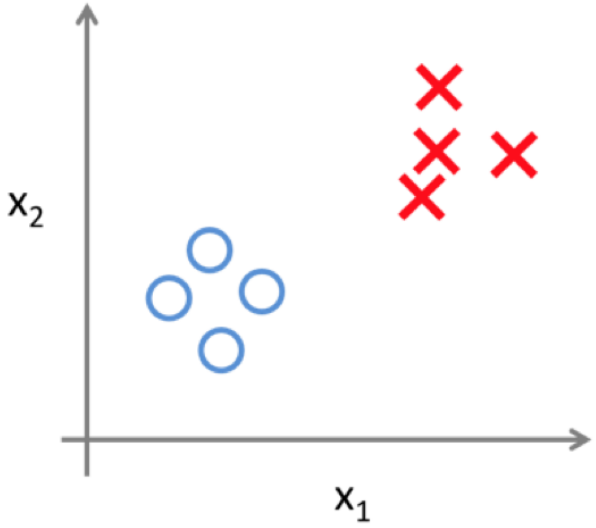


https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

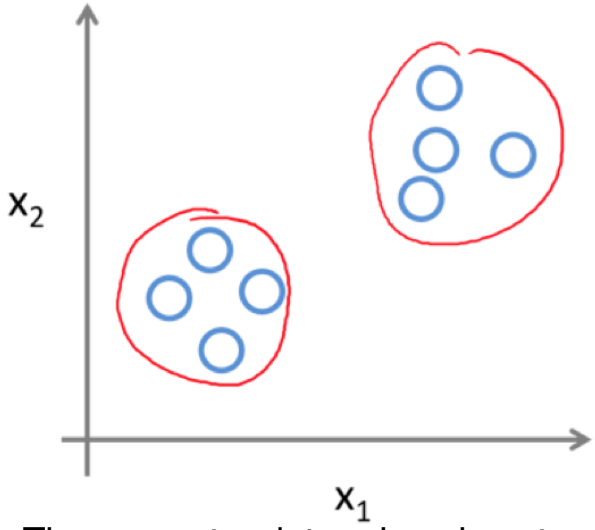
Unsupervised Learning

To modes of machine learning

Supervised Learning



Unsupervised Learning



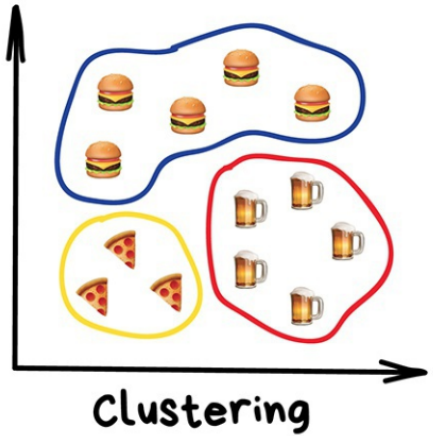
The computer determines how to classify based on properties within the data

Clustering

"Divides objects based on unknown features. Machine chooses the best way"

Nowadays used:

- For market segmentation (types of customers, loyalty)
- To merge close points on a map
- For image compression
- To analyze and label new data
- To detect abnormal behavior



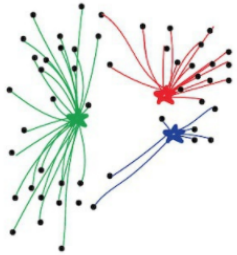
Popular algorithms: K-means_clustering, Mean-Shift, DBSCAN

Image source: https://vas3k.com/blog/machine_learning/

PUT KEBAB KIOSKS IN THE OPTIMAL WAY

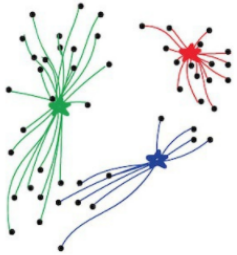
(also illustrating the K-means method)

1. Put kebab kiosks in random places in city



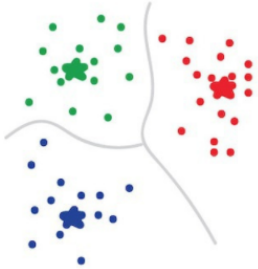
2. Watch how buyers choose the nearest one

3. Move kiosks closer to the centers of their popularity



4. Watch and move again

5. Repeat a million times



6. Done!
You're god of kebabs!

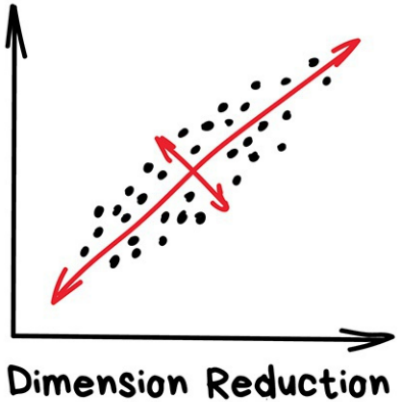
Image source: https://vas3k.com/blog/machine_learning/

Dimensionality Reduction (Generalization)

"Assembles specific features into more high-level ones"

Nowadays is used for:

- Recommender systems (★)
- Beautiful visualizations
- Topic modeling and similar document search
- Fake image analysis
- Risk management



Popular algorithms: Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Latent Dirichlet allocation (LDA), Latent Semantic Analysis (LSA, pLSA, GLSA), t-SNE (for visualization)

Image source: https://vas3k.com/blog/machine_learning/

Modern dimensionality reduction algos
have hyper parameters that can make you think
patterns are there when they really aren’t (and vv.)

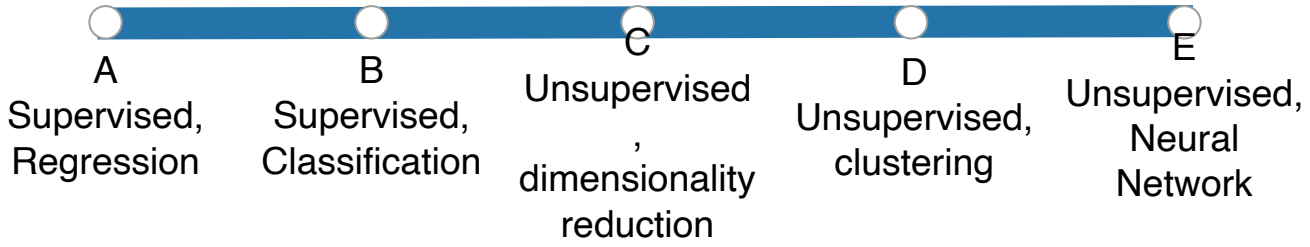
https://distill.pub/2016/misread-tsne/

https://pair-code.github.io/understanding-umap/

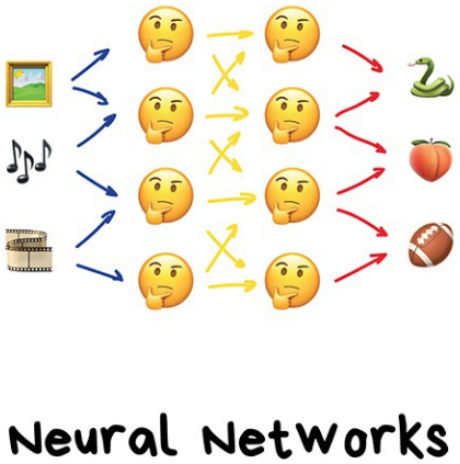
Prediction Approach

You want to predict someone's emotion based on an image.

How would you approach this with machine learning?



"We have a thousand-layer network, dozens of video cards, but still no idea where to use it. Let's generate cat pics!"



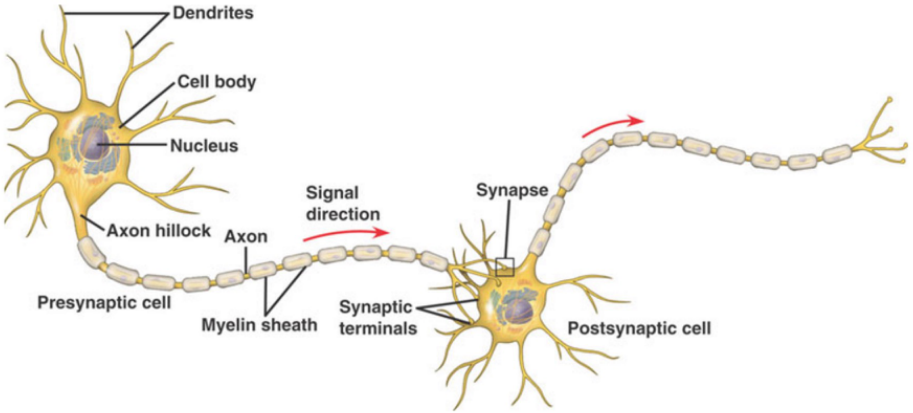
Used today for:

- Replacement of all algorithms above
- Object identification on photos and videos
- Speech recognition and synthesis
- Image processing, style transfer
- Machine translation

Popular architectures: Perceptron, Convolutional Network (CNN), Recurrent Networks (RNN), Autoencoders

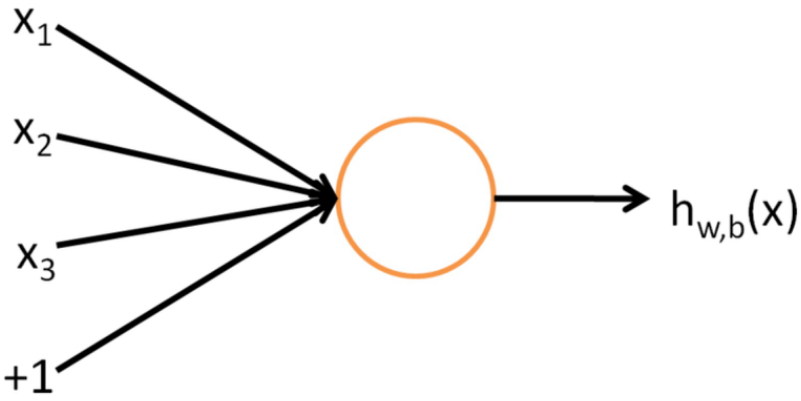
https://vas3k.com/blog/machine_learning/

WHAT IS A NEURON?



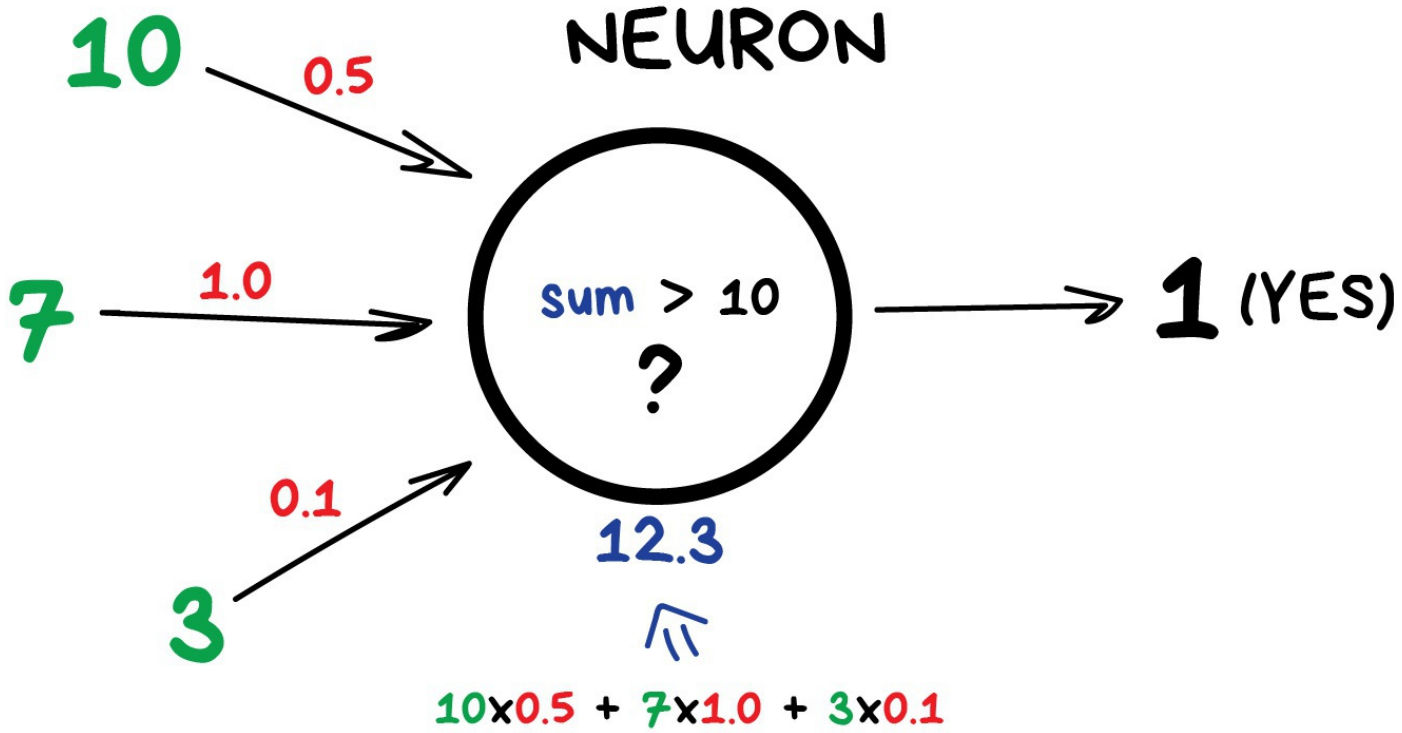
- Receives signal on synapse
- When trigger sends signal on axon

MATHEMATICAL NEURON



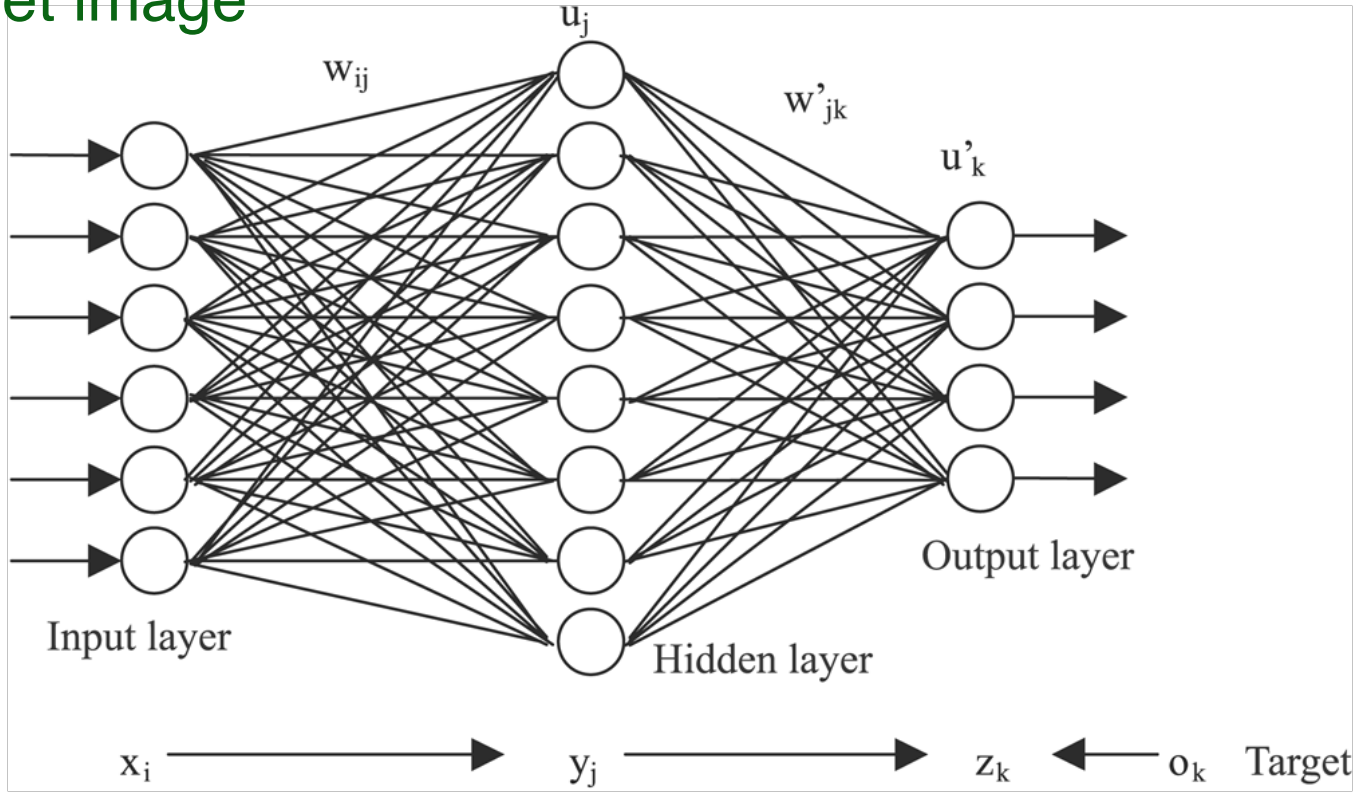
- Mathematical abstraction, inspired by biological neuron
- Either on or off based on sum of input

These weights tell the neuron to respond more to one input and less to another. Weights are adjusted when training — that's how the network learns. Basically, that's all there is to it.

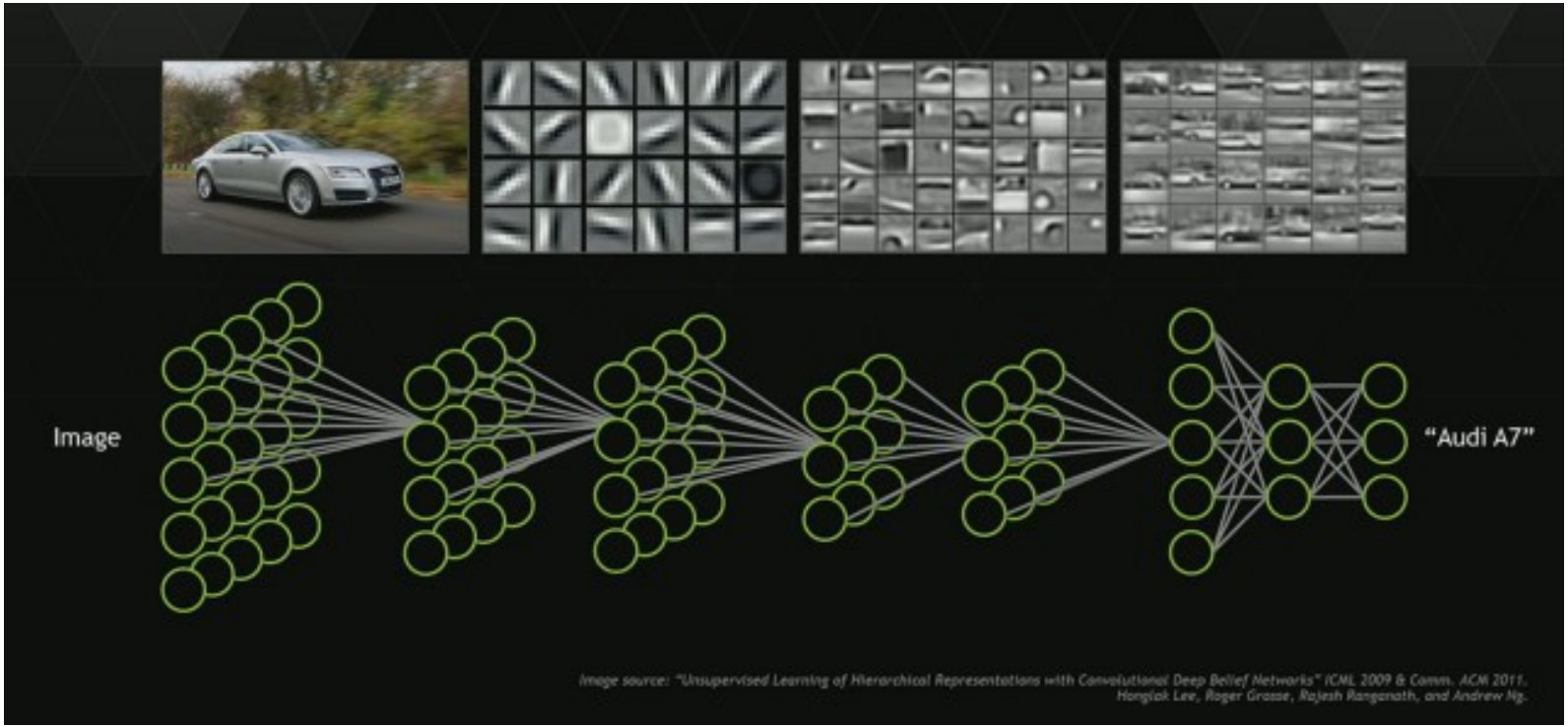


https://vas3k.com/blog/machine_learning/

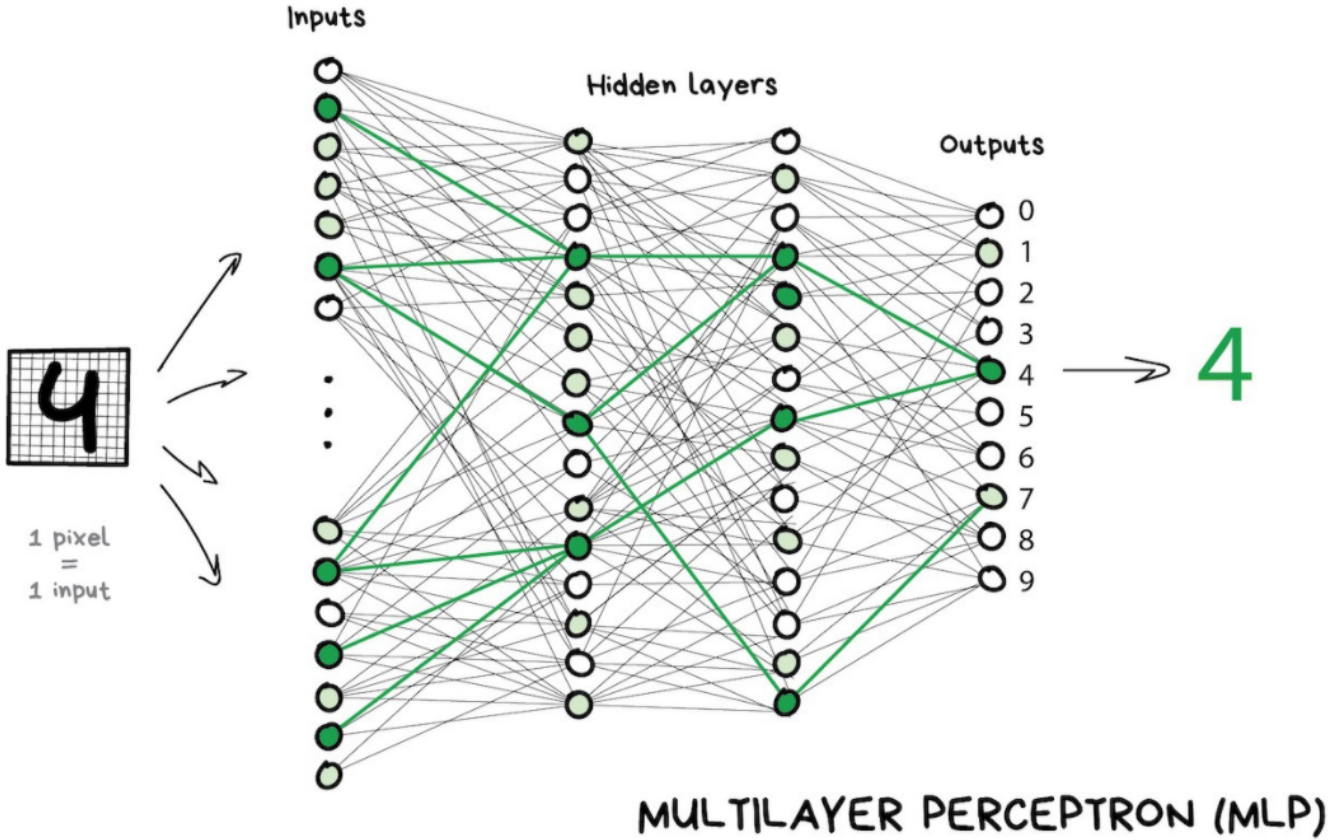
This will likely not be the last time you see this (mostly unhelpful) neural net image



Convolutional neural networks



https://towardsdatascience.com/understanding-residual-networks-9add4b664b03



https://vas3k.com/blog/machine_learning/

Manually labeling used to be the way...

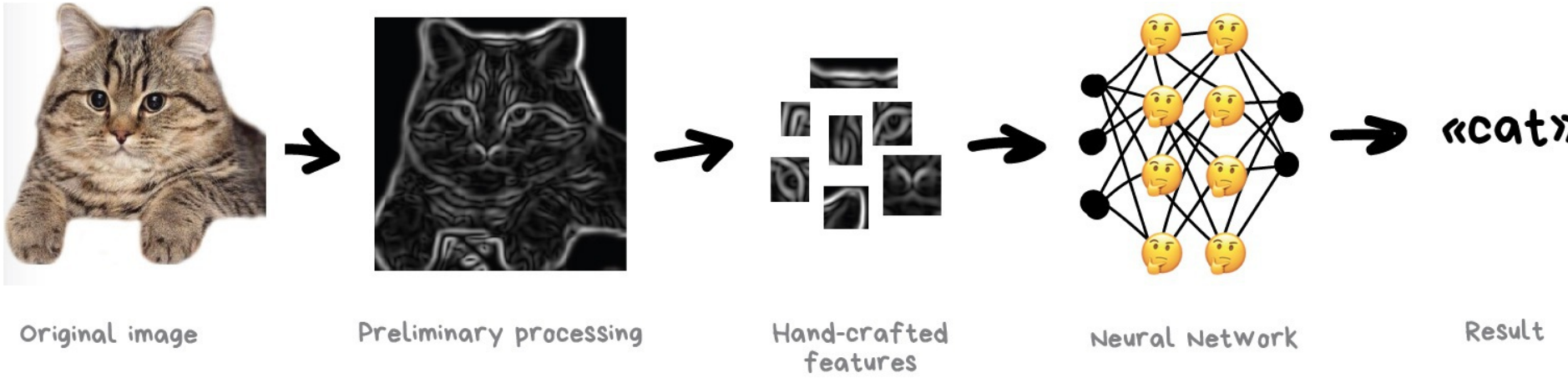
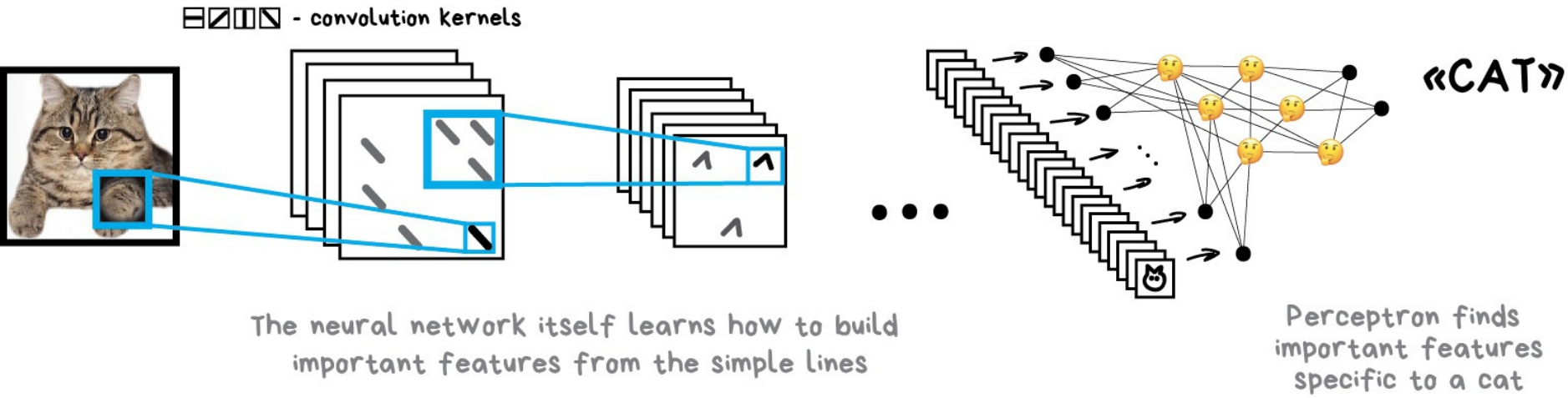


Image source: https://vas3k.com/blog/machine_learning/

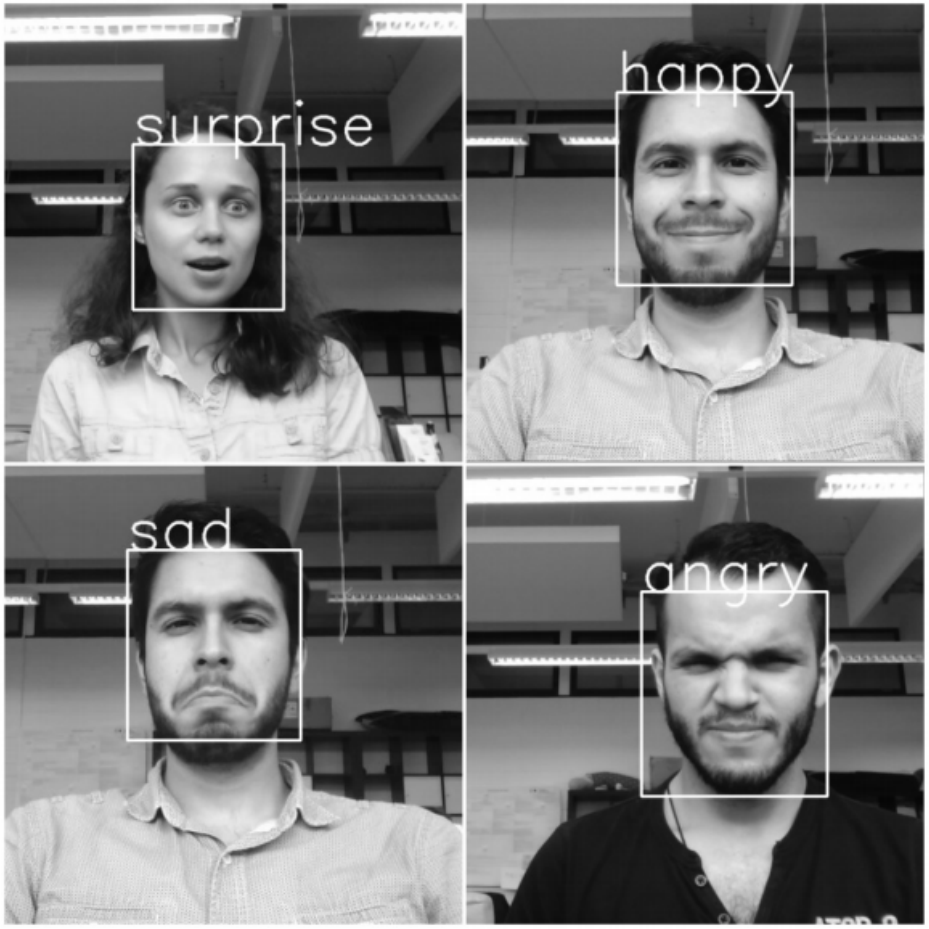
CNNs avoid manual features



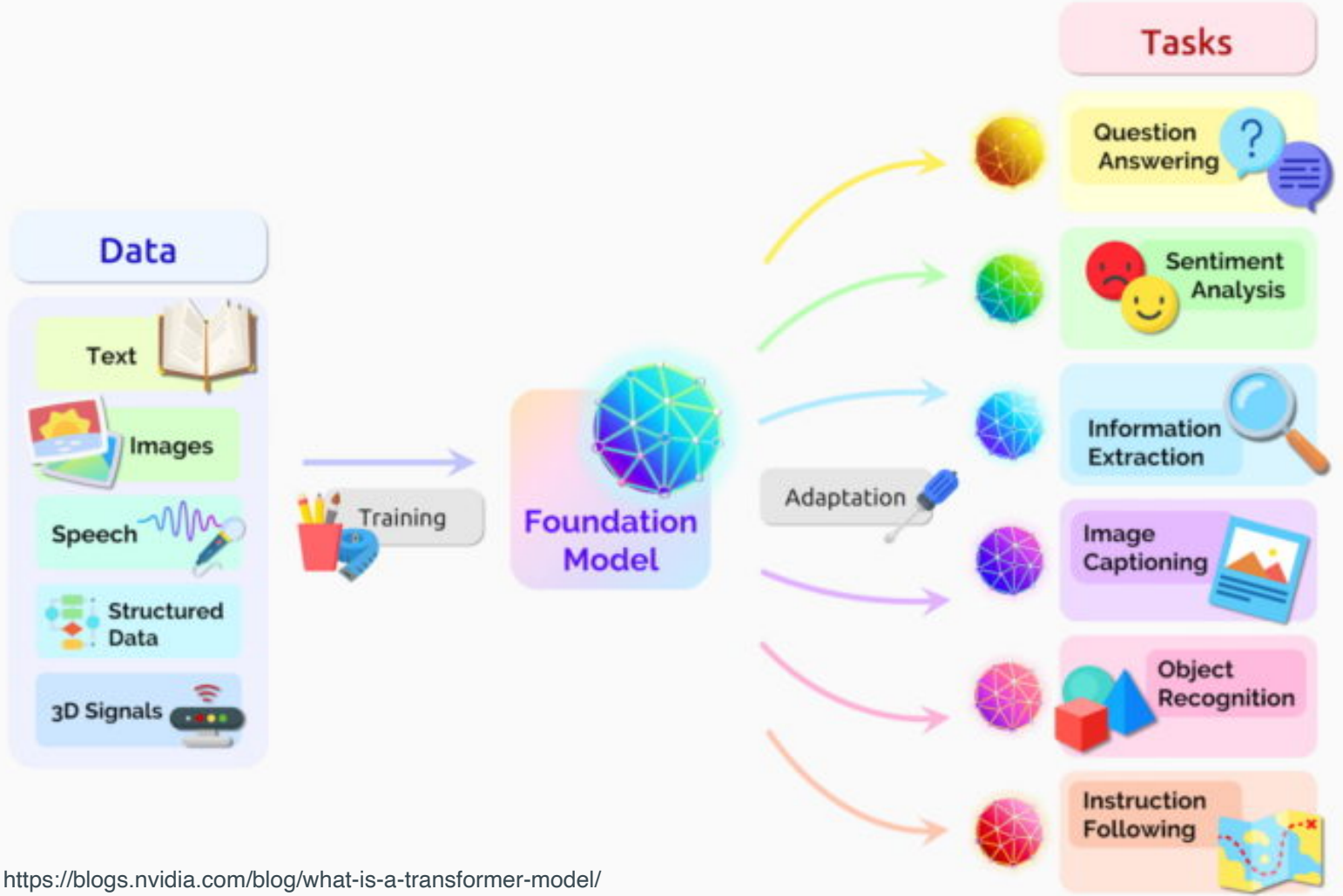
“CNNs are all the rage right now. They are used to search for objects on photos and in videos, face recognition, style transfer, generating and enhancing images, creating effects like slow-mo and improving image quality. Nowadays CNNs are used in all the cases that involve pictures and videos.”

CONVOLUTIONAL NEURAL NETWORK (CNN)

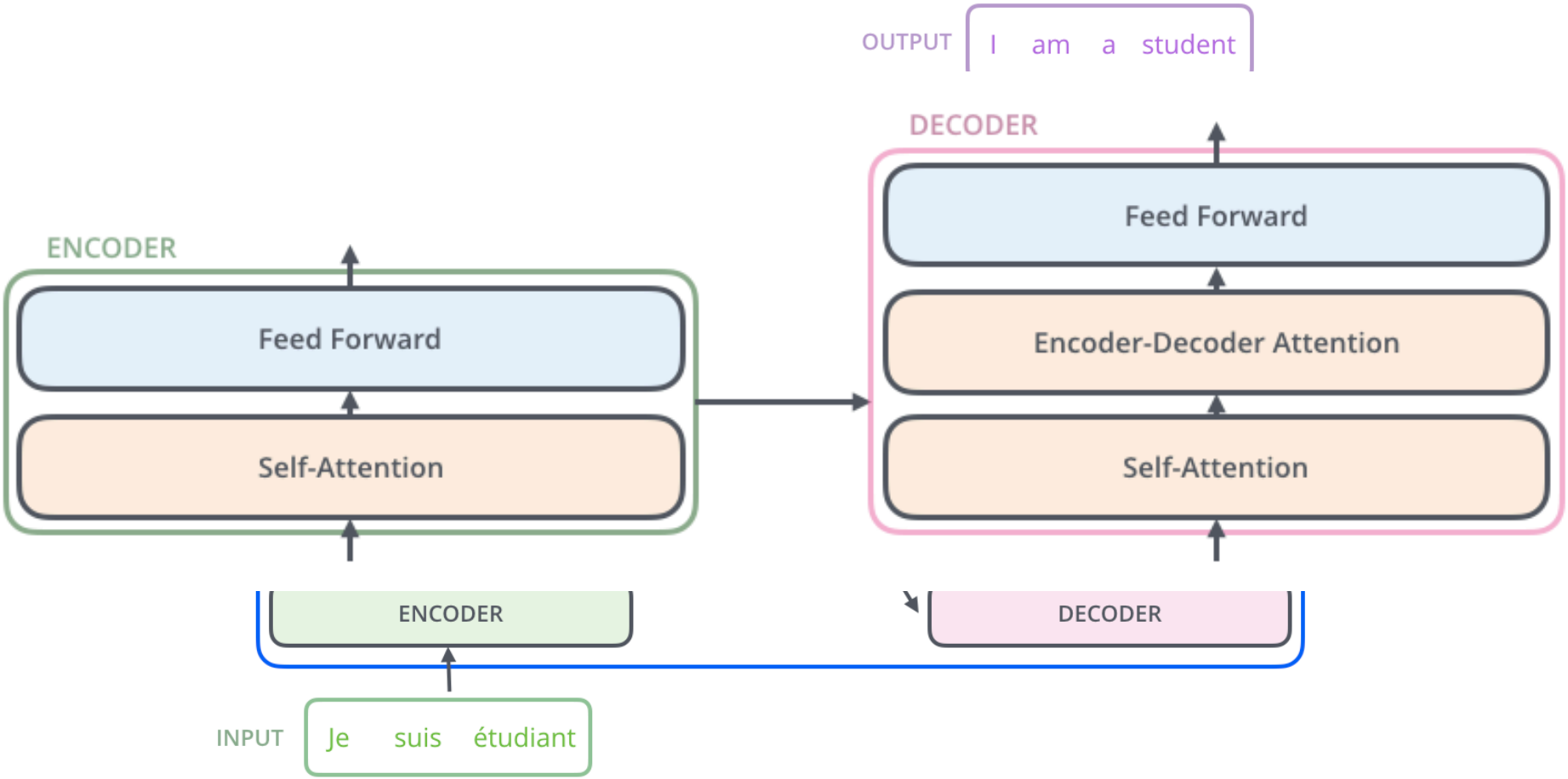
Image source: https://vas3k.com/blog/machine_learning/



https://github.com/oarriaga/face_classification



https://blogs.nvidia.com/blog/what-is-a-transformer-model/

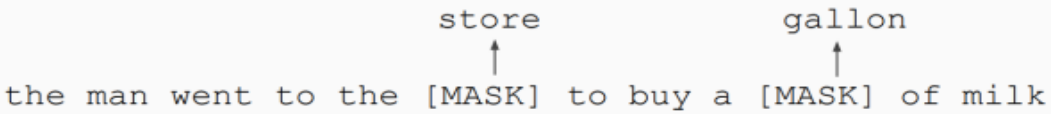


https://jalammar.github.io/illustrated-transformer/

Much of DL success comes from semi-supervised tricks to avoid large hand labelled datasets

Masked LM

- **Solution**: Mask out *k*% of the input words, and then predict the masked words
  - We always use *k* = 15%



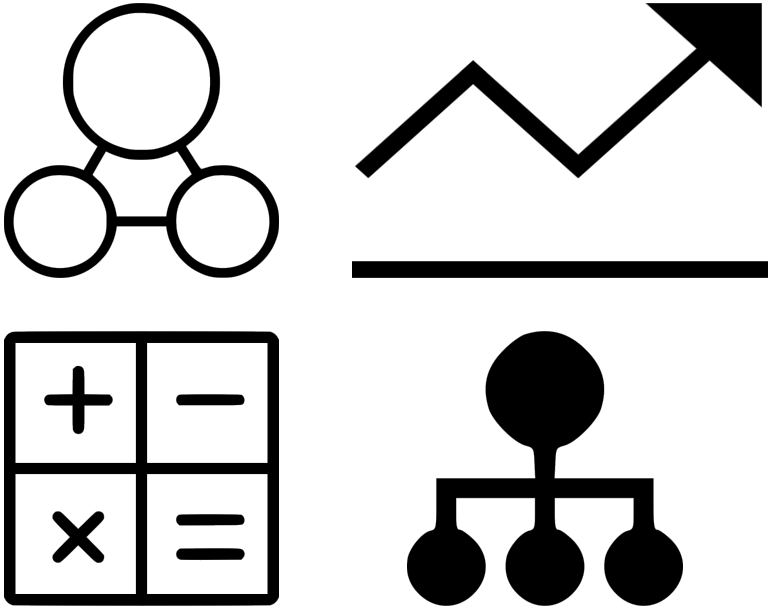
- Too little masking: Too expensive to train
- Too much masking: Not enough context

https://neptune.ai/blog/unmasking-bert-transformer-model-performance

ML jargon

The model is the algorithm + hyperparameters; parameters are the result of training the model

- Algorithm:
    - some math to optimally solve a problem given some training data
- Solver:
    - The software implementation/approximation of the math; many different ways to get the job done
- Hyperparameters:
    - Knobs/Dials that affect how the algorithm solves the problem, the solver is one such
- Parameters:
    - the solution to the optimization problem being solved on the training data



model selection

OUR GOAL IS TO PREVENT OVERFITTING!!!!

https://arstechnica.com/health/2023/11/ai-with-90-error-rate-forces-elderly-out-of-rehab-nursing-homes-suit-claims/

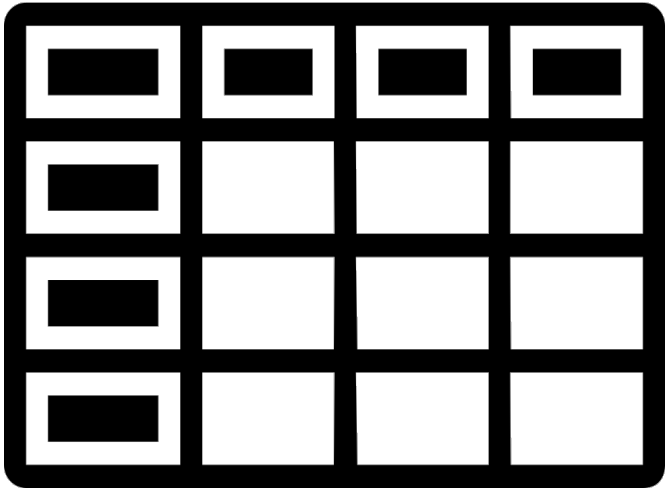
Preventing overfitting is important!

When AI models make errors it is typically because the dataset is a bad sample, leading to increased likelihood of overfitting patterns that are particular to this bad sample.

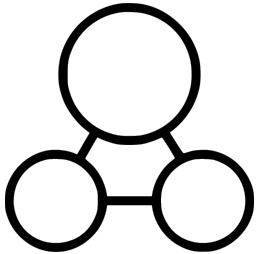
Generically there are two issues at play here.

Bad procedures (or carelessness) led to a bad model

Then ethical failings led to this bad model being put into service and kept in service

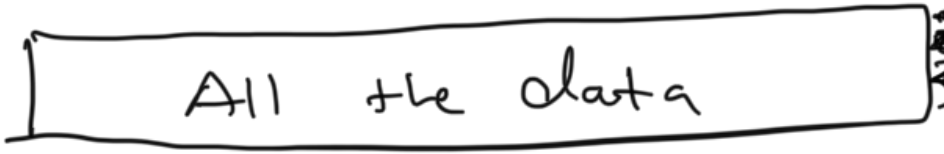


BIGGER
datasets



SIMPLER
models

1

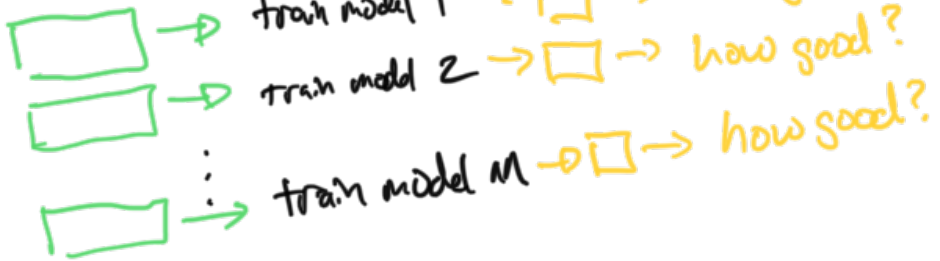


randomly shuffle rows of dataset

2

Split into training, validation and test sets

3



- if you have M models to try do this once for each

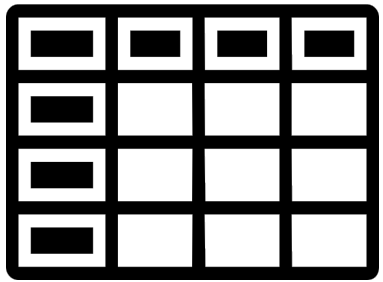
4

Pick best model from validation set performance

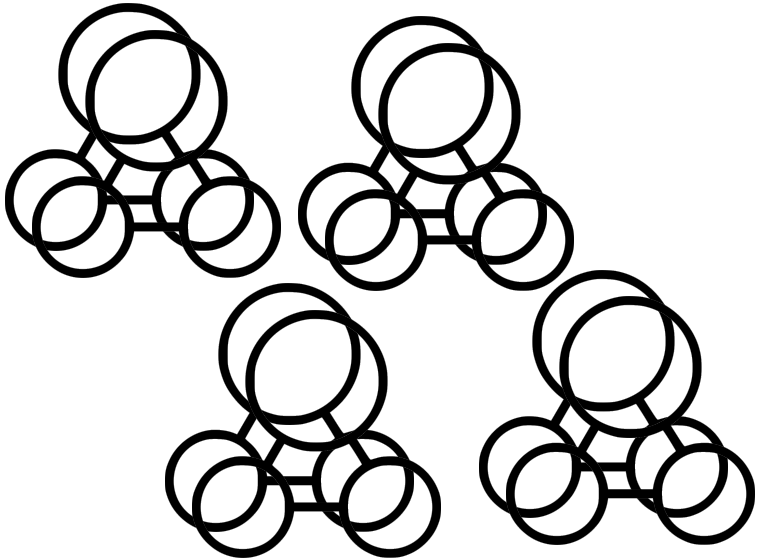
train it on T+V then test it on Test set

Model, feature, and hyperparameter selection

- Needs: Don’t leak information from training/selection process into the test set!
- Trade-offs: Usually not enough data to have completely separate train, validation, test sets. Which one do we prioritize?
    - Low training data -> bad fit
    - Low validation data -> bad selection of model/feature/hparam
    - Low test data -> poor estimate of generalization performance



SMALLER
datasets



COMPLEX
models

1. All the data

randomly shuffle rows of dataset

2. Test

split off a test set

3. fold 1, fold 2, fold 3, ..., fold K

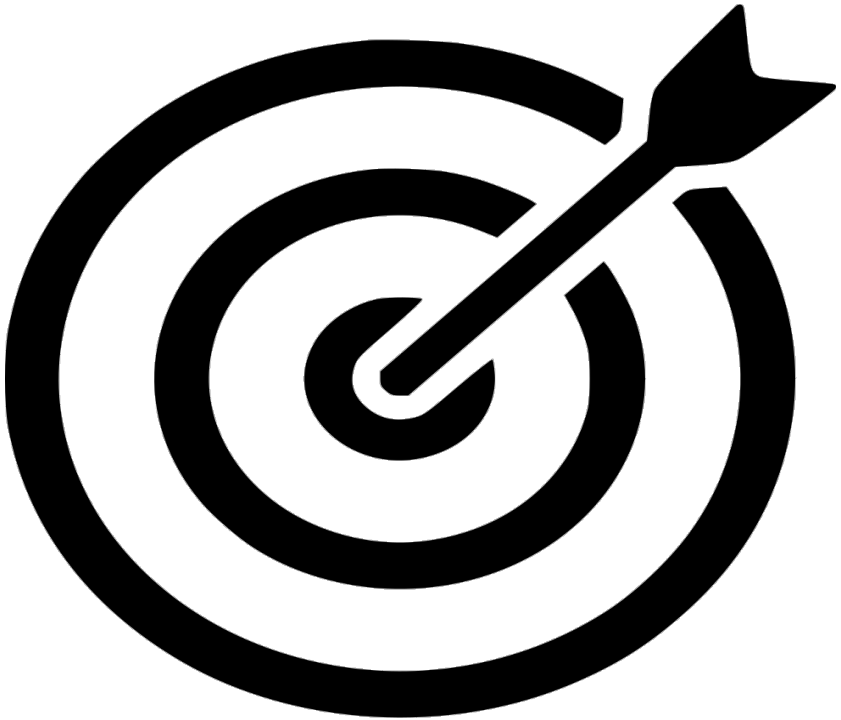
split remaining data into K folds with even(ish) amounts of data w/ same(ish) distributions

4. split 1 → train / validate; split 2 → train / valid; split 3 → train; ... split K; etc ...

Every split gets its turn as the validation set, use all other splits as training.

Validation performance is the mean across splits

5. Do steps 3+4 M times to test M models, pick best validation performance model → trained best model → Estimated performance in future



model assessment

continuous variable prediction

Squared Error

Computationally simpler

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

Same units as y

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}$$

A few outliers can lead to a big increase in squared errors… even if all the other predictions are pretty good

continuous variable prediction

Alternatively

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

If you care more about most errors, and less about the outliers

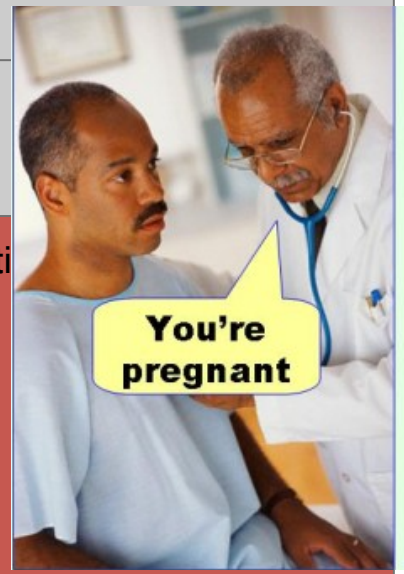
categorical variable prediction

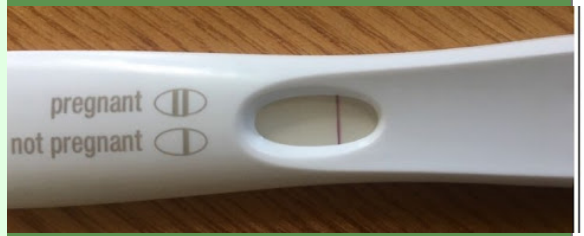
$$\text{Accuracy} = \frac{\#\text{ of samples predicted correctly}}{\#\text{ of samples predicted}} * 100$$

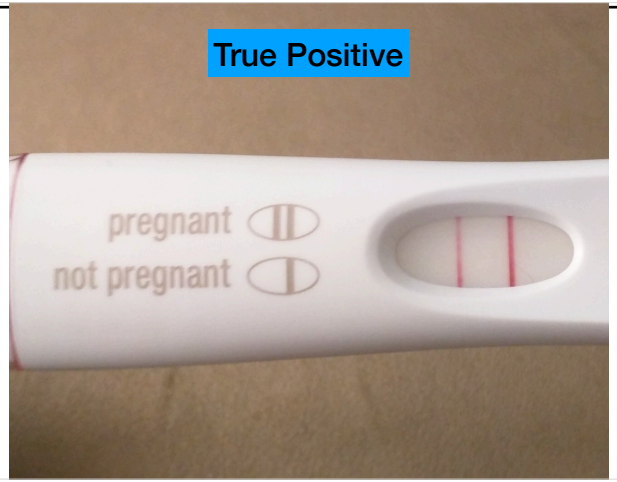
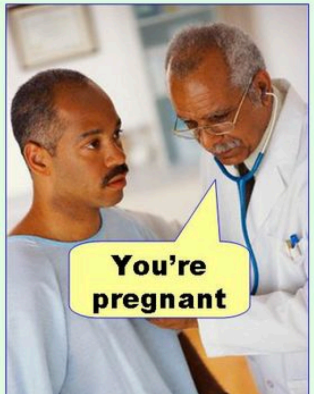
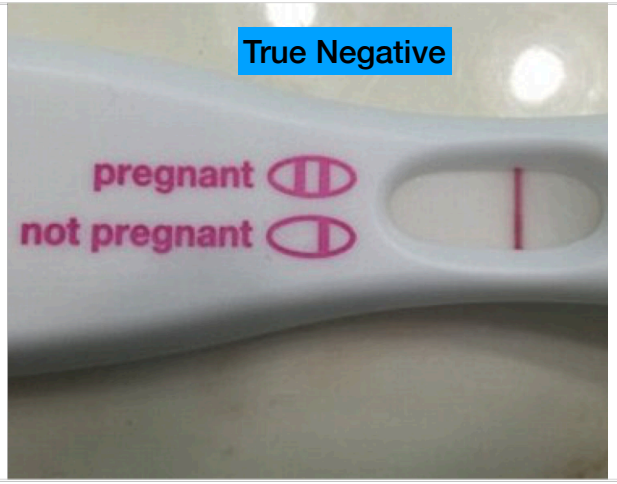
Accuracy can mislead

- If classes are imbalanced, what would “chance performance” be?
- The classic example: detect cancer
    - 1/1000 actually have cancer
    - your prediction algorithm misdiagnoses 1% of healthy people
    - Do the math… your algorithm tells 10 people who are healthy they are sick for every 1 person who is actually sick

		Actual	
		Positive	
Predicted	Positive	True Positive (TP)	False Positi
	Negative	False Negativ	True Negative (TN)





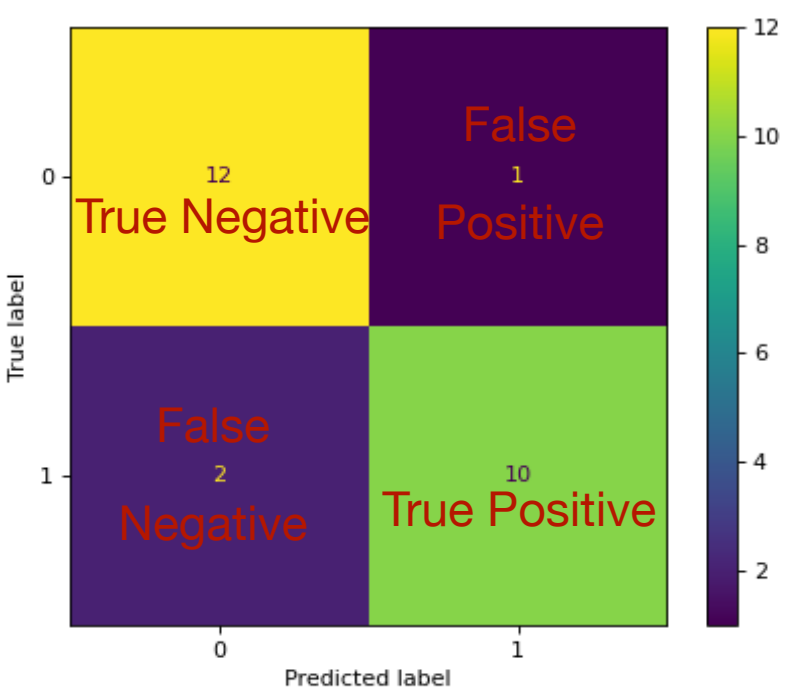
	Predicted +	Predicted -
Truly +	True Positive 	Type II error (false negative) 
Truly -	Type I error (false positive) 	True Negative 

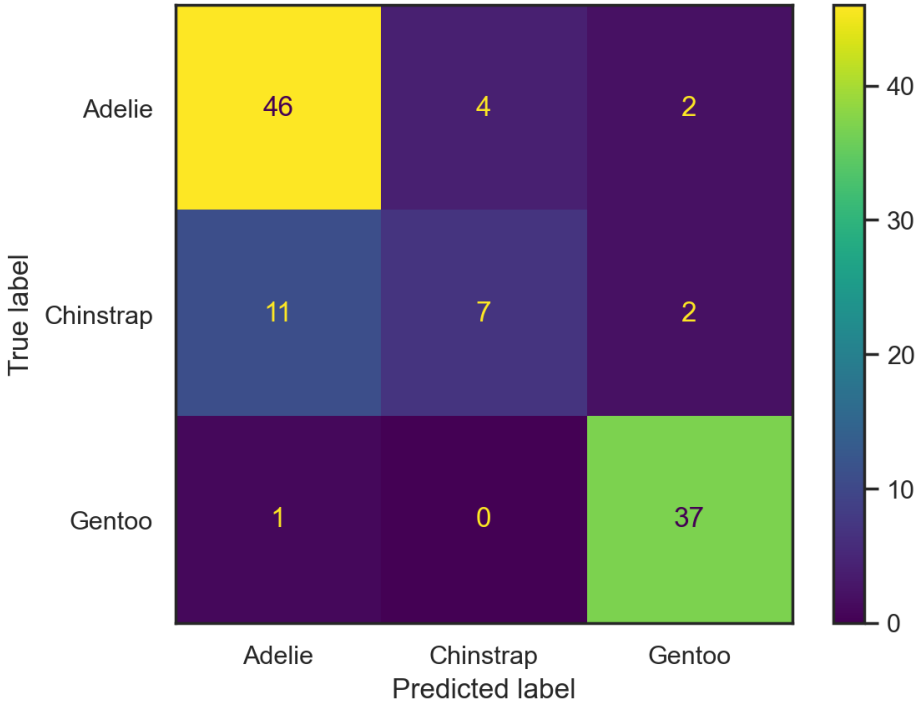
categorical variable prediction

confusion matrix

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

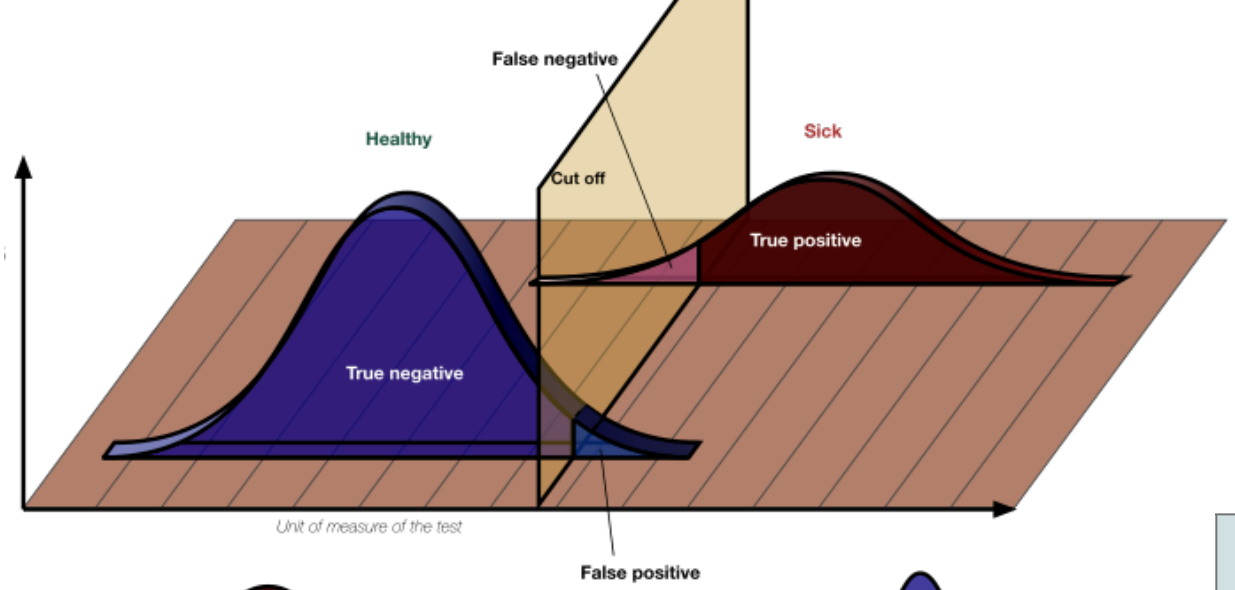
Confusion matrix





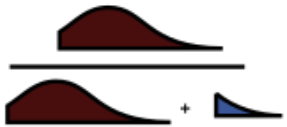
Accuracy	What % were predicted correctly?
Recall a.k.a. Sensitivity	Of those that *were* positives, what % were predicted to be positive?
Specificity	Of those that were *negatives*, what % were predicted to be negative?
Precision a.k.a. PPV	Of those that we predicted positive, what % were positives?

categorical variable prediction

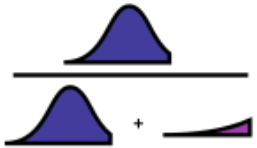


		Actual	
		Positive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

PPV =
aka recall

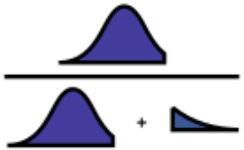


NPV =



Sensitivity =
aka recall

Specificity =



Prediction Approach

https://forms.gle/ubN7LFy3FdbSmcEi6