Supervised Learning - Regression

WHAT IS REGRESSION?

Regression: Predict a numerical outcome ("dependent variable") from a set of inputs ("independent variables").

Statistical Sense: Predicting the expected value of the outcome.

Casual Sense: Predicting a numerical outcome, rather than a discrete one.

WHAT IS REGRESSION?

How many patients will come to the emergency unit on a Sunday evening? (**Regression**)

Is this histopathological image classified as "cancer" or "non-cancer" type? (**Classification**)

How many days will this patient spend in the hospital? (**Regression**)

LINEAR REGRESSION (SUPERVISED)

Regression algorithms can be used for example when some continuous value needs to be computed as compared to classification where the output is categorical.

So whenever there is a need to predict some future value of a process which is currently running, regression algorithm can be used.

Operating on a two dimensional set of observations (two continuous variables), simple linear regression attempts to fit, as best as possible, a line through the data points.

The regression line (our model) becomes a tool that can help uncover underlying trends in our dataset.

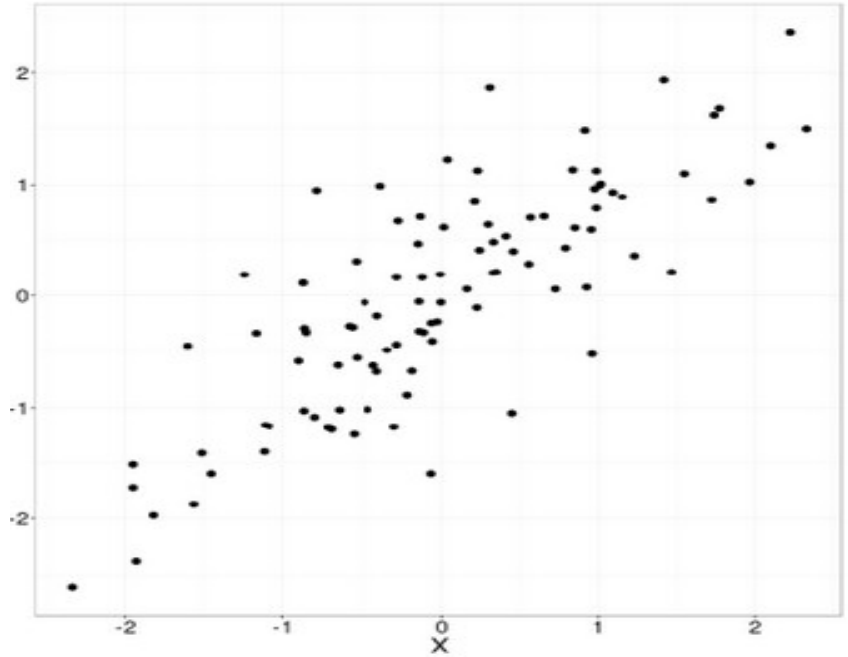
The regression line, when properly fitted, can serve as a predictive model for new events.

Linear Regressions are however unstable in case features are redundant, i.e. if there is multicollinearity

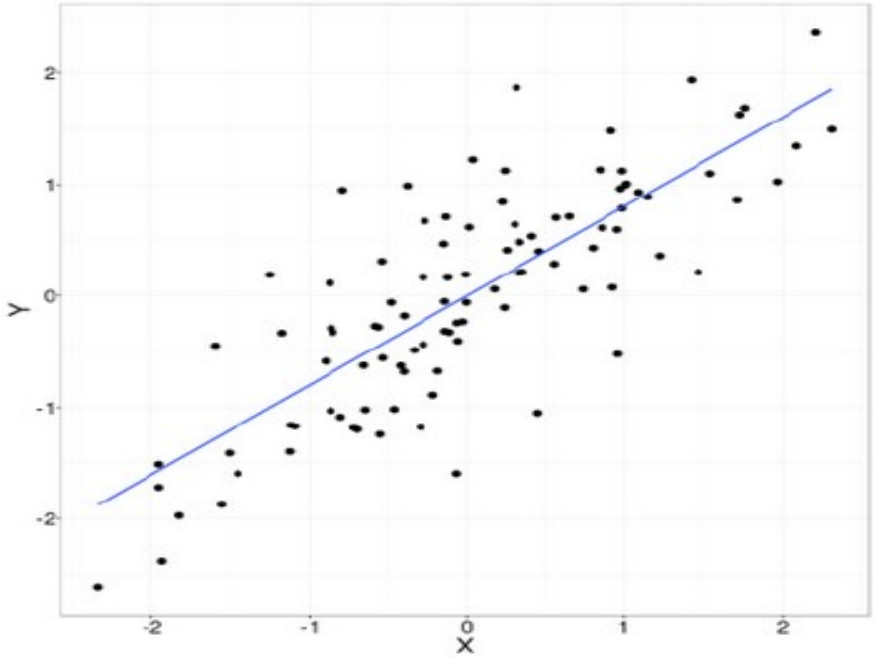
Example where linear regression can be used is:

- predicting drug resistance by correlating genotypic information with phenotypic profiles

APPLYING LINEAR REGRESSION



Scatterplot of our dataset.



Fitting of the regression line (blue).

LINEAR REGRESSION – PROS AND CONS

Pros

- Easy to fit and apply
- Concise
- Less prone to over-fitting
- Interpretable

```
Call:
lm(formula = blood_pressure ~ age + weight, data = bloodpressure)

Coefficients:
(Intercept)          age       weight
    30.9941       0.8614       0.3349
```

Cons

- Can only express linear and additive relationships

LOGISTIC REGRESSION (SUPERVISED)

It is a regression model that predicts probabilities

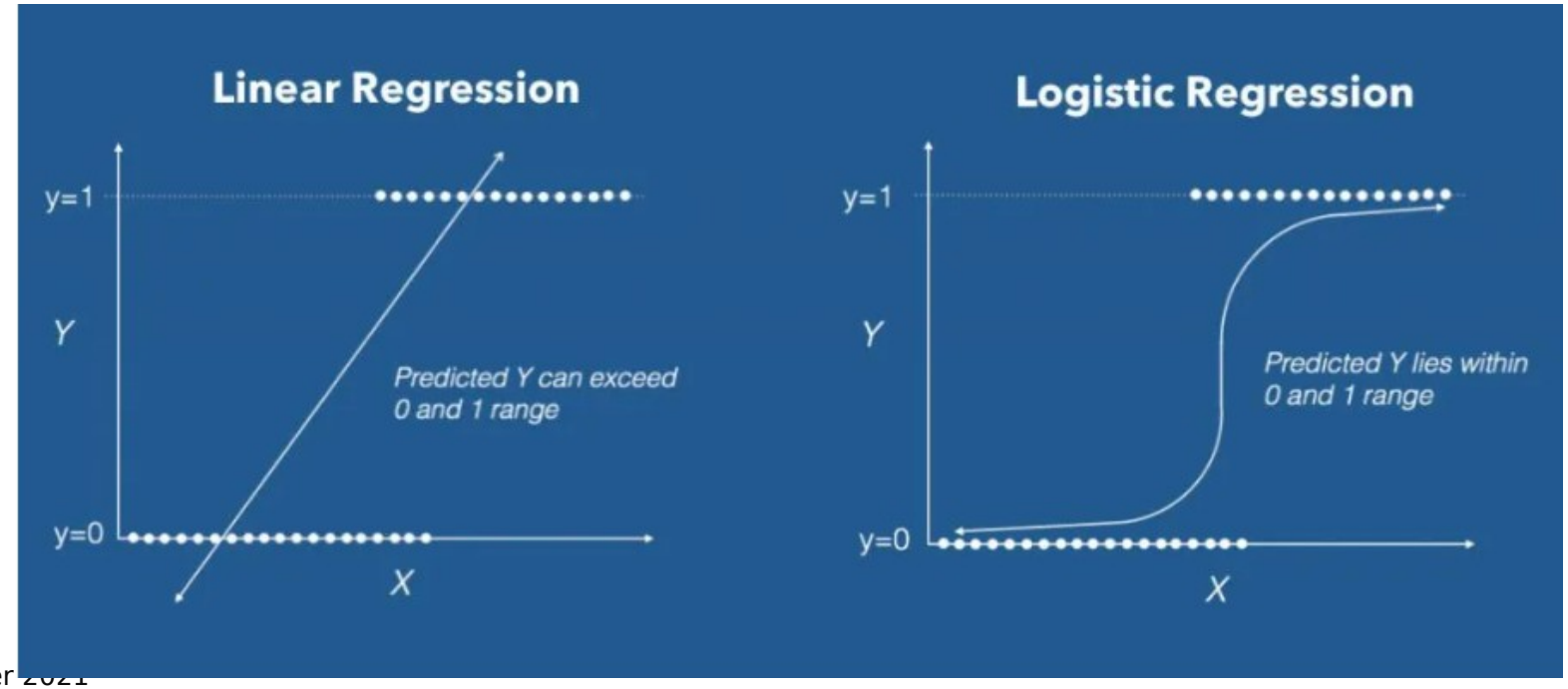
- Predicting *whether* an event occurs (yes/no): **classification**
- Predicting *the probability* that an event occurs: **regression**
- Linear regression: predicts values in $[-\infty, \infty]$
- Probabilities: limited to [0,1] interval
  - So we'll call it non-linear

Note: **Classification** refers to predicting whether an event will occur (Yes/No). While **regression** refers to the probability that an event will occur.

EXAMPLE OF LOGISTIC REGRESSION – PREDICTING DUCHENNE MUSCULAR DYSTROPHY (DMD)

We want to develop a test to detect the gene for DMD in women.

- The test uses the measurements of 2 enzymes in the blood (CK and H).
- What is the probability that a woman is a DMD carrier based on her CK and H levels?
- We cannot use linear regression (where the outcome is 0:False and 1:True), because the linear model will predict probabilities outside the range of 0 and 1.



GENERALIZED LINEAR MODELS (GLM)

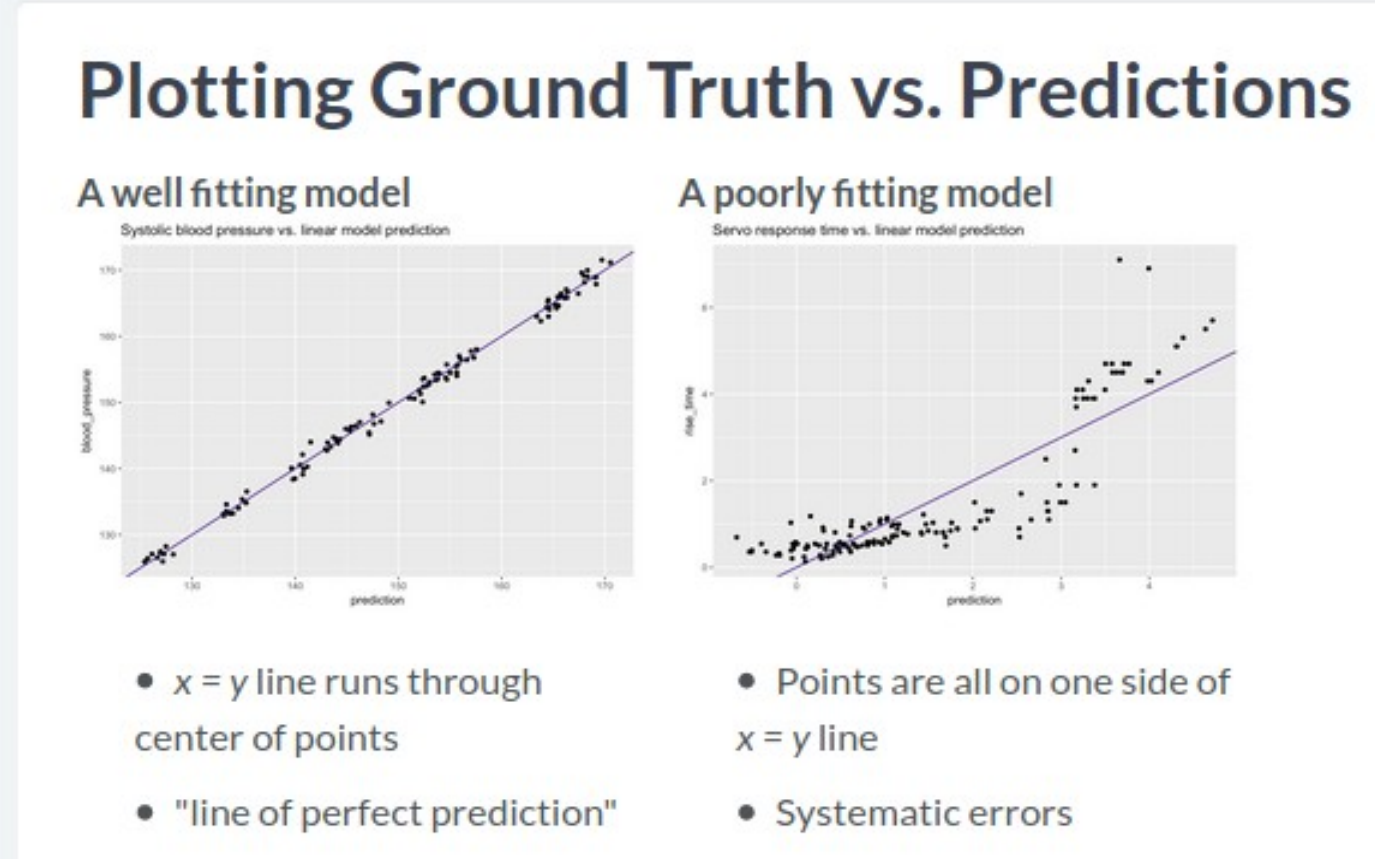
- The term ***generalized linear model*** (GLIM or GLM) refers to a larger class of models
- In these models, the response variable y_i is assumed to follow an exponential family distribution with mean μ_i, which is assumed to be some (often nonlinear) function.
- GLMs are a broad class of models that include linear regression, ANOVA, Poisson regression, log-linear models etc.
- Some of the models are:

Model	Probability Distribution
Linear Regression	Normal
Logistic Regression	Binomial
Poisson Regression	Poisson

Evaluating a Regression Model

EVALUATING OUR REGRESSION MODEL GRAPHICALLY

First of all we can visualize our ground truths vs the predicted values to see how well our model has performed the predictions.

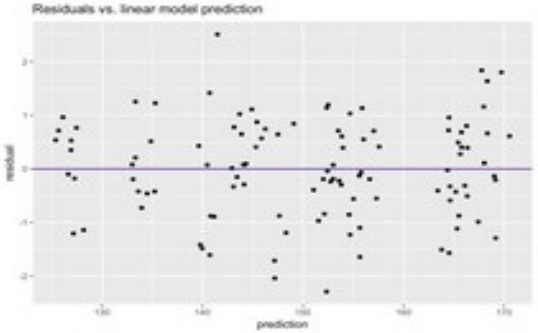


EVALUATING OUR REGRESSION MODEL GRAPHICALLY

Secondly we can also visualize the residuals against the predictions

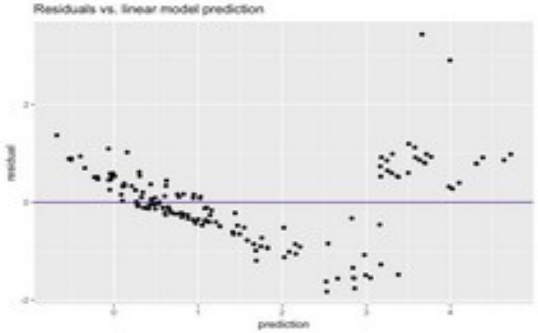
The Residual Plot

A well fitting model



- Residual: actual outcome - prediction
- Good fit: no systematic errors

A poorly fitting model



- Systematic errors

EVALUATION OF OUR REGRESSION MODEL – USING RMSE (ROOT MEAN SQUARE ERROR)

$$RMSE = \sqrt{\overline{(pred - y)^2}}$$

where

- $pred - y$: the error, or residuals vector
- $\overline{(pred - y)^2}$: mean value of $(pred - y)^2$

EVALUATION OF OUR REGRESSION MODEL – USING R^2

Coefficient of Determination or R^2 is another metric used for evaluating the performance of a regression model.

It helps us to compare our current model with a constant baseline and tells us how much our model is better.

The constant baseline is chosen by taking the mean of the data and drawing a line at the mean.

R^2 is a scale-free score that implies it doesn't matter whether the values are too large or too small, the R^2 will always be less than or equal to 1.

The closer the value of R^2 to 1, the better is our model

EVALUATION OF OUR REGRESSION MODEL – USING R^2

Calculating R^2

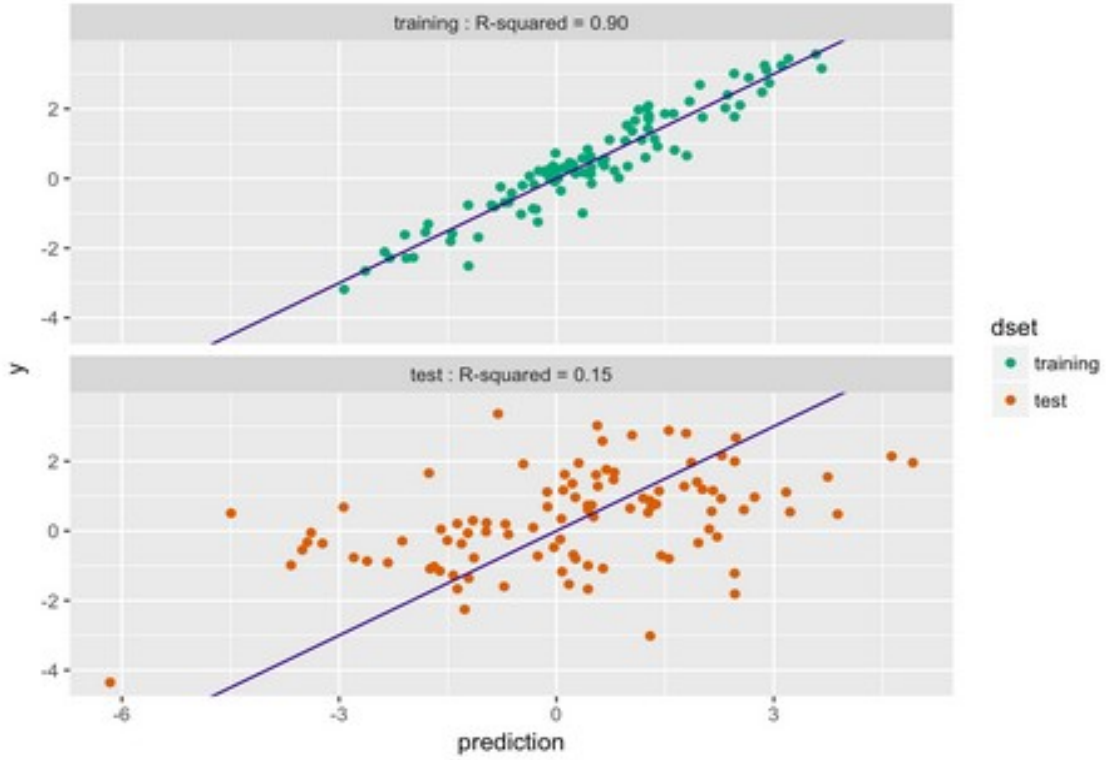
R^2 is the *variance explained by the model.*

$$R^2 = 1 - \frac{RSS}{SS_{Tot}}$$

where

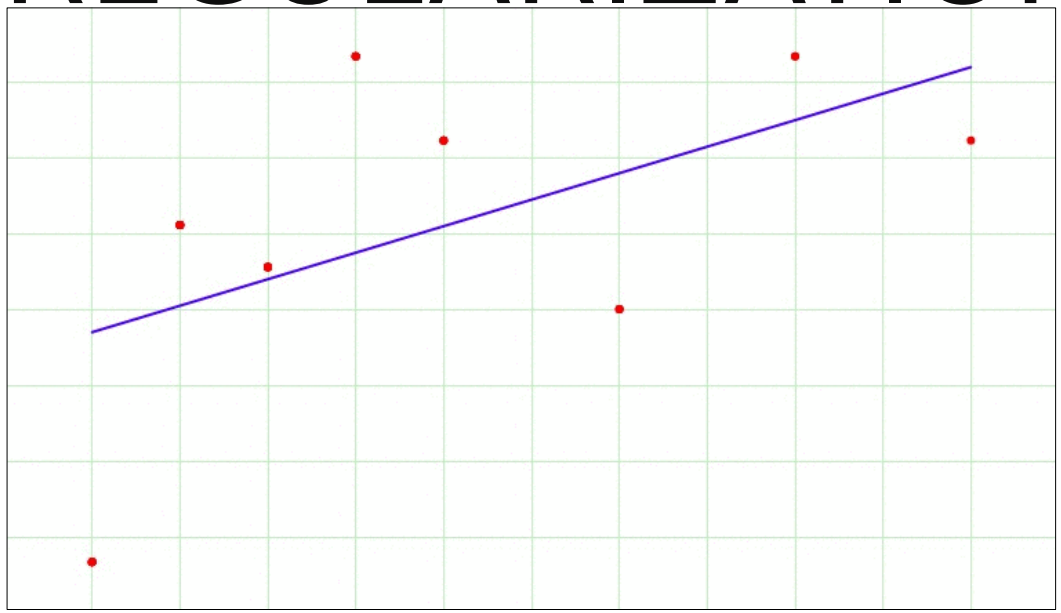
- $RSS = \sum (y - prediction)^2$
  - Residual sum of squares (variance from model)
- $SS_{Tot} = \sum (y - \overline{y})^2$
  - Total sum of squares (variance of data)

REGRESSION – **PROPERLY TRAINING A MODEL**

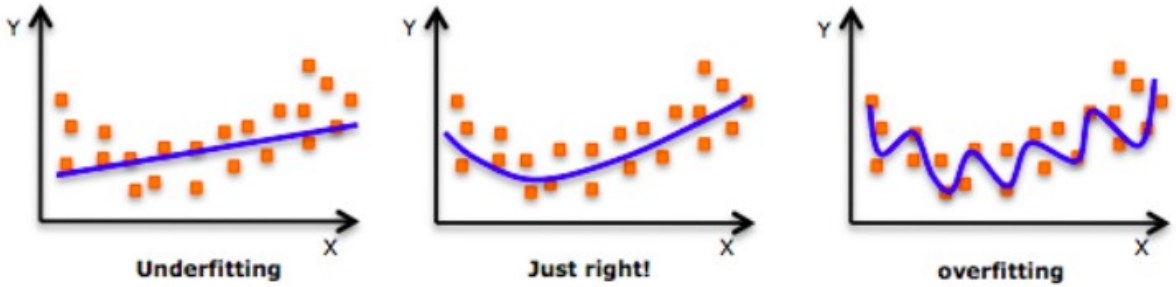


- Training R^2: 0.9; Test R^2: 0.15 -- **Overfit**

REGRESSION – OVERFITTING AND REGULARIZATION



Source: https://towardsdatascience.com



Regularization:

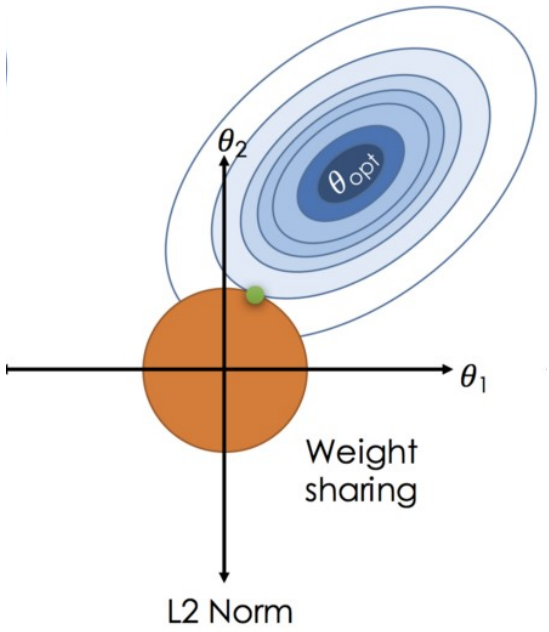
- Any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.
- Regularization technique is to penalize complex models.

Types of penalty:

- L2 Norm (Ridge regression)
- L1 Norm (Lasso Regression)
- Elastic Net regression

REGRESSION – L2 NORM (RIDGE REGRESSION)

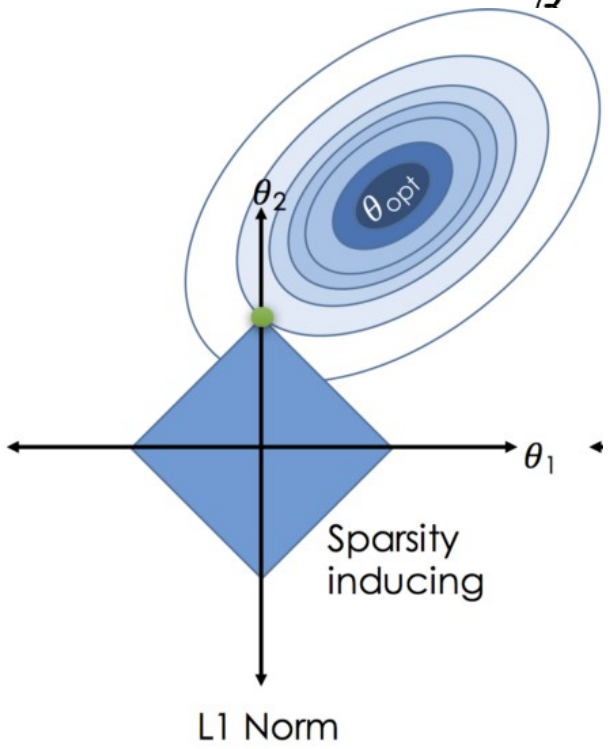
$$L_{Ridge}\left(\hat{\beta}\right)=\sum_{i-1}^{n}\left(Y_i-x_i^{'}\hat{\beta}\right)^2+\lambda\sum_{i-1}^{n}\hat{\beta}^2=\left\|y-X\hat{\beta}\right\|^2+\lambda\left\|\hat{\beta}\right\|^2$$



- **λ** becomes larger, the variance decreases, and the bias increases
- Ridge regression decreases the complexity of a model but does not reduce the number of variables, it rather just shrinks their effect.

REGRESSION – L1 NORM (LASSO REGRESSION)

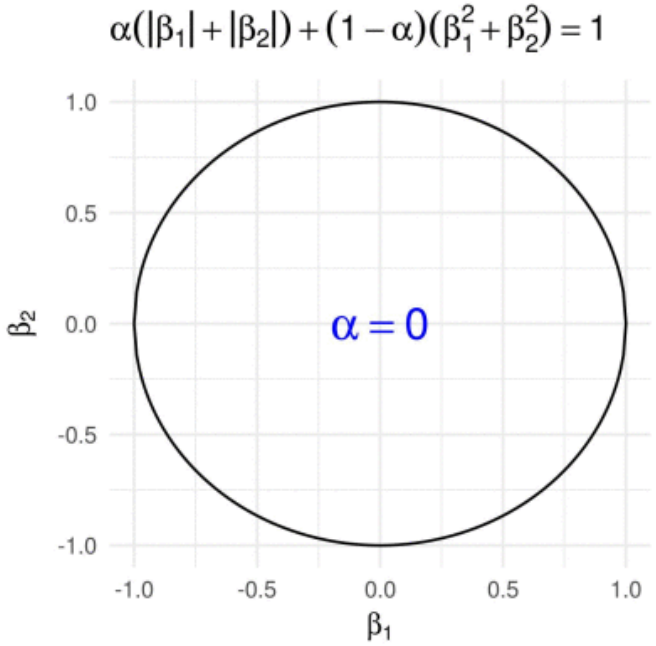
$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 \quad \text{s.t. } \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \leq t$$



- The larger the value of lambda the more features are shrunk to zero
- If p>n, the lasso selects at **most n variables**. The number of selected “genes” – for example - is bounded by the number of samples.
- **Grouped variables:** the lasso fails to do grouped selection. It tends to **select one variable from a group and ignore the others.**

REGRESSION – ELASTIC NET REGRESSION

$$\hat{\beta} = \arg\min_{\beta} |\mathbf{y} - \mathbf{X}\beta|^2, \quad \text{subject to } (1-\alpha)\,|\beta|_1 + \alpha\,|\beta|^2 \leqslant t \text{ for some } t.$$



- Elastic net is good when there are **correlations between parameters**
  - Lasso regression tends to pick just one of the correlated terms and eliminate the other
  - Ridge regression tends to shrink all of the parameters for the correlated variables together
- By combining Lasso and Ridge regression, elastic net regression groups and shrink the parameters associated with the correlated variables and leaves then in the equation or removes then once.

Shape of the penalty can give some idea of the type of shrinkage impose on the model:

- Sharp corners = sparsity
- Round corners = only shrinkage

REGRESSION – **PROPERLY TRAINING A MODEL**

In general models can perform much better on training than on data they have not yet seen.

For simple models, this difference between training data and test data results is often not severe.

But for more complex models or even for linear model with too many variables, using only the training data to evaluate the model can produce misleading results.

In the previous slide example we get the value of R^2 as 0.9 on training data but 0.15 on new data.

- It means this model was overfit.

When we have a lot of data, the best thing to do is to split your data into 2, one set to train the model and another set to test it.

When we don’t have enough data we must do cross-validation

Going into the “grey” area

SEMI-SUPERVISED MACHINE LEARNING ALGORITHMS

In supervised learning, the algorithm receives as input a collection of data points, each with an associated label, whereas in unsupervised learning the algorithm receives the data but no labels.

- The semi-supervised setting is a mixture of these two approaches: the algorithm receives a collection of data points, but only a subset of these data points have associated labels.

So, they fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – **typically a small amount of labeled data and a large amount of unlabeled data**.

The systems that use this method are able to considerably improve learning accuracy.

SEMI-SUPERVISED MACHINE LEARNING ALGORITHMS

Consider the gene finding model where the system is provided with labeled data and unlabeled data.

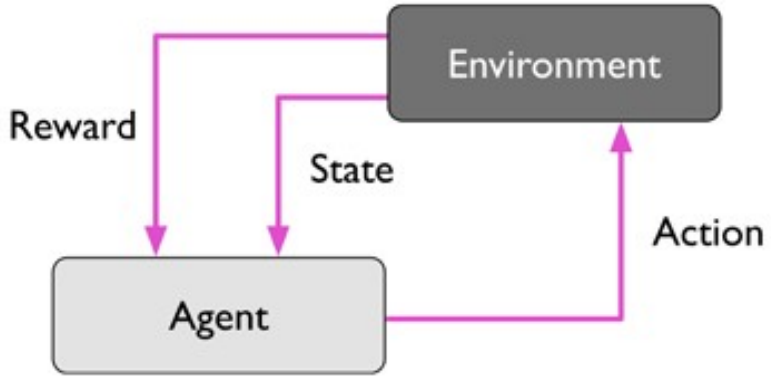
- The learning procedure begins by constructing an initial gene-finding model on the basis of the labeled subset of the training data alone.
- Next, the model is used to scan the genome, and tentative labels are assigned throughout the genome.
- These tentative labels can then be used to improve the learned model, and the procedure iterates until no new genes are found.

SEMI-SUPERVISED MACHINE LEARNING ALGORITHMS

In practice, gene-finding systems are often trained using a semi-supervised approach, in which the input is a collection of annotated genes and an unlabeled whole-genome sequence.

The semi-supervised approach can work much better than a fully supervised approach because the model is able to learn from a much larger set of genes — all of the genes in the genome — rather than only the subset of genes that have been identified with high confidence.

REINFORCEMENT MACHINE LEARNING ALGORITHMS



The learning system interacts with the environment by producing actions and discovers errors or rewards.

- The goal is to develop a system (agent) that improves its performance based on interactions with its environment.

Through its interaction with the environment, an agent can then use reinforcement learning to learn a series of actions that maximizes this reward via an exploratory trial-and-error approach or deliberative planning.

REINFORCEMENT MACHINE LEARNING ALGORITHMS

The idea behind ***Reinforcement Learning*** is that an agent will learn from the environment by interacting with it and receiving rewards for performing actions.

Learning from interaction with the environment comes from our natural experiences.

- Consider a child in a living room who sees a fireplace and approaches it.
- It's warm, it's positive, the child feels good (*Positive Reward* +1) and understands that fire is a positive thing.
- Next he tries to touch the fire and it burns his hand (Negative reward -1). He then understands that fire is positive when he is a sufficient distance away, because it produces warmth. But getting too close to it, he will be burned.

DEEP LEARNING ALGORITHMS

Also known as deep structured learning or hierarchical learning

It is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

Can perform learning in supervised and/or unsupervised manners.

Teach computers to do what comes naturally to humans: **learn by example**

- key technology behind driverless cars, enabling them to recognize a stop sign, or to distinguish a pedestrian from a lamppost.
- **In medical Research**
  - Cancer researchers are using deep learning to automatically detect cancer cells.
  - Teams at UCLA built an advanced microscope that yields a high-dimensional data set used to train a deep learning application to accurately identify cancer cells.

DEEP LEARNING ALGORITHMS

While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

- Deep learning requires large amounts of labeled data.
  - For example, driverless car development requires millions of images and thousands of hours of video.
- Deep learning requires substantial computing power.
  - High-performance GPUs have a parallel architecture that is efficient for deep learning.
  - When combined with clusters or cloud computing, this enables development teams to reduce training time for a deep learning network from weeks to hours or less.

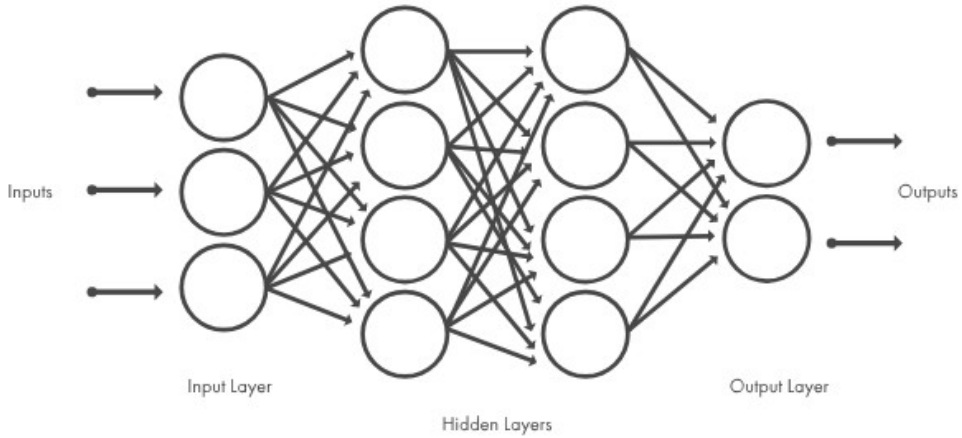
DEEP LEARNING ALGORITHMS

Most deep learning methods use neural network architectures, which is why **deep learning models** are often referred to as **deep neural networks**.

The term "**deep**" usually refers to the number of hidden layers in the neural network.

- Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150.

Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.



DEEP LEARNING ALGORITHMS

Deep learning is now one of the most active fields in machine learning and has been shown to improve performance in image and speech recognition.

The potential of deep learning in high-throughput biology is clear

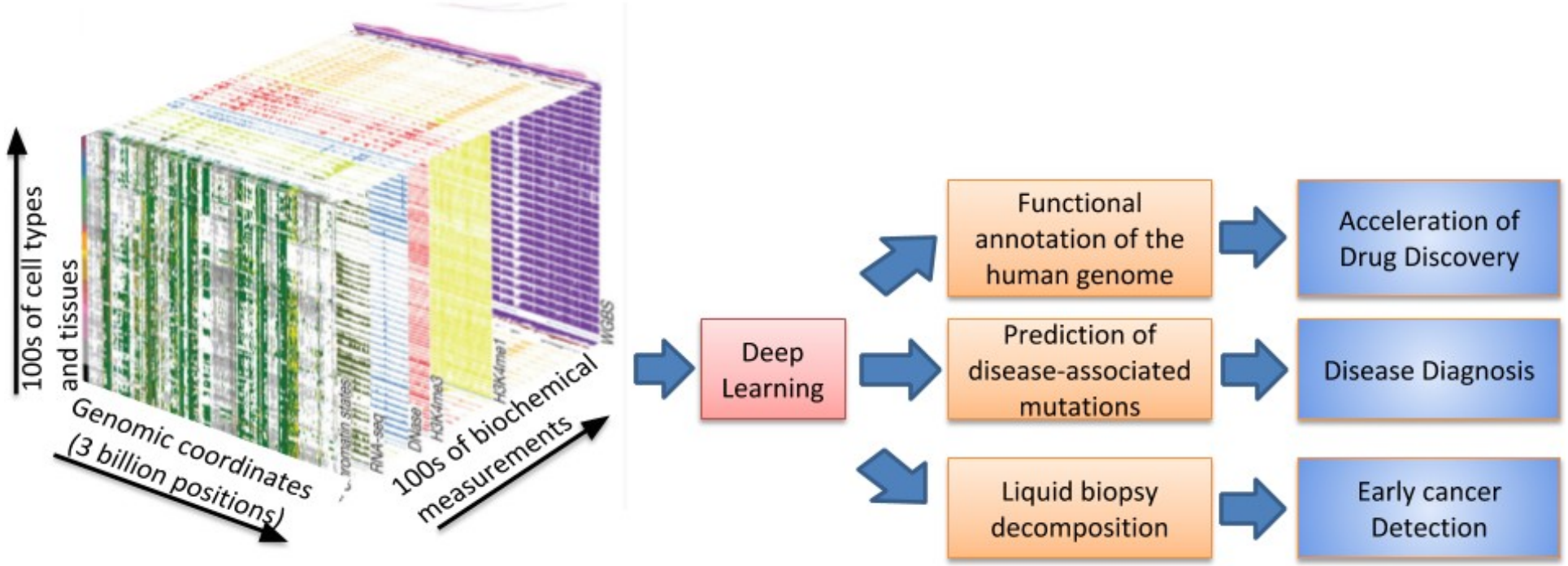
- it allows to better exploit the availability of increasingly large and high-dimensional data sets (e.g. from DNA sequencing, RNA measurements, flow cytometry or automated microscopy) by training complex networks with multiple layers that capture their internal structure

DEEP LEARNING ALGORITHMS

Example

- **Multi-label Deep Learning for Gene Function Annotation in Cancer Pathways** [Renchu Guan, Xu Wang, Mary Qu Yang, Yu Zhang, Fengfeng Zhou, Chen Yang & Yanchun Liang Scientific Reports volume 8, (2018)]
- Applied deep learning to explore full texts of biomedical articles containing detailed methodologies, experimental results, critical discussions and interpretations can be found, for the analysis of gene multi-functions relevant to cancer pathways derived from full-text biomedical publications.
  - Without the involvement of a biologist to do a feature study about the data.
- Experimental results on eight KEGG cancer pathways revealed that this new system is not only superior to classical multi-label learning models, but it can also achieve numerous gene functions related to important cancer pathways.

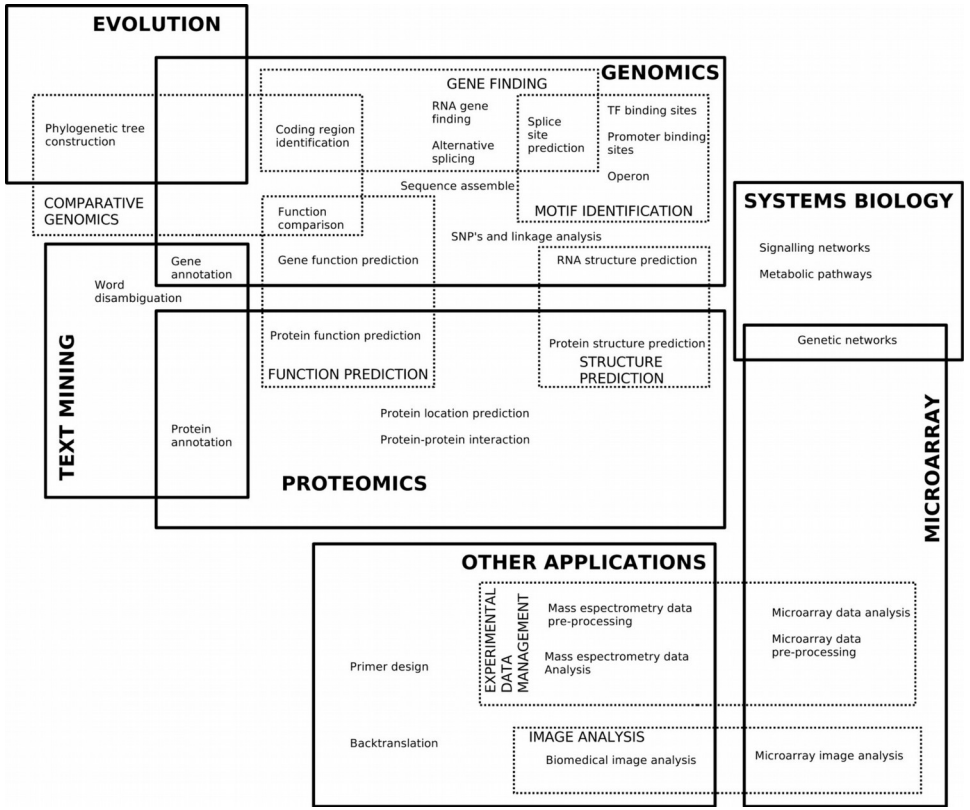
OPPORTUNITIES FOR DEEP LEARNING IN GENOMICS



https://towardsdatascience.com/opportunities-and-obstacles-for-deep-learning-in-biology-and-medicine-6ec914fe18c2

In closing

APPLICATIONS OF ML IN BIOINFORMATICS



From: Machine learning in bioinformatics
Brief Bioinform. 2006;7(1):86-112. doi:10.1093/bib/bbk007

IS THERE A PERFECT ML TECHNIQUE?

There is not one solution (one machine learning algorithm) or one approach that fits all problems.

For each problem, there is not one single solution.

WHICH TECHNIQUE TO USE?

Size, quality and nature of the data to be analysed.

The question, the answer expected, and also expected accuracy.

How the result will be used

Time and computing resources available.

Always good to check performance of different algorithms and compare results.

WHAT KIND OF DATA DO YOU HAVE?

If the data to be analysed is unlabelled and the aim is to find structure, it is an unsupervised learning problem.

If the aim is to optimize an objective function by interacting with an environment, it is a reinforcement learning problem.

When supervised learning is feasible, it is often the case that additional, unlabelled data points are easy to obtain.

How do you decide whether it’s a supervised or semi-supervised approach?

A good rule of thumb is to use semi-supervised learning if you do not have very much labelled data and you have a very large amount of unlabelled data

WHAT IS THE EXPECTED OUTPUT?

If the output of your model is a number, it is a regression problem.

- Two-class classification of gene expression data

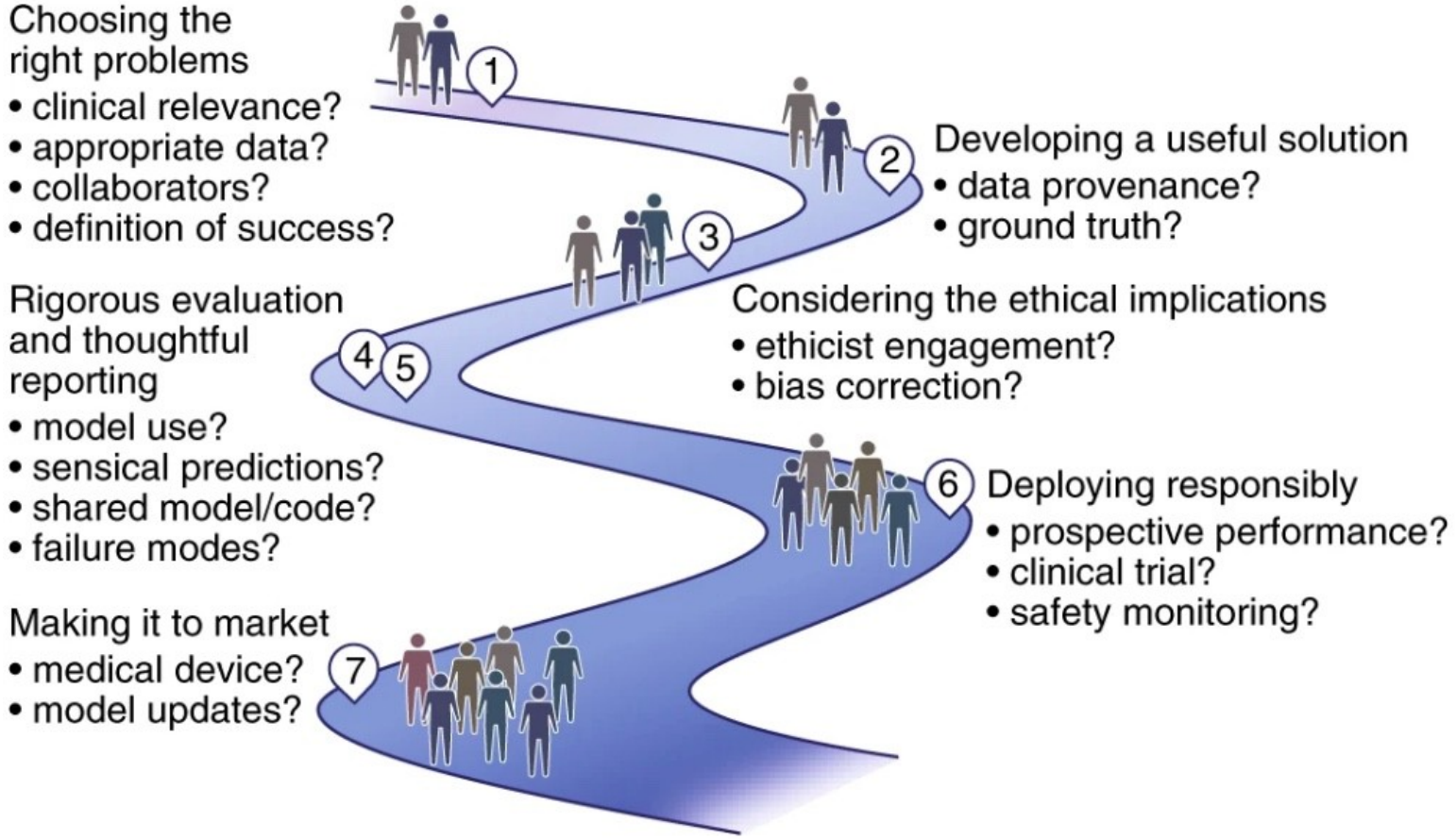
If the output of your model is a class, it is a classification problem.

- Genomic classification of AML

If the output of your model is a set of input groups, it is a clustering problem.

- Patterns in gene expression at different developmental stages of zebrafish

DO NO HARM: A ROADMAP FOR RESPONSIBLE MACHINE LEARNING FOR HEALTH CARE



https://www.nature.com/articles/s41591-019-0548-6/figures/1

TOOLS

All the methods listed above are already available either in Python, R (https://www.r-project.org/about.html) or Matlab using existing packages. Some basic code needs to be written.

If you are not used to writing code, you may use a tool like WEKA (https://www.cs.waikato.ac.nz/ml/weka/) or RapidMiner (https://rapidminer.com/) – the methods are already implemented and you simply need to load your data in either csv, arff,... format and run the selected methods.

Some useful R packages R implementing many ML techniques: https://cran.r-project.org/web/views/MachineLearning.html

SOME ONLINE RESOURCES

https://machinelearningmastery.com/start-here/

https://www.datascience.com/blog

https://www.mathworks.com/discovery/machine-learning.html

https://www.coursera.org/browse/data-science

SOURCES

http://www.sthda.com/english/articles/29-cluster-validation-essentials/97-cluster-validation-statistics-must-know-methods/#data-preparation

https://medium.mybridge.co/30-amazing-machine-learning-projects-for-the-past-year-v-2018-b853b8621ac7

Shakuntala Baichoo and Zahra Mungloo slides (H3ABionet, ML group)

NOW GO FORTH AND ML!