A Crash Course in Statistical Learning Methods

Olga Demler, PhD
Division of Preventive Medicine
Brigham and Women's Hospital
Harvard Medical School

Lecture Summary

- Linear Discriminant Analysis
- Logistic Regression
- Survival Analysis
- Shrinkage methods: Ridge Regression, Lasso, Elastic Net

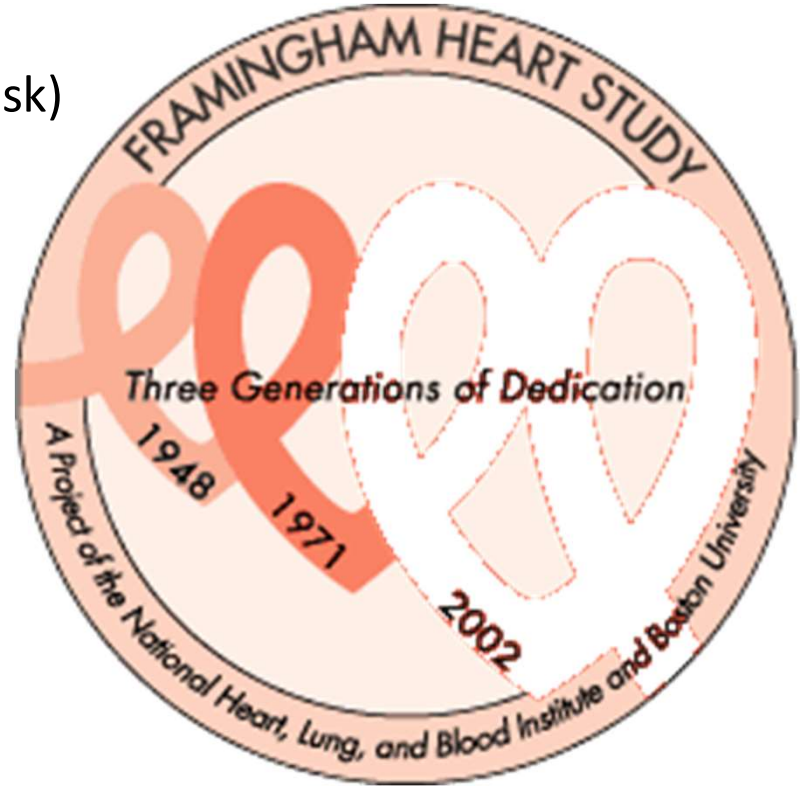
- Classification and Regression Trees
- Random Forests
- Support Vector Machines
- Brief Overview of Neural Networks

- Framingham Heart Study:

•Coronary Heart Disease (10-year risk)

•Hard Coronary Heart Disease (8-year risk)

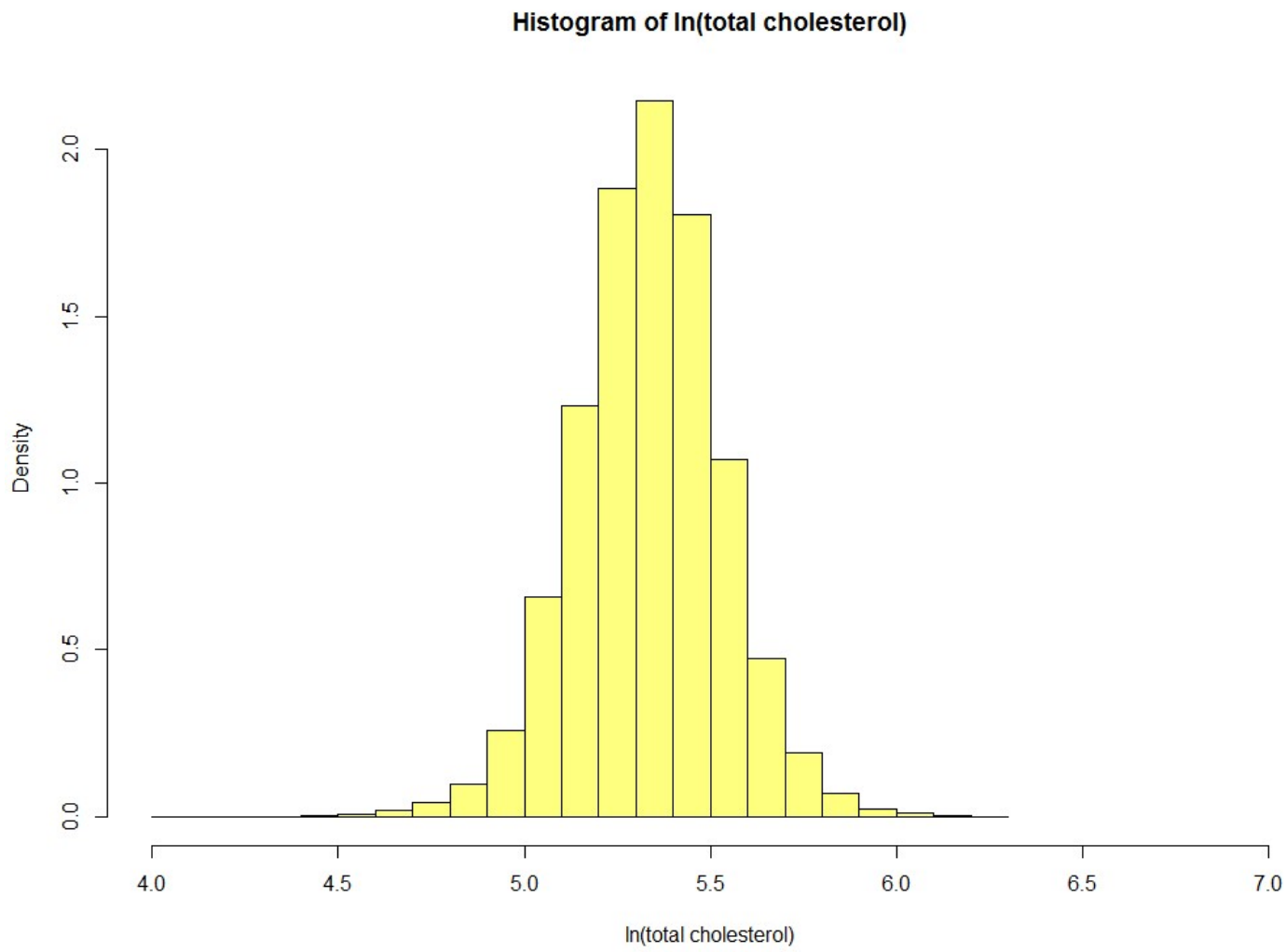
•Stroke



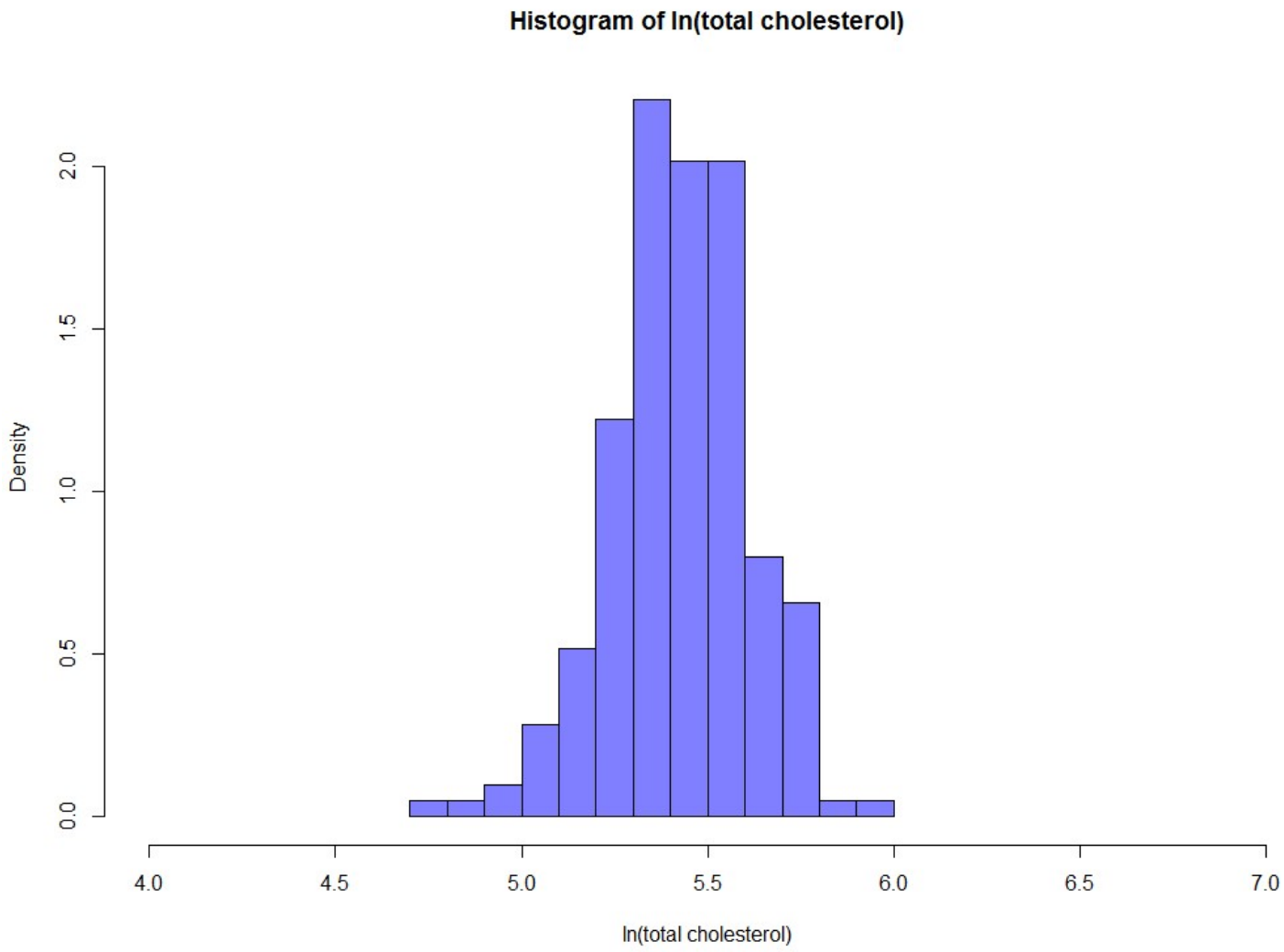
- Gail Model for 5-year risk of Breast Cancer

Linear Discriminant Analysis

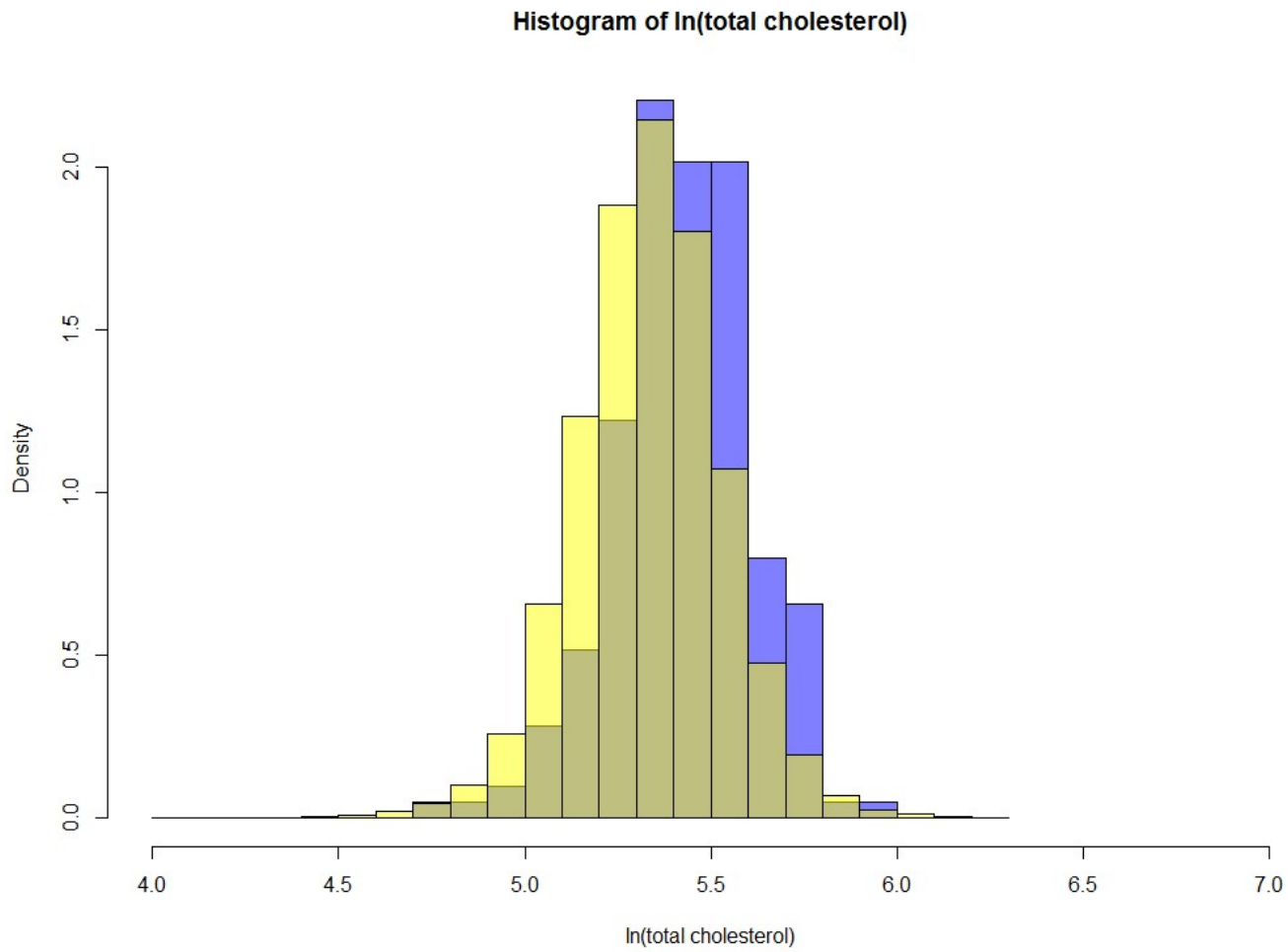
Linear Discriminant Analysis



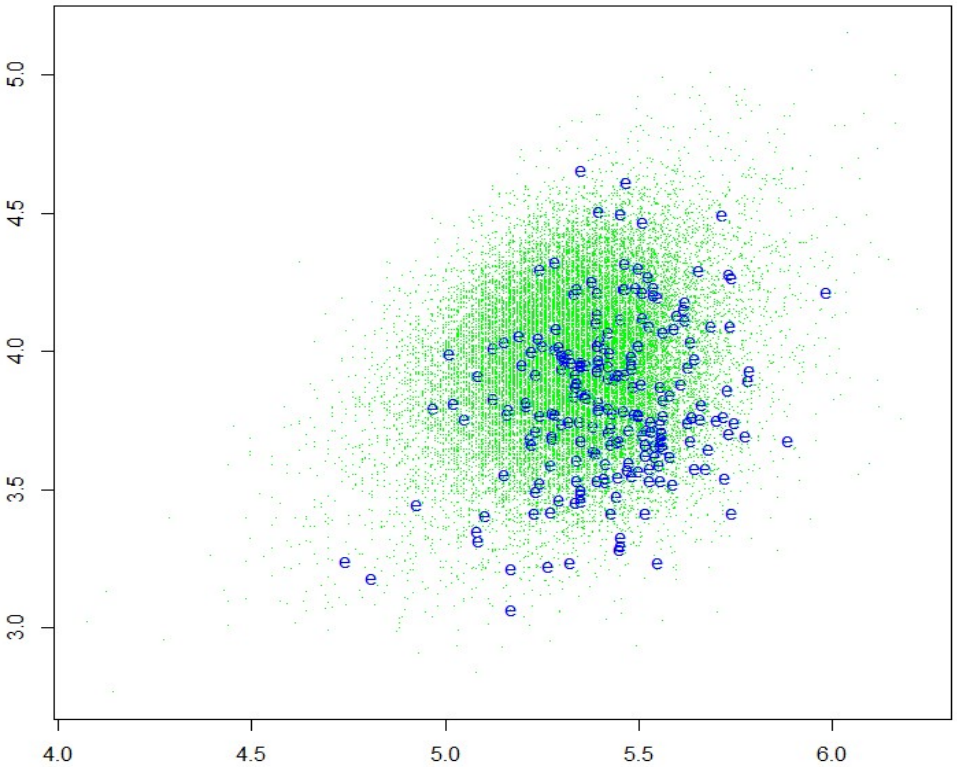
Linear Discriminant Analysis



Short review of predictive models: Linear Discriminant Analysis

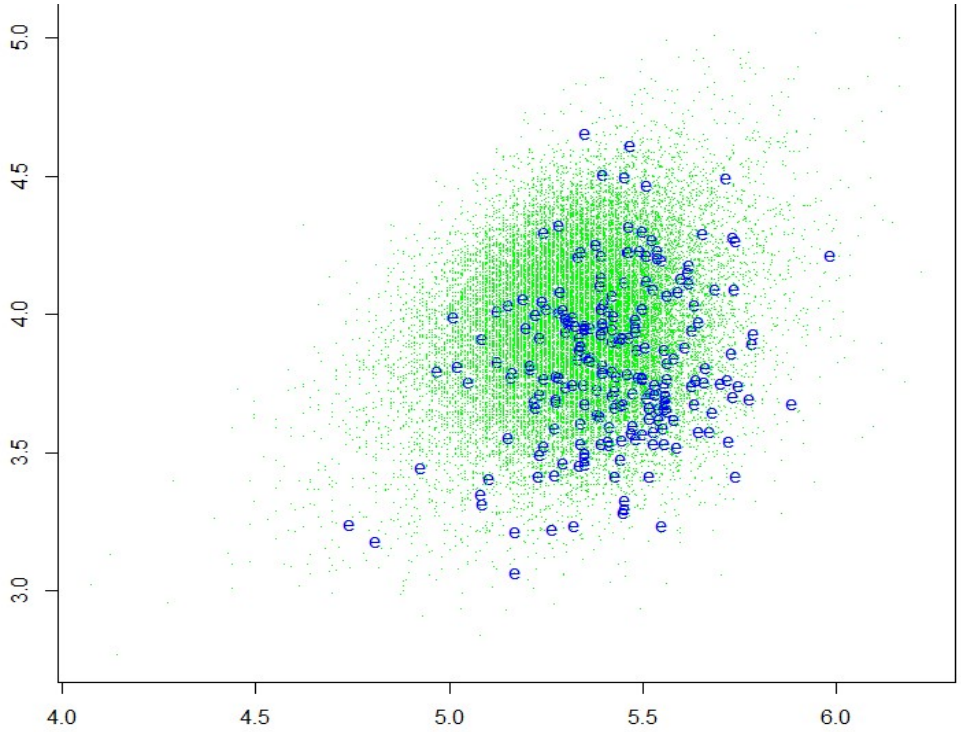


Short review of predictive models: Linear Discriminant Analysis



Short review of predictive models: Linear Discriminant Analysis

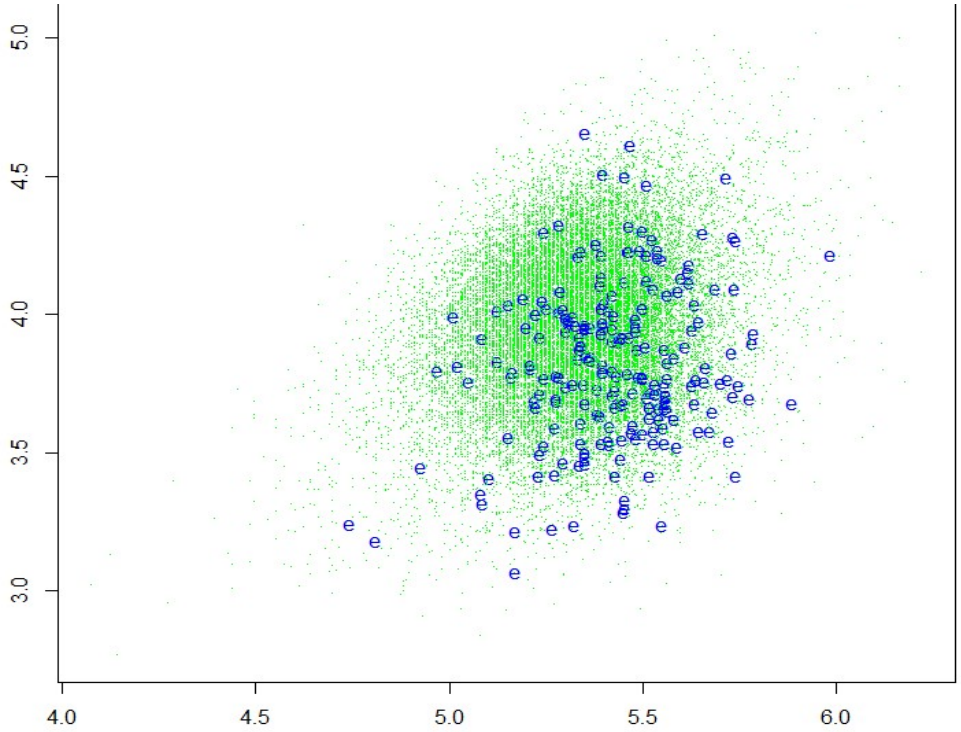
It will be reasonable to combine biomarkers into a single measure.
We would like to separate events from non-events as much as possible.



Short review of predictive models: Linear Discriminant Analysis

Risk score

It will be reasonable to combine biomarkers into a single measure.
We would like to separate events from non-events as much as possible.

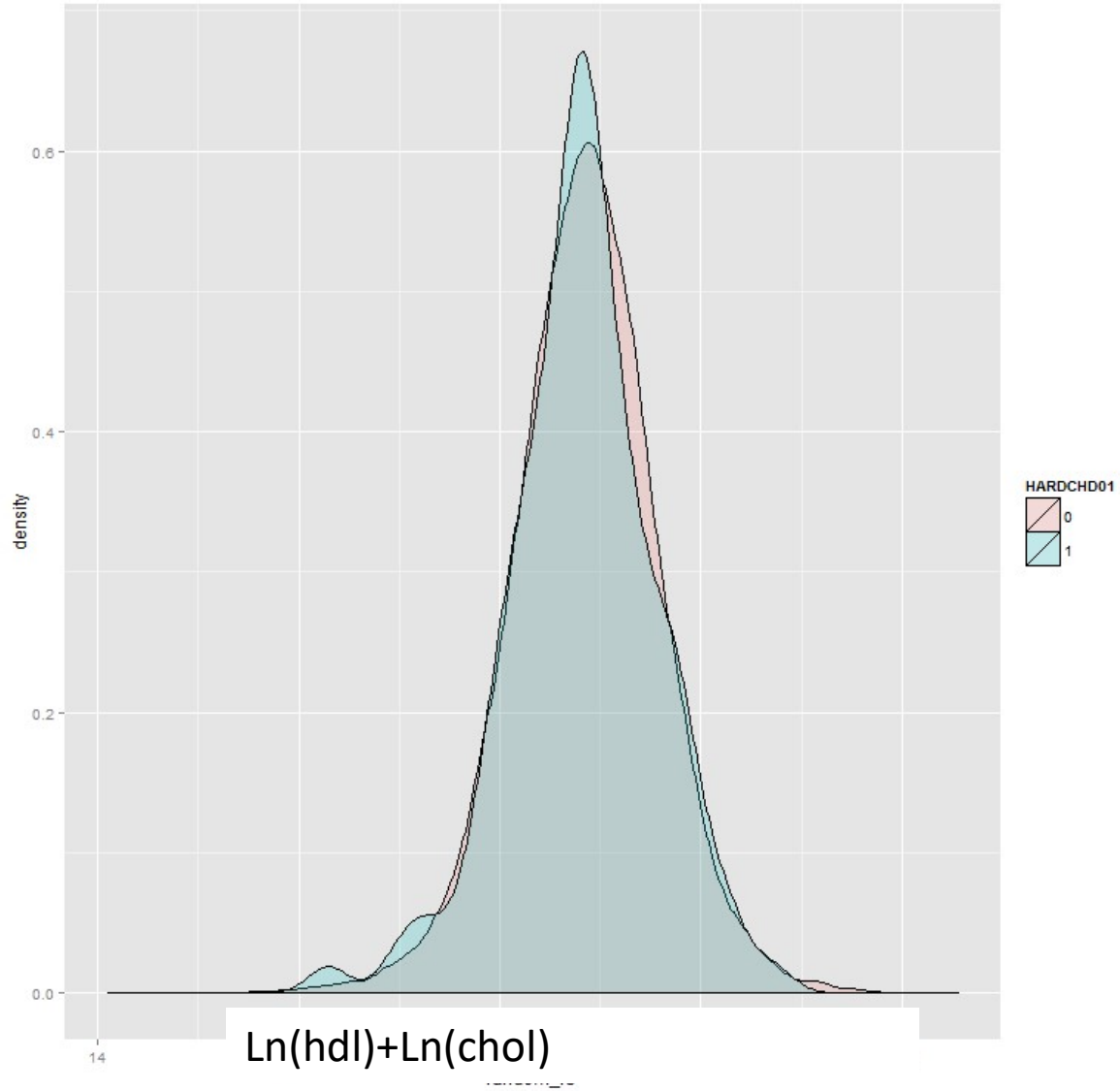


Ln(hdl)

Ln(hdl)+Ln(chol)

Ln(chol)

Short review of predictive models: Linear Discriminant Analysis



Short review of predictive models: Linear Discriminant Analysis (1936)

R.A. Fisher

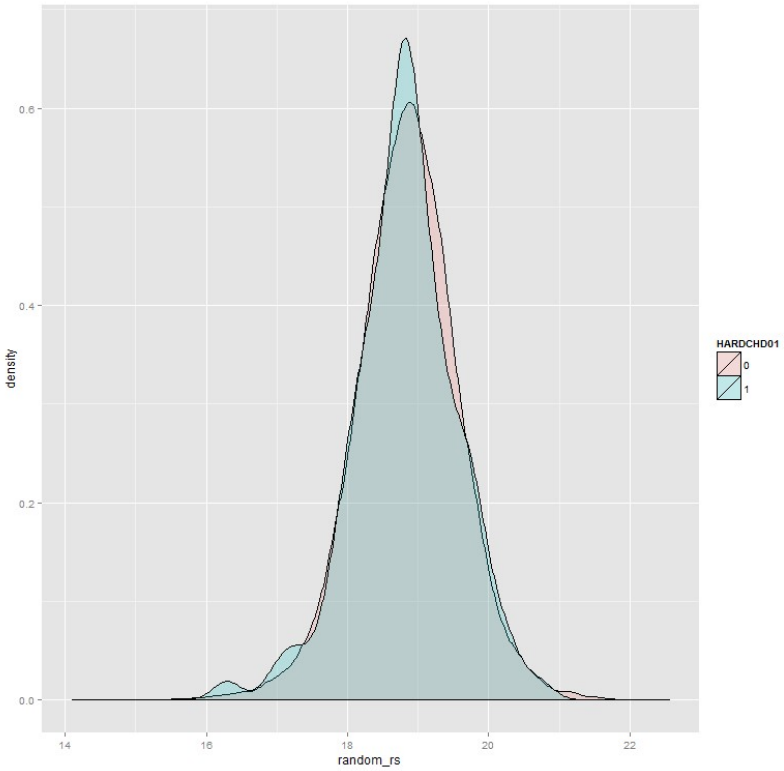
Outcome: 0 or 1 (event or non-event; 10 year CHD or no CHD in 10 years of followup)

Predictors:
Will work with any kind of predictors
Confidence Intervals and p-values are valid **only** for normally distributed predictors

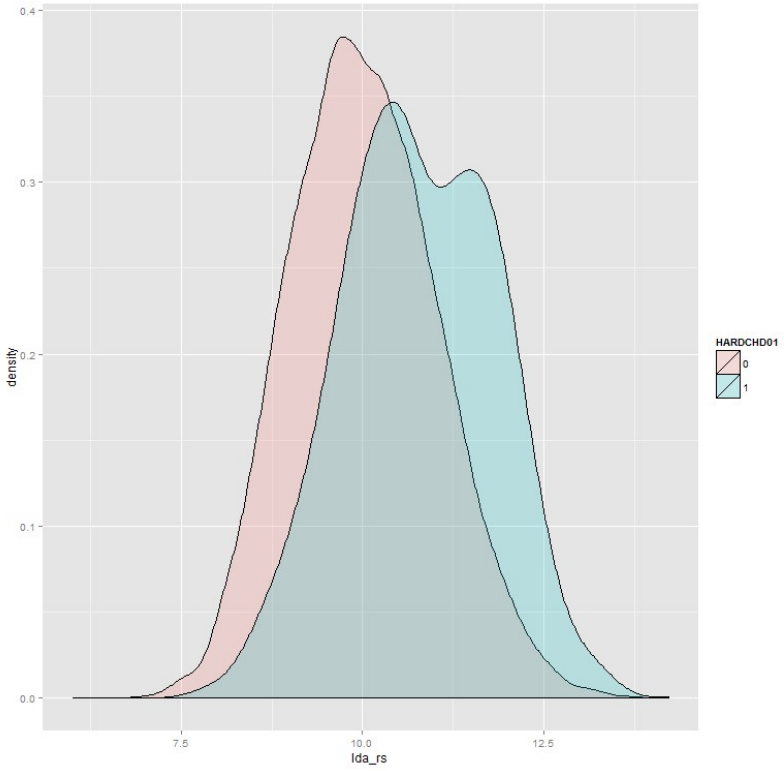
It will be reasonable to find coefficients that separate groups as much as possible.
Linear Discriminant analysis coefficients separate group mean among events from group mean among non-events.

Ln(hdl)+Ln(chol) → α_1 Ln(hdl)+ α_2 Ln(chol)

Short review of predictive models: Linear Discriminant Analysis



$$\ln(chol) + \ln(hdl)$$

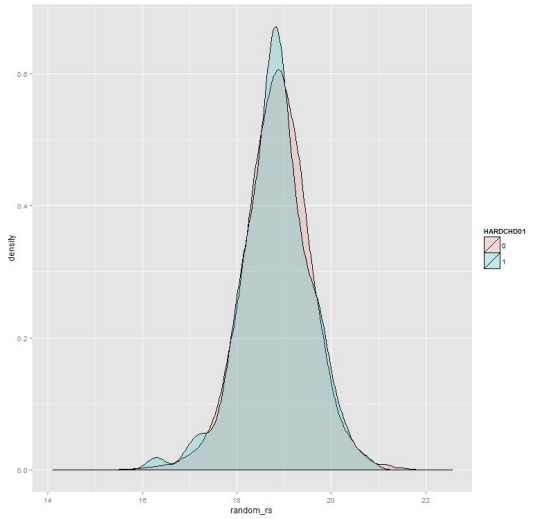


$$\alpha_1 \ln(chol) + \alpha_2 \ln(hdl)$$

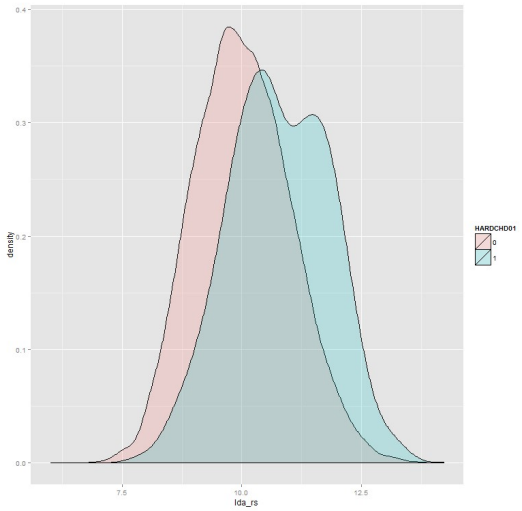
LDA

Short review of predictive models: Linear Discriminant Analysis

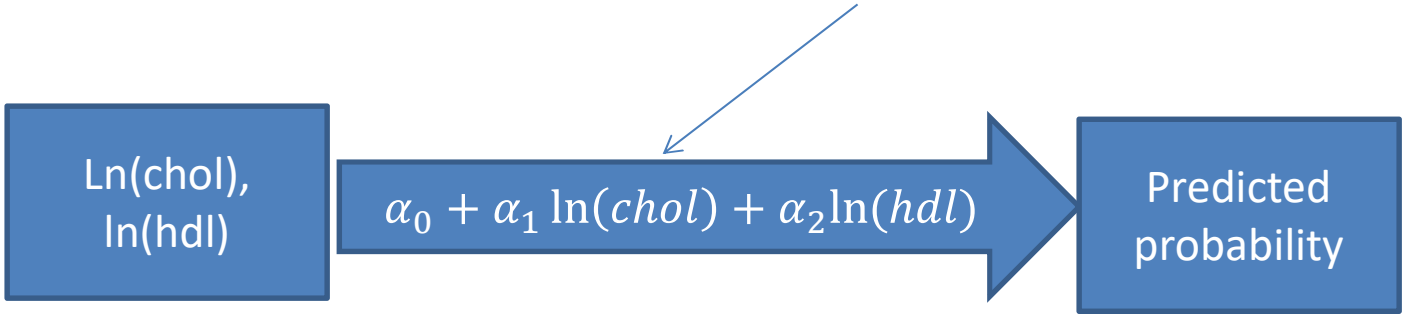
$\ln(chol) + \ln(hdl)$



$4.1\ln(chol) - 3.0\ln(hdl)$



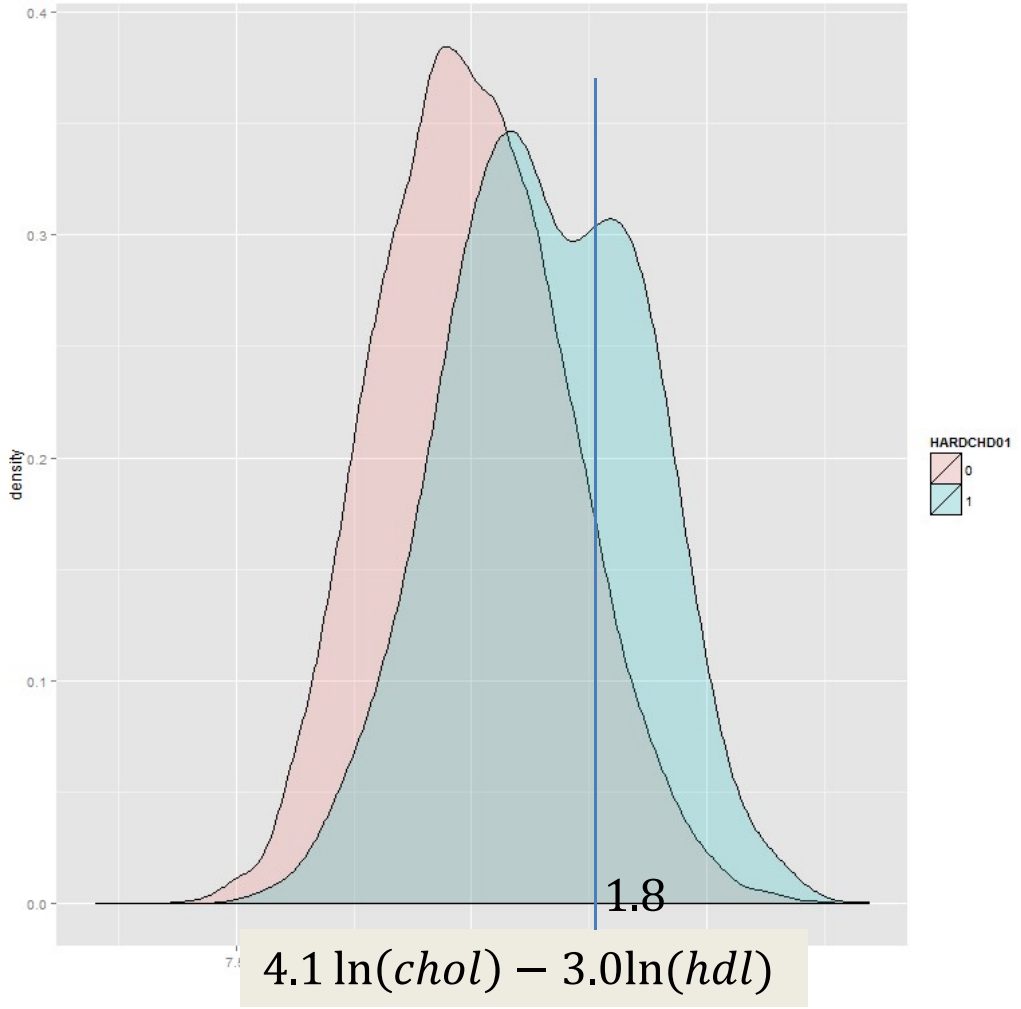
Linear Discriminant analysis coefficients separate as much as possible group mean among events from group mean among non-events.



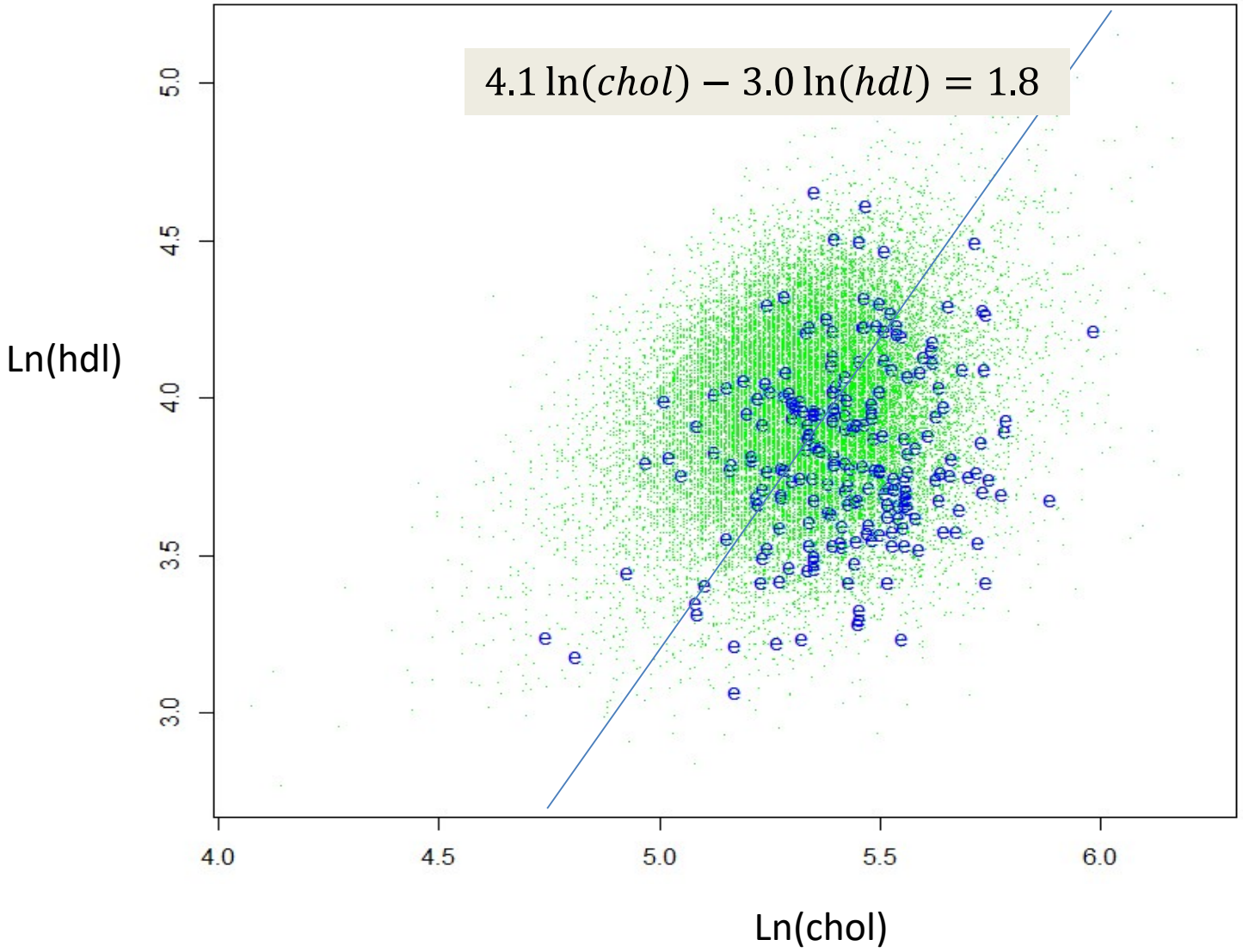
Short review of predictive models: Linear Discriminant Analysis

```
library(MASS)

lda(tbl_contin$HARDCHD ~ tbl_contin$LNCHOL + tbl_contin$LNHDL, data=tbl_contin)

lda(tbl_contin$HARDCHD ~ tbl$LNAGE + tbl$LNCHOL + tbl$LNHDL + tbl$LNSBPMED +
    tbl$TRTSBP + tbl$CURRSMK + tbl$LNCHOLAG + tbl$SMKLNAGE, data=tbl_contin)
```

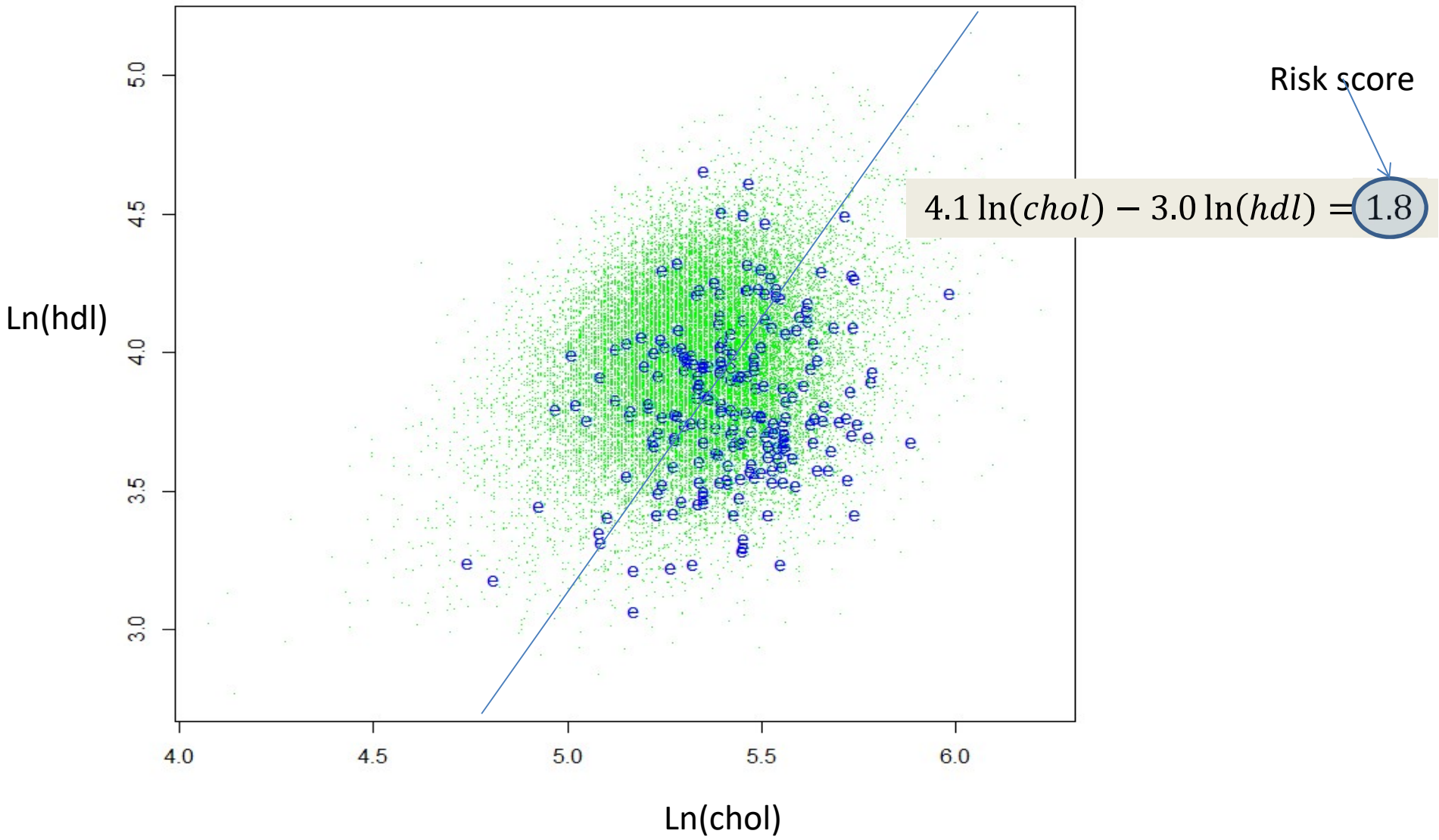
Short review of predictive models: Linear Discriminant Analysis



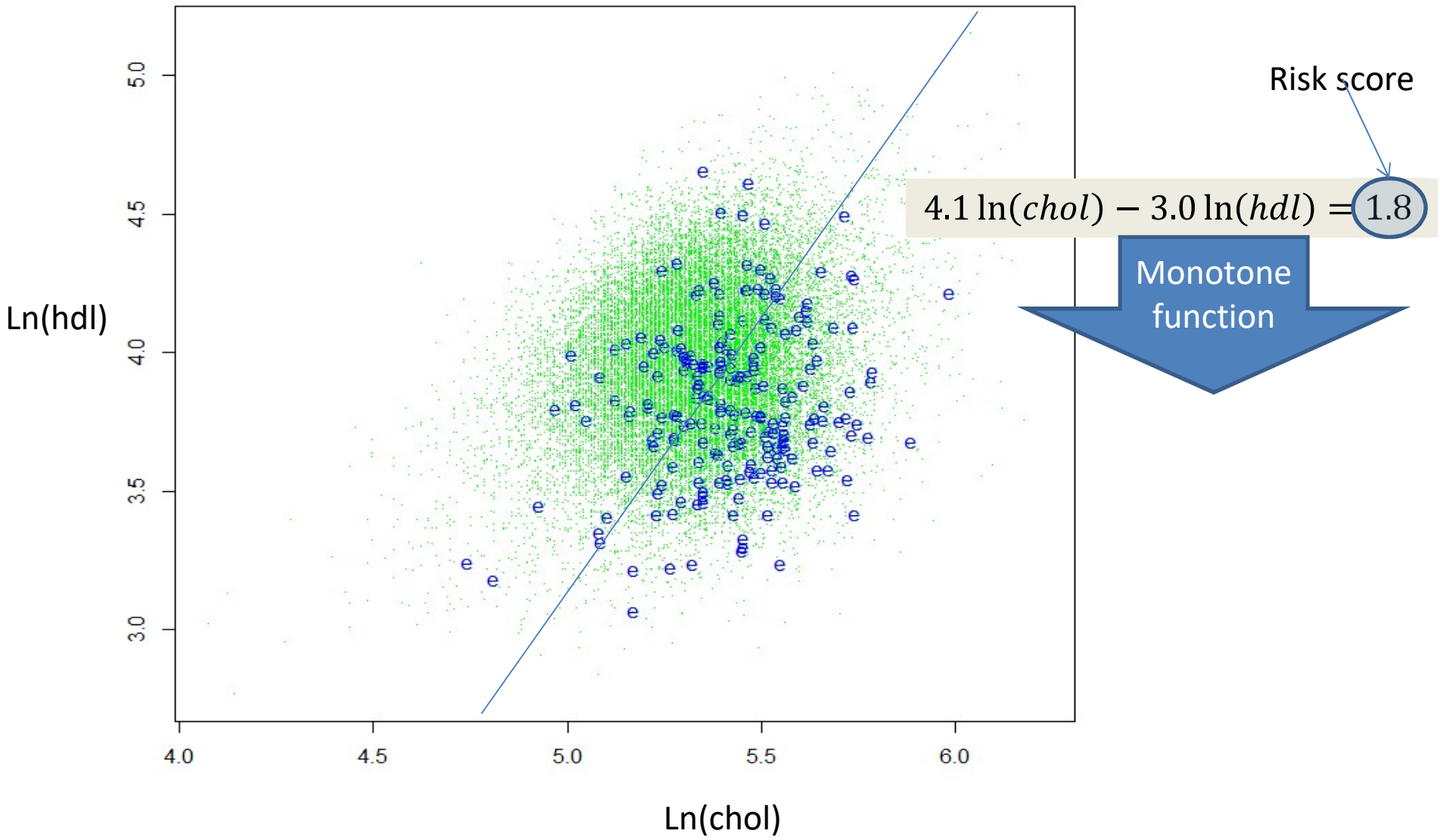
Short review of predictive models: Linear Discriminant Analysis



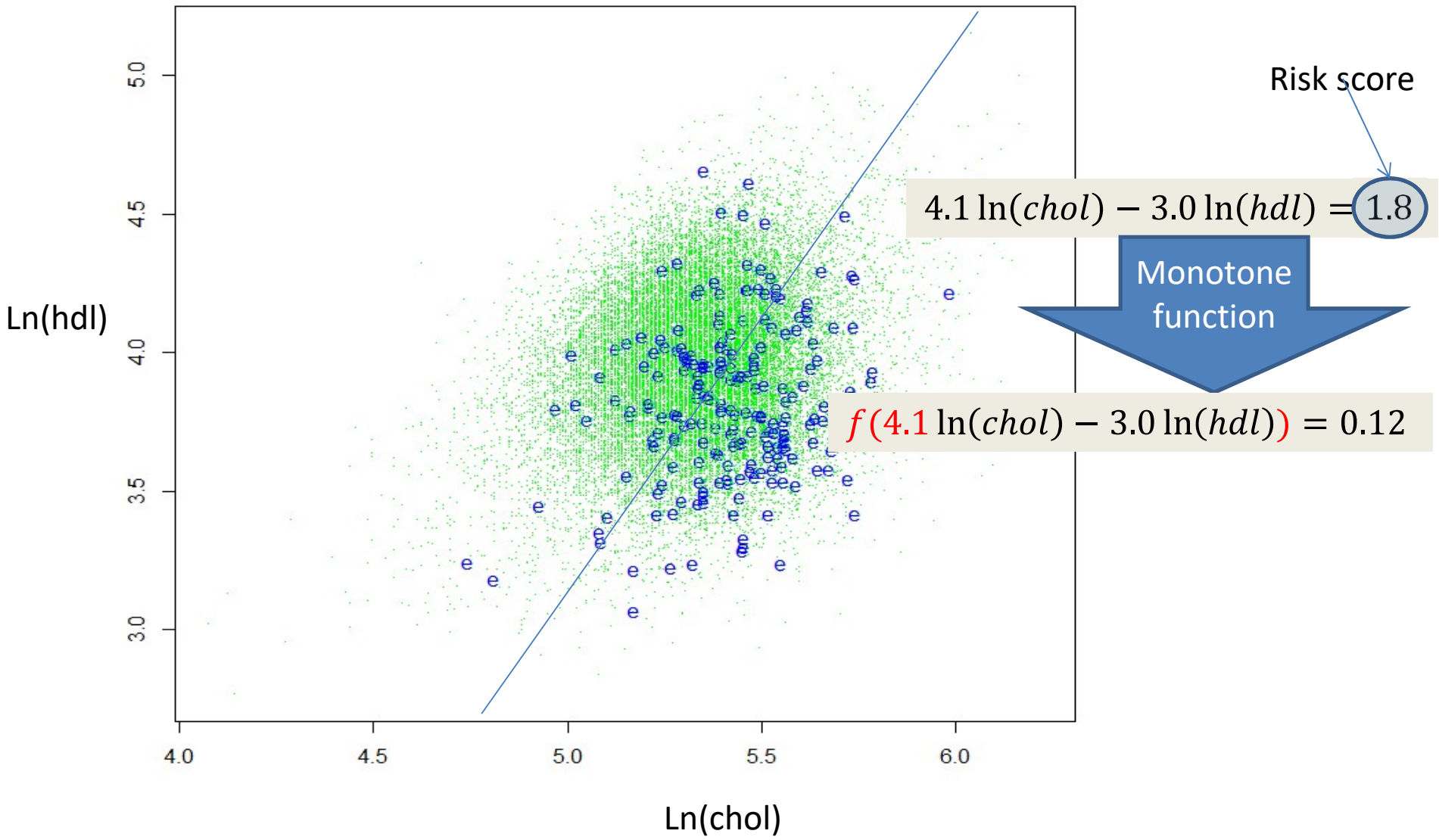
Linear Discriminant Analysis



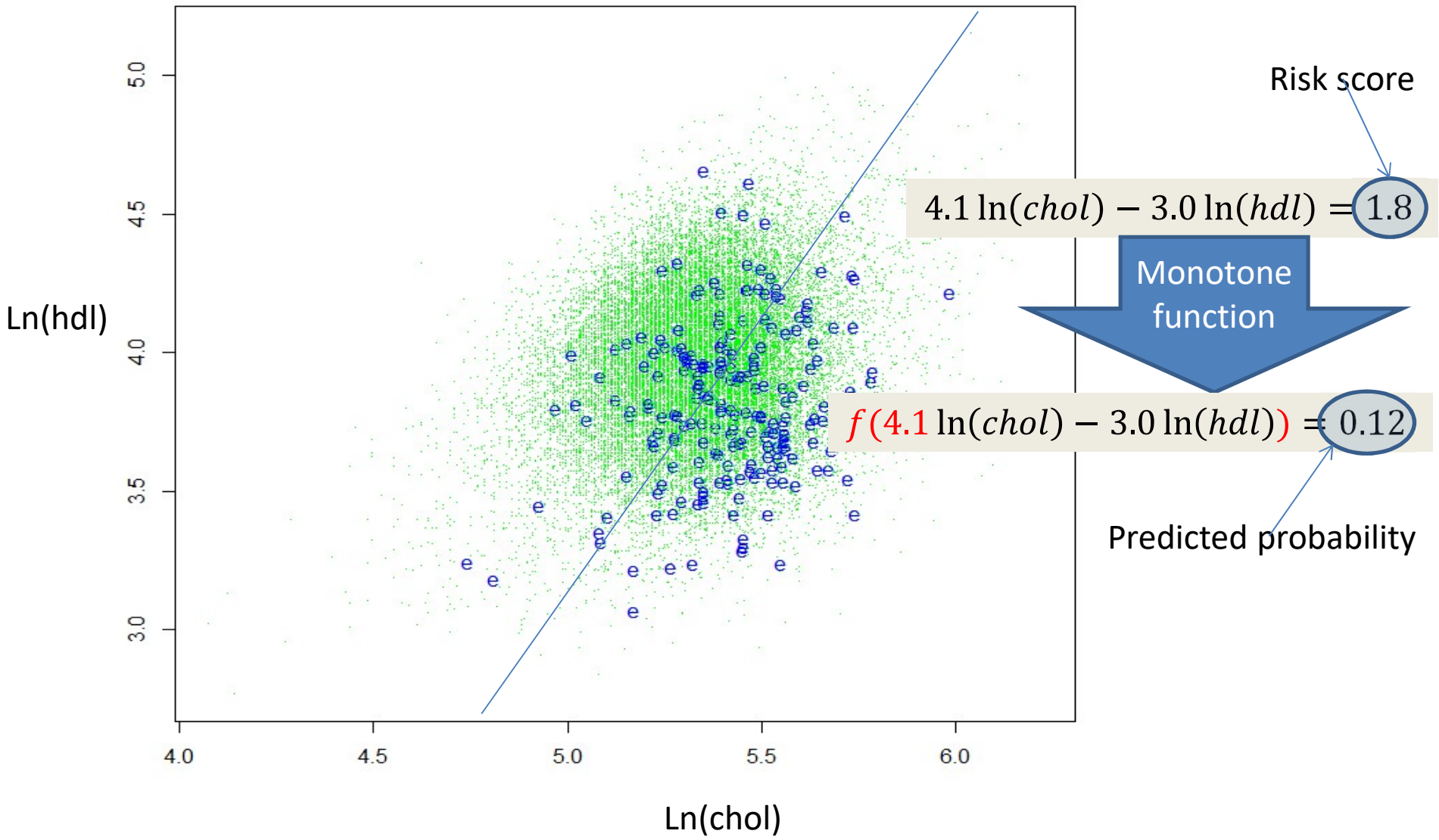
Linear Discriminant Analysis



Linear Discriminant Analysis

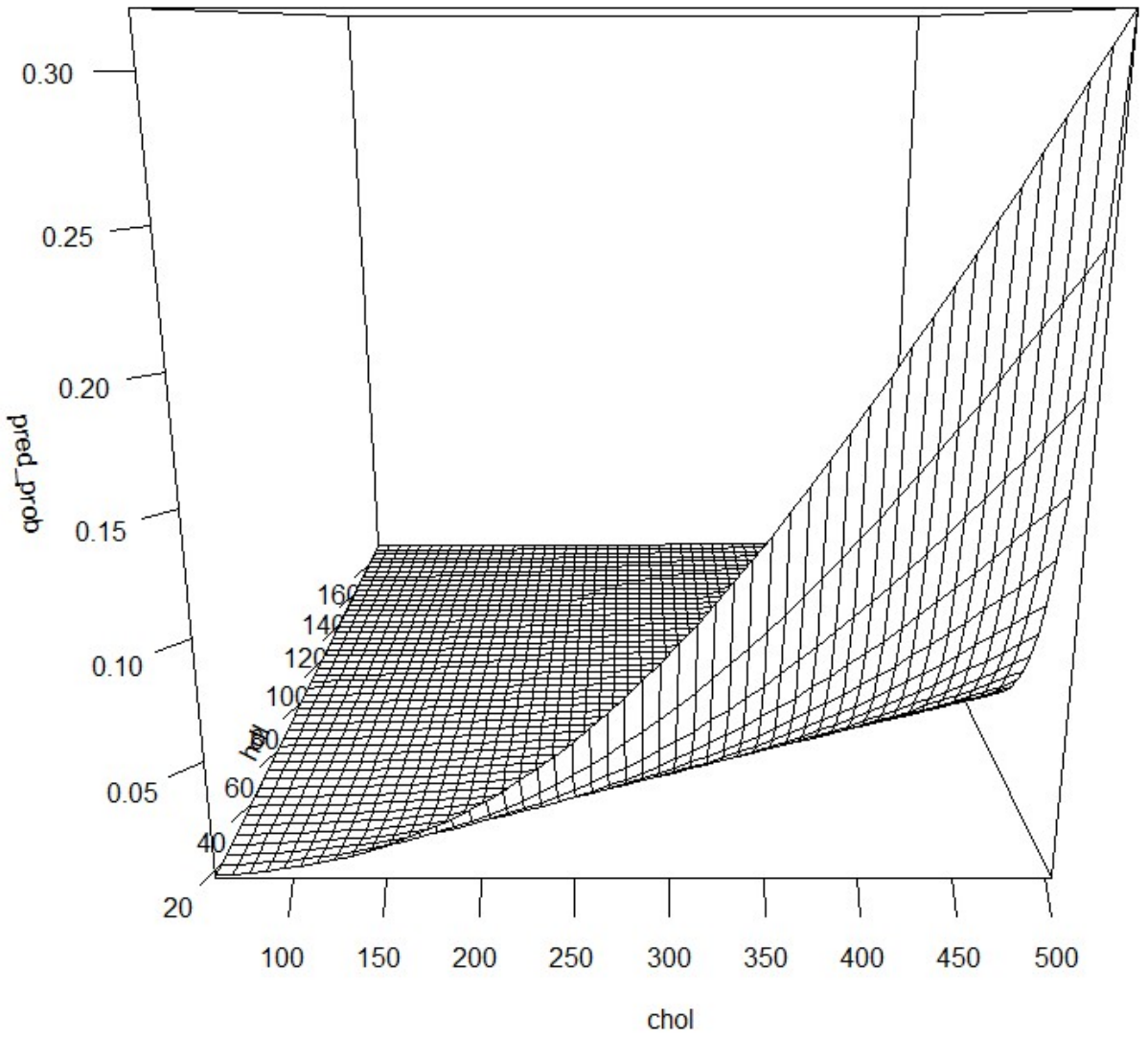


Linear Discriminant Analysis



Linear Discriminant Analysis

Assumptions: normally distributed predictors

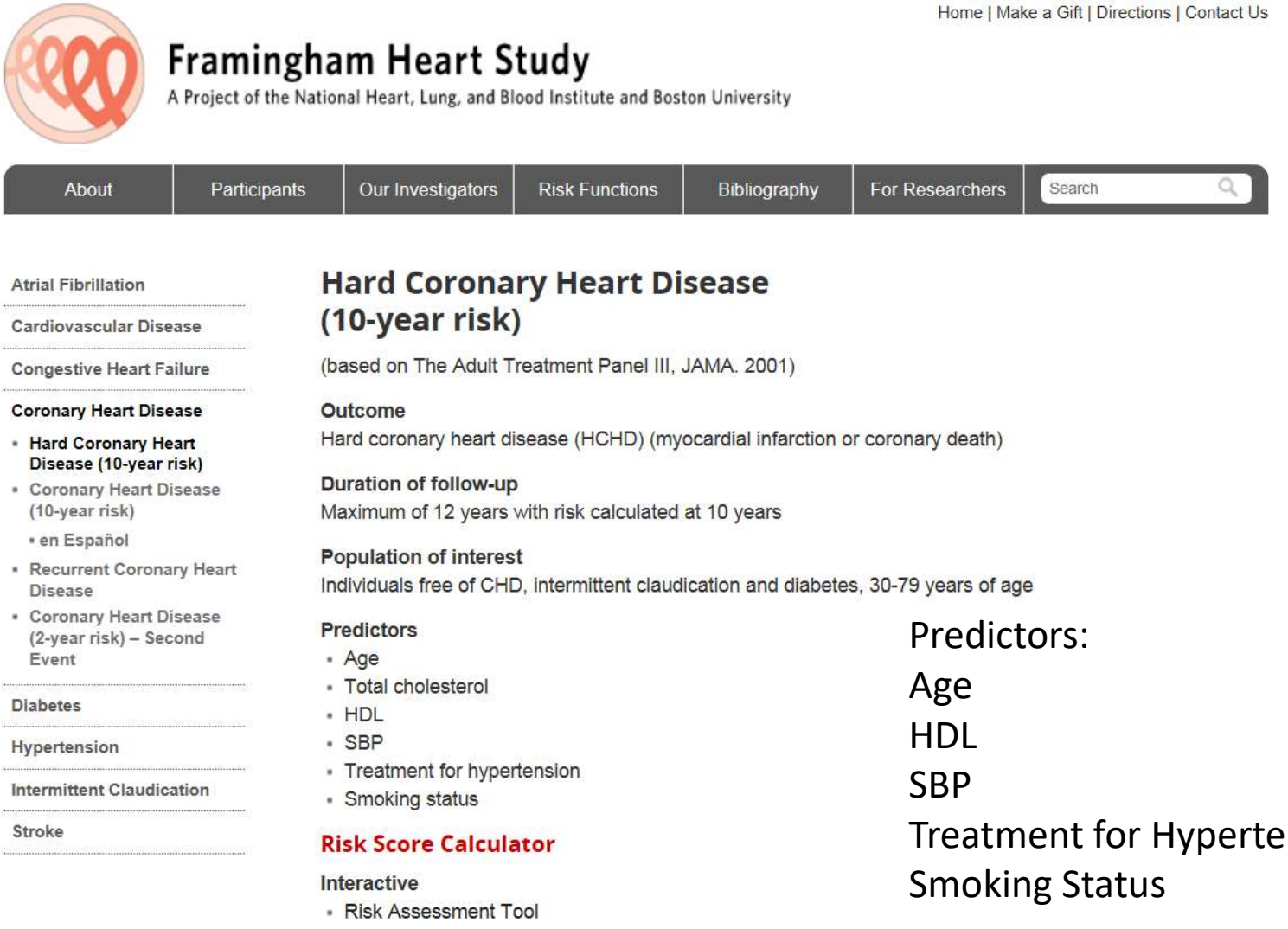


Predicted probability versus total cholesterol

hdl cholesterol

Logistic Regression

From Linear Discriminant Analysis to Logistic Regression



Home | Make a Gift | Directions | Contact Us

Framingham Heart Study
A Project of the National Heart, Lung, and Blood Institute and Boston University

About | Participants | Our Investigators | Risk Functions | Bibliography | For Researchers | Search

- Atrial Fibrillation
- Cardiovascular Disease
- Congestive Heart Failure
- Coronary Heart Disease
  - Hard Coronary Heart Disease (10-year risk)
  - Coronary Heart Disease (10-year risk)
    - en Español
  - Recurrent Coronary Heart Disease
  - Coronary Heart Disease (2-year risk) – Second Event
- Diabetes
- Hypertension
- Intermittent Claudication
- Stroke

Hard Coronary Heart Disease (10-year risk)

(based on The Adult Treatment Panel III, JAMA. 2001)

Outcome
Hard coronary heart disease (HCHD) (myocardial infarction or coronary death)

Duration of follow-up
Maximum of 12 years with risk calculated at 10 years

Population of interest
Individuals free of CHD, intermittent claudication and diabetes, 30-79 years of age

Predictors
- Age
- Total cholesterol
- HDL
- SBP
- Treatment for hypertension
- Smoking status

Risk Score Calculator

Interactive
- Risk Assessment Tool

Predictors:
Age
HDL
SBP
Treatment for Hypertension
Smoking Status

Logistic Regression

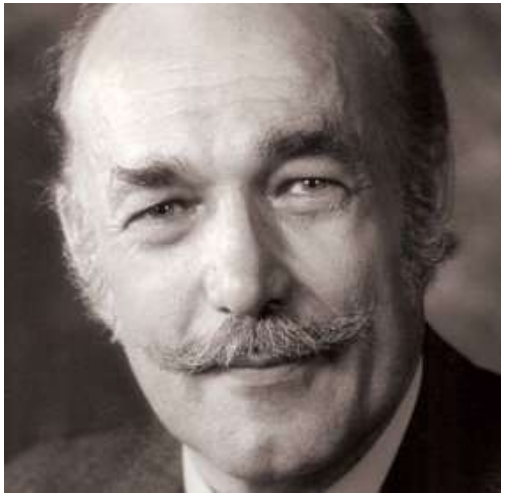
Sir R.A. Fisher

LDA (1936):
Outcome: categorical (we consider binary outcomes in this workshop)

Predictors:
Confidence Intervals and p-values are valid **only** for normally distributed predictors

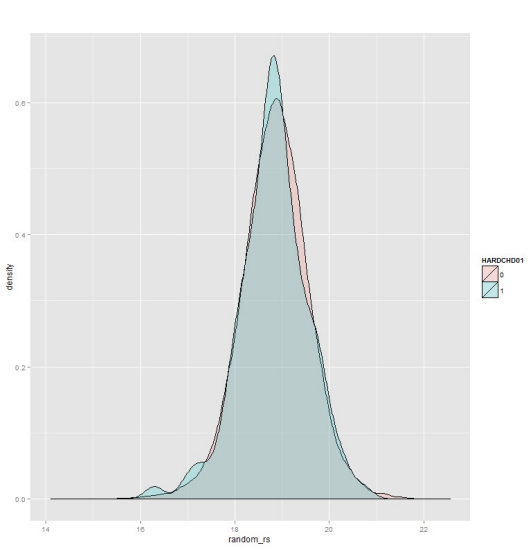
Logistic regression (1940-1970ies)
Outcome: categorical (we consider binary outcomes in this workshop)

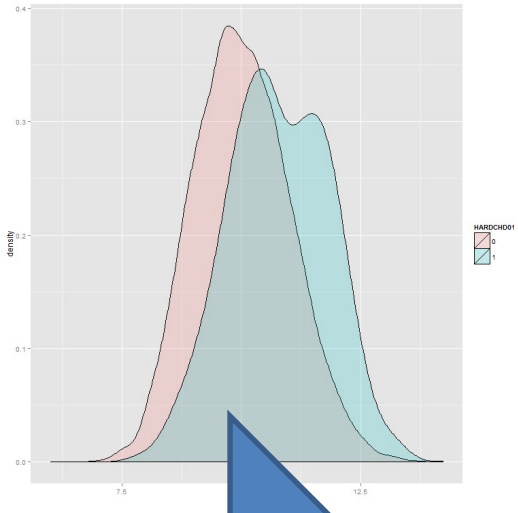
Predictors:
Will work with any kind of predictors
Confidence Intervals and p-values are valid for a wider class of distributions including discrete families

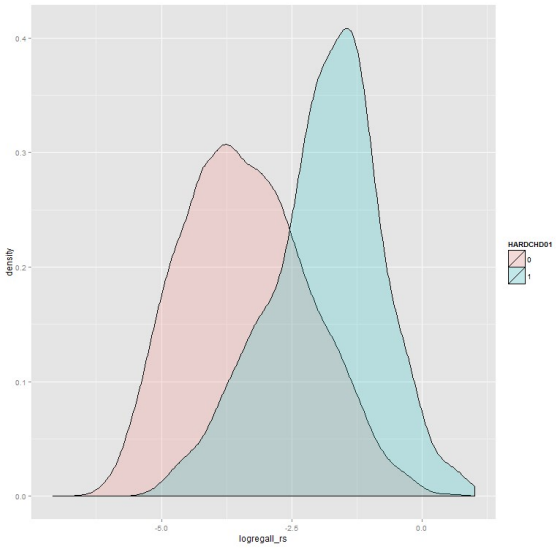


J. Nelder, Wedderburn et. al.

Logistic Regression:
works for continuous and categorical predictor variables







Ln(chol), ln(hdl)

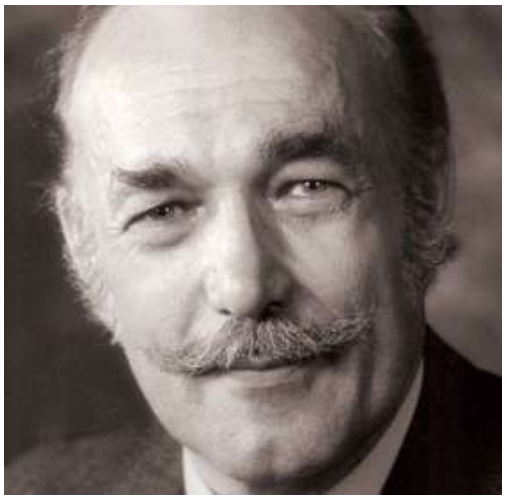
$$\alpha_0 + \alpha_1 \ln(chol) + \alpha_2 \ln(hdl) + \alpha_3 smoking\ status + \alpha_4 treatment\ status$$

Predicted probability

Logistic Regression

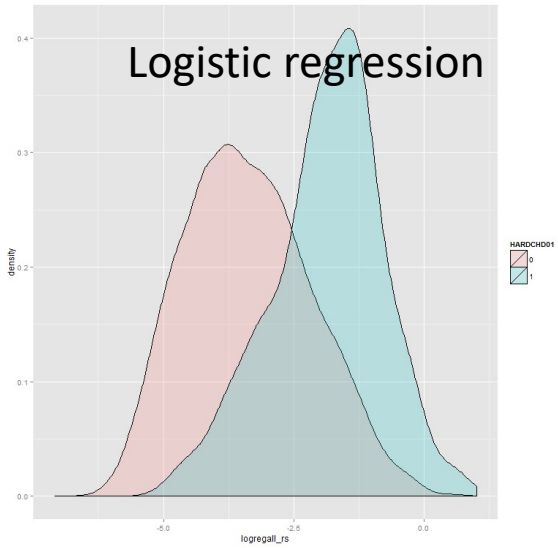
Logistic regression (1940-1970ies)
Outcome: categorical (we consider binary outcomes in this workshop)

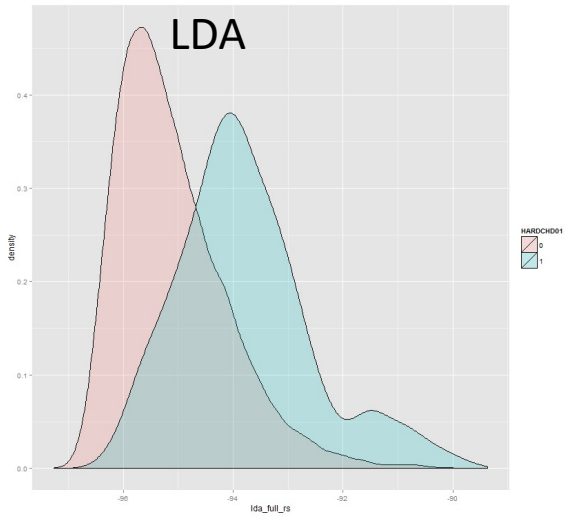
Predictors:
Will work with any kind of predictors
Confidence Intervals and p-values are valid for a wider class of distributions including discrete families



J. Nelder, Wedderburn et. al.

Linear Discriminant Analysis versus Logistic Regression





Pros:

Works with a wider range of predictors

Cons:

Less efficient than LDA (wider confidence intervals)

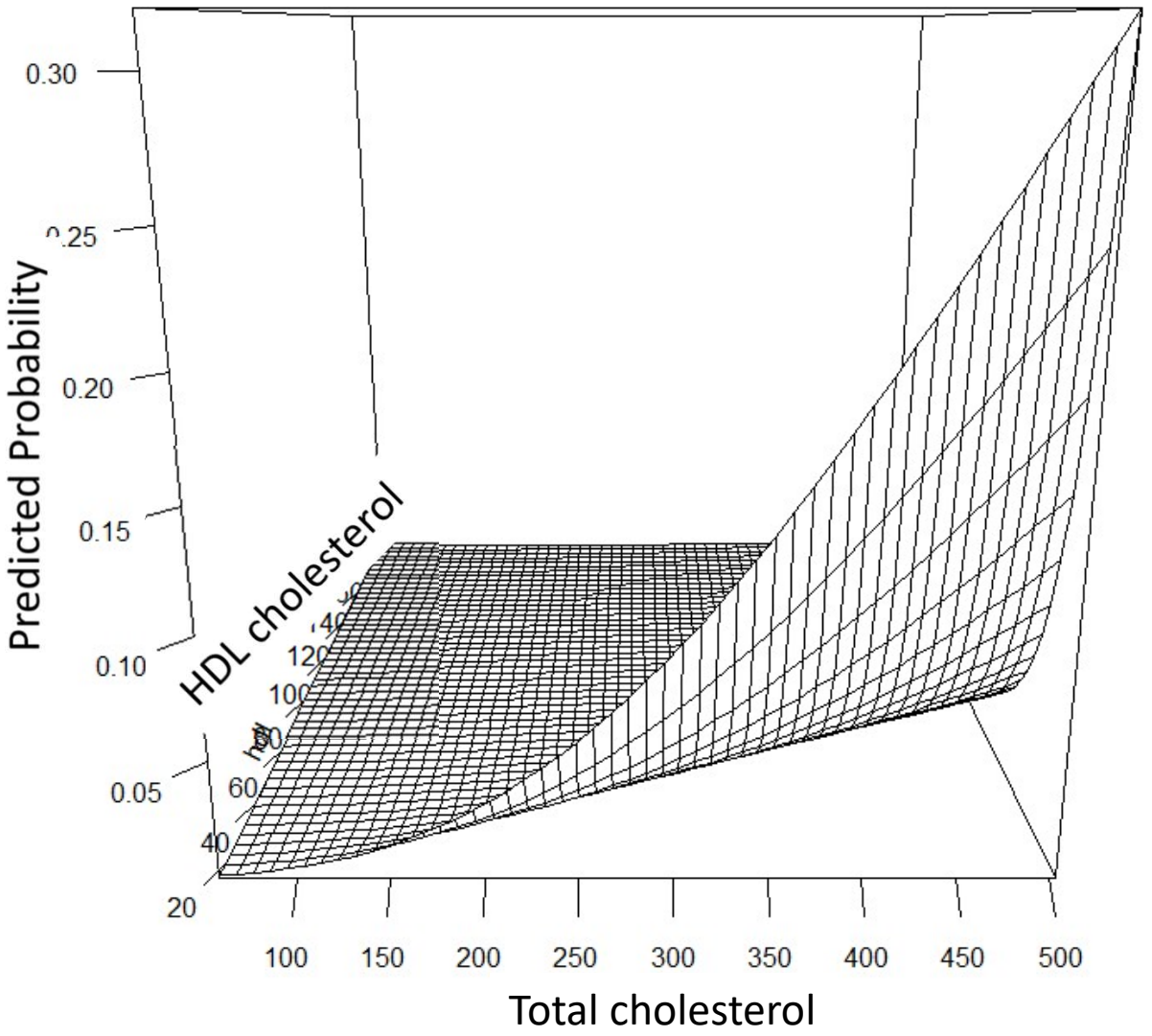
Logistic Regression. Model Selection.

Akaike Information Criteria; Bayesian Information Criteria

- Log likelihood of model – penalty for number of predictors
- BIC has bigger penalty

From Logistic Regression to Supervised Learning Methods

Predicted probability versus total cholesterol and hdl cholesterol



Survival Analysis

Survival Analysis (brief overview)

Now we have the time of the event and also some observations are censored

- Non-parametric Models:
- Kaplan-Meier
- Nelson-Aalen

- Semi-parametric models:
- Cox proportional hazards model

Survival Analysis

Women's Health Study (WHS):

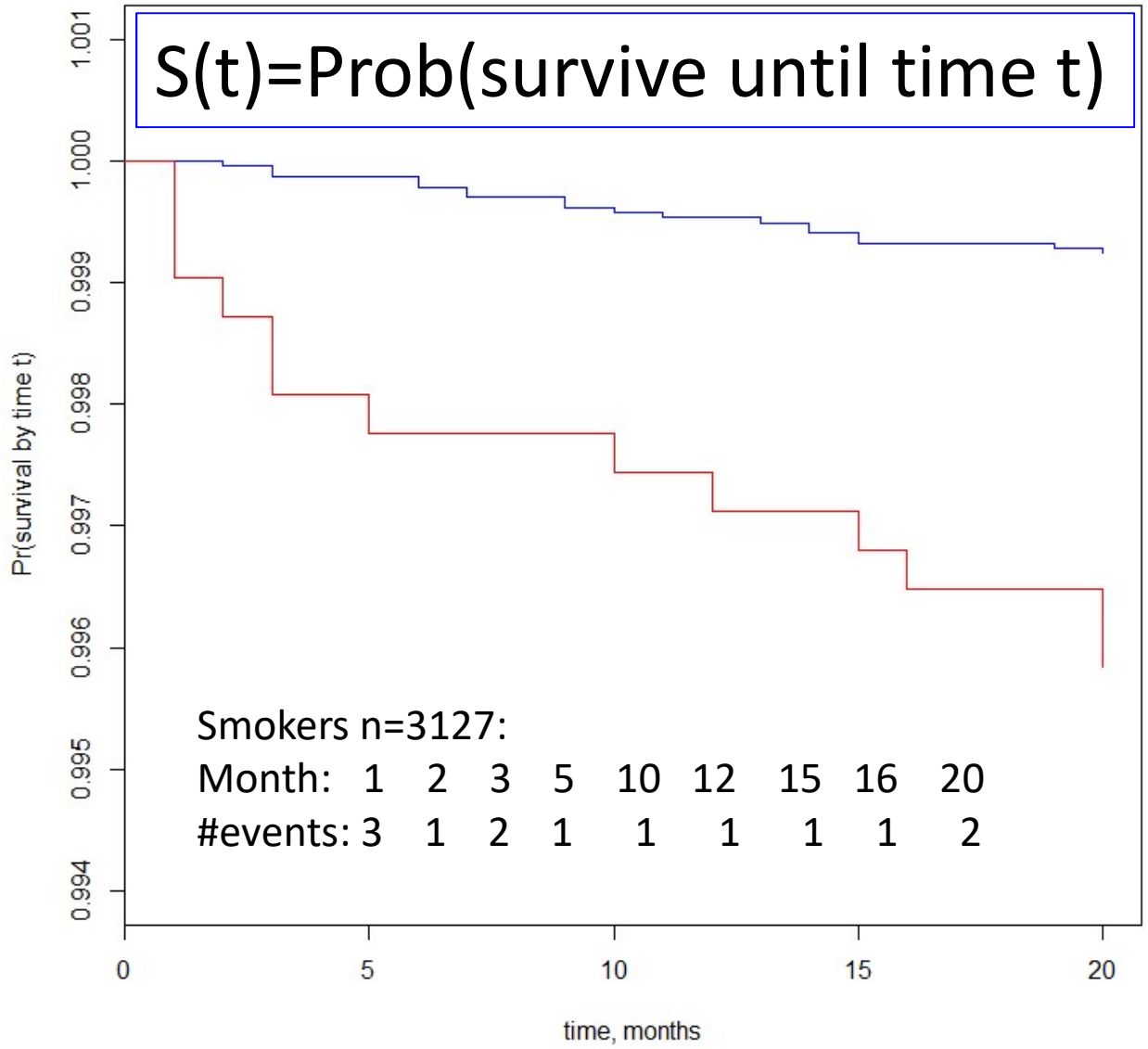
- a cohort study
- 8-year cardiovascular events
- Some subjects dropped out of the study before 8 years (censoring).

Outcome of censored observations is unknown

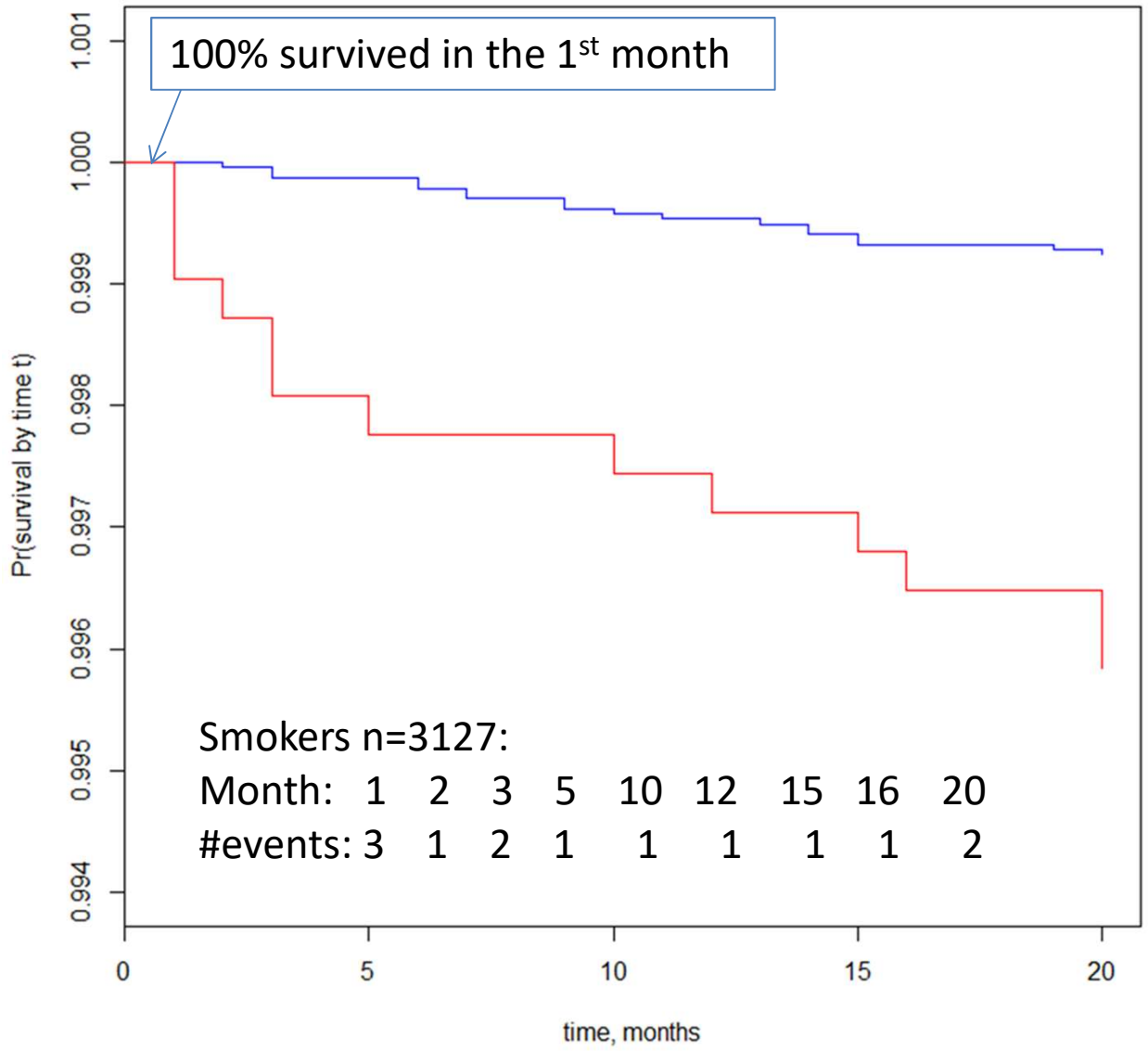
Survival Analysis

S(t)=Prob(survive until time t)

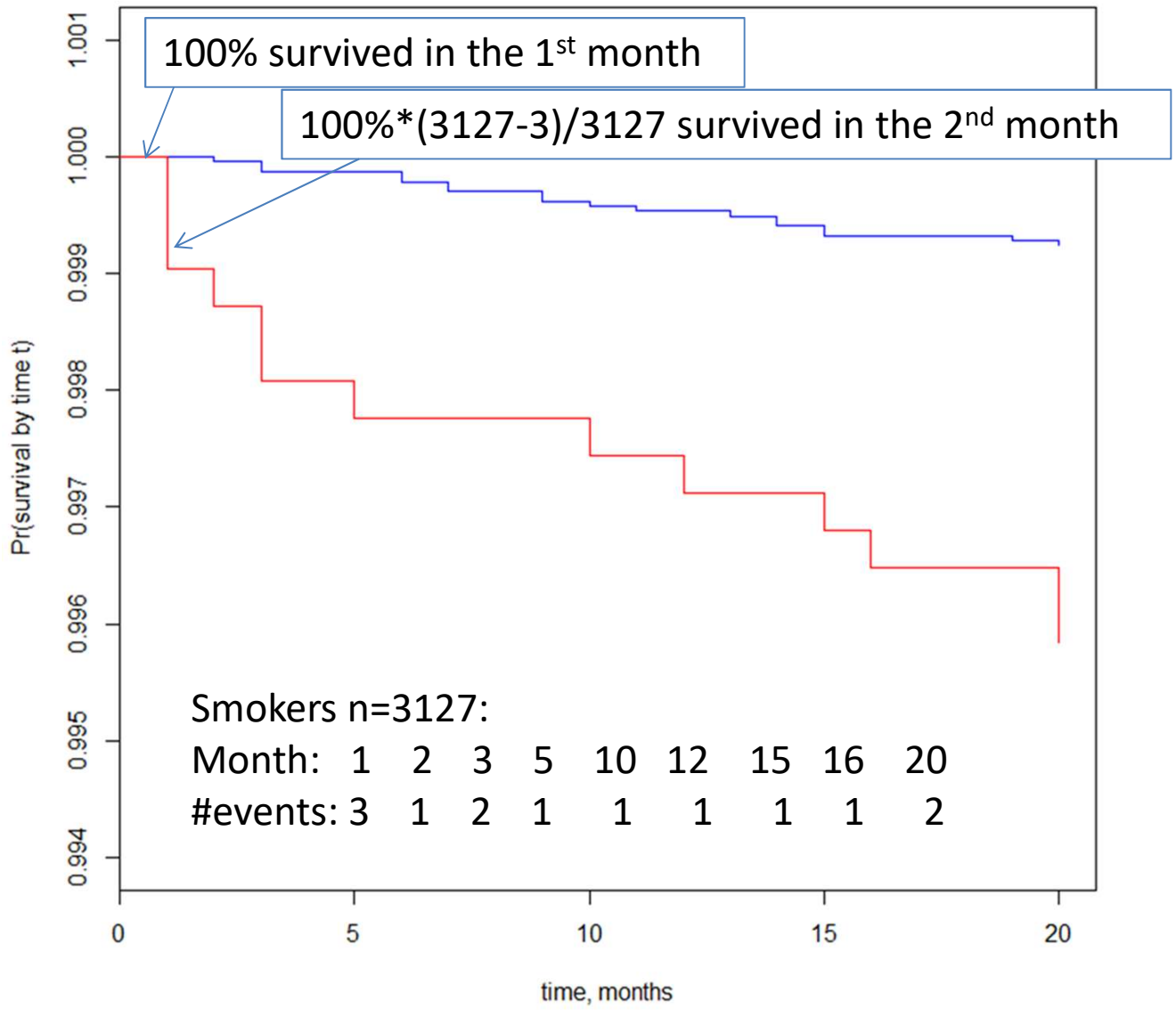
Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator



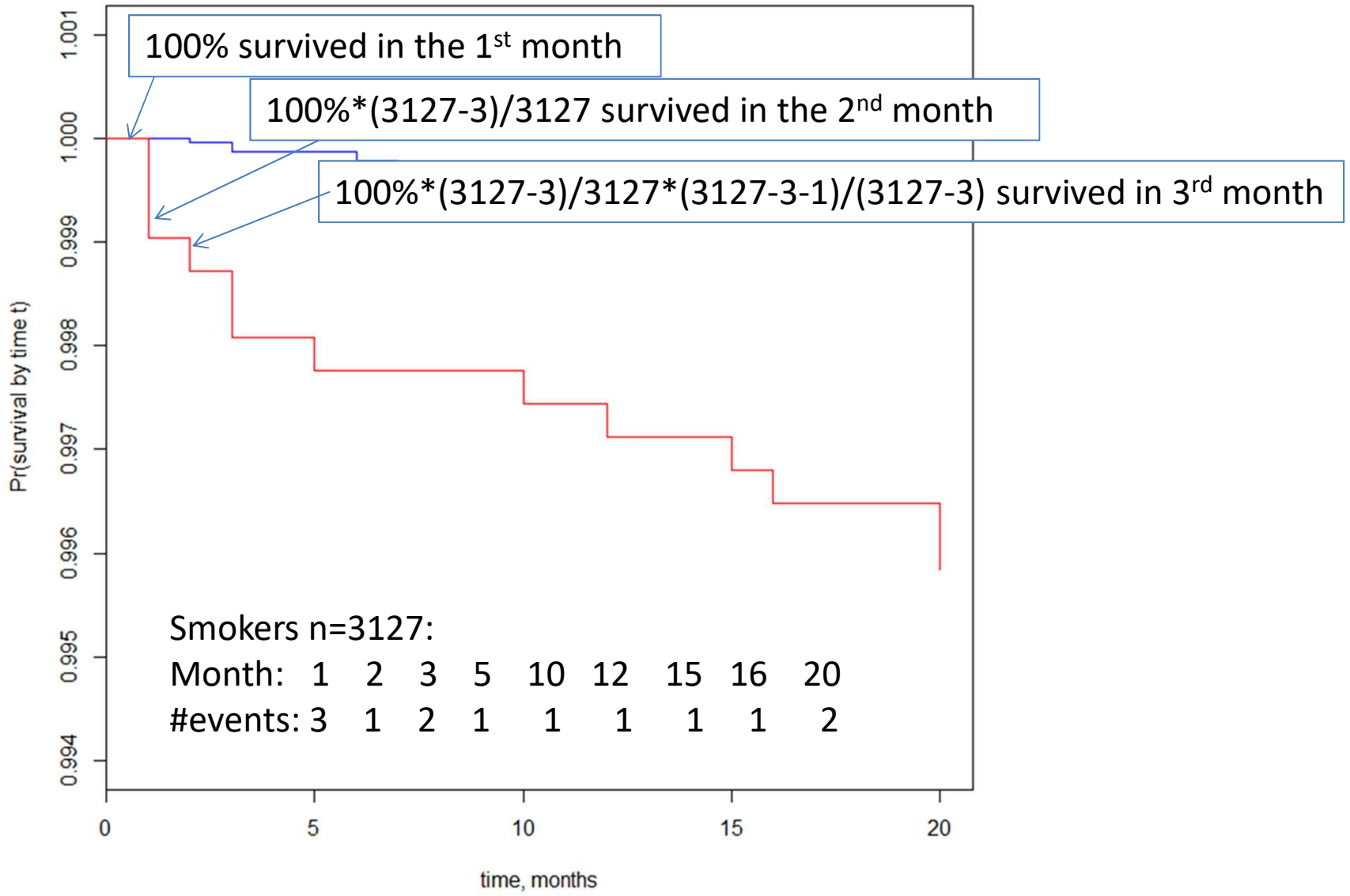
Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator



Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator



Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator

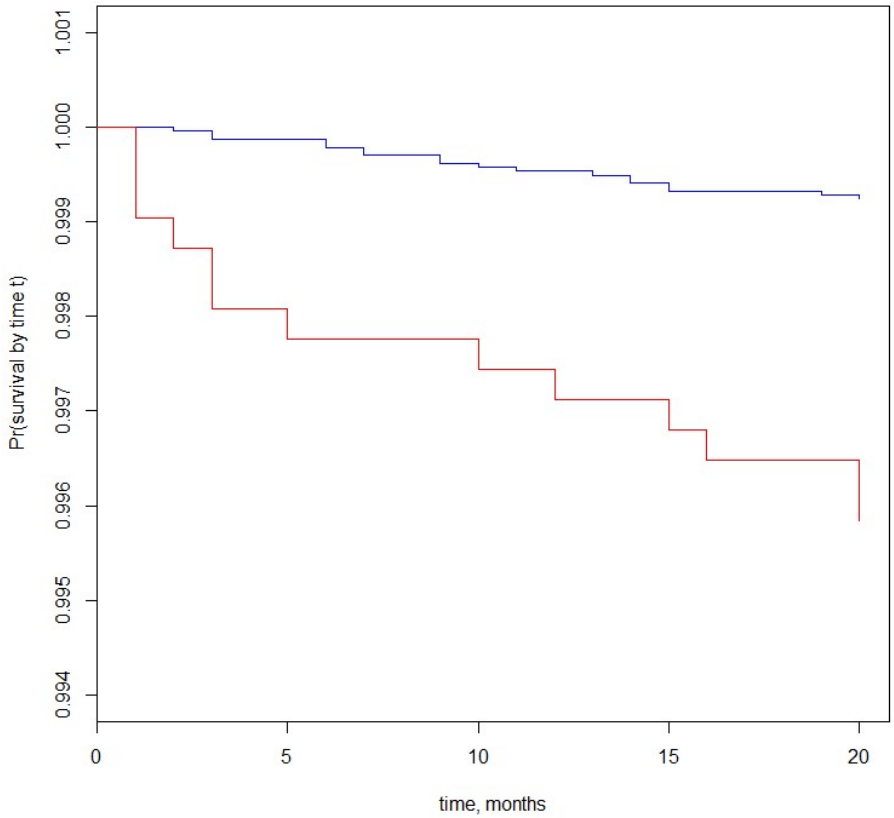


Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator

Non-parametric model:

Kaplan-Meier estimator (1958) or product-limit estimator

P. Meier



Survival Analysis.
Kaplan-Meier estimator (1958) or product-limit estimator

Pros:

- Intuitive
- Can (crudely) test survival rates in several groups (logrank test)

Cons:

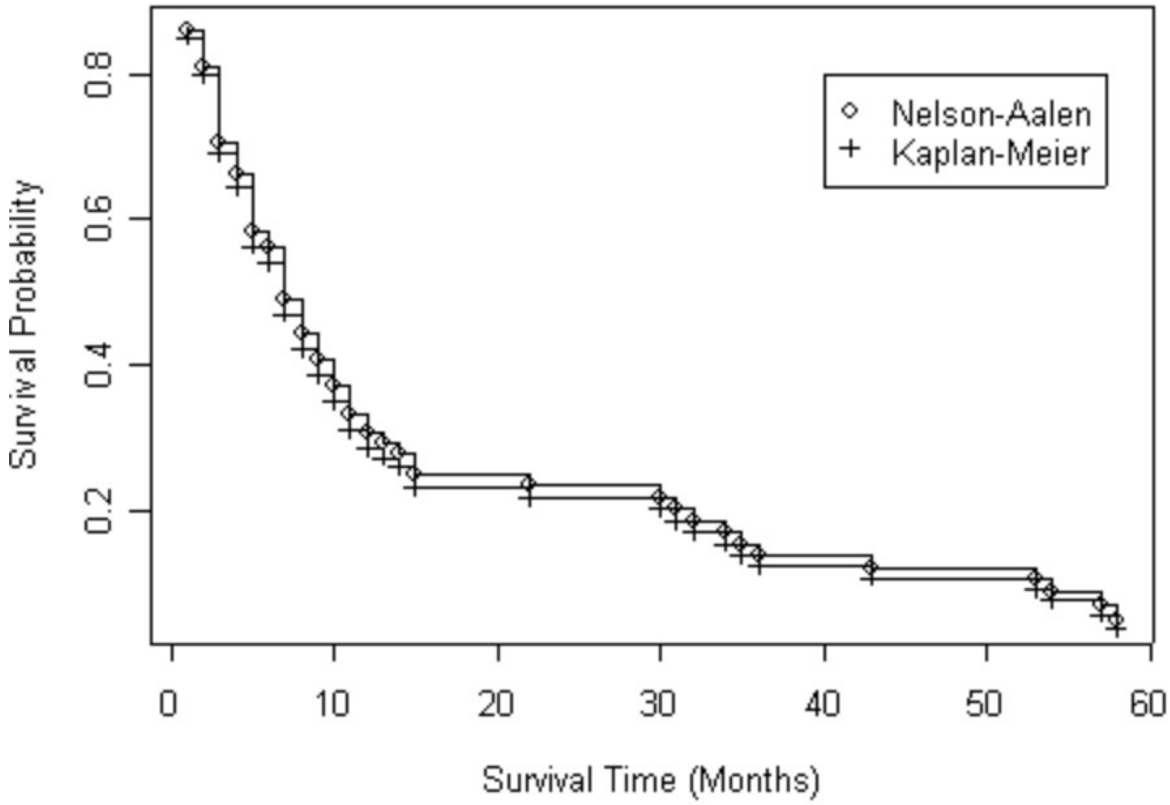
- Cannot accommodate continuous predictors
- Can calculate only group-level predicted probabilities

Survival Analysis.
Nelson-Aalen estimator

- Another non-parametric survival model:

Nelson-Aalen

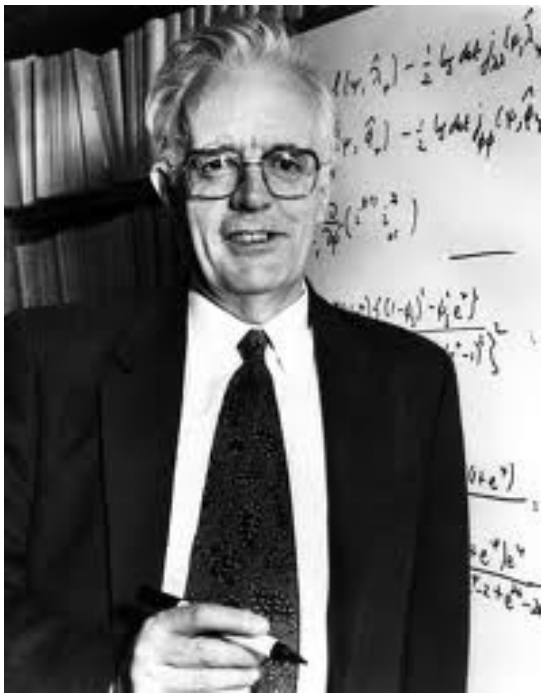
O. Aalen



Very similar to Kaplan-Meier

Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

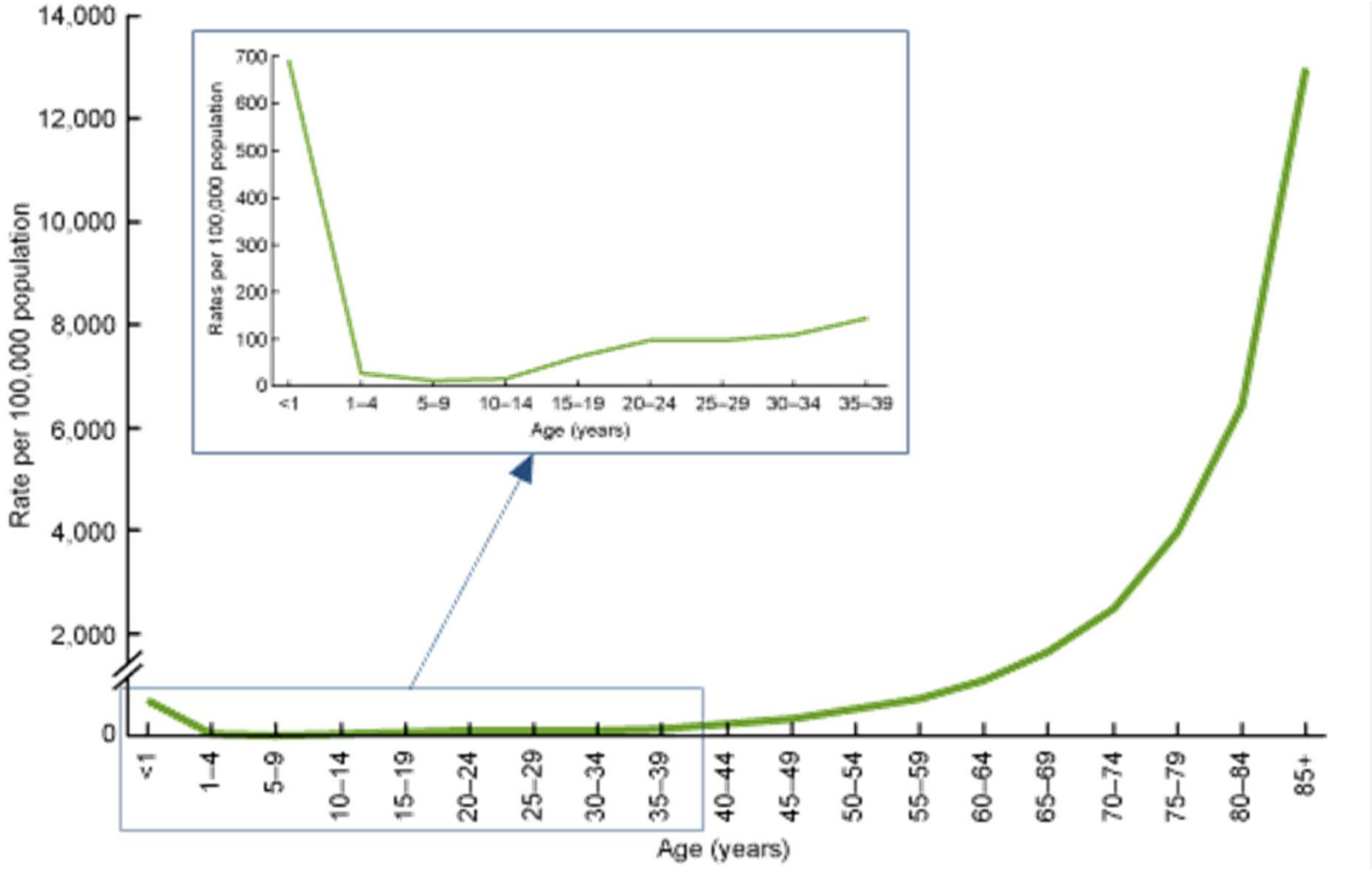


Sir David Cox

Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time. All have common Baseline Hazard which may change over time

http://www.cdc.gov/nchs/data/databriefs/db26.pdf



Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

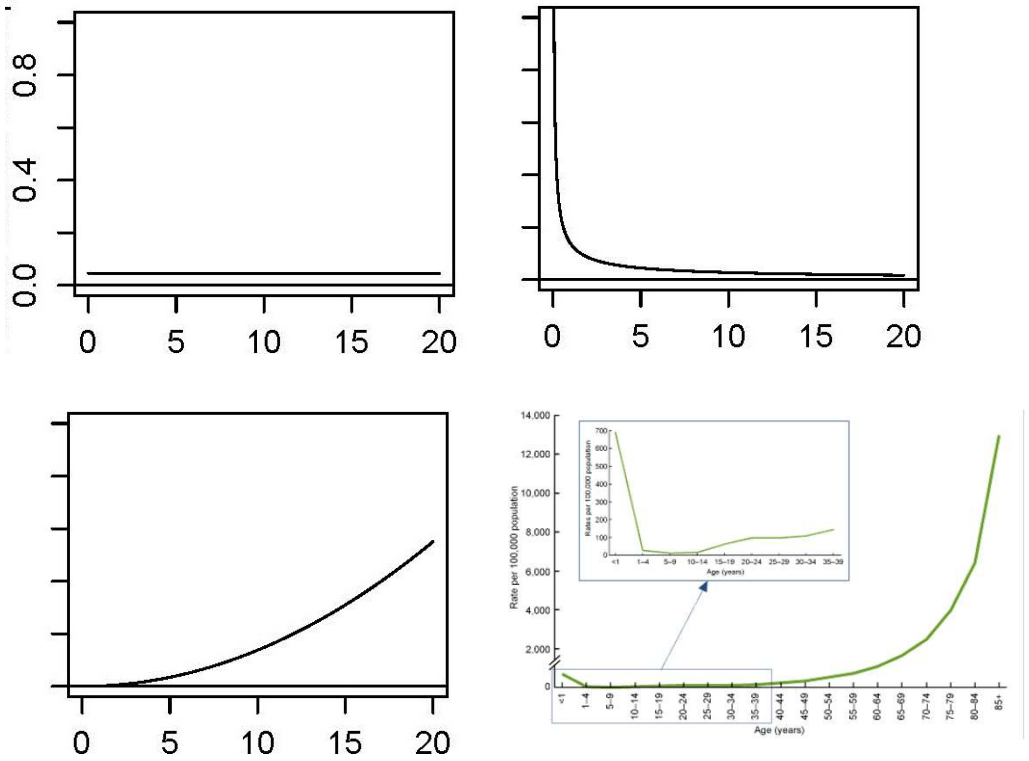
Hazard(t)=BazelineHazard(t) x positive number

Survival Analysis.
Cox Proportional Hazards Regression Model

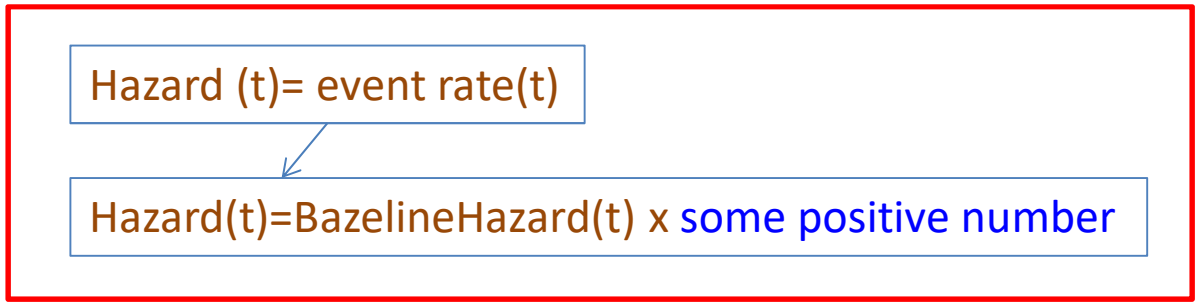
Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

Hazard(t)=BazelineHazard(t) x positive number

Time (horizontal) versus hazard rate

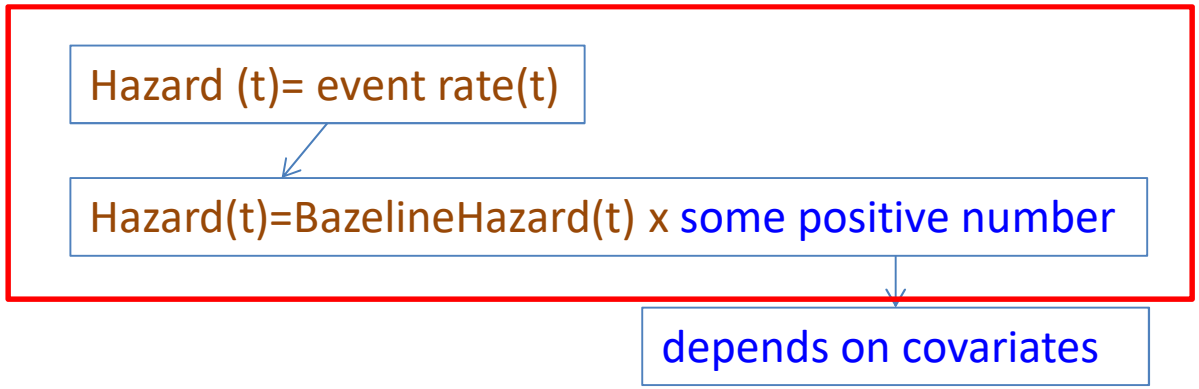


Survival Analysis.
Cox Proportional Hazards Regression Model

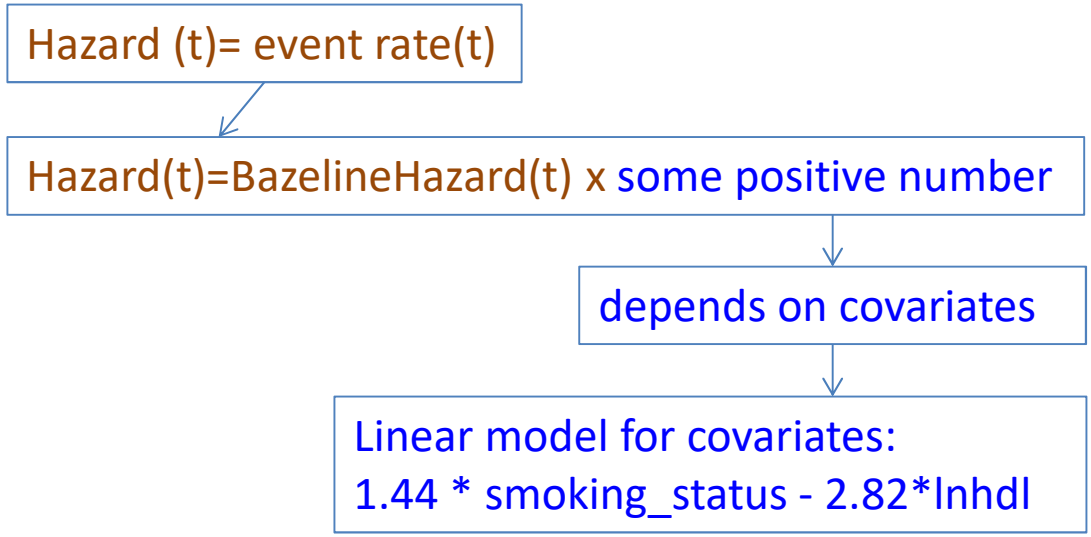


Proportional Hazards assumption

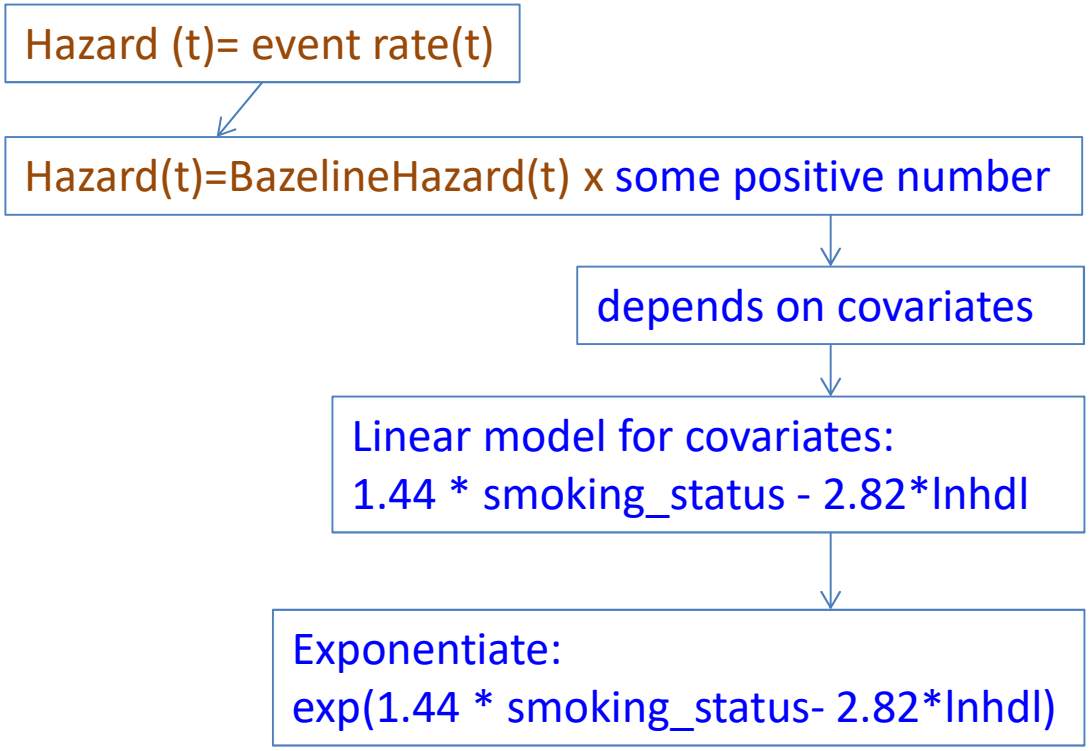
Survival Analysis.
Cox Proportional Hazards Regression Model



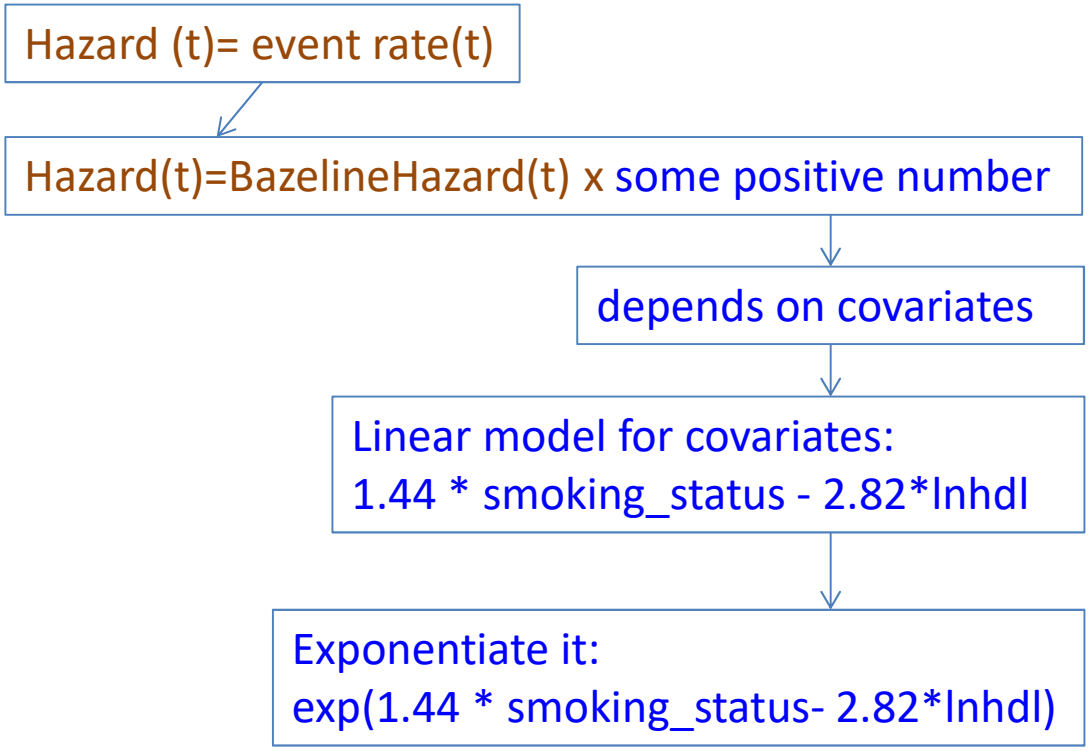
Survival Analysis.
Cox Proportional Hazards Regression Model



Survival Analysis.
Cox Proportional Hazards Regression Model

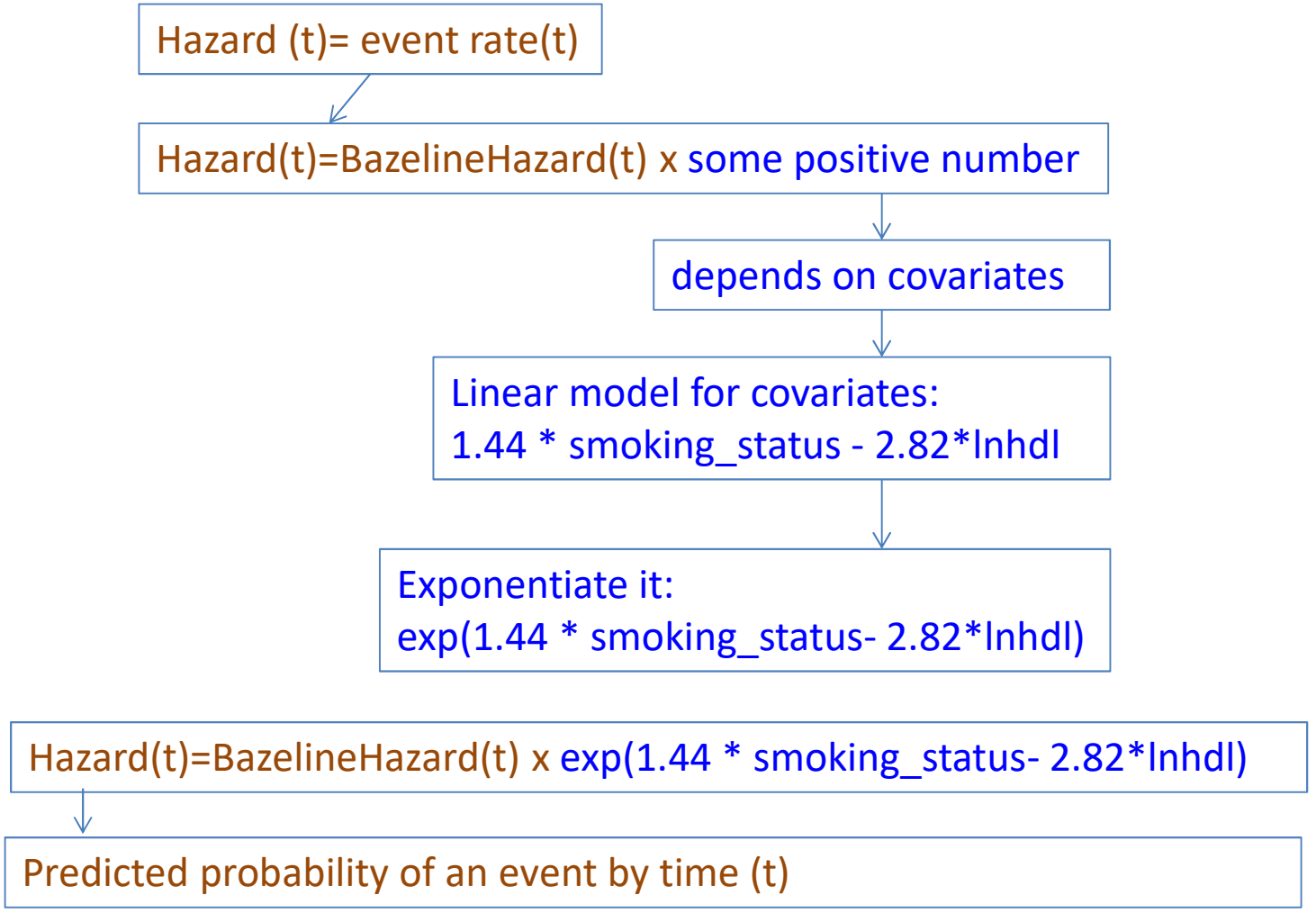


Survival Analysis.
Cox Proportional Hazards Regression Model



Hazard(t)=BazelineHazard(t) x exp(1.44 * smoking_status- 2.82*lnhdl)

Survival Analysis.
Cox Proportional Hazards Regression Model



Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

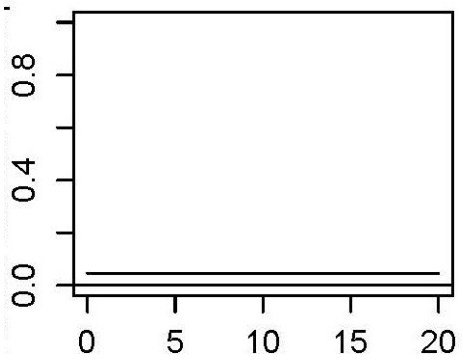
Time (horizontal) versus hazard rate

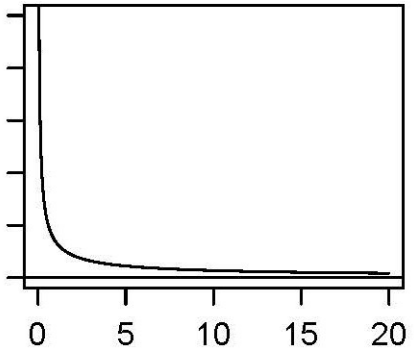
Hazard(t)=BazelineHazard(t) x positive number

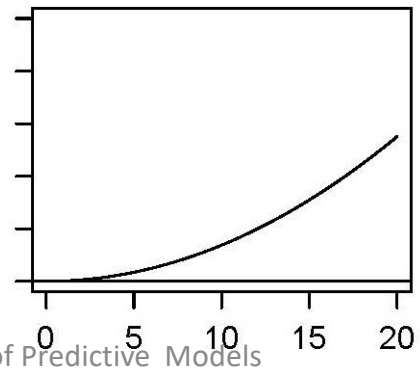
Some positive number depends on Covariates

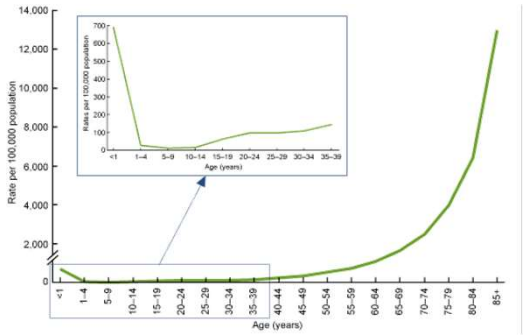
Covariates can be

- Categorical(smoking y/n; medication_type)
- Ordinal(education category)
- Continuous(age, lipid levels etc)









Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

Time (horizontal) versus hazard rate

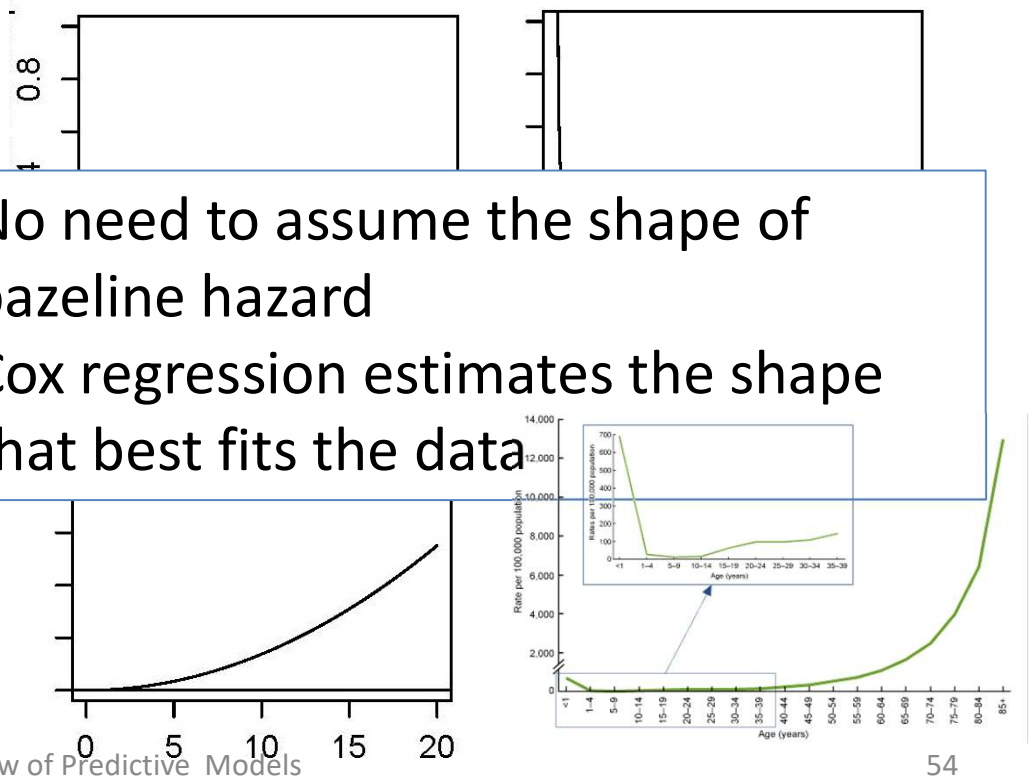
Hazard(t)=BazelineHazard(t) x positive number

Some positive number depends on Covariates

Covariates can be

- Categorical(smoking y/n; medication_type)
- Ordinal(education category)
- Continuous(age, lipid levels etc)

No need to assume the shape of bazeline hazard
Cox regression estimates the shape that best fits the data



Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard Function = event rate per unit of time
All have some common Baseline Hazard which may change over time

Time (horizontal) versus hazard rate

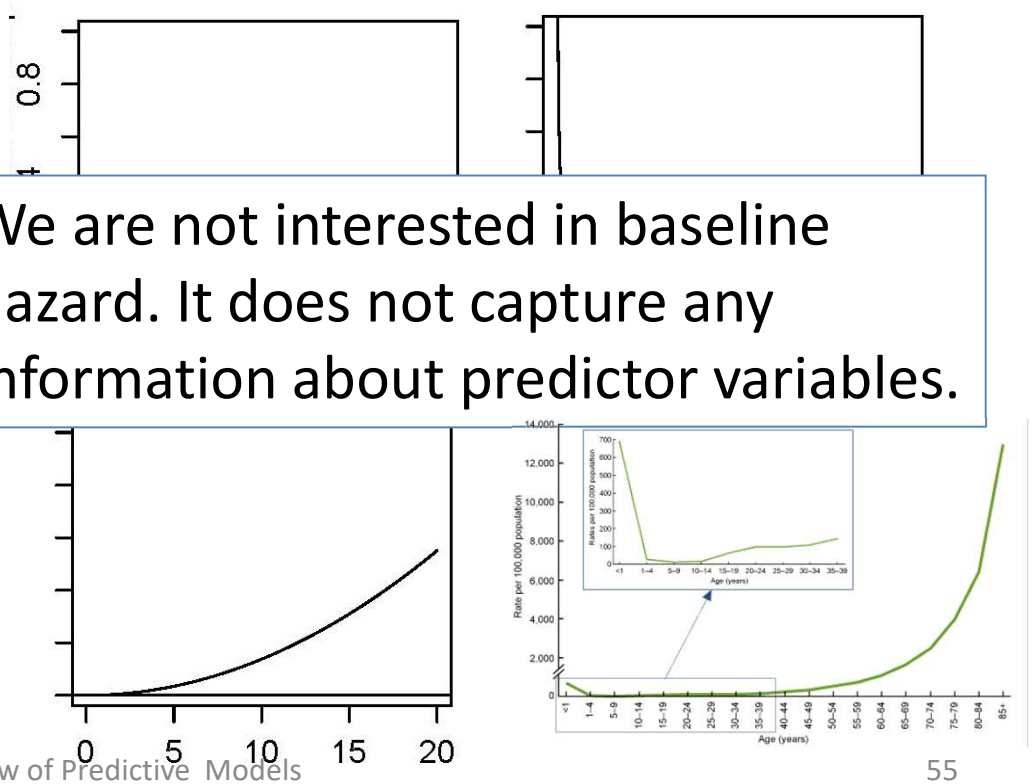
Hazard(t)=BazelineHazard(t) x positive number

Some positive number depends on Covariates

Covariates can be

- Categorical(smoking y/n; medication_type)
- Ordinal(education category)
- Continuous(age, lipid levels etc)

We are not interested in baseline hazard. It does not capture any information about predictor variables.



Survival Analysis.
Cox Proportional Hazards Regression Model

```
             coef  exp(coef)
tbl$CURRSMK 1.44   4.2333
tbl$LNHDL   -2.82  0.0597
```

Hazard(t)=BazelineHazard(t) x positive number

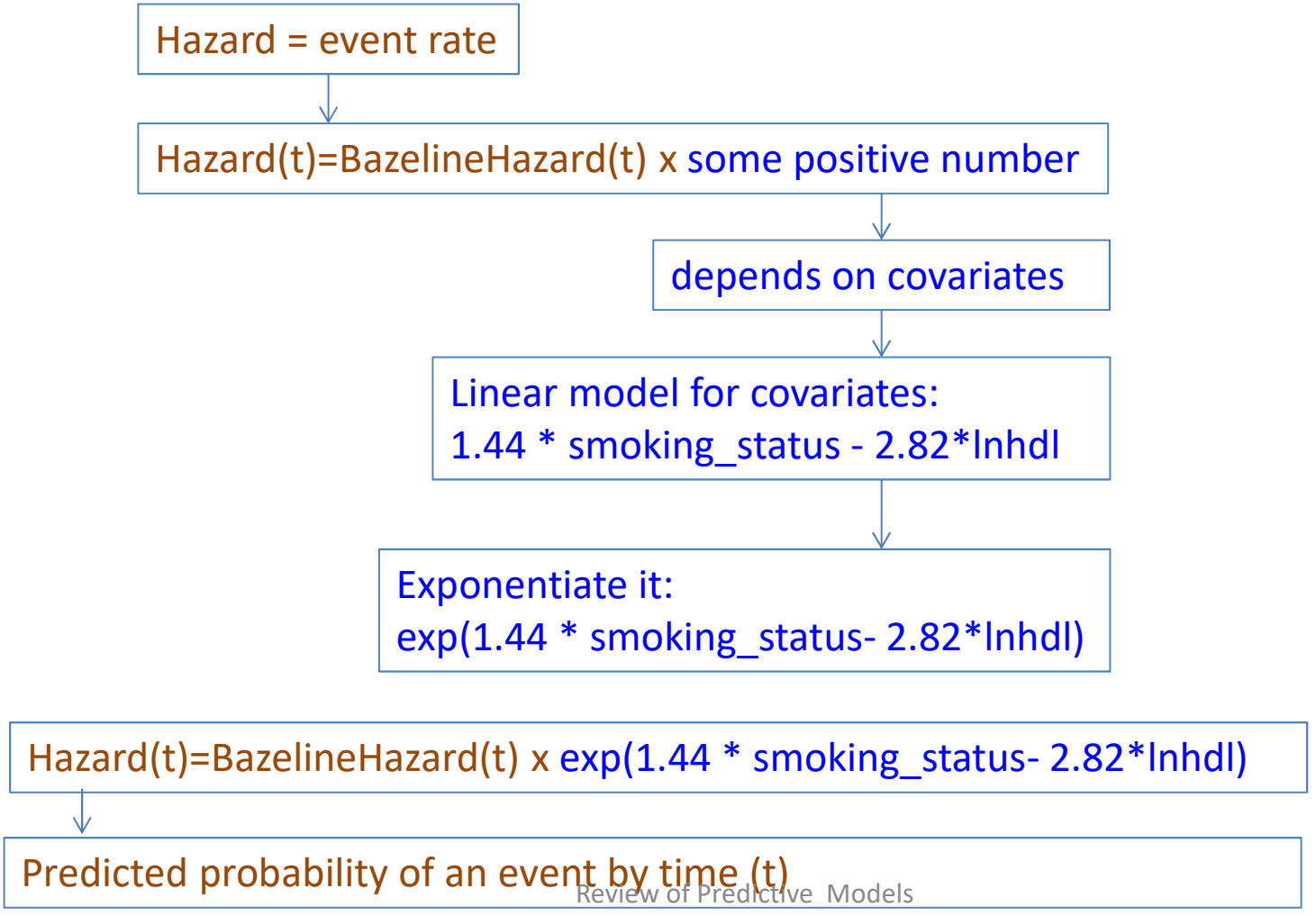
Some positive number depends on Covariates

Covariates can be

- Categorical (smoking y/n; medication_type)
- Ordinal (education category)
- Continuous (age, lipid levels etc)

a smoker with lnhdl=4.4:
1.44 - 2.82*4.4
It is negative!
Exponentiate:
exp(1.44 - 2.82*4.4)
Hazard(t)=BazelineHazard(t) x exp(1.44 - 2.82*4.4)

Survival Analysis.
Cox Proportional Hazards Regression Model



Survival Analysis.
Cox Proportional Hazards Regression Model

Hazard = event rate

↓

Hazard(t)=BazelineHazard(t) x some positive number

probability at month 20	CURRSMK	CHOL	HDL	AGE	TRTSBP	SBP*MEDS
0.075	1	204	31	60	1	185
0.108	1	270	30	63	0	185
0.080	1	233	36	75	1	145
0.077	1	255	34	69	1	145
0.084	1	287	32	59	1	165
0.080	1	211	32	66	1	165

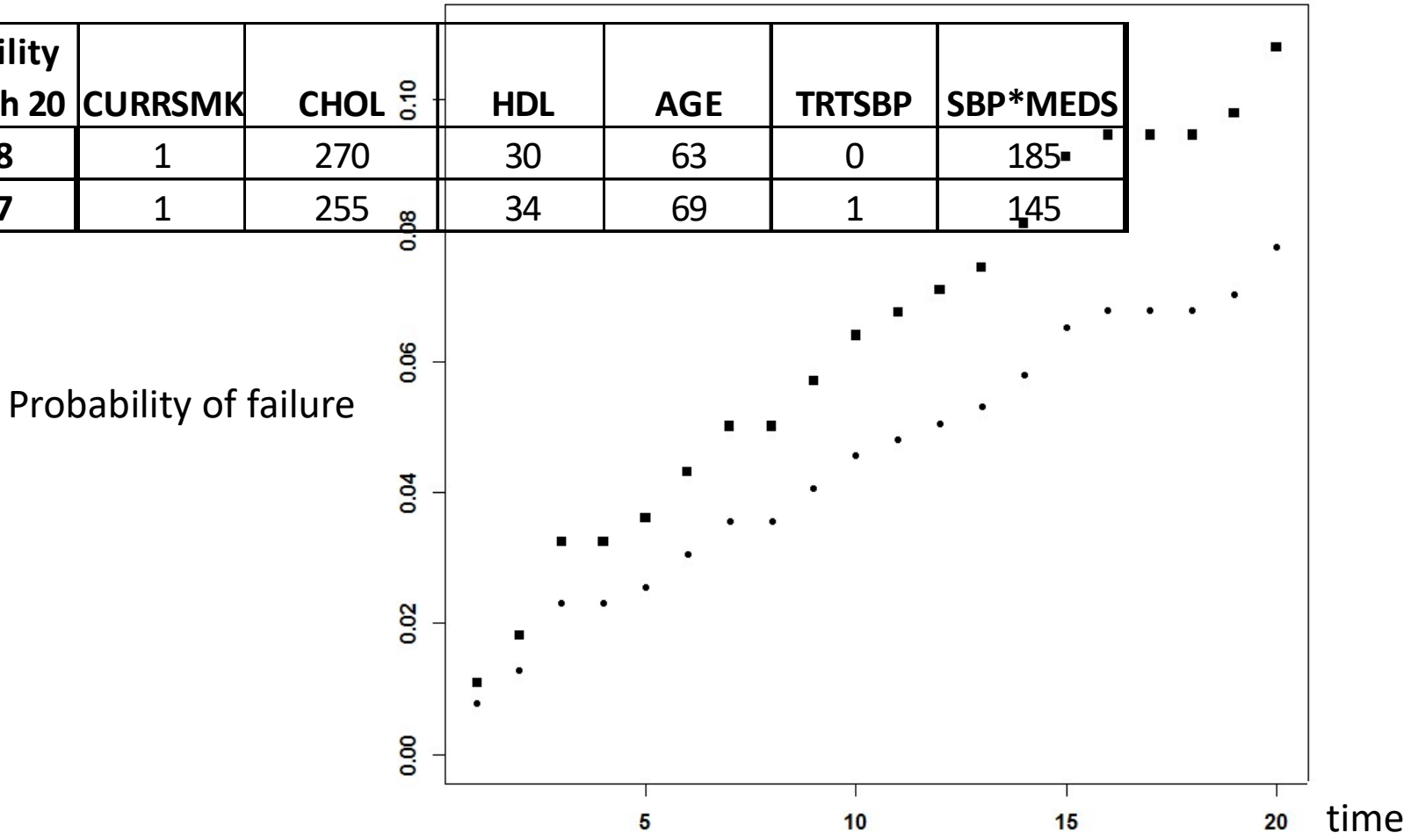
Hazard(t)=BazelineHazard(t) x exp(1.44 * smoking_status- 2.82*lnhdl)

↓

Predicted probability of an event by time (t)

Survival Analysis.
Cox Proportional Hazards Regression Model

probability at month 20	CURRSMK	CHOL	HDL	AGE	TRTSBP	SBP*MEDS
0.108	1	270	30	63	0	185
0.077	1	255	34	69	1	145



Predicted probability of an event by time (t)

Survival Analysis.
Cox Proportional Hazards Regression Model

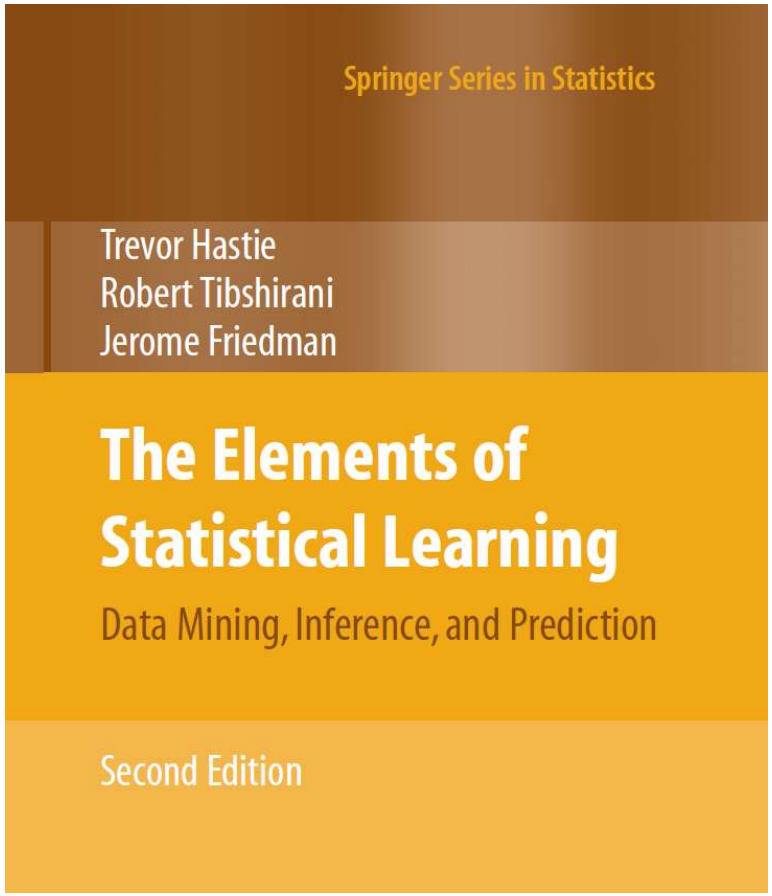
Pros:

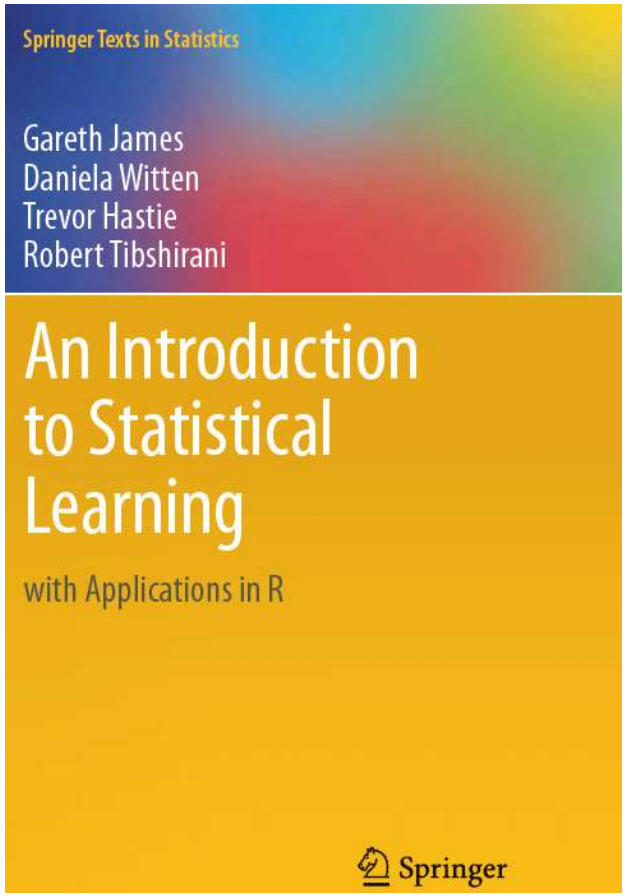
1. Very widely used: many statistical tests are available, implemented in most software packages,
2. Models various types covariates
3. Extended for time-varying covariates
4. Accommodates censoring

Cons:

1. Proportionality of hazards assumption (PH assumption) – stratify
2. Censoring should not be related to probability of the outcome

Statistical Learning Methods





http://www.stanford.edu/~hastie/local.ftp/Springer/OLD/ESLII_print4.pdf

Statistical Learning Methods

- Unsupervised learning (outcome is not known – try to see if there are clusters)
- Supervised Learning (outcome is known – separate data into groups with the goal of predicting the outcome)

Statistical Learning Methods

- Unsupervised learning (outcome is not known – try to see if there are clusters)
- Supervised Learning (outcome is known – separate data into groups with the goal of predicting the outcome)

Statistical Learning Methods

Supervised Learning Methods:

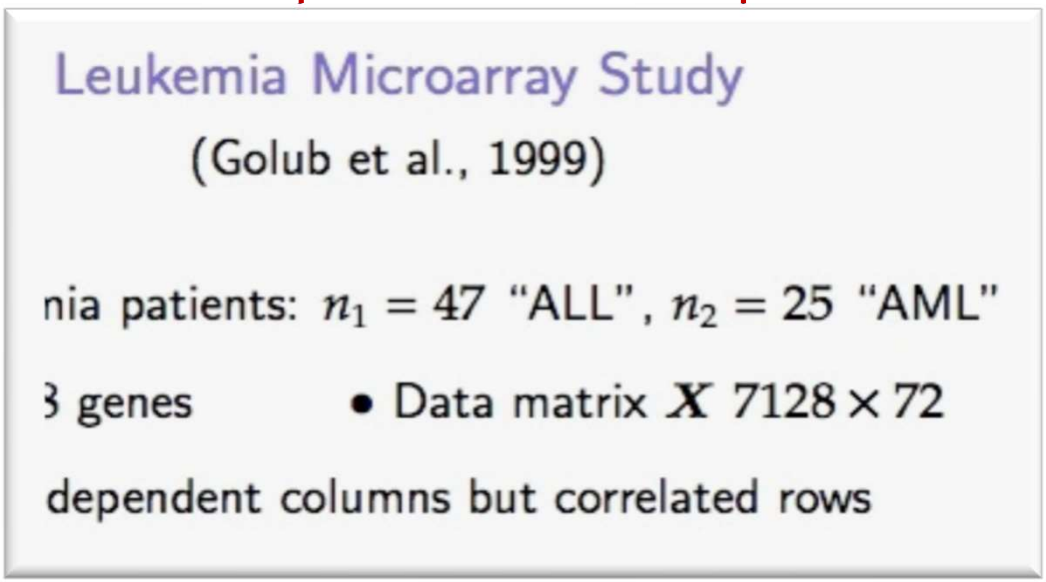
- CART
- Random Forests
- Ridge regression
- Lasso
- Elastic Net
- Neural Networks
- Support Vector Machines

All have a "learning" stage: construct tree(s), build a NN, use iterations to calculate the boundary function. Hence Statistical Learning Methods. Also called training stage.

Ridge Regression
Lasso
Elastic Net

Megavariate data: Feature Selection

- For megavariate data, matrix-based methods won't converge: correlation matrix is not full-rank (Linear Discriminant Analysis-based, Logistic Regression-based)
- Other methods (Support Vector Machines) will overfit: produce a model with perfect separation into classes. Overfitting will create problems with replication.



Shrinkage Methods for Regression-Based Methods

$Log(L)$+penalty#features

- Fits a regression with a boundary on the sum of the coefficients
- Variables that contribute little have reduced coefficients, some are set to 0
- Cross validation used to set boundary

Shrinkage Methods for Regression-Based Methods

$Log(L)$+penalty#features

- Ridge Regression, Lasso regression, Elastic Net were developed for continuous outcome
- Adapted for 0/1 outcomes and retain the name or can be called also:
  - Ridge Regression: logistic regression with quadratic regularization, logistic regression with L_2 penalty
  - Lasso: logistic regression with L_1 penalty
  - Elastic Net: logistic regression with elastic net penalty

Ridge Regression; Lasso Regression

- β^{ridge} minimizes

$$Log(L) + \lambda \sum_{j=1}^{p} \beta_j^2$$

- β^{lasso} minimizes

$$Log(L) + \lambda \sum_{j=1}^{p} |\beta_j|$$

Ridge Regression; Lasso Regression

- β^{ridge} minimizes

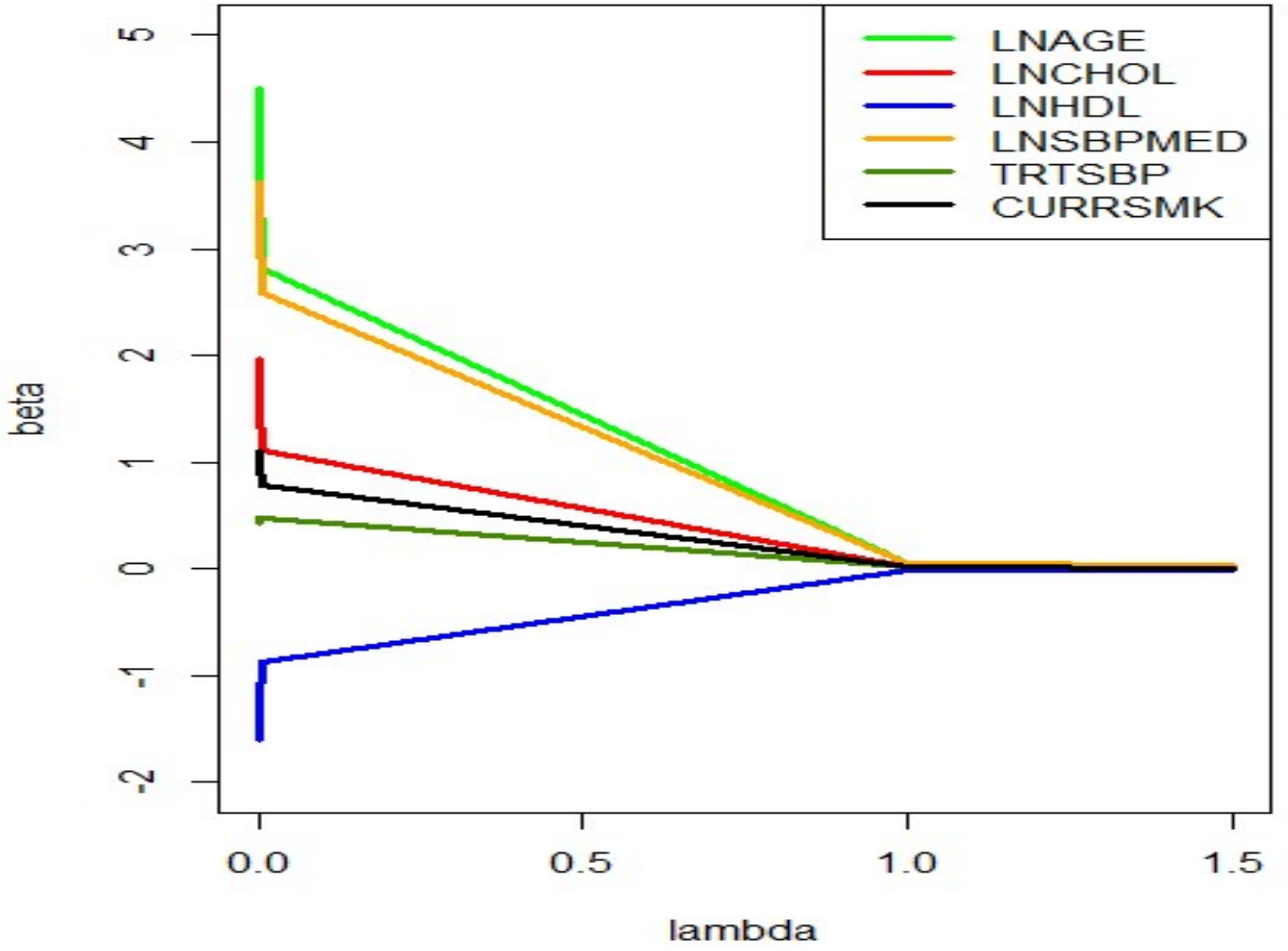
$$Log(L) + \lambda \sum_{j=1}^{p} \beta_j^2$$

- β^{lasso} minimizes

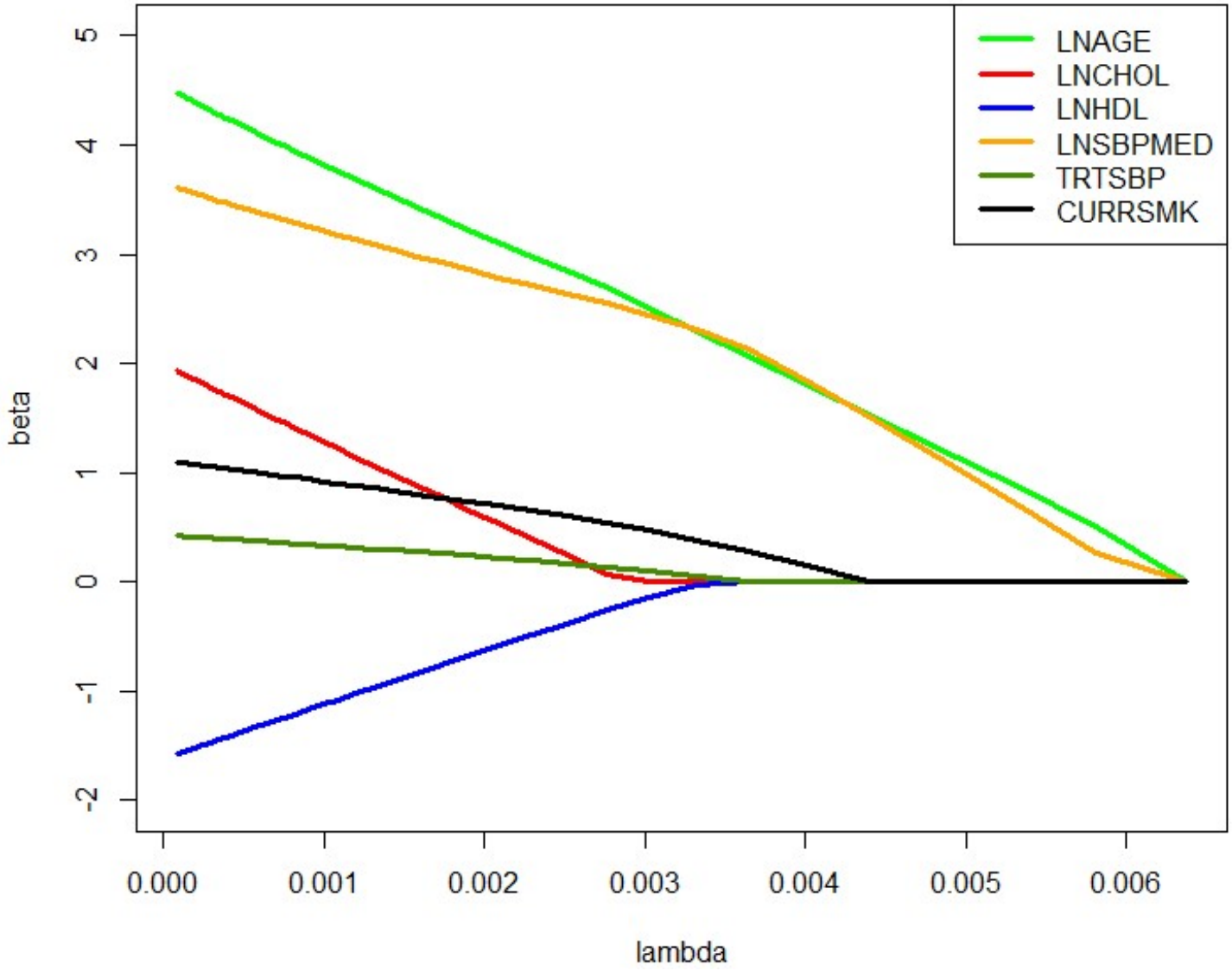
$$Log(L) + \lambda \sum_{j=1}^{p} |\beta_j|$$

Always standardize because scale-dependent (R does it by default)

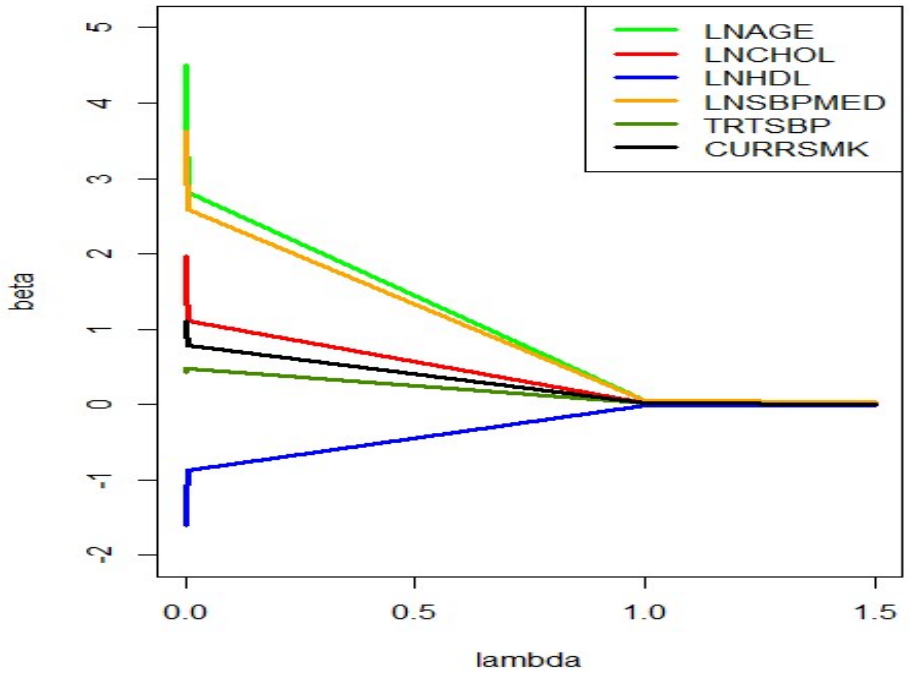
Ridge Regression



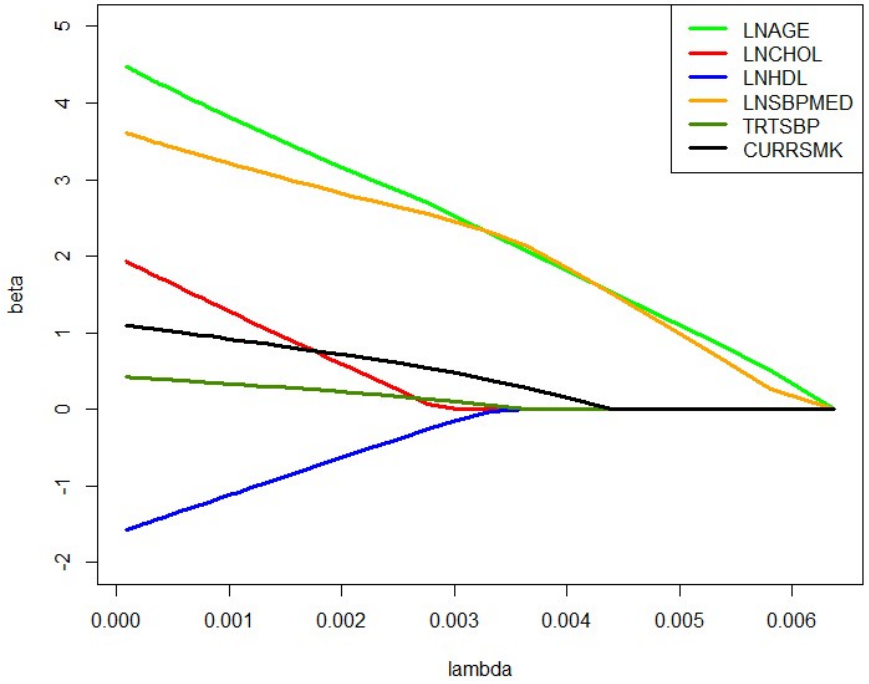
Lasso Regression



Ridge



Lasso



Ridge regression keeps all variables in the model hence need to perform feature selection separately

Lasso sets some coefficients to zero and was designed specifically for the purposes of feature selection

Elastic Net

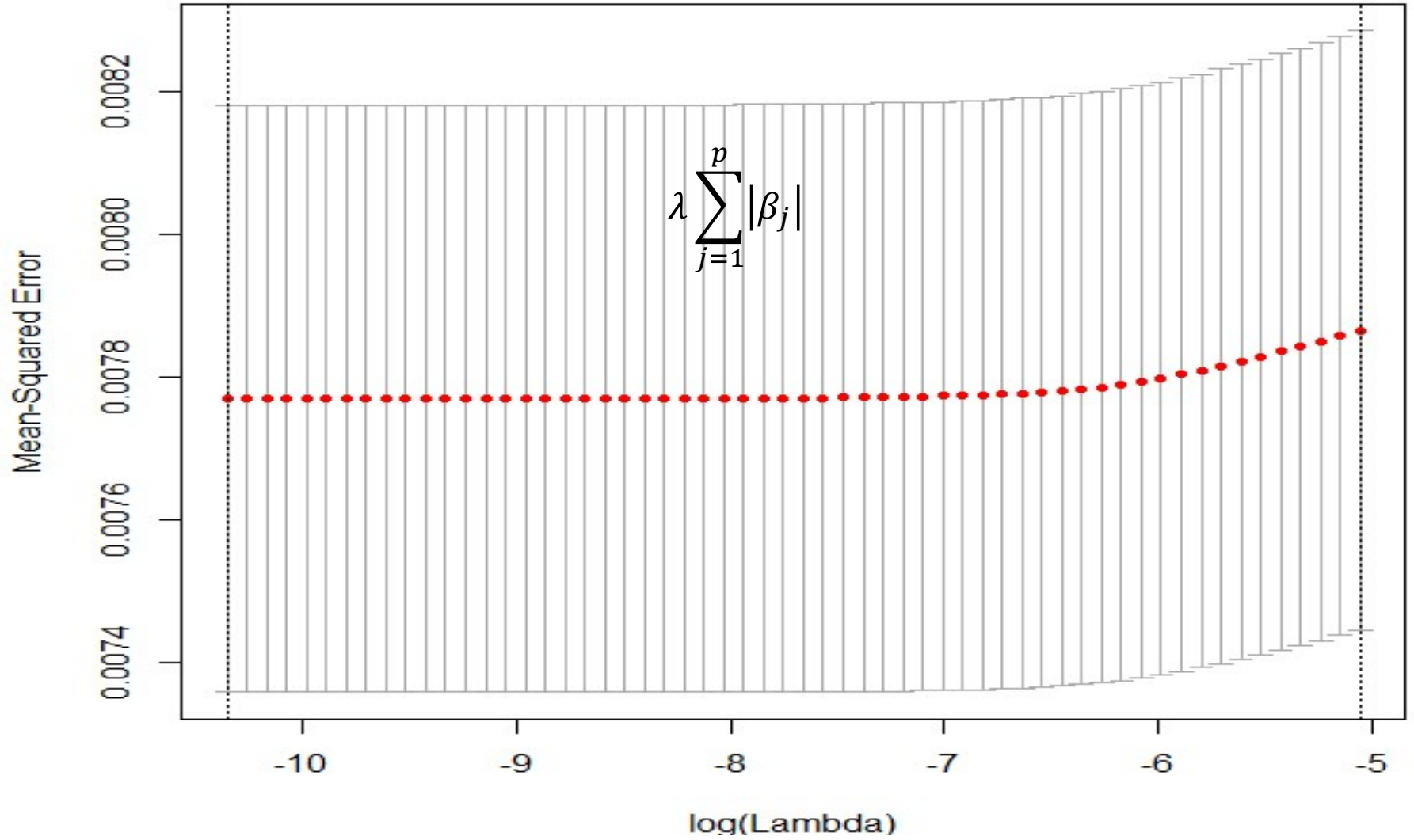
- Elastic Net is a compromise between Ridge and Lasso:

$$\lambda \left[\sum_{j=1}^{p} \alpha \, \beta_j^2 + (1 - \alpha) \left| \beta_j \right| \right]$$

- λ is a shrinkage parameter
- α is mixing parameter

Ridge and Lasso

Select optimal λ using cross-validation



Elastic Net

Select optimal α, λ using cross-validation

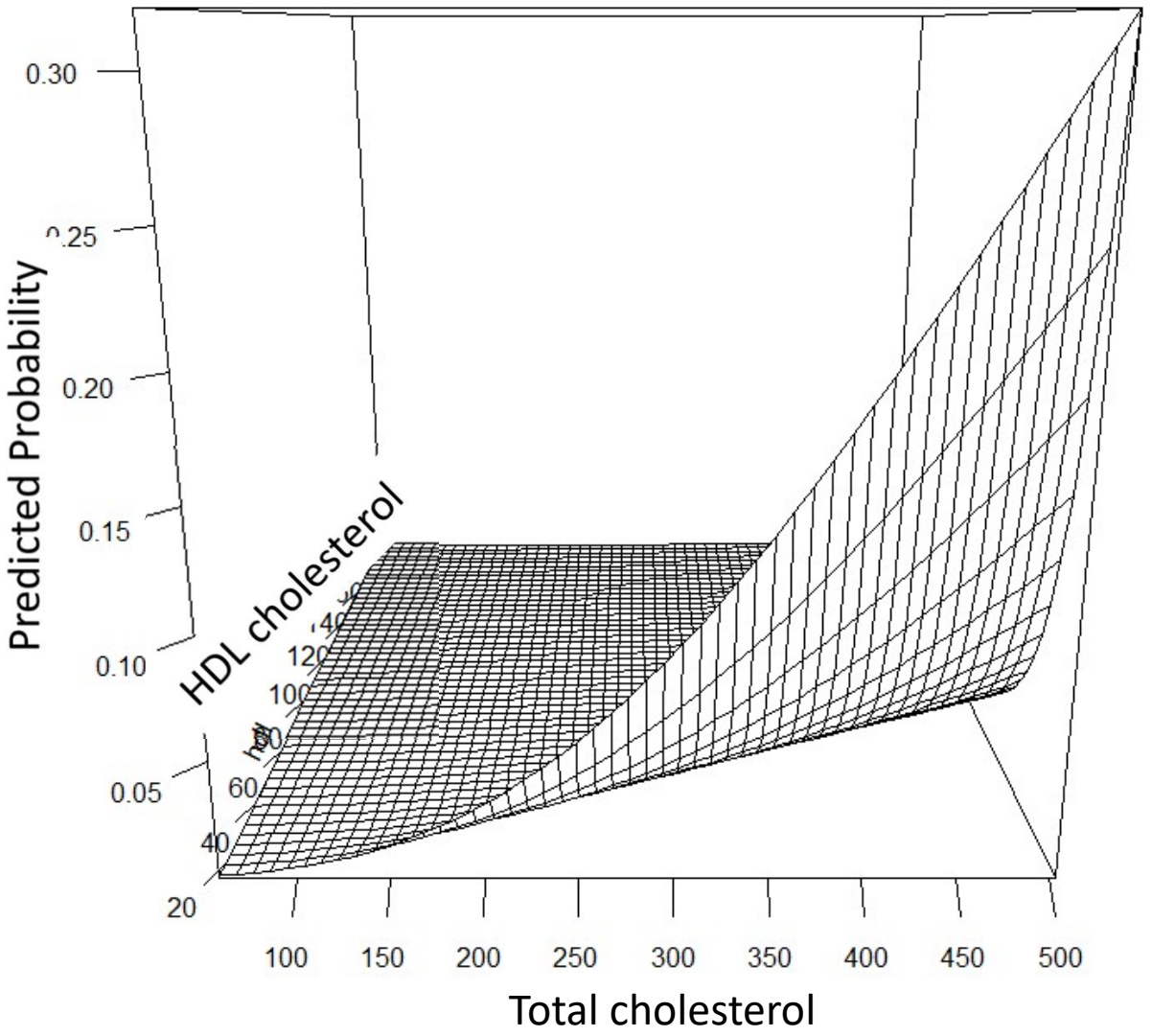
$$\lambda \left[\sum_{j=1}^{p} \alpha \, \beta_j^2 + (1-\alpha) \left| \beta_j \right| \right]$$

If more than one meta-parameter then do multidimensional optimization

Classification And Regression Trees (CART)

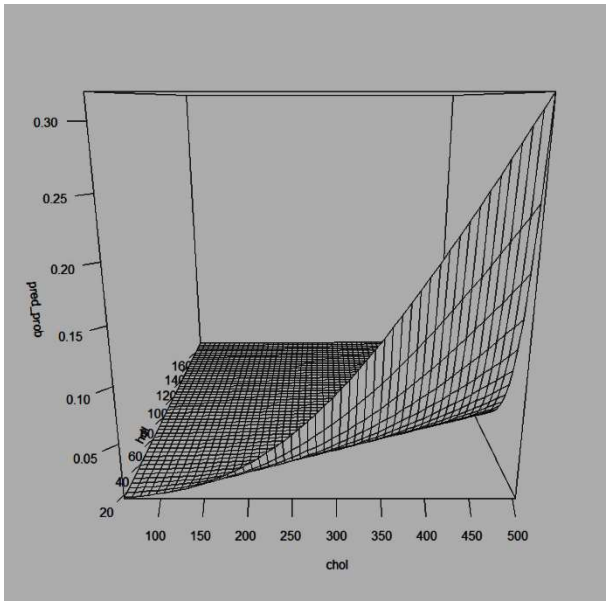
From Logistic Regression to Supervised Learning Methods

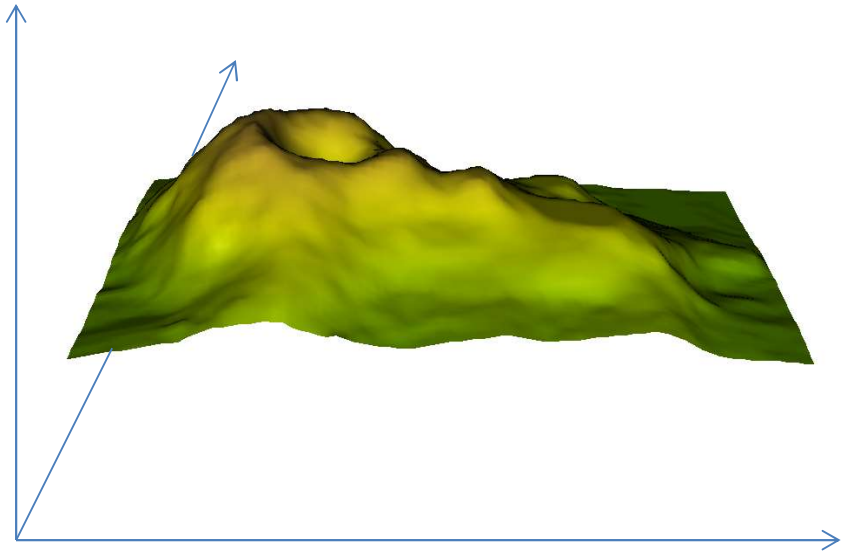
Predicted probability versus total cholesterol and hdl cholesterol



From Logistic Regression to Supervised Learning Methods

Pred. prob

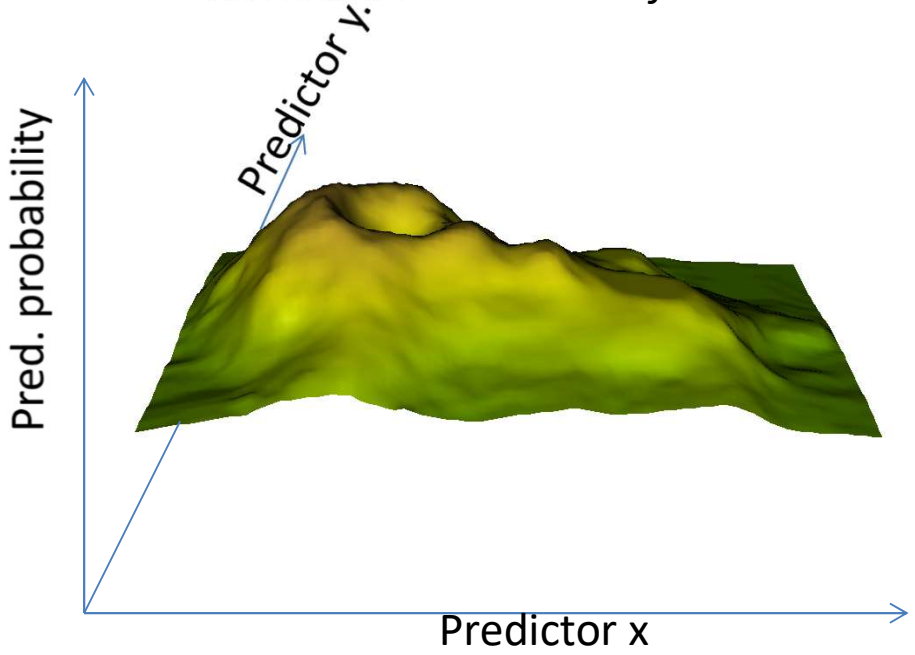




Supervised Learning.
Classification and Regression Trees (CART)

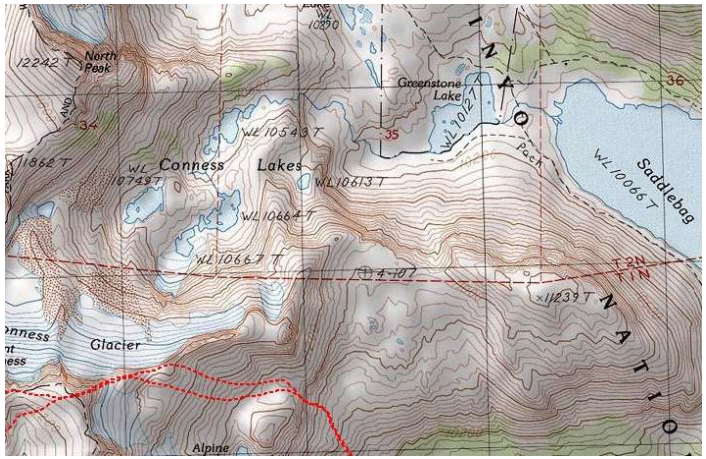
1980s Breiman, Friedman, Olshen and Stone introduced *classification and regression trees*

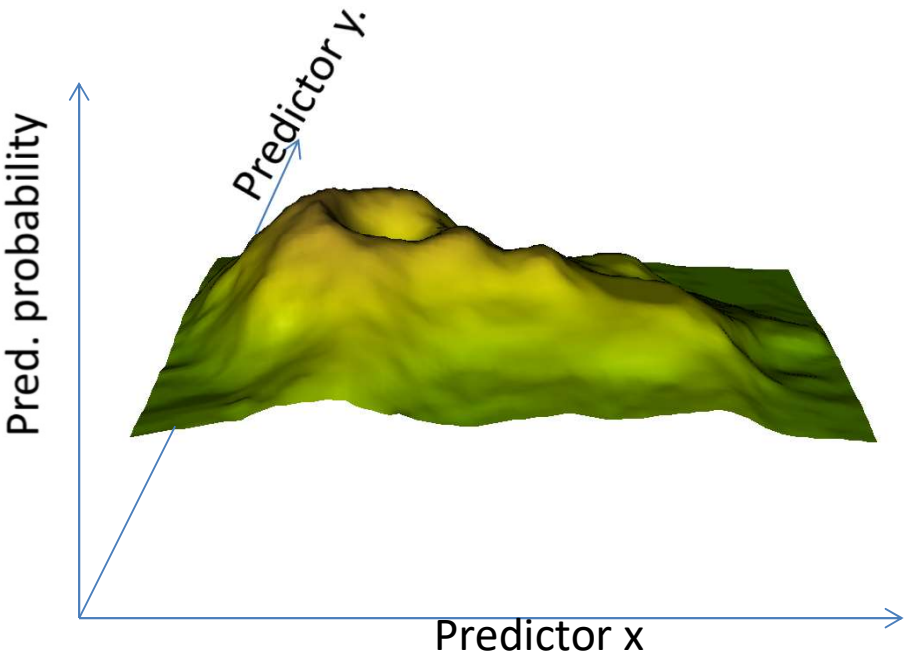
JH Friedman



Era of more powerful computers. The first example in this talk of a statistical learning method

Supervised Learning.
Classification and Regression Trees (CART)

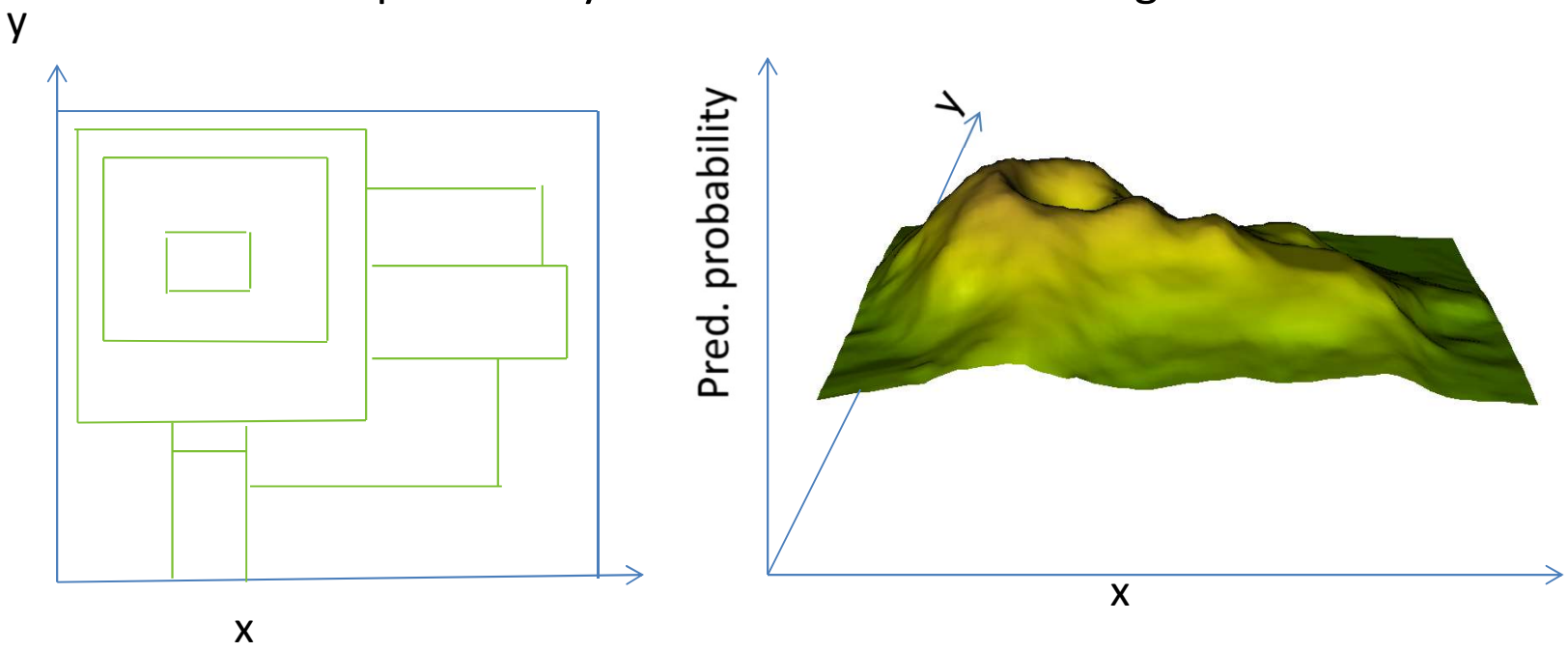




Main idea: Split the space into regions similarly to topographic maps and model constant probability of an outcome in each region

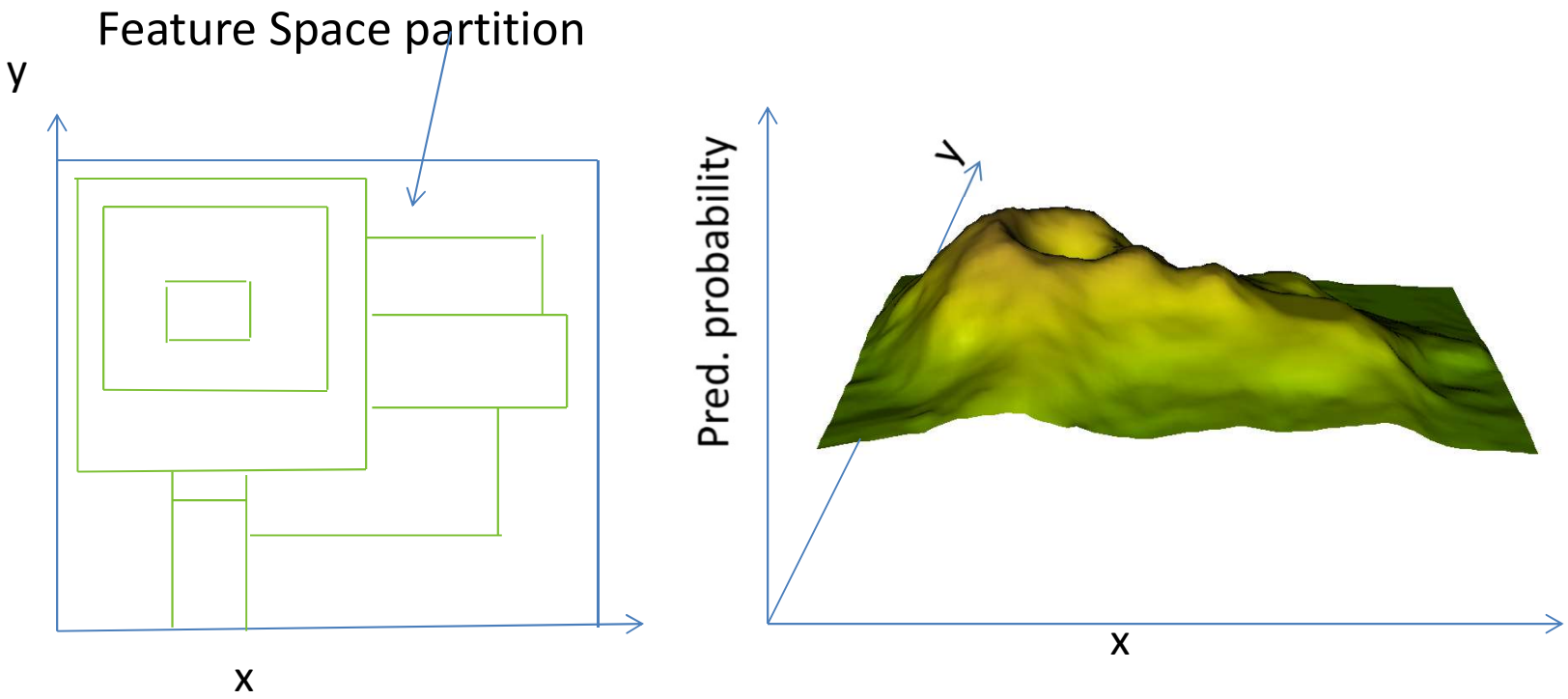
Supervised Learning.
Classification and Regression Trees (CART)

Main idea: Split the space into regions and model constant probability of an outcome in each region



Supervised Learning.
Classification and Regression Trees (CART)

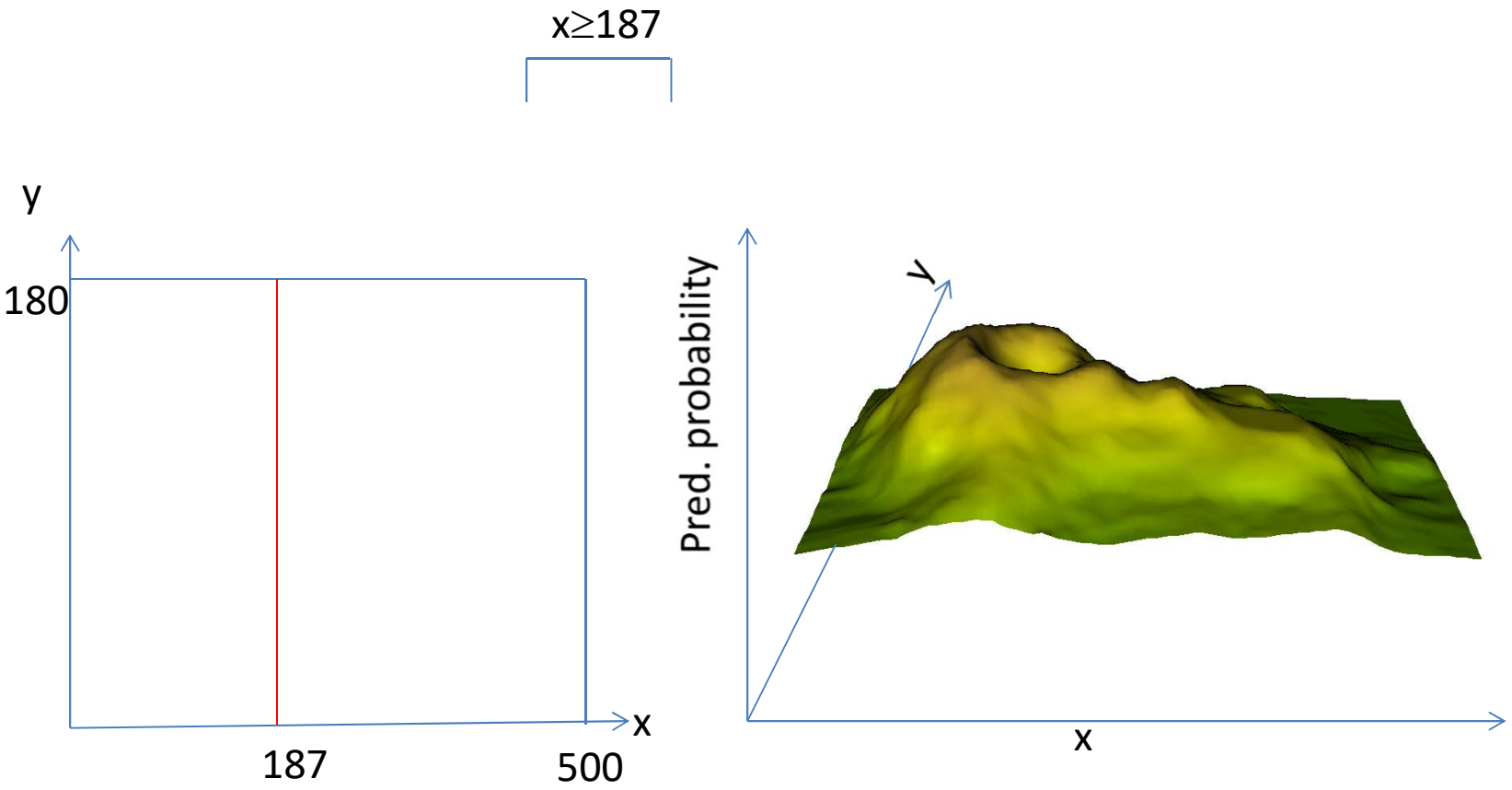
Terminology



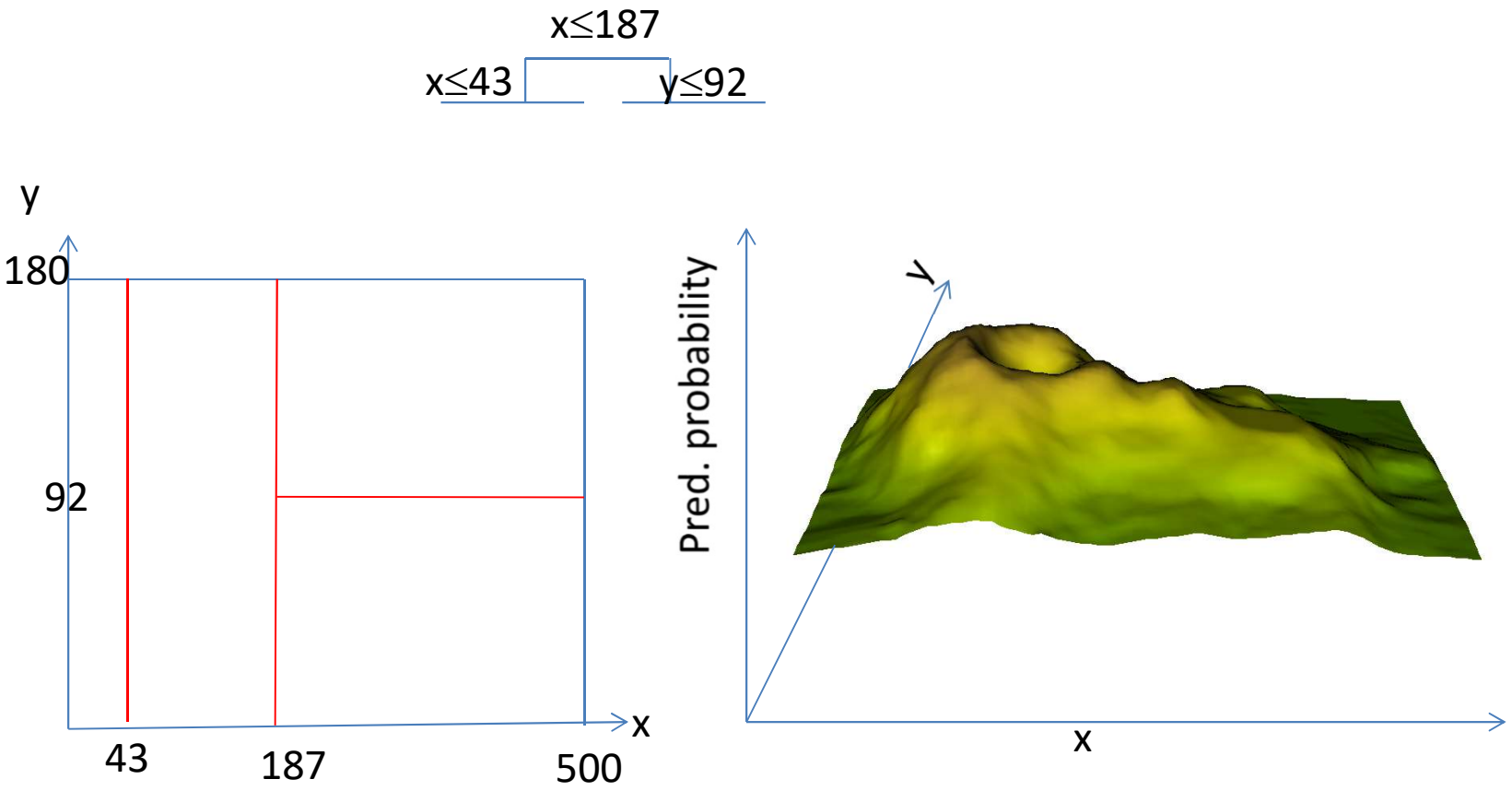
Features, Inputs, Attributes =predictor variables

Response=outcome variable

Supervised Learning.
Classification and Regression Trees (CART)

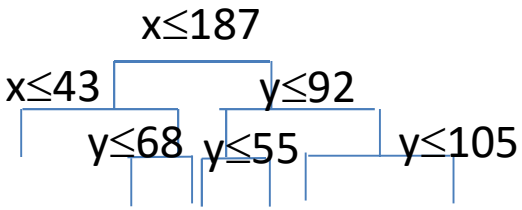


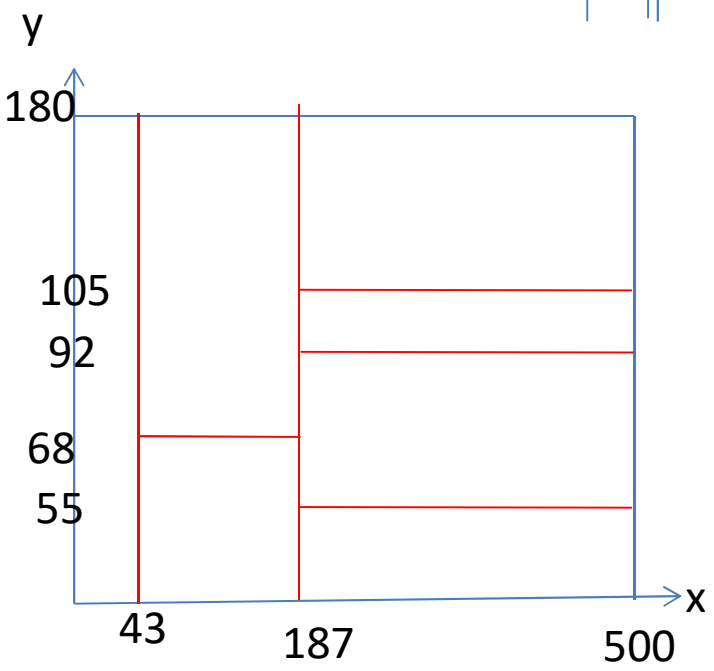
Supervised Learning.
Classification and Regression Trees (CART)

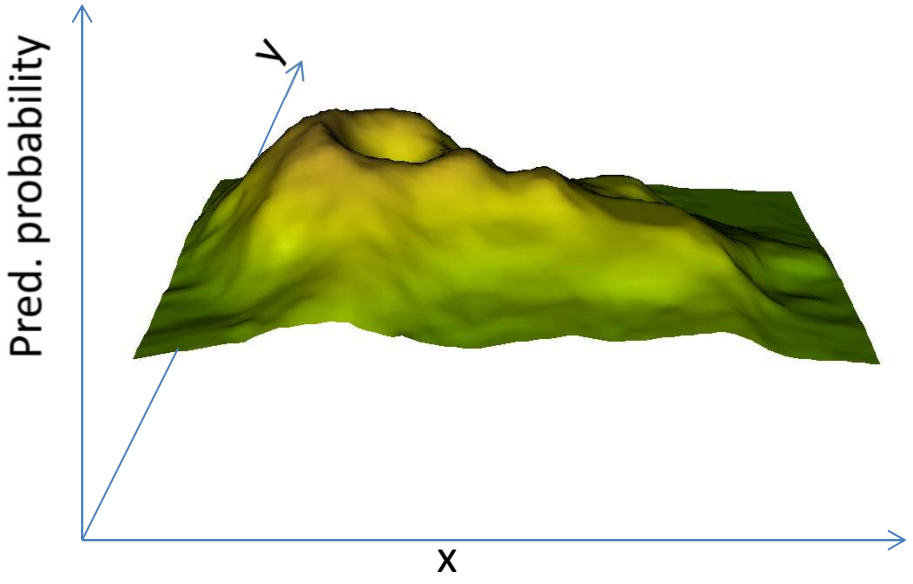


At each node select **the best predictor** for splitting **and the best cutoff value**

Supervised Learning.
Classification and Regression Trees (CART)

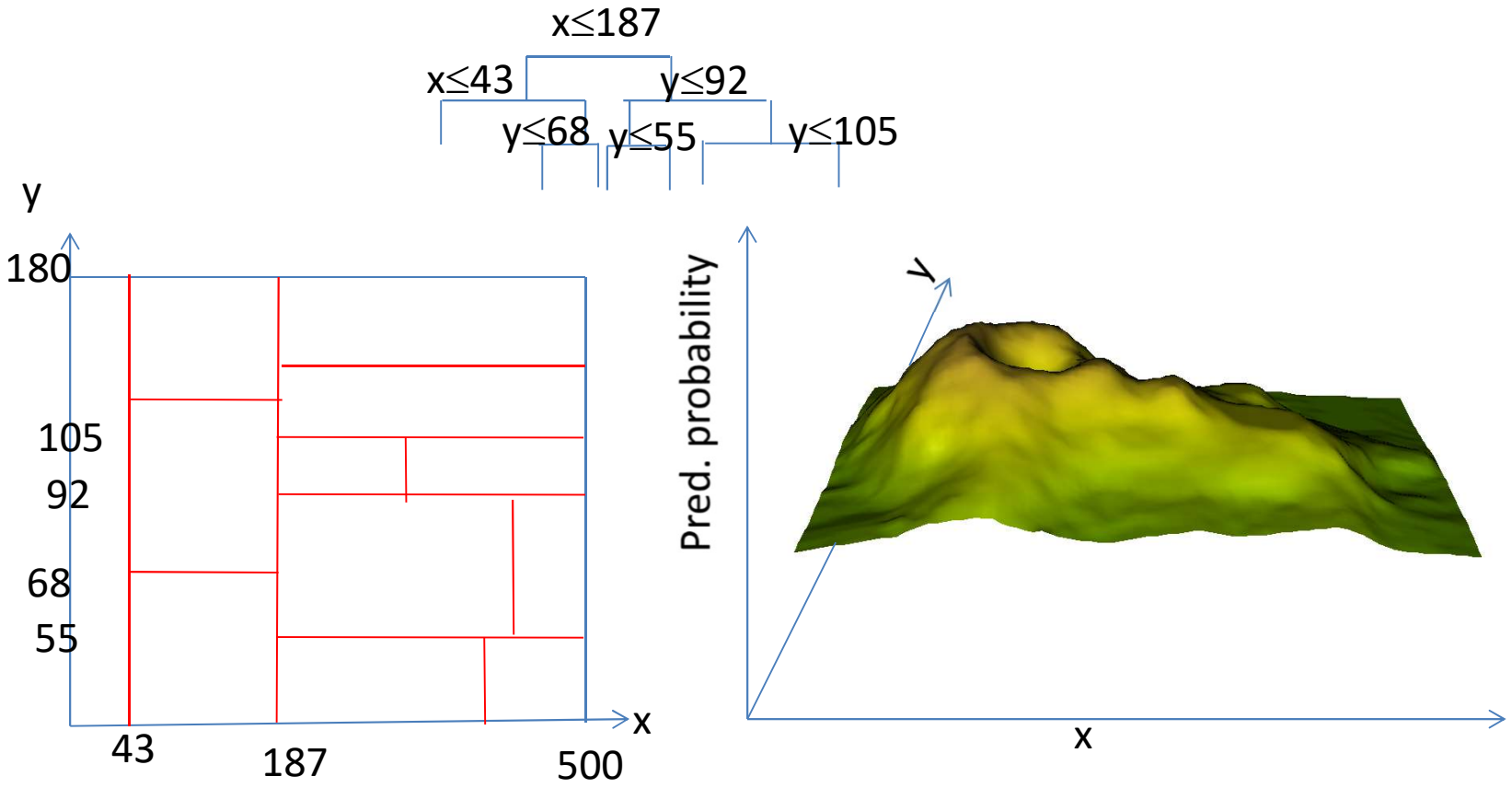






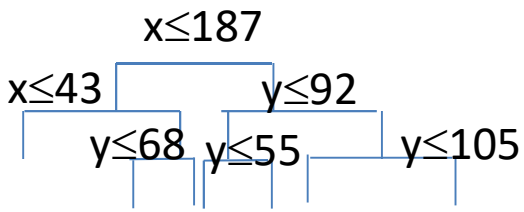
At each node select **the best predictor** for splitting and **the best cutoff value**

Supervised Learning.
Classification and Regression Trees (CART)



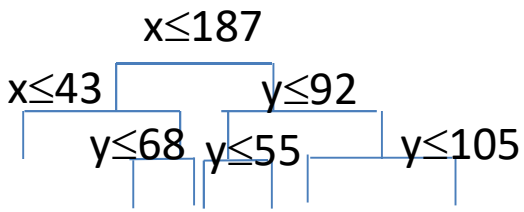
At each node select **the best predictor** for splitting and **the best cutoff value**

Supervised Learning.
Classification and Regression Trees (CART)



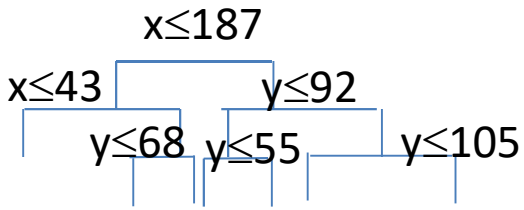
- At each node select **the best predictor** for splitting and **the best cutoff value**

Supervised Learning.
Classification and Regression Trees (CART)



- At each node select **the best predictor** for splitting and **the best cutoff value**
- The goal is to separate cases from non-cases as much as possible.

Supervised Learning.
Classification and Regression Trees (CART)



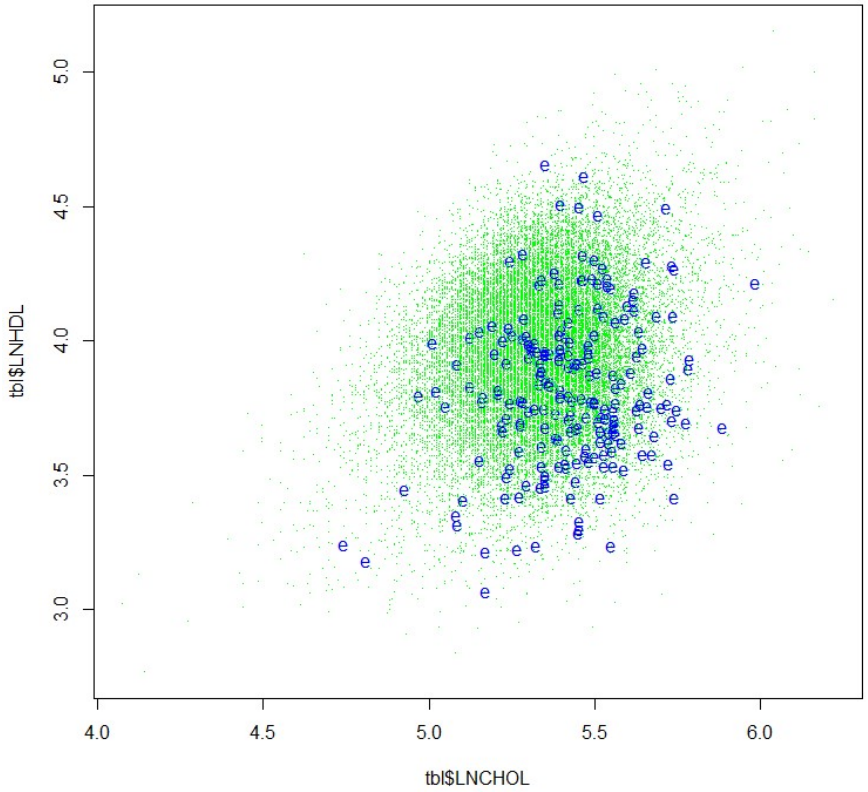
- At each node select **the best predictor** for splitting and **the best cutoff value** based on the node impurity measures: misclassification error or Gini index or cross-entropy or deviance
- The goal is to separate cases from non-cases as much as possible.

CART algorithm is an example of the **greedy algorithm.**

A **greedy algorithm** is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce
an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time.

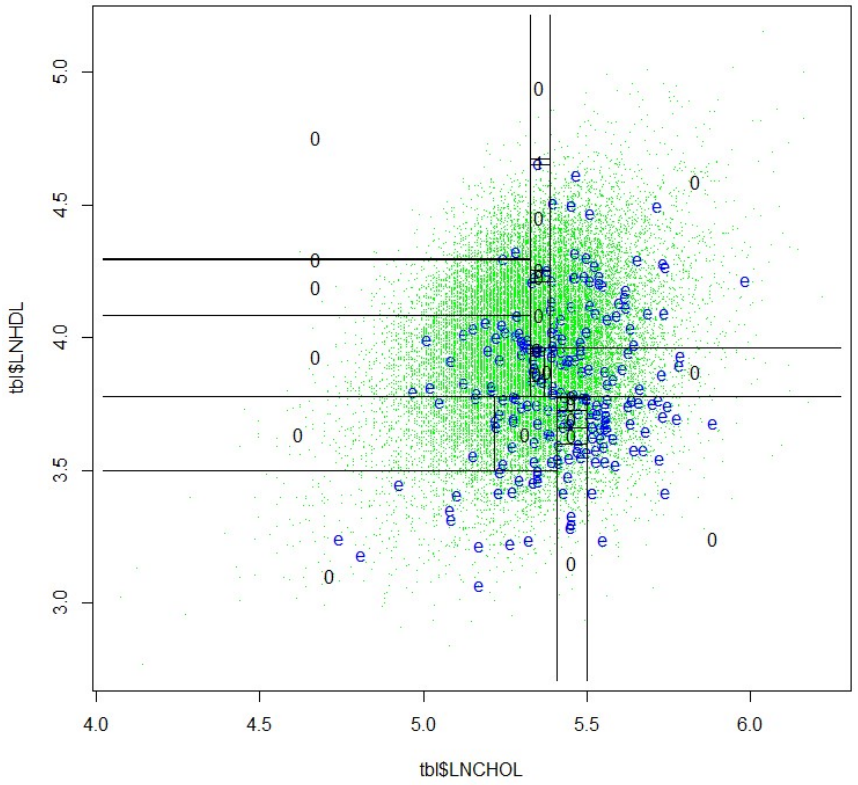
Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)



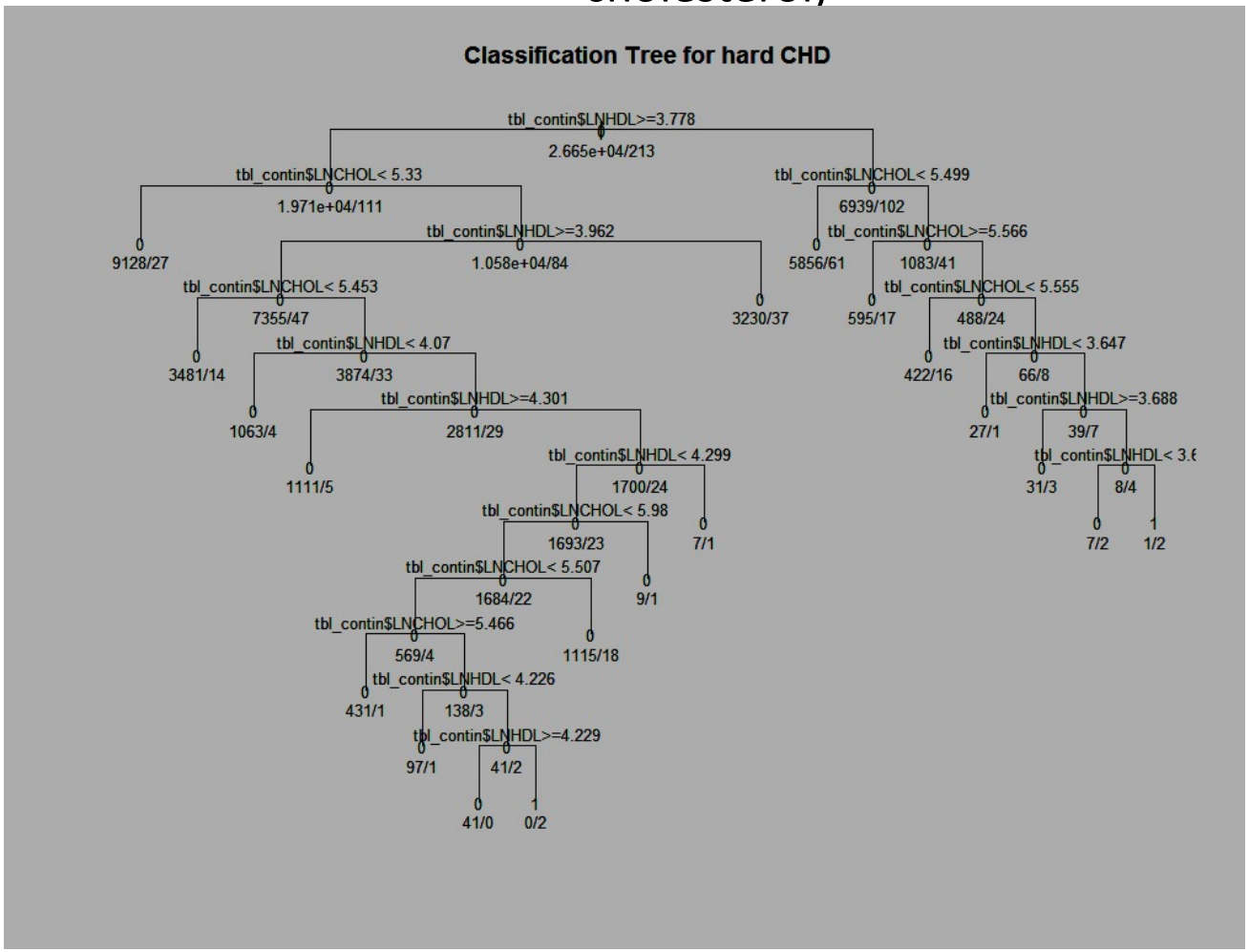
Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)

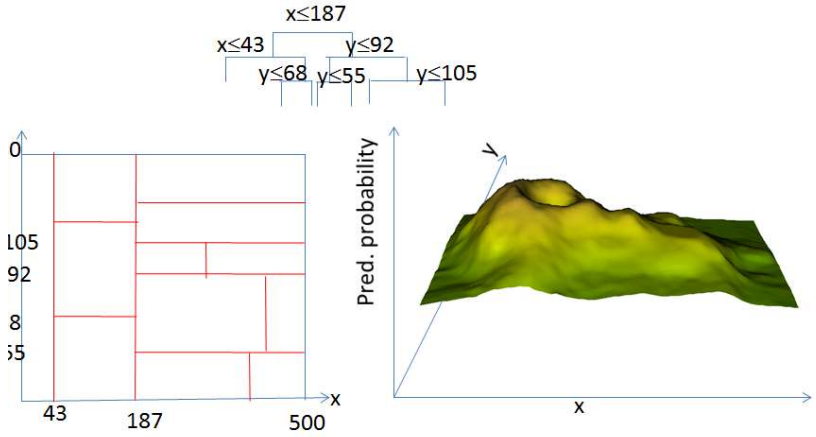


Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)



Supervised Learning.
Classification and Regression Trees (CART)



It is very easy to create a very large tree.
Large tree results in over-fitting the data.
So **prune** the tree!

Cost-complexity pruning:
By collapsing internal nodes, select the **least complex** but **most accurate** tree:
Tradeoff between the **size** and **goodness of fit of the tree**

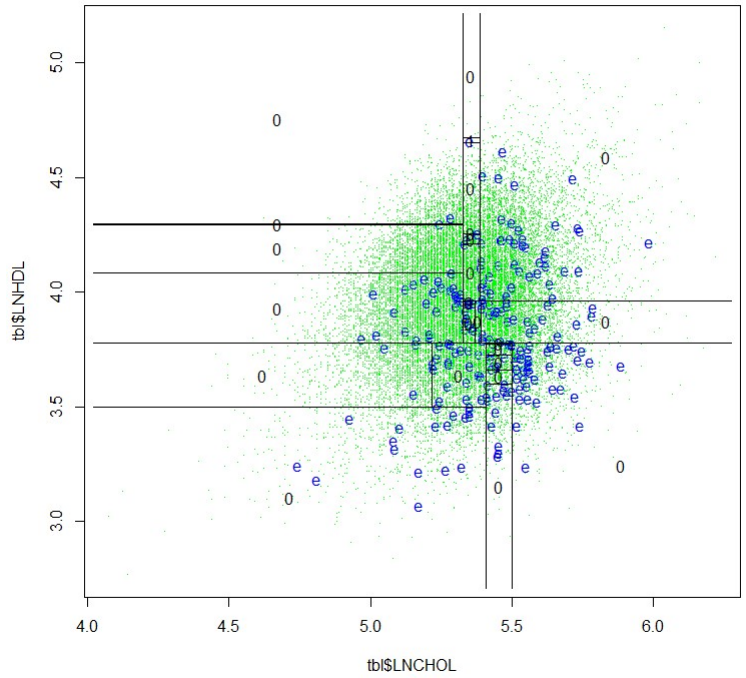
Misclassification error; Gini index; Cross-entropy or deviance

Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)

Advantages:

- these models go after interactions immediately
- Easy to explain and interpret



Supervised Learning.
Classification and Regression Trees (CART)

CARTS can be unstable.

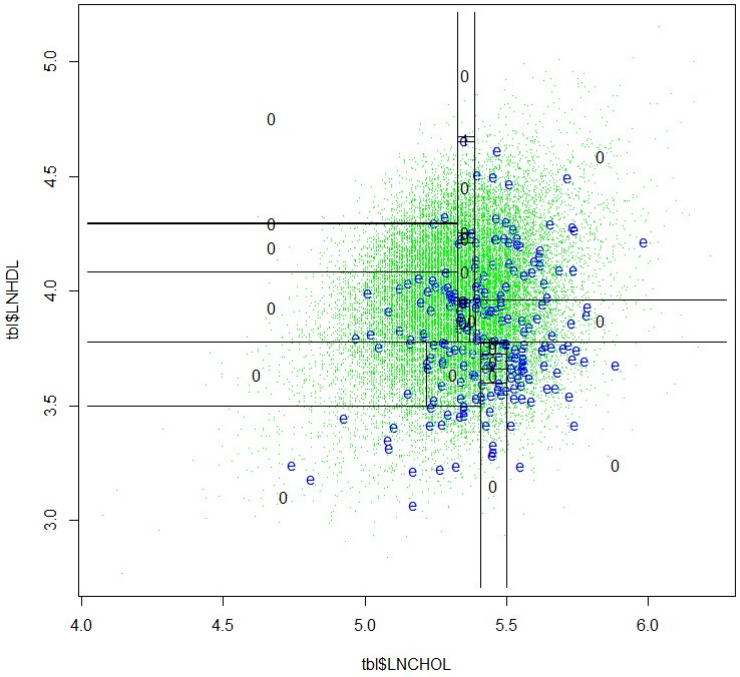
whole sample

removed 1% of obs

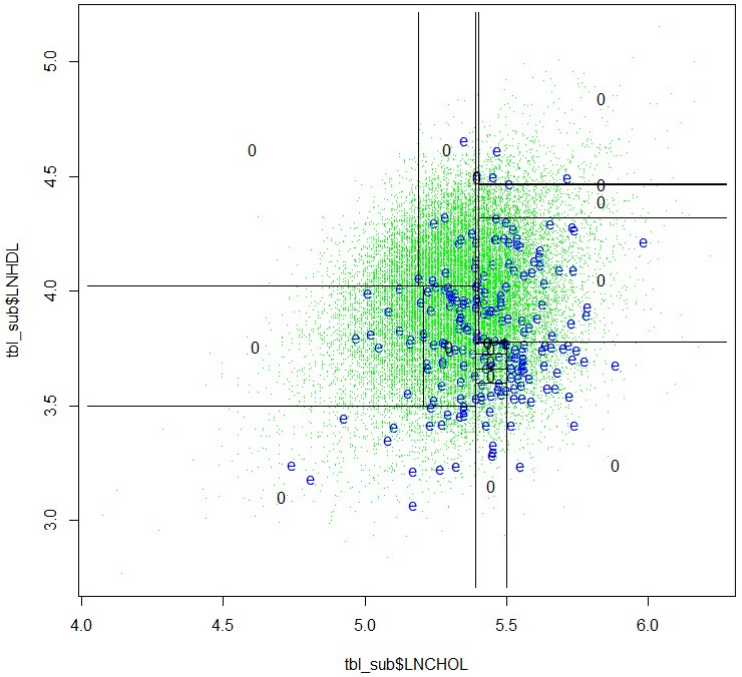
Supervised Learning.
Classification and Regression Trees (CART)

CARTS can be unstable.

whole sample



removed 1% of obs



Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)

Advantages:

- These models go after interactions immediately
- Easy to explain

Disadvantages:

- Results can be unstable

Supervised Learning.
Classification and Regression Trees (CART)

Example: predicting hard CHD using log(total cholesterol) and log(HDL cholesterol)

Advantages:

- These models go after interactions immediately
- Easy to explain

Disadvantages:

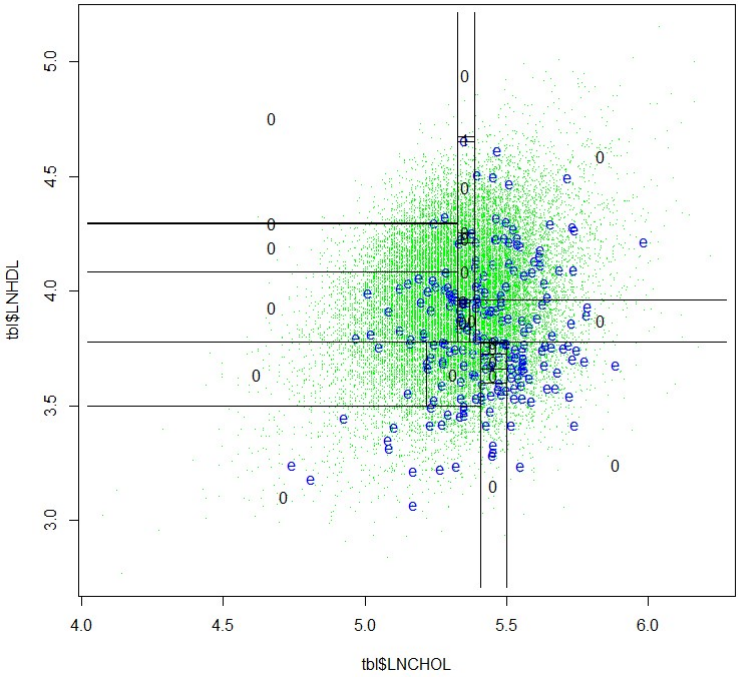
- Results can be unstable
- Hard to access uncertainty in inference about trees
- Hard to model linear relationships – allow linear splits

Random Forests

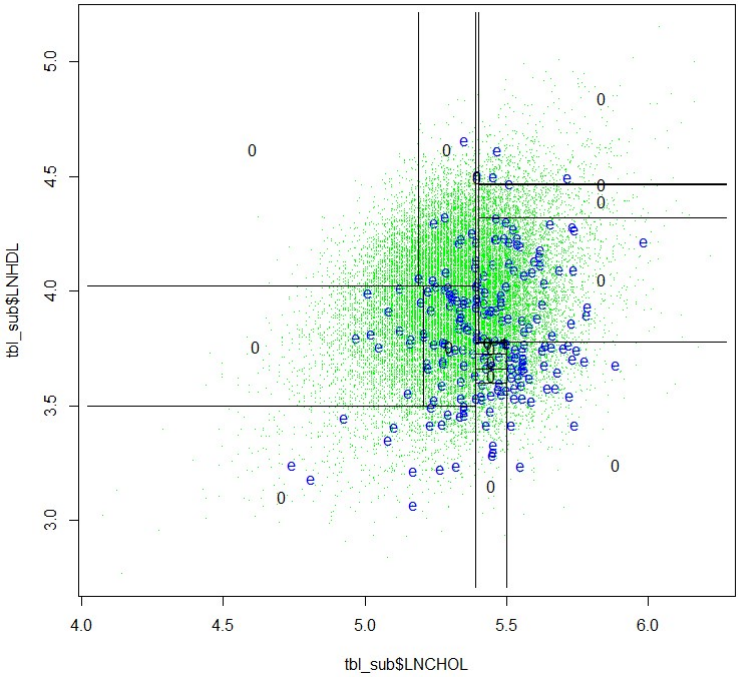
Extensions of CARTs: Random Forests. Main Idea.

1. CARTS can be unstable.

whole sample



removed 1% of obs



Extensions of CARTs: Random Forests. Main Idea.

1. CARTS can be unstable.
2. To stabilize them: main idea:

An average has lower variance than the random variable, therefore grow several trees and classify an observation by "majority vote" (called "bagging")

Also called a committee of trees

Decision is made by consensus

Extensions of CARTs: Random Forests. Main Idea

1. In 2001 Leo Breiman and Adele Cutler

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)

Showed that final prediction works better (is much more stable) if trees are uncorrelated.

Ways to de-correlate the trees:

- Select random set of candidate predictors

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node

   Can be as many as sqrt(p) and as few as one

   1. select a set of candidate predictors at random (hence Random)

Showed that final prediction works better (is much more stable) if trees are uncorrelated.

Ways to de-correlate the trees:

- Select random set of candidate predictors

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)

Showed that final prediction works better (is much more stable) if trees are uncorrelated.

Ways to de-correlate the trees:

- Select random set of candidate predictors
- Use slightly different datasets (bootstrapping)

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node and optimal threshold

Extensions of CARTs: Random Forests. Main Idea

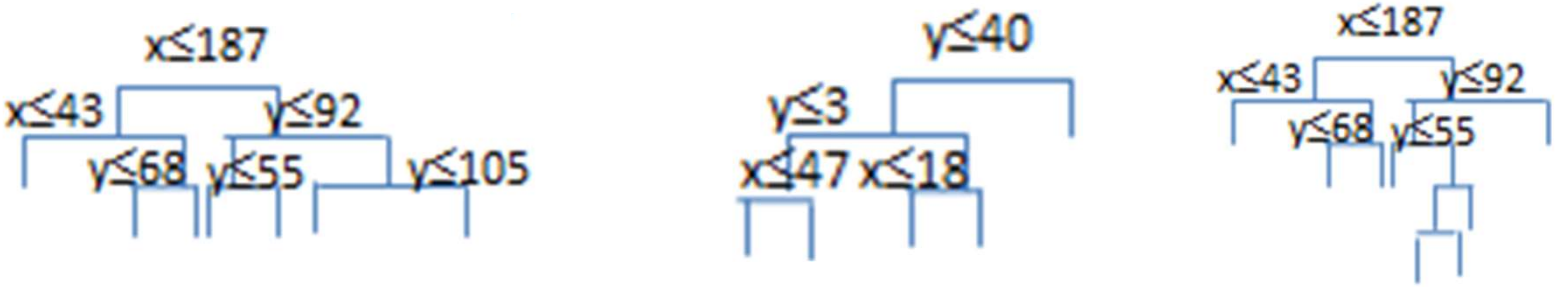
Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node and optimal threshold
    3. Repeat

Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

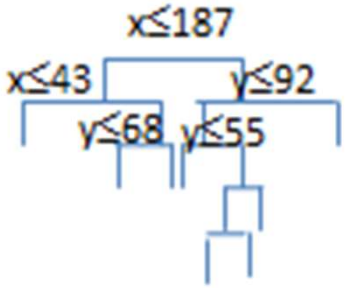
1. At each node
    1. select a set of candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node and optimal threshold
    3. Repeat
2. Create several trees (hence Forest)



Extensions of CARTs: Random Forests. Main Idea

Add two more steps:

1. At each node
    1. select a set of candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node
    3. Repeat
2. Create several trees (hence Forest)
3. Classify an observation by taking the majority vote from the trees



Extensions of CARTs: Random Forests. Main Idea

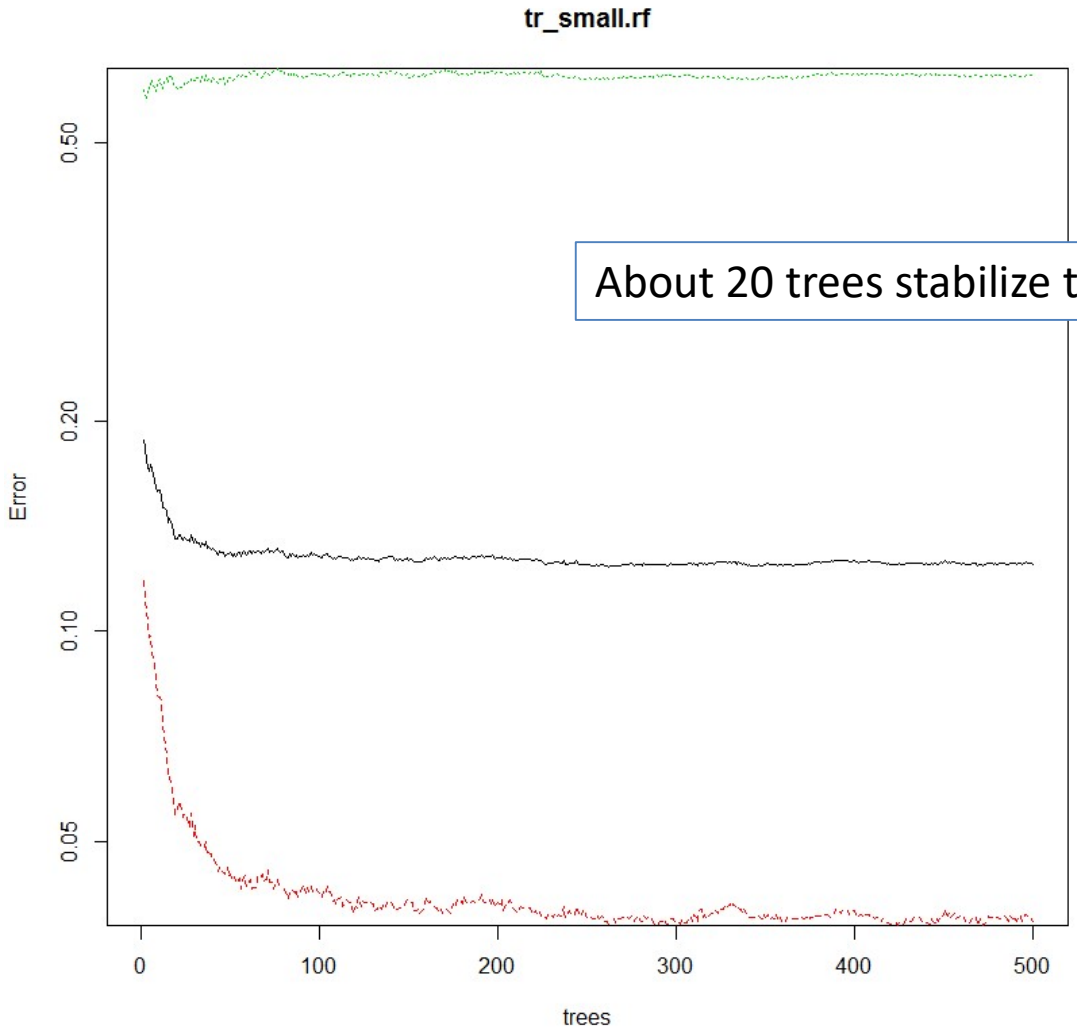
Add two more steps:

Can be as many as sqrt(p) and as few at one

1. At each node
    1. select candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node
    3. Repeat
2. Create several trees (hence Forest)
3. Classify an observation by taking the majority vote from the trees

Called Bagging

Extensions of CARTs: Random Forests. Main Idea



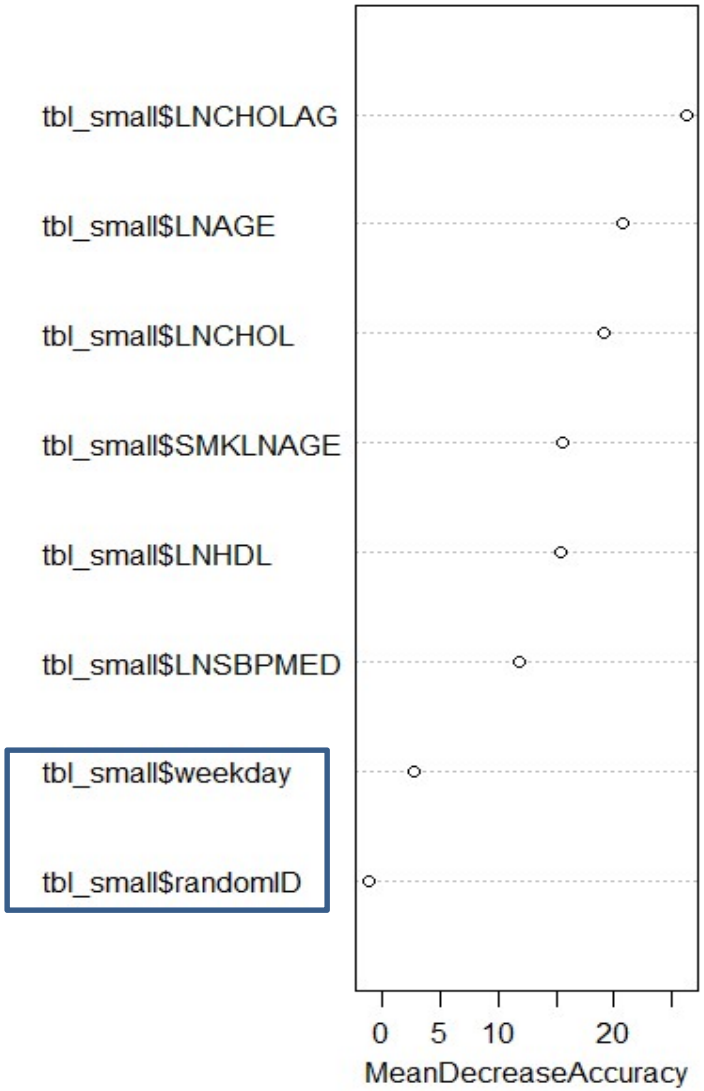
About 20 trees stabilize the prediction in our data

Extensions of CARTs: Random Forests.

Pros

1. One of the best-performing methods – stable
2. Accurate
3. Robust to outliers
4. Internal estimates of error, strength, variable importance
5. Simple to train and tune
6. Often used in variable selection

Extensions of CARTs: Random Forests.



Random Forests are

often used in variable selection

Short review of predictive models. Random Forests

Pros:

1. One of the best-performing methods – stable
2. Accurate
3. Robust to outliers
4. Internal estimates of error, strength, variable importance
5. Simple to train and tune
6. Often used in variable selection
7. Popular
8. Implemented in several software packages

Extensions of CARTs: Random Forests.

Pros:

1. One of the best-performing methods – stable
2. Accurate
3. Robust to outliers
4. Internal estimates of error, strength, variable importance
5. Simple to train and tune
6. Often used in variable selection
7. Popular
8. Implemented in several software packages

Cons:

1. A “black-box” model – does not give the estimates of model or function
2. Limited interpretability

Extensions of CARTs: Random Forests. Boosting. Main Idea.

Add two more steps:

1. At each node
    1. select candidate predictors at random (hence Random)
    2. Find the best predictor to use in the split at that node
2. Create several trees (hence Forest)
3. Classify an observation by taking the majority vote from the trees. Average predicted probabilities.

Called Bagging

Additional way to combine trees: do boosting.

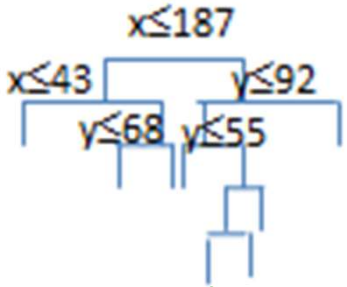
Hastie et al: "Boosting is one of the most powerful learning ideas introduced in the last twenty years"

Extensions of CARTs: Random Forests. Boosting. Main Idea.

1. Calculate the first tree.
2. Use this tree to classify observation in your data.
3. Make misclassified observations “more important” by assigning them large weight.
4. Calculate new tree using weighted data.
5. Repeat M times.
6. For each tree calculate its weight – trees that perform better get more weight.
7. Calculate final prediction by taking weighted average of trees’s predictions.

weight=.03

weight=.2



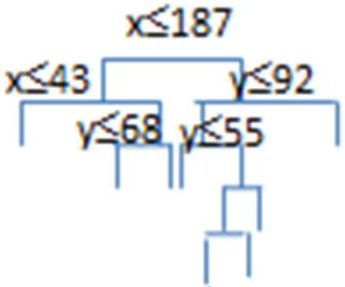
weight=.77

Extensions of CARTs: Random Forests. Boosting. Main Idea.

Boosting is not limited to CART but can be used with other prediction rules.

weight=.03

weight=.2

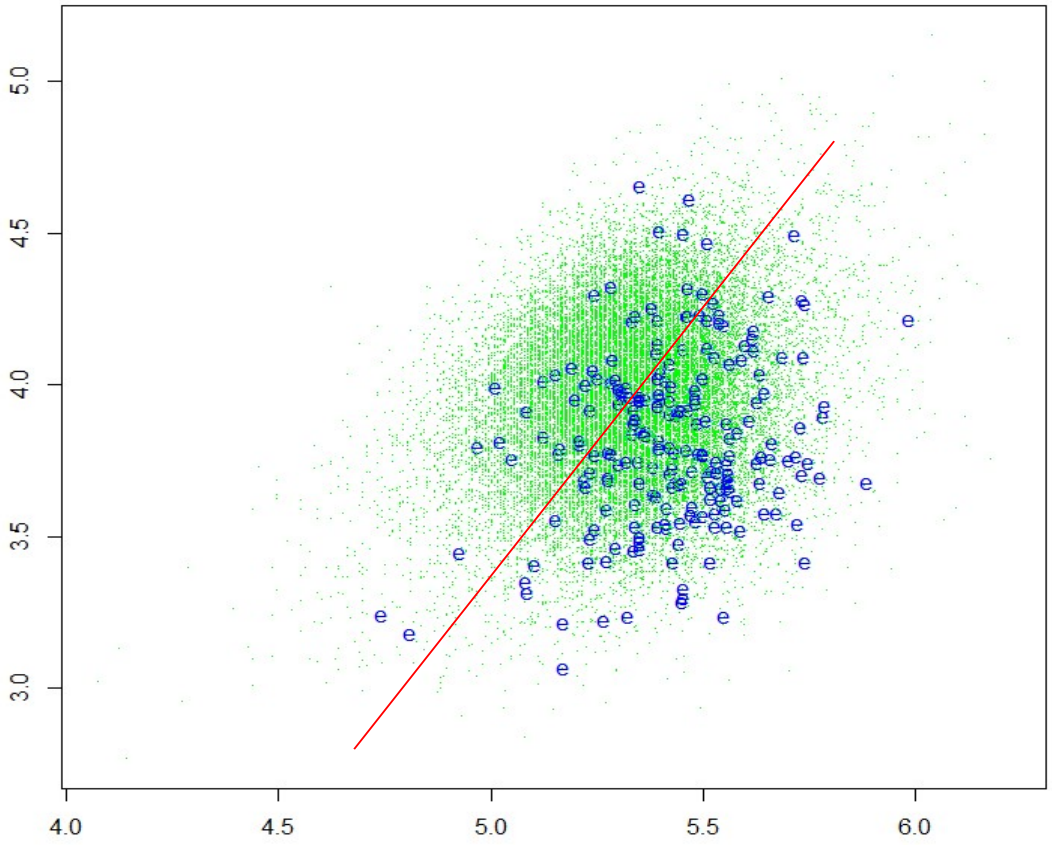


weight=.77

Support Vector Machines (SVM)

Support Vector Machines (SVM)

LDA, logistic regression results in linear boundary:



Support Vector Machines

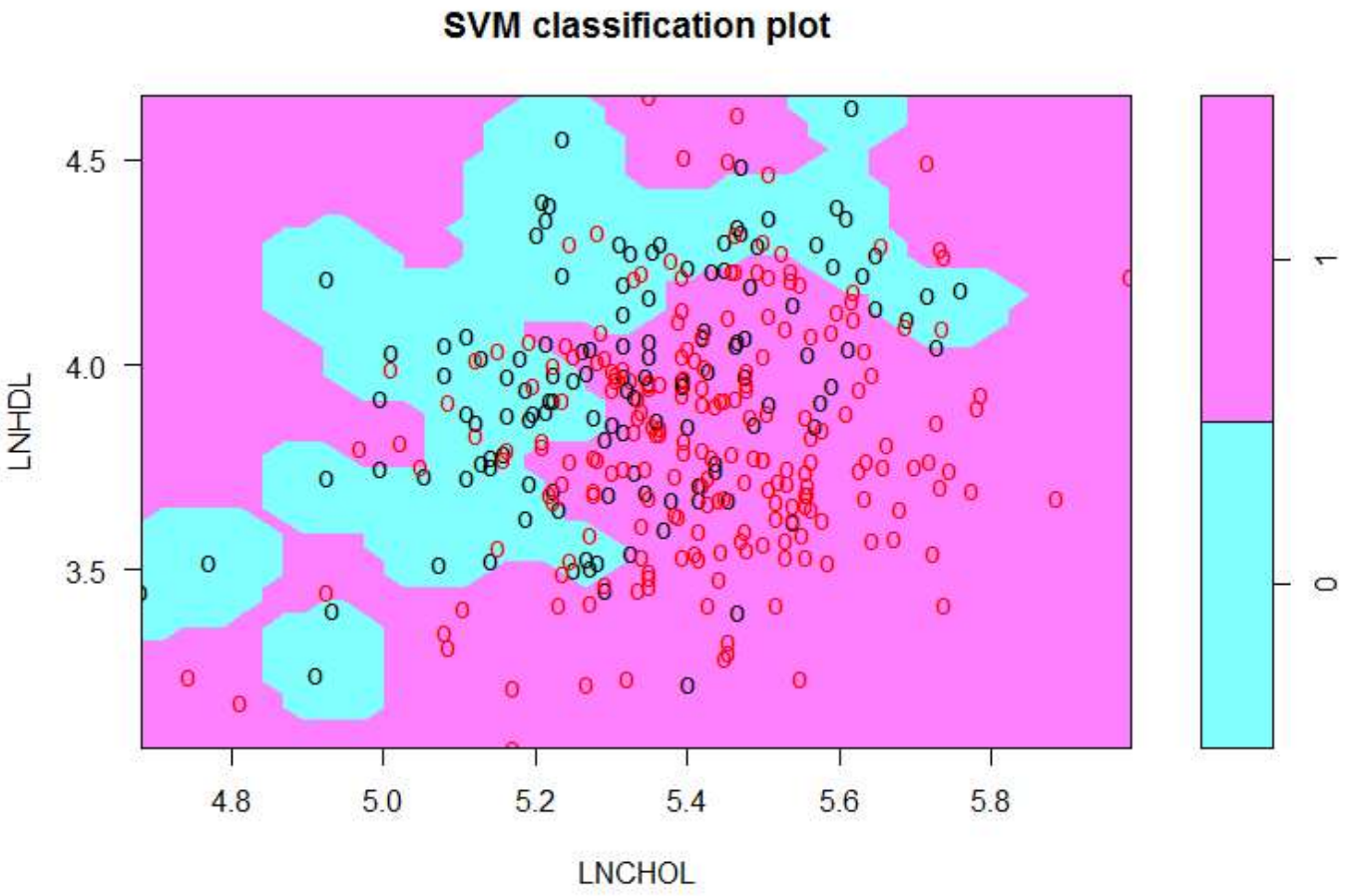
SVM Vapnik, Cortez (1996)

Support Vector Machines

SVM finds best, possibly non-linear boundaries between events and non-events:

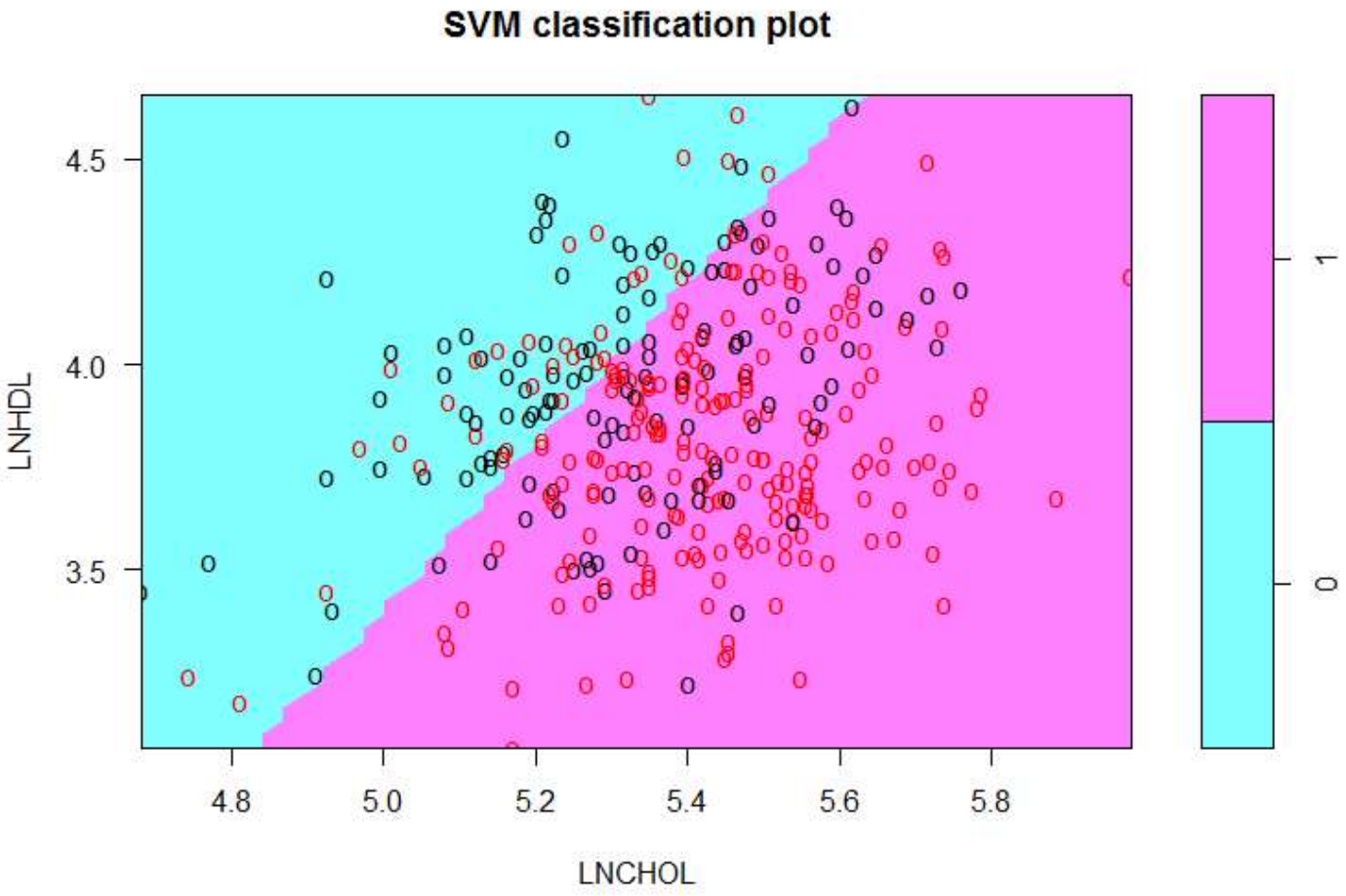
Support Vector Machines

SVM finds best, possibly non-linear boundaries between events and non-events:



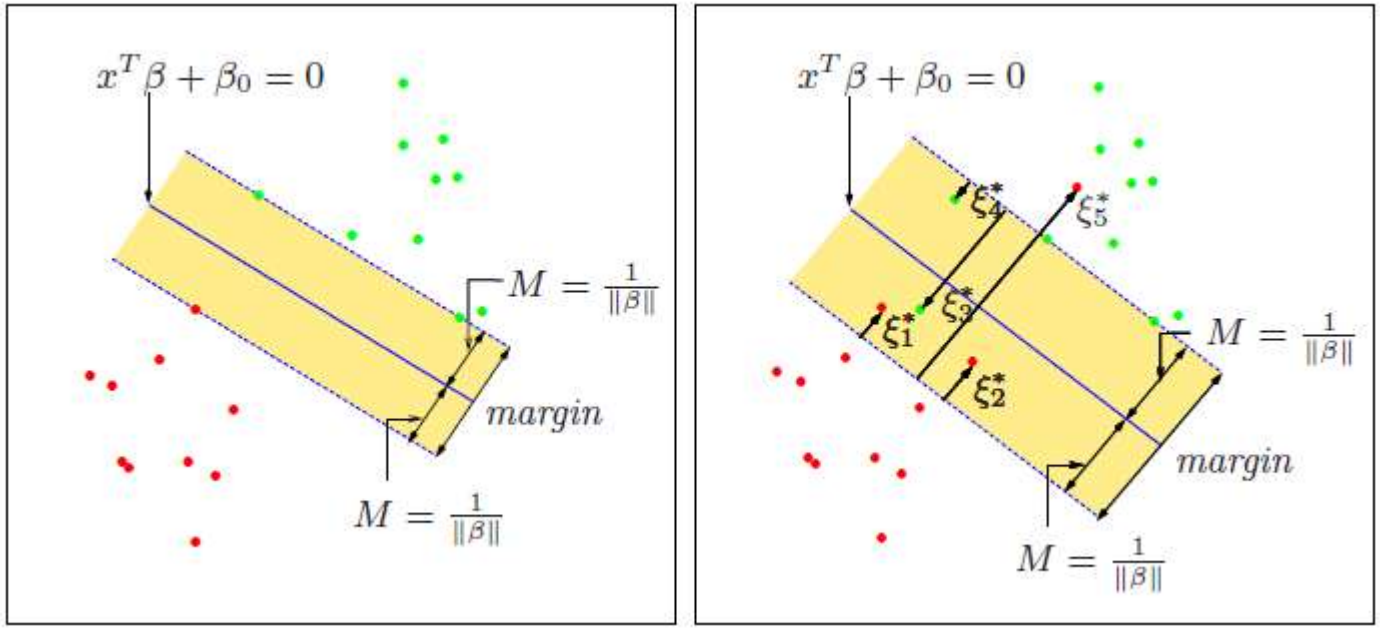
Support Vector Machines

Can control degree of non-linearity:



Support Vector Machines

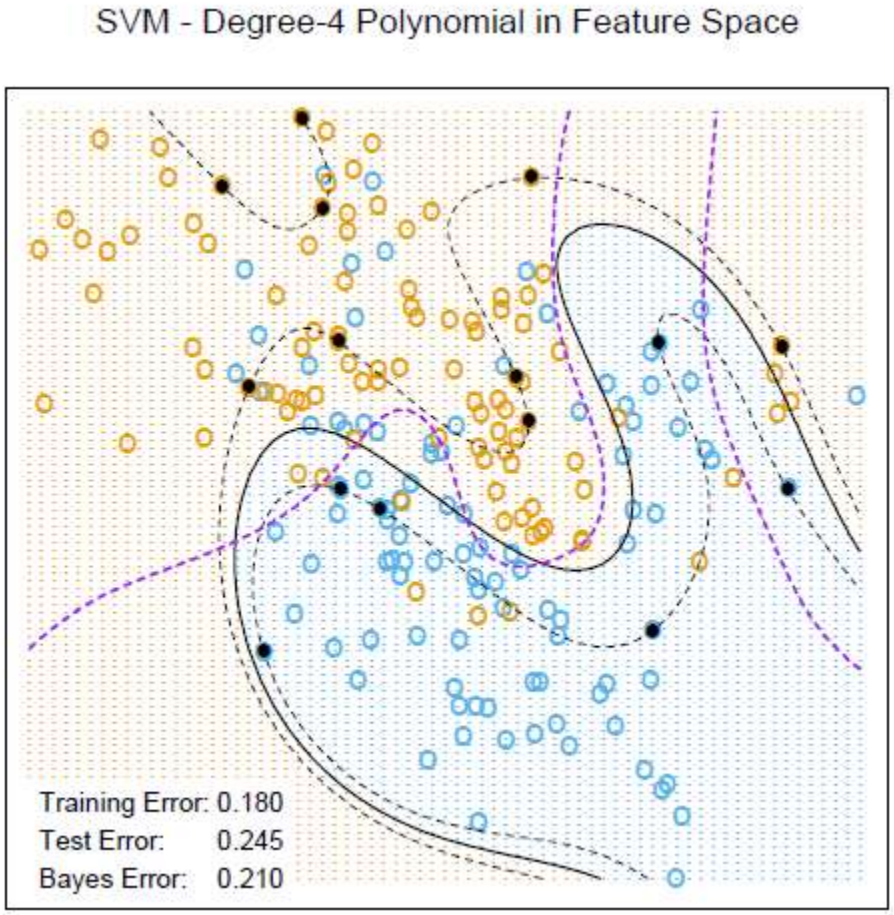
Main Idea:



Hastie et al

Left: separable case. Find the line that separates two groups by the widest margin.
Right: Groups overlap. Some observations are misclassified. Fix amount of allowed misclassification while find the boundary that has the largest margin

Support Vector Machines

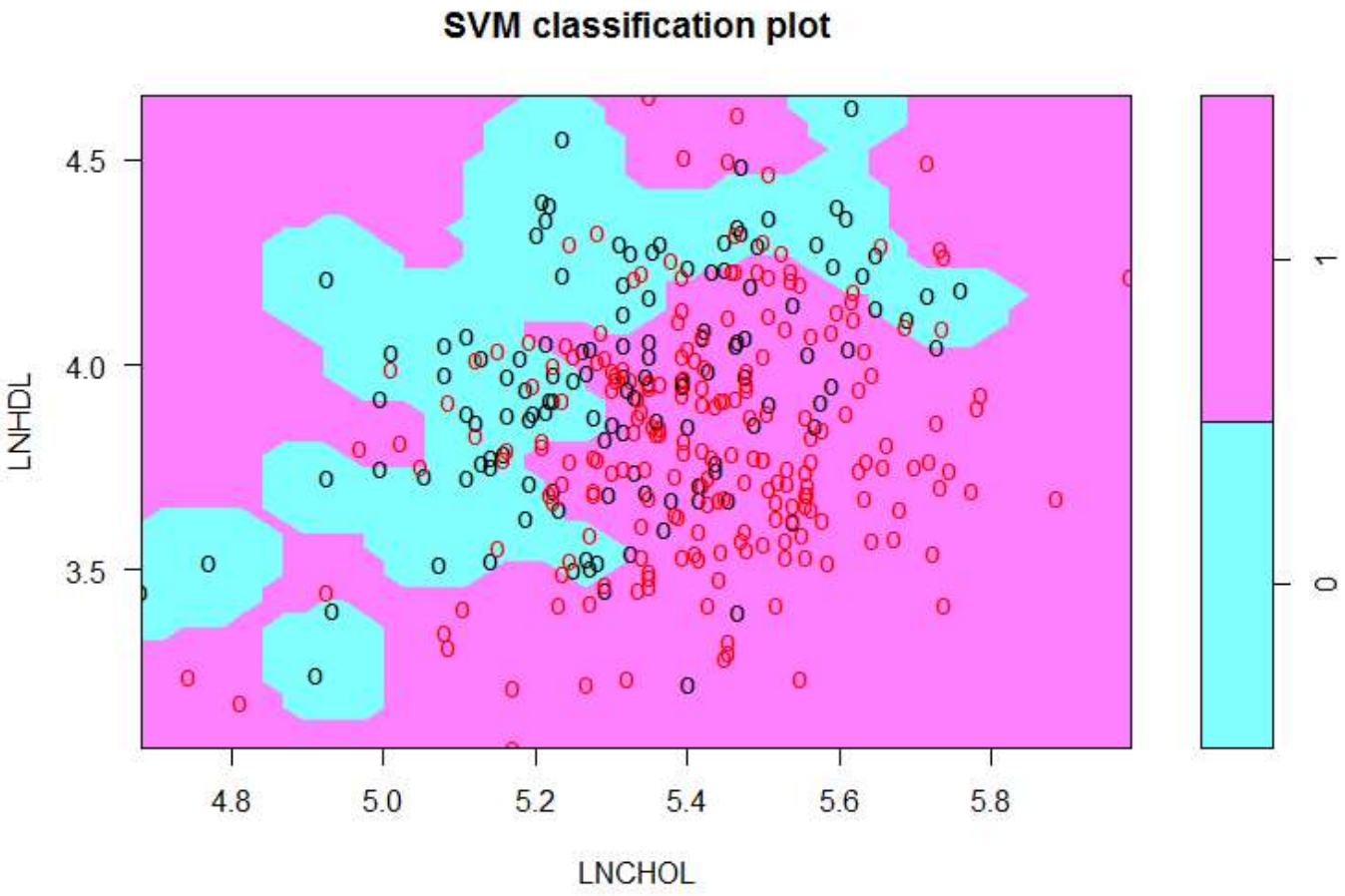


Hastie et al

1. Fix amount of allowed misclassification, called cost (C)
2. Split into two regions with non-linear boundary by maximizing the margin while staying within specified cost C

Support Vector Machines

SVM finds best, possibly non-linear boundaries between events and non-events:



Support Vector Machines

Pros:

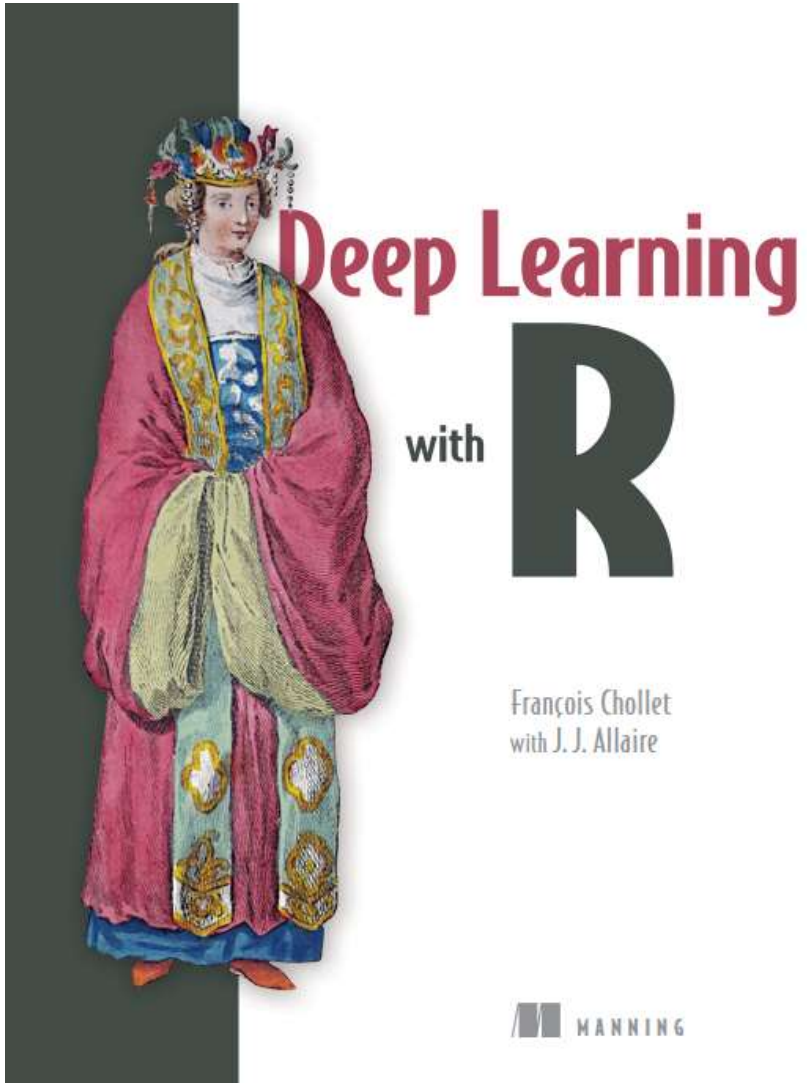
- Accommodates a wide range of boundaries and therefore can separate into two categories many different configuration of classes

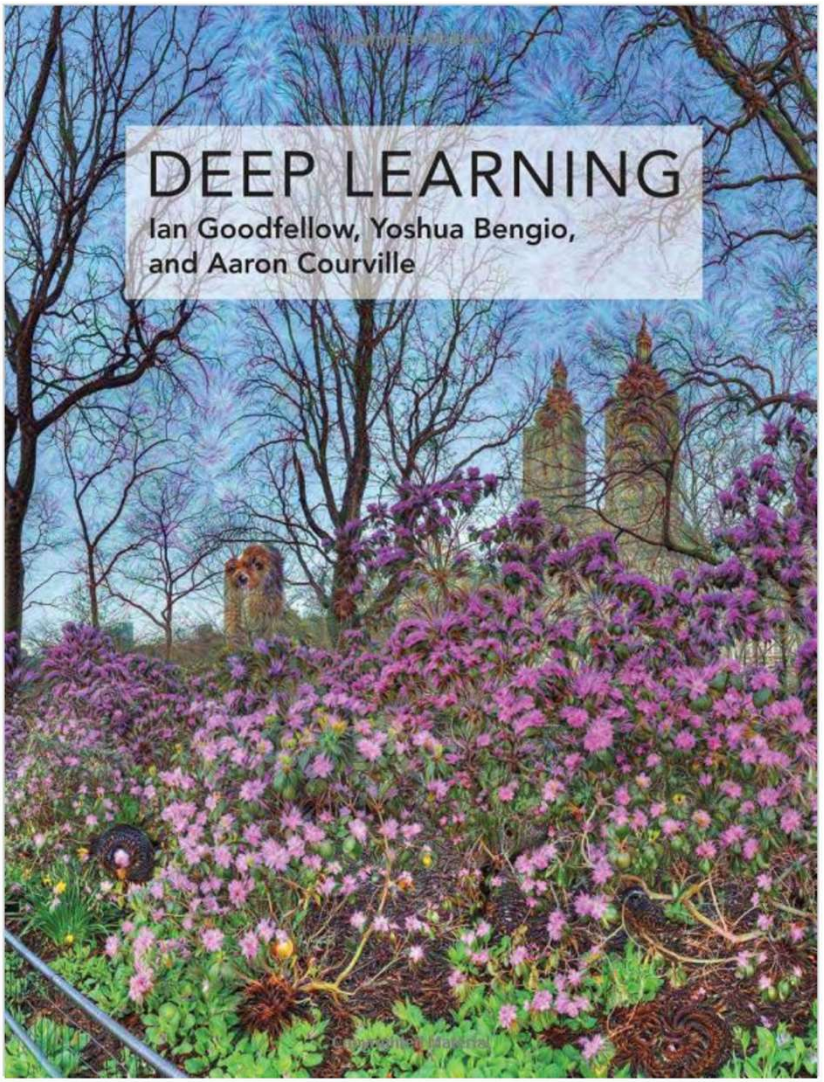
Cons:

- Cannot select variables (yet): if two predictors are important, have to build this knowledge into the model (curse of dimensionality)
- Can be computationally intensive

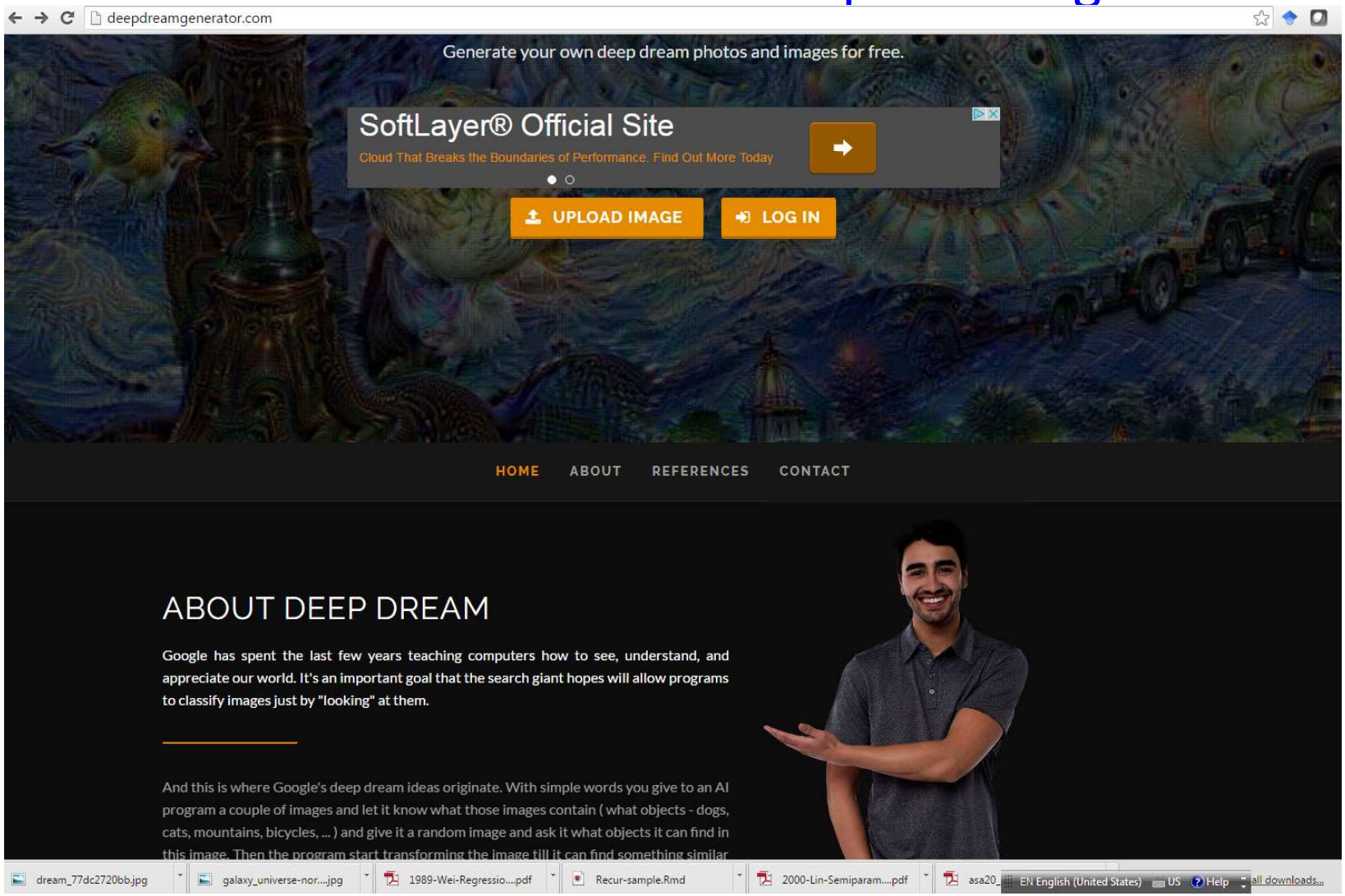
Deep Learning

Deep Learning



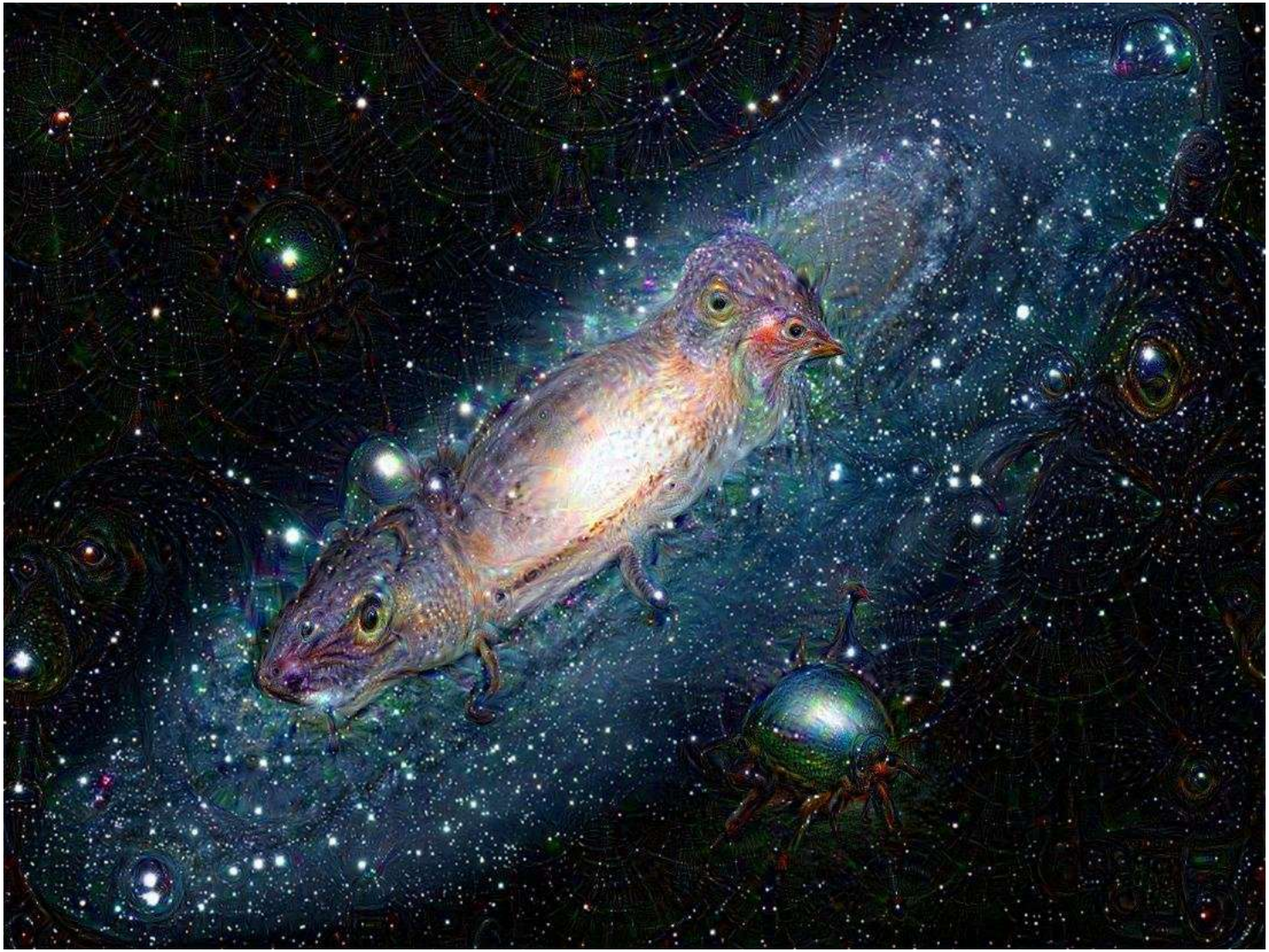


Neural Networks and Deep Learning



http://deepdreamgenerator.com/

http://deepdreamgenerator.com/



Gulshan, Varun, et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." *Jama* 316.22 (2016): 2402-2410.

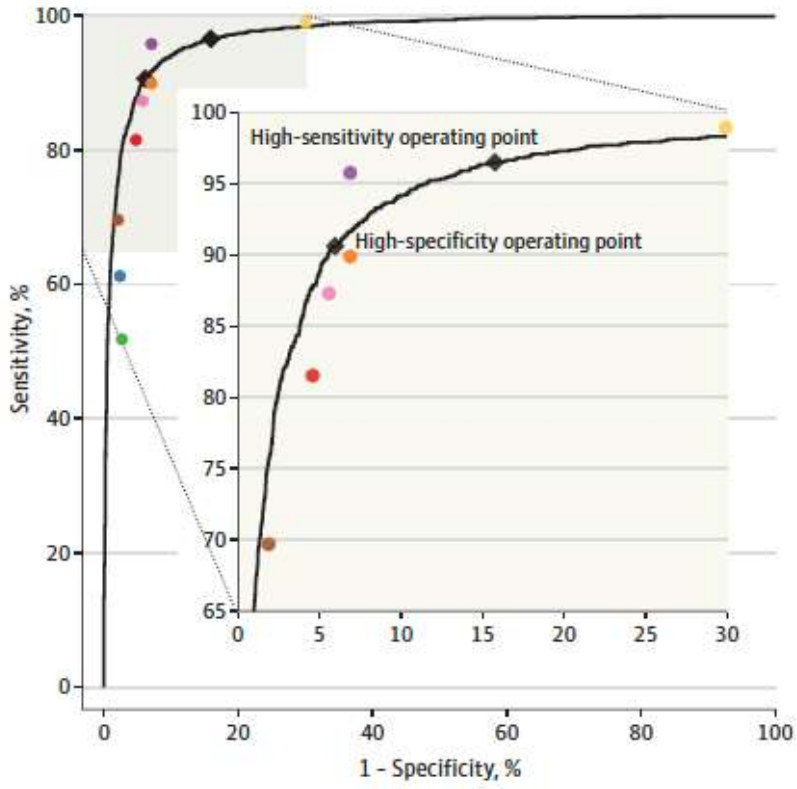
JAMA | Original Investigation | INNOVATIONS IN HEALTH CARE DELIVERY

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs

Varun Gulshan, PhD; Lily Peng, MD, PhD; Marc Coram, PhD; Martin C. Stumpe, PhD; Derek Wu, BS; Arunachalam Narayanaswamy, PhD; Subhashini Venugopalan, MS; Kasumi Widner, MS; Tom Madams, MEng; Jorge Cuadros, OD, PhD; Ramasamy Kim, OD, DNB; Rajiv Raman, MS, DNB; Philip C. Nelson, BS; Jessica L. Mega, MD, MPH; Dale R. Webster, PhD

Performance of the Neural Network in the external dataset

Figure 3. Validation Set Performance for All-Cause Referable Diabetic Retinopathy in the EyePACS-1 Data Set (9946 Images)



Black curve – CNN Algorithm
Colored dots – ophtalmologists

AUC 97.4% (95% CI, 97.1%-97.8%).

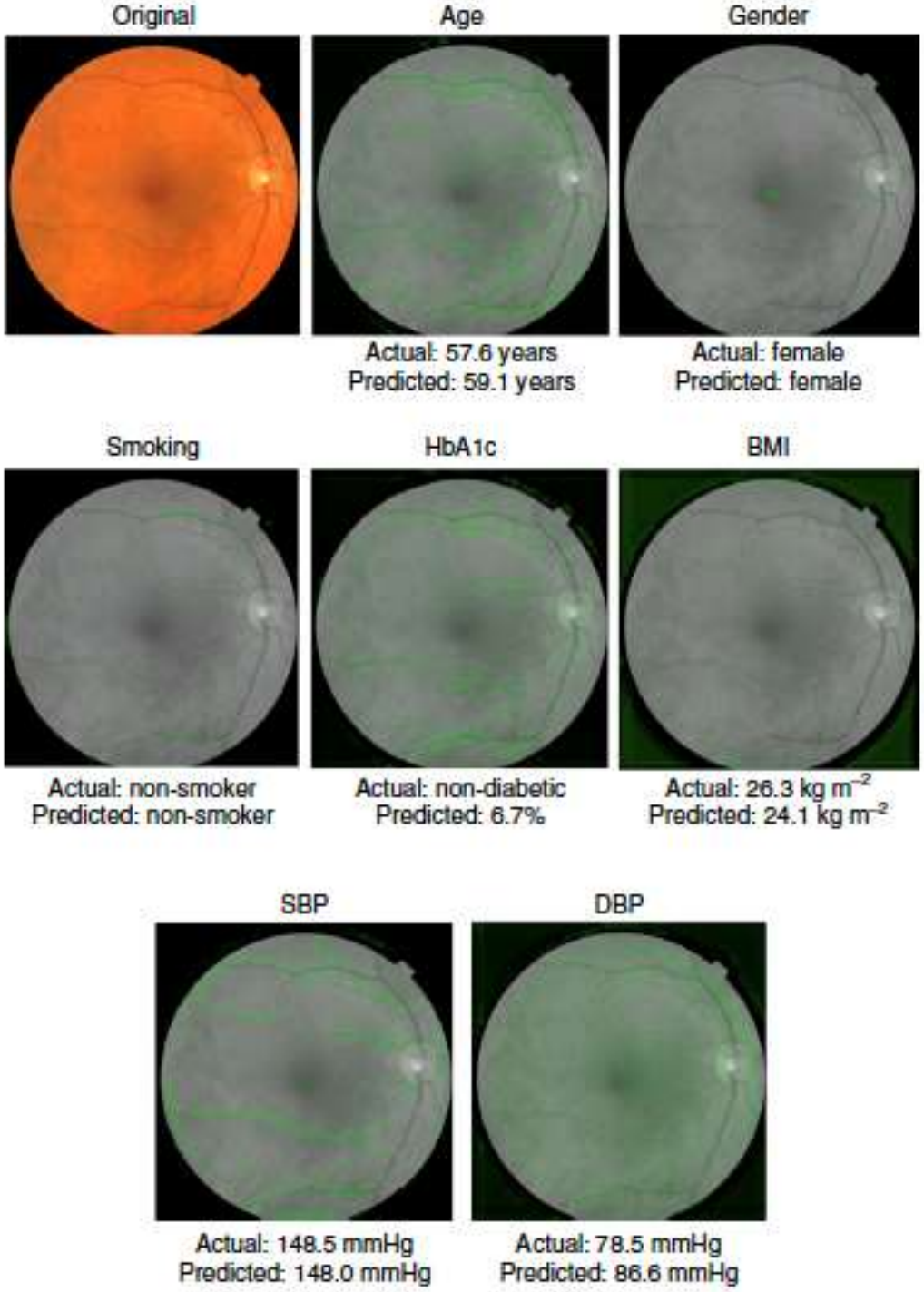
Poplin, Ryan, et al. "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning." *Nature Biomedical Engineering* 2.3 (2018): 158.

ARTICLES
https://doi.org/10.1038/s41551-018-0195-0

nature biomedical engineering

Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning

Ryan Poplin[1,4], Avinash V. Varadarajan[1,4], Katy Blumer[1], Yun Liu[1], Michael V. McConnell[2,3], Greg S. Corrado[1], Lily Peng[1,4*] and Dale R. Webster[1,4]



Neural Network predicted:

Age – Mean abs error within 3.67 years

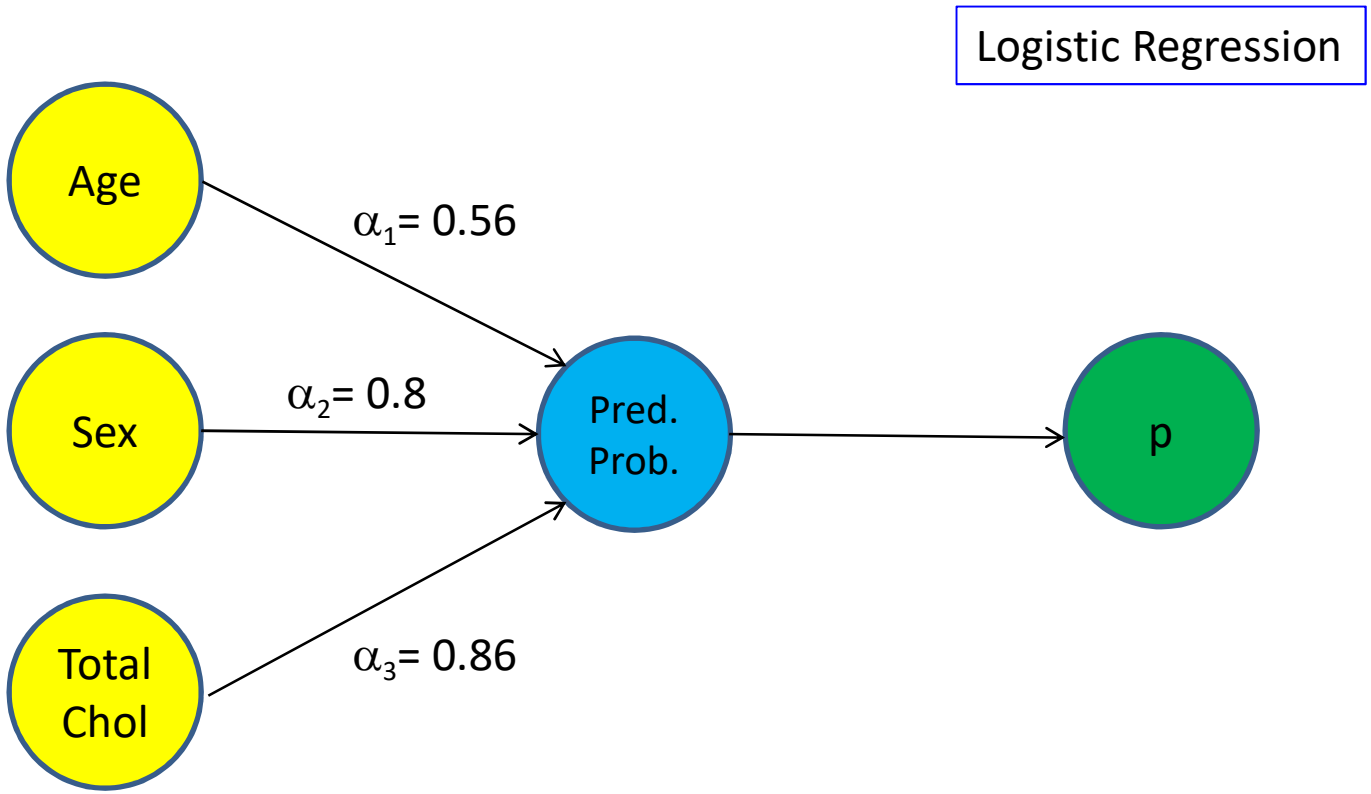
Sex – AUC=0.97

Smoking status – AUC=0.72

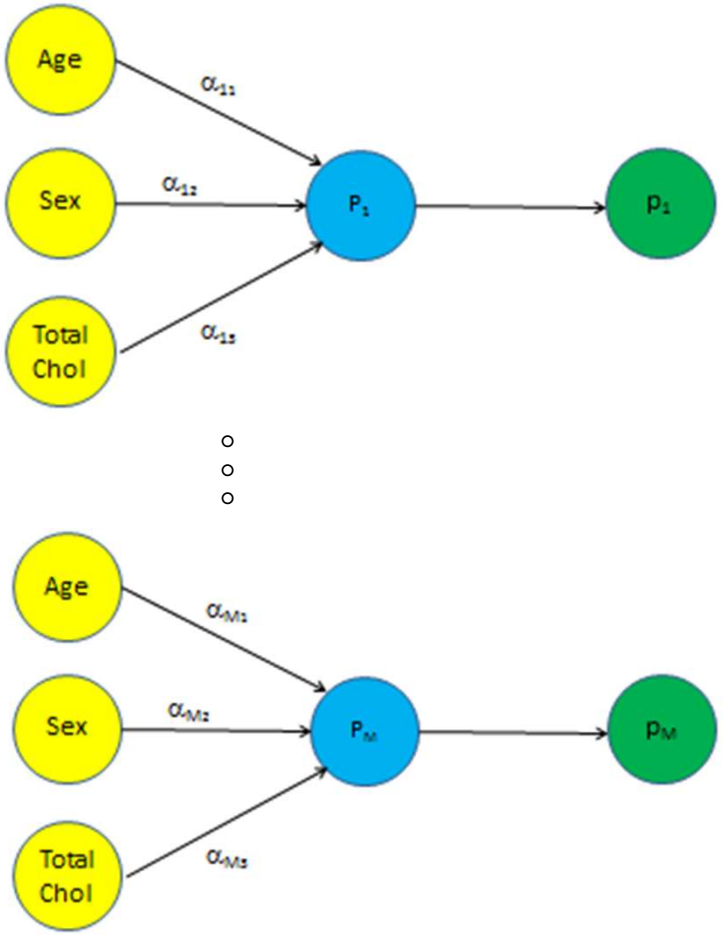
SBP – Mean absolute error of 11.2 mmHg

Poplin, Ryan, et al. "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning." *Nature Biomedical Engineering* 2.3 (2018): 158.

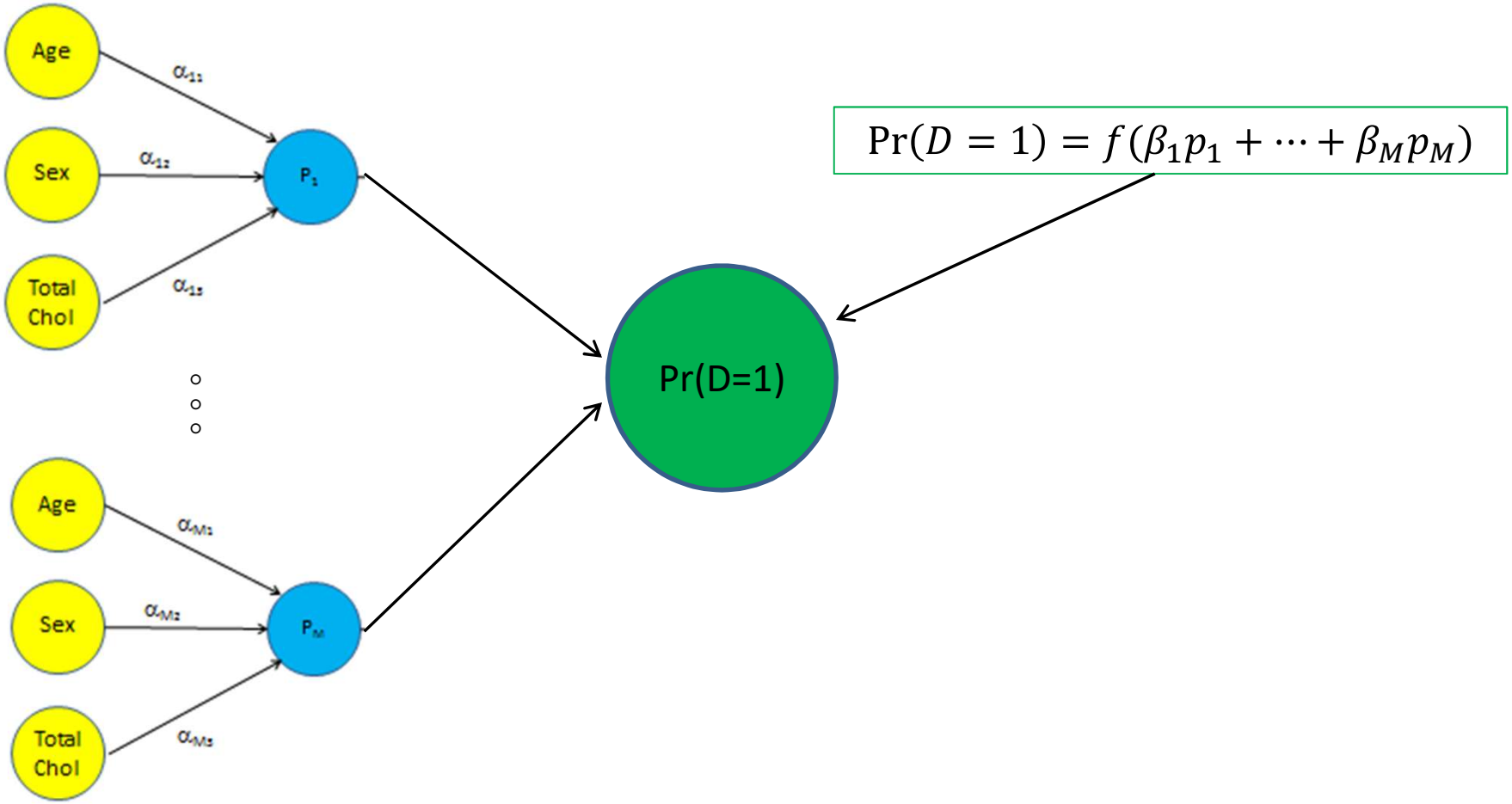
Logistic Regression



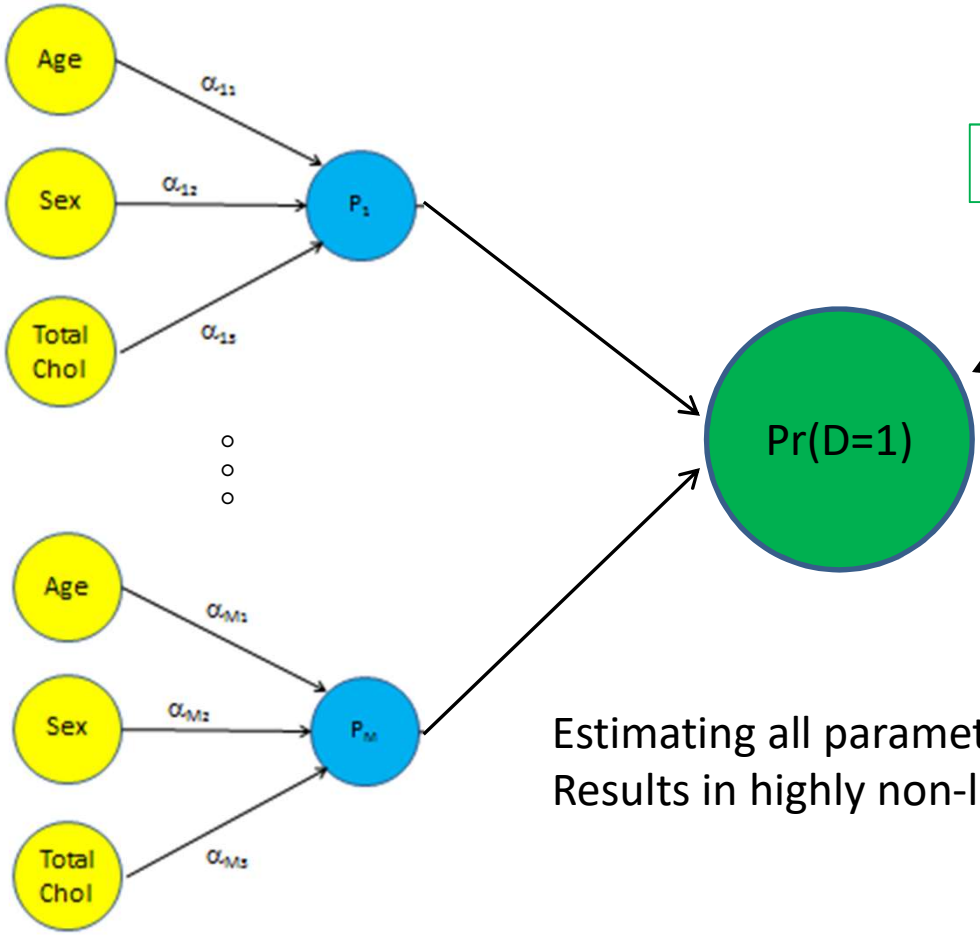
Neural Networks



Neural Networks



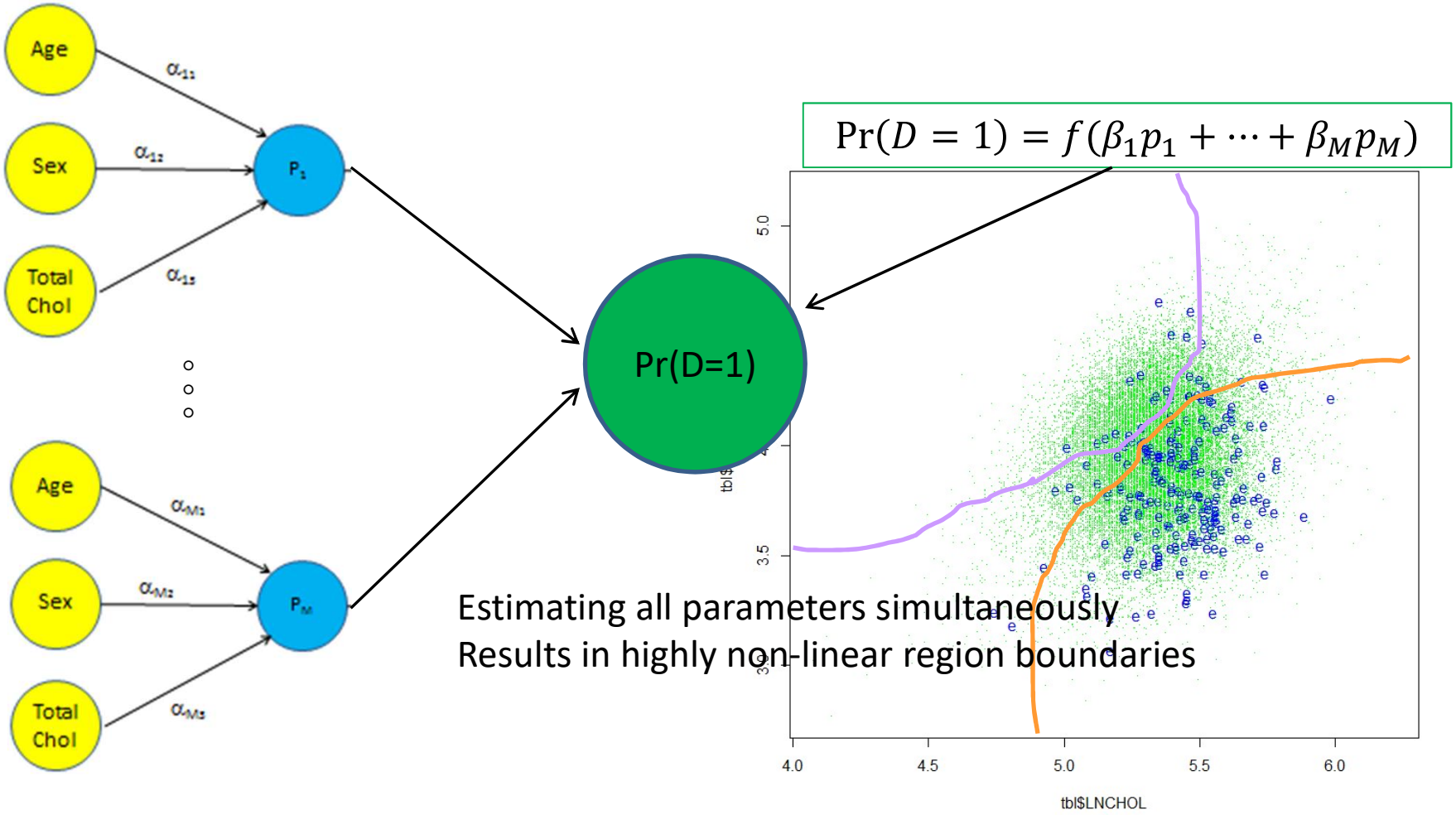
Neural Networks



$$\Pr(D = 1) = f(\beta_1 p_1 + \cdots + \beta_M p_M)$$

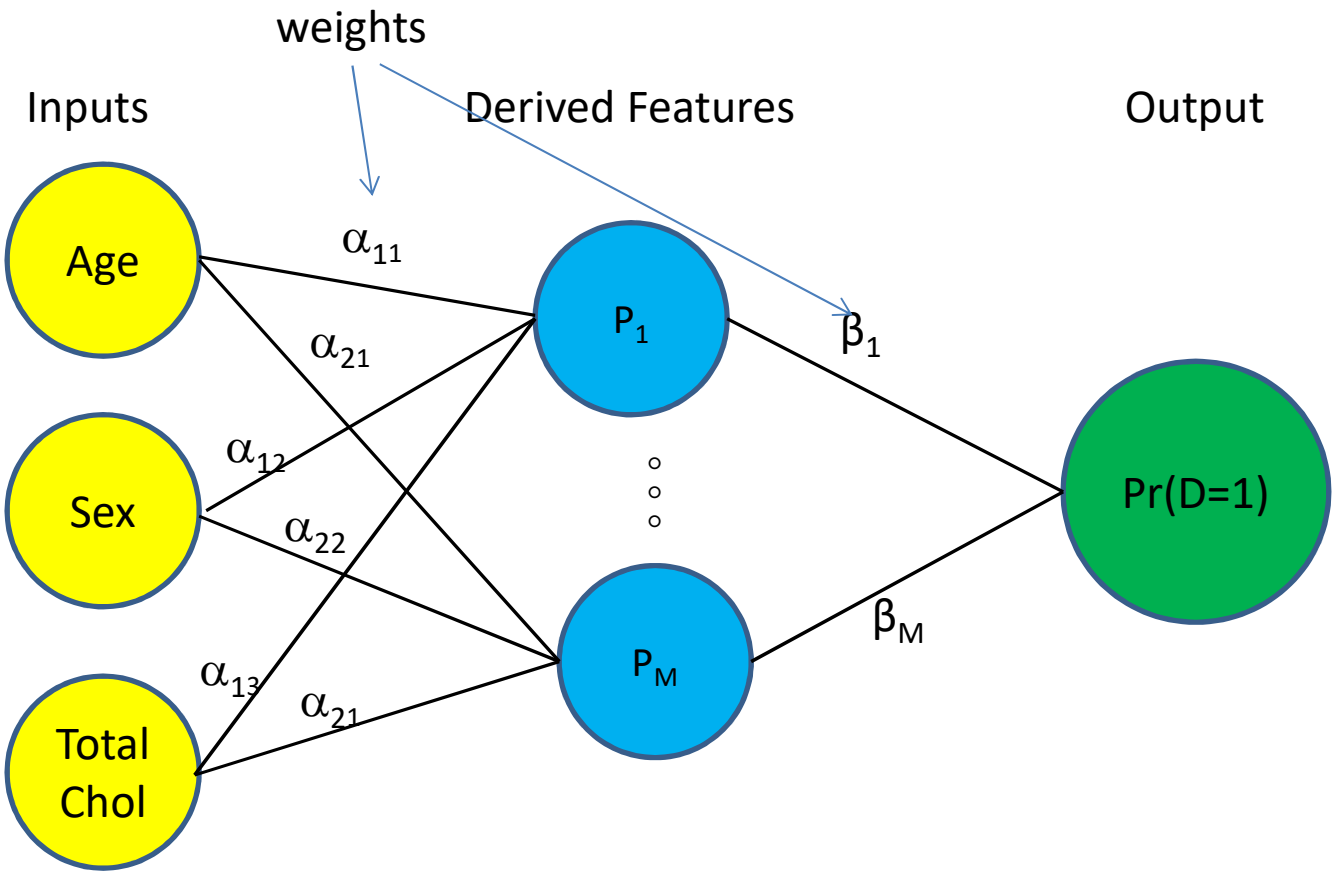
Estimating all parameters simultaneously
Results in highly non-linear region boundaries

Neural Networks

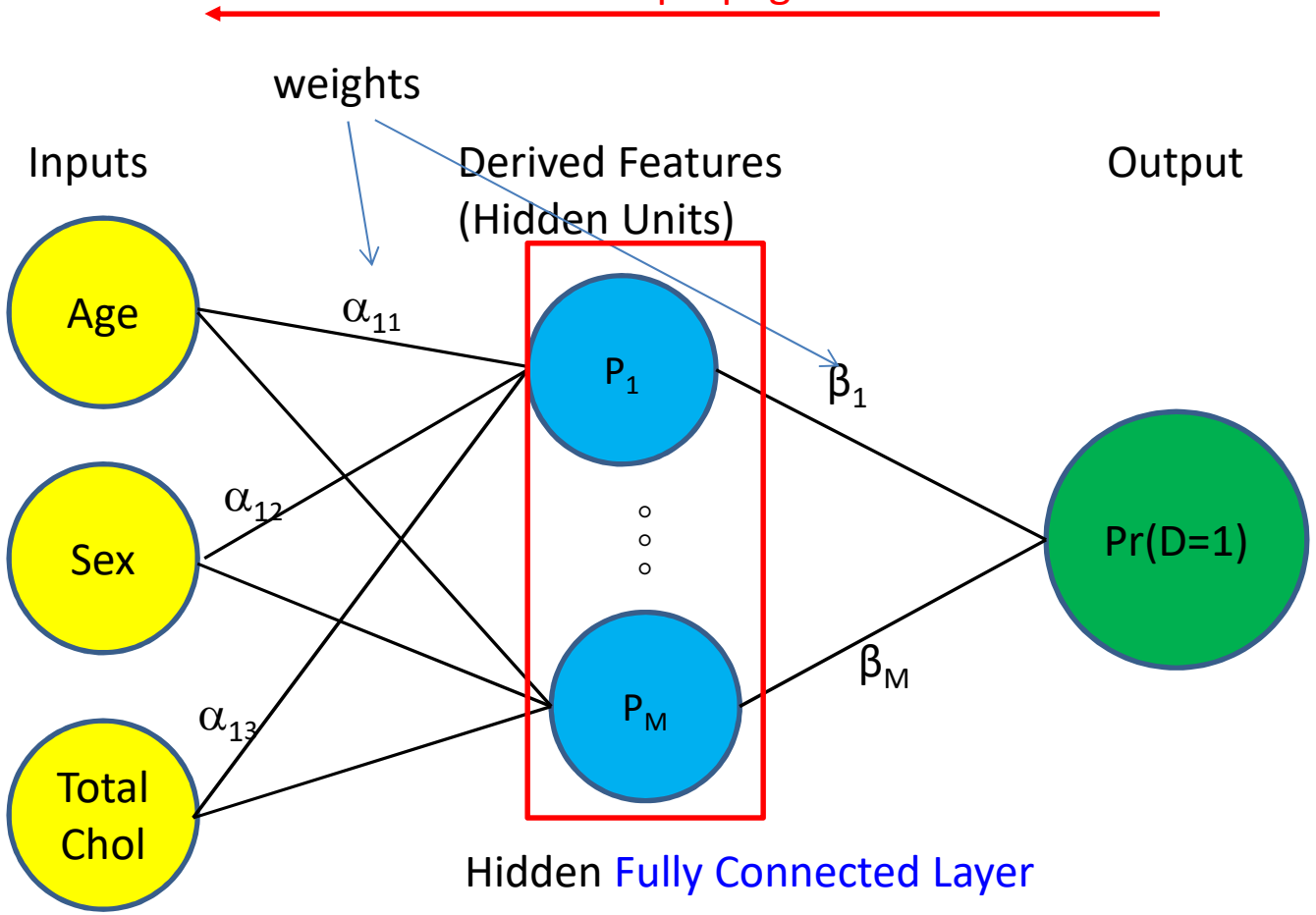


Estimating all parameters simultaneously
Results in highly non-linear region boundaries

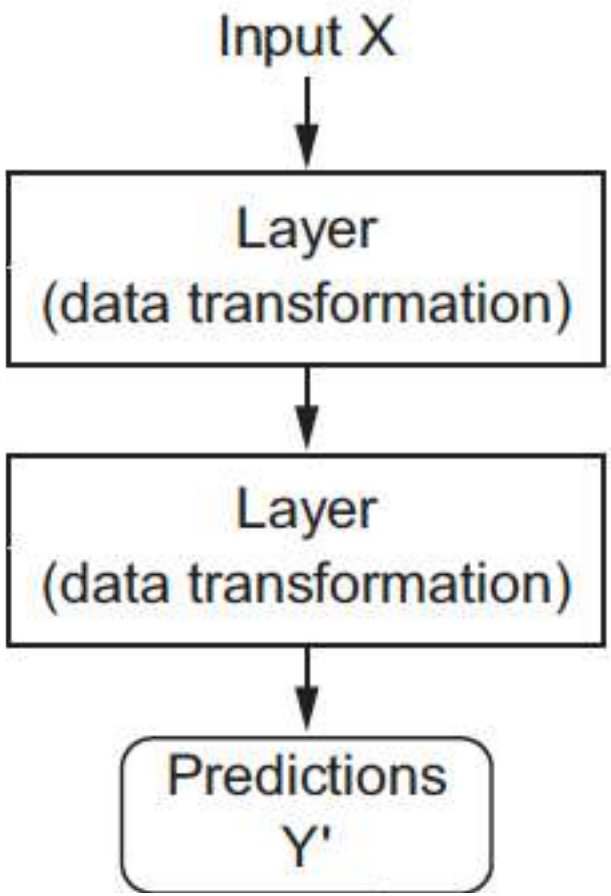
Neural Networks

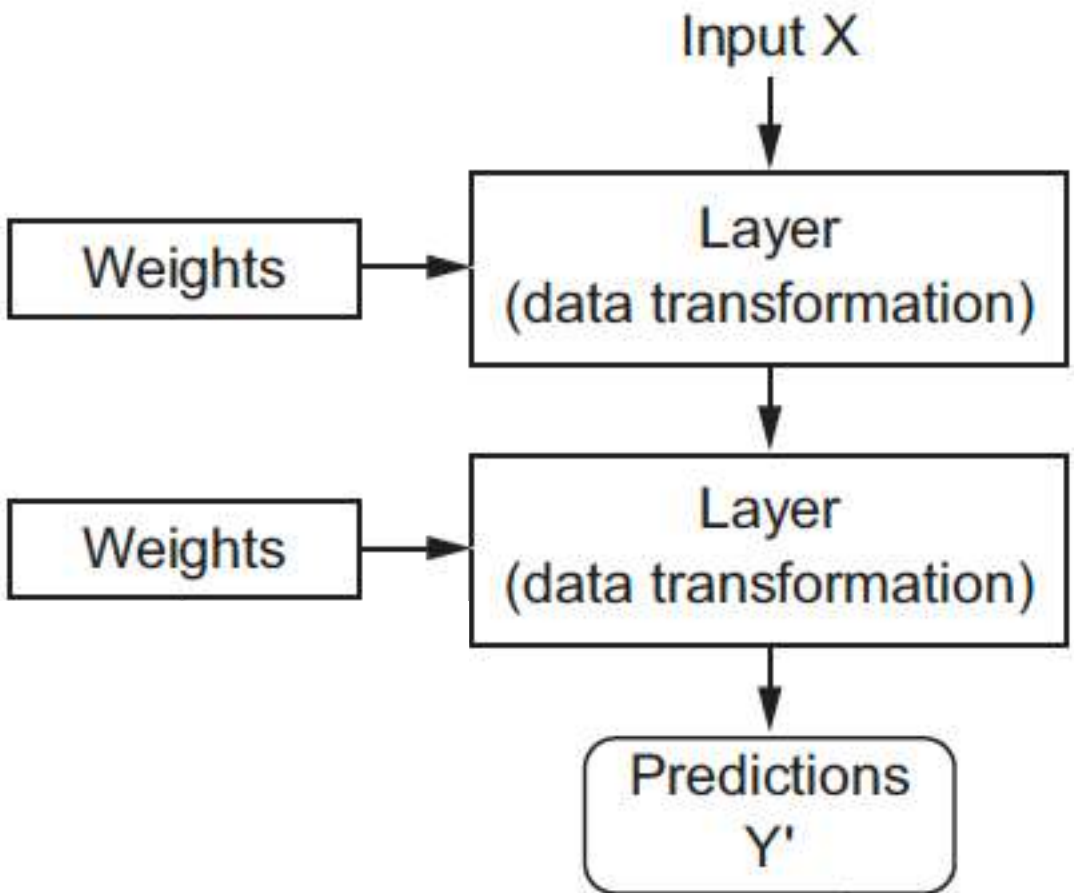


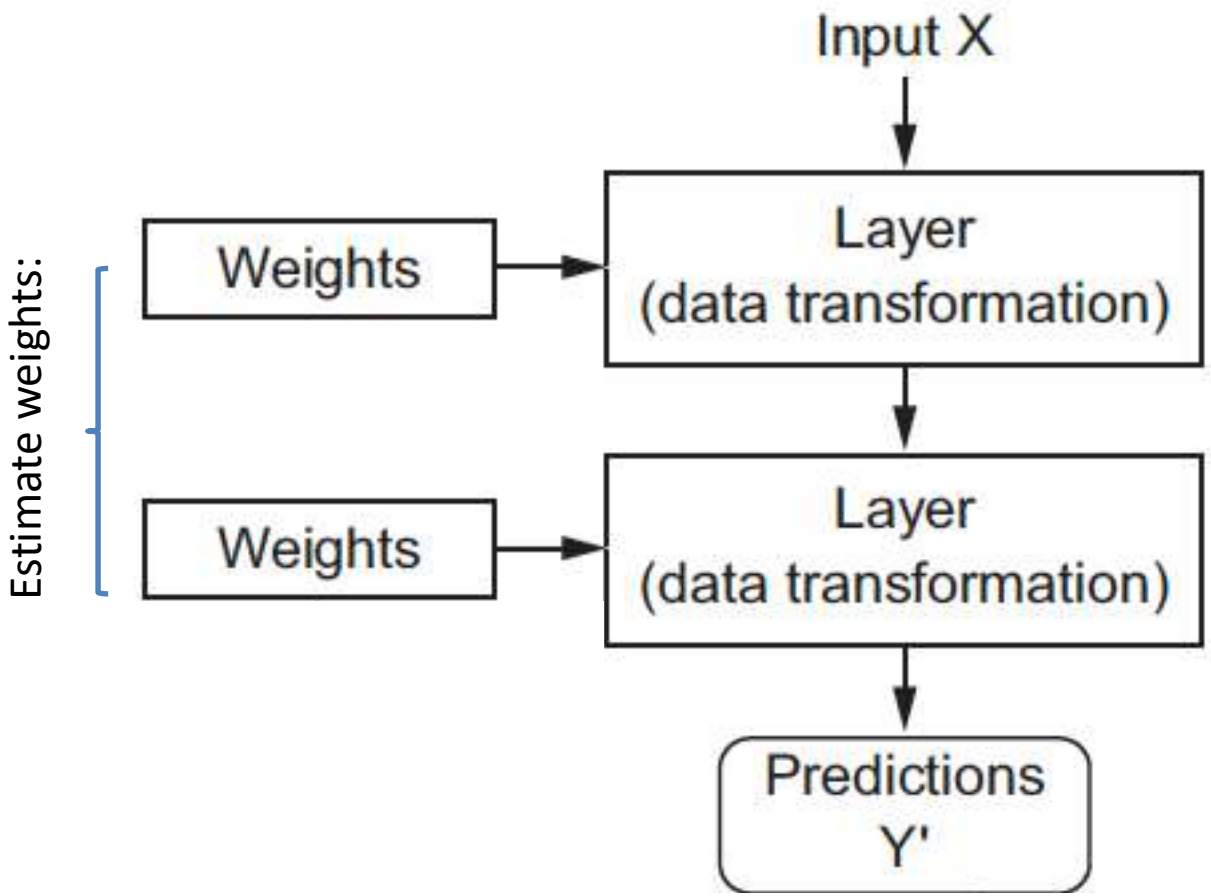
Short review of predictive models: Neural Networks

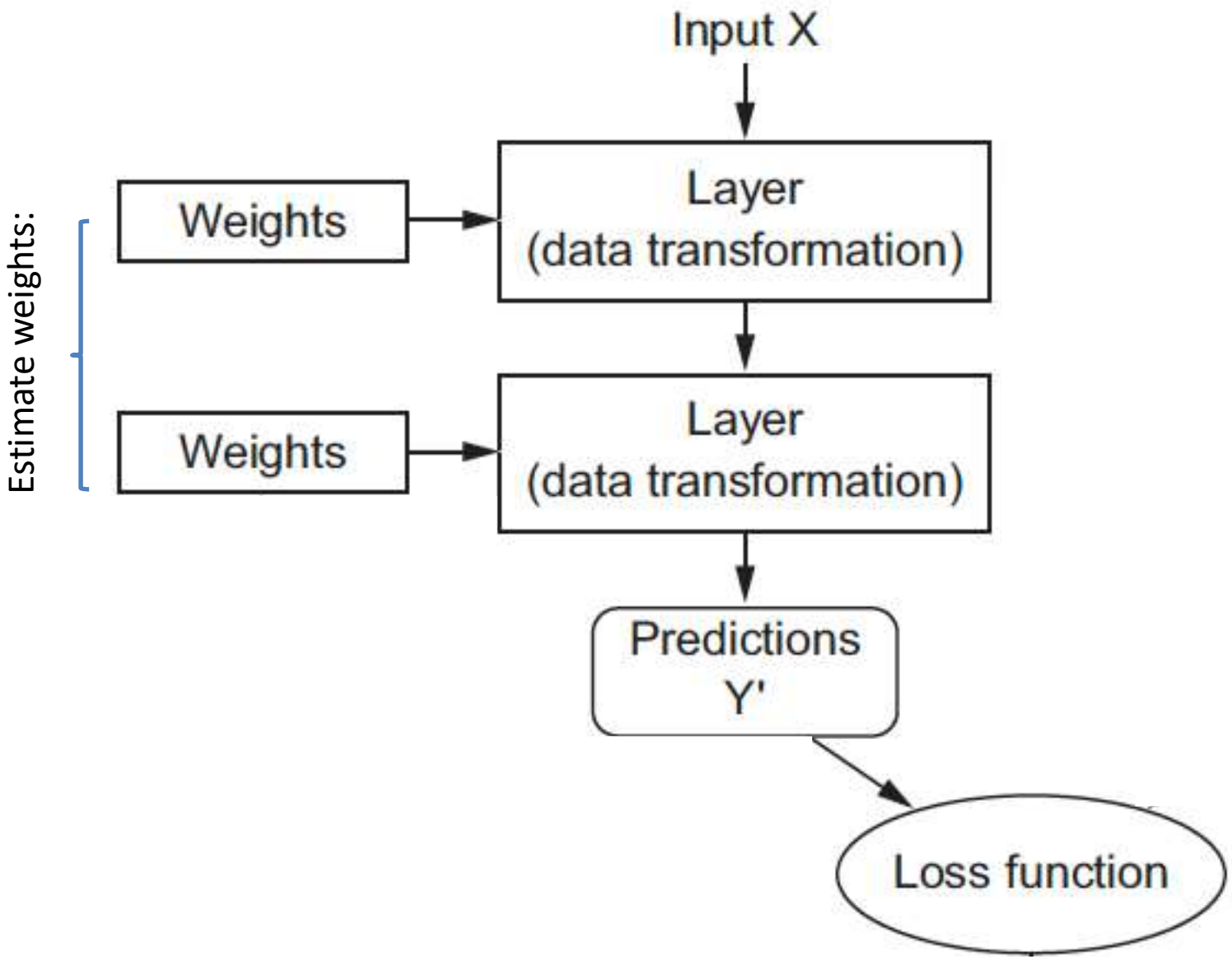


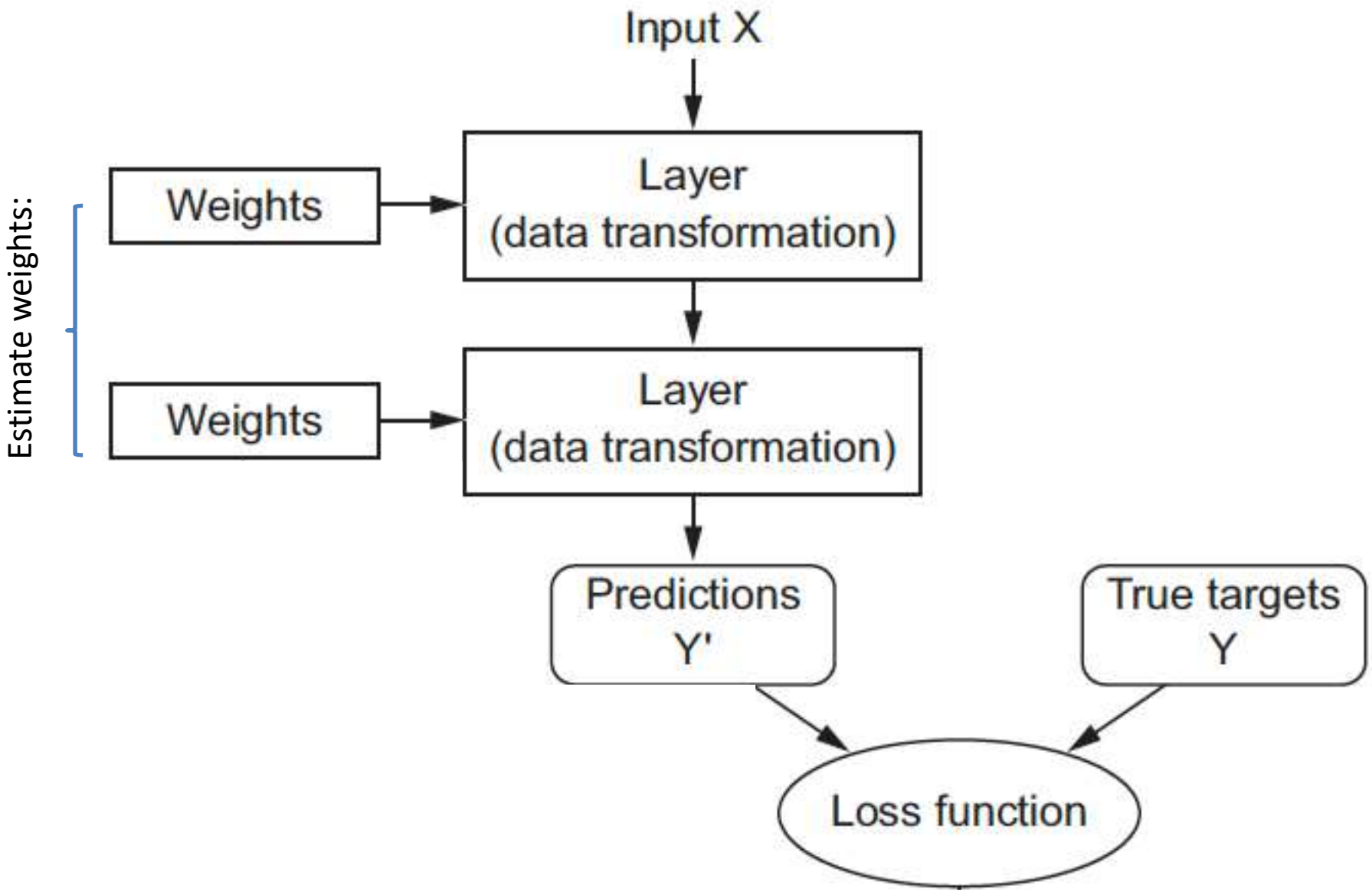
Might have more than one hidden layer

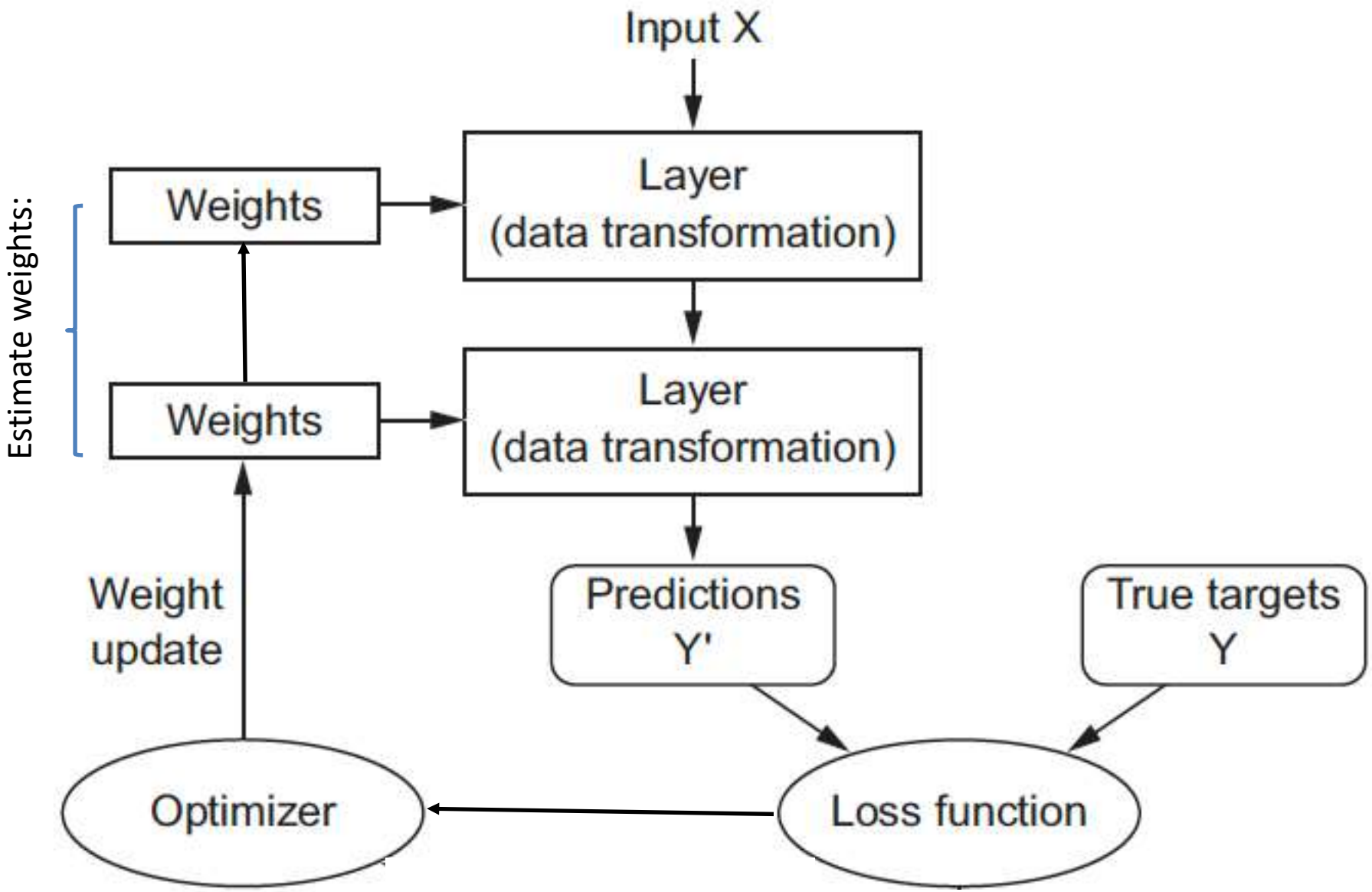

Input X
Layer
(data transformation)
Layer
(data transformation)
Predictions
Y'

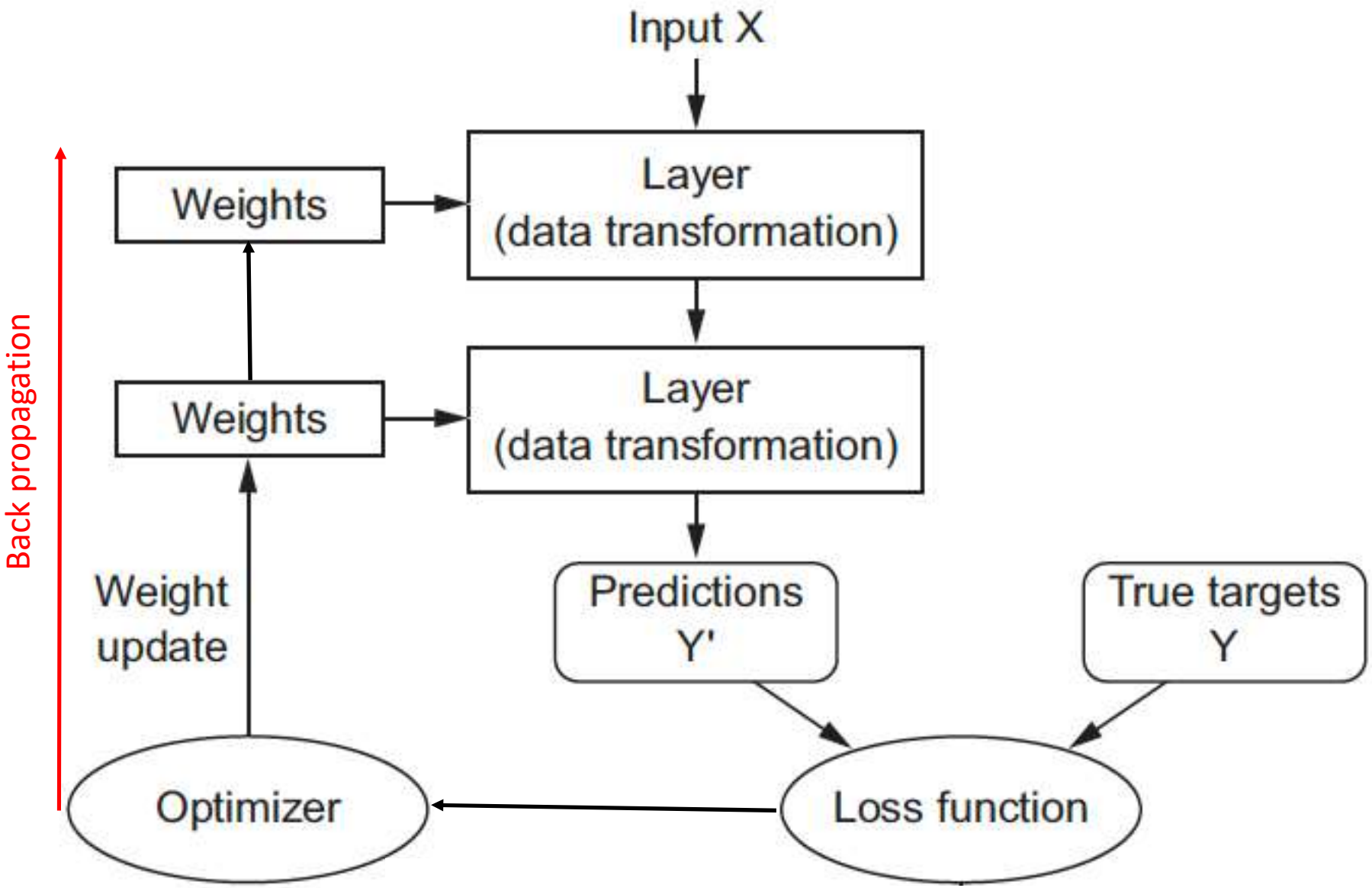

Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Predictions
Y'

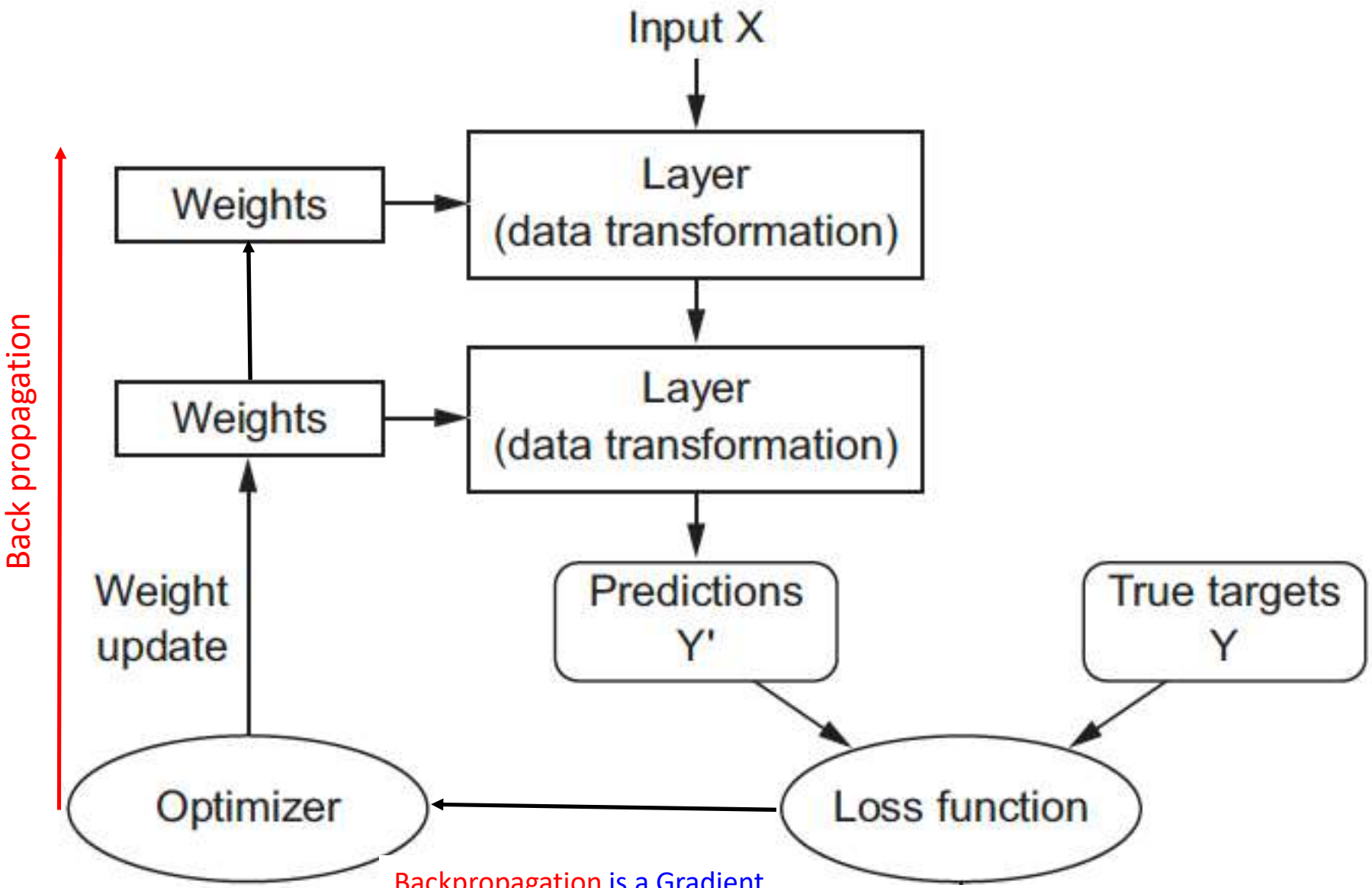

Input X
Layer
(data transformation)
Weights
Layer
(data transformation)
Weights
Predictions
Y'
Estimate weights:


Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Predictions
Y'
Loss function
Estimate weights:


Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Predictions
Y'
True targets
Y
Loss function
Estimate weights:

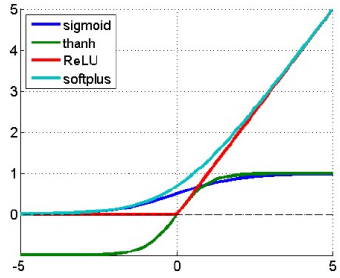

Input X
Weights
Layer
(data transformation)
Estimate weights:
Weights
Layer
(data transformation)
Weight
update
Predictions
Y'
True targets
Y
Optimizer
Loss function


Input X
Weights
Layer
(data transformation)
Weights
Layer
(data transformation)
Predictions
Y'
True targets
Y
Weight
update
Optimizer
Loss function
Back propagation



Backpropagation is a Gradient Descent Optimization by Stochastic Gradient Descent or RMSprop or other methods

http://ruder.io/optimizing-gradient-descent/



Input X

Weights

Layer
(data transformation)

Activation Function

https://dashee87.github.io/data science/deep learning/visualising-activation-functions-in-neural-networks/

Weights

Layer
(data transformation)

Activation Function

Back propagation

Weight update

Predictions
Y'

True targets
Y

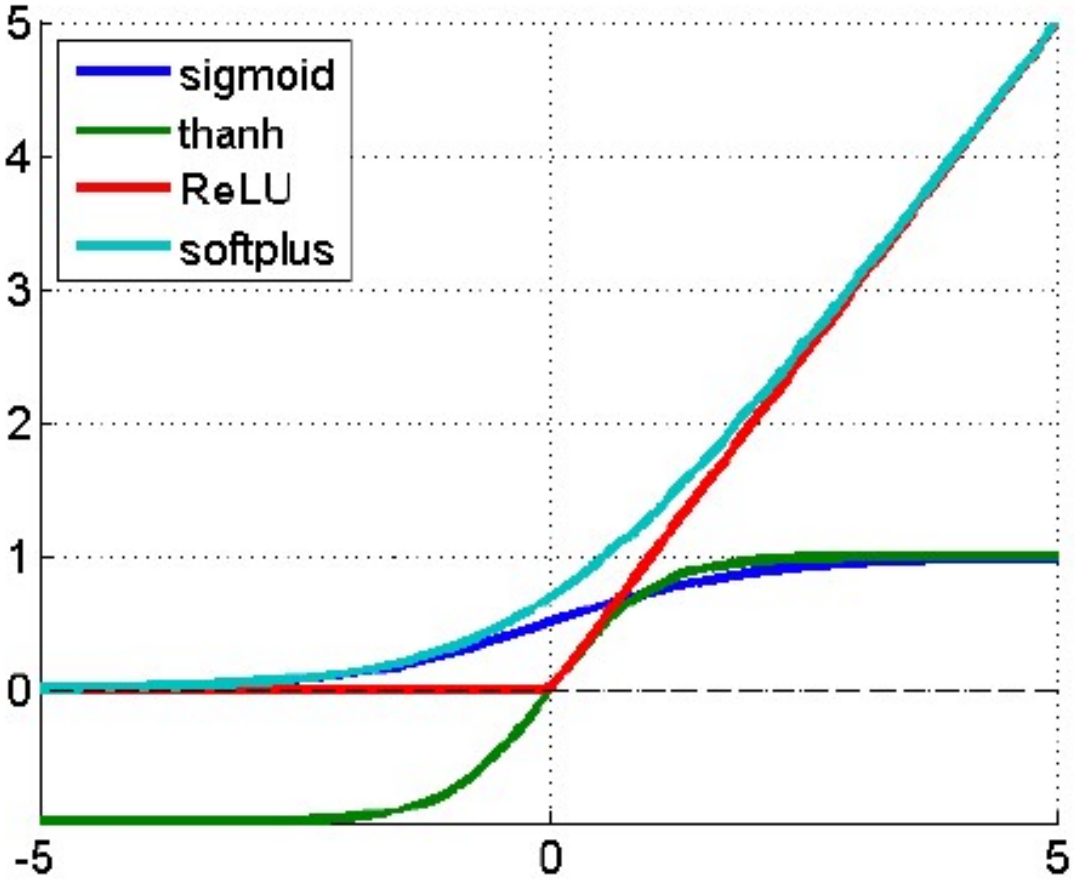
Optimizer

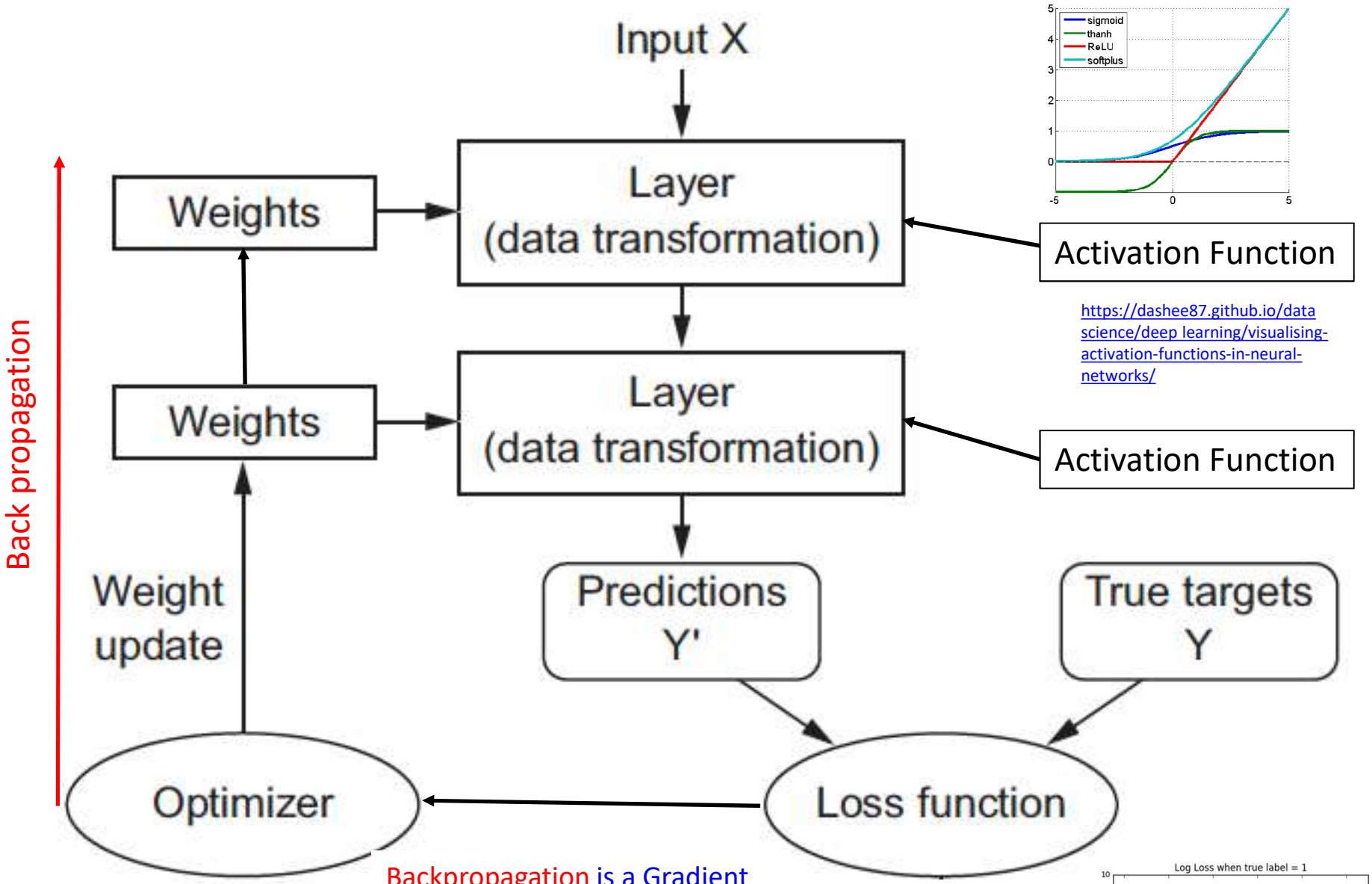
Loss function

Backpropagation is a Gradient Descent Optimization by Stochastic Gradient Descent or RMSprop or other methods

http://ruder.io/optimizing-gradient-descent/

Activation Function – adding non-linearity to the model

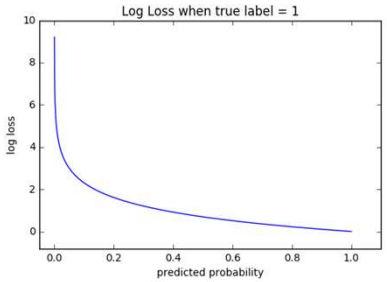




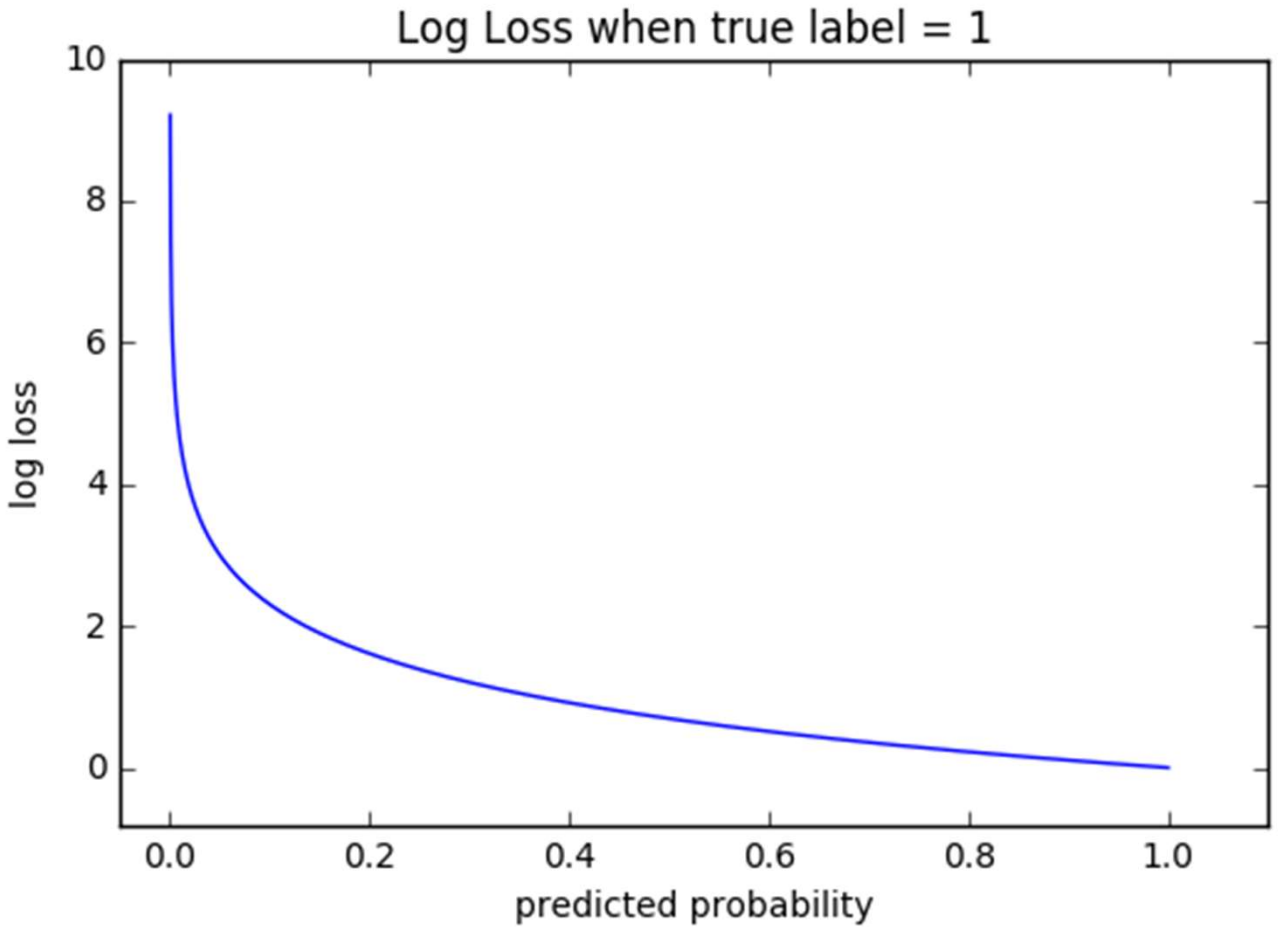
https://dashee87.github.io/data science/deep learning/visualising-activation-functions-in-neural-networks/

Backpropagation is a Gradient Descent Optimization by Stochastic Gradient Descent or RMSprop or other methods

http://ruder.io/optimizing-gradient-descent/



Cross Entropy Loss



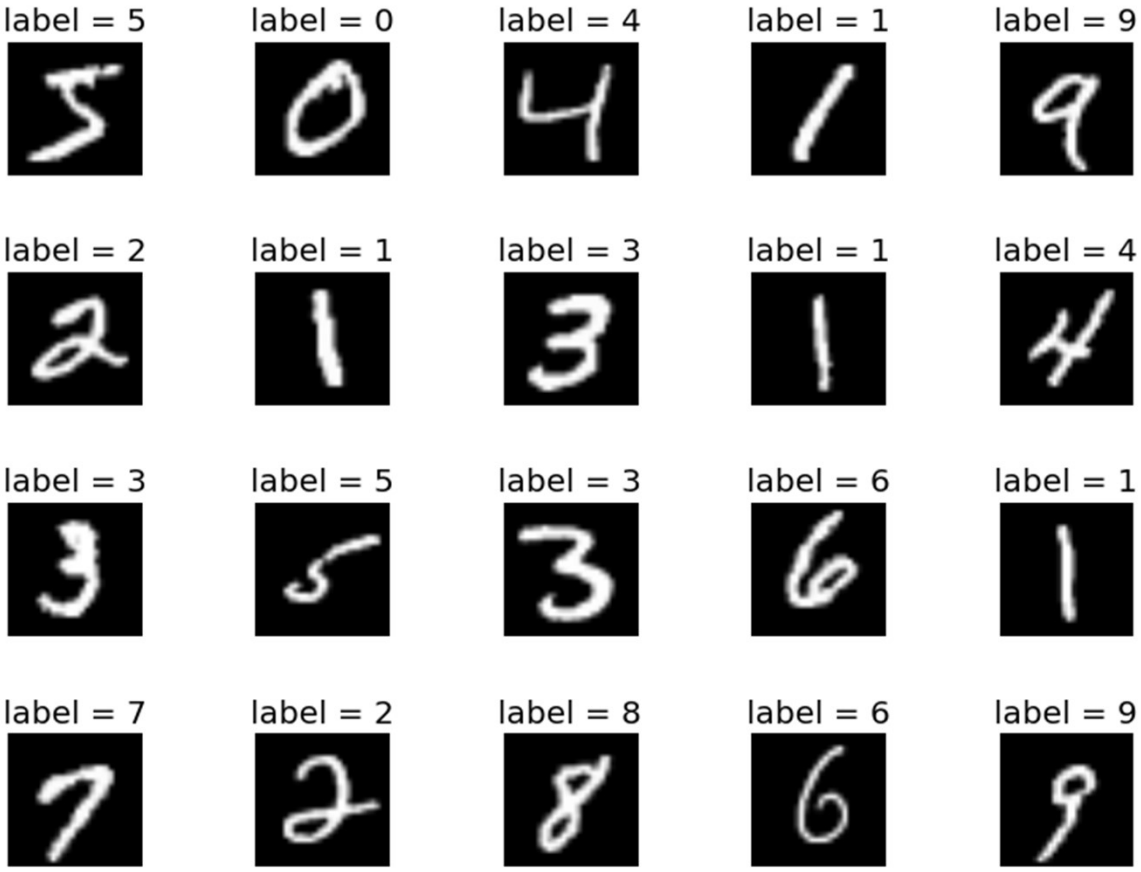
Neural Networks

How the weights are estimated?

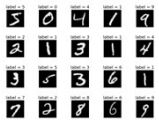
- Start with some set of weights as initial starting values
- Calculate predicted probabilities for one observation in your data and identify incorrect classifications.
- Do "Back Propagation": change more those weights which are responsible for incorrect classification of a given observation
- Repeat for other observations until overall error is minimized
- One pass through data is called an "epoch"
- Duration of the training is measured in epochs
- Also use weight decay: allow algorithm dynamically adjust the complexity of the network by setting some weights to zero.

Backpropagation is a Gradient Descent Optimization by Stochastic Gradient Descent or RMSprop or other methods

MNIST dataset used for image recognition



The MNIST database contains 60,000 training images and 10,000 testing images
MNIST is Modified National Institute of Standards and Technology database



For 1st batch

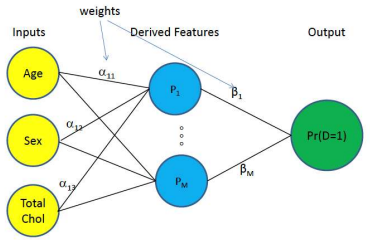
use random weights

Batch 1…1000

label = 5

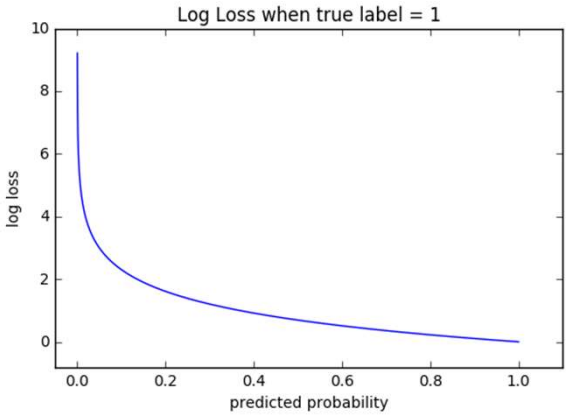
Training Instance 1…6

Run through the Network, Predict the Label, Calculate the Loss



Update the weights

Use Loss to calculate weights adjustment by Backpropagation



Calculate Loss

Epoch 1…5

For 1st batch in the 1st epoch use random weights

Batch 1…1000

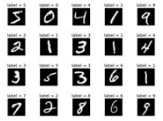
Training Instance 1…6

Run through the Network, Predict the Label, Calculate the Loss

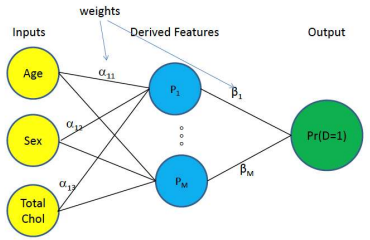
Update the weights

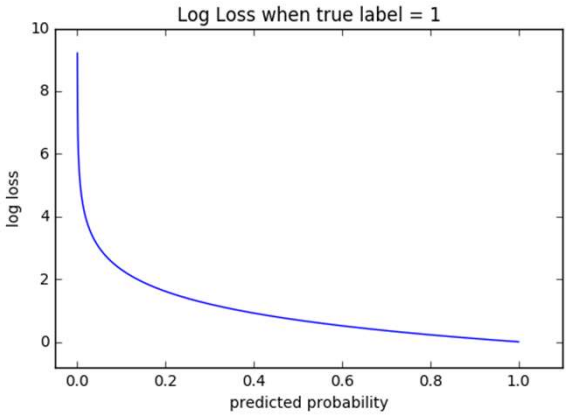
Use Loss to calculate weights adjustment by Backpropagation

Calculate Loss



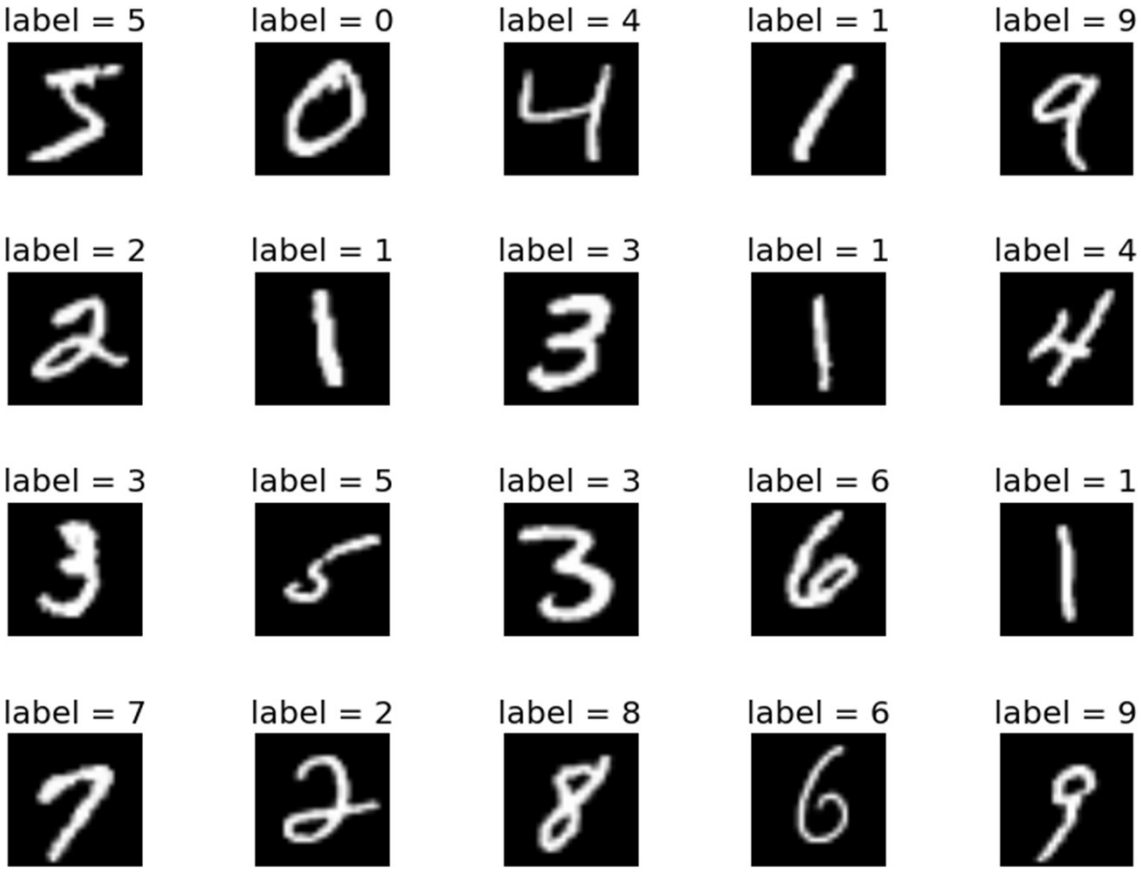
label = 5





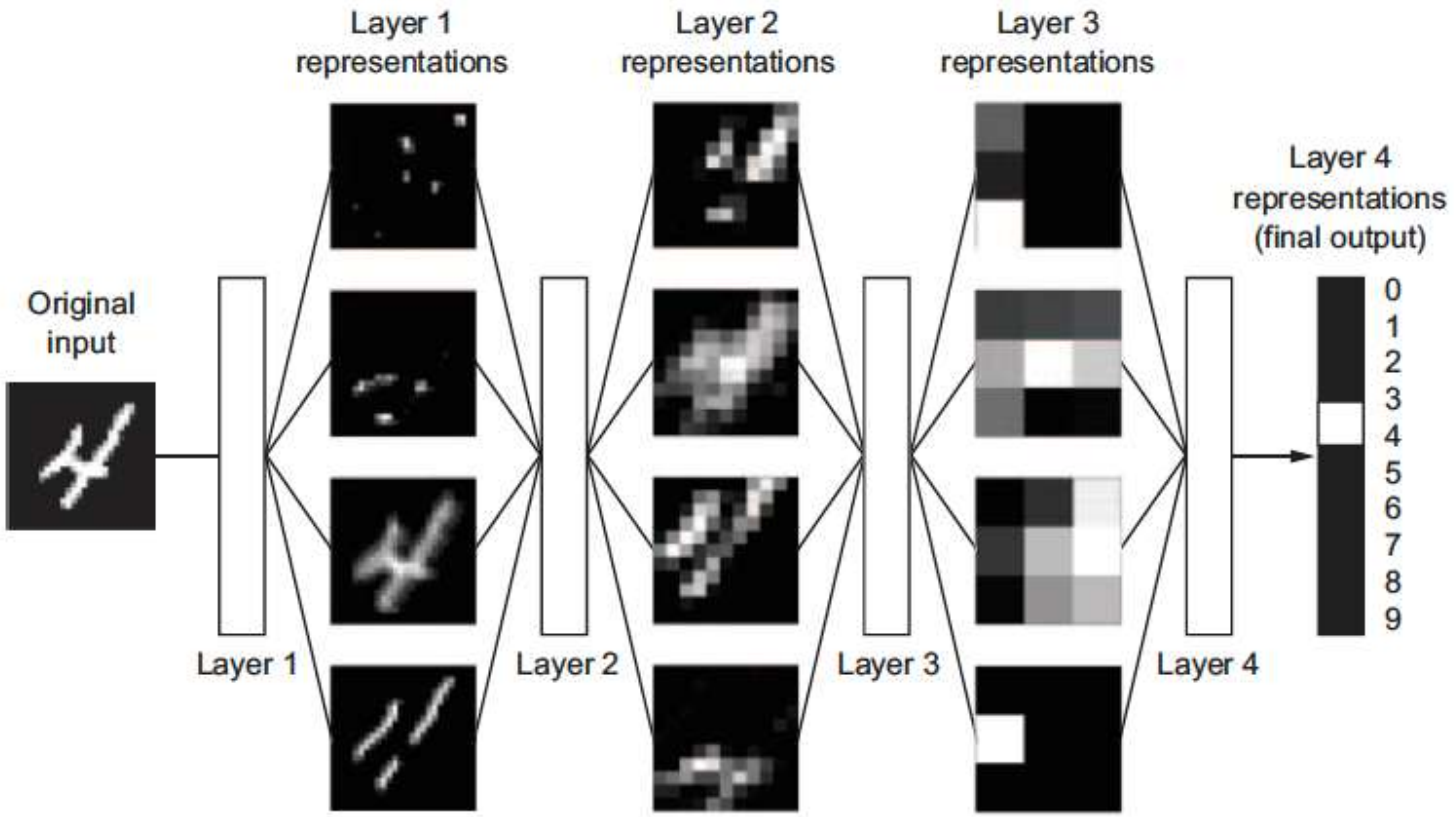
Convolutional Neural Networks (CNN) for Image Recognition

MNIST dataset used for image recognition

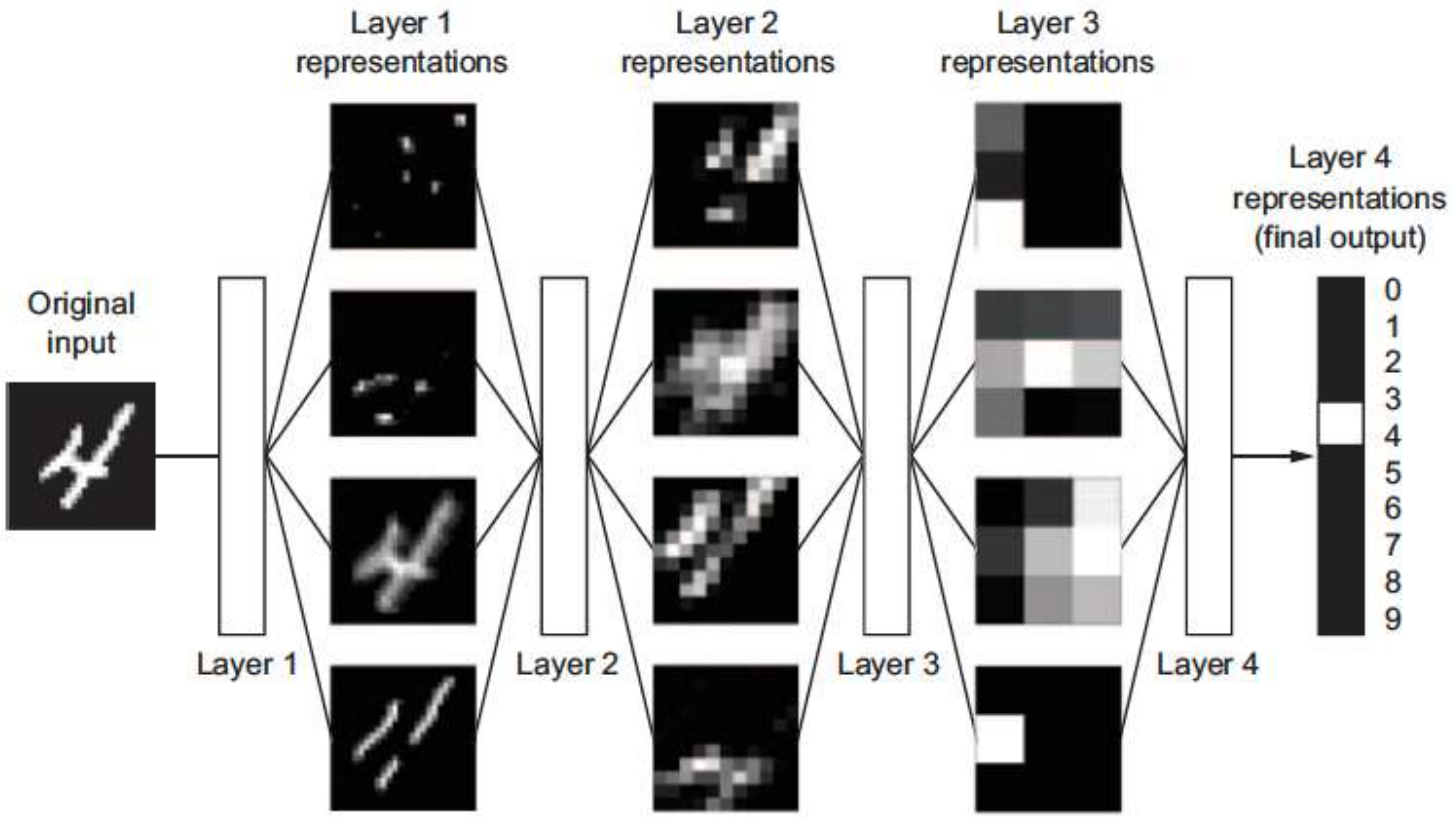


The MNIST database contains 60,000 training images and 10,000 testing images
MNIST is Modified National Institute of Standards and Technology database

Convolutional Neural Networks (CNN) for Image Recognition



Convolutional Neural Networks (CNN) for Image Recognition



Convolution Layers

Dense Layer = Fully Connected Layer

Convolution is a function used in Signal Processing. In most CNNs convolution is equivalent to filtering.

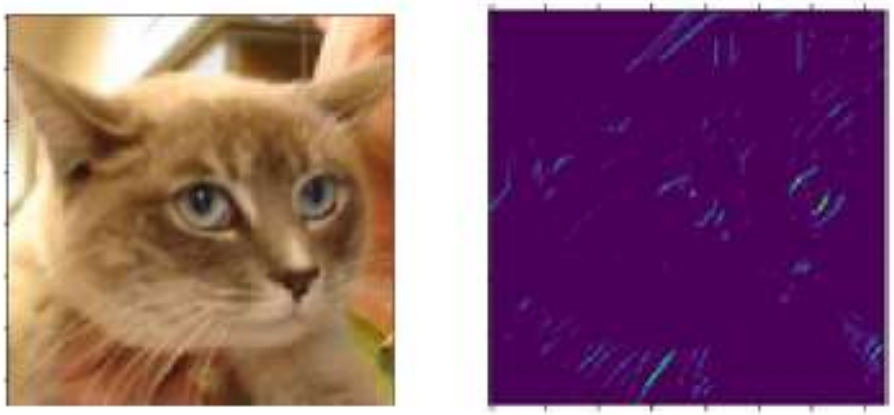
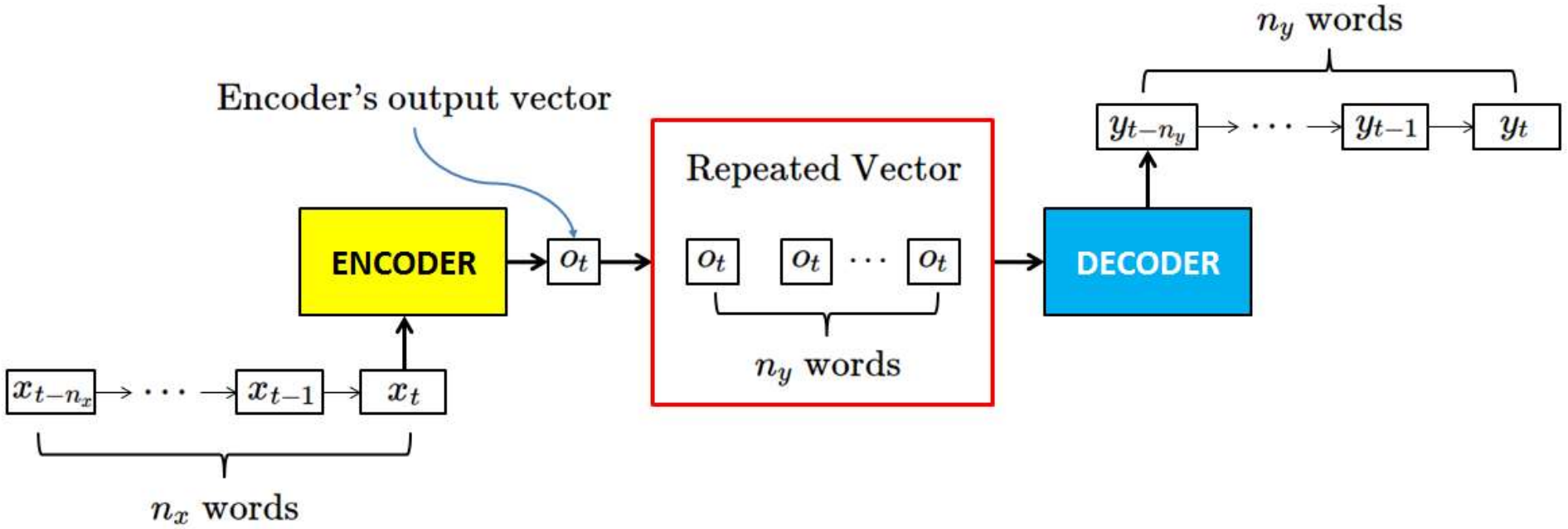


Image of a cat modified by the first convolutional layer.
We can interpret it as an eye filter.

More about Convolutional Neural Networks:
https://www.youtube.com/watch?v=aircAruvnKk

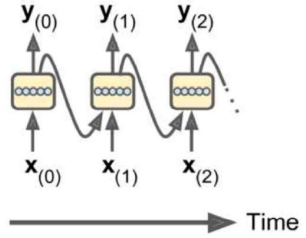
Some Other Types of Neural Networks

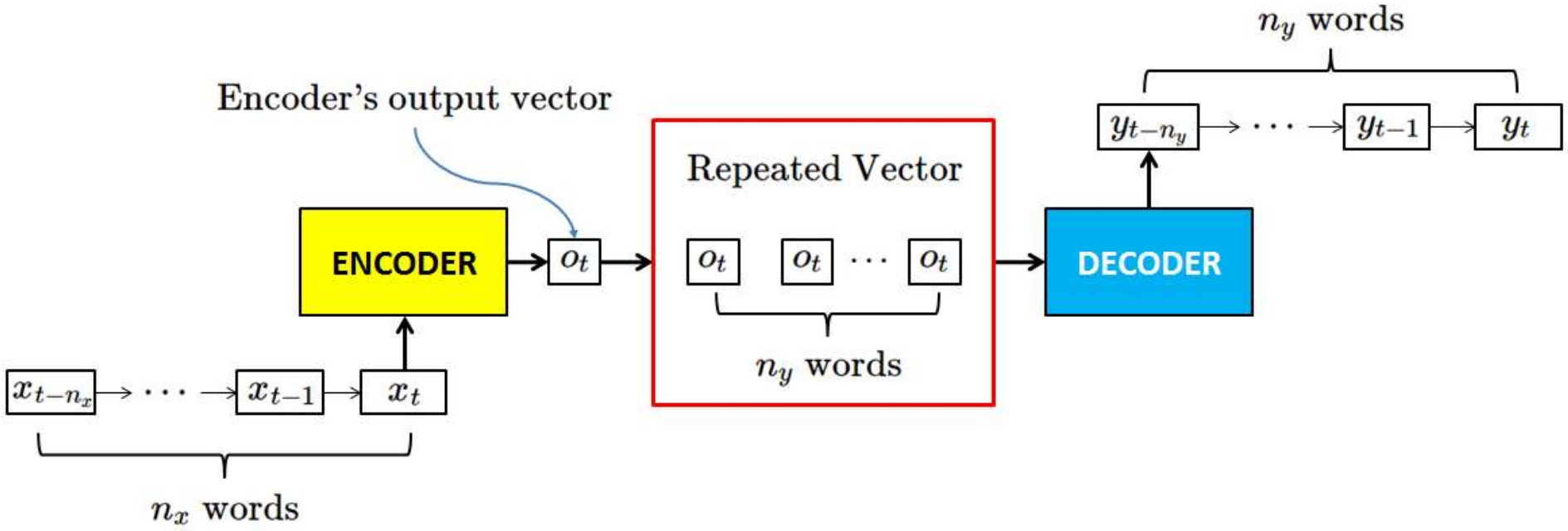
- Recurrent Neural Networks: speech recognition, language translation



Some Other Types of Neural Networks

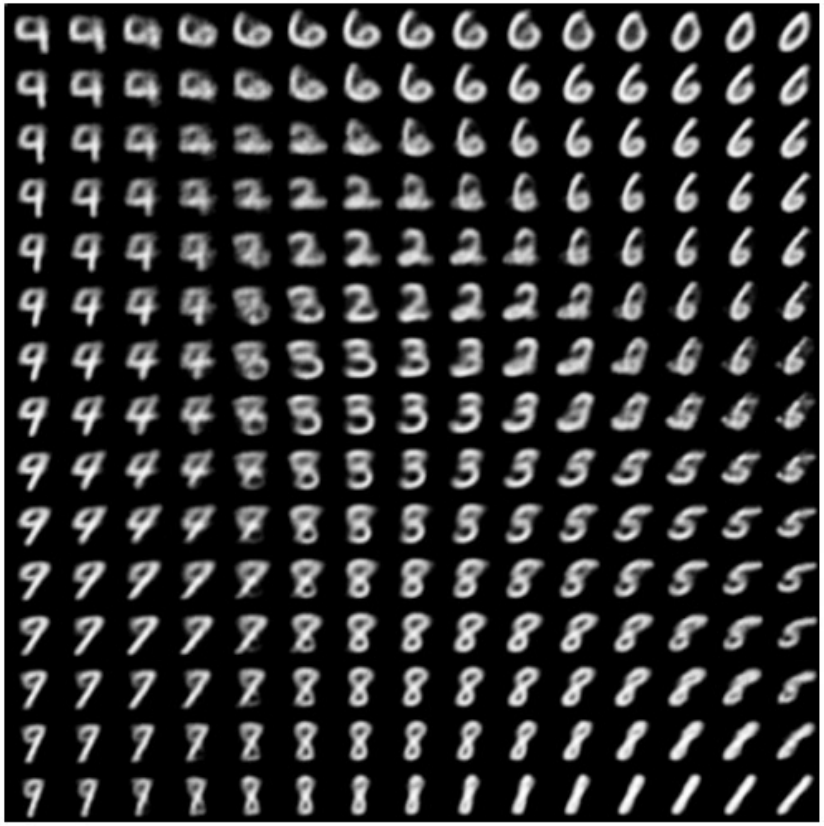
- Recurrent Neural Networks: speech recognition, language translation



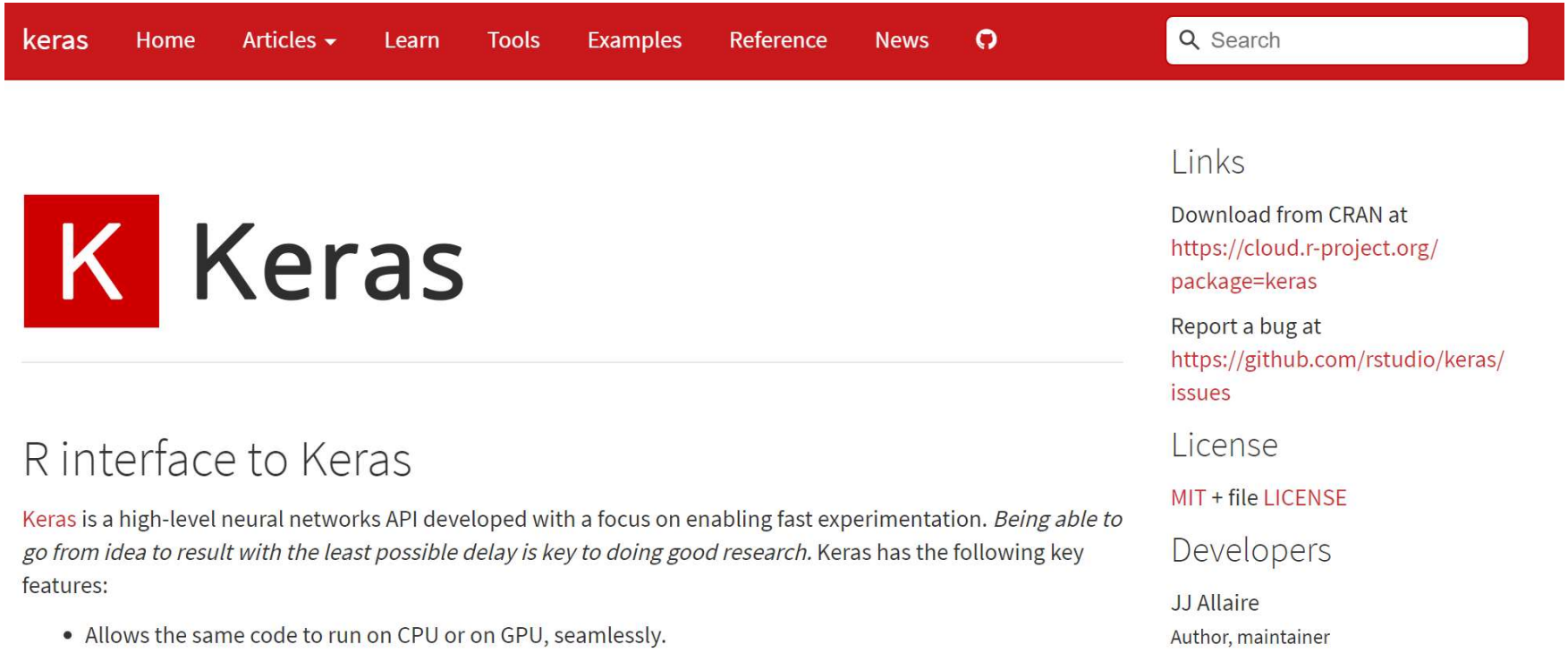


Some Other Types of Neural Networks

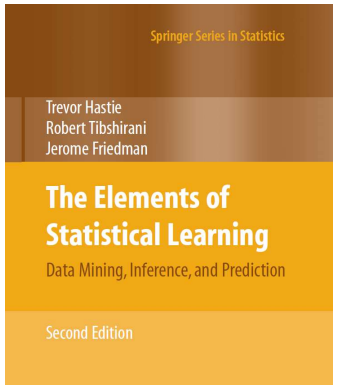
- Autoencoders: unsupervised learning, finding common classes, image generation



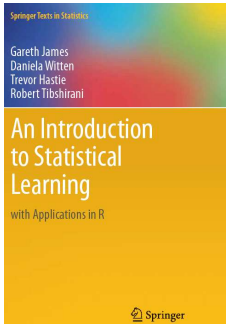
Further Reading



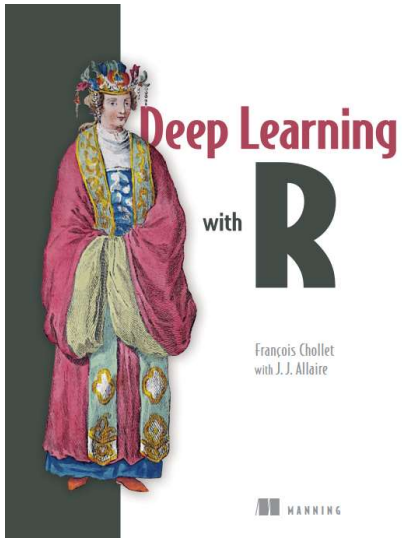
https://keras.rstudio.com/
https://conda.io/docs/user-guide/install/index.html
https://medium.freecodecamp.org/why-you-need-python-environments-and-how-to-manage-them-with-conda-85f155f4353c

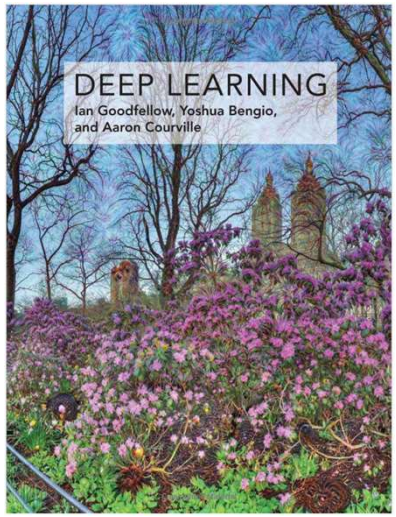


Further Reading



- http://statweb.stanford.edu/~tibs/ElemStatLearn/index.html
- http://www-bcf.usc.edu/~gareth/ISL/





https://www.youtube.com/watch?v=aircAruvnKk

Thank you!