MACHINE LEARNING IN R:

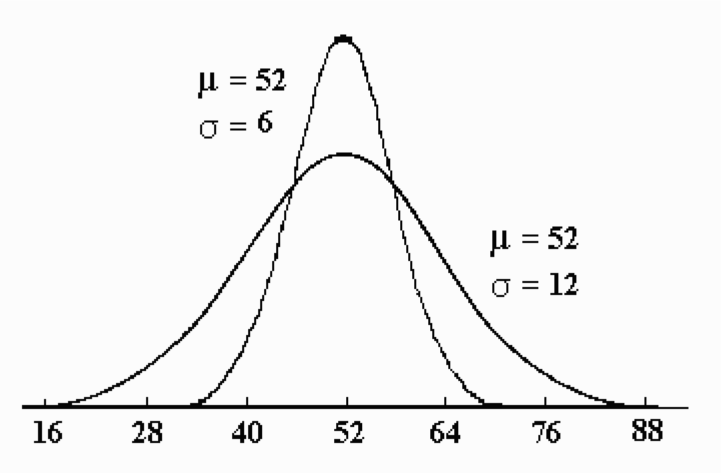
supervised classification

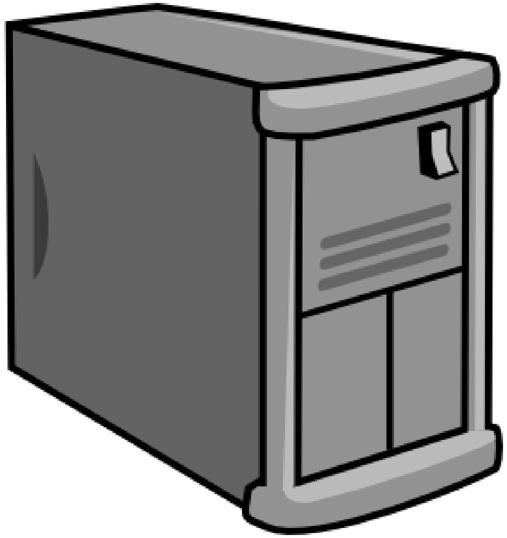
“I think we should be very careful about artificial intelligence. If I had to guess at what our biggest existential threat is, it’s probably that. So we need to be very careful,”

Elon Musk

Machine Learning	Statistics
network, graphs	model
focus on prediction	focus on inference
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

http://statweb.stanford.edu/~tibs/stat315a/





Statistics ⟷ **Machine Learning**

T-Test

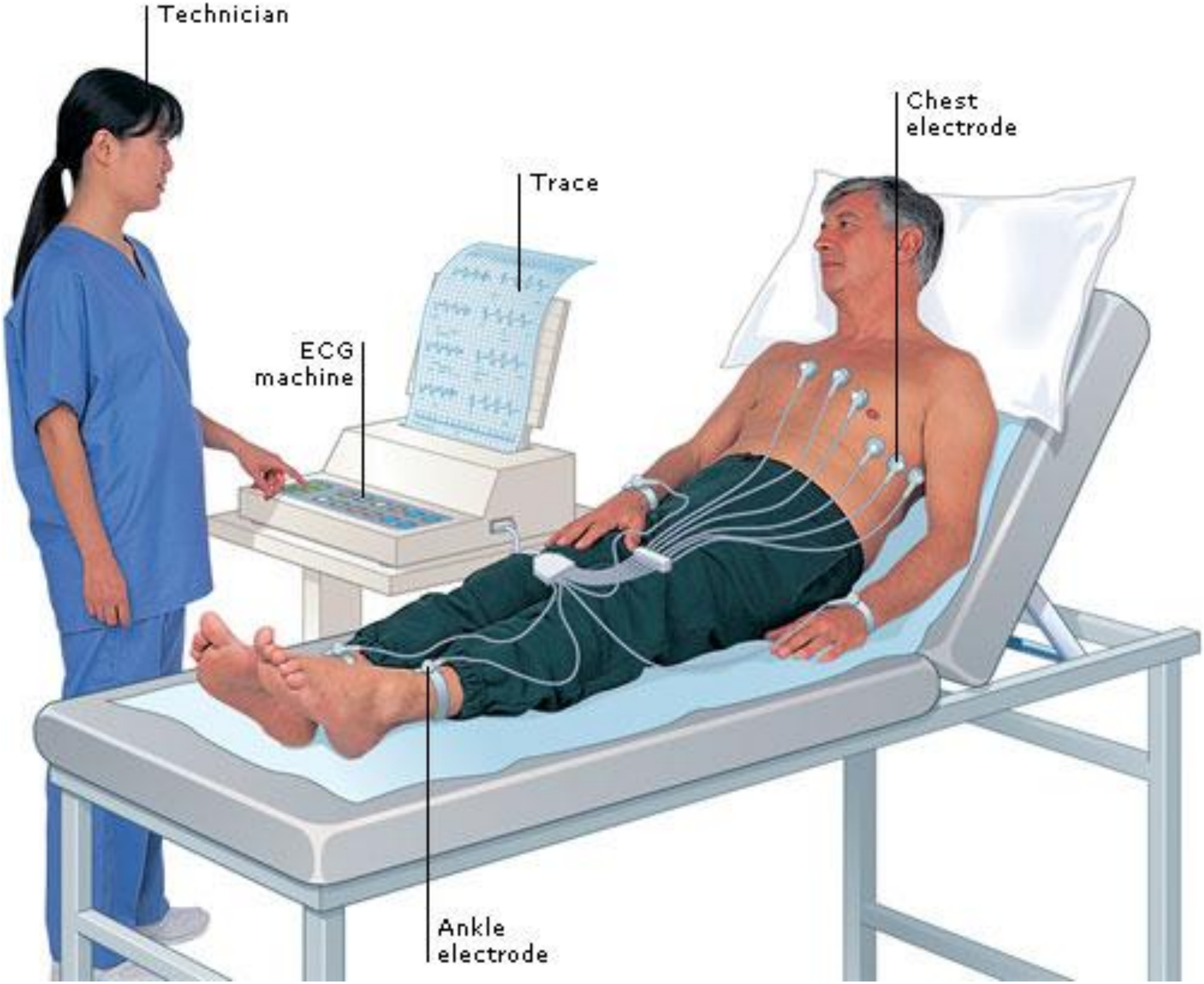
Logistic Regression

Elastic Net

Gradient Boosting

Deep Learning

Arrhythmia Data



http://archive.ics.uci.edu/ml/datasets/Arrhythmia

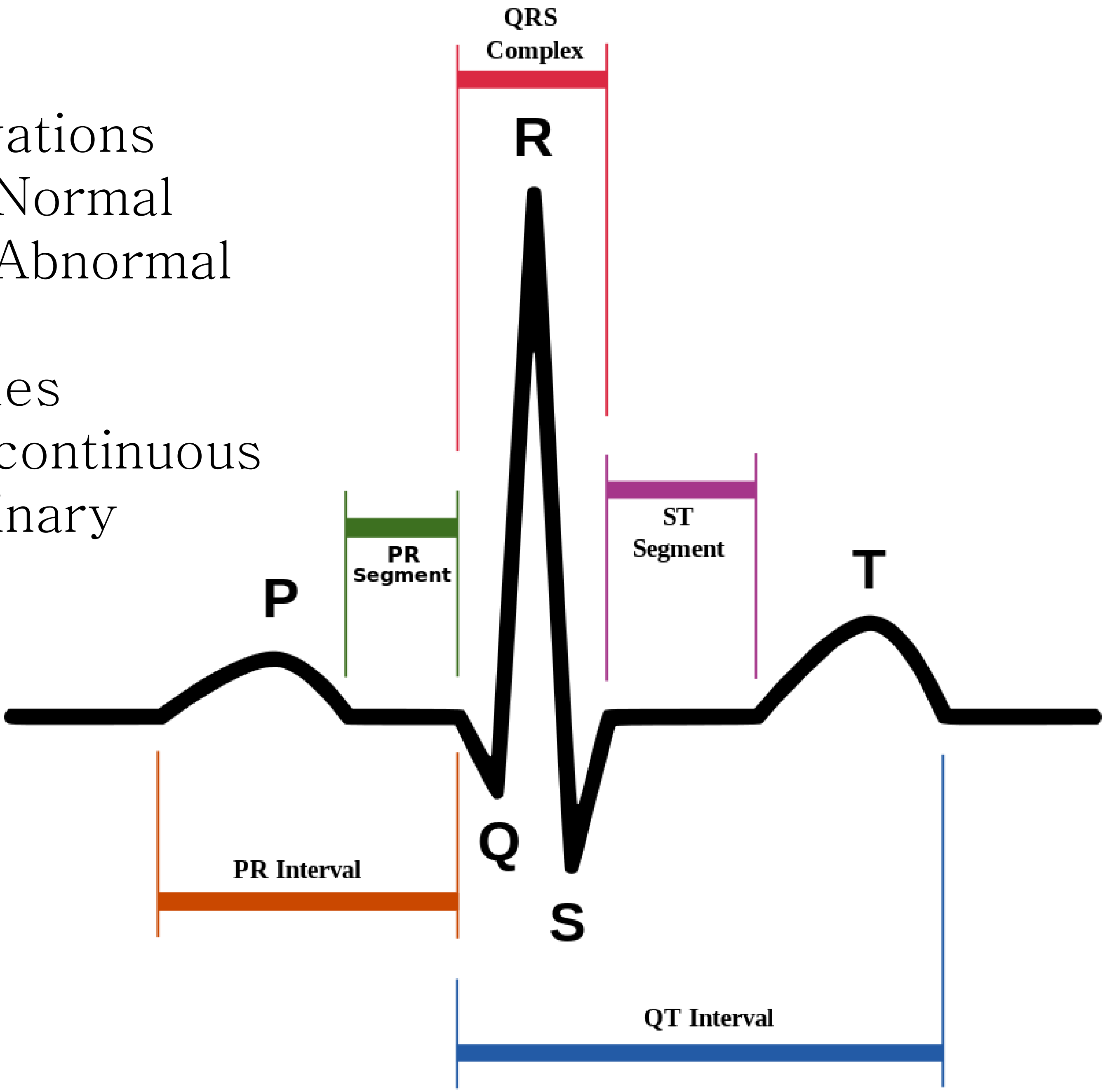
Data

451 observations

- 245 Normal
- 206 Abnormal

263 variables

- 198 continuous
- 65 binary



Data pre-processing

```
arrhythmia = read.csv('arrhythmia.csv')

set.seed(120)

# Make folds vector to specify holdout set
folds = sample(1:10, nrow(arrhythmia ), replace = TRUE)
response = arrhythmia$abnormal

# Formula for use in model matrix function
fmla <- abnormal ~ sex + di_width_ragged_r_wave +
di_width_diphasic_derivation_of_r_wave + …



# Make sparse model matrix
mm = Matrix::sparse.model.matrix( fmla, data =
arrhythmia)
```

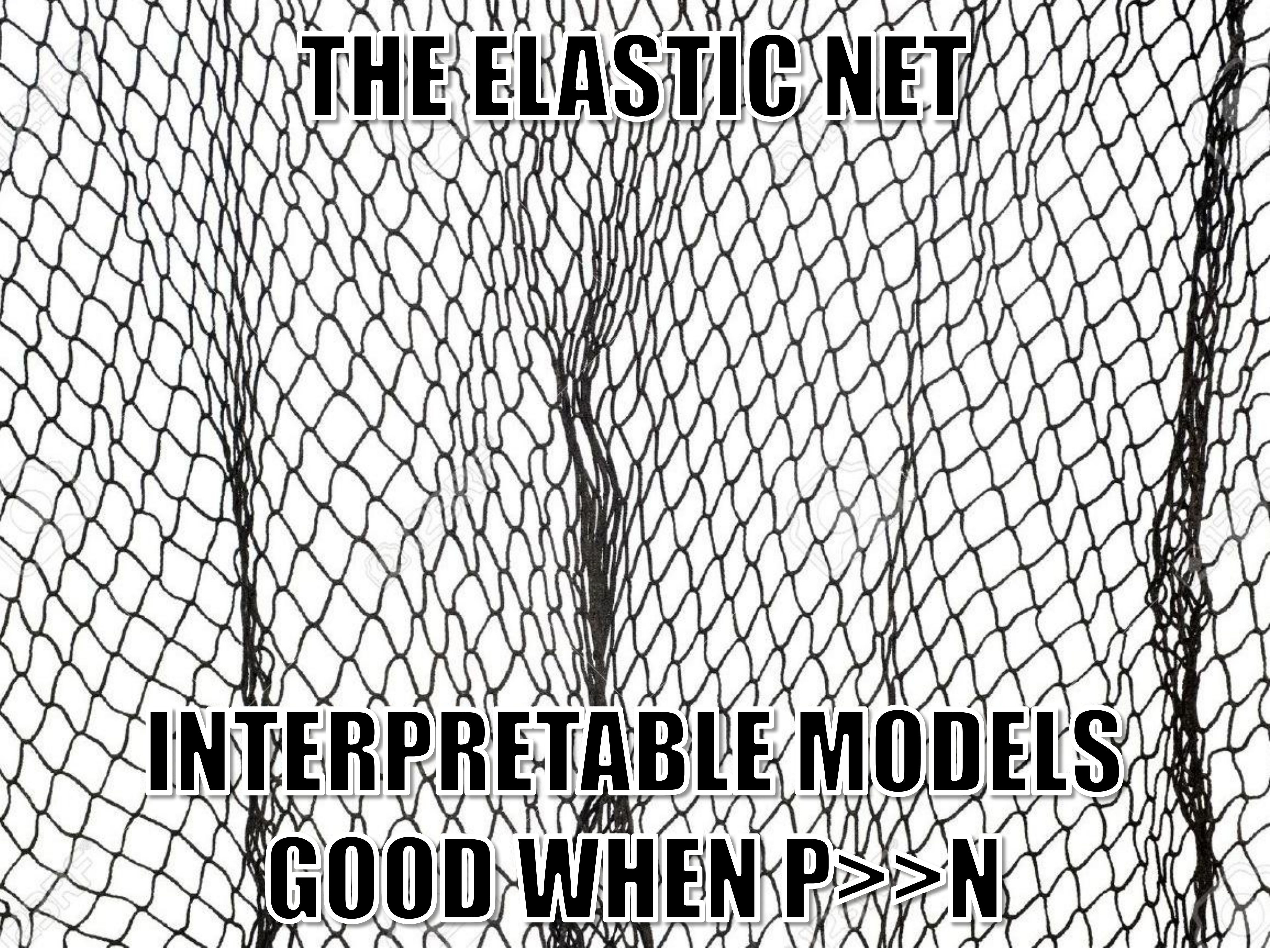
Logistic regression

10 events per variable?
Choose 20 ‘best variables’

```
#fit a logistic regression
glm1 = glm(fmla2,
      data = arrhythmia[folds < 9, , drop = TRUE],
      family = binomial())

#make predictions
p_logistic = predict(glm1,
      arrhythmia[folds >= 9, ],
      type = "response")

#get predictive accuracy
auc(p_logistic, response[folds >= 9] )
accuracy(p_logistic > 0.5, response[folds >= 9] )
```

AUC = 70%

THE ELASTIC NET

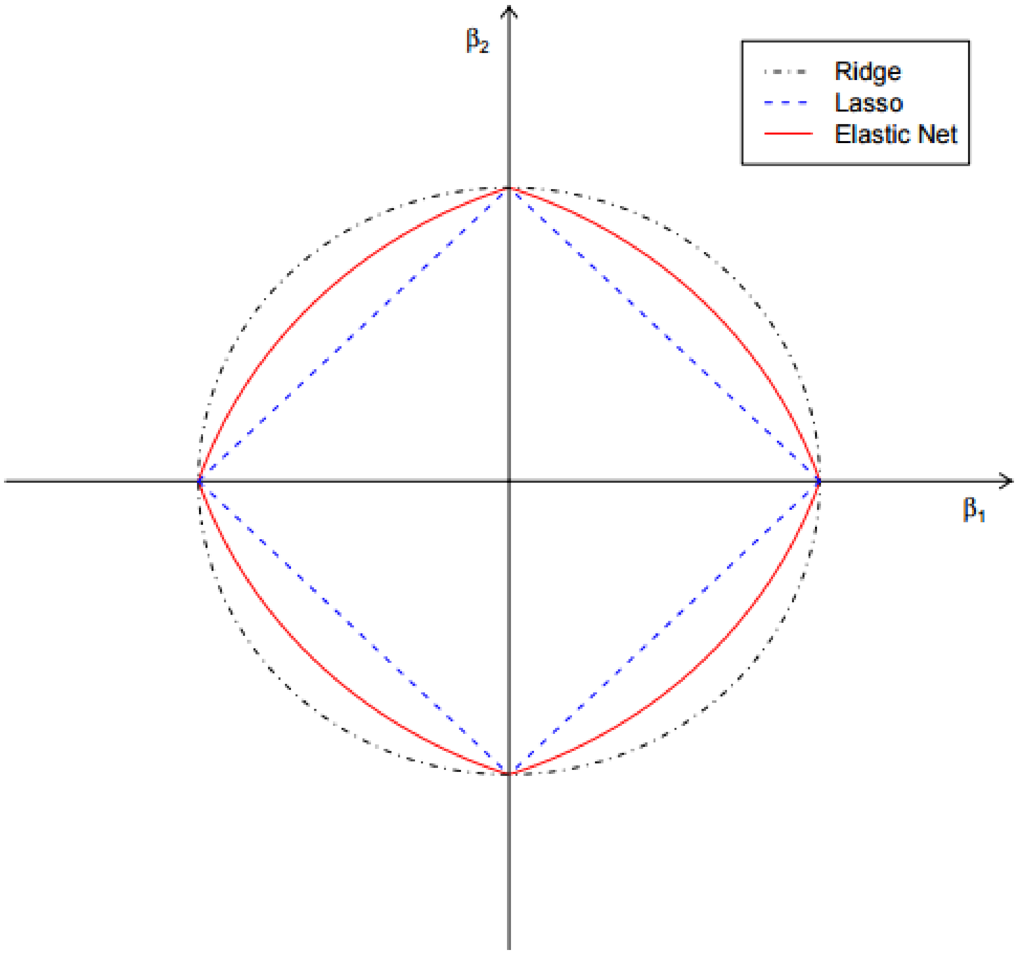


INTERPRETABLE MODELS

GOOD WHEN P>>N

Elastic Net

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1)$$



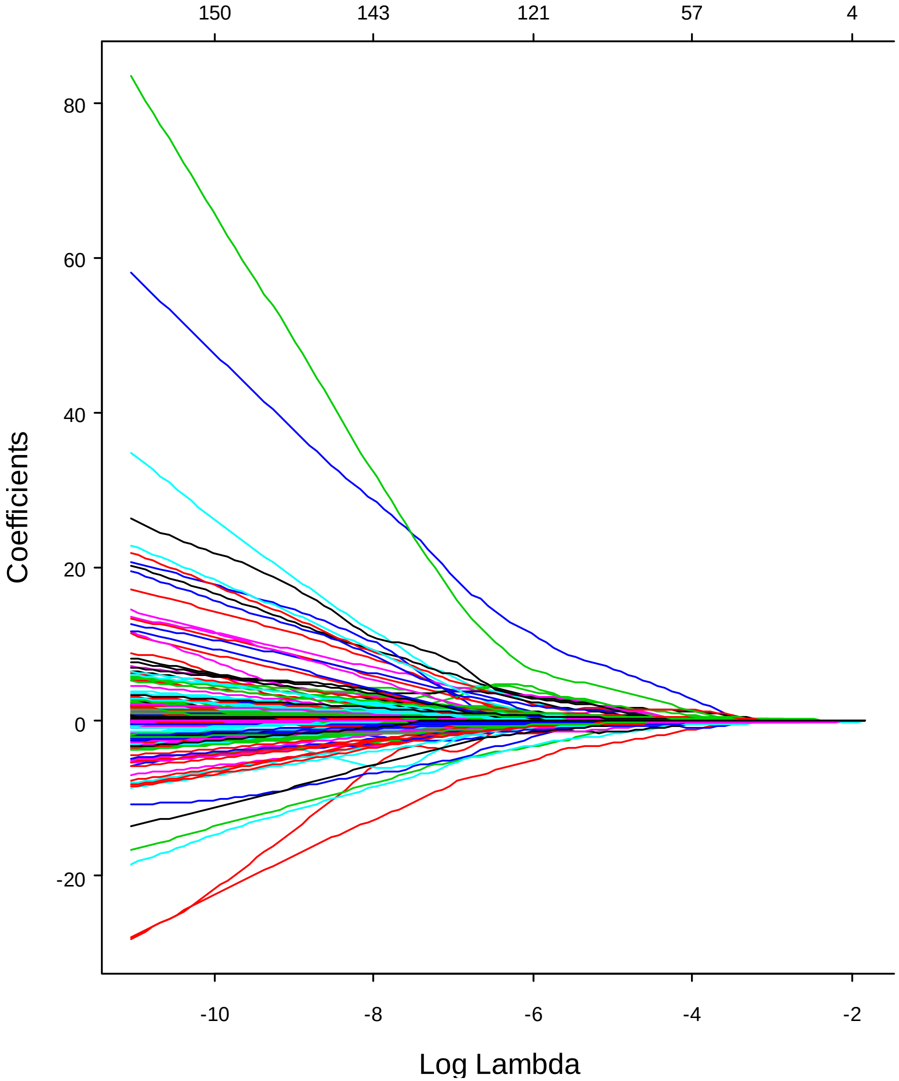
Elastic Net in R

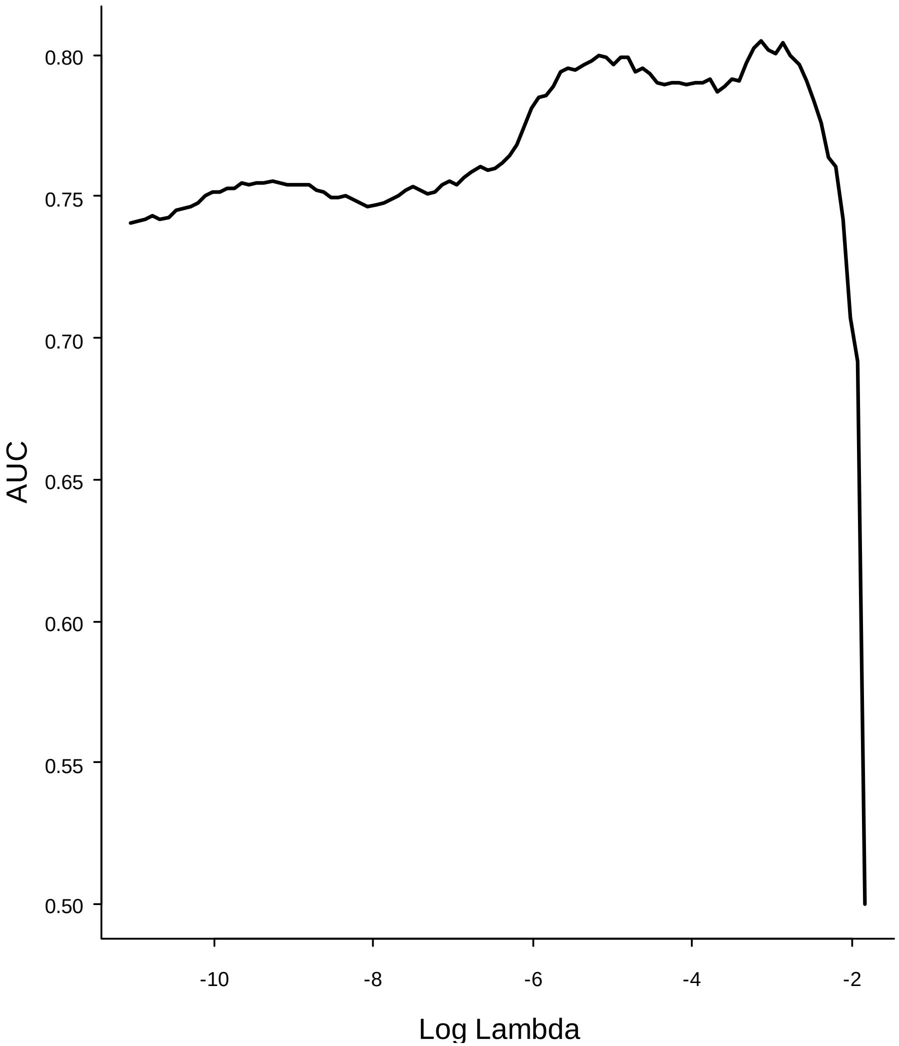
```
library(glmnet)
#fit model on training set
e_net_default = glmnet( x = mm[folds < 9,],
                  y = response[folds < 9],
                  family = "binomial")

#make predictions on test set
p_e_net_default = predict(e_net_default,
                  newx = mm[folds >= 9,],
                  type = "response")

str(p_e_net_default )
# num [1:98, 1:100] 0.467 0.467 0.467 0.467 0.467 ...
# - attr(*, "dimnames")=List of 2
# ..$ : chr [1:98] "11" "12" "15" "17" ...
# ..$ : chr [1:100] "s0" "s1" "s2" "s3" ...
```

Elastic Net in R

```r
#get auc for each prediction

e_net_default_auc = aaply(p_e_net_default, 2,
      function(.x){
            auc(y = response[ folds >= 9], prob = .x)
      })

#plot these results

par(mfrow = c(1,2))
plot(e_net_default, xvar = "lambda")
plot(y = e_net_default_auc ,
      x = log(e_net_default$ lambda ),
      xlab = 'Log Lambda',
      ylab = 'AUC', type = 'l', lwd = 2)
```

Elastic net plots



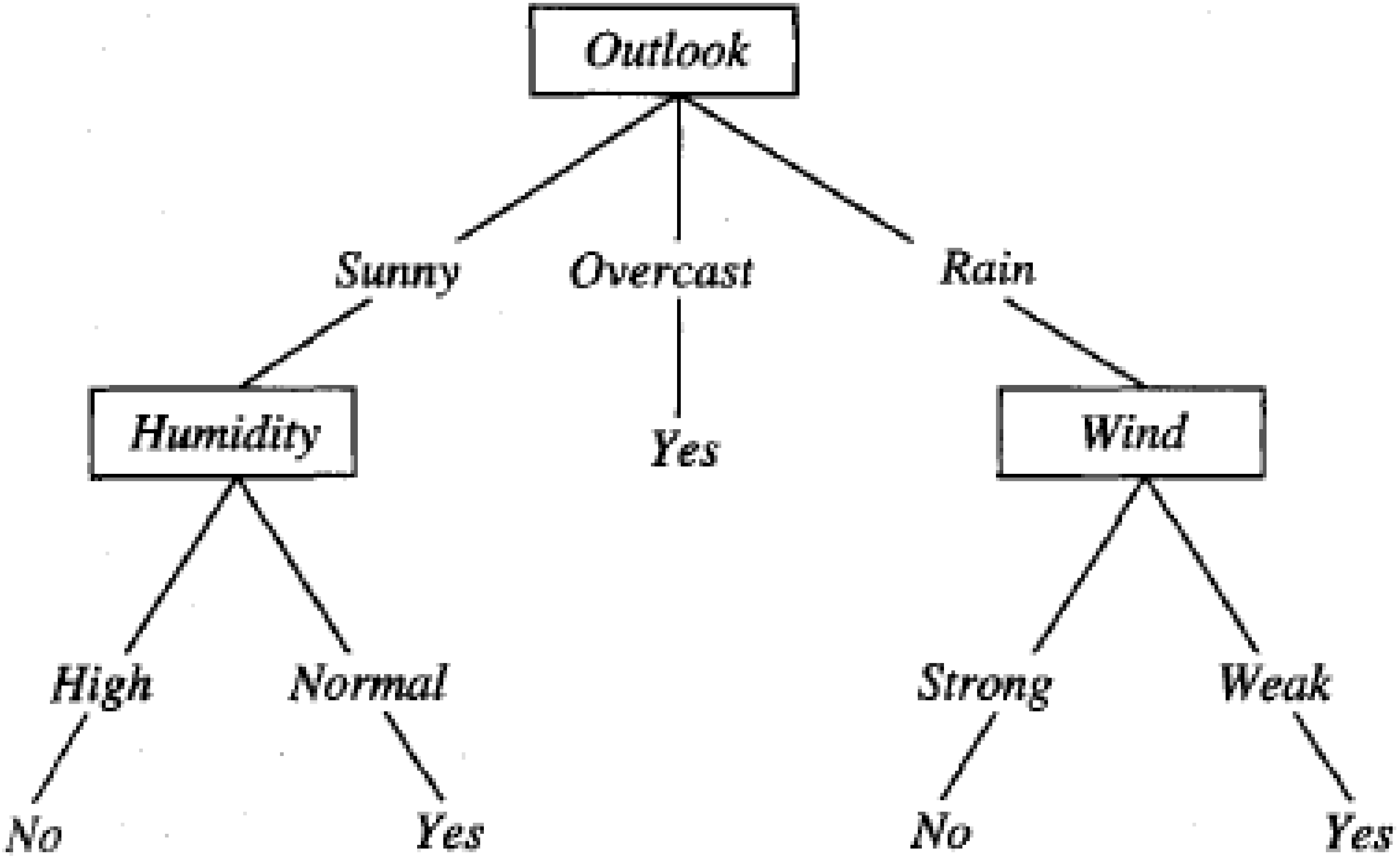


AUC = 80%

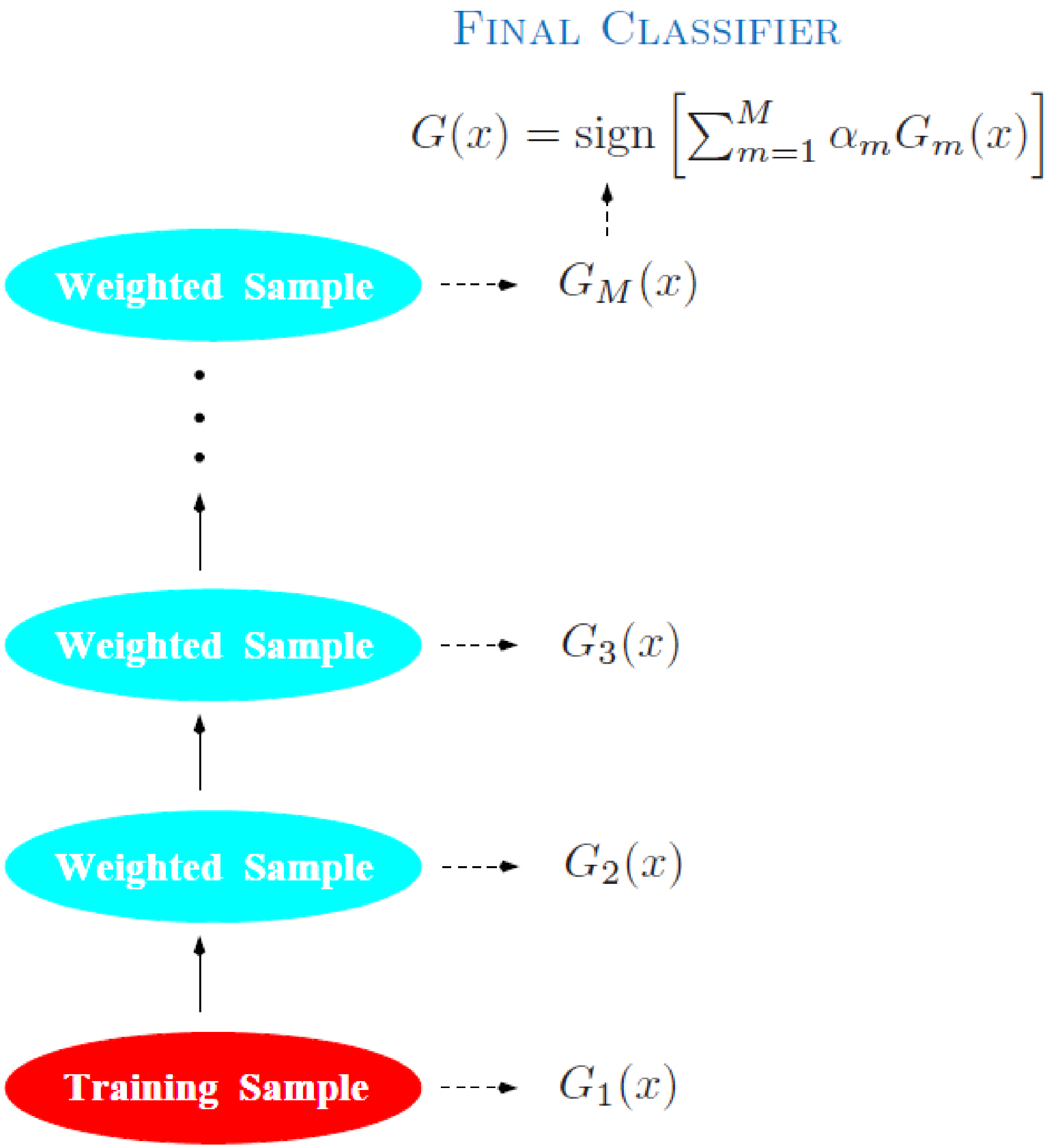
GRADIENT BOOSTING!

THE BEST OFF-THE-SHELF CLASSIFIER IN THE WORLD!

Trees


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Strong
Weak
No
Yes
No
Yes

Boosting



FINAL CLASSIFIER

$$G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$$

Weighted Sample $G_M(x)$

Weighted Sample $G_3(x)$

Weighted Sample $G_2(x)$

Training Sample $G_1(x)$

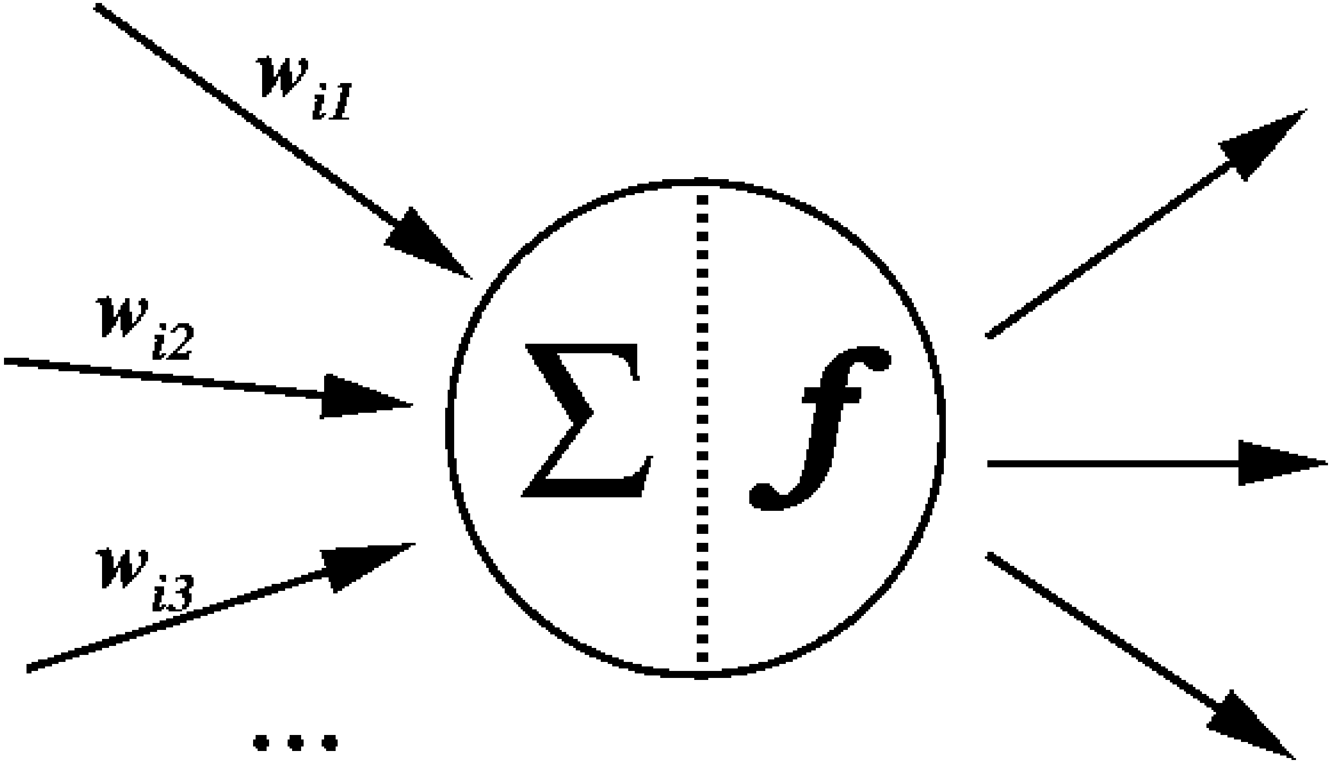
Gradient Boosting in R

```r
library(xgboost)

#fits model
boost_default = xgboost(data = mm[folds < 9,],
                  label = response[folds < 9],
                  nrounds = 50,
                  objective = "binary:logistic")

#makes predictions
p_boost_default = predict(boost_default,
                  mm[folds >= 9,])

auc (p_boost_default, response[folds >= 9] )
accuracy (p_boost_default > 0.5, response[folds >= 9] )
```

AUC = 83%

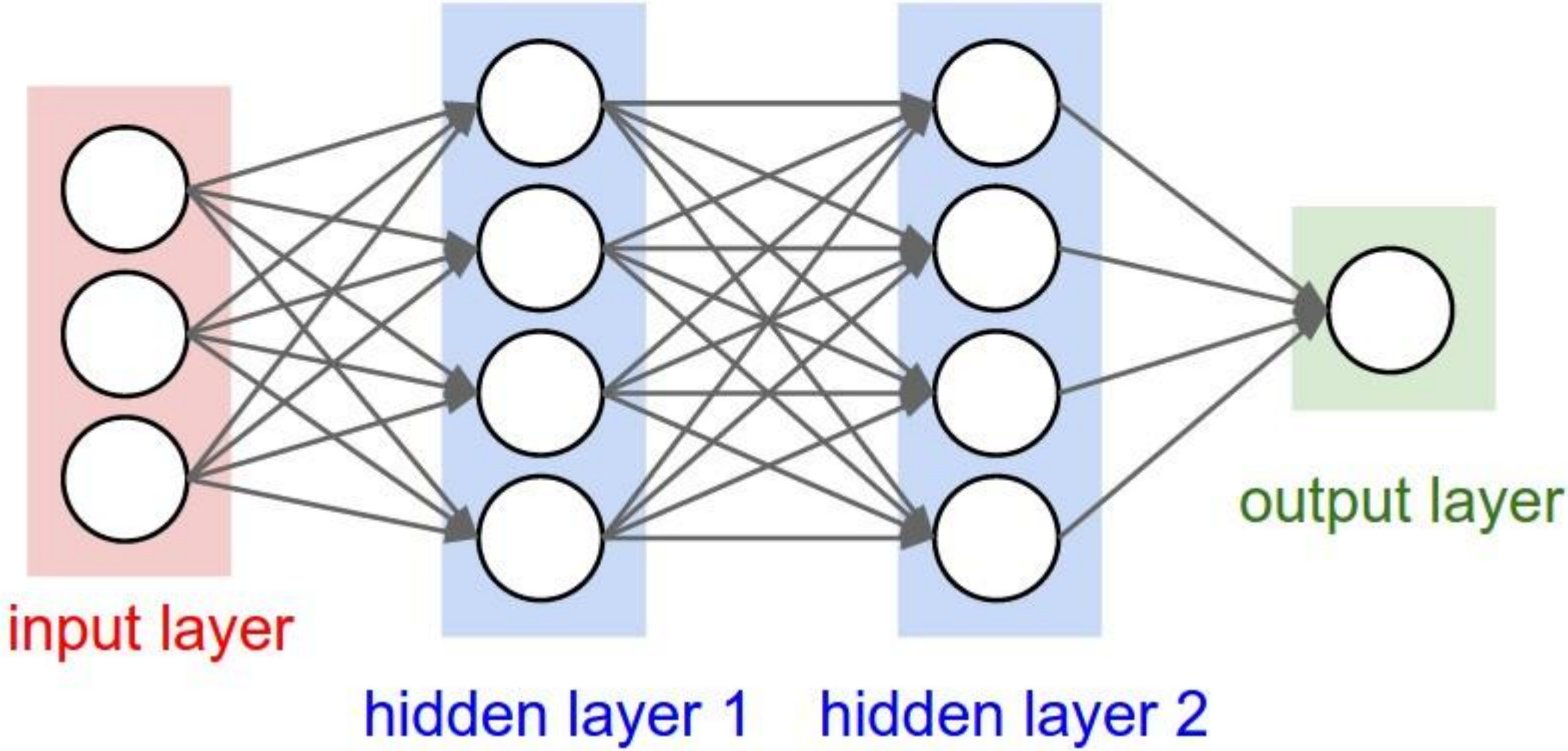
DEEP LEARNING IS LIKE LOVE
NO ONE IS SURE WHAT IT IS,
BUT EVERYONE WANTS IT

Neural Networks



$$y_i = f(net_i)$$

‘Deep’ Neural Networks



Deep Learning in R

```r
library(h2o)

# initialise h2o
localH2O <-h2o.init(ip = "localhost",
                  port = 54321, startH2O = TRUE)

#get data in h2o format
dat_h2o <- as.h2o(arrhythmia)

#get vector of predictor locations
predictors = which(!names(arrhythmia) %in%
                        c("arrhythmia", "abnormal" ))
```

Deep Learning

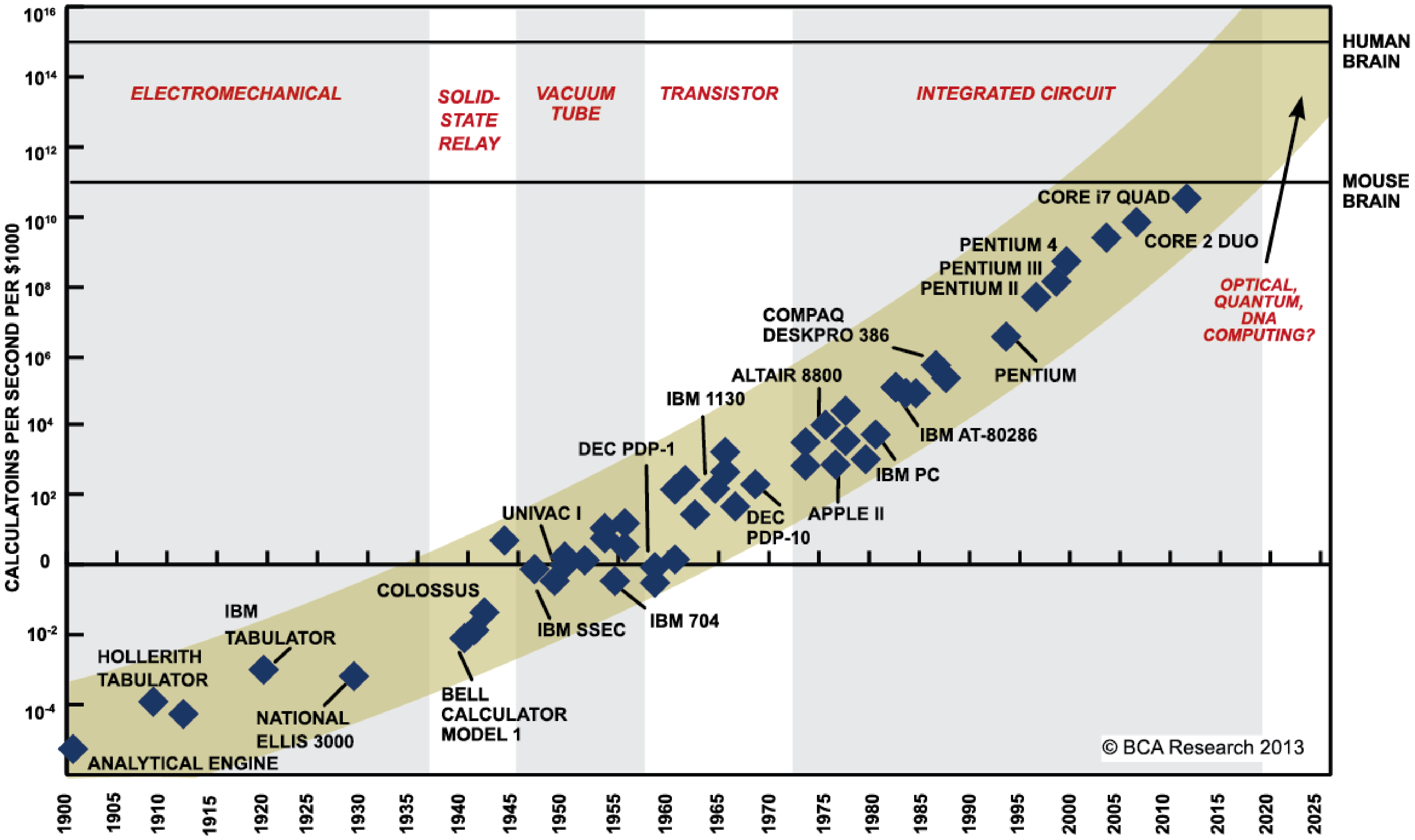
```
#fit model
dl_fit <- h2o.deeplearning(x = predictors,
         y = 'abnormal',
         training_frame = dat_h2o[which(folds < 9),],
         epochs = 50)

#make predictions
pred_dl <- h2o.predict(dl_fit,
            dat_h2o[which(folds >= 9) , ])

pred_dl <- as.data.frame(pred_dl)

auc( pred_dl$`TRUE.`, response[folds >= 9])
accuracy( pred_dl$`TRUE.`, response[folds >= 9])
```

AUC = 84%

Results

Model	AUC
Logistic Regression	70
Elastic Net	80
Gradient Boosting	83
Deep Learning	84
Combined model	85

THE SINGULARITY IS NEAR?



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Other topics

- Cross validation
- Tuning parameters
- Bias/Variance trade-off
- Distributed learning/big data
- Unsupervised Learning
- Unstructured data

- … many more…

Thanks!

Scripts and data available here:
https://github.com/tomliptrot/Machine_Learning_ManchesterR/