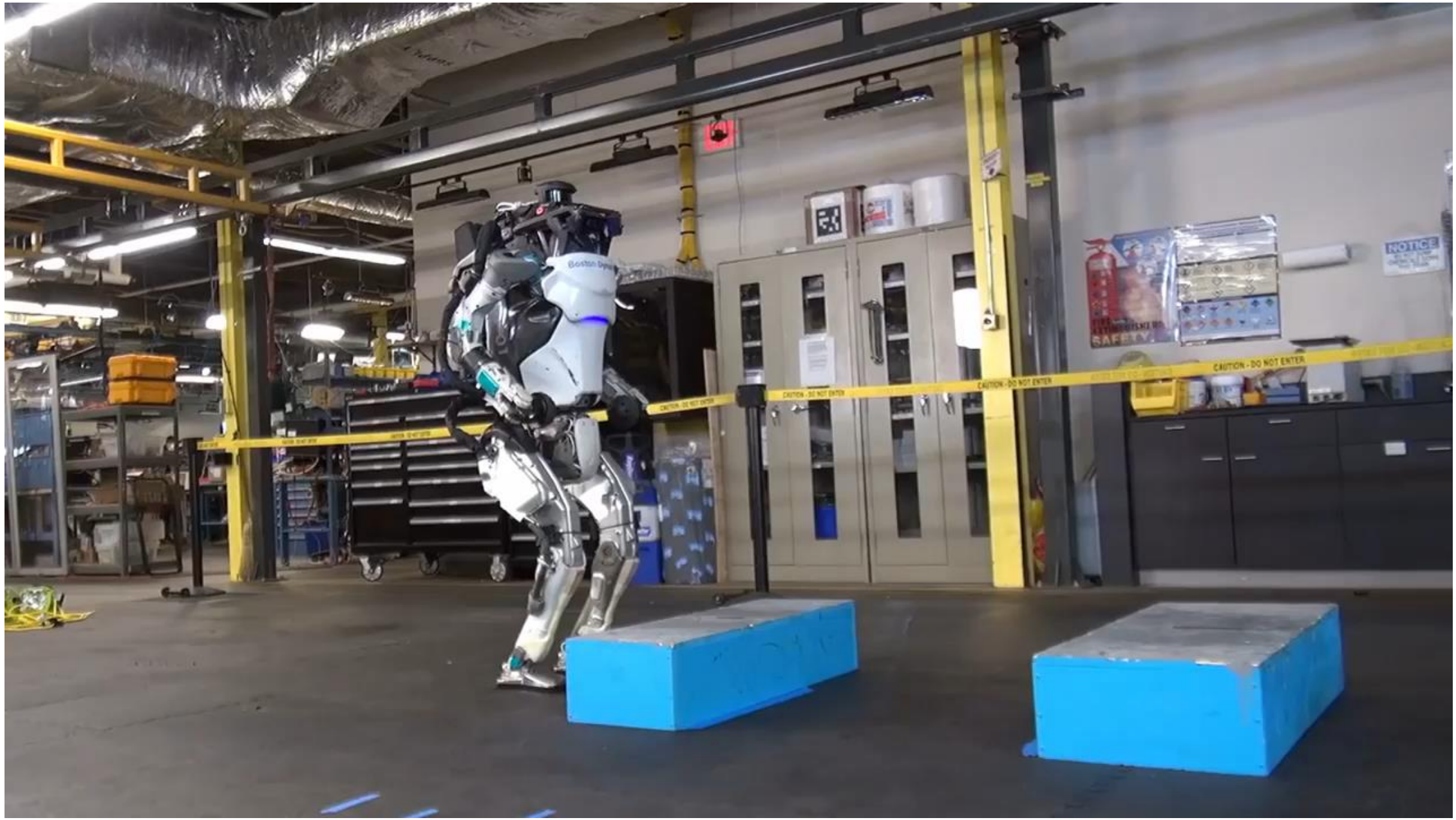
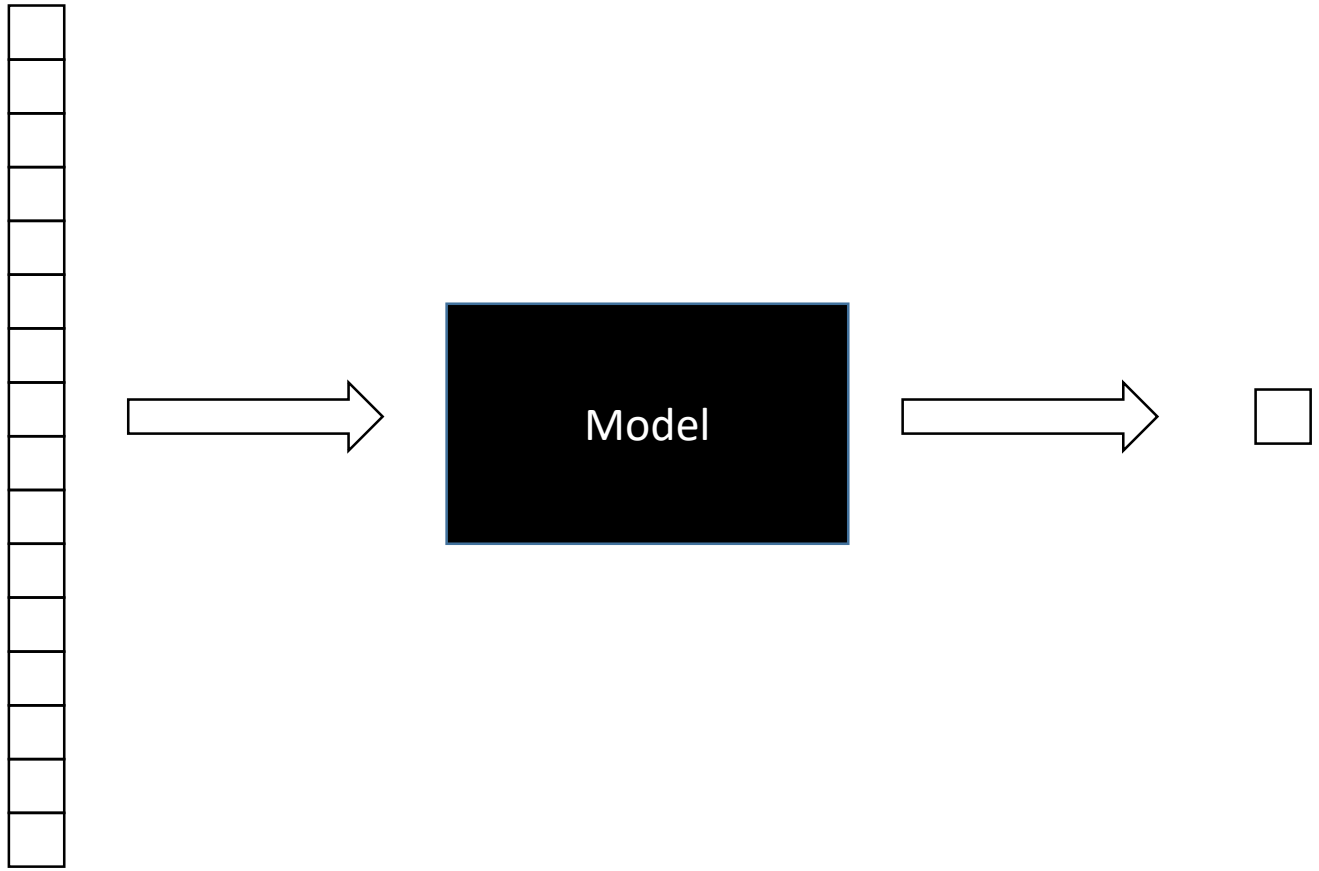
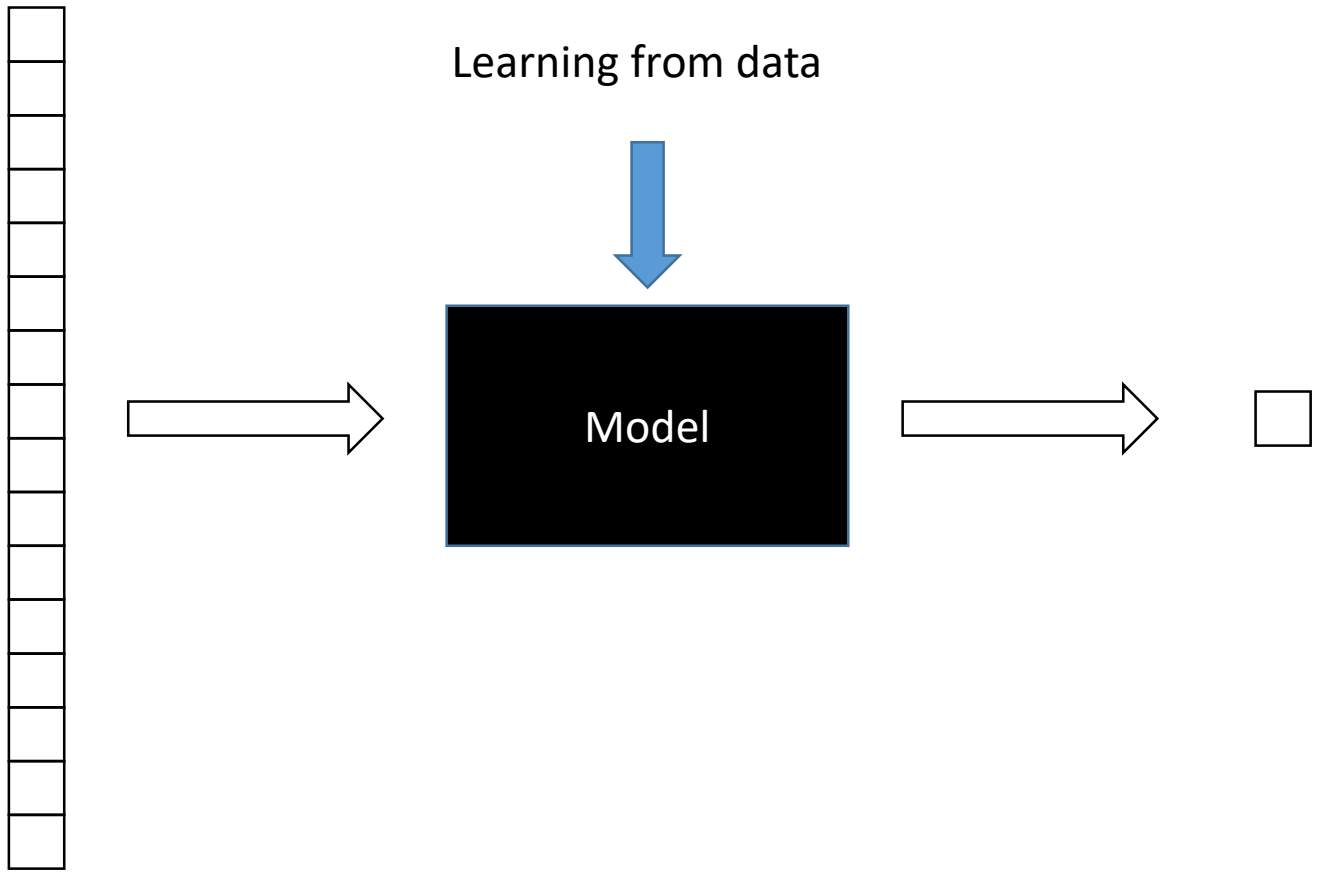
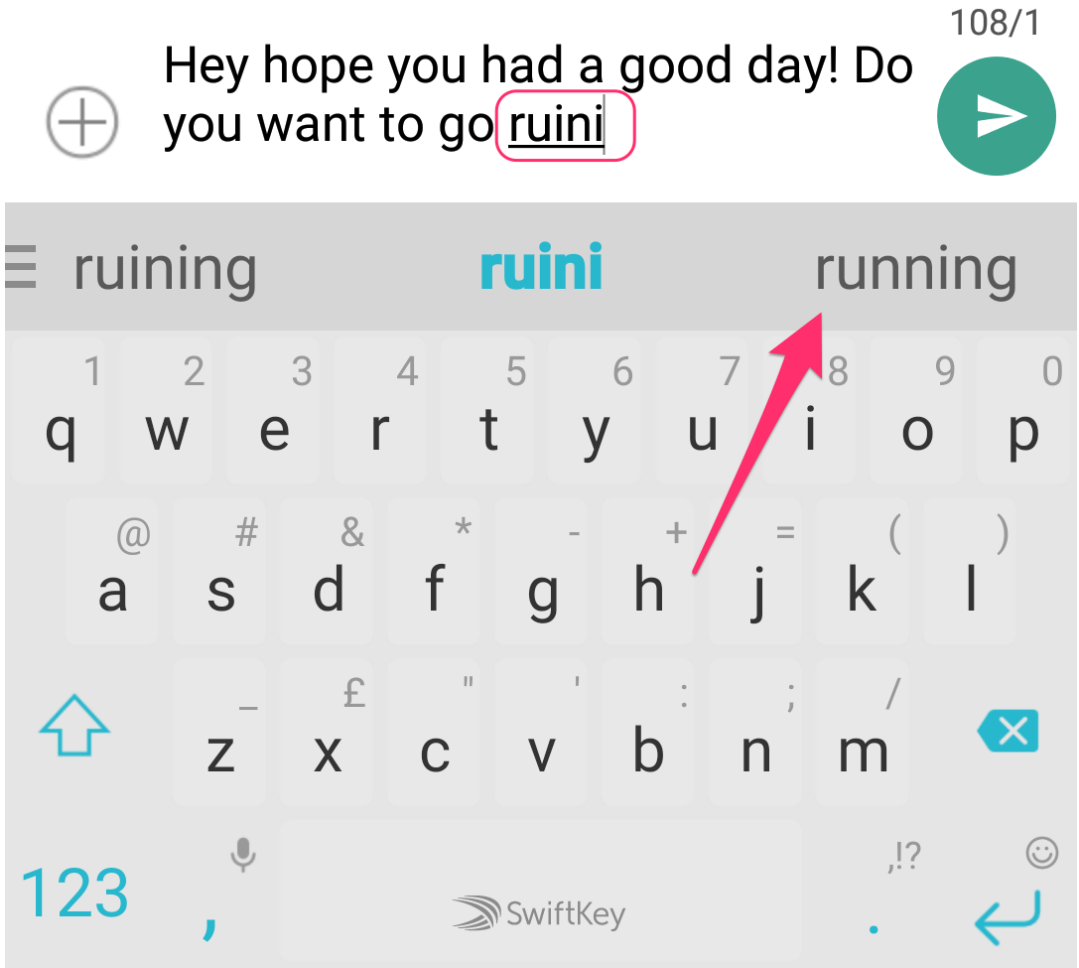

CAUTION - DO NOT ENTER
NOTICE

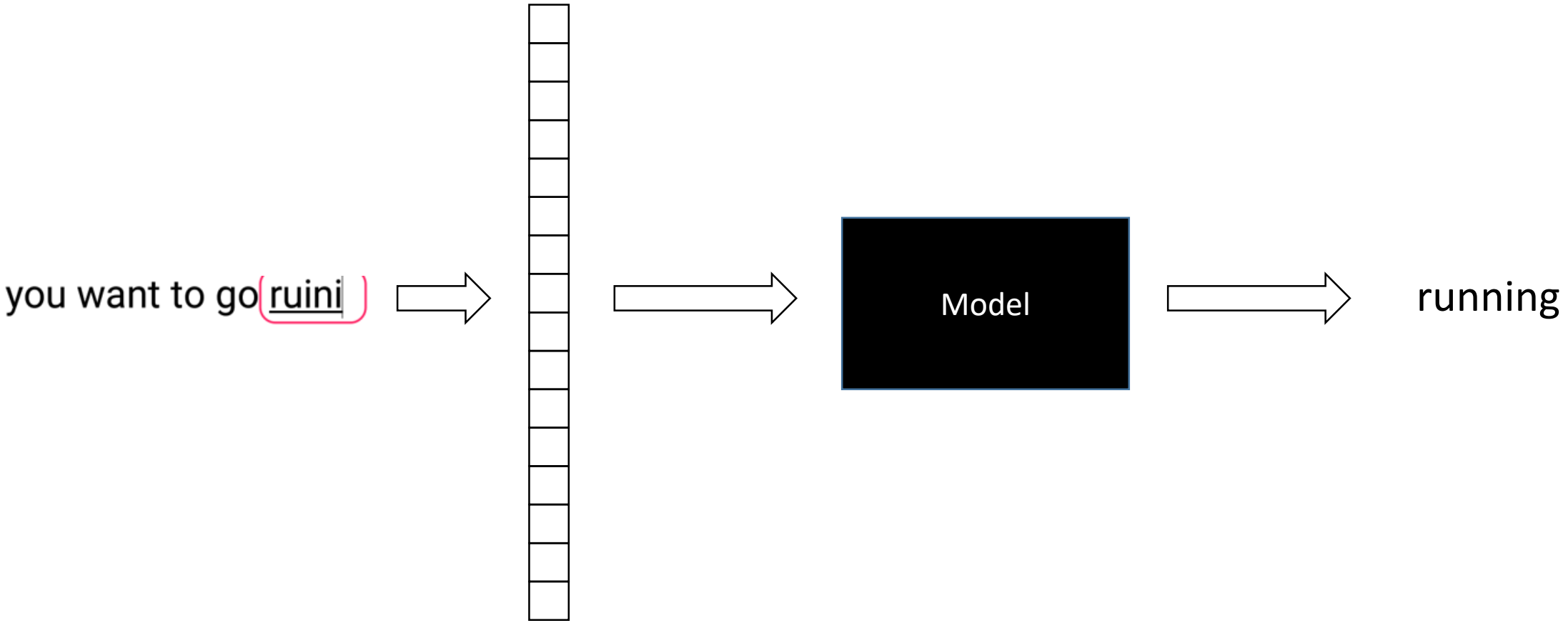
Boston Dynamics

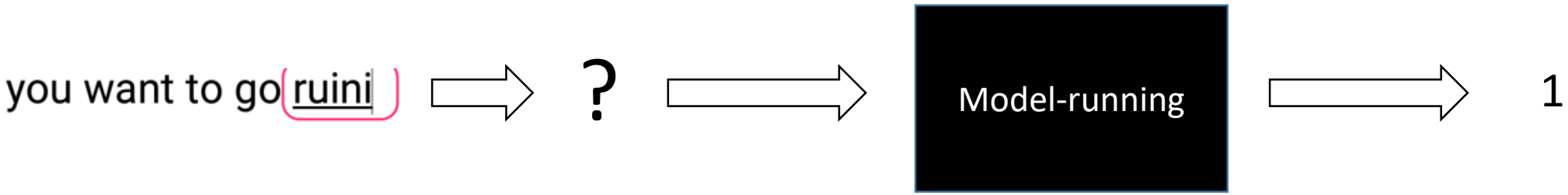

Model


Learning from data
Model

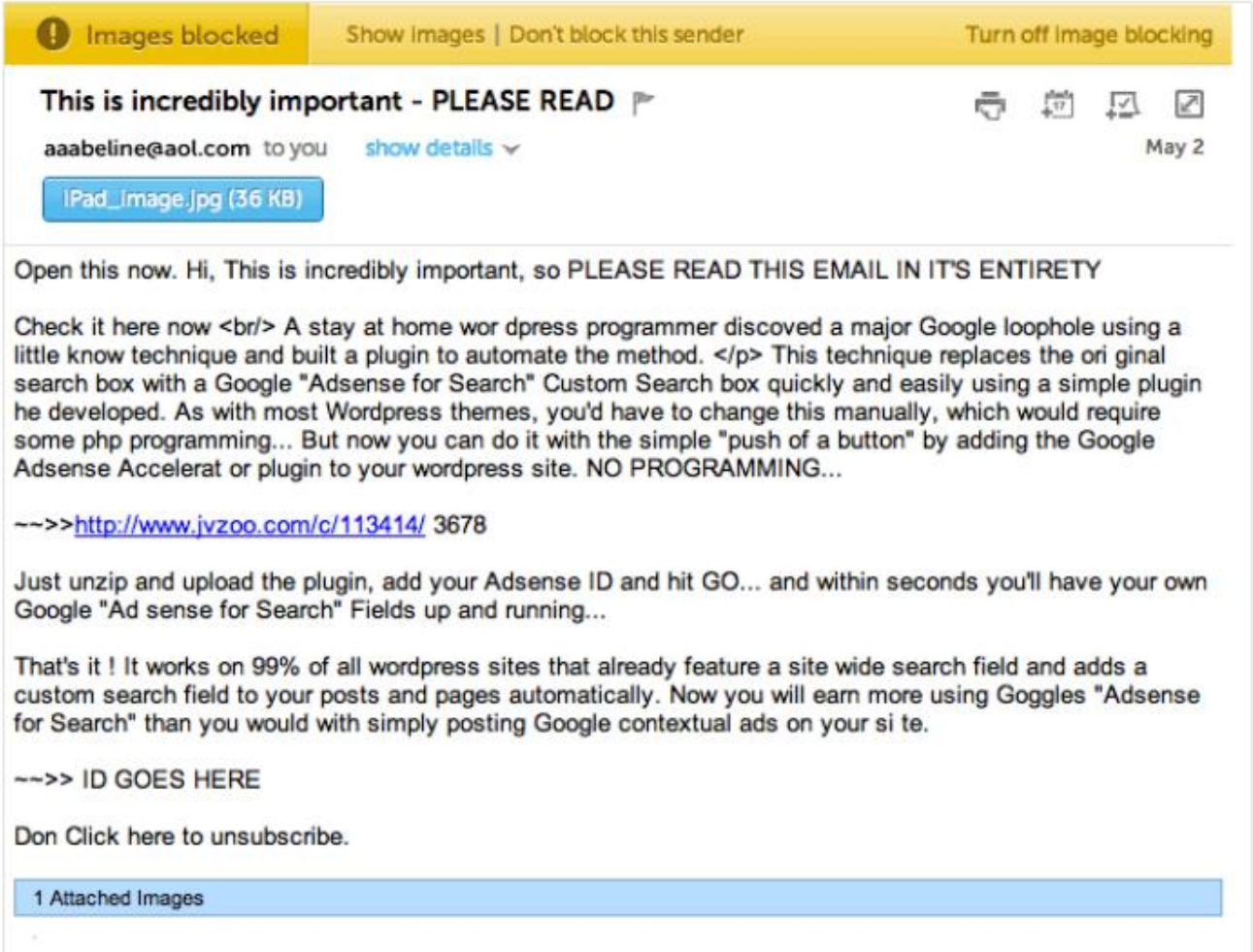
SwiftKey


108/1
Hey hope you had a good day! Do you want to go ruini
ruining
ruini
running
SwiftKey


you want to go ruini
Model
running


you want to go ruini
?
Model-running
1

Sent Mail
Spam (372)
Trash



Images blocked | Show images | Don't block this sender | Turn off image blocking

This is incredibly important - PLEASE READ

aaabeline@aol.com to you show details May 2

IPad_Image.jpg (36 KB)

Open this now. Hi, This is incredibly important, so PLEASE READ THIS EMAIL IN IT'S ENTIRETY

Check it here now
 A stay at home wor dpress programmer discoved a major Google loophole using a little know technique and built a plugin to automate the method. </p> This technique replaces the ori ginal search box with a Google "Adsense for Search" Custom Search box quickly and easily using a simple plugin he developed. As with most Wordpress themes, you'd have to change this manually, which would require some php programming... But now you can do it with the simple "push of a button" by adding the Google Adsense Accelerat or plugin to your wordpress site. NO PROGRAMMING...

~~>>http://www.jvzoo.com/c/113414/ 3678

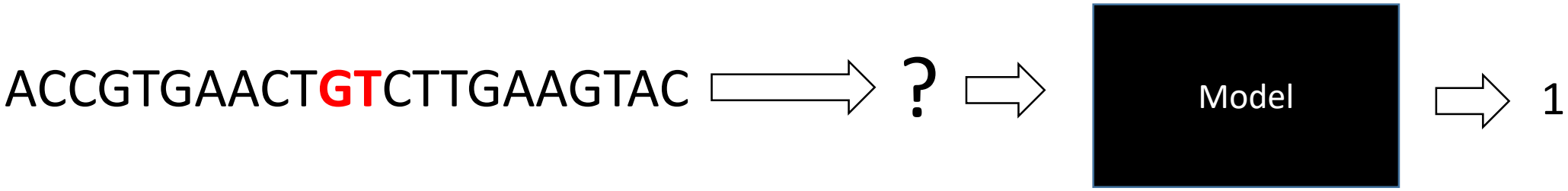
Just unzip and upload the plugin, add your Adsense ID and hit GO... and within seconds you'll have your own Google "Ad sense for Search" Fields up and running...

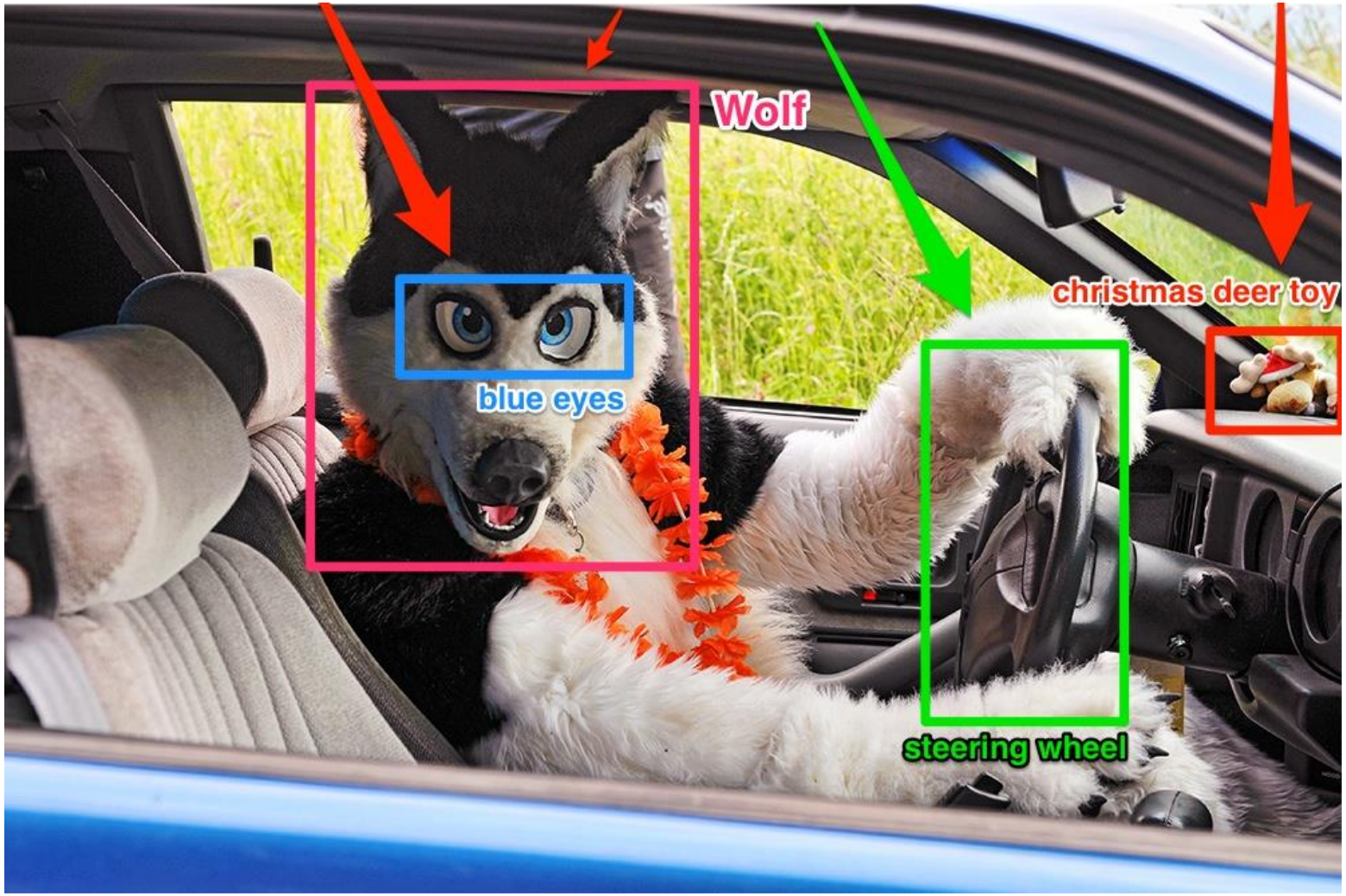
That's it ! It works on 99% of all wordpress sites that already feature a site wide search field and adds a custom search field to your posts and pages automatically. Now you will earn more using Goggles "Adsense for Search" than you would with simply posting Google contextual ads on your si te.

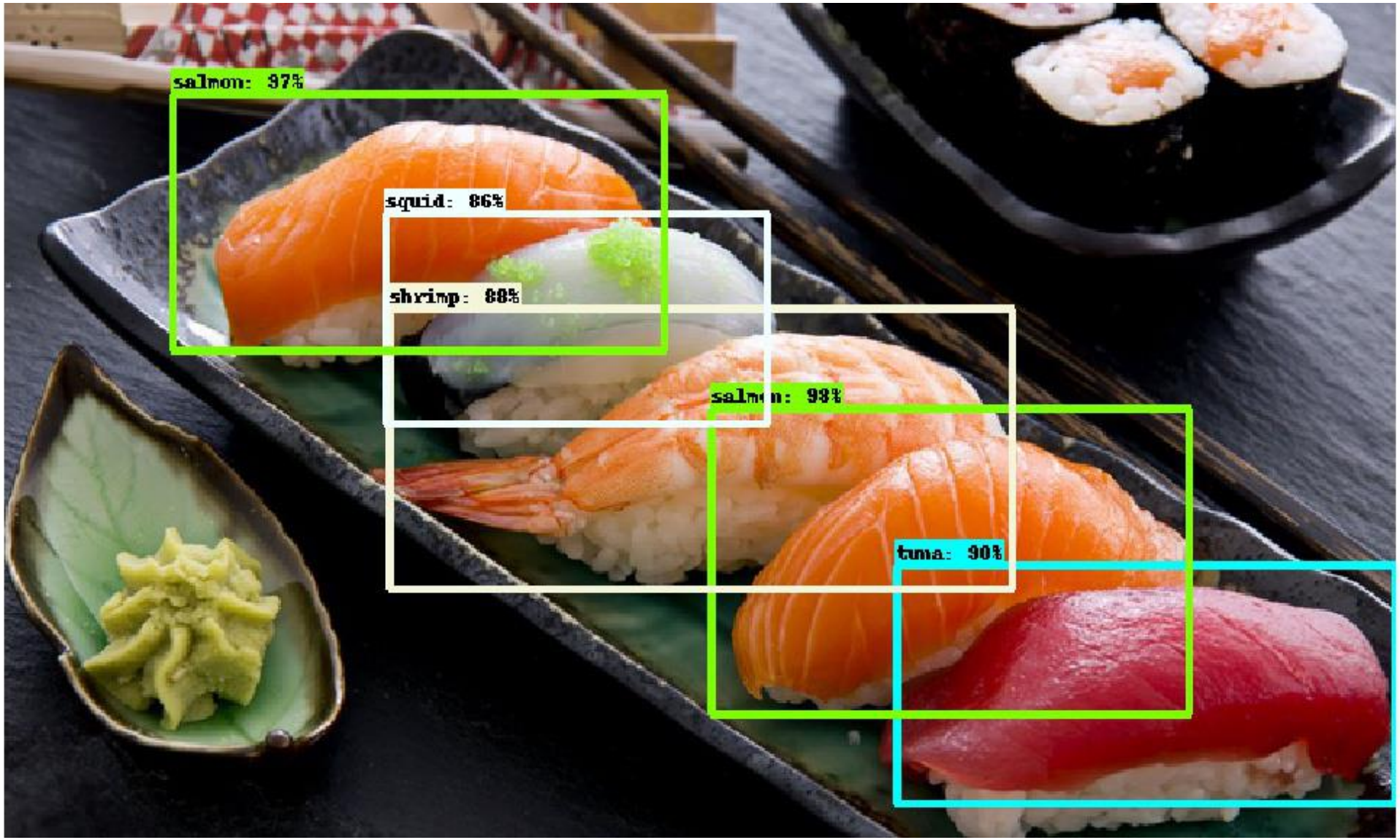
~~>> ID GOES HERE

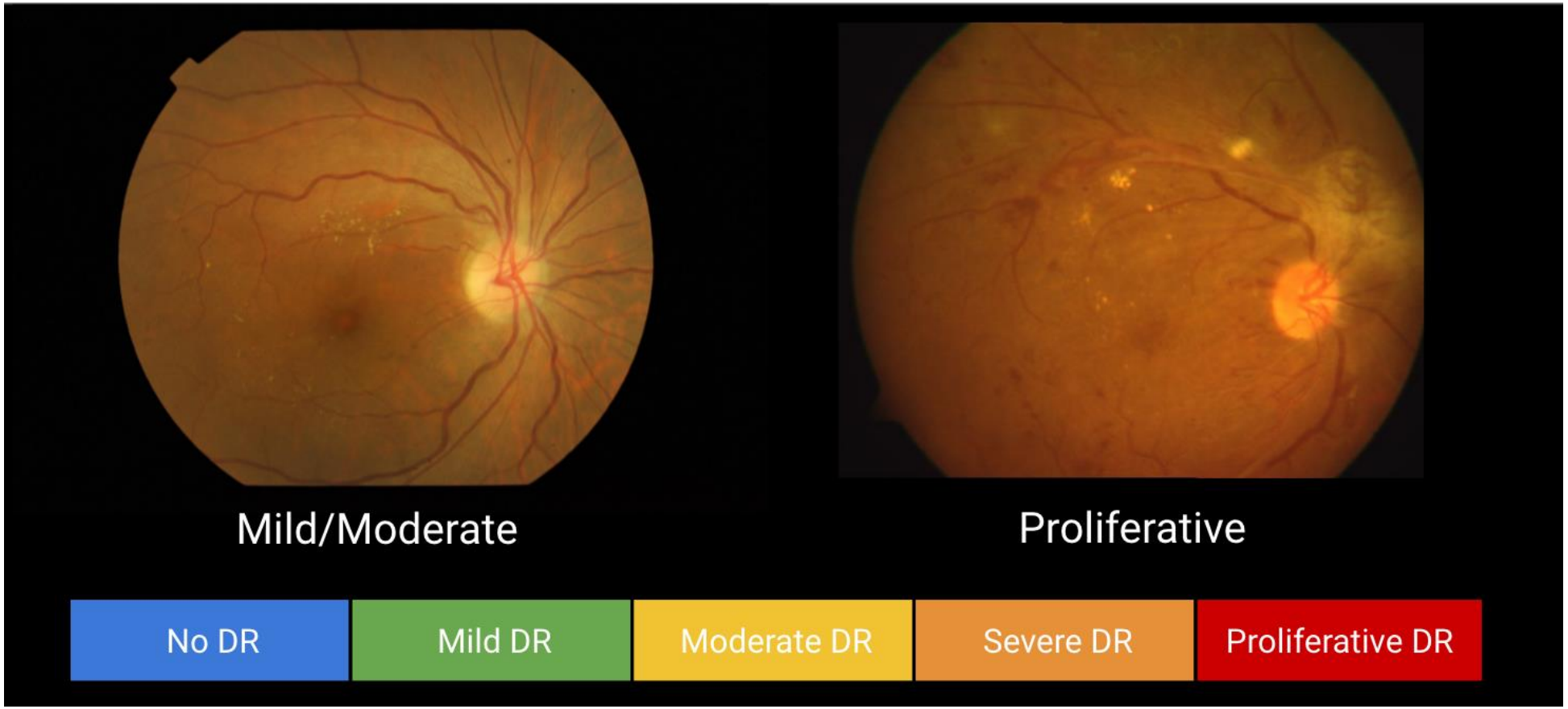
Don Click here to unsubscribe.

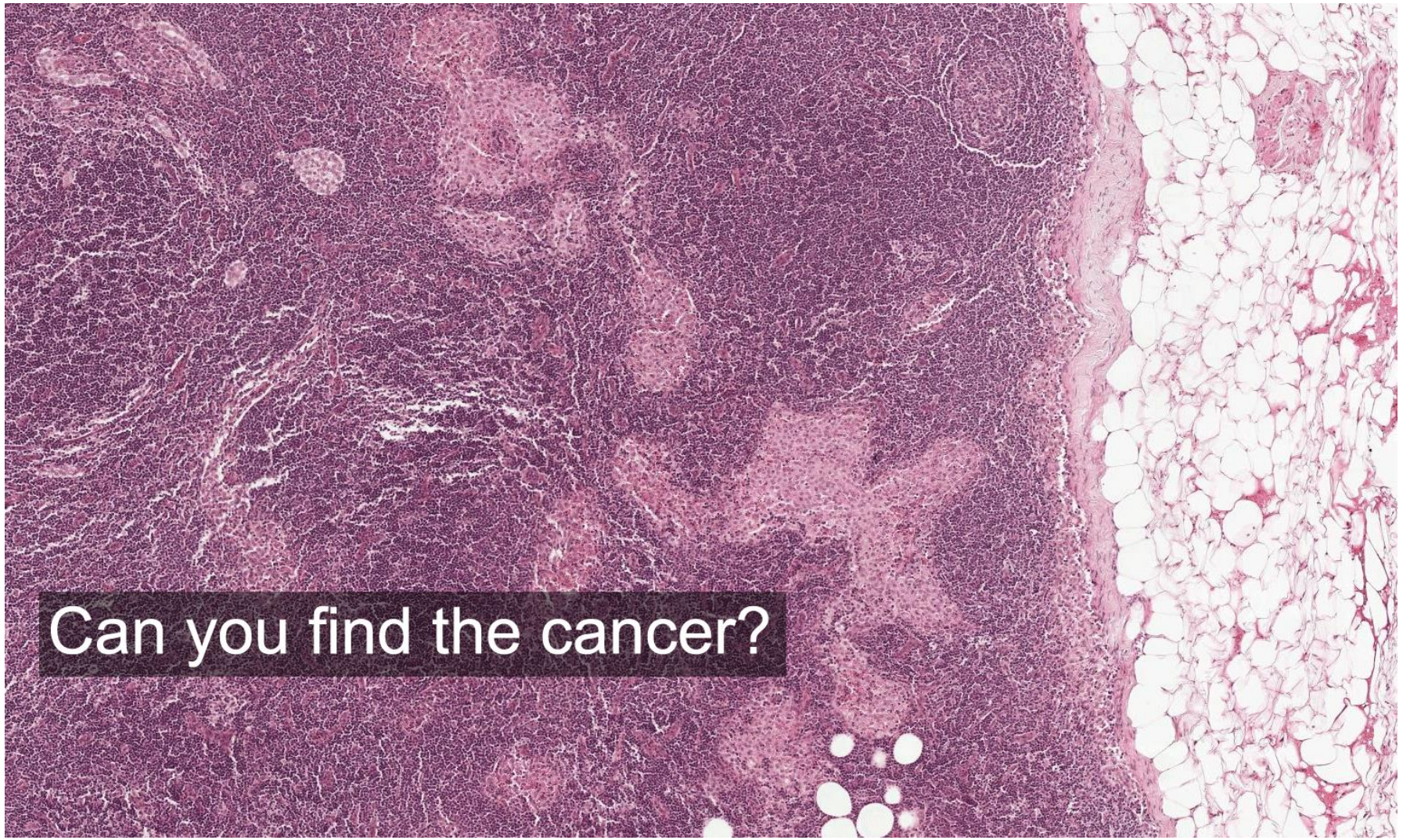
1 Attached Images

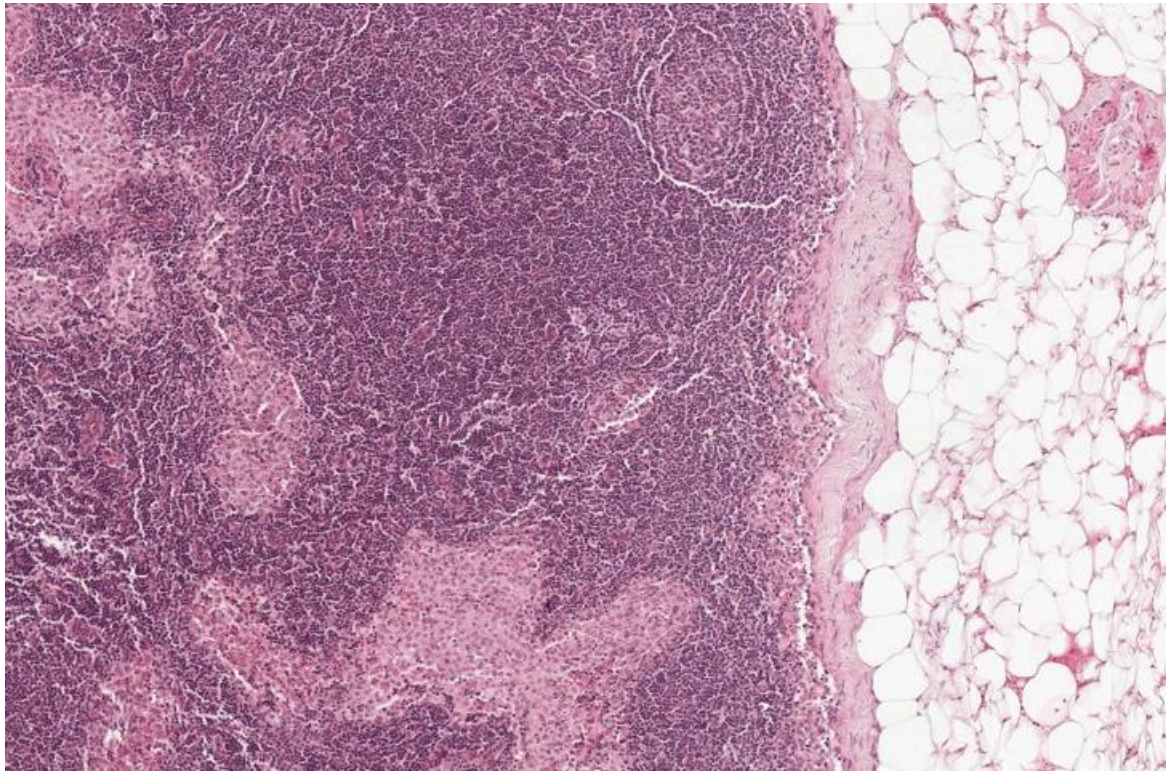

ACCGTGAACTGTCTTGAAGTAC
?
Model
1


Wolf
blue eyes
christmas deer toy
steering wheel


salmon: 97%
squid: 86%
shrimp: 88%
salmon: 93%
tuna: 90%


Mild/Moderate
Proliferative
No DR
Mild DR
Moderate DR
Severe DR
Proliferative DR

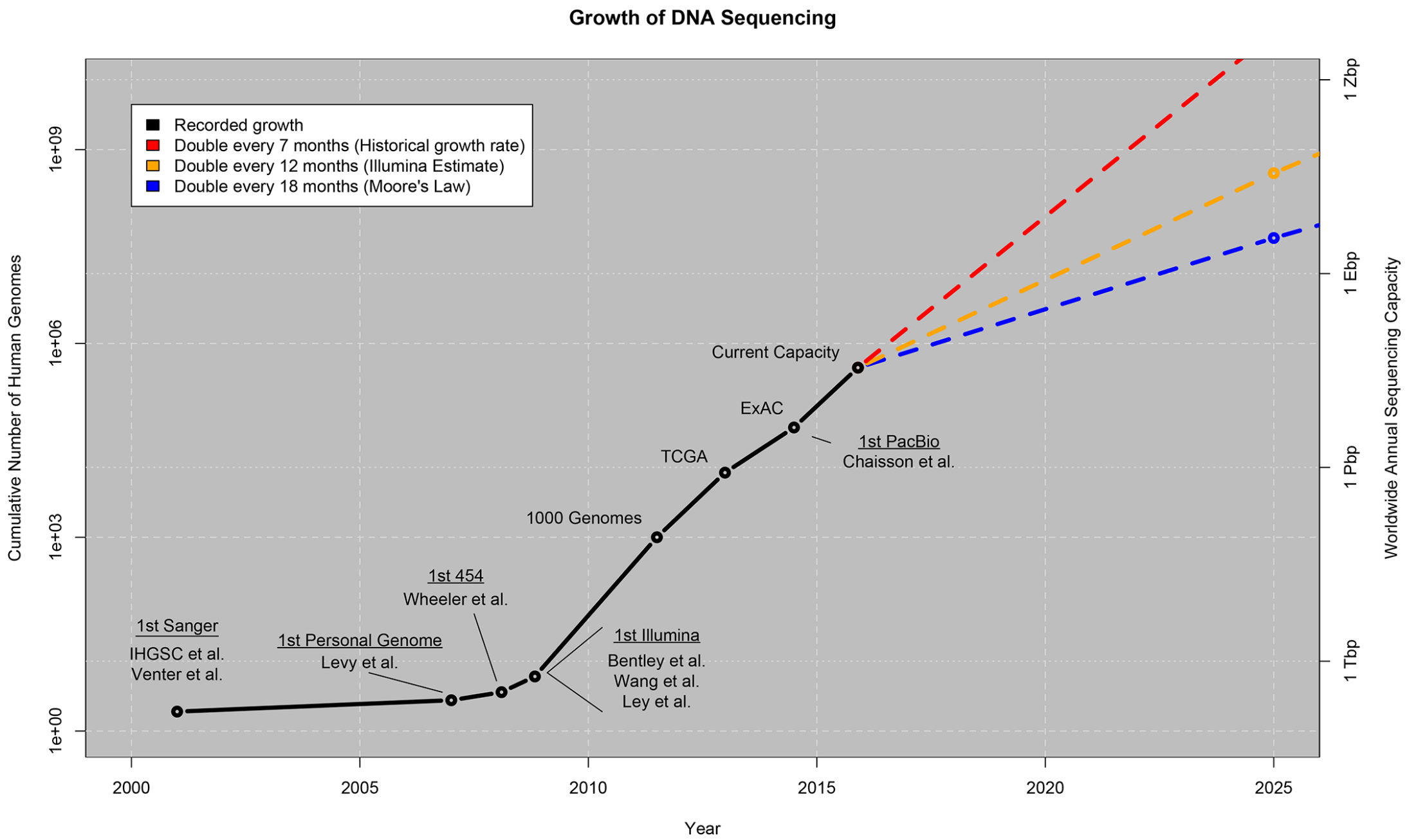

Can you find the cancer?



?

Model

1



Stephens ZD, Lee SY, Faghri F, Campbell RH, Zhai C, Efron MJ, et al. (2015) Big Data: Astronomical or Genomical? PLoS Biol 13(7): e1002195.

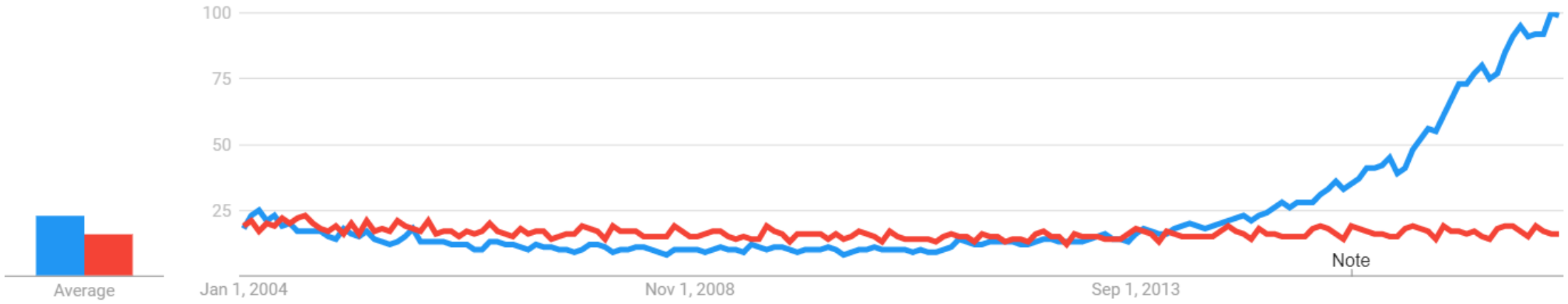
Machine learning
Field of study

Biomedical Sciences
Applied discipline

+ Add comparison

Worldwide | 2004 - present | All categories | Web Search

Interest over time



Nucleic acid sequence data

- DNA DataBank of Japan (DDBJ)
- European Nucleotide Archive (ENA)
- GenBank
- dbSNP
- European Variation Archive (EVA)
- dbVar
- Database of Genomic Variants Archive (DGVa)
- EBI Metagenomics
- NCBI Trace Archive
- NCBI Sequence Read Archive (SRA)
- NCBI Assembly

Protein sequence

- UniProtKB

Proteomics

- PRIDE
- PeptideAtlas
- ProteomeXchange

Chemistry & chemical biology

- caNanoLab
- ChEMBL
- NCBI PubChem BioAssay
- NCBI PubChem Substance

Functional genomics

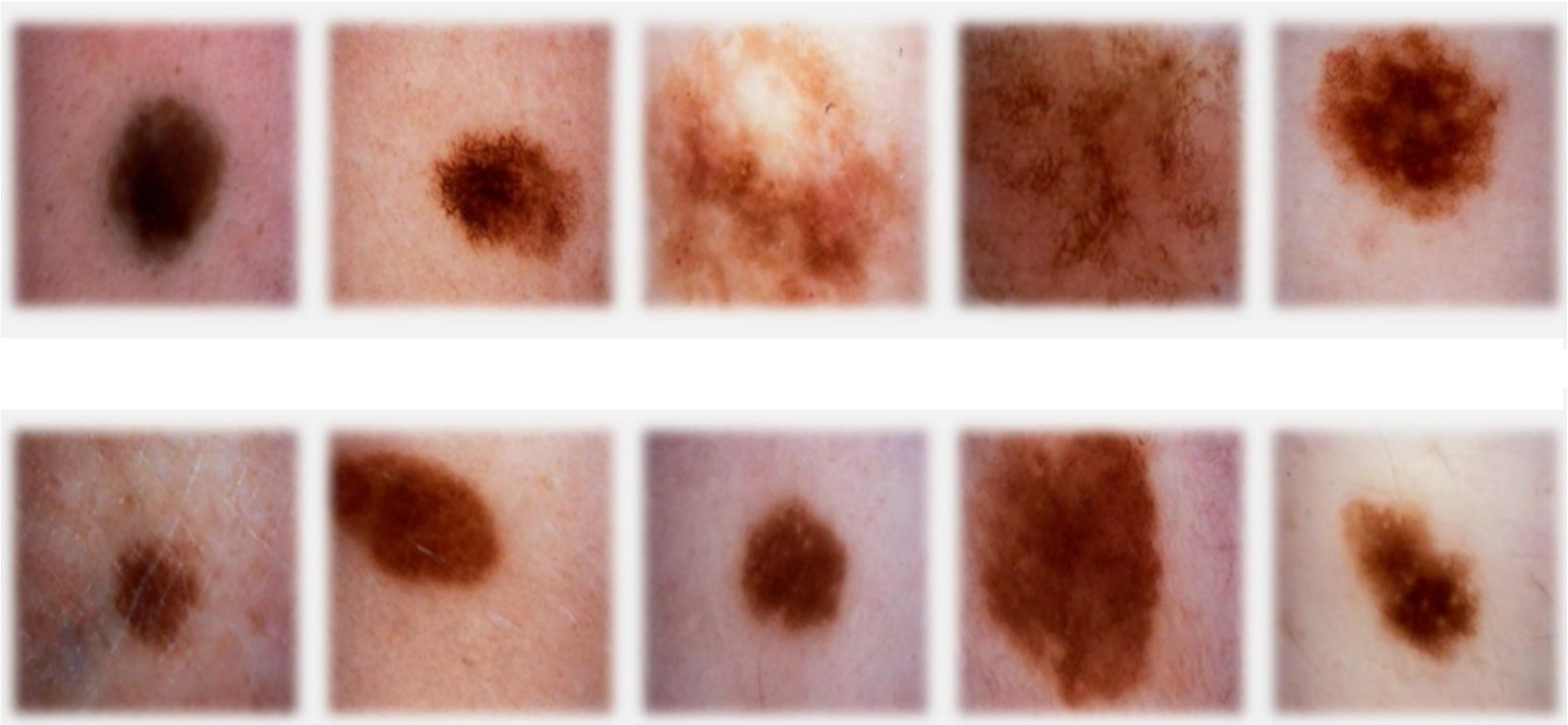
- ArrayExpress
- Gene Expression Omnibus (GEO)
- GenomeRNAi
- dbGAP
- The European Genome-phenome Archive (EGA)
- Database of Interacting Proteins (DIP)
- IntAct
- Japanese Genotype-phenotype Archive (JGA)
- Biological General Repository for Interaction Datasets
- NCBI PubChem BioAssay

Molecular & supramolecular structure

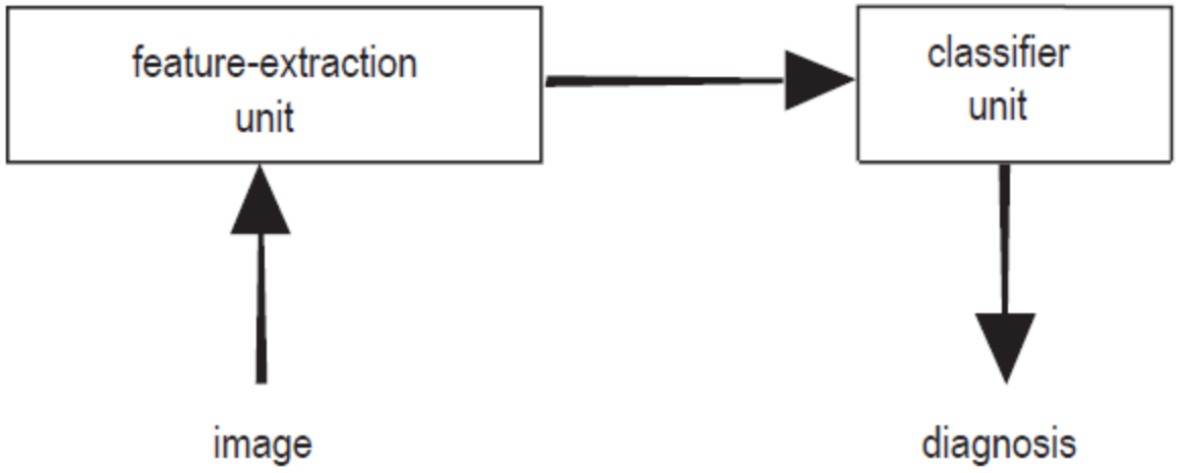
- Protein Circular Dichroism Data Bank (PCDDB)
- Crystallography Open Database (COD)
- Coherent X-ray Imaging Data Bank (CXIDB)
- Biological Magnetic Resonance Data Bank (BMRB)
- EMDataBank (EMDB)
- Worldwide Protein Data Bank (wwPDB)
- Structural Biology Data Grid

https://www.nature.com/sdata/policies/repositories

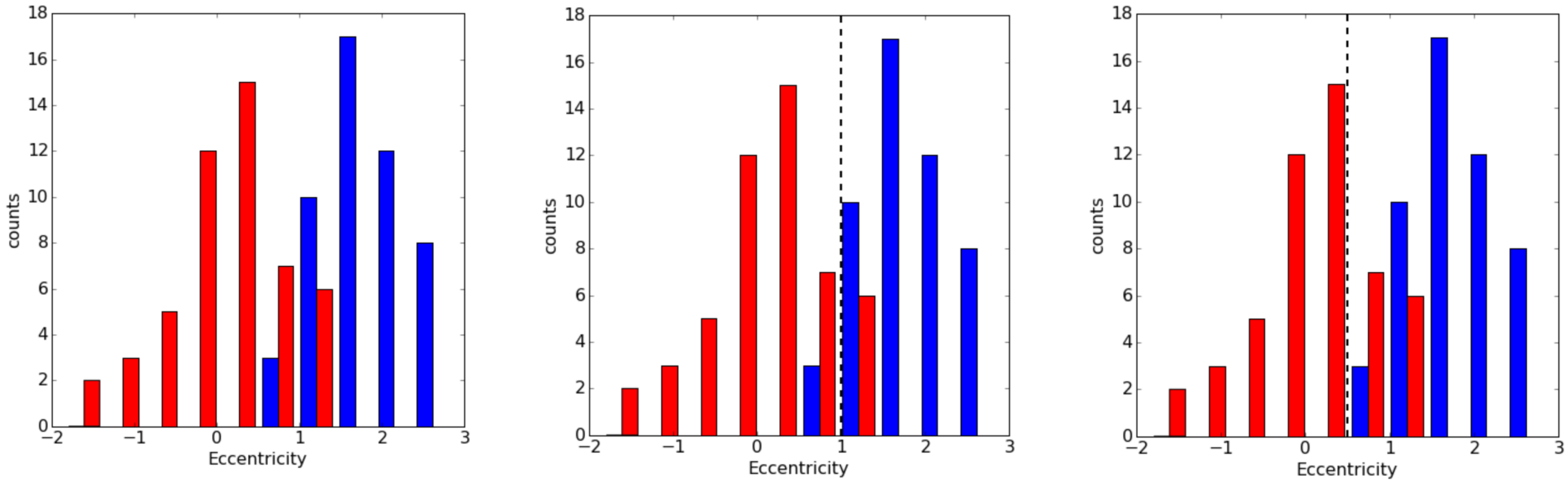
Introduction: concepts



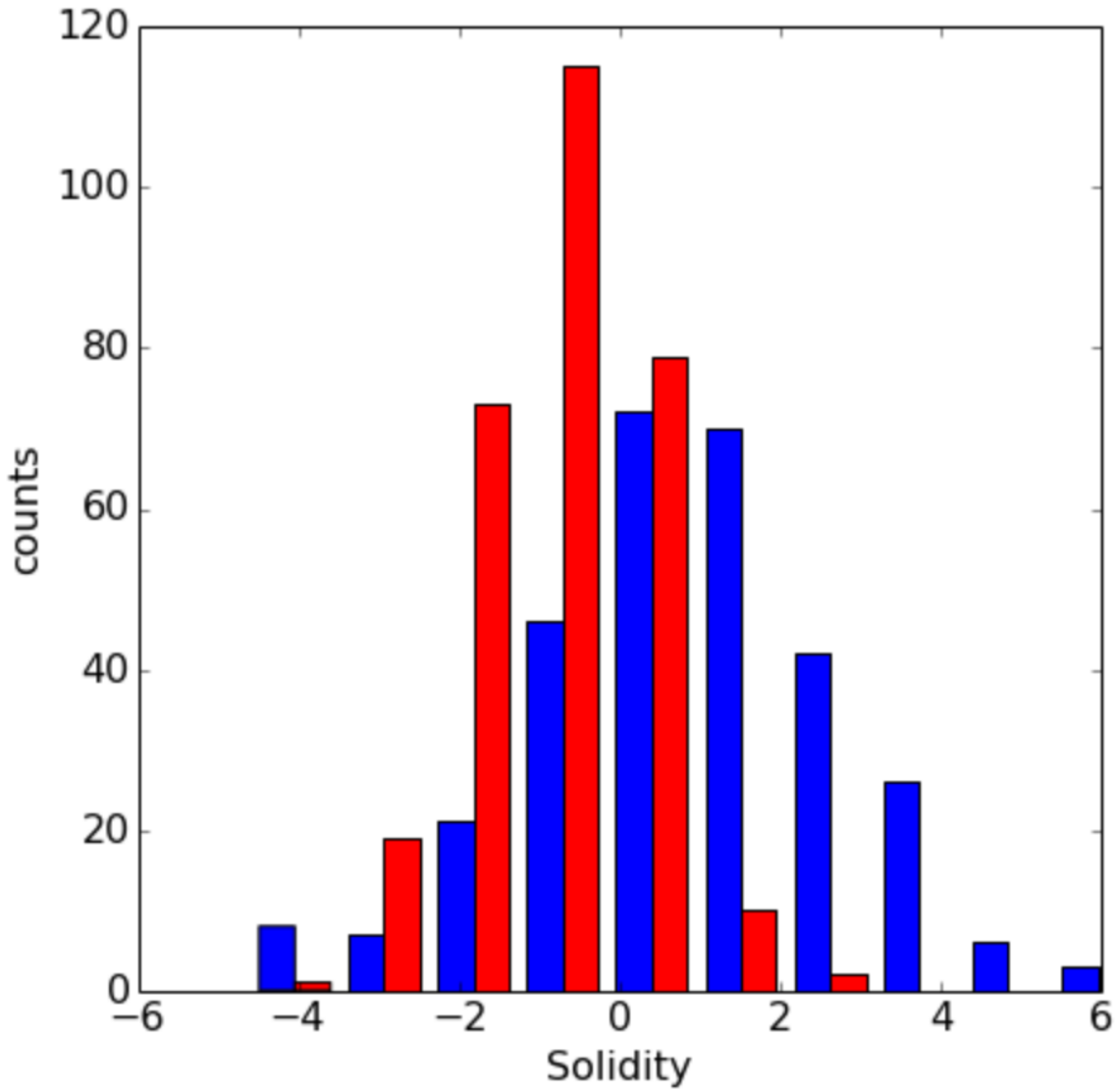
- sign of cancer
- top row malignant
- bottom row benign



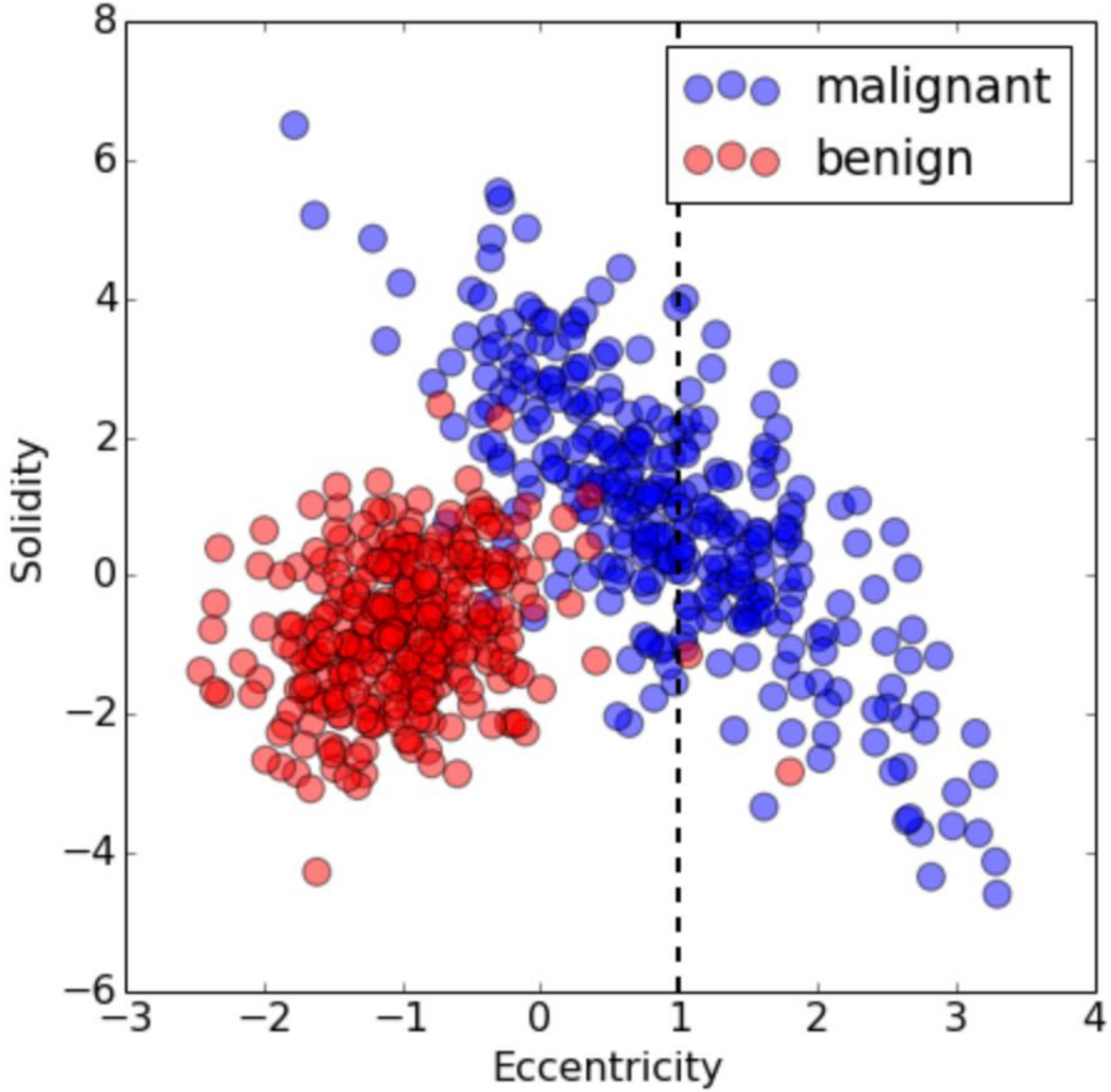
- feature extraction: features (a.k.a. properties or attributes)
- data set, sample (a.k.a. example, instance or data point), label (a.k.a. target)



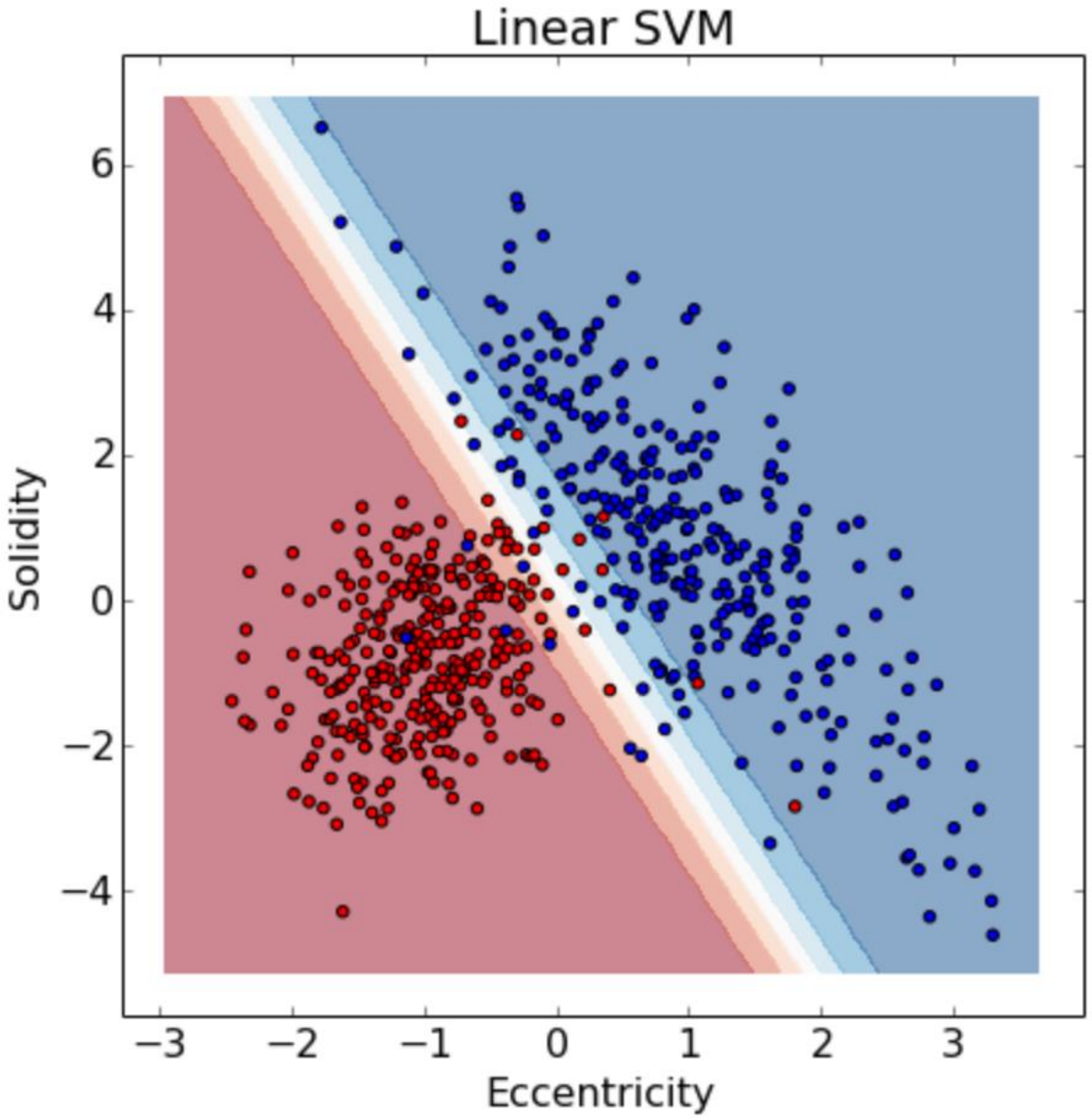
- eccentricity of lesion (how nearly circular the lesion is)
- threshold ***t***
- consequence of the predictions



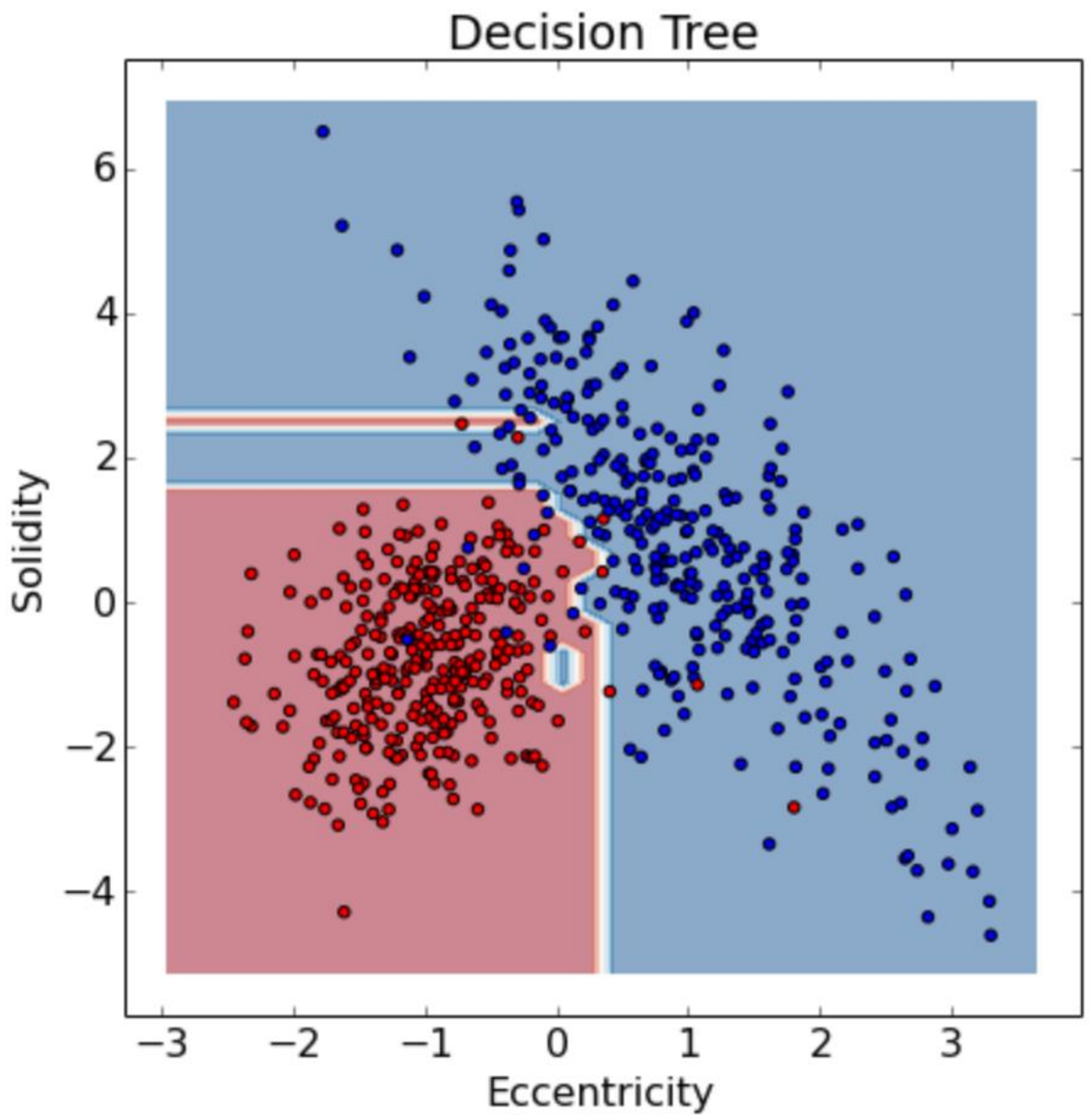
- add another feature?
- feature vector X
- Euclidean vector space



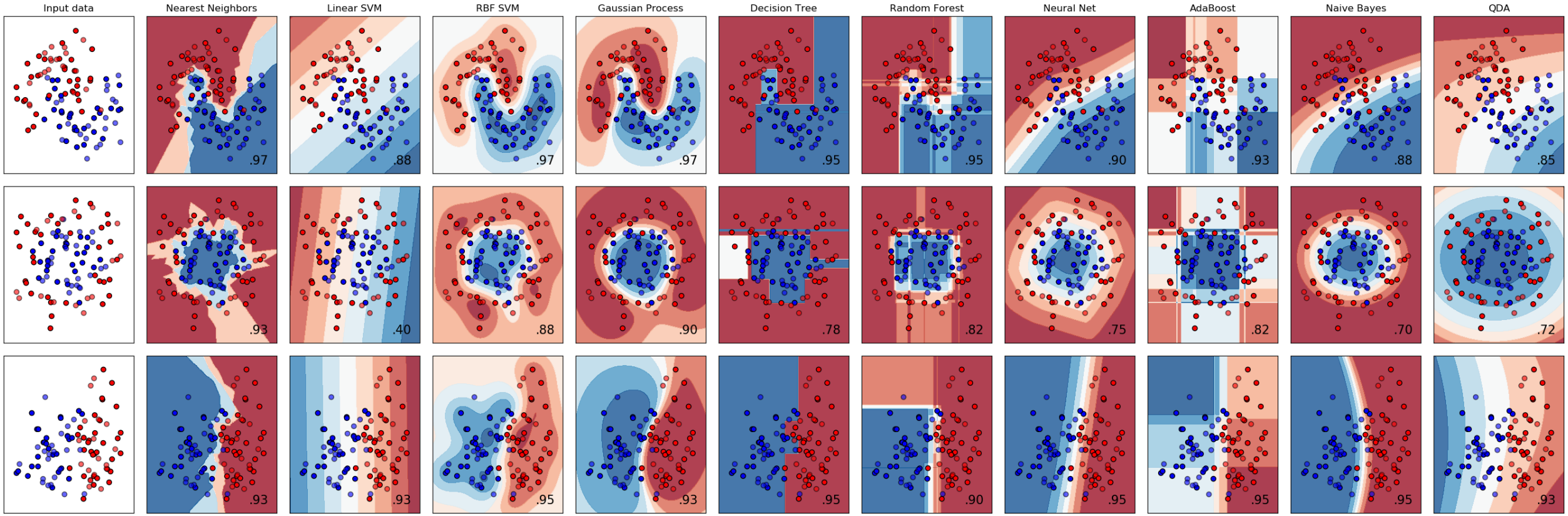
- feature vector X
- Euclidean vector space



- linear decision boundary
- blue region malignant, red region benign
- yet more features
- can’t look at the decision boundary
- more complex



- unseen external images
- generalization
- overfitting

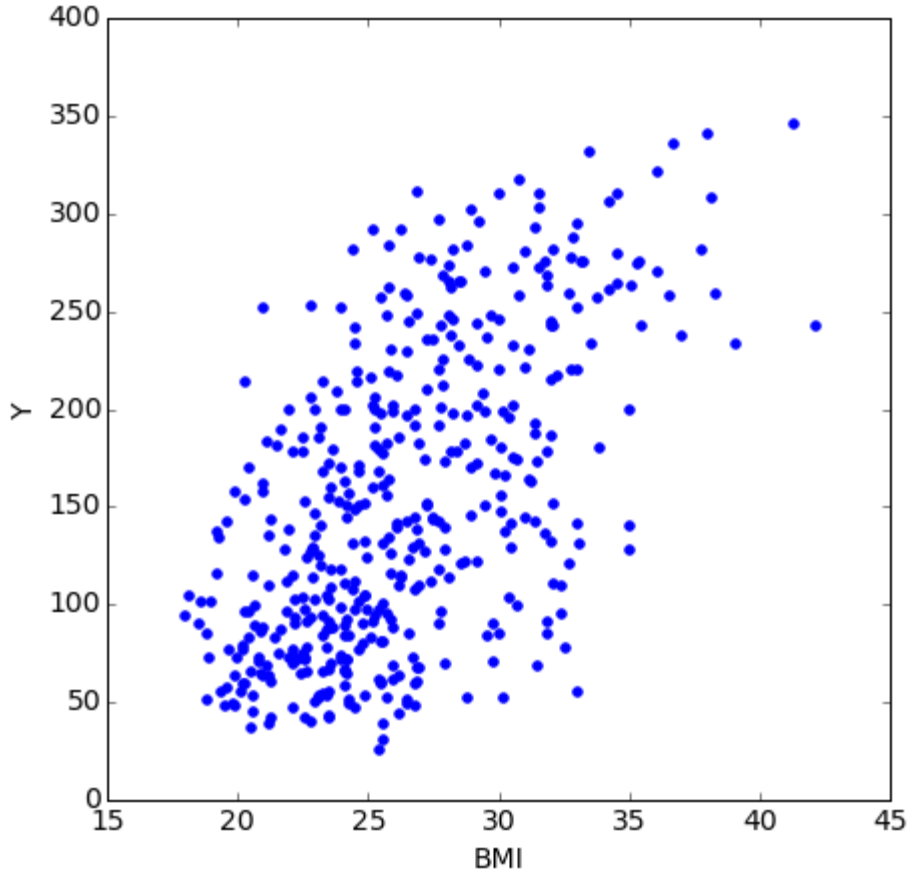

Input data
Nearest Neighbors
Linear SVM
RBF SVM
Gaussian Process
Decision Tree
Random Forest
Neural Net
AdaBoost
Naive Bayes
QDA
.97
.88
.97
.97
.95
.95
.90
.93
.88
.85
.93
.40
.88
.90
.78
.82
.75
.82
.70
.72
.93
.93
.95
.93
.95
.90
.95
.95
.95
.93

Preprocess data

https://www.emotiv.com/epoc/

Introduction: regression

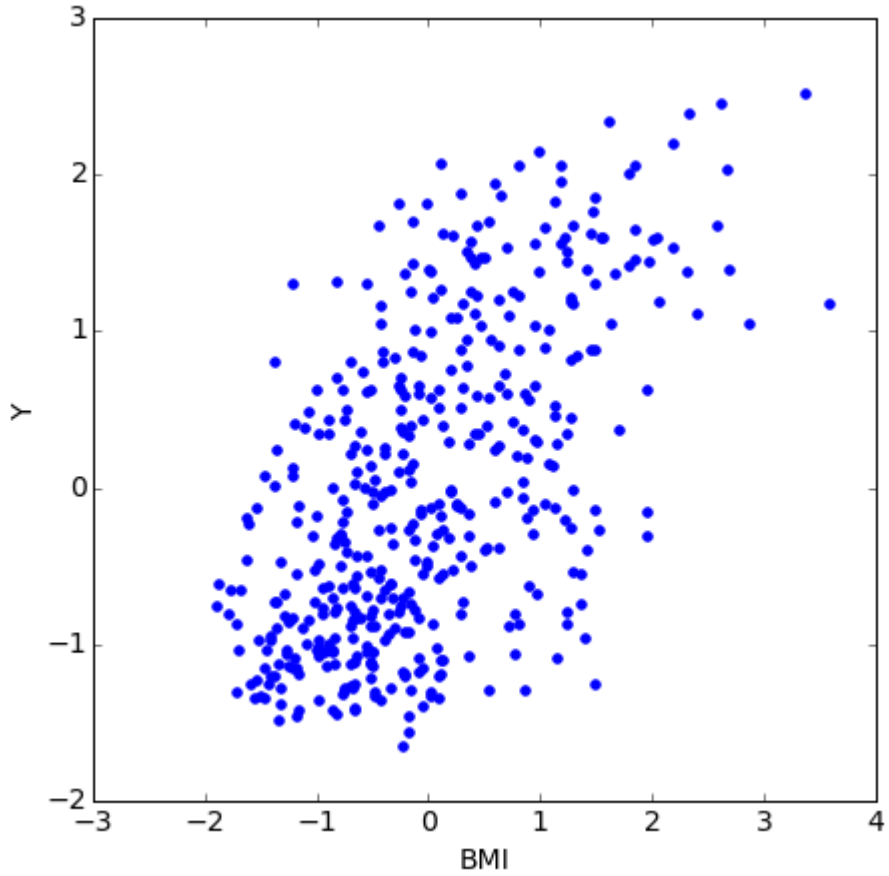
regression



- Can BMI explain Y?
- Can BMI predict Y?
- How does Y vary with BMI?

Regression is a general term for **modeling the relationship** between a **label** (a.k.a. dependent variable or target) and one or more **features** (a.k.a. the attributes, the independent or explanatory variable(s)).

linear regression



We need to make

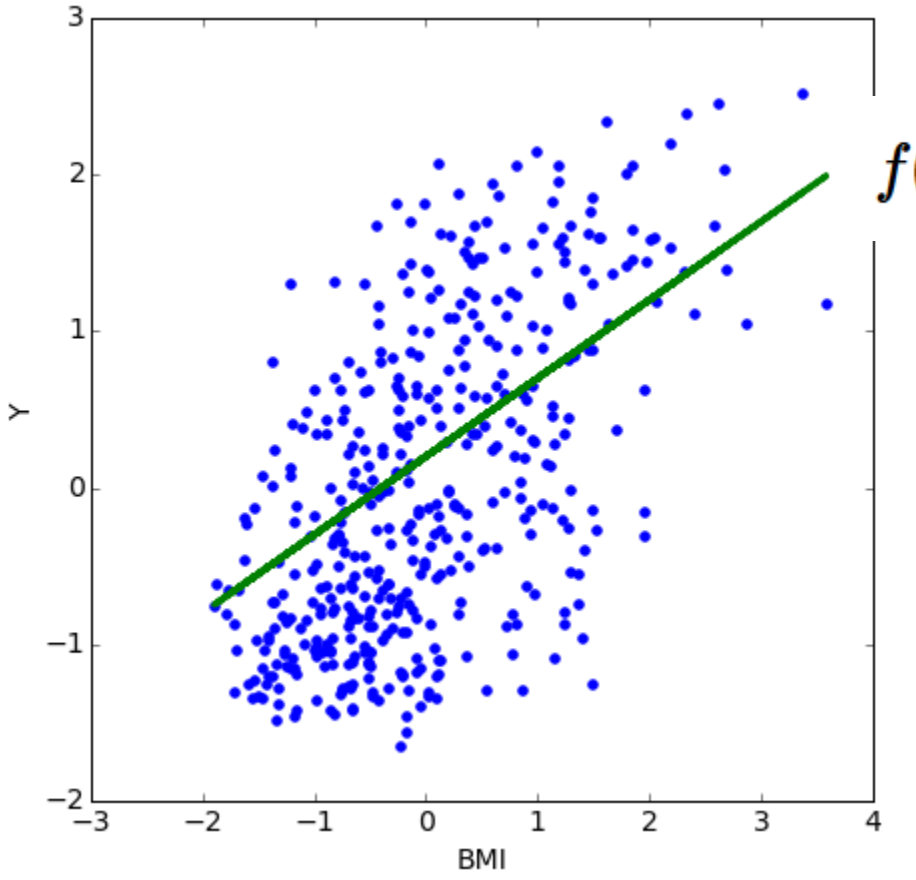
assumptions *linear relationship*

about the

model *linear model*

that generated the data.

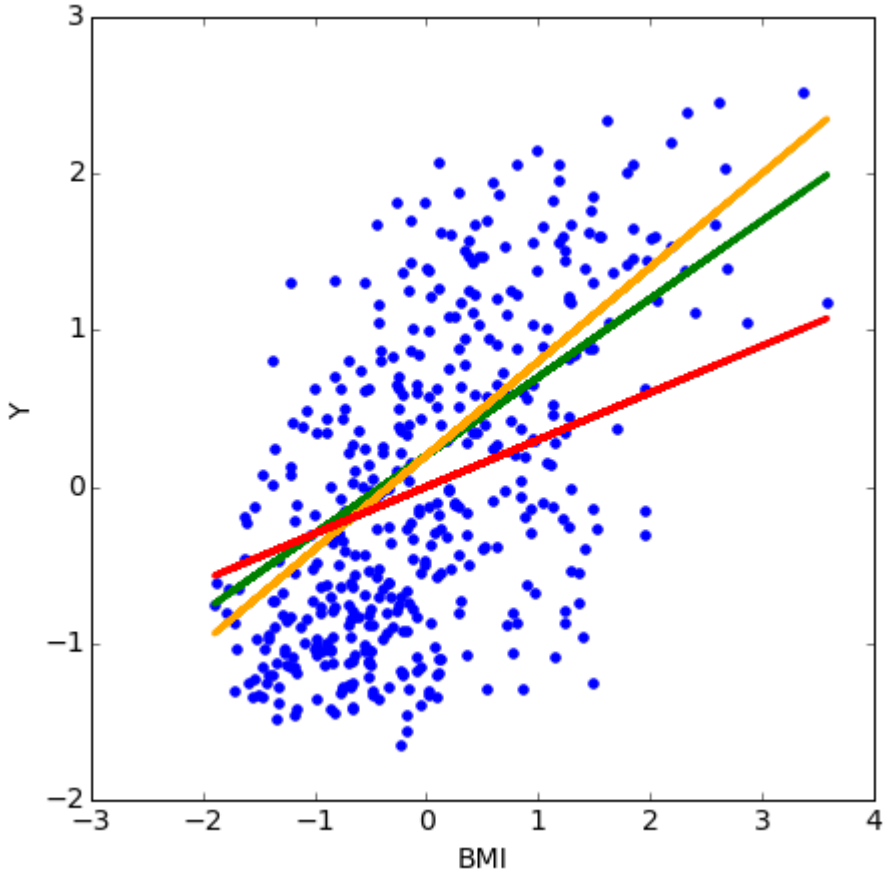
linear regression



$$f(x) = 0.5 \cdot x + 0.2 = ax + b$$

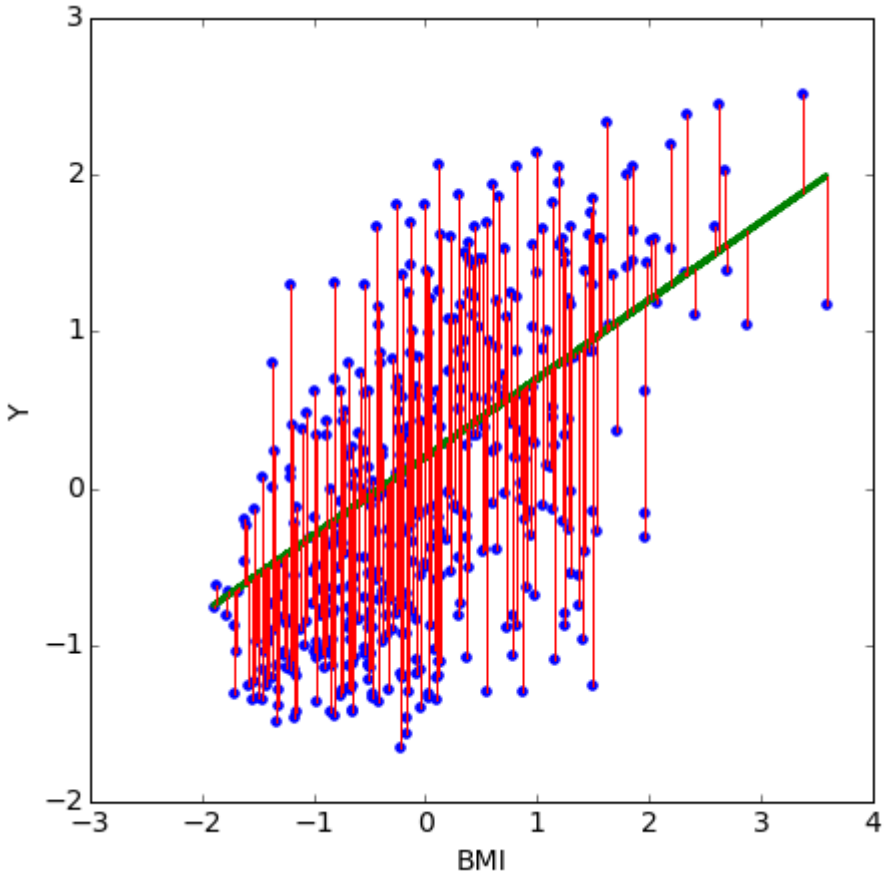
- *a* and *b* are model parameters
- *a* is the slope to the line (direction)
- *b* is the intercept or bias (position)

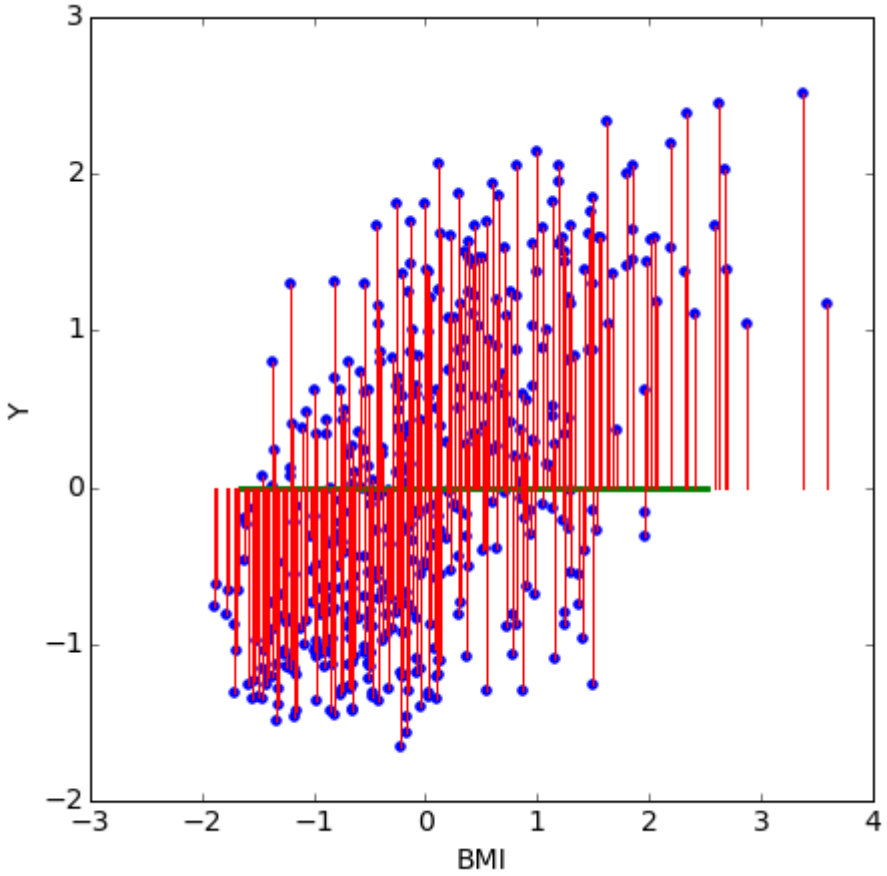
linear regression: fitting



- What values for *a* and *b* fit the data best?
- How can we evaluate them?

linear regression: fitting





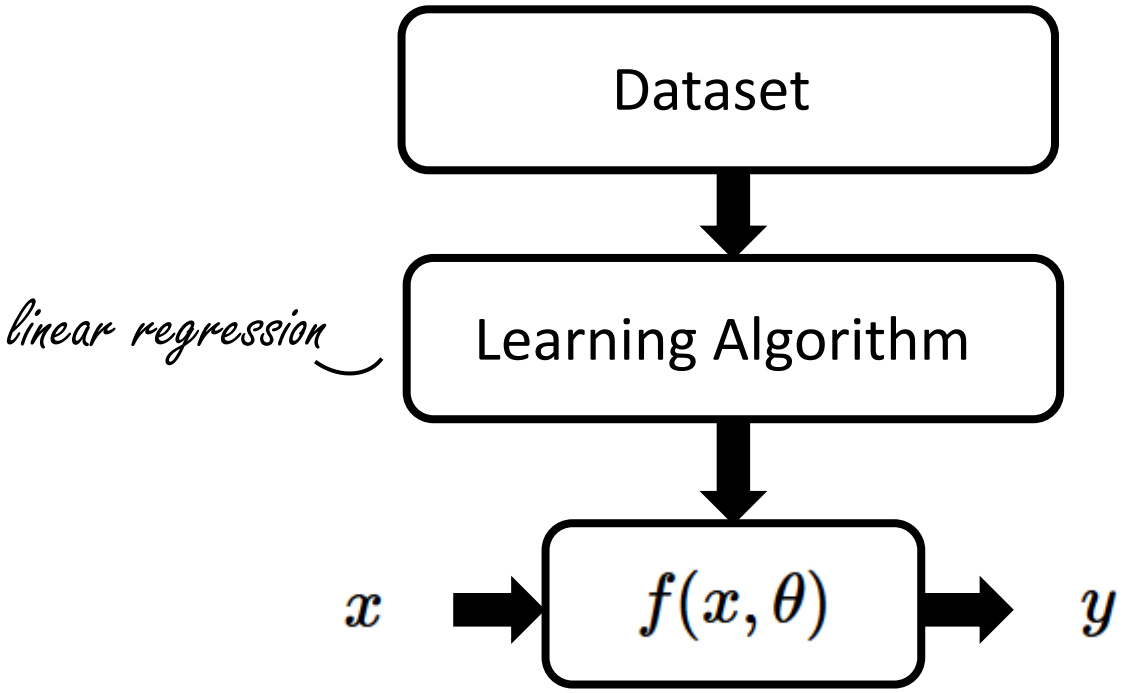
$$R^2 = 1 - \frac{SS_{error}}{SS_{total}}$$

$$R^2 = 0.296$$

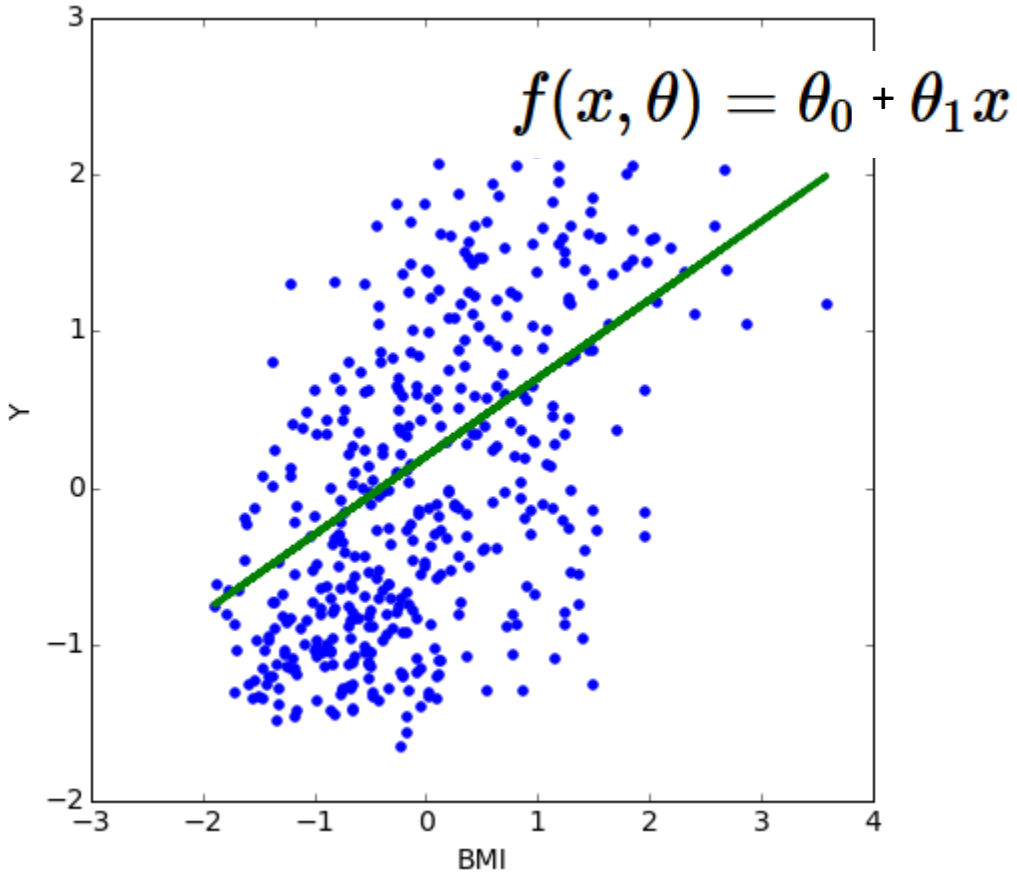
$$SS_{error} = \sum_{i=1}^{n} (y^{(i)} - f(x^{(i)}))^2$$

$$SS_{total} = \sum_{i=1}^{n} (y^{(i)} - \bar{y})^2$$

linear regression



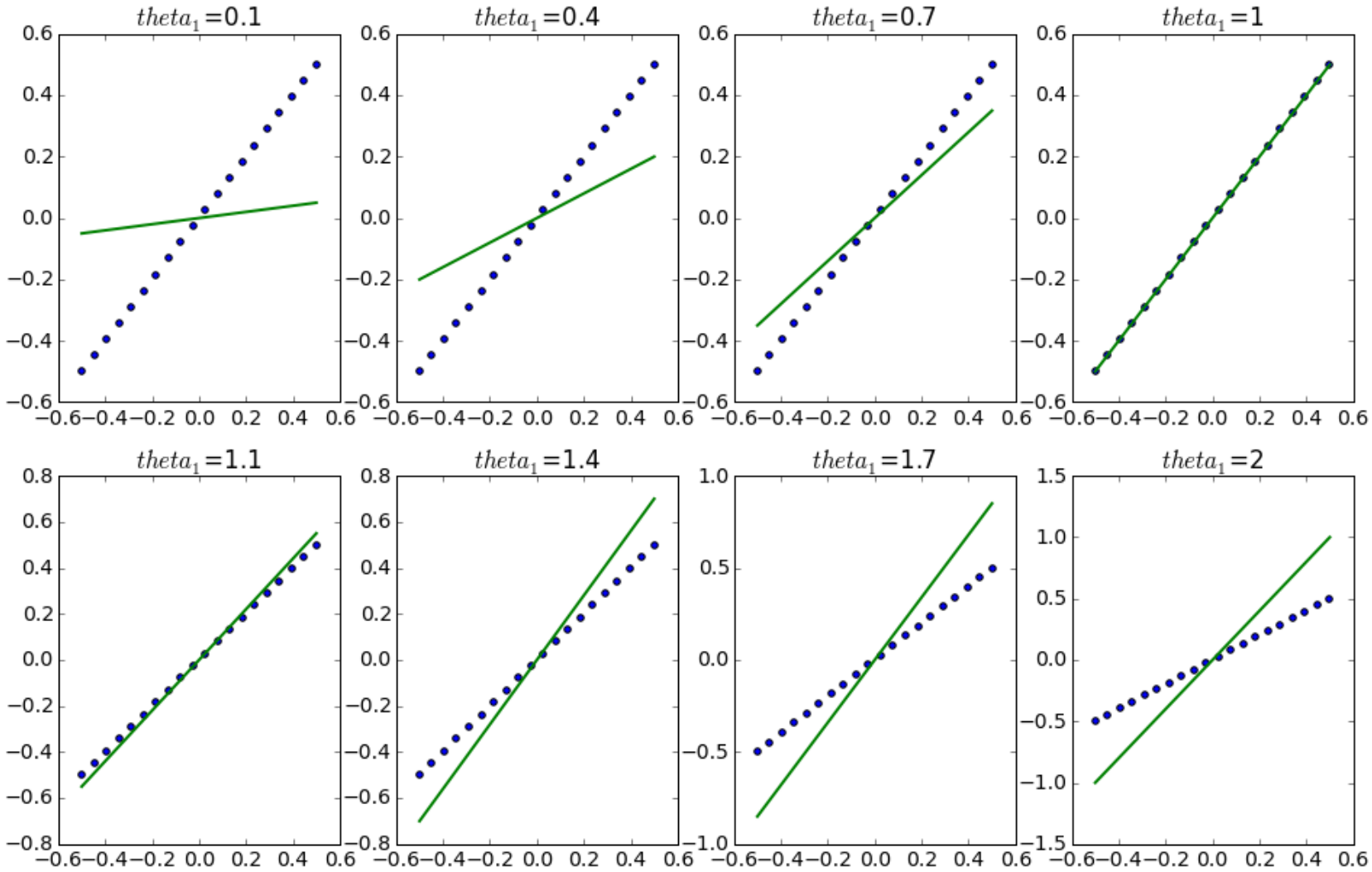
$$f(x,\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m = \theta' x$$



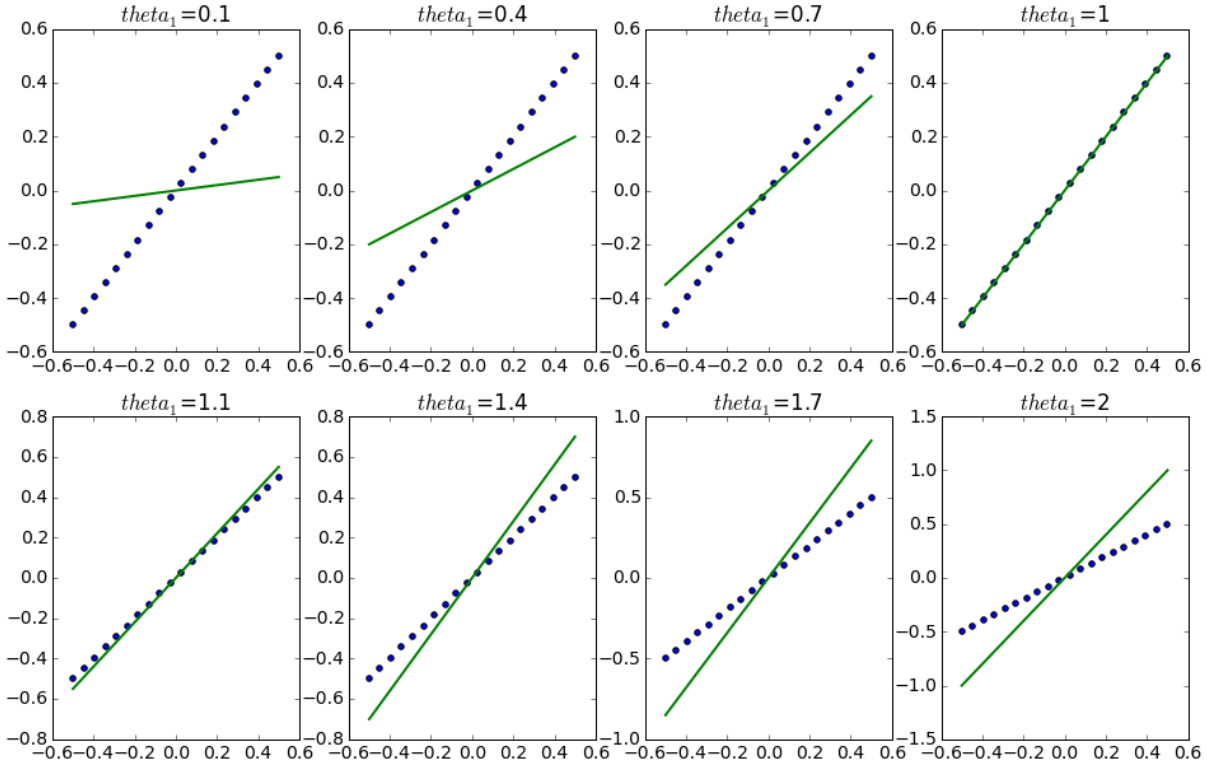
linear regression: cost

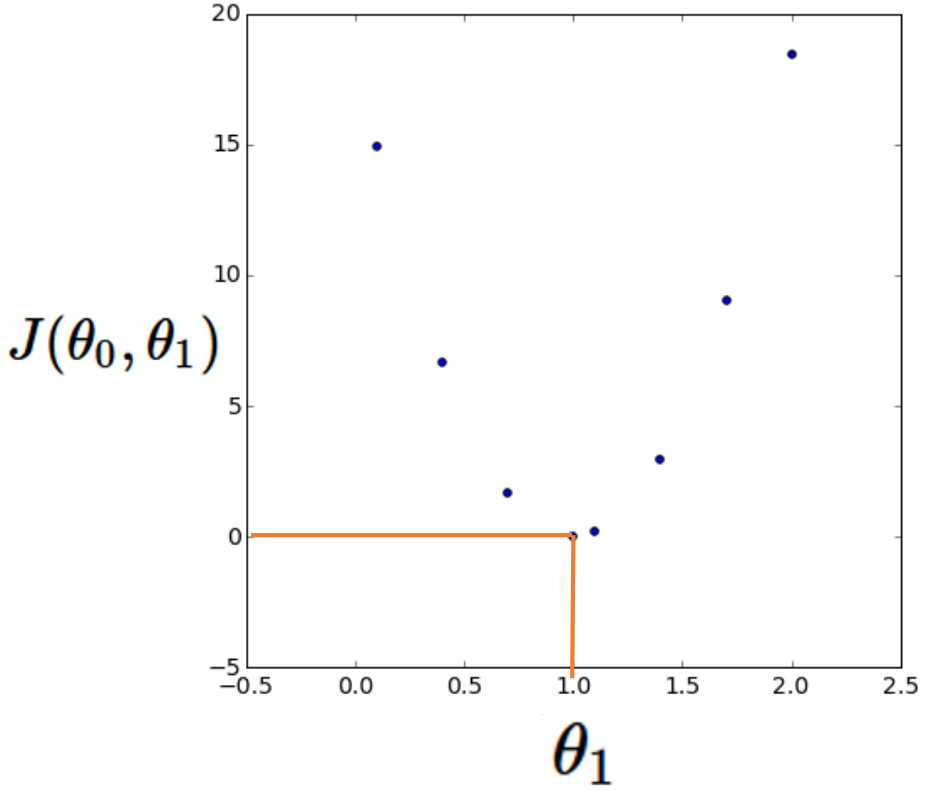
$$f(x) = x$$

$$(\theta_1 = 1, \theta_0 = 0)$$



linear regression: cost





$$J(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} (f(x^{(i)}, \theta) - y^{(i)})^2$$

linear regression

Fit a linear model

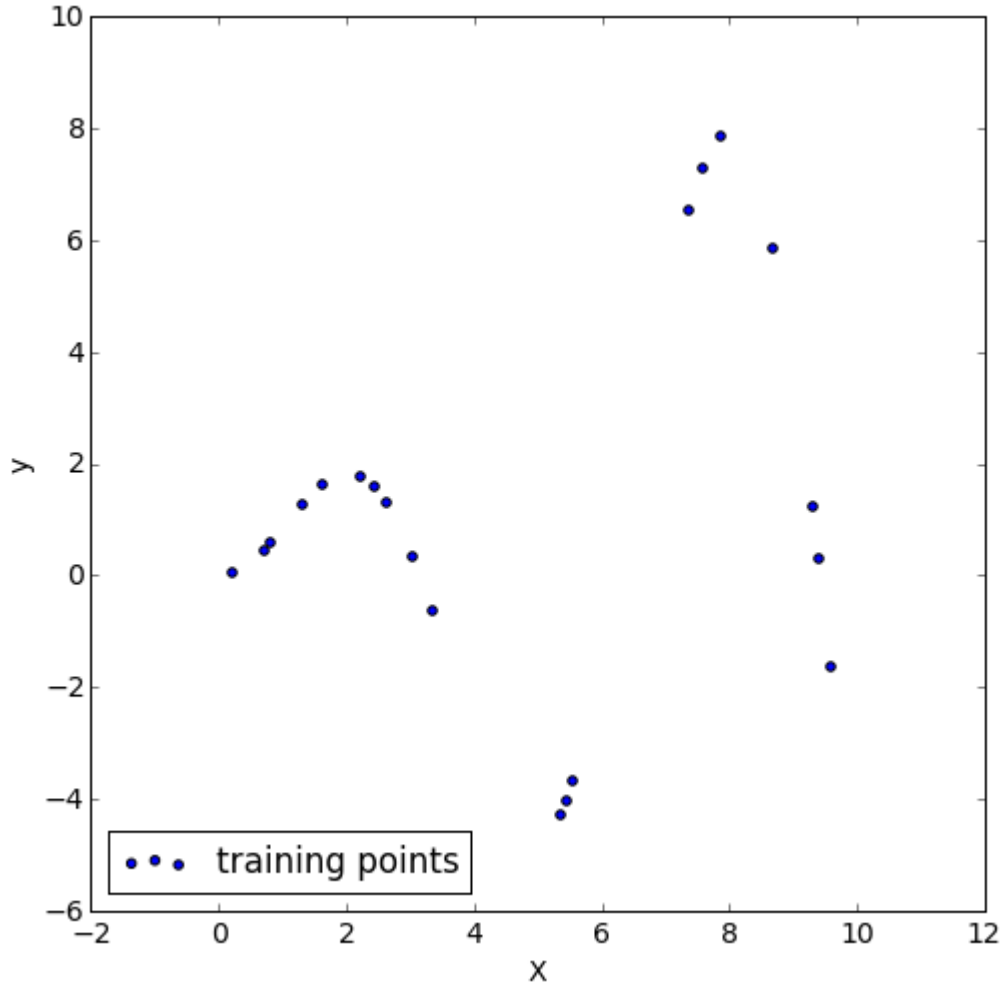
$$f(x, \theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m = \theta' x$$

to the data set such that the cost function

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (f(x^{(i)}, \theta) - y^{(i)})^2$$

is minimal.

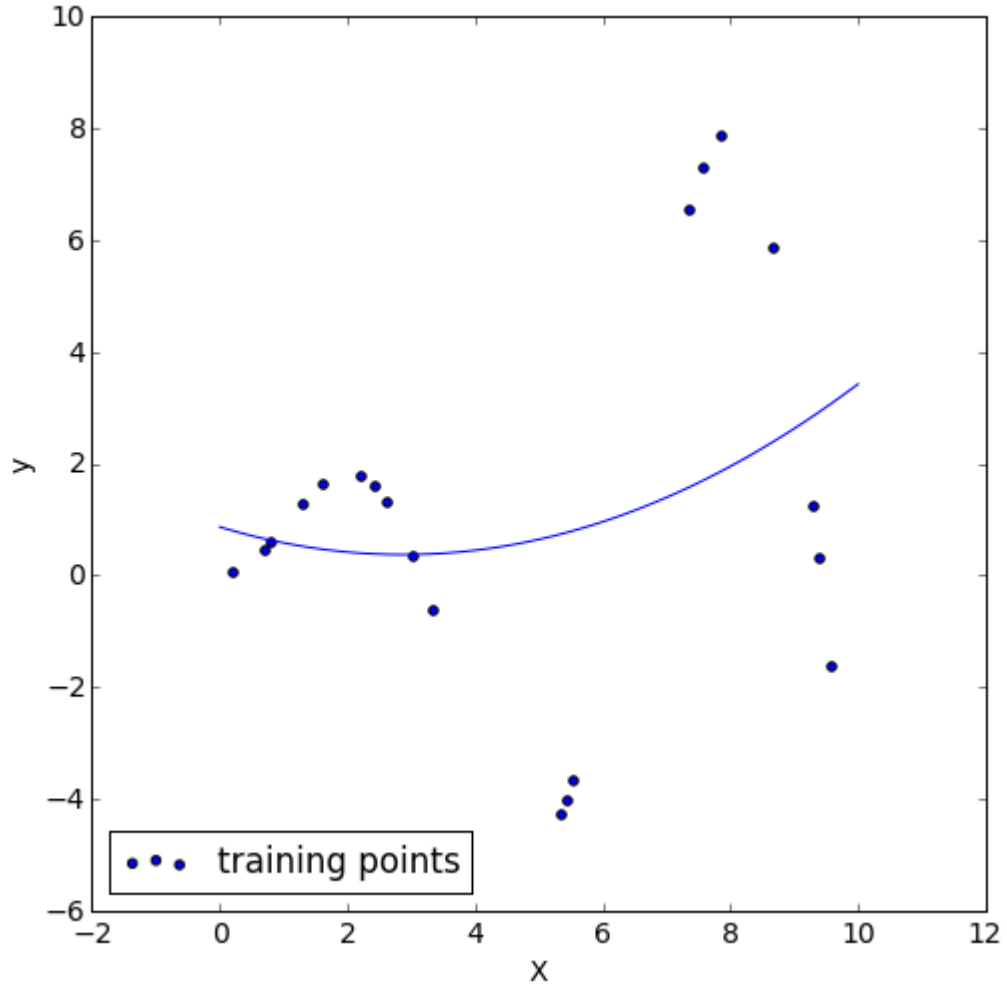
non-linear regression



- Does *y* vary linearly with *X*?
- Can we fit a non-linear model? Yes
- Or we could add polynomial transformations of the features.

$$f(x, \theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_1^2$$

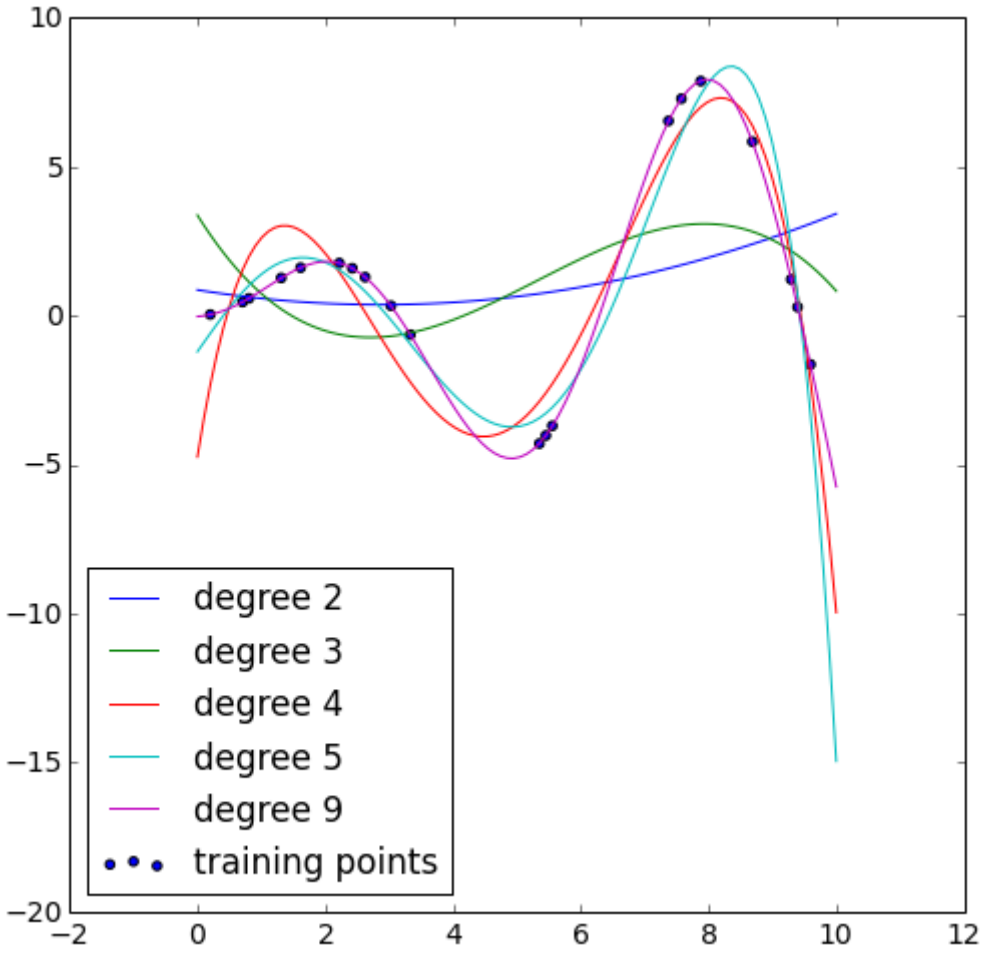
non-linear regression



- Does *y* vary linearly with *X*?
- Can we fit a non-linear model? Yes
- Or we could add polynomial transformations of the features.

$$f(x, \theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_1^2$$

non-linear regression



$$f(x,\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3 + \theta_4 x_1^4 + \theta_5 x_1^5 + \theta_6 x_1^6 + \theta_7 x_1^7 + \theta_8 x_1^8 + \theta_9 x_1^9$$

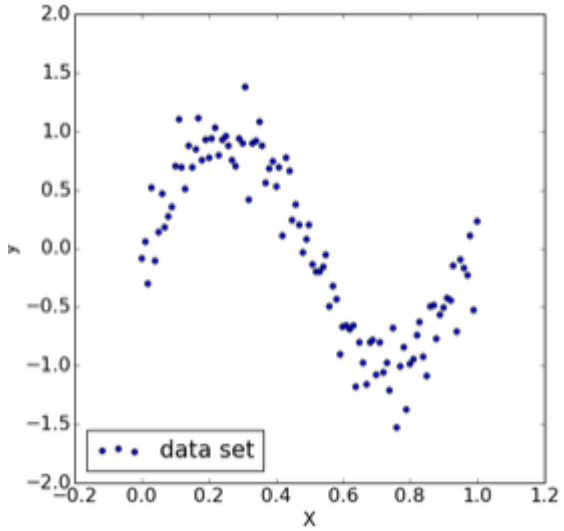
validation

unseen external data

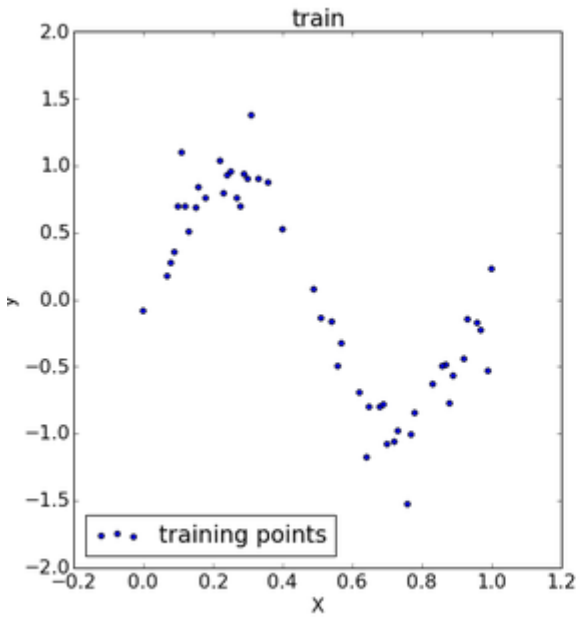
(data not seen during training)

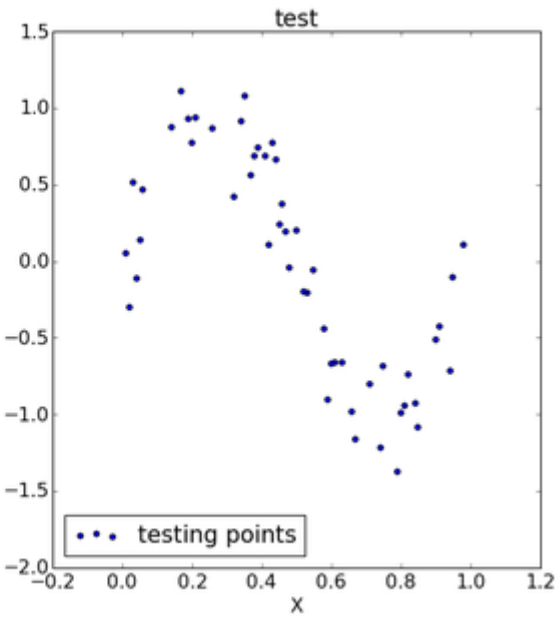
For instance, when we augment the features in a data set by polynomial features of a certain degree *d* we need to set *d* such that it allows for training a model that performs best on all unseen external data.

train-test split

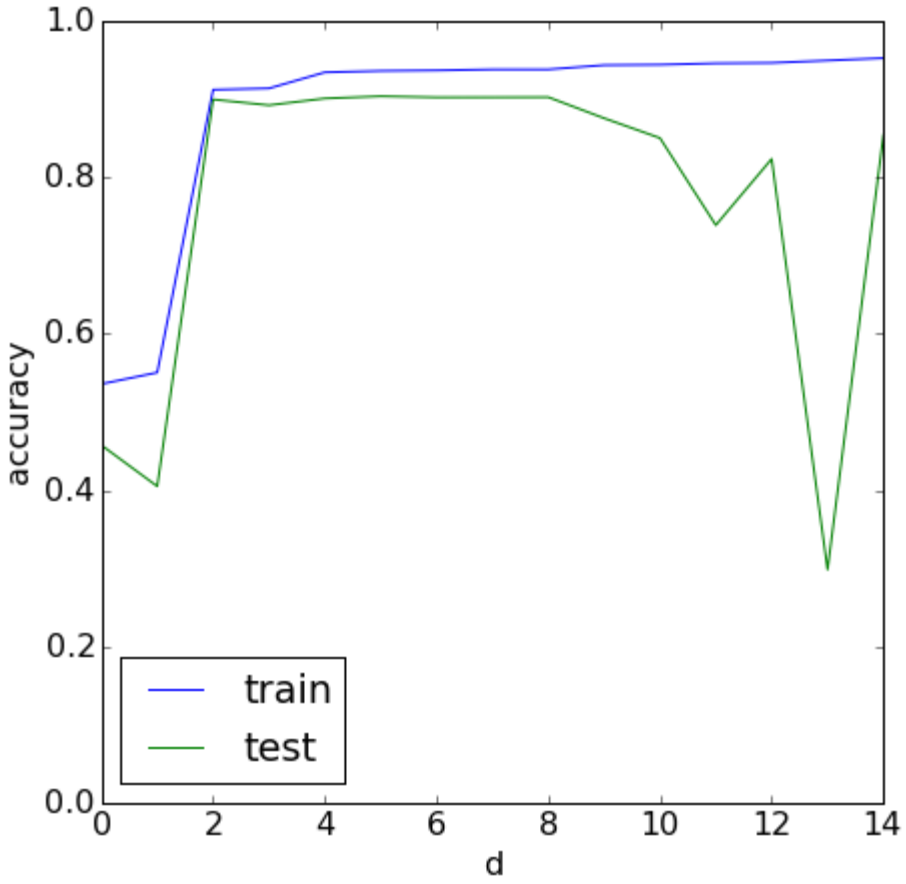


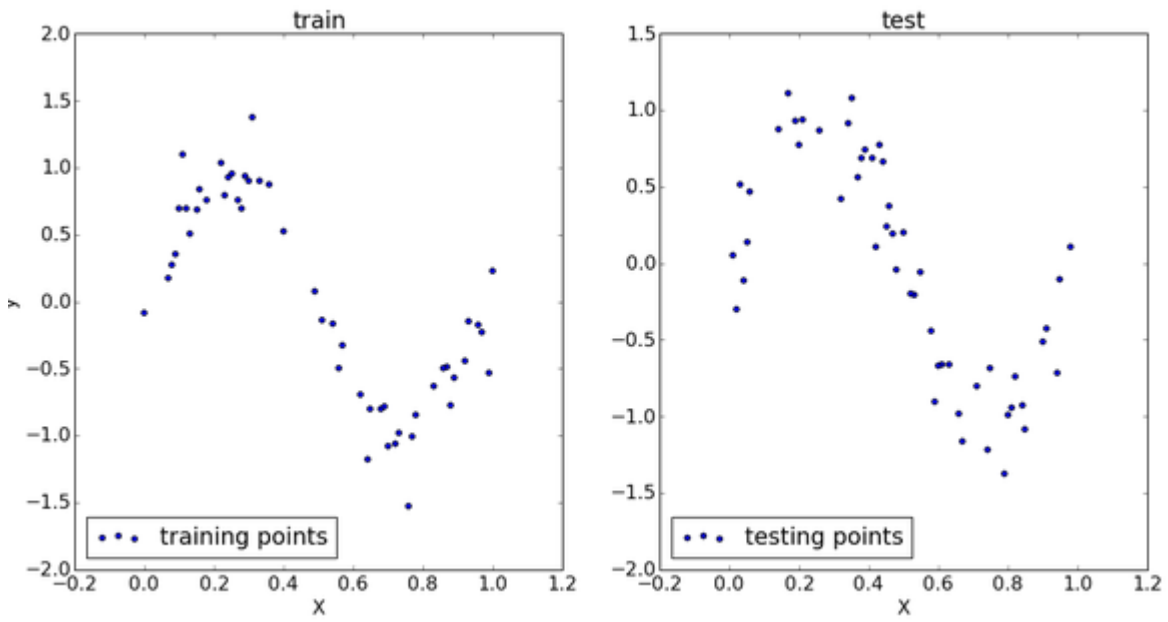
=





model selection and validation



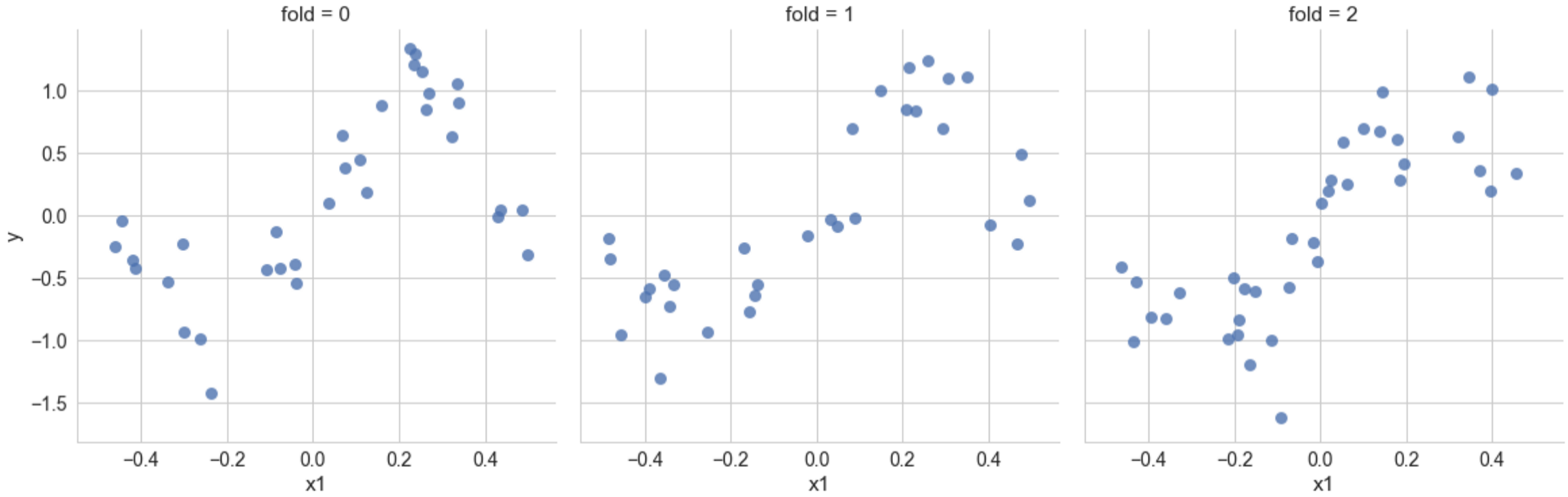


- $d > 8$ overfitting
- $d = 1$ underfitting
- generalization performance?
- validation set

k-fold cross-validation (CV)

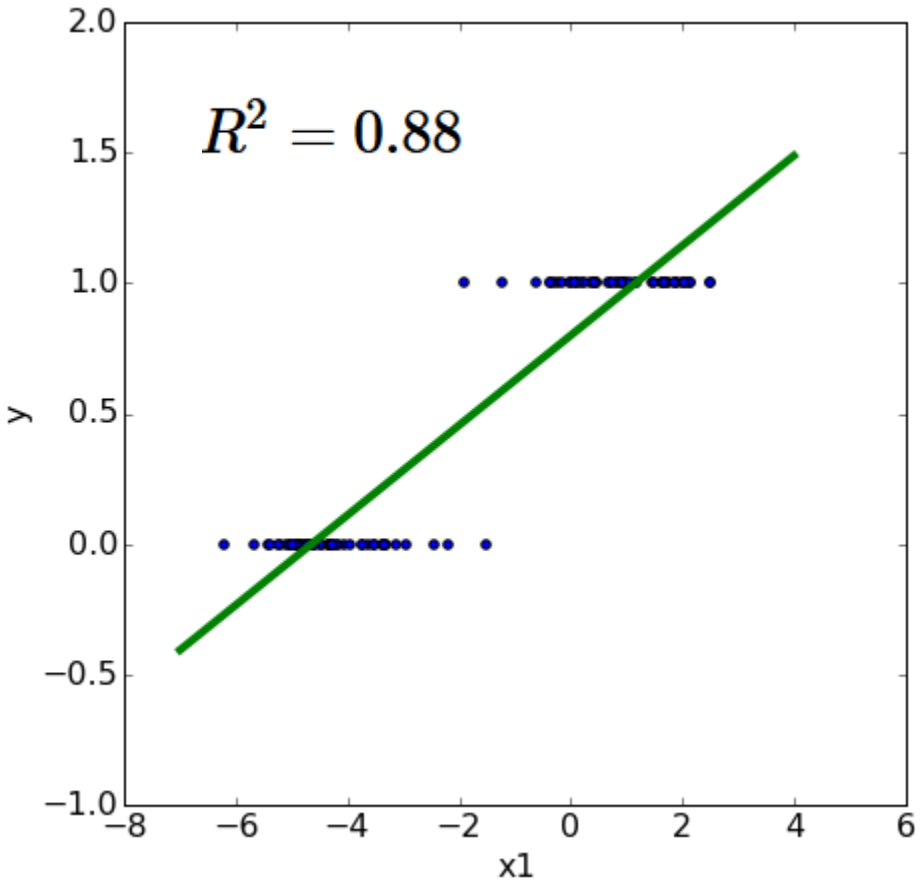
- fewer data points for training
- performance results can depend strongly on a particular random choice of the data set splits
- *k*-fold CV:
  - partition data set into *k* smaller sets (folds)
  - For each fold D_i $(i = 1 \ldots k)$ do
    1. train a model m_i on the data points in folds D_j with $j \neq i$,
    2. use m_i to compute predictions for D_i.

3-fold cross-validation (CV)



classification

logistic regression



We need to make

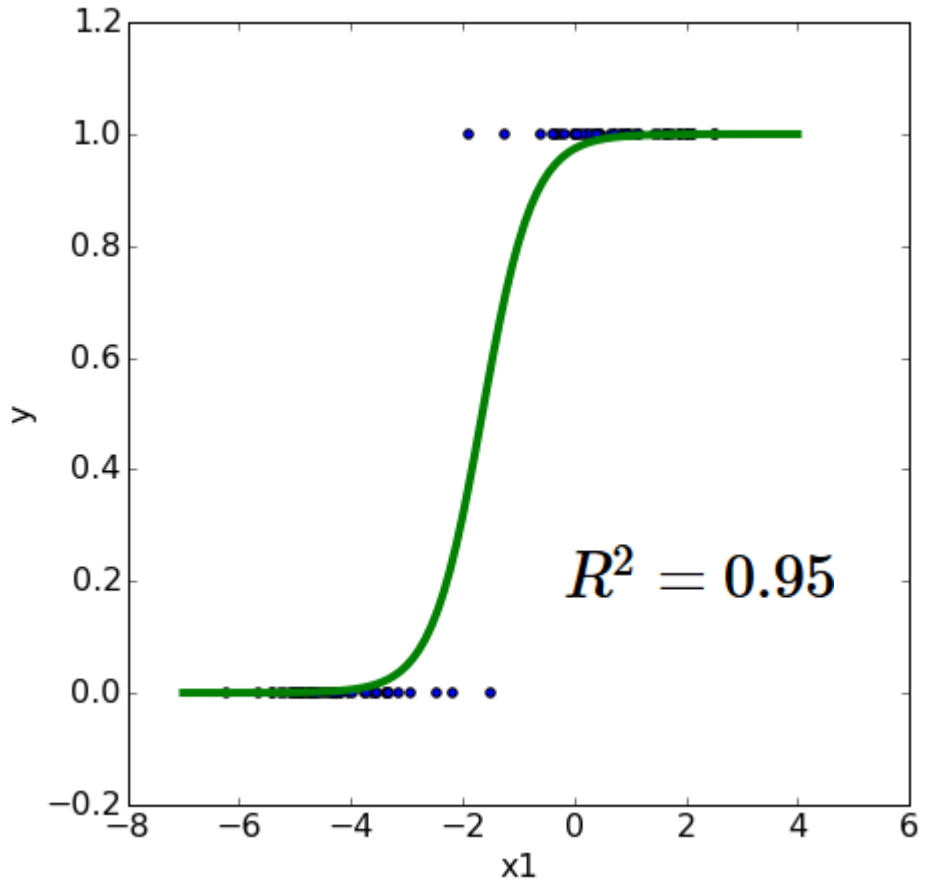
assumptions *linear relationship*

about the

model *linear model*

that generated the data.

logistic regression



We need to make

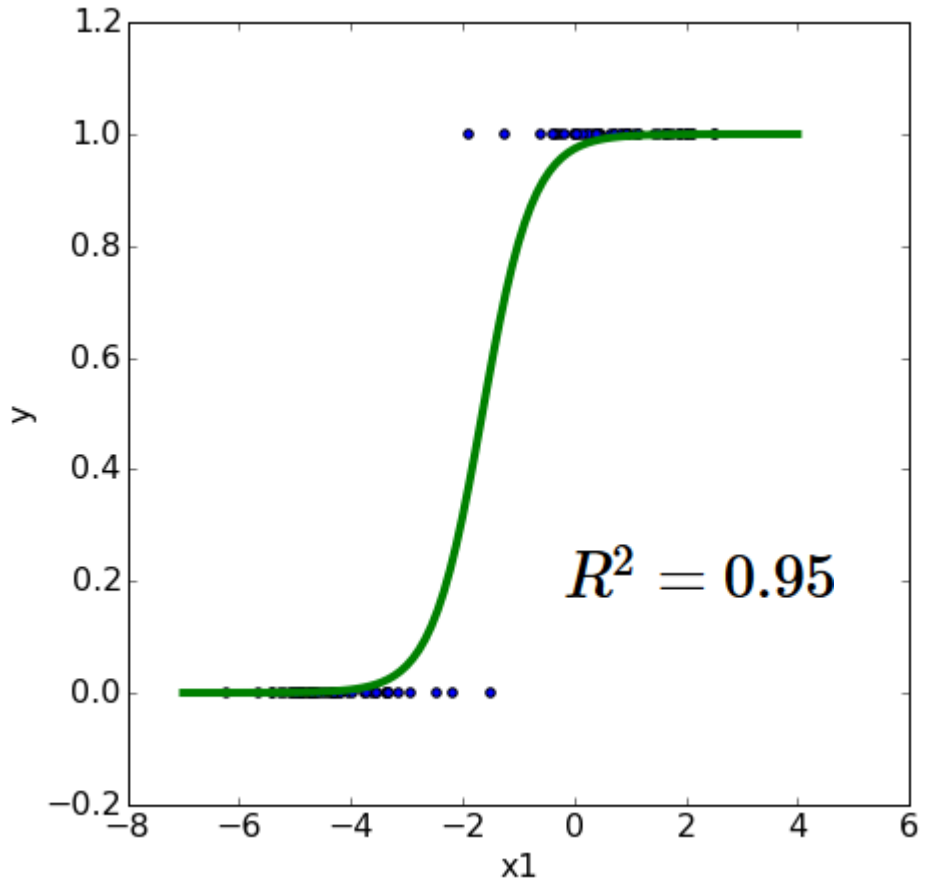
assumptions *linearly separable*

about the

model *logistic model*

that generated the data.

logistic regression: logistic model



$$f(x, \theta) = g(\theta_0 + \theta_1 x_1)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

logistic regression: cost function

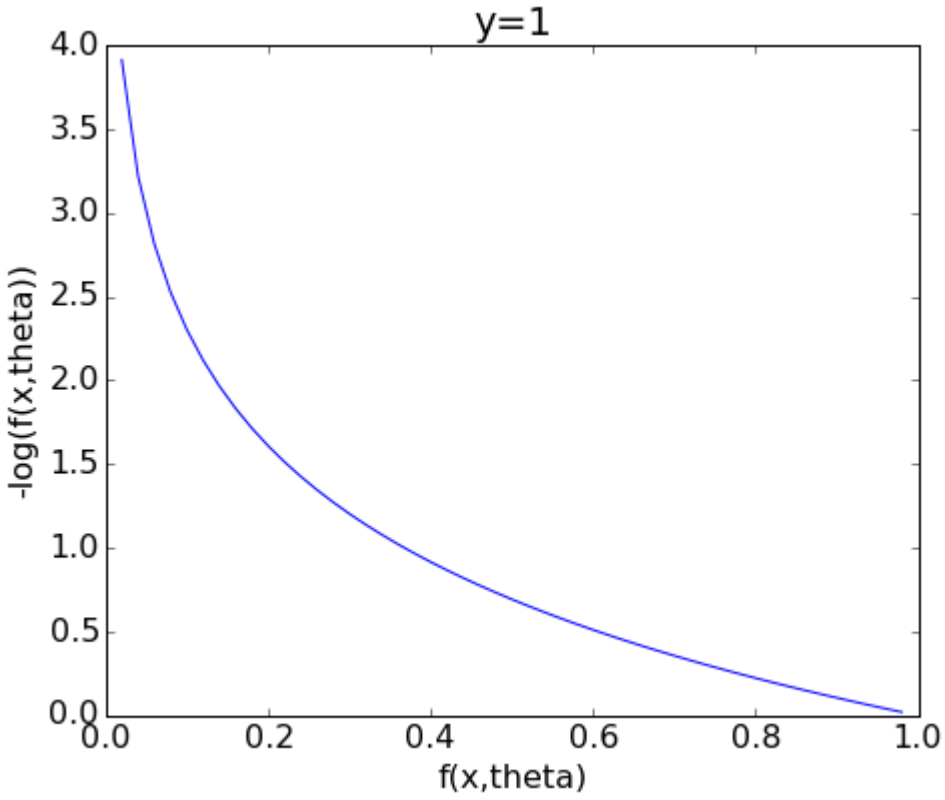
$$J(\theta) = -[\frac{1}{n}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)}, \theta)) + (1 - y^{(i)}) log(1 - f(x^{(i)}, \theta))]$$

We know that $y^{(i)}$ is either 0 or 1. If $y^{(i)} = 1$ then the cost function $J(\theta)$ is incremented by $-log(f(x^{(i)}, \theta))$.

Similarly, if $y^{(i)} = 0$ then the cost function $J(\theta)$ is incremented by $-log(1 - f(x^{(i)}, \theta))$.

logistic regression: cost function

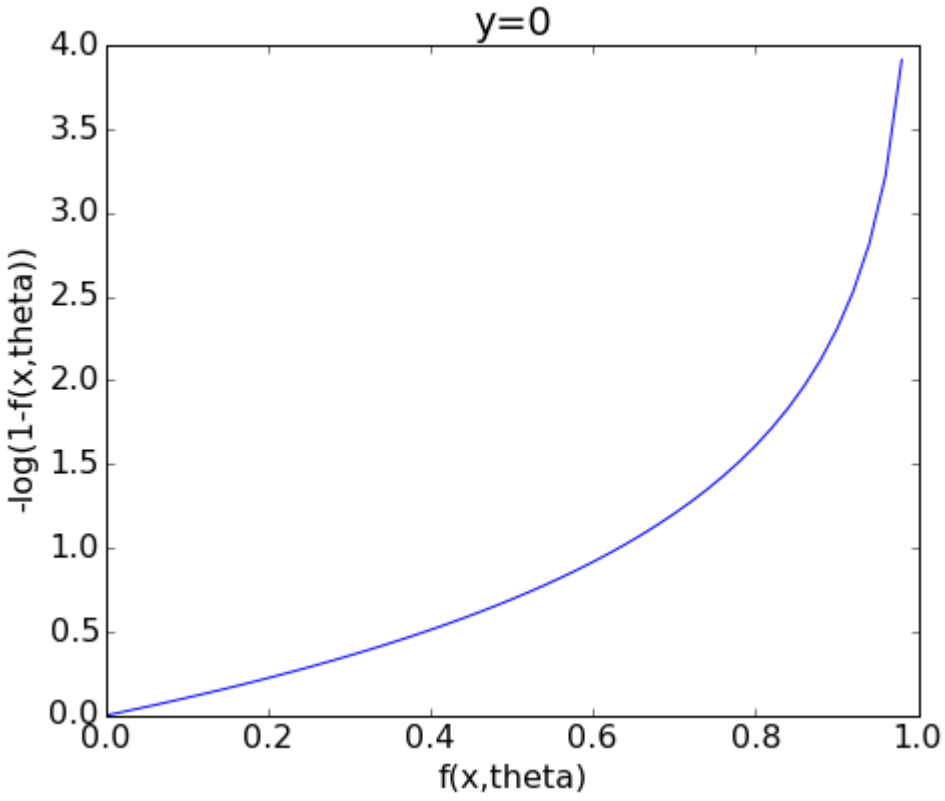
We know that $y^{(i)}$ is either 0 or 1. If $y^{(i)} = 1$ then the cost function $J(\theta)$ is incremented by $-log(f(x^{(i)}, \theta))$.



logistic regression: cost function

Similarly, if $y^{(i)} = 0$ then the cost function $J(\theta)$ is incremented by

$$-log(1 - f(x^{(i)}, \theta)).$$



logistic regression

Fit a logistic model

$$f(x,\theta) = g(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m) = g(\theta' x)$$

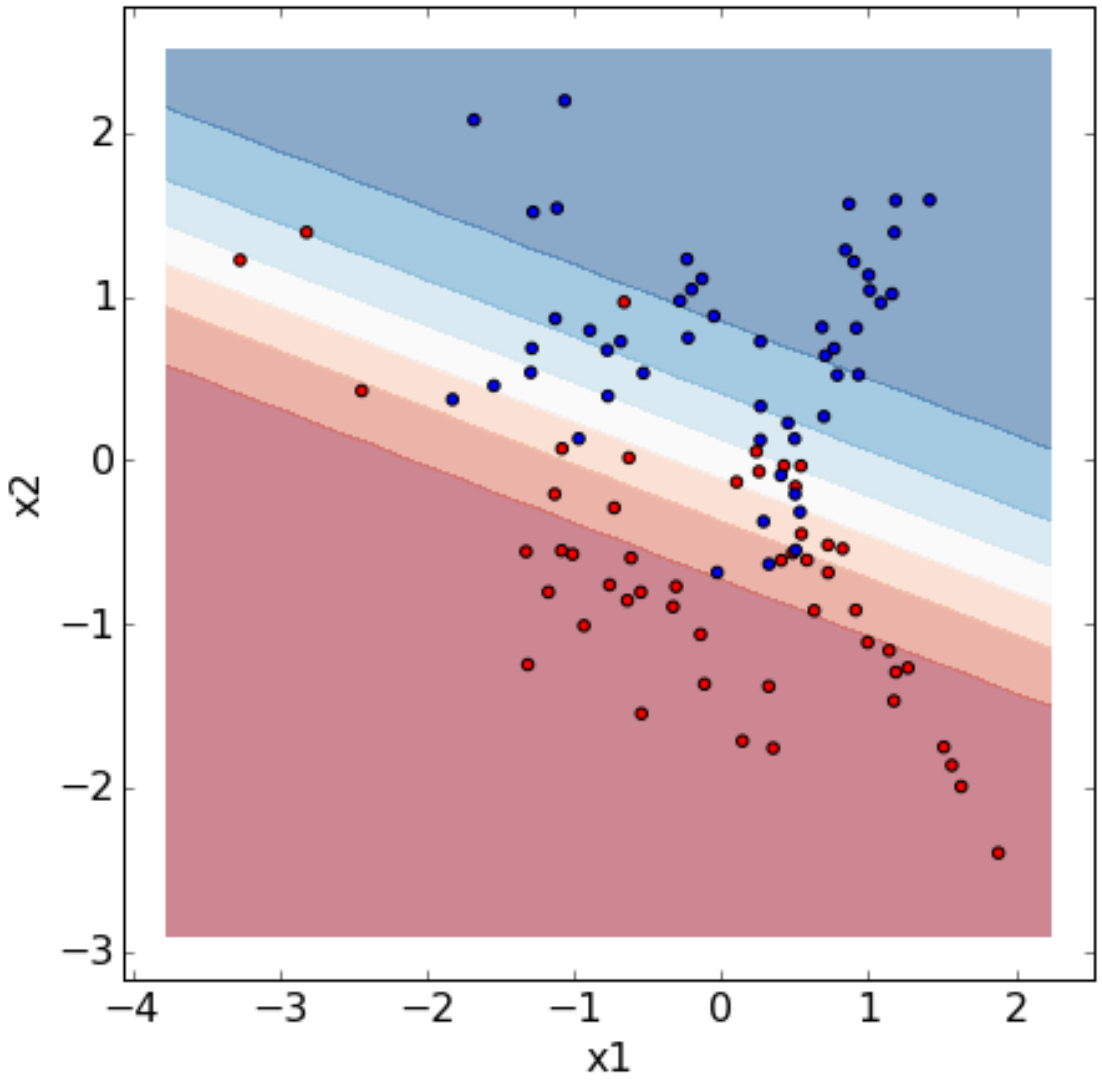
to the data set such that the cost function

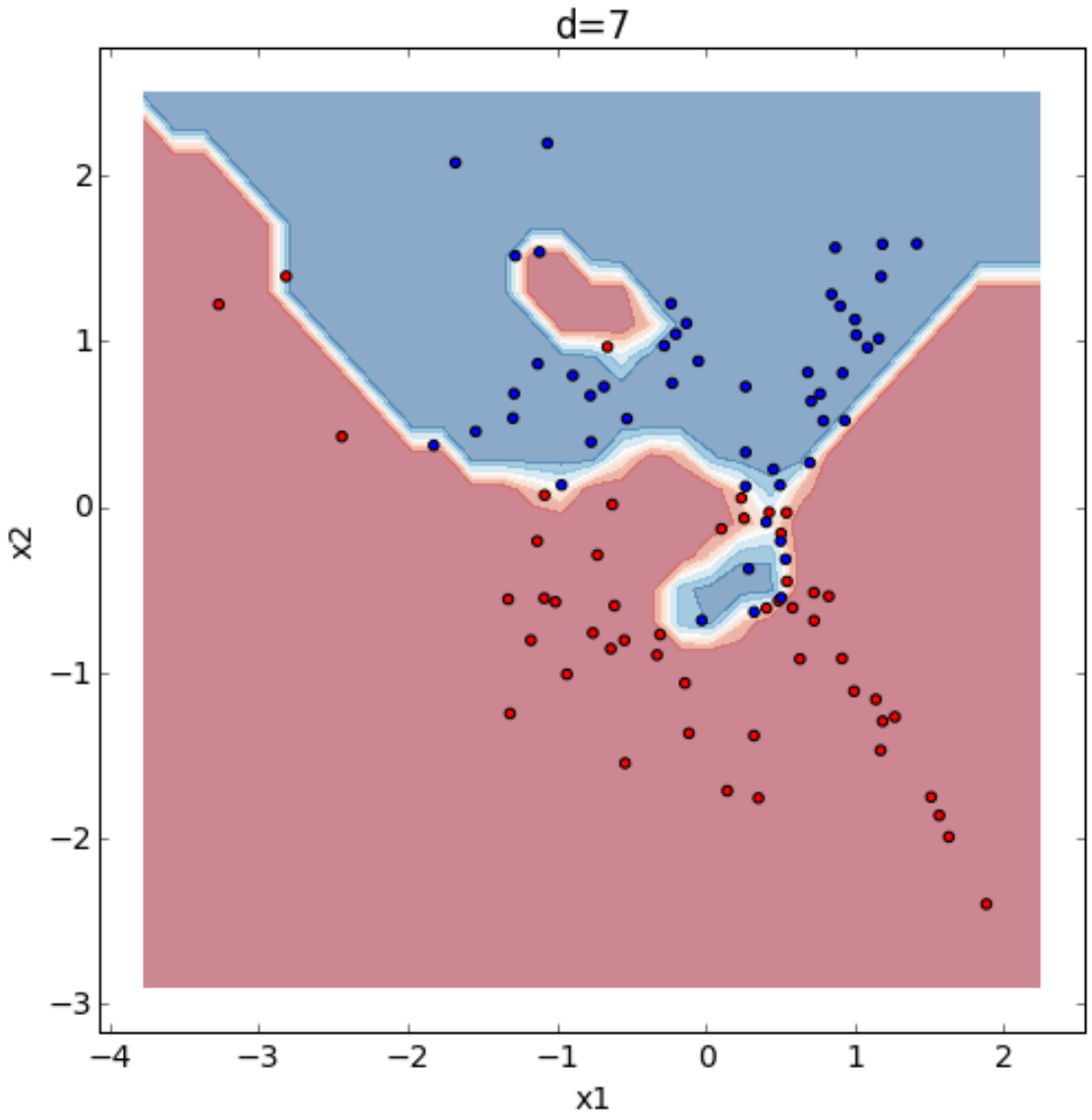
$$J(\theta) = -[\frac{1}{n}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)},\theta)) + (1 - y^{(i)}) log(1 - f(x^{(i)},\theta))]$$

is minimal using gradient descent

$$\theta_j := \theta_j - \alpha \frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})x_j^{(i)}$$

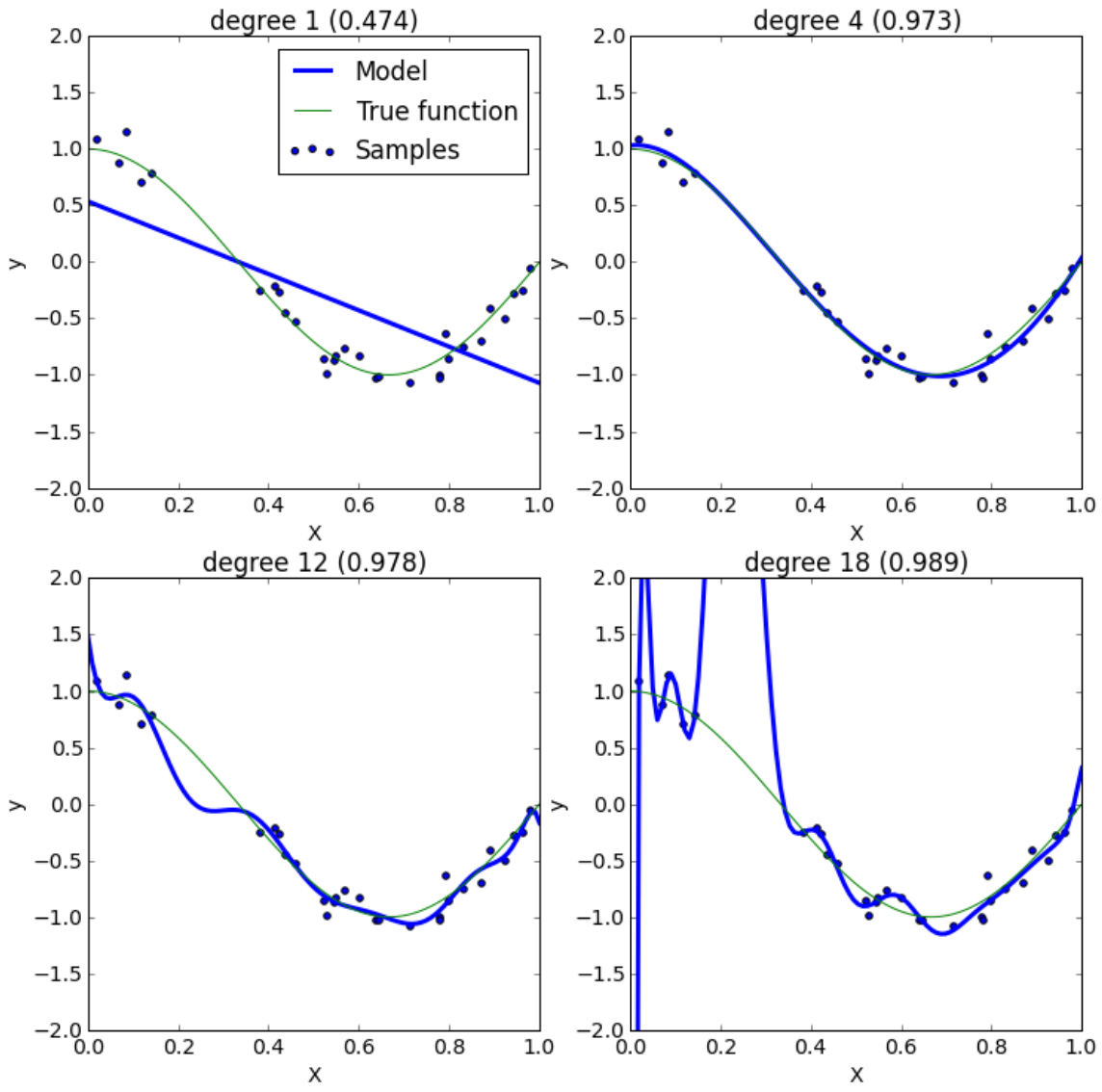
logistic regression





regularization

regularization



- noise
  - label
  - feature
- feature relevance
- relatively small train sets
- don't try to fit the data perfectly
- don't try to use all features
- notice how R^2 on the train set does increase with *d*

regularized linear regression

$$f(x,\theta) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m$$

cost function

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})^2$$

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})^2 + \lambda\sum_{j=1}^{m}\theta_j^2$$

regularized cost function

$$\theta_0 := \theta_0 - \alpha\frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\frac{1}{n}\sum_{i=1}^{n}(f(x^{(i)},\theta) - y^{(i)})x_j^{(i)} - \frac{\lambda}{n}\theta_j$$

regularized logistic regression

$$f(x,\theta) = g(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m)$$

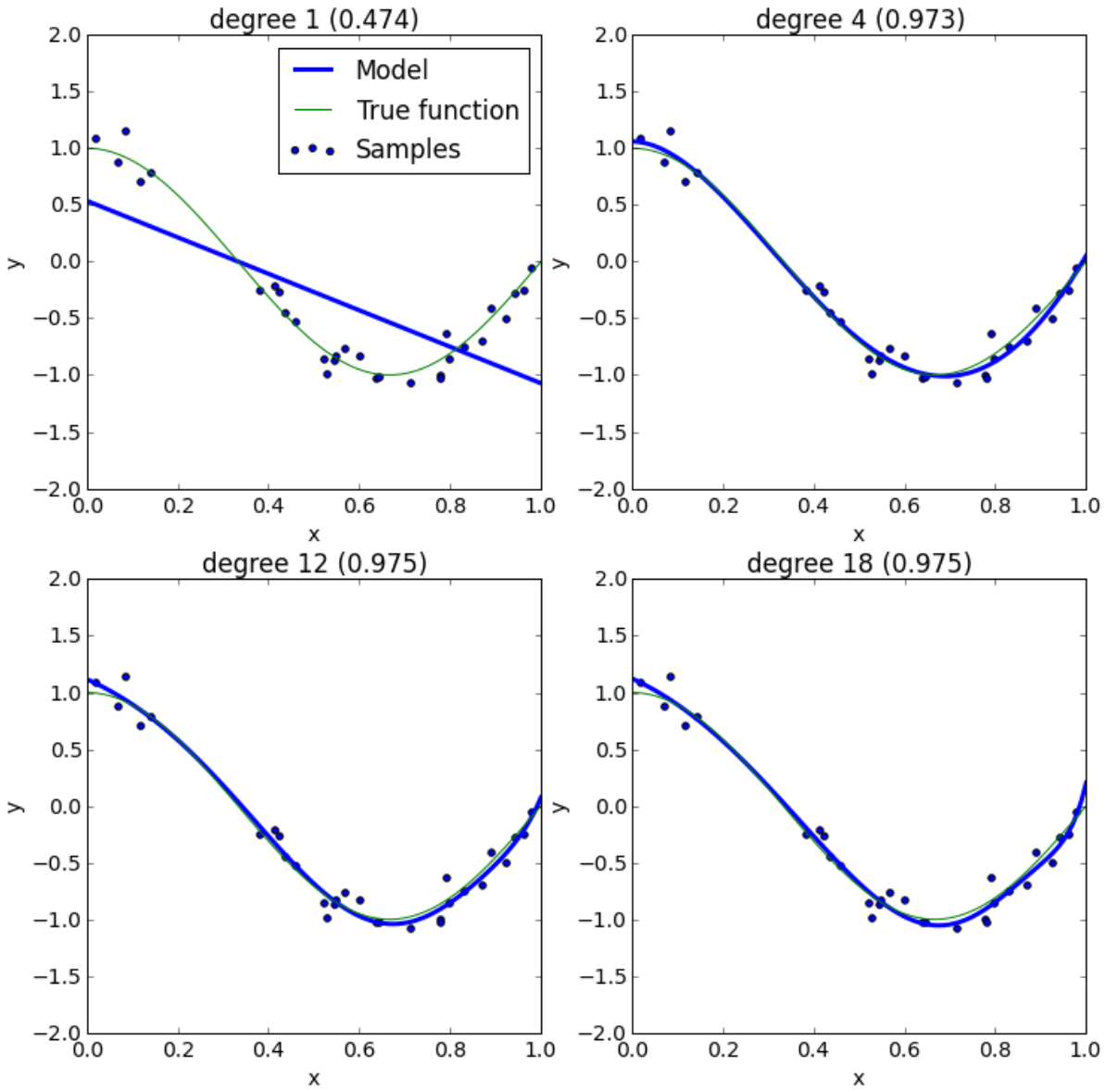
cost function

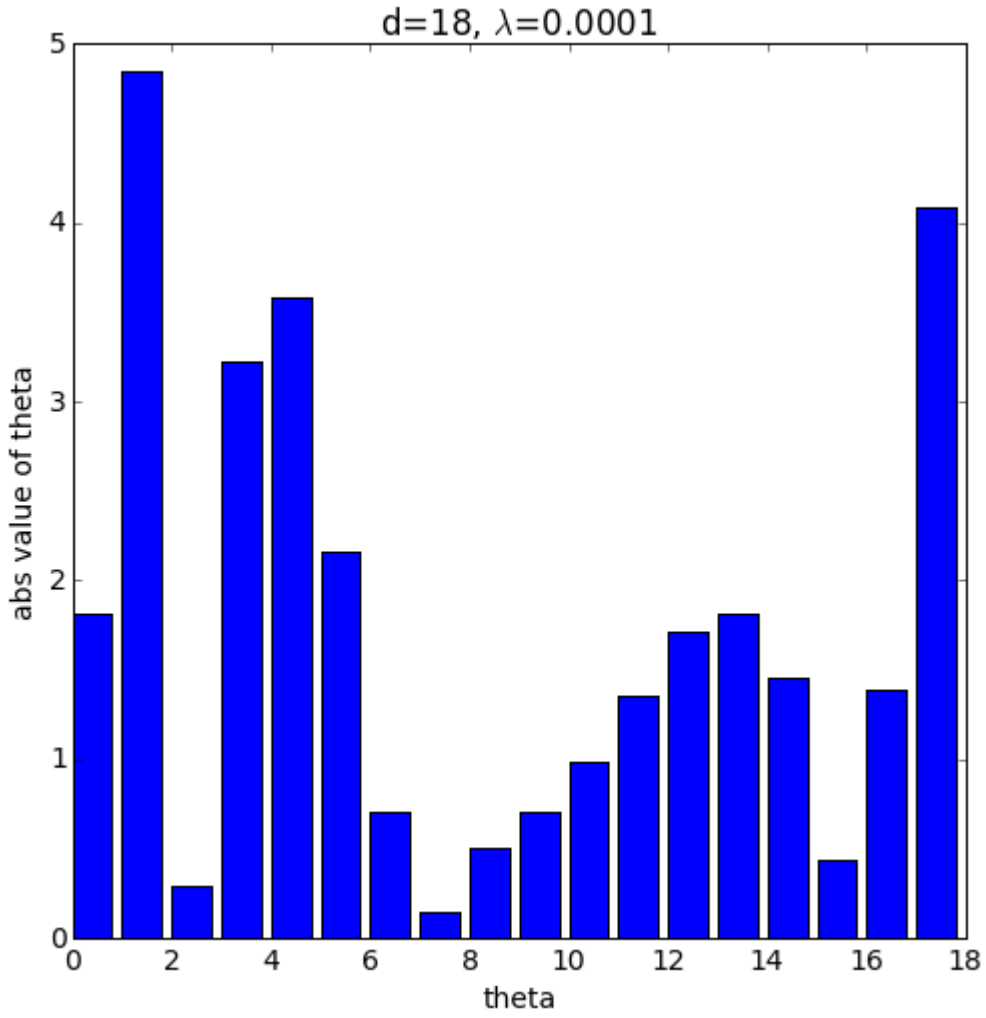
$$J(\theta) = -[\frac{1}{n}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)},\theta)) + (1-y^{(i)}) log(1-f(x^{(i)},\theta))]$$

$$J(\theta) = -[\frac{1}{n}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)},\theta)) + (1-y^{(i)}) log(1-f(x^{(i)},\theta))] + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

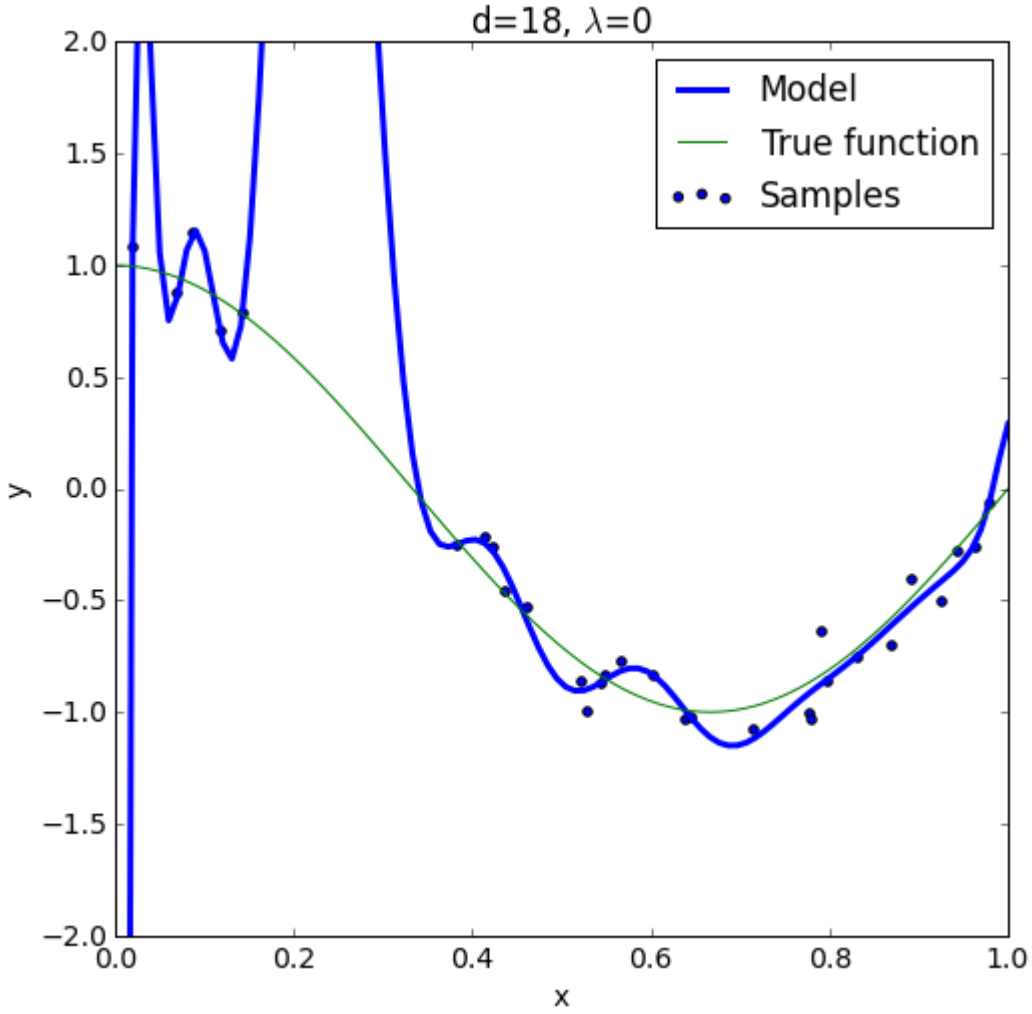
regularized cost function

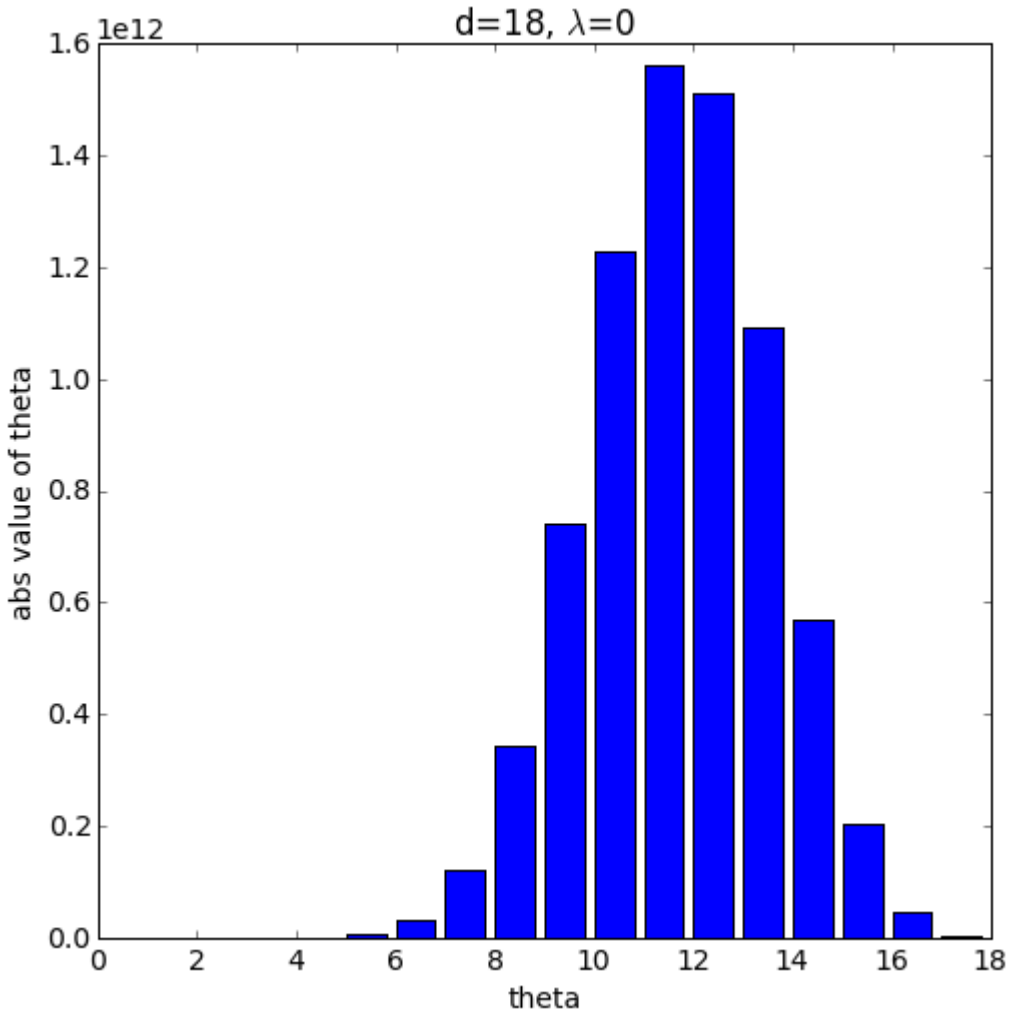
regularized linear regression



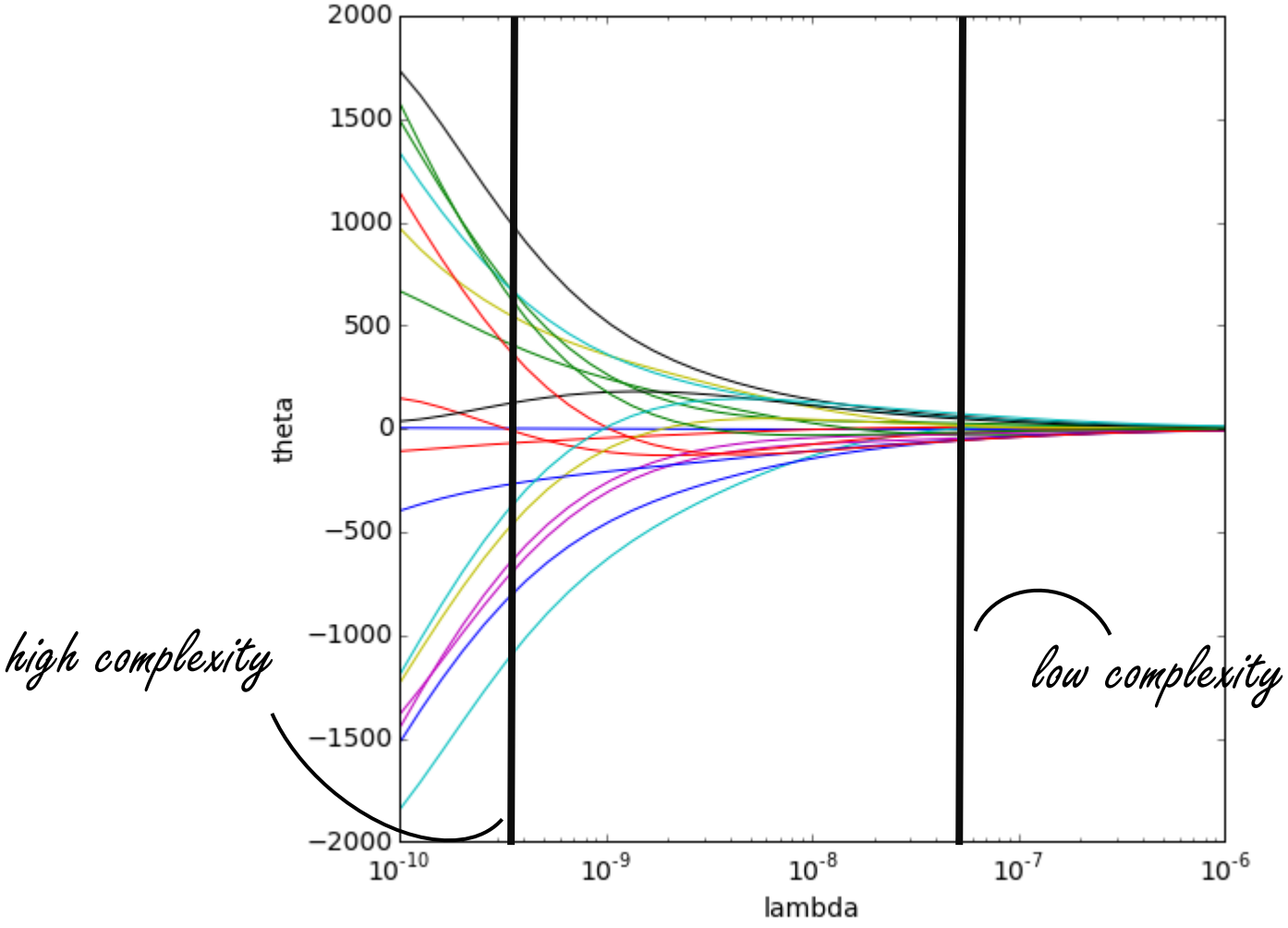


regularized linear regression

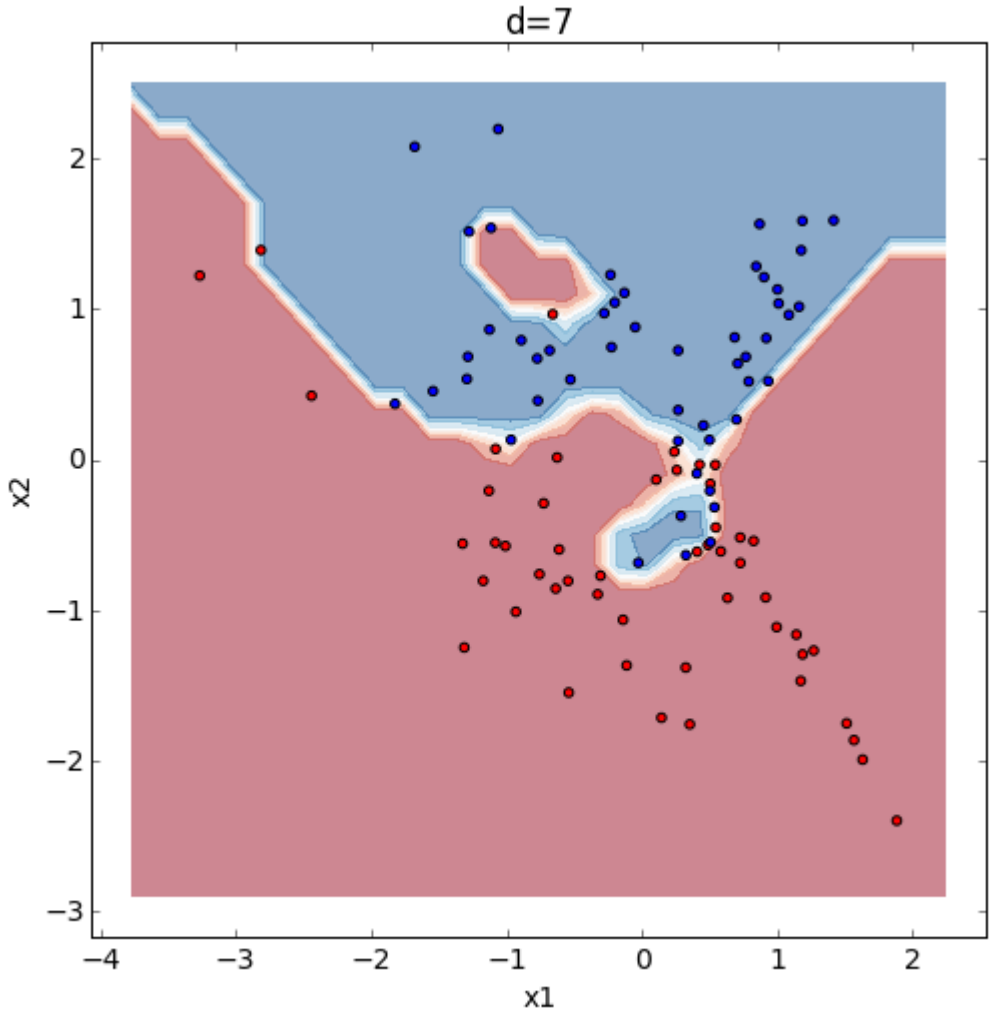


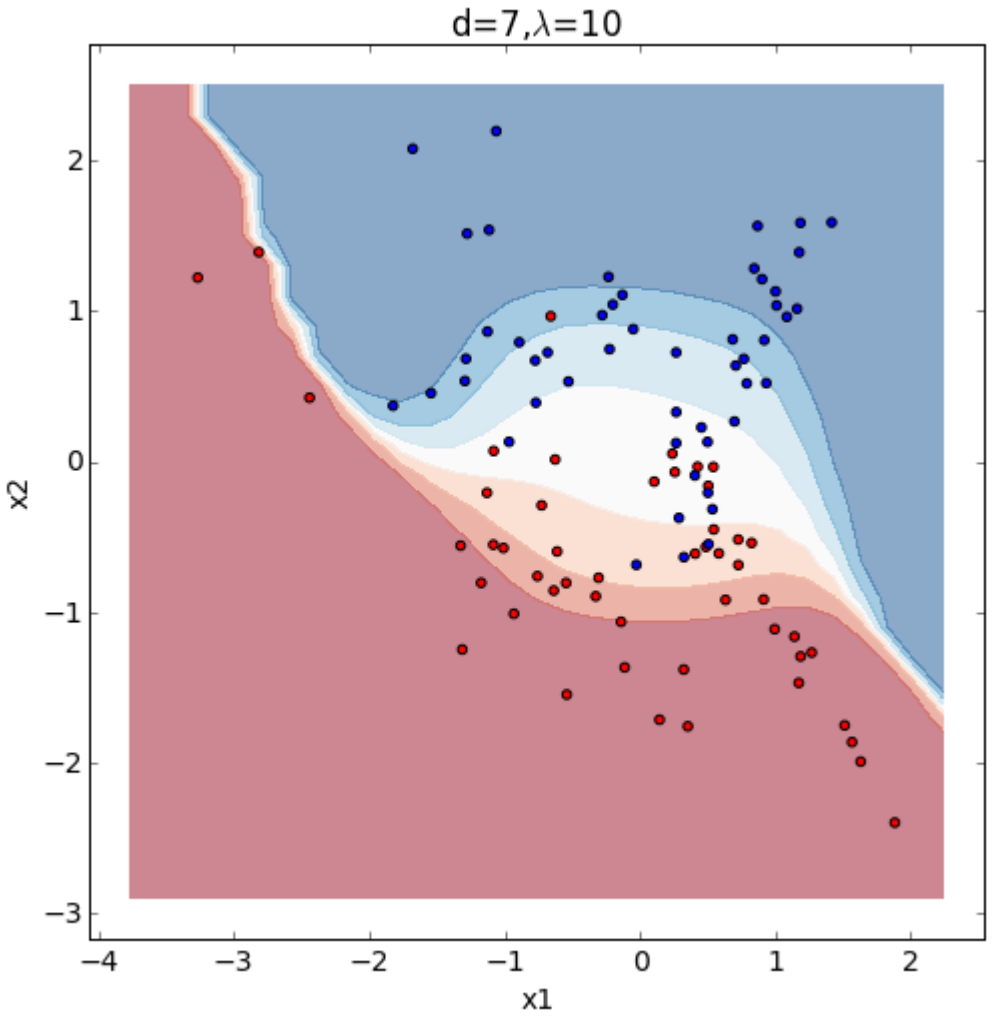


regularized linear regression



regularized logistic regression





support vector machines

Fit a linear model

$$f(x, \theta) = \theta' x$$

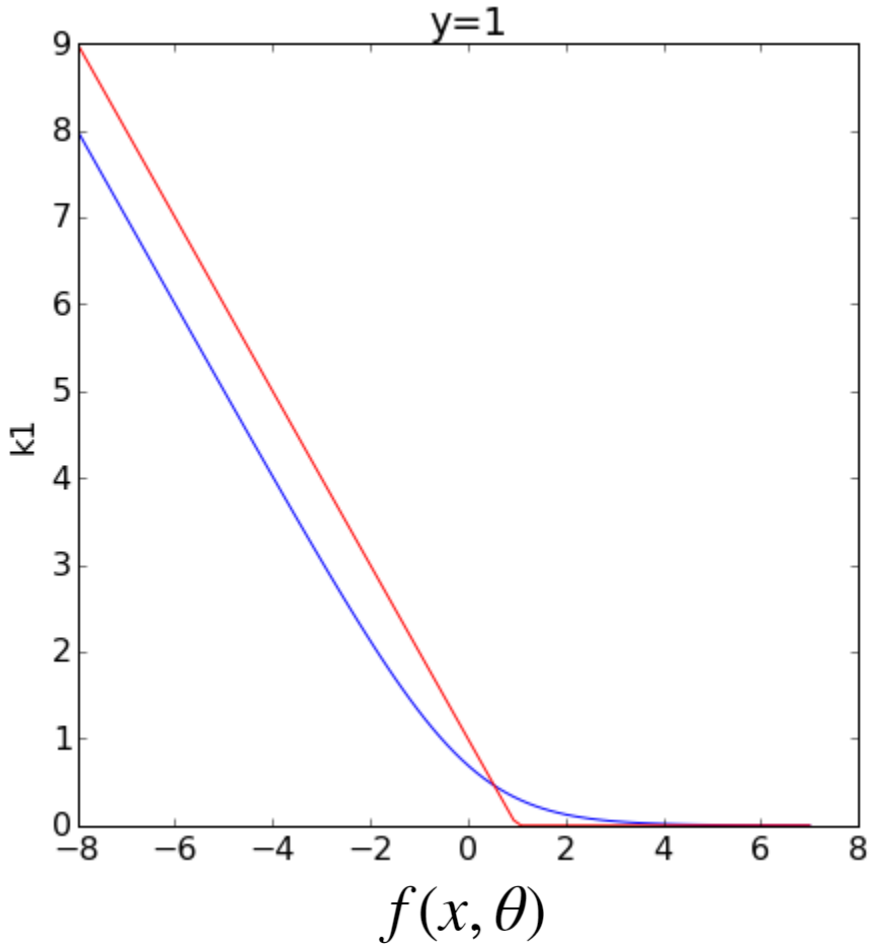
such that

$$J(\theta) = \left[C \sum_{i=1}^{n} y^{(i)} k_1(\theta' x^{(i)}) + (1 - y^{(i)}) k_0(\theta' x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2$$

with $k_1(\theta' x) = max(0, 1 - \theta' x)$ and $k_0(\theta' x) = max(0, 1 + \theta' x)$

is minimized.

support vector machines

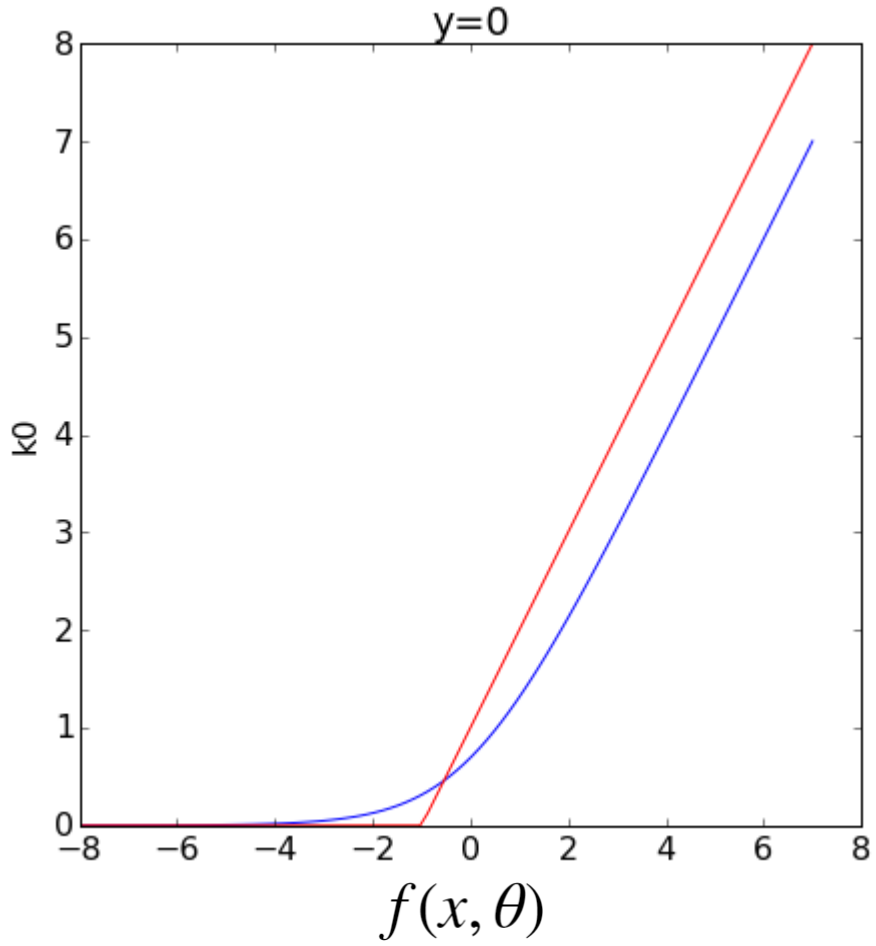


- replace cost function by piecewise linear function

- if *y* = 1 then the contribution to the cost is

$$k_1(f(x,\theta)) = max(0, 1 - f(x,\theta))$$

support vector machines



- replace cost function by piecewise linear function

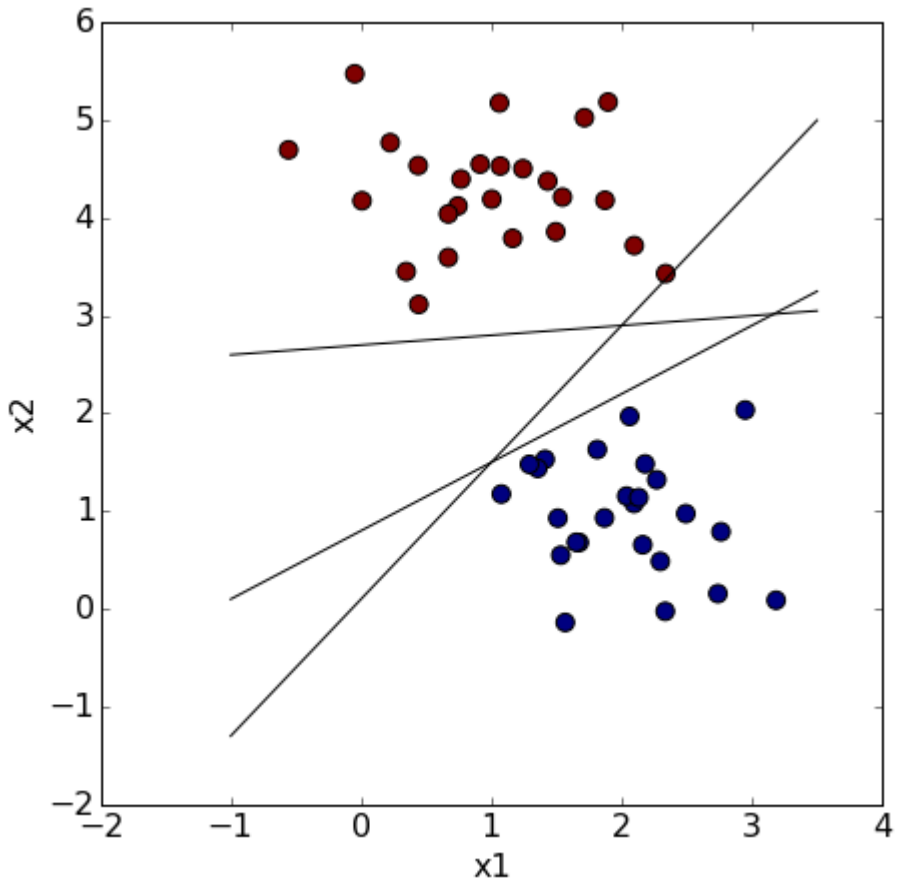
- if *y* = 1 then the contribution to the cost is

$$k_1(f(x,\theta)) = max(0, 1 - f(x,\theta))$$

- if *y* = 0 then the contribution to the cost is

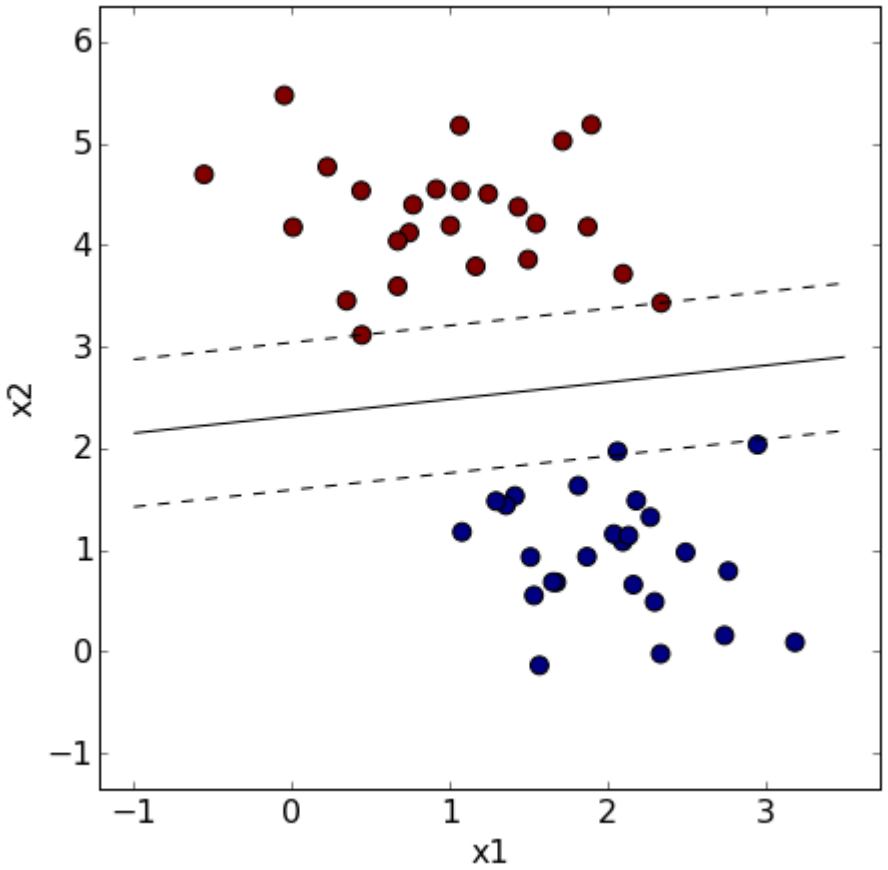
$$k_0(f(x,\theta)) = max(0, 1 + f(x,\theta))$$

support vector machines



- Consider a train set with two classes that are perfectly linearly separable.
- many possible decision boundaries

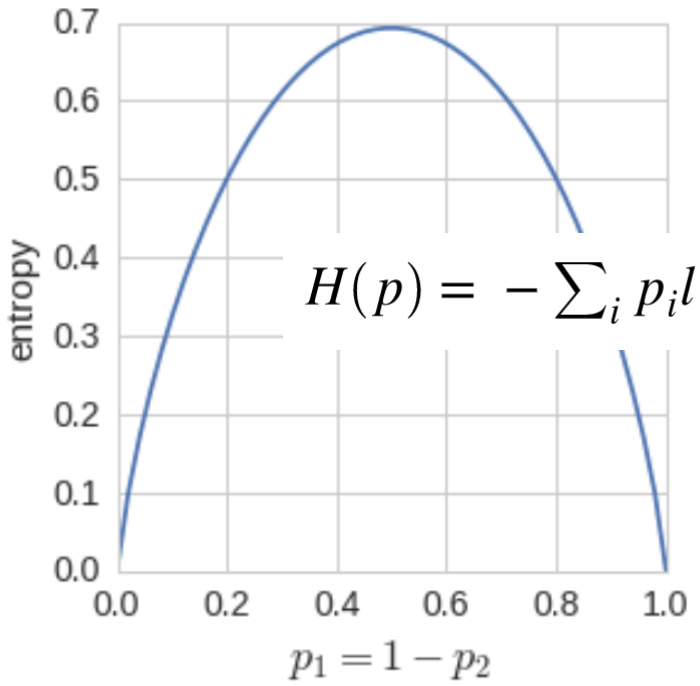
support vector machines



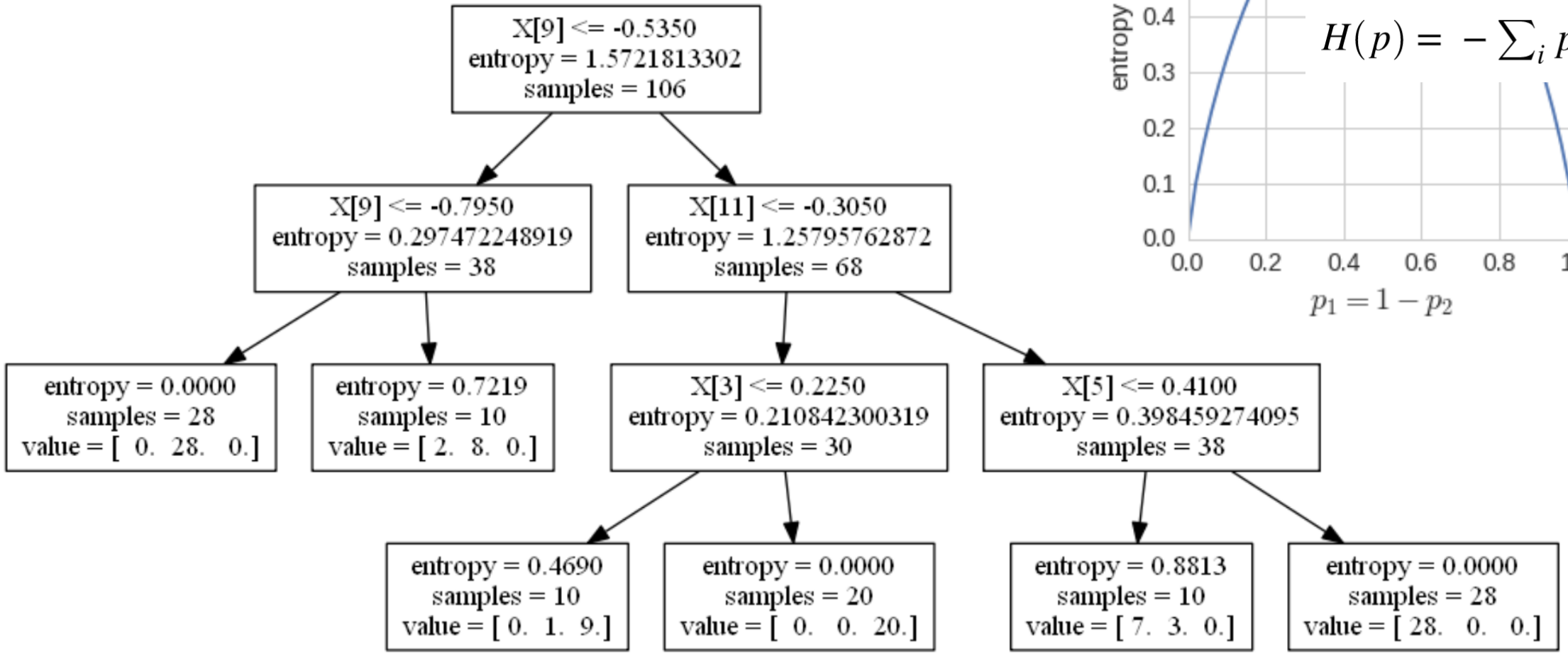
- Consider a train set with two classes that are perfectly linearly separable.
- many possible decision boundaries
- SVM picks to one that maximizes the margin between the classes

Non-linear classification models

Decision trees

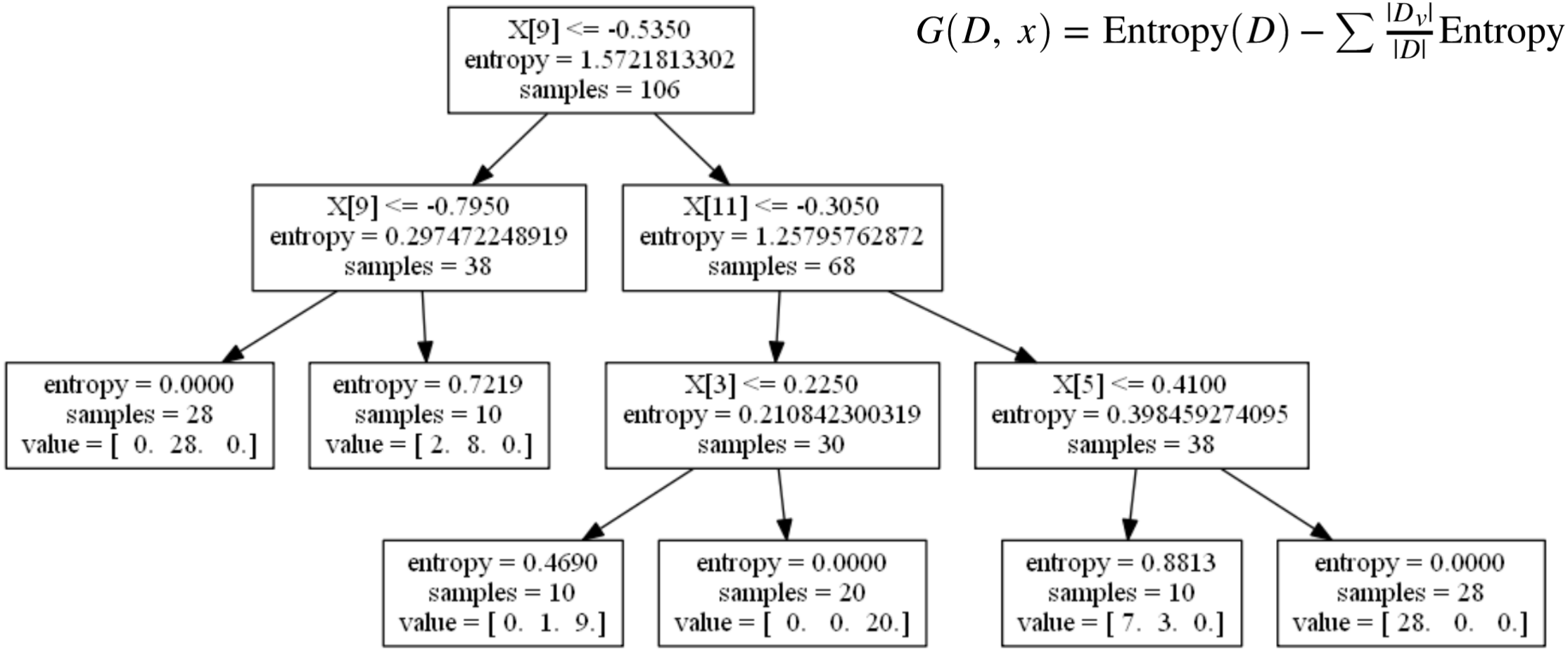


$$H(p) = -\sum_i p_i log_2(p_i)$$

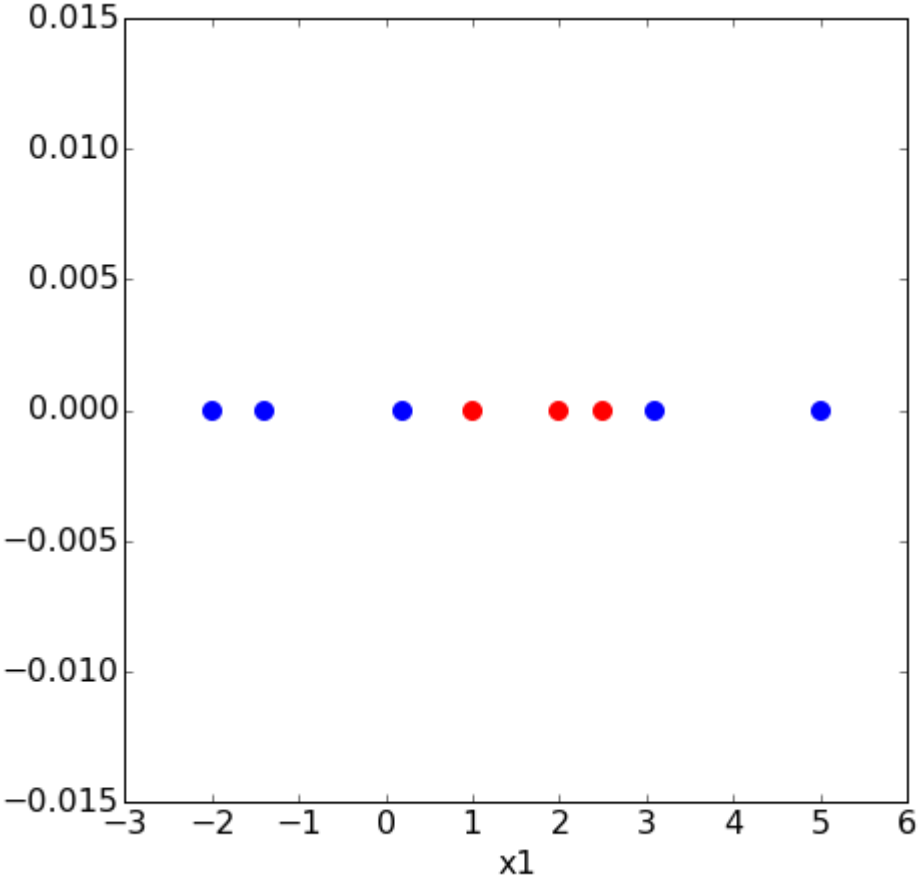


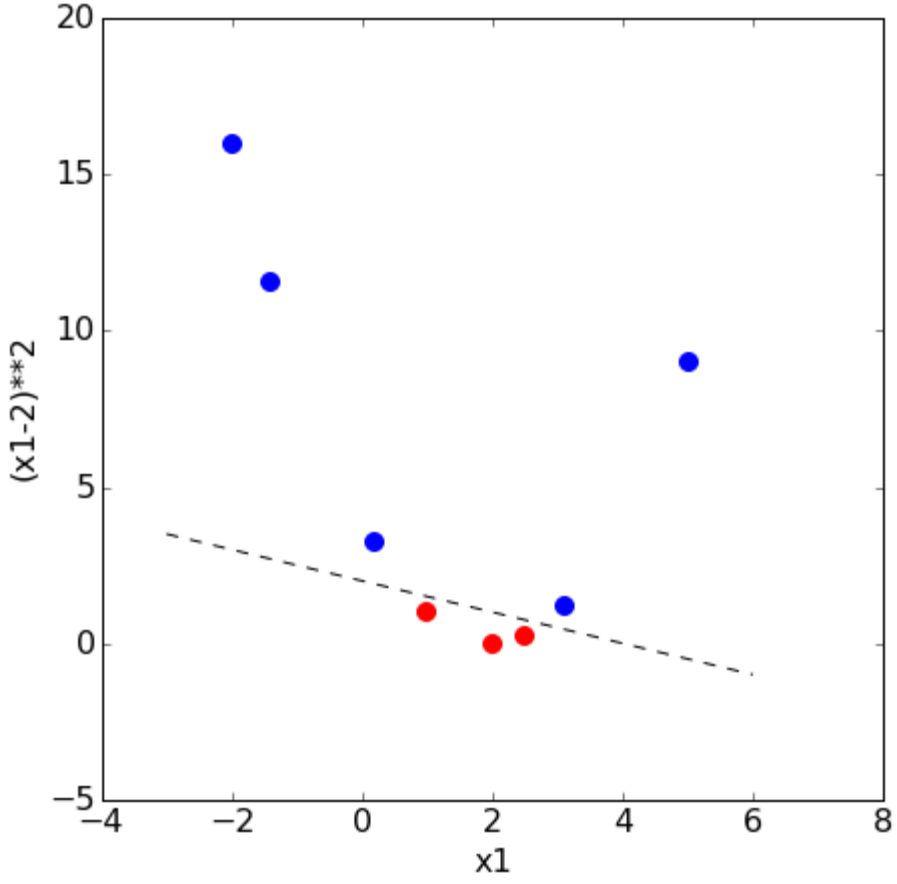
Decision trees

$$G(D,\ x) = \text{Entropy}(D) - \sum \frac{|D_v|}{|D|}\text{Entropy}(D_v)$$



kernel support vector machines





kernel support vector machines

SVMs can also be formulated as a linear function of the samples (dual form) instead of the features as

$$f(x, \theta) = \sum_{i=1}^{n} \theta_i (x \cdot x^{(i)}) + \theta_0$$

that can be reformulated as a non-linear function using what is know as a kernel function

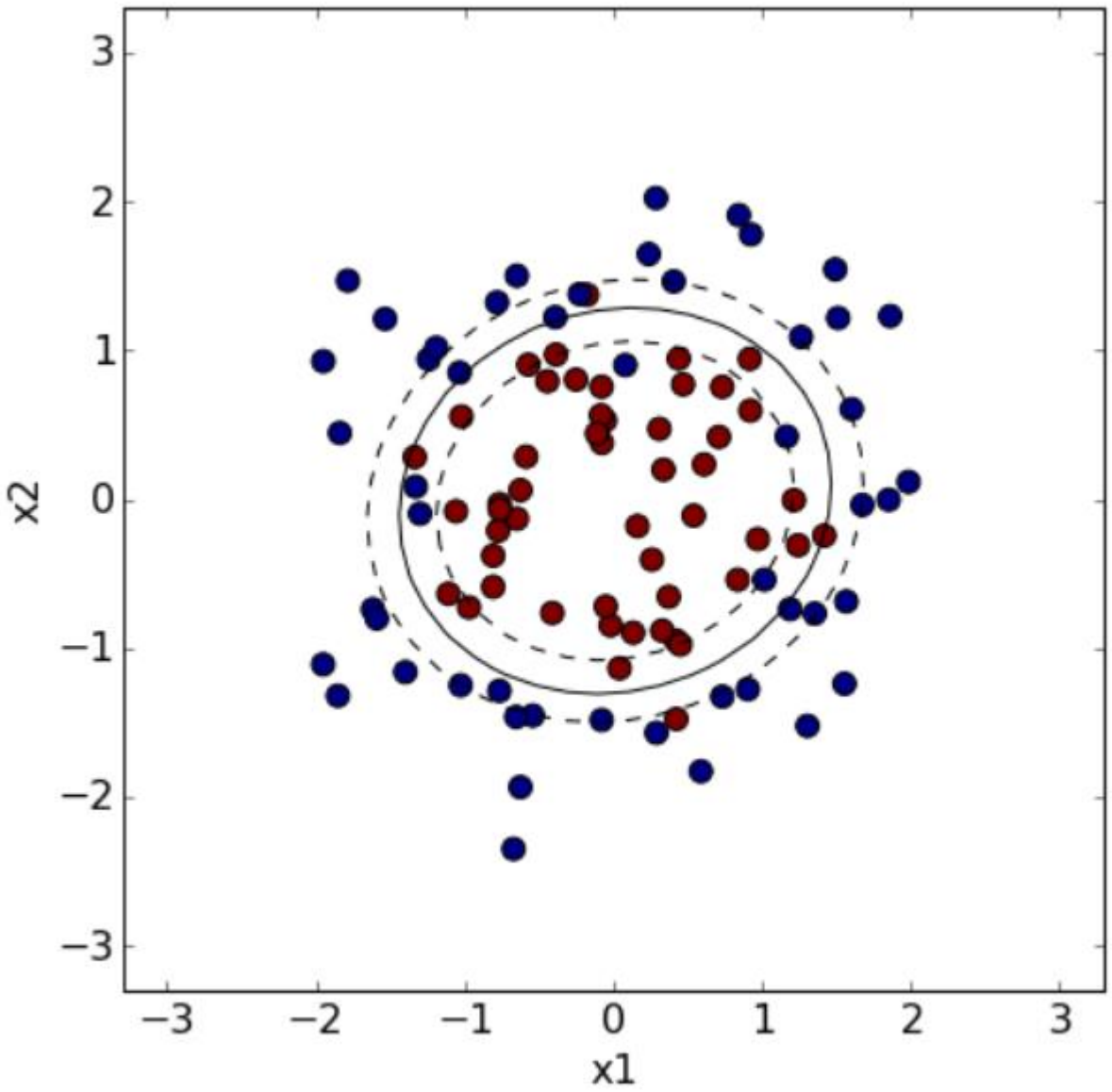
$$K(x^{(i)}, x^{(j)})$$

to become

$$f(x, \theta) = \sum_{i=1}^{n} \theta_i K(x, x^{(i)}) + \theta_0$$

The data points $x^{(i)}$ for which $\theta_i > 0$ are called the support vectors.

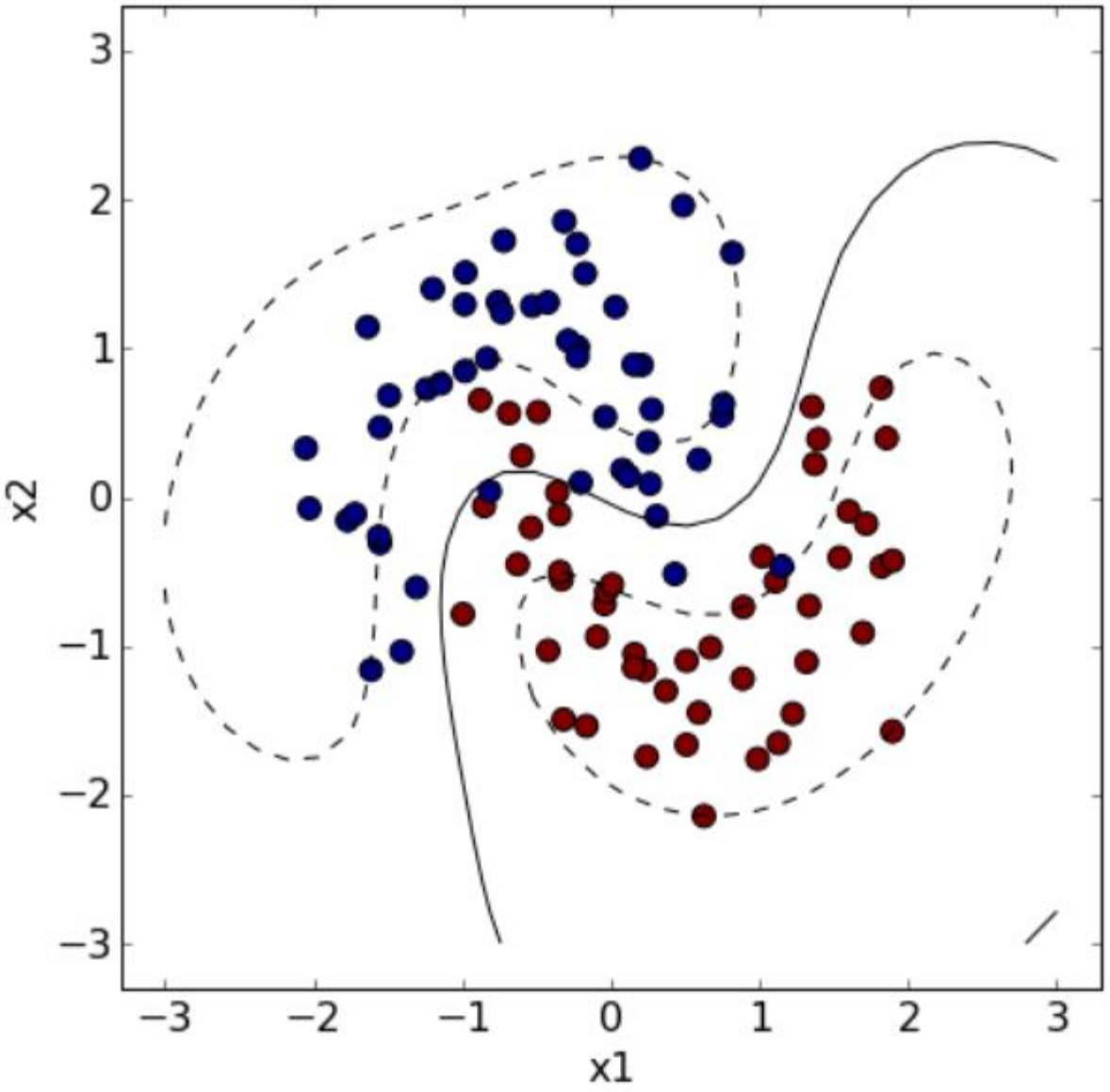
kernel support vector machines



$$f(x, \theta) = \sum_{i=1}^{n} \theta_i K(x, x^{(i)}) + \theta_0$$

$$K(x^{(i)}, x^{(j)}) = (x^{(i)} \cdot x^{(j)} + c)^d$$

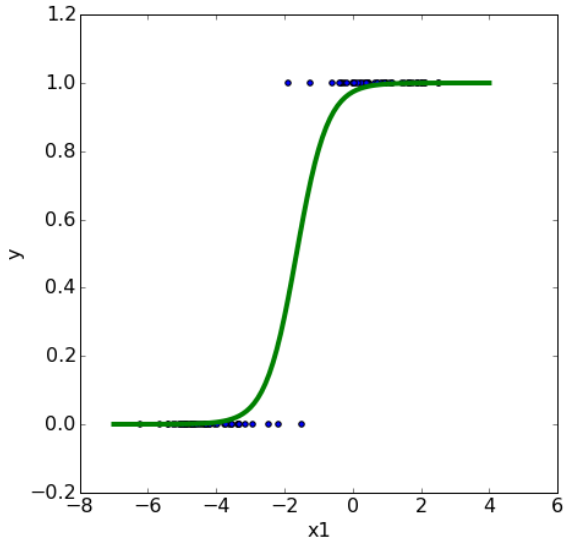
kernel support vector machines

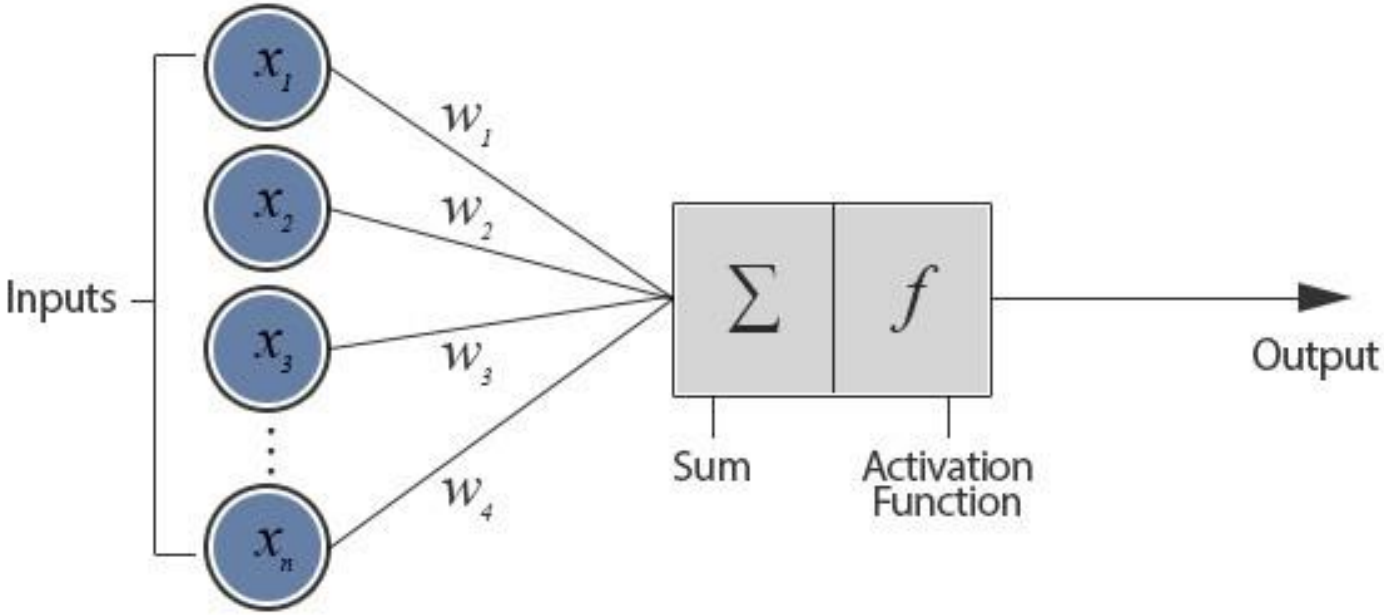


$$f(x, \theta) = \sum_{i=1}^{n} \theta_i K(x, x^{(i)}) + \theta_0$$

$$K(x^{(i)}, x^{(j)}) = \exp\left[-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right]$$

$$f(x,\theta) = g(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_m x_m) = g(\theta' x)$$



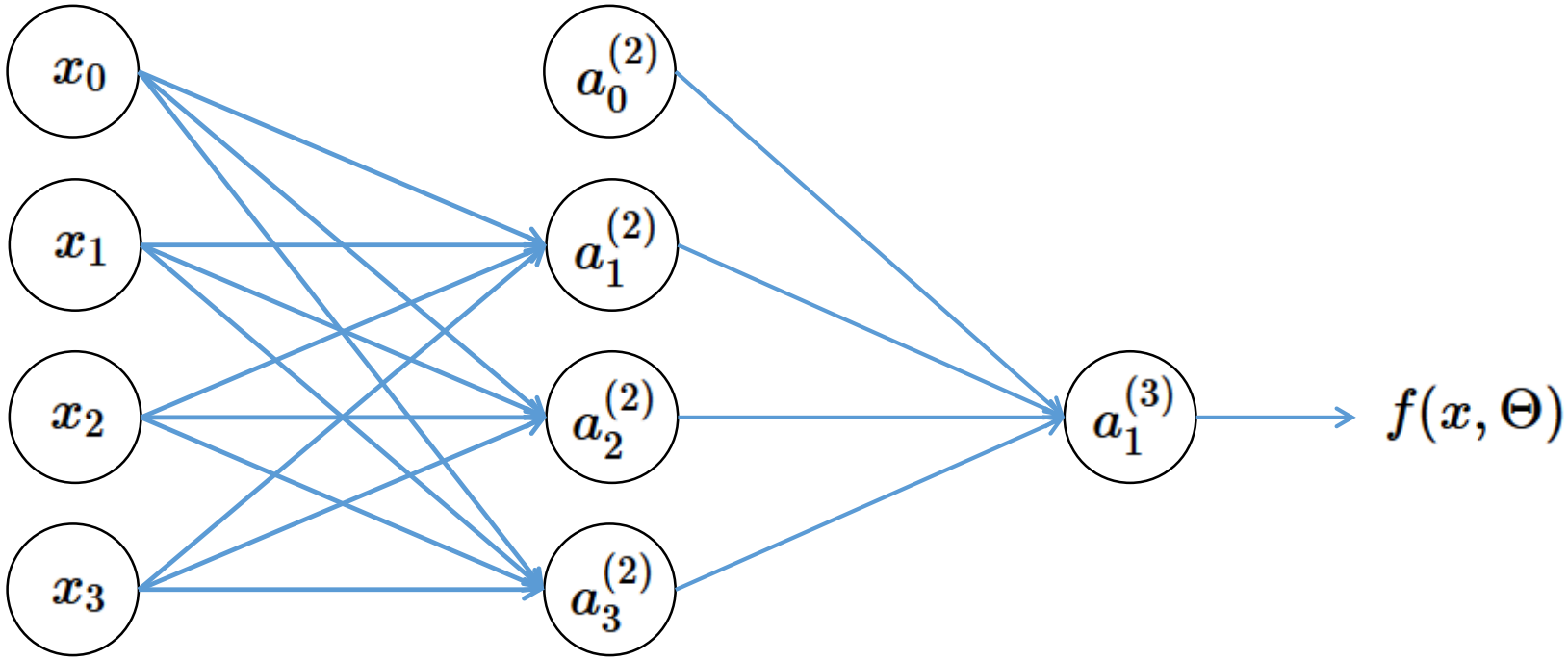


Model: $f(x,\Theta) = g(\Theta_{10}^{(2)} a_0 + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$



Cost function logistic regression:

$$J(\theta) = -[\frac{1}{m}\sum_{i=1}^{n} y^{(i)} log(f(x^{(i)}, \theta) + (1 - y^{(i)}) log(1 - f(x^{(i)}, \theta))] + \frac{\lambda}{2n}\sum_{j=1}^{n} \theta^2$$

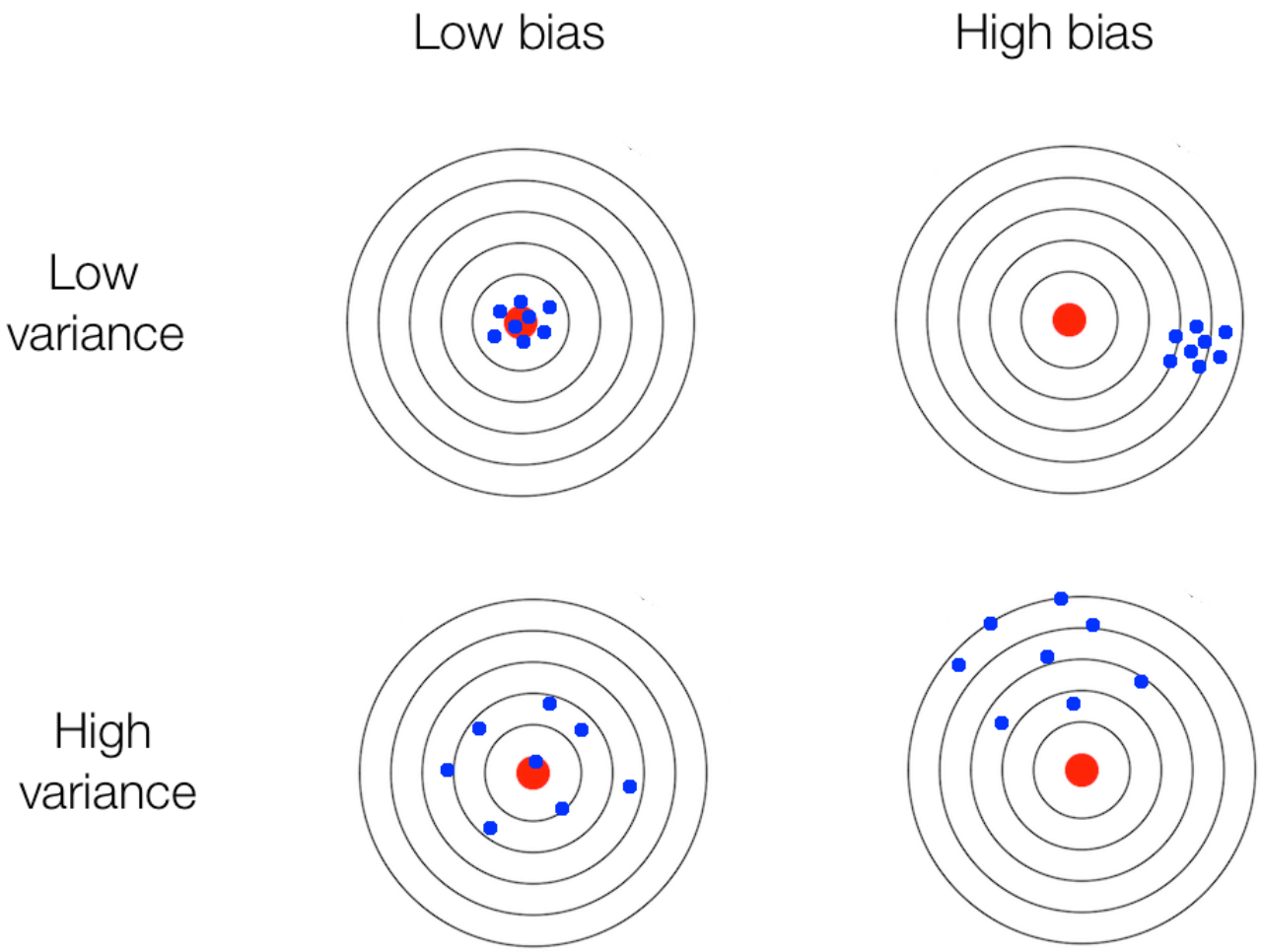
Cost function feedforward neural network:

$$J(\theta) = -[\frac{1}{m}\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} log(f(x^{(i)}, \Theta)_k) + (1 - y_k^{(i)}) log(1 - f(x^{(i)}, \Theta)_k)] + \frac{\lambda}{2n}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}} (\Theta_{ji}^{(l)})^2$$

Bias and variance

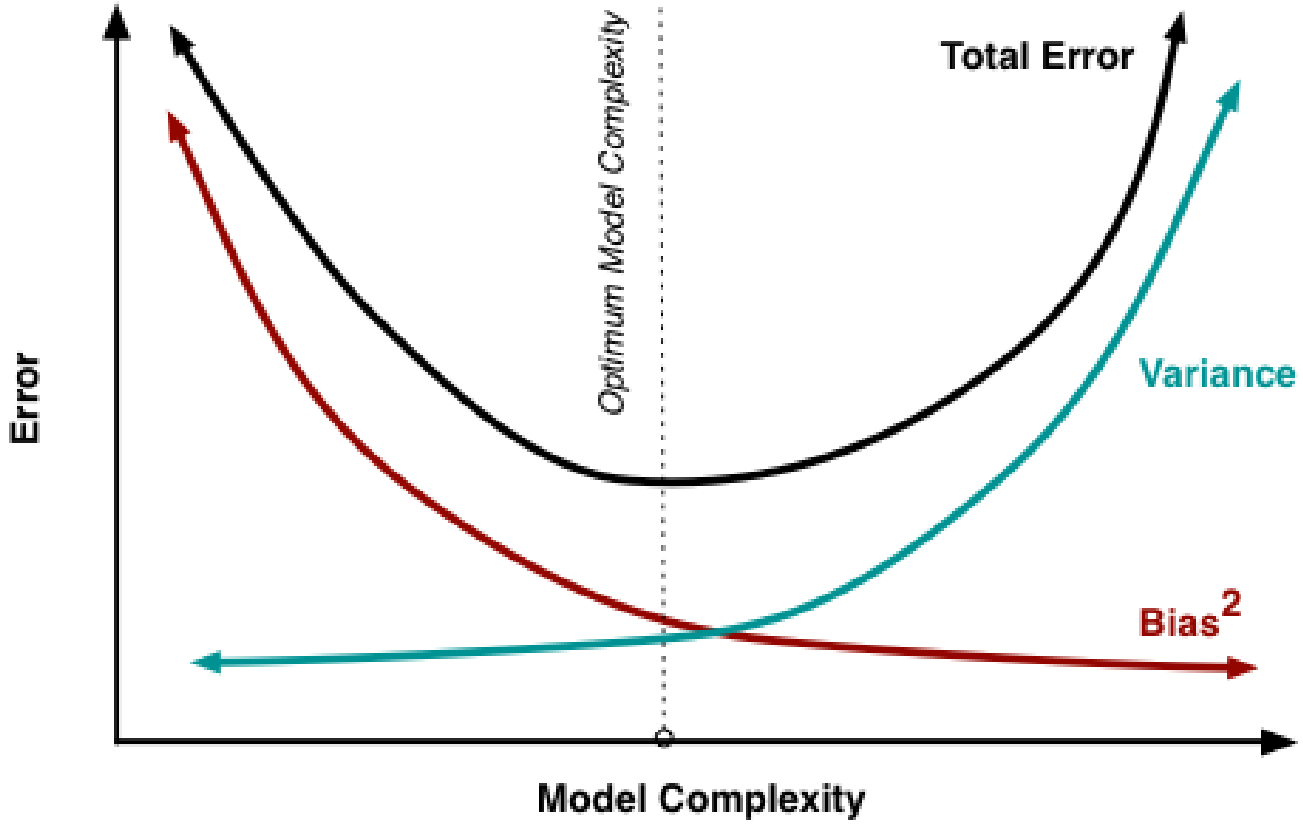
Bias-variance tradeoff

- Error due to bias
- Error due to variance
- Irreducible error

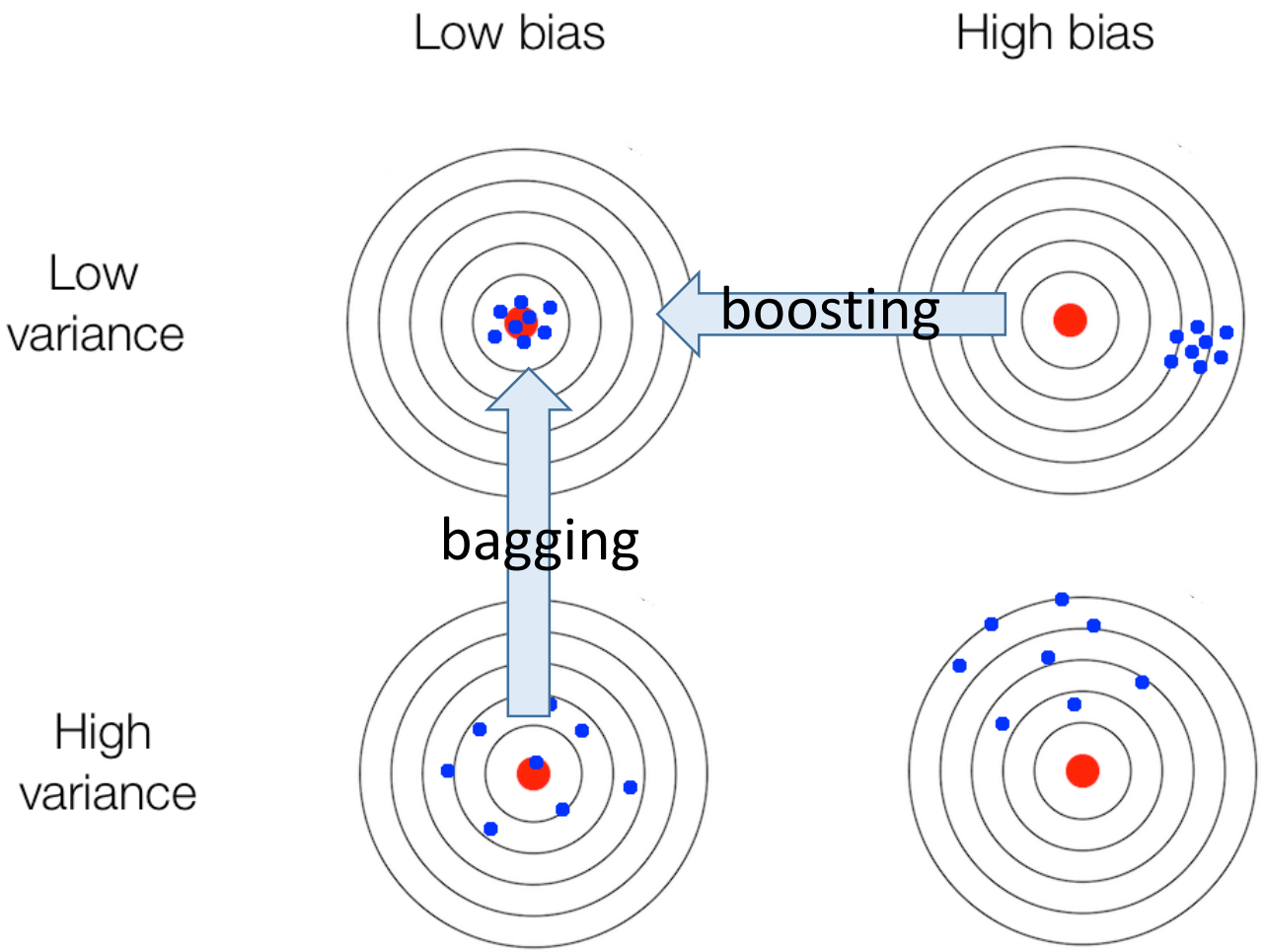


Bias-variance tradeoff

- Error due to bias
- Error due to variance
- Irreducible error



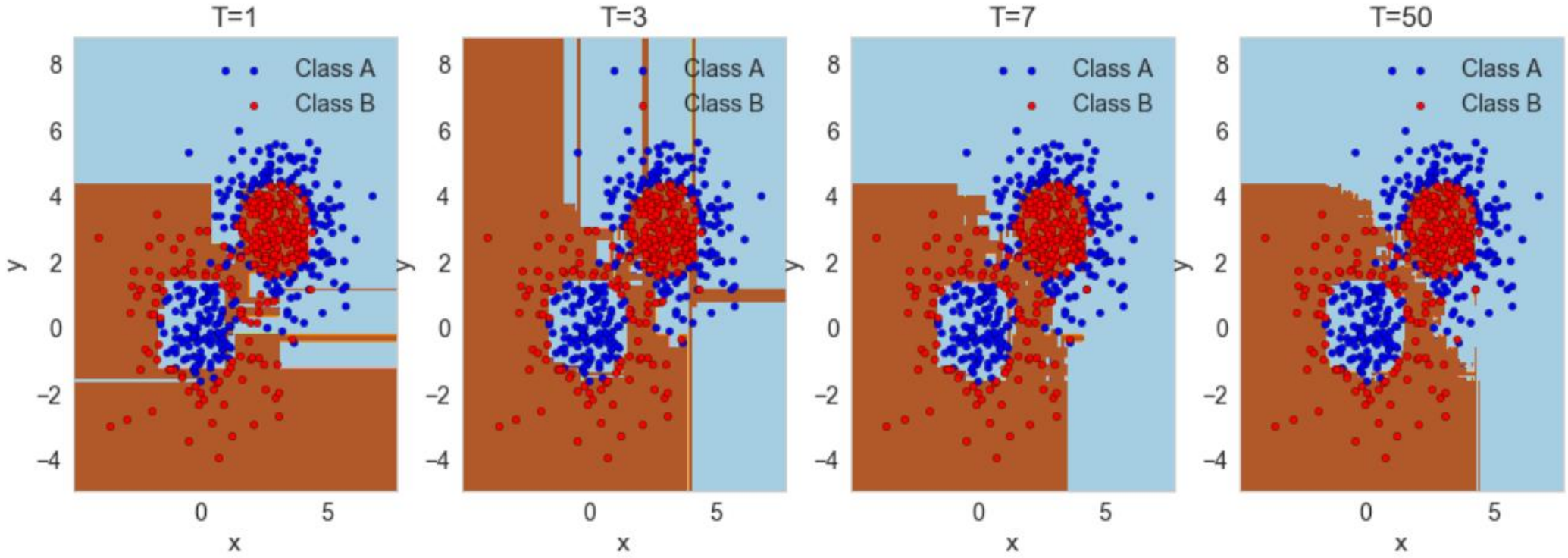
bagging/boosting



Bagging: Random Forests

- Train set contains *n* data points with *m* features.
- Construct *T* low bias high variance decision trees by following these steps:
  1. Sample ***n*** data points at random **with replacement** from the train set.
  2. At each node, select $h \ll m$ features at random and compute the best split using only these *h* features.
  3. Each tree is grown to the largest extent possible. There is no pruning or early stopping.
- Step 3 ensures that the bagged models are low bias by learning deep complex decision trees.

Bagging: Random Forests



Boosting

- reduce the bias of a high bias low variance model
- turning an ensemble of weak learners into a strong learner
- the meta-model is additive, i.e. adaboost:

$$\hat{f}(x, \theta) = \sum_{t=1}^{T} \alpha_t f_t(x, \theta)$$

Boosting: adaboost

$$\boxed{y_i \in \{-1, +1\}}$$

Initialize the weights $D_1(i) = 1/n, i = 1, 2, \ldots, n$.

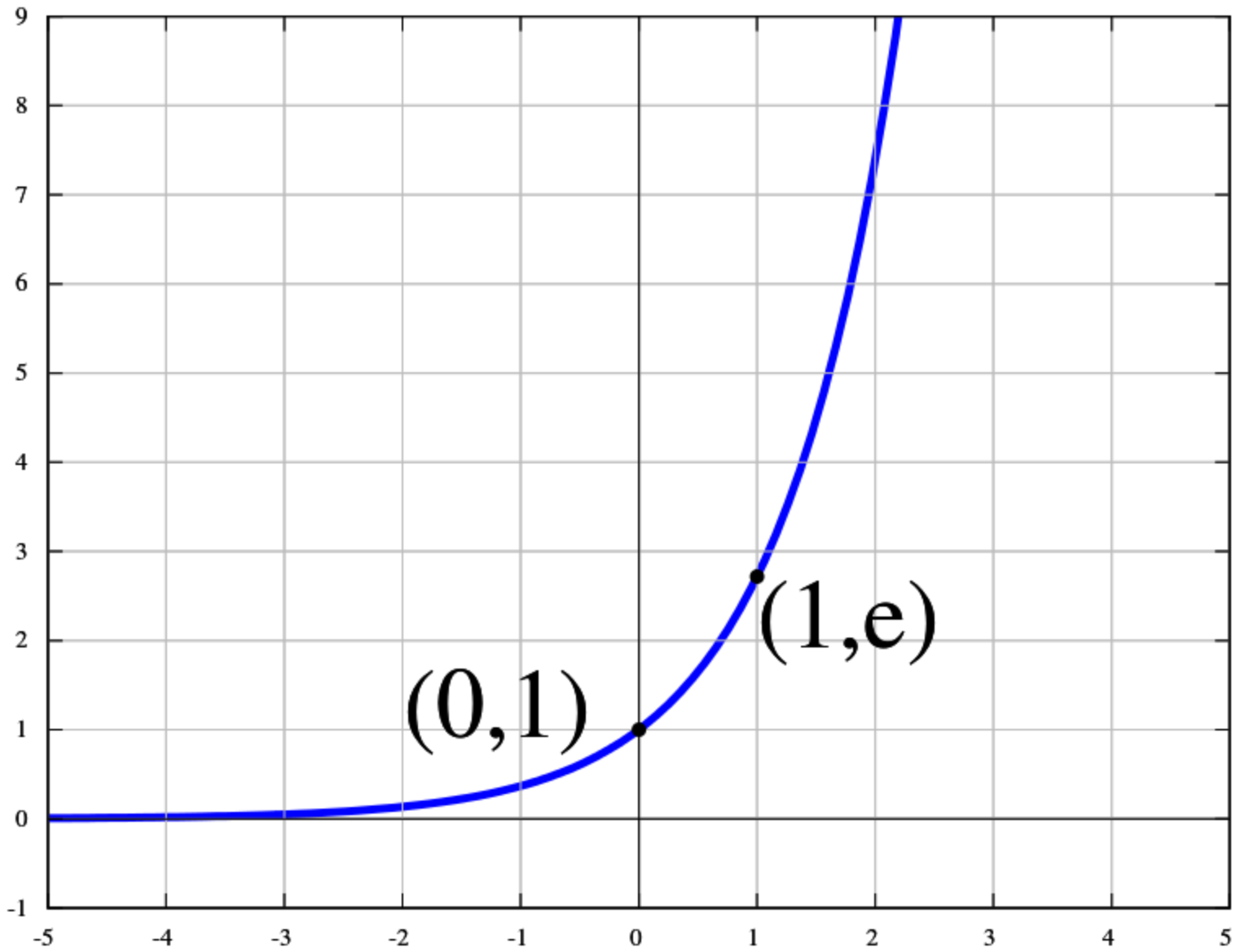
For $t = 1$ to T:

1. Fit a weak classifier $f_t(x, \theta)$ to the trainset data using weights $D_1(i)$.
2. Set $\alpha_t = \frac{1}{2} ln(\frac{1-error}{error})$.
3. Update weights:

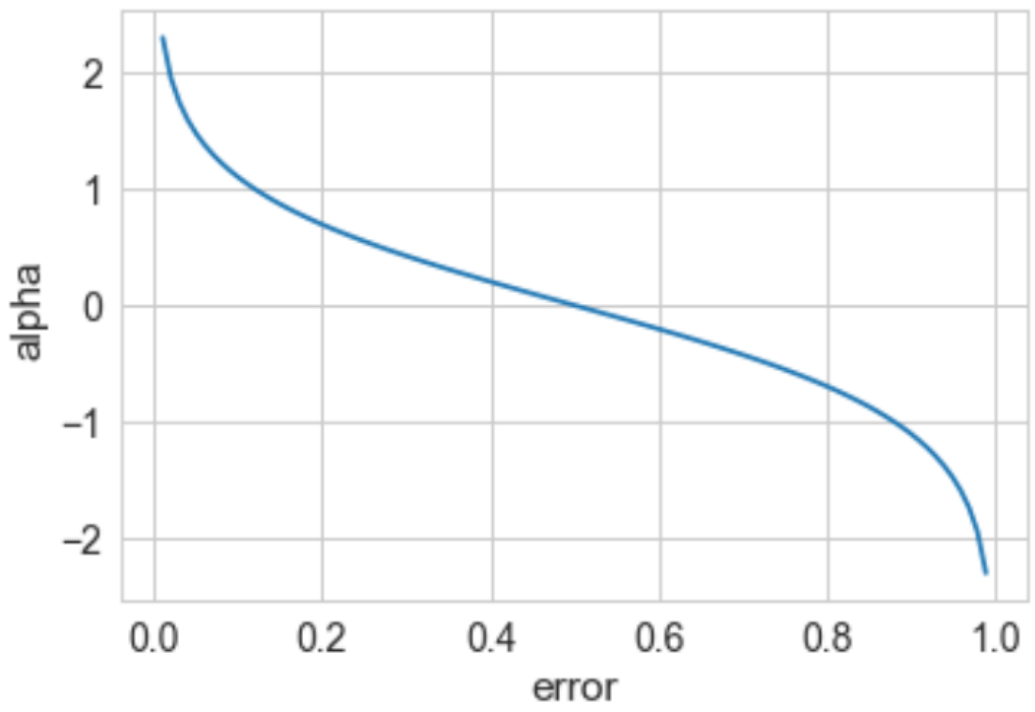
$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i f_t(x_i, \theta))}{Z_t},$$

where Z_t is a normalizing factor that makes sure that $\sum D_{t+1}(i) = 1$.

$$\hat{f}(x, \theta) = \sum_{t=1}^{T} \alpha_t f_t(x, \theta)$$


9
8
7
6
5
4
3
2
1
0
-1
-5
-4
-3
-2
-1
0
1
2
3
4
5
(0,1)
(1,e)

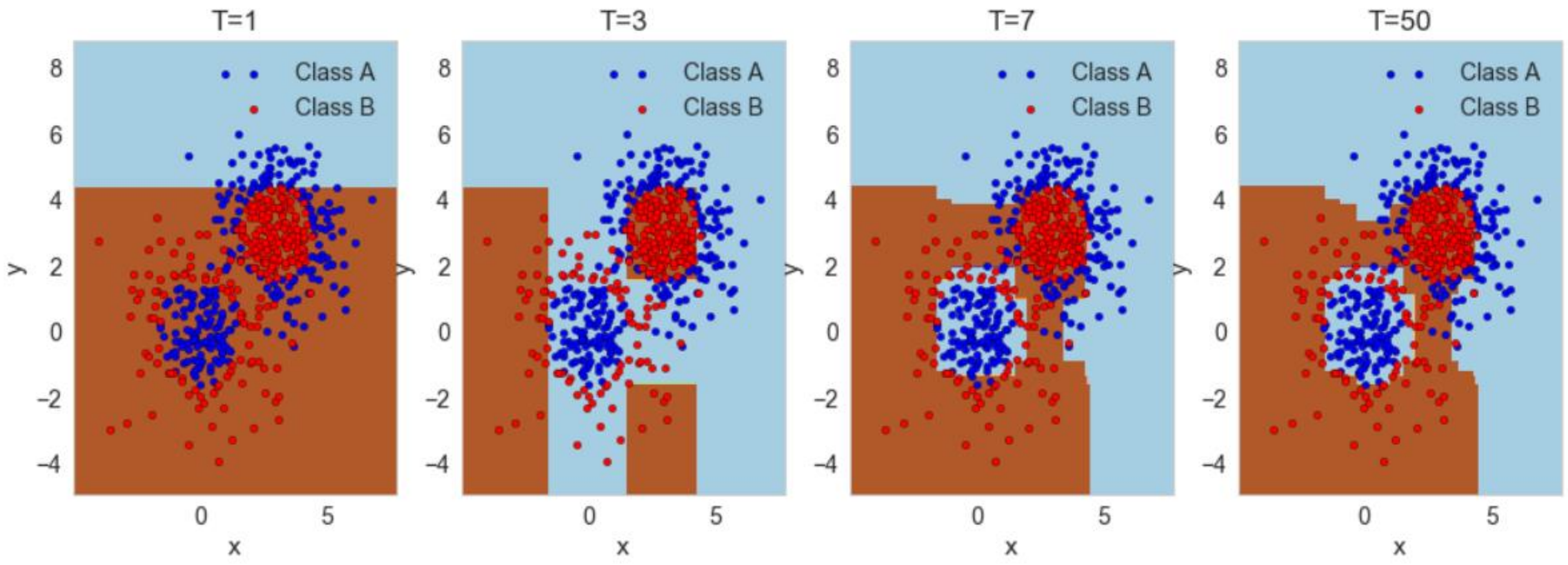
Boosting: adaboost



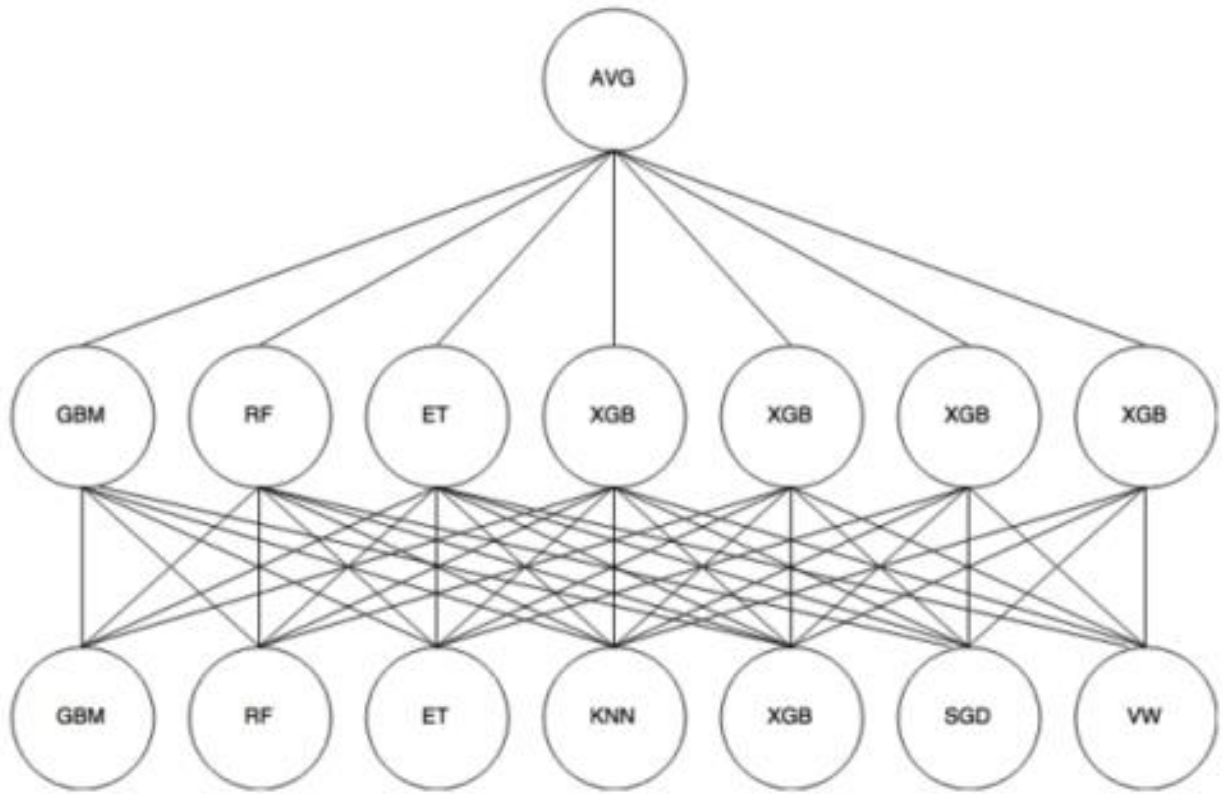
- The weight of a weak model in the boosted meta-model increases exponentially as the error approaches 0. Better models are given exponentially more weight.
- The weight is zero if the error rate is 0.5. A model with 50% accuracy is no better than random guessing, so it is ignored.
- The weight decreases exponentially as the error approaches 1. A negative weight is given to classifiers with worse than 50% accuracy. “Whatever that classifier says, do the opposite!”.

$$\hat{f}(x, \theta) = \sum_{t=1}^{T} \alpha_t f_t(x, \theta)$$

Boosting: adaboost



Blending/Stacking



https://mlwave.com/kaggle-ensembling-guide/

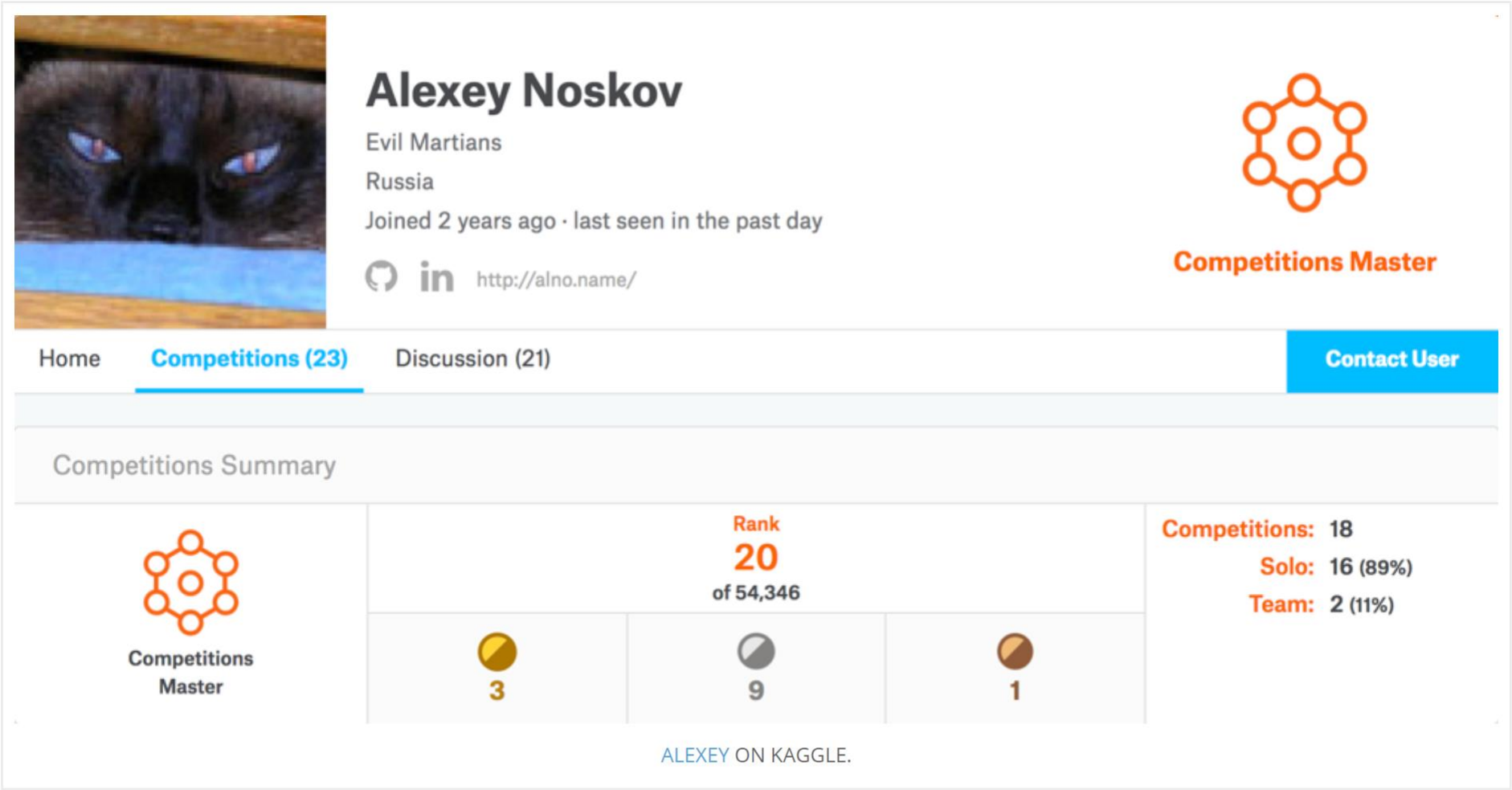
Allstate Claims Severity Competition, 2nd Place Winner's Interview: Alexey Noskov

Kaggle Team | 02.27.2017

Each row in this dataset represents an insurance claim. You must predict the value for the 'loss' column. Variables prefaced with 'cat' are categorical, while those prefaced with 'cont' are continuous.

File descriptions

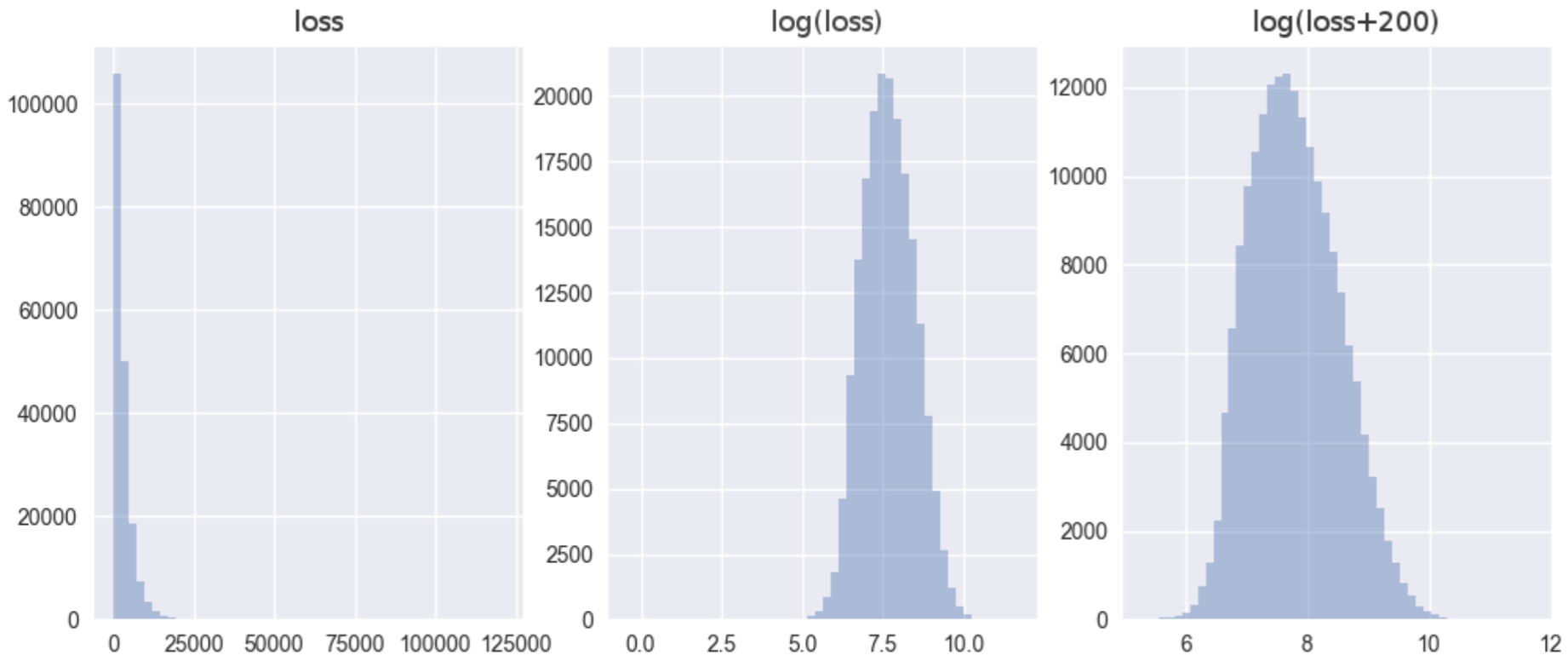
- **train.csv** - the training set
- **test.csv** - the test set. You must predict the loss value for the ids in this file.
- **sample_submission.csv** - a sample submission file in the correct format



ALEXEY ON KAGGLE.

I became interested in data science about 4 years ago - first I watched Andrew Ng's famous course, then some others, but I lacked experience with real problems and struggled to get some. But things changed when around beginning of 2015 I got to know Kaggle, which seem to be the missing piece, as it allowed me to get experience in complex problems and learn from the others, improving my data science and machine learning skills.

First, I needed to fix skew in target variable. Initially I applied log-transform, and it worked good enough, but some time after I switched to other transformations like `log(loss + 200)` or `loss ^ 0.25`, which worked somewhat better.



TARGET VARIABLE AND ITS TRANSFORMATIONS.

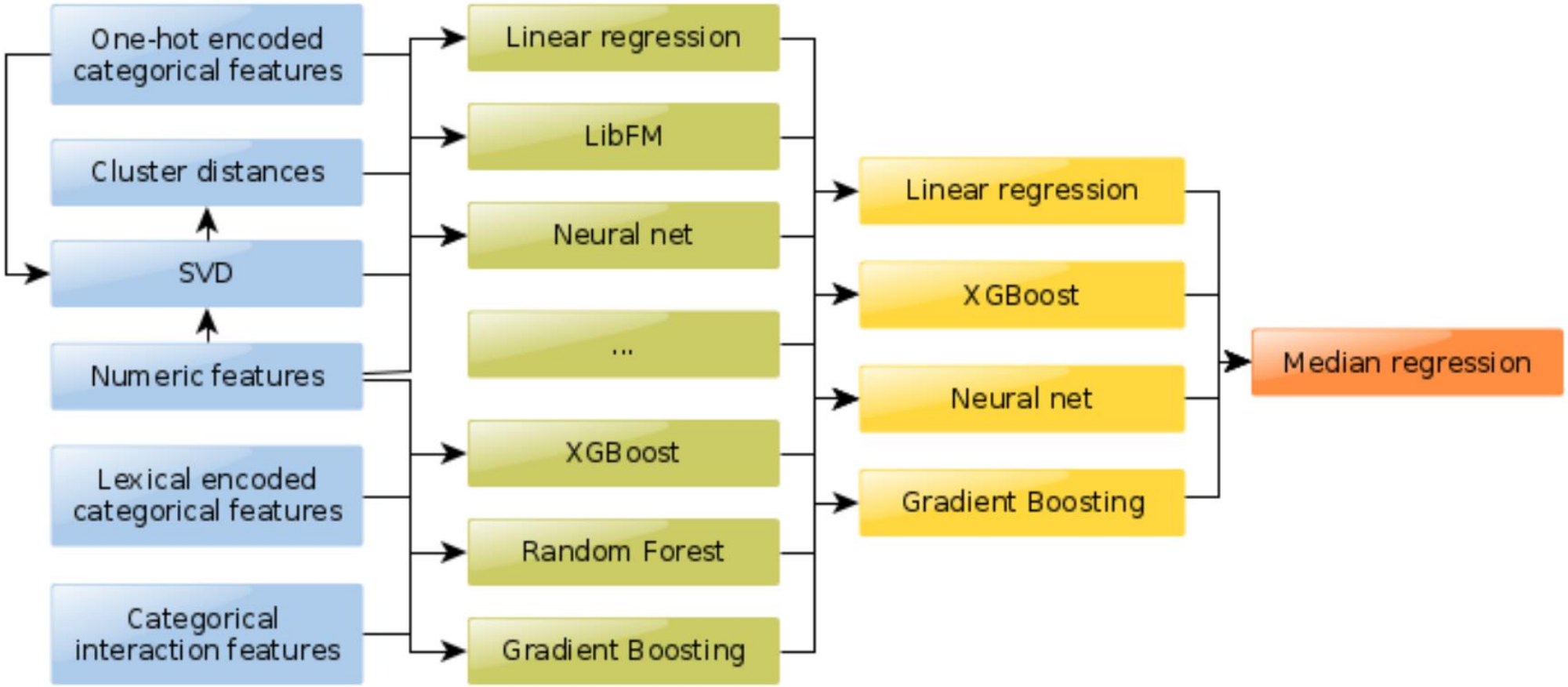
For features - first of all, I needed to encode categorical variables. For this I used basic one-hot encoding for some models, but also so-called lexical encoding, when value of encoded category is produced from its name (A becomes 0, B - 1, Z - 26, AA - 27, and so on).

Model tuning

In this competition, model hyperparameter tuning was very important, so I've contributed a lot of time in it. There are three main approaches here:

- Manual tuning, which works good when you have some intuition about parameter behaviour and may estimate model performance before its training completes by per-epoch validation scores;
- Uninformed parameter search - using GridSearchCV or RandomizedSearch from sklearn package, or similar - most simple of all;
- Informed search using HyperOpt or BayesOptimization or similar package - it tries to fit some model to scores of different parameter sets and selects the most promising point for each next try - so it usually finds the optimum a lot faster than uninformed search.

I used manual tuning for XGBoost and NN models which provide per-epoch validation scores and bayes optimization package for the others.


One-hot encoded categorical features
Cluster distances
SVD
Numeric features
Lexical encoded categorical features
Categorical interaction features
Linear regression
LibFM
Neural net
...
XGBoost
Random Forest
Gradient Boosting
Linear regression
XGBoost
Neural net
Gradient Boosting
Median regression

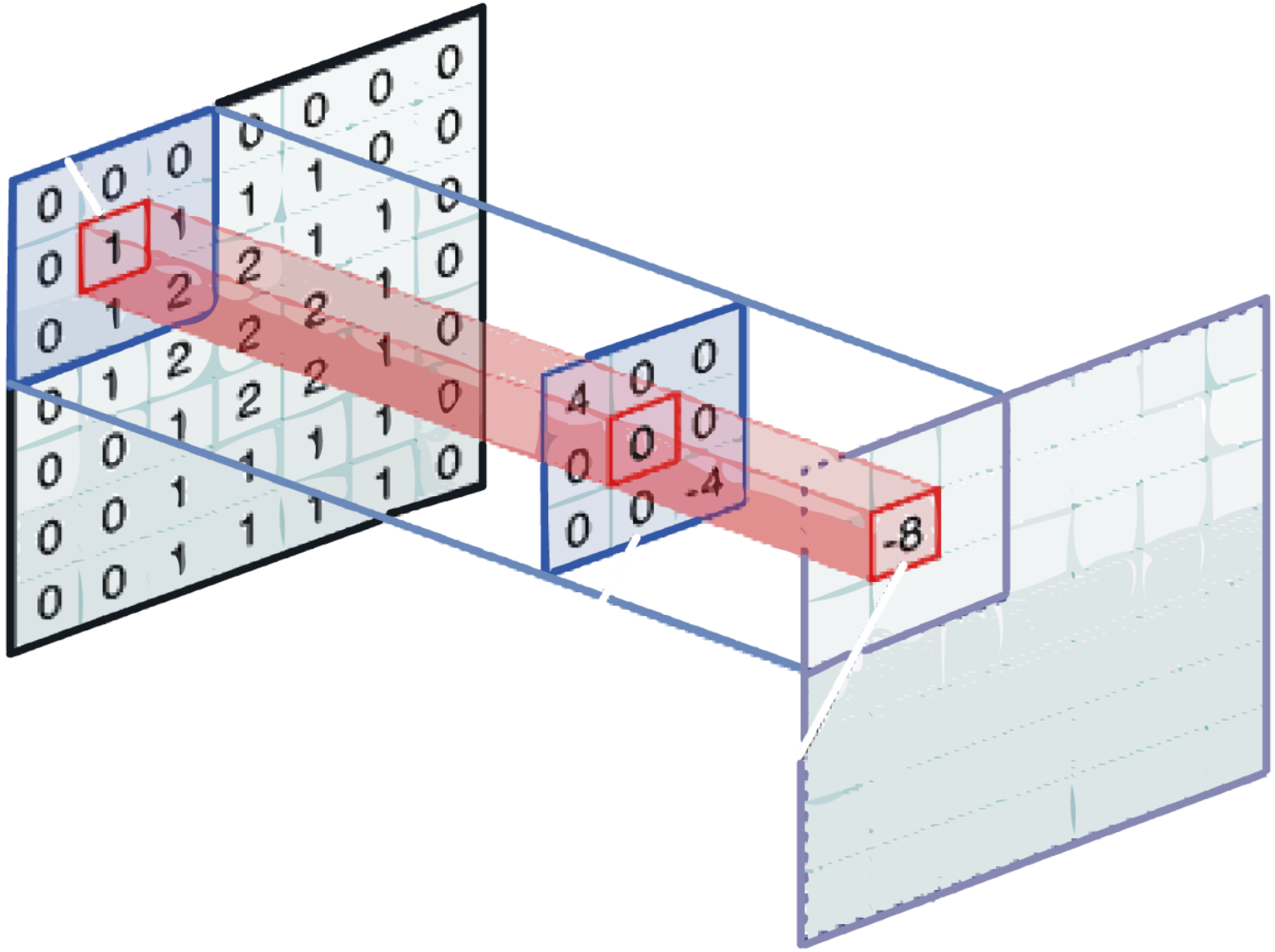
Convolution

What We See

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	91	08
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	04	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	24	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	63	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	32	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
88	36	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	38	25	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	99	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	86	81	16	23	57	05	54
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

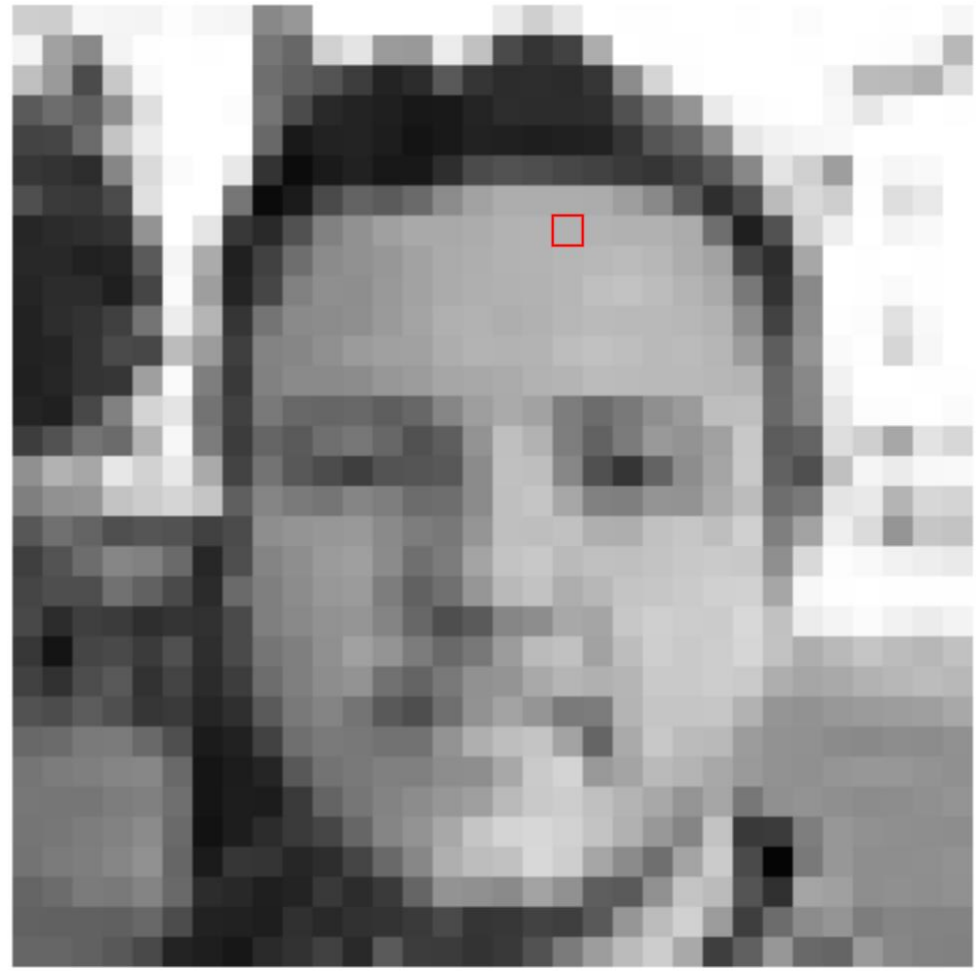
What Computers See

https://github.com/leriomaggio



https://github.com/leriomaggio

208	205	247	245	244	253	247	245	138	151	255	255	255	255	255	255	234	207	231	255	254	254	255	255	254	255	252	255	255	254	255	247
244	181	137	244	254	255	254	255	118	103	209	228	156	153	238	193	74	52	88	173	255	254	254	255	255	255	254	255	254	253	244	184
192	154	75	200	249	255	255	255	110	98	84	81	35	44	89	53	44	45	43	54	140	213	253	255	255	255	255	245	187	188	178	223
90	109	98	143	223	255	255	252	117	75	41	35	31	24	25	38	45	44	44	48	81	118	148	234	252	254	255	248	231	248	255	254
67	89	107	198	238	255	255	255	104	25	34	35	29	20	25	34	32	30	32	34	53	85	100	142	231	242	247	249	255	255	255	255
55	51	45	134	218	251	255	232	51	12	28	33	24	24	48	75	82	78	71	88	58	53	67	90	138	228	208	158	253	248	249	255
79	58	58	75	224	255	255	118	11	27	74	99	91	108	140	162	173	173	173	172	158	137	92	48	78	187	217	208	254	222	233	255
38	43	47	52	147	255	229	58	41	81	129	146	180	189	189	172	178	179	178	179	177	177	172	110	31	82	209	238	255	244	249	255
40	40	33	38	90	245	171	32	65	110	139	145	151	162	171	174	178	179	182	184	187	183	173	162	71	45	187	255	254	255	254	255
37	44	44	31	89	250	158	38	70	129	143	142	153	162	171	175	177	178	182	191	194	188	180	170	120	51	137	255	254	250	254	255
34	45	51	84	118	237	181	53	118	138	140	143	154	164	178	178	174	177	183	188	185	185	183	178	140	88	141	254	252	225	249	255
34	38	52	74	71	188	156	83	131	134	144	155	160	161	173	179	178	179	189	193	190	185	187	182	158	93	148	250	254	214	247	255
32	38	52	54	159	250	128	57	129	138	138	140	151	156	168	168	171	178	180	187	188	185	185	183	180	102	138	242	255	255	254	254
38	32	72	129	212	228	115	65	121	104	102	104	94	103	134	158	170	162	125	108	121	143	155	190	191	104	134	230	253	253	255	251
61	82	118	107	179	247	124	60	101	90	111	119	103	81	94	147	191	178	128	98	123	153	147	161	200	92	100	222	207	187	227	215
144	178	187	231	210	232	170	67	115	88	78	62	83	85	88	139	192	190	135	80	53	99	141	165	201	97	79	192	245	235	248	249
127	145	149	195	204	213	197	95	133	122	117	133	128	108	110	139	191	197	167	129	127	148	147	171	188	110	121	228	233	180	215	212
87	112	100	79	85	82	65	75	142	148	151	153	138	125	120	149	191	190	193	175	174	193	198	190	208	127	163	239	219	149	198	195
63	83	109	134	129	108	39	78	132	142	155	159	139	111	124	184	195	200	188	192	191	195	200	202	200	143	217	253	249	242	238	234
69	78	78	113	97	74	43	108	127	140	152	155	125	97	112	150	185	194	174	183	198	198	202	208	209	188	247	254	255	254	254	254
72	44	63	59	48	52	49	74	127	137	148	149	132	103	78	90	134	141	188	165	199	207	204	203	218	193	238	244	251	242	238	243
55	20	69	73	59	80	48	74	117	127	144	161	148	124	105	120	156	187	193	162	189	208	201	205	214	194	174	185	197	188	183	193
65	49	77	89	50	68	43	61	109	127	141	147	113	100	121	145	148	169	181	178	181	201	201	205	202	174	188	189	178	183	188	184
82	78	92	79	54	58	37	47	90	121	132	118	89	78	111	148	163	149	122	124	180	197	197	198	178	149	148	152	156	157	159	168
104	107	122	123	105	79	27	33	68	111	122	120	114	114	147	175	190	198	183	101	170	200	187	185	156	148	145	139	137	141	140	145
117	124	127	133	135	105	21	28	37	88	115	121	128	128	141	142	168	202	212	153	164	188	180	168	154	148	144	149	151	151	147	144
119	118	118	125	128	111	21	29	28	58	100	118	131	140	151	159	186	201	205	192	180	168	149	168	119	144	147	143	140	141	144	148
117	119	125	130	139	108	18	29	44	58	70	102	133	147	168	197	212	215	210	195	177	152	133	195	57	59	128	151	145	143	142	141
115	123	128	134	145	102	27	54	52	38	45	69	105	135	175	189	193	218	208	188	139	111	184	203	74	5	121	151	142	142	143	148
101	108	123	121	132	105	44	40	31	35	57	44	58	101	147	144	138	163	145	94	90	145	198	187	84	48	165	180	142	144	142	145
98	97	97	98	104	78	34	33	30	48	41	49	51	58	74	53	55	88	83	89	150	188	209	158	62	108	140	149	125	133	131	131
102	102	97	88	73	35	30	23	42	50	65	41	90	80	59	51	57	82	123	157	187	205	169	62	98	151	105	101	154	135	130	129

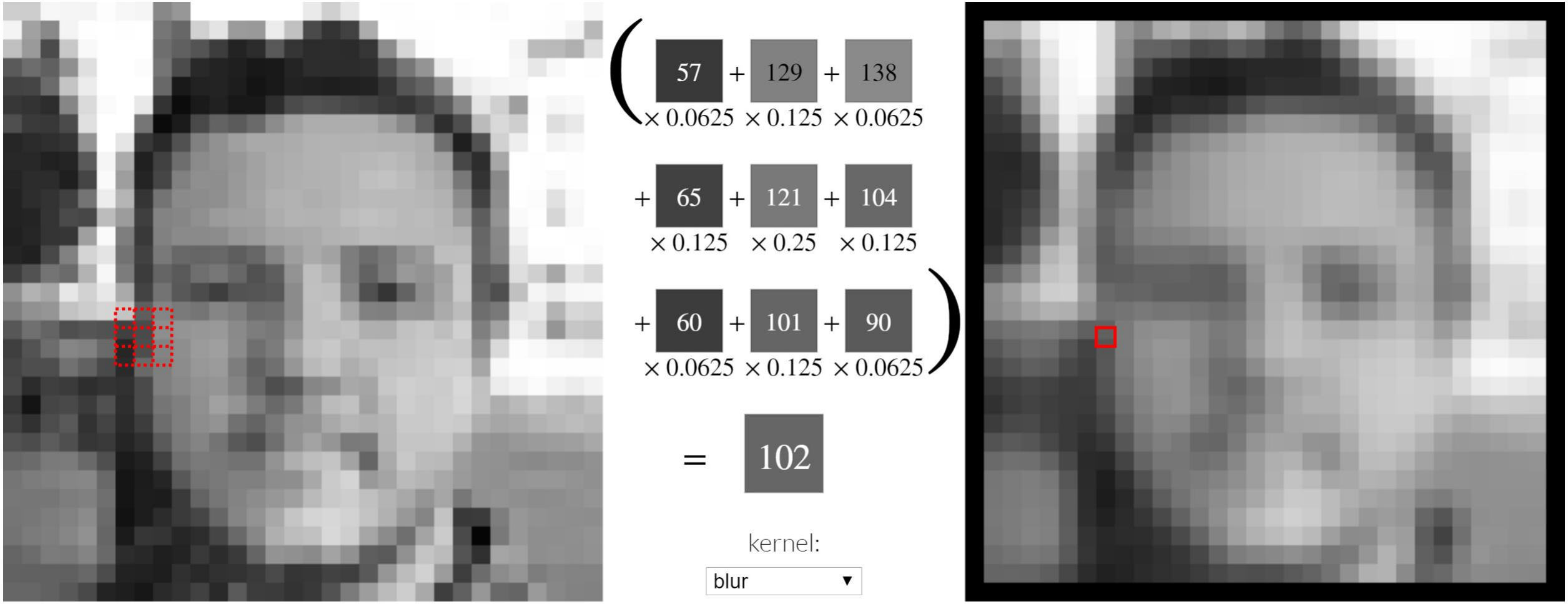


http://setosa.io/ev/image-kernels/

blur ▼

$$\begin{pmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{pmatrix}$$

http://setosa.io/ev/image-kernels/



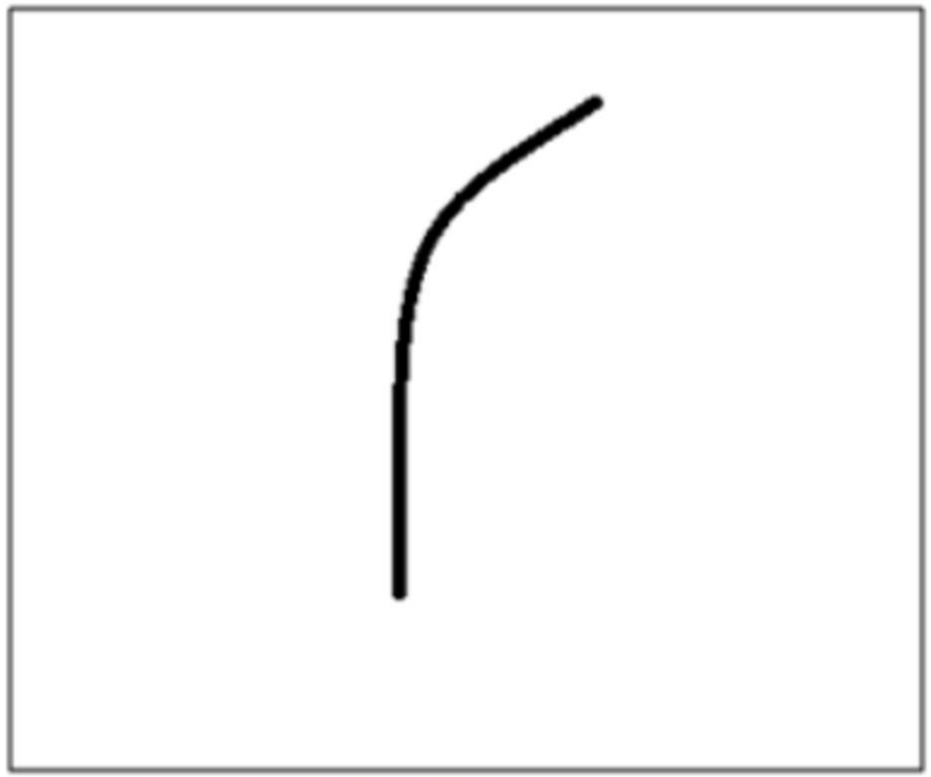
input image

output image

http://setosa.io/ev/image-kernels/

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

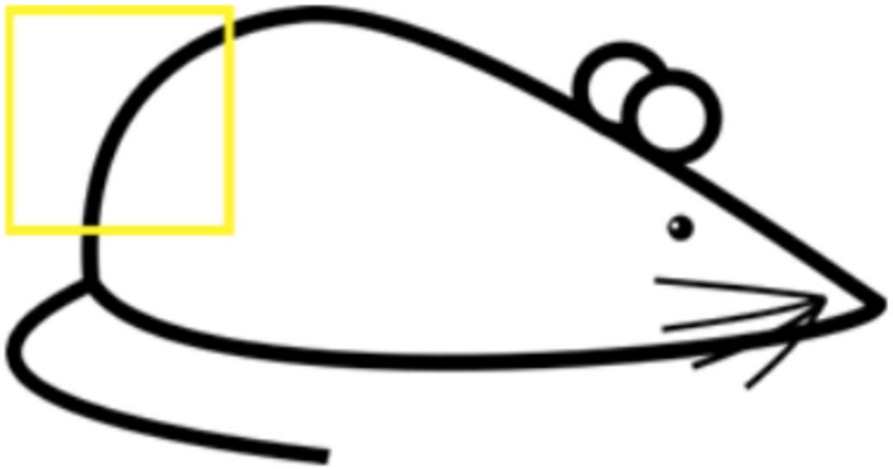
Pixel representation of filter



Visualization of a curve detector filter

https://github.com/leriomaggio

Original image



Visualization of the filter on the image

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

https://github.com/leriomaggio

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

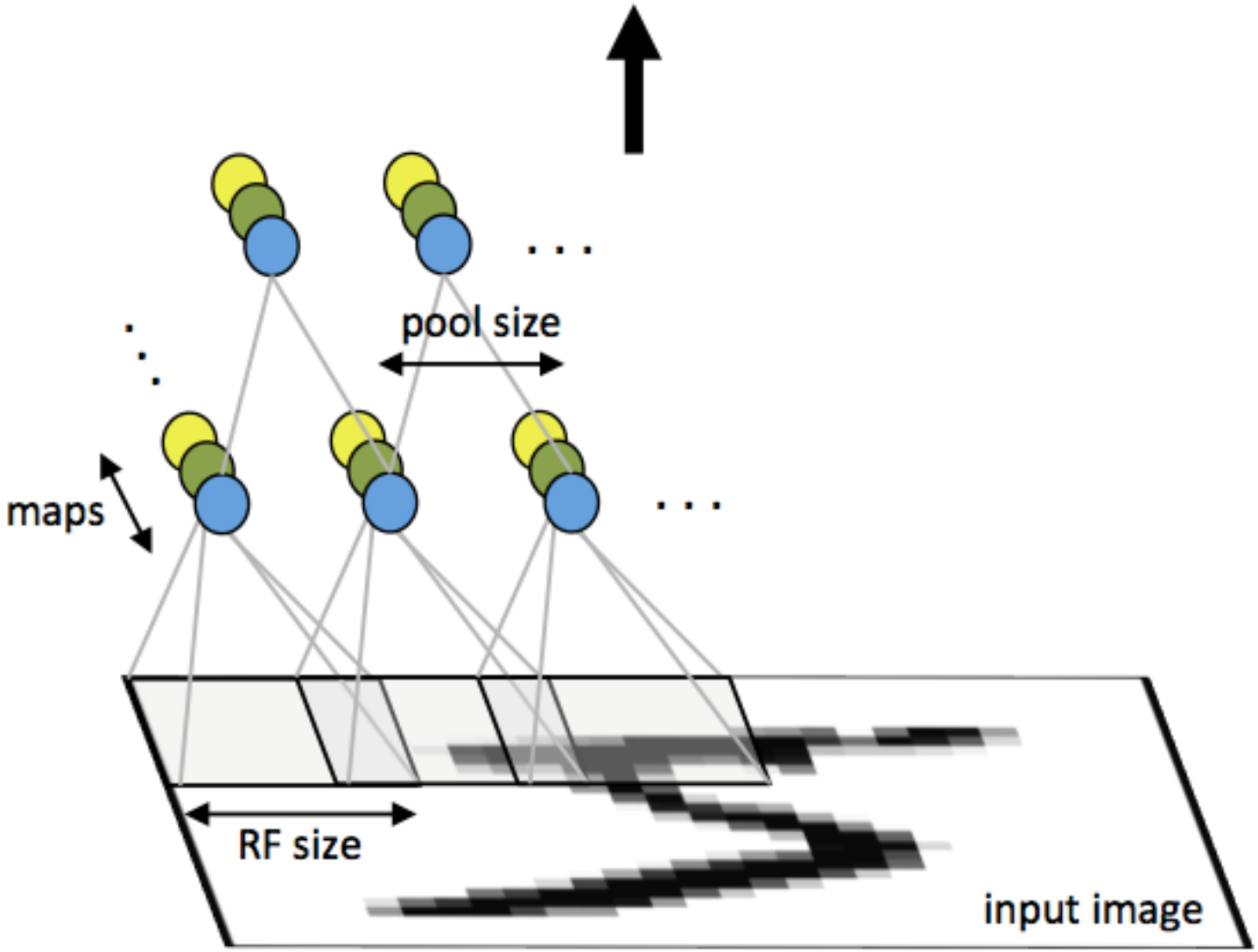
Pixel representation of receptive field

$*$

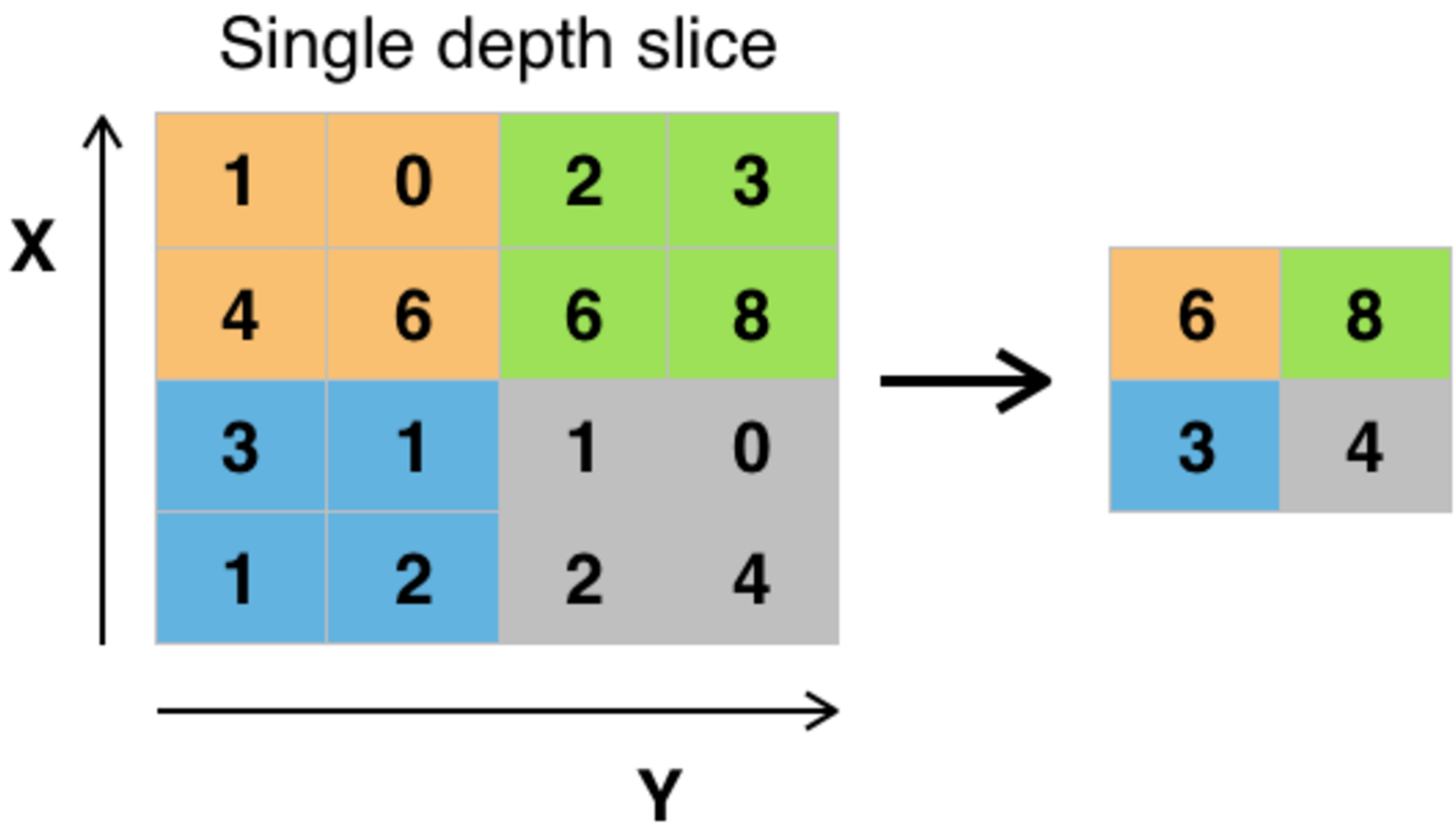
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

https://github.com/leriomaggio

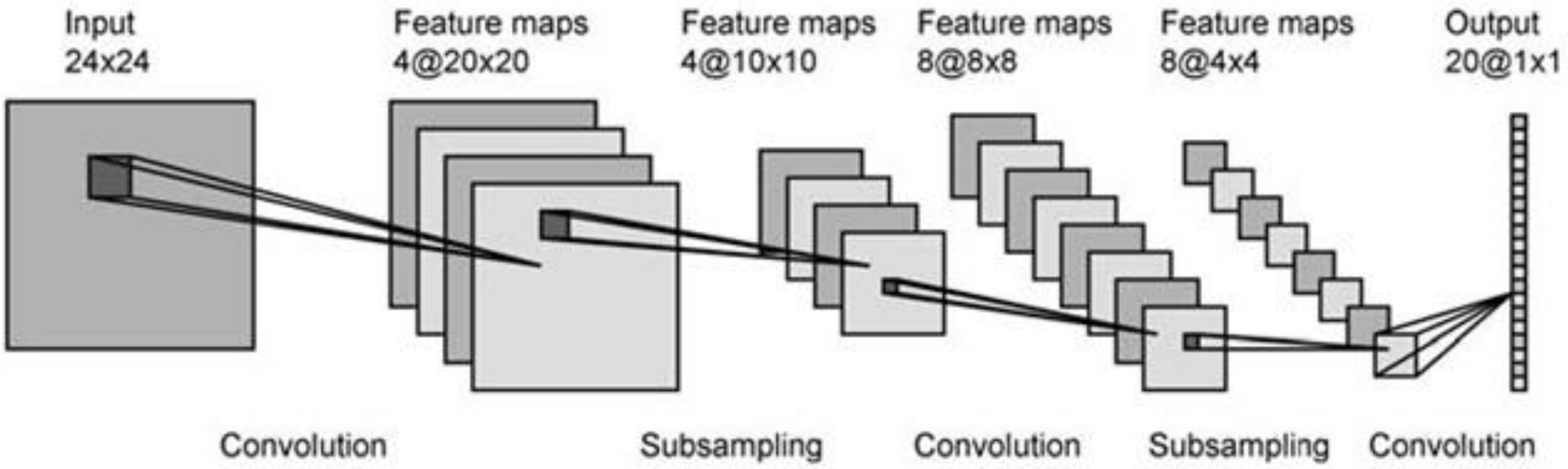


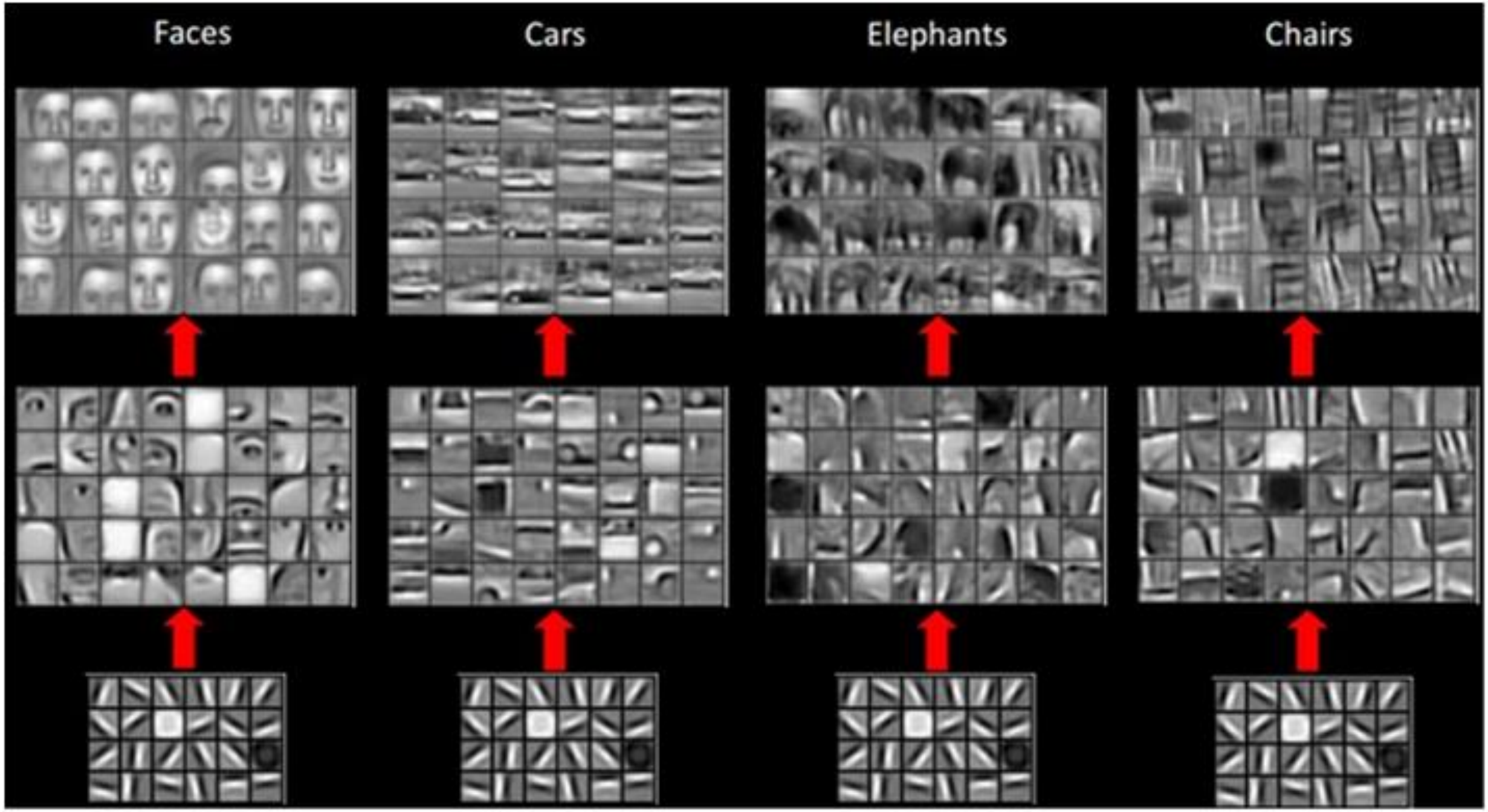
https://github.com/leriomaggio



Example of Maxpool with a 2x2 filter and a stride of 2

https://github.com/leriomaggio

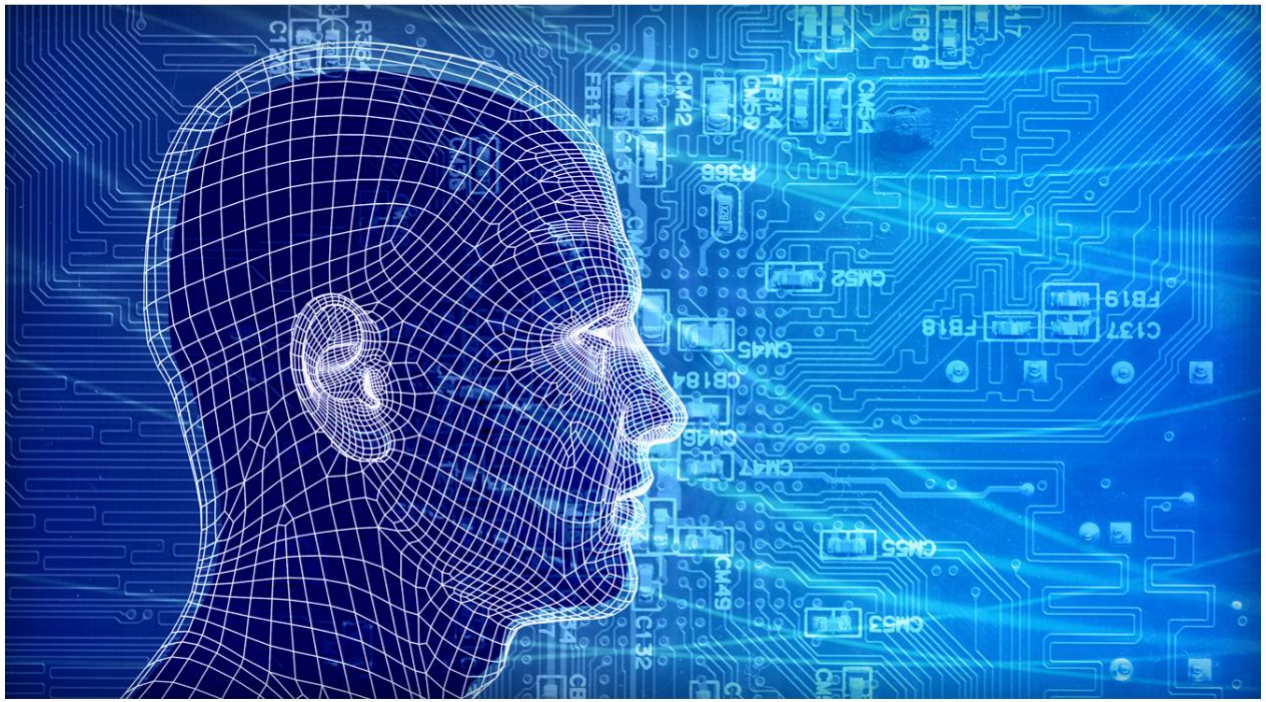

Input
24x24
Feature maps
4@20x20
Feature maps
4@10x10
Feature maps
8@8x8
Feature maps
8@4x4
Output
20@1x1
Convolution
Subsampling
Convolution
Subsampling
Convolution


Faces
Cars
Elephants
Chairs

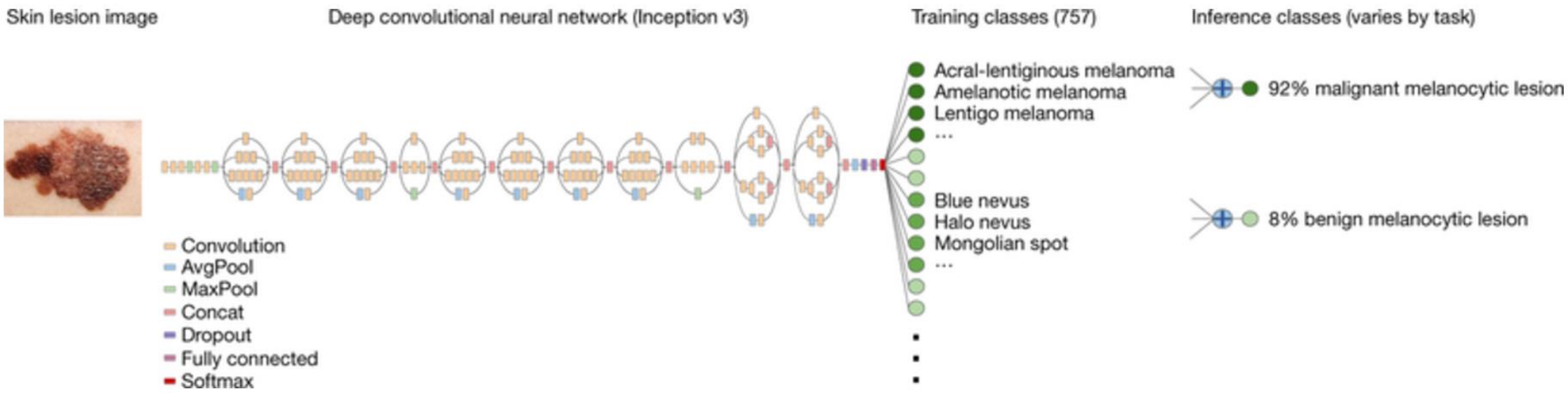
Deep learning algorithm diagnoses skin cancer as well as seasoned dermatologists

By Jessica Hall on January 25, 2017 at 1:25 pm | 16 Comments

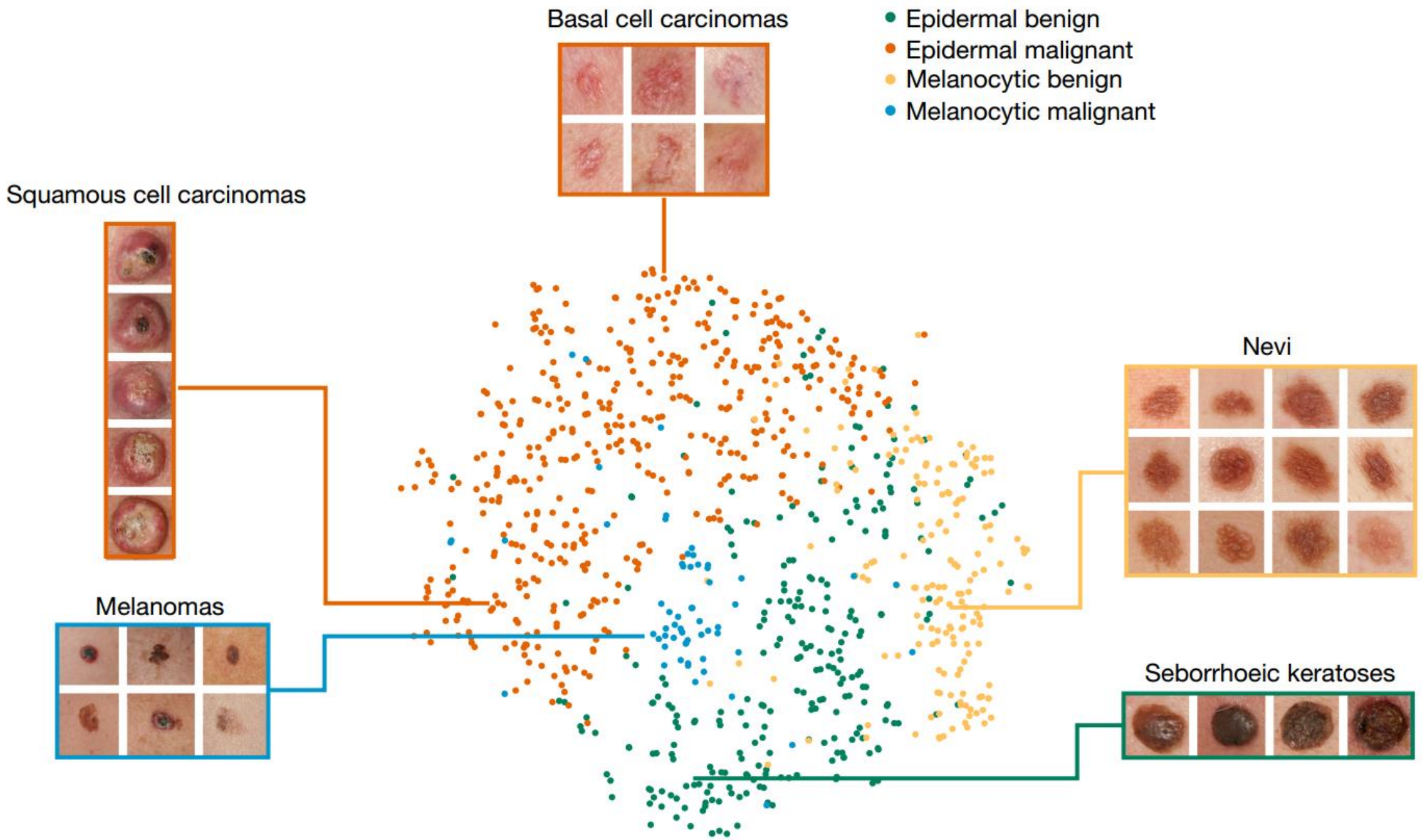
422 shares



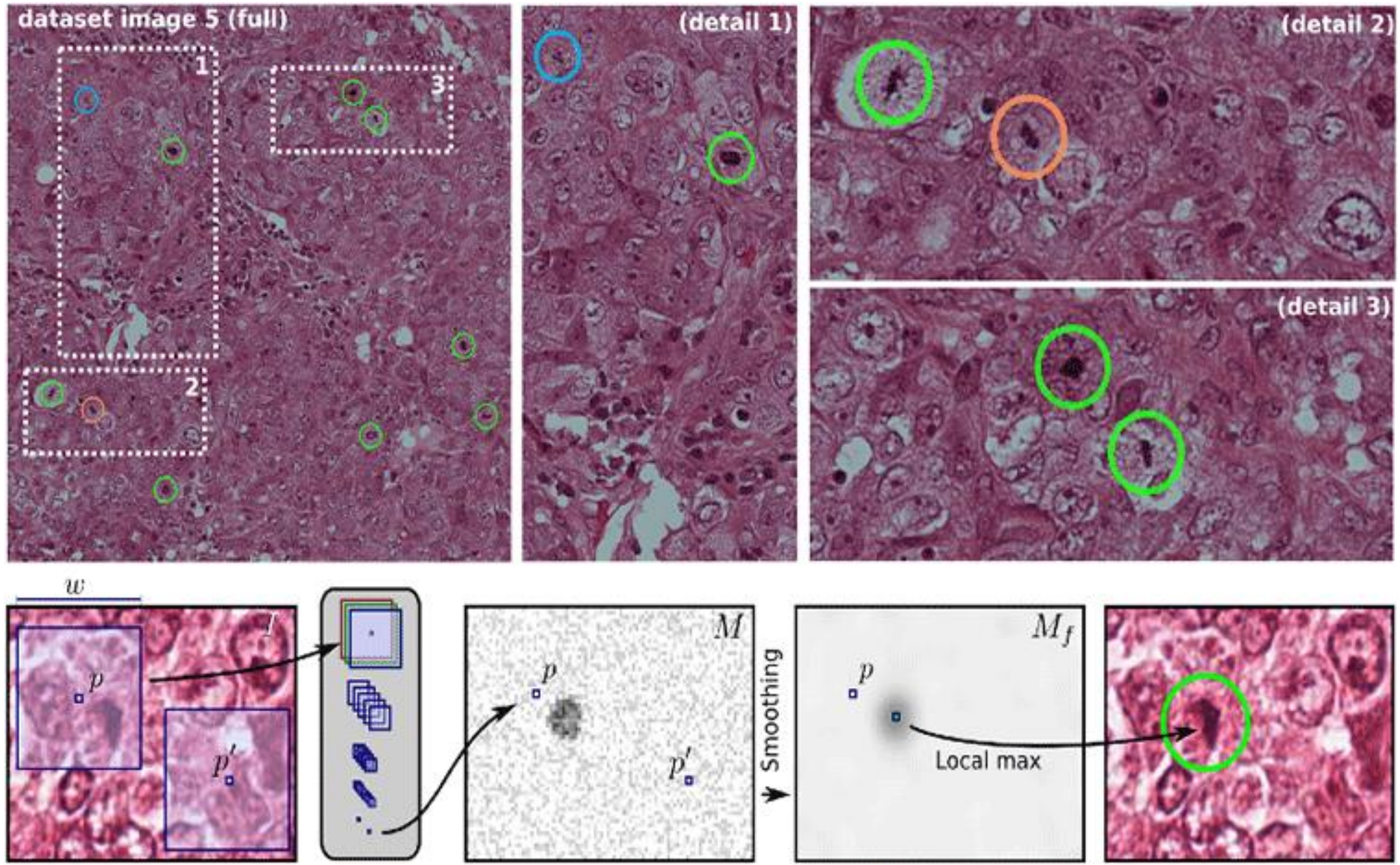
Thrun et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, January 2017 DOI: 10.1038/nature21056



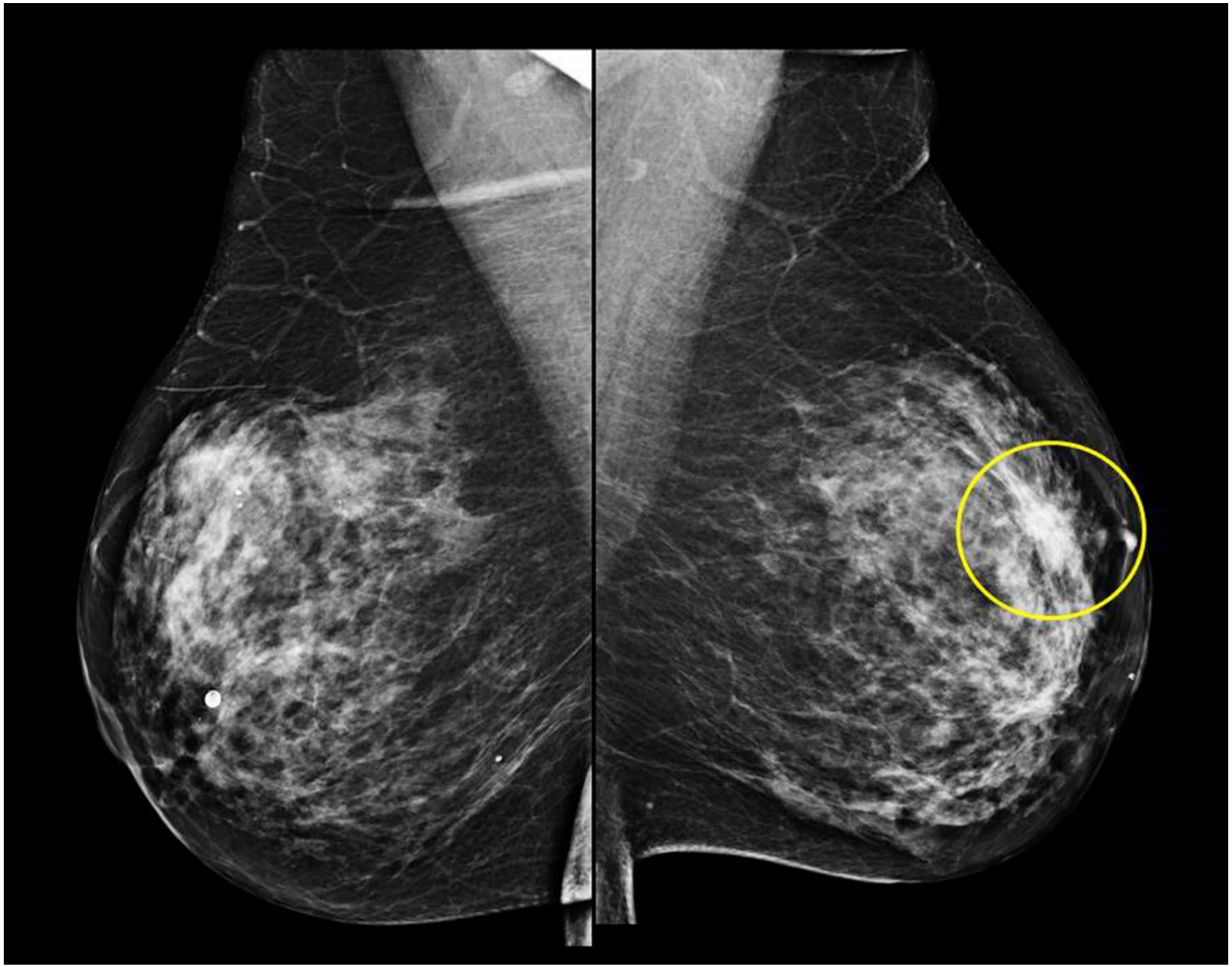
Thrun et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, January 2017 DOI: 10.1038/nature21056

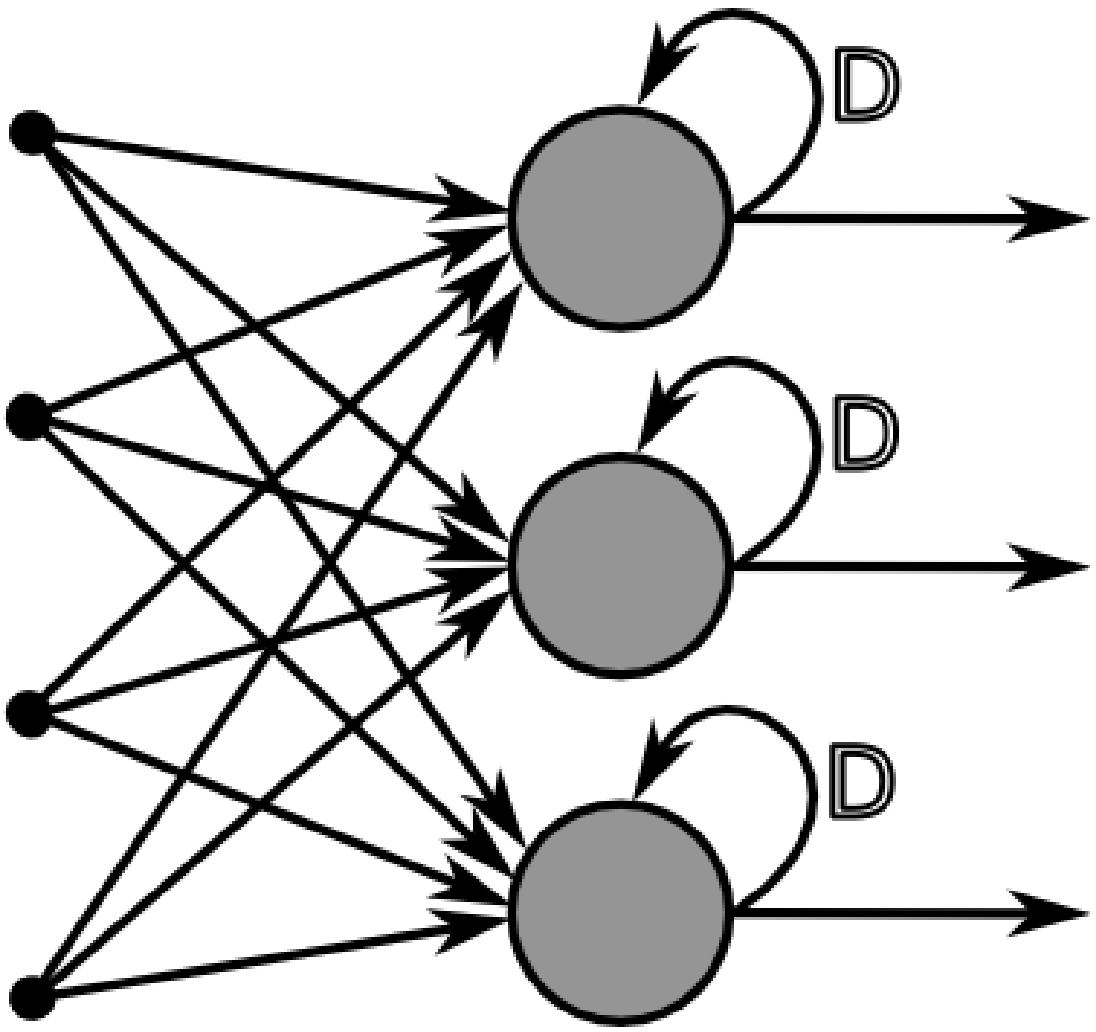


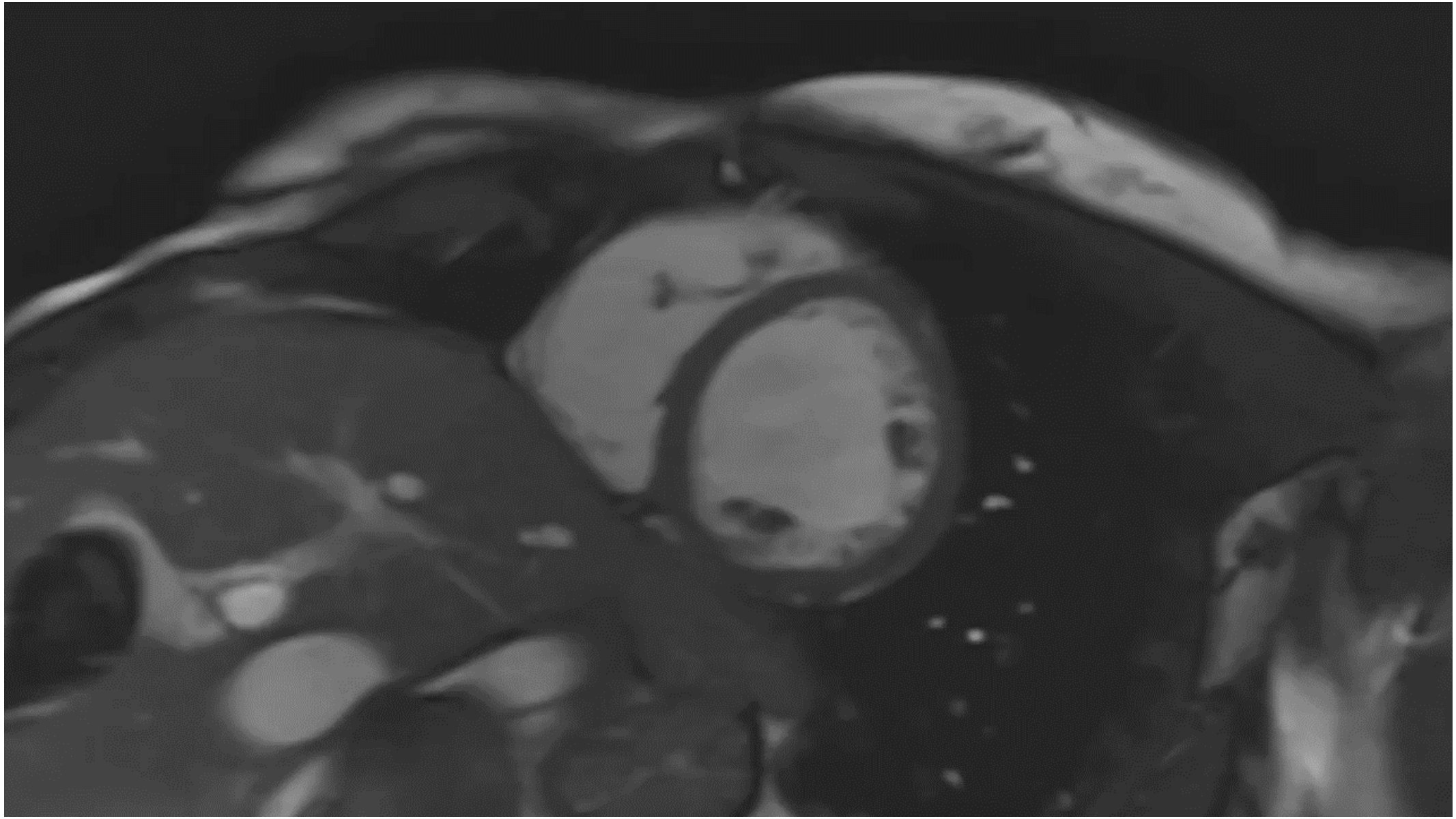
Thrun et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature, January 2017 DOI: 10.1038/nature21056



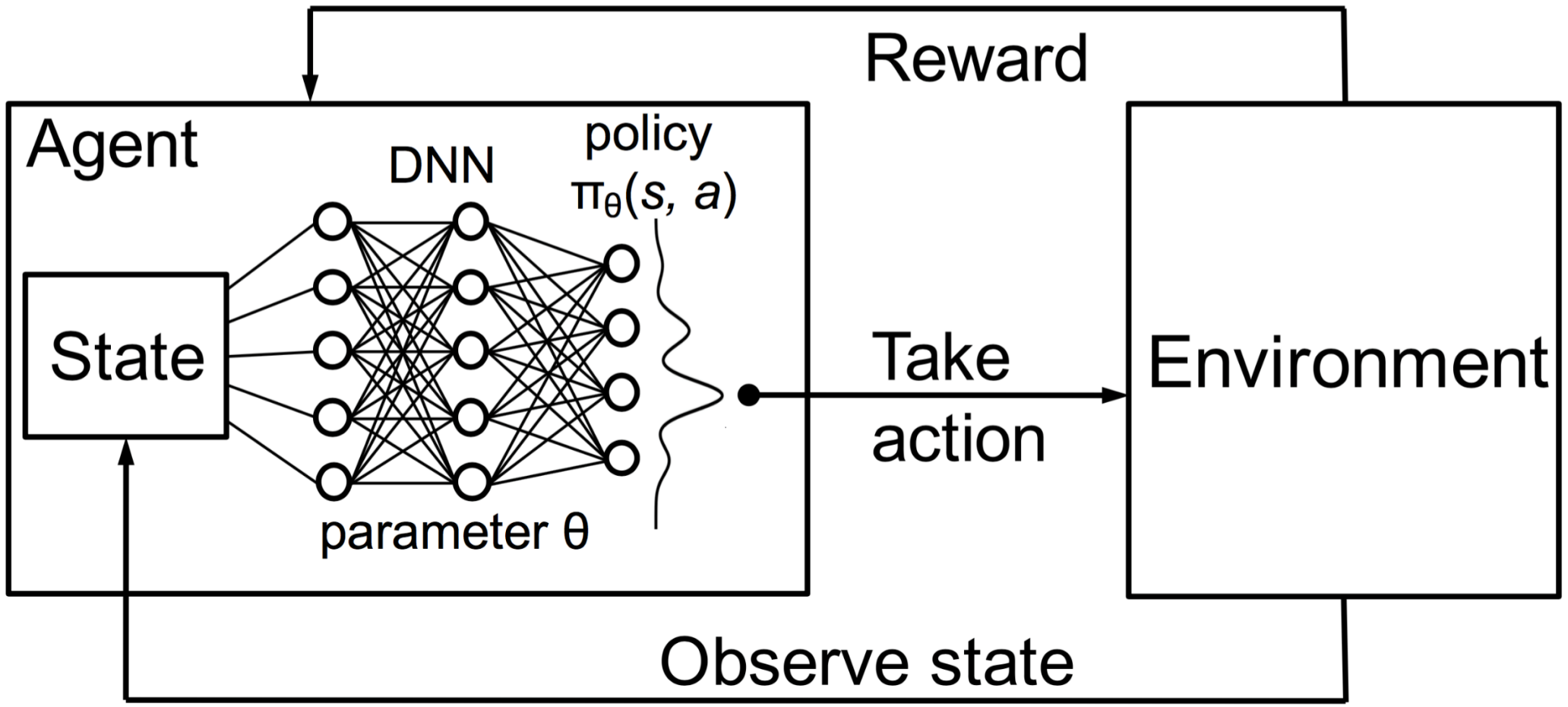
http://people.idsia.ch/~juergen/deeplearningwinsMICCAIgrandchallenge.html




D
D
D

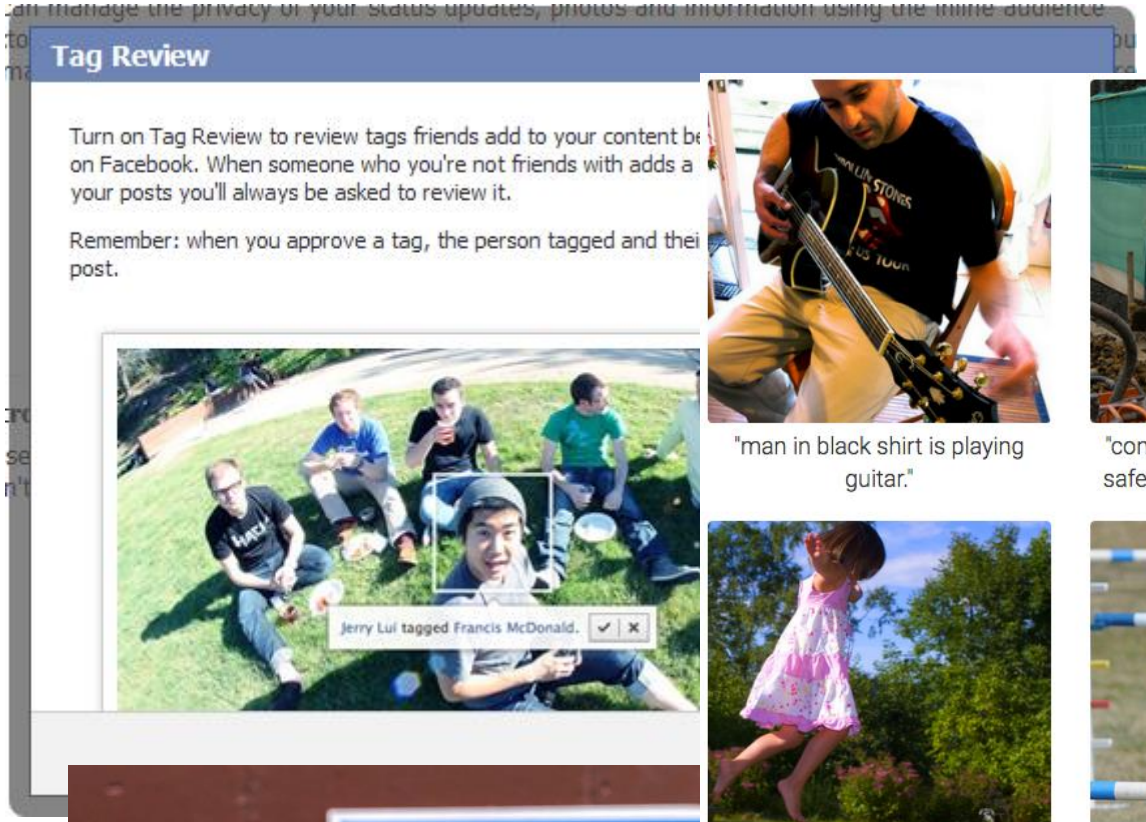


https://www.kaggle.com/c/second-annual-data-science-bowl/data



http://people.csail.mit.edu/hongzi/

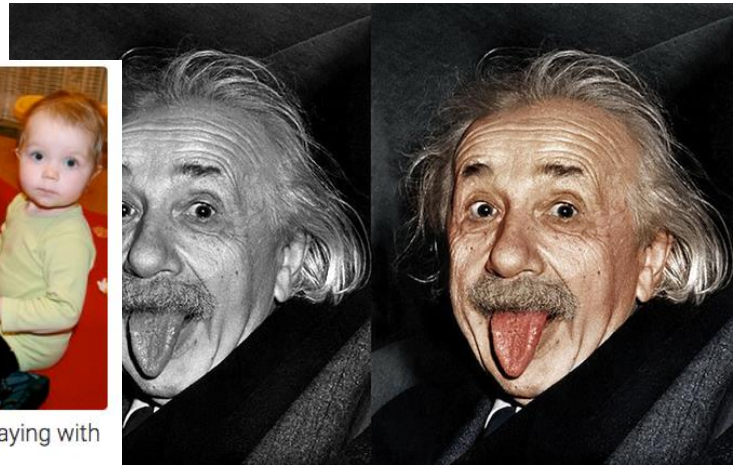
[redacted]



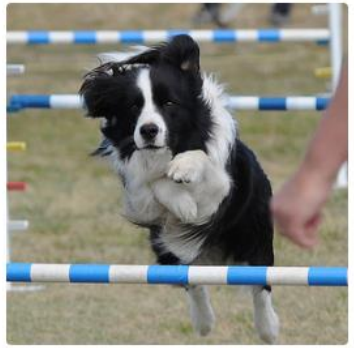
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

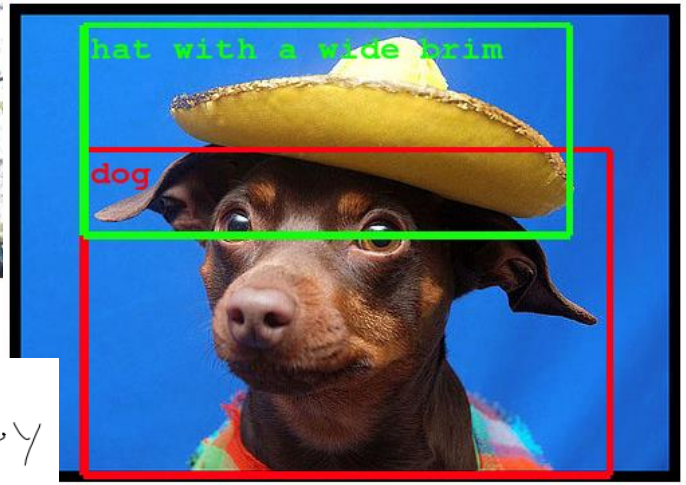


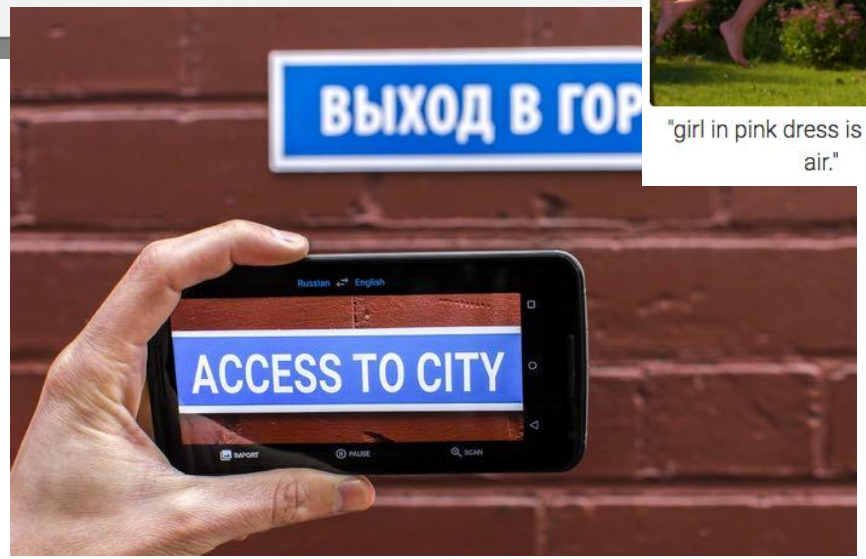
"girl in pink dress is jumping in air."



"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."





Machine Learning Mastery

Madine Learning Mastery

Htachnne Learning Mastery