Building meaningful machine learning models for disease prediction

Dr Shirin Glander

Dep. of Genetic Epidemiology
Institute of Human Genetics
University of Münster

shirin.glander@wwu.de

https://shiring.github.io
https://github.com/ShirinG

Friday, 31st March 2017

About me

since 2015	Bioinformatics Postdoc Next Generation Sequencing autoinflammatory diseases & innate immunity
2011 - 2015	PhD in Biology Is the immune system of plants required to adapt to flowering time change?
2005 - 2011	BSc and MSc of Science in Biology evolutionary genetics, immune memory in Drosophila

Table of contents

Building meaningful machine learning models for disease prediction

1. Machine Learning (ML) in disease modeling
2. What makes a model meaningful?
3. A quick recap of ML basics
4. How to build ML models in R
5. Evaluating model performance

Machine Learning (ML) in disease modeling

ML in disease modeling

- tools that can interpret "big medical data"
- and provide fast, accurate and actionable information
- for precision or personalized medicine

Examples:

- computer-aided diagnosis of breast cancer from mammograms[1]
- identifying gene defects with facial recognition software[2]
- identifying signatures of Brain Cancer from MRSI[3]
- ... and many more ...

Image source: Wikimedia Commons

[1] Doi 2007.

[2] Levenson 2014.

[3] Sadja 2006.

What makes a model meaningful?

Can we trust a model?

- most ML algorithms model high-degree interactions between variables
- ML models are hard (or impossible) to interpret!
- we often don't know WHY they make decisions
- therefore, it is crucial that our models are meaningful

Image source: Pixabay

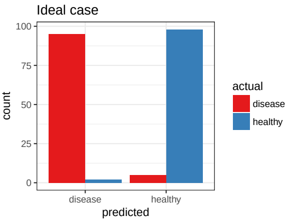
What makes a model meaningful?

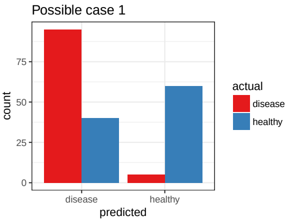
- creating ML models is relatively easy
- creating good or meaningful models is hard

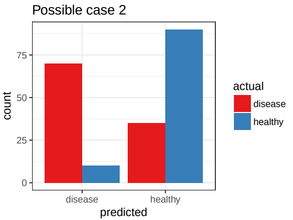
Meaningful models

- are generalizable
- answer the question(s) posed...
- ... with sufficient accuracy to be trustworthy

Accuracy depends on the problem!


Ideal case

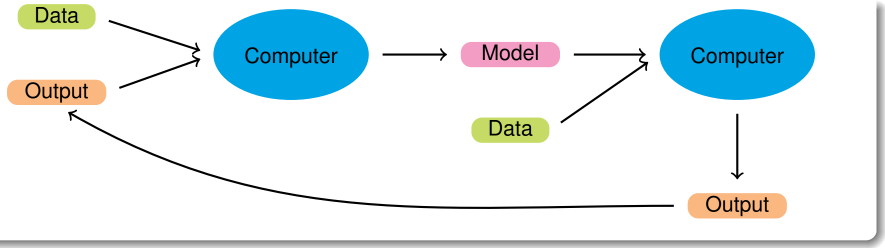

Possible case 1


Possible case 2

A quick recap of ML basics

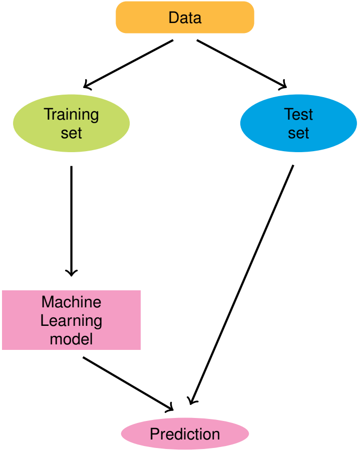
Machine learning

- artificial intelligence (AI)
- data-driven
- algorithms learn by being trained on observed data...
- ... and predict unknown data
- the increase in computational capacity has made ML more accessible

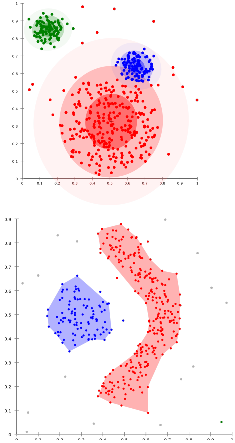


Supervised vs Unsupervised learning

Supervised



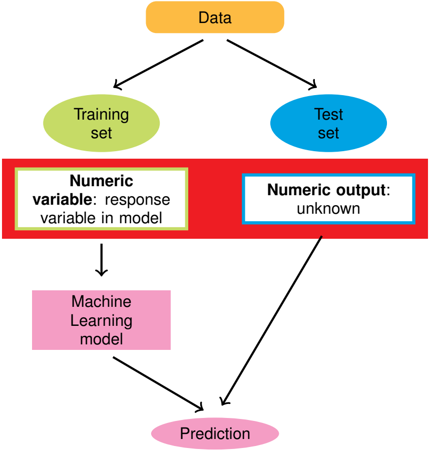
Unsupervised



Wikipedia

Classification vs Regression

Regression
e.g. weight loss

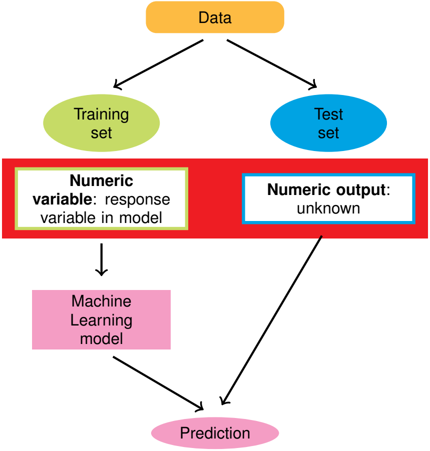


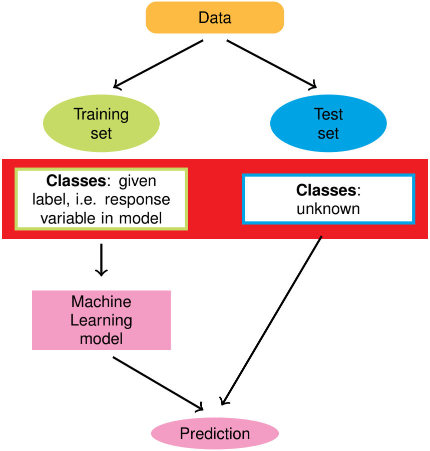
Data
Training set
Test set
Numeric variable: response variable in model
Numeric output: unknown
Machine Learning model
Prediction

Classification vs Regression

Regression
e.g. weight loss

Classification
e.g. healthy vs disease


Data
Training set
Test set
Numeric variable: response variable in model
Numeric output: unknown
Machine Learning model
Prediction



Features

- variables used for model training.
- using the right features is crucial.

- More is not necessarily better (overfitting)!

- feature selection
- feature extraction/ engineering

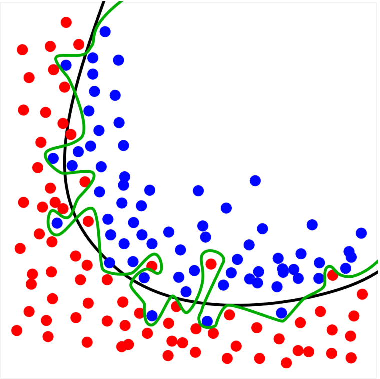
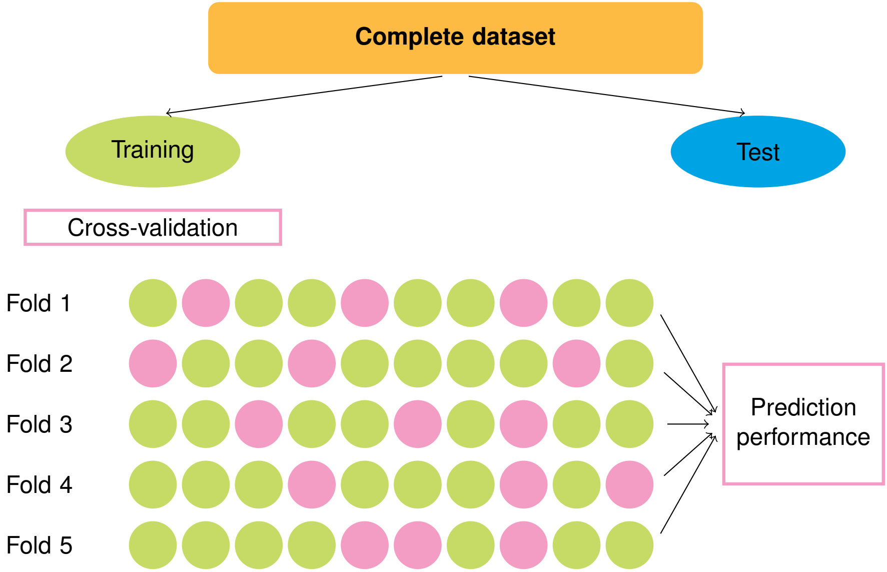


Image Source: Wikipedia

Training, (cross-) validation and test data



Complete dataset

Training

Test

Cross-validation

Fold 1

Fold 2

Fold 3

Fold 4

Fold 5

Prediction performance

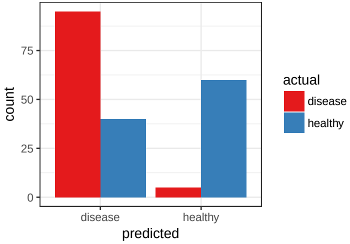
Take home messages:

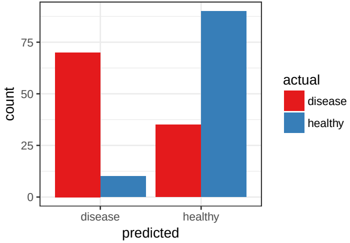
- ML models learn on observed data
- and predict unknown data

- creating ML models is easy
- creating good models is hard

Meaningful models

- answer specific questions
- are able to generalize to unseen data
- can be trusted

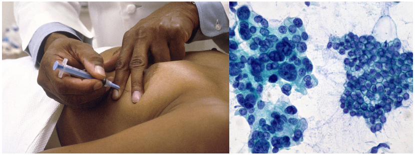

Possible case 1


Possible case 2

How to build ML models in R

Session setup

- Breast Cancer Wisconsin Dataset[4]



- caret[5]
- h2o[6]

Code will be available on my website and on Github

[4] W. H. Wolberg and O. L. Mangasarian (1990). “Multisurface method of pattern separation for medical diagnosis applied to breast cytology.” In: *Proceedings of the National Academy of Sciences* 87.23, pp. 9193–9196.

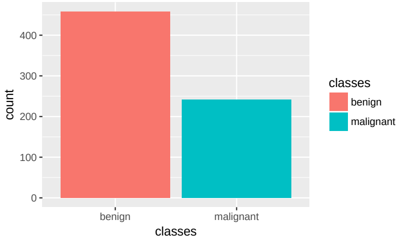
[5] M. Kuhn et al. (2016). *caret: Classification and Regression Training*. R package version 6.0-71.

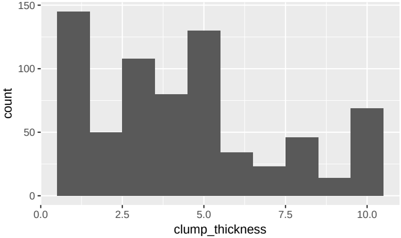
[6] H2O.ai (2017). *h2o: R Interface for H2O*. . R package version 3.10.3.6.

Get to know your data

Response variable

- Is it balanced?



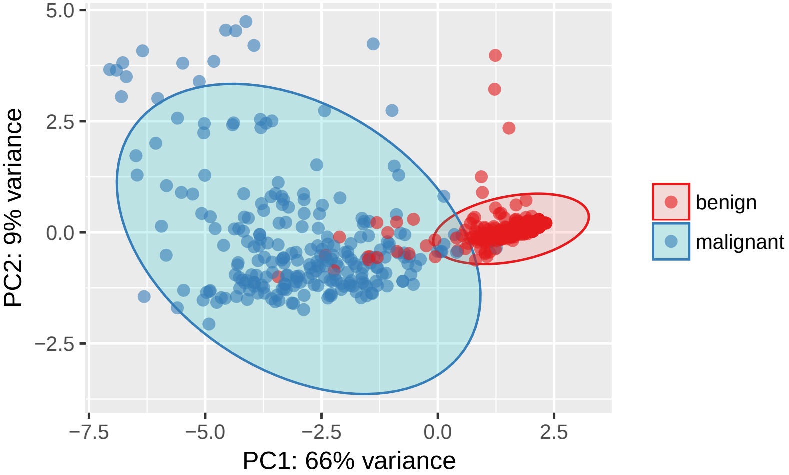


Missing data

- Is there missing data?
- Can we afford to loose data points?
- Or do we use imputation (and introduce additional uncertainty)?

Get to know your data

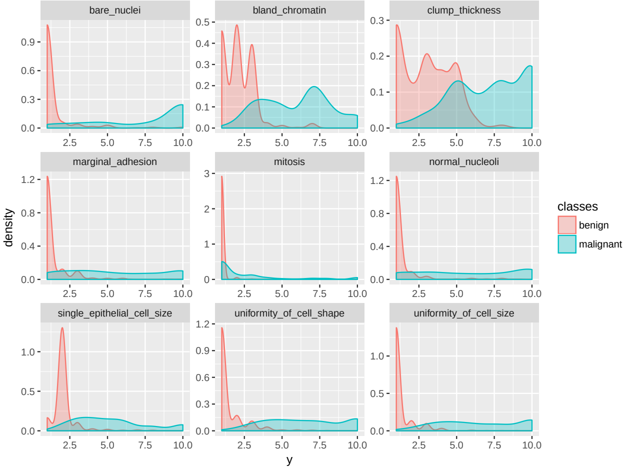
Principal Component Analysis (PCA)



Get to know your data

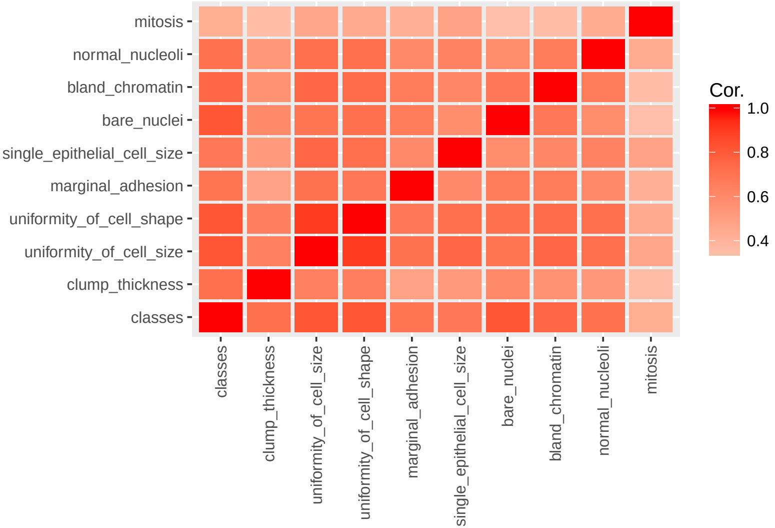
Features

- factors or numeric
- pre-processing



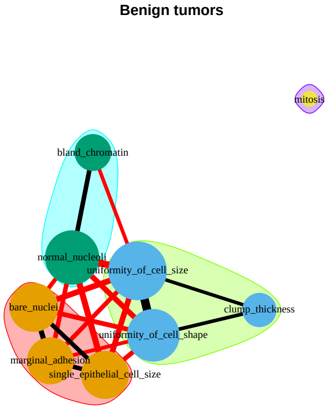
Get to know your data

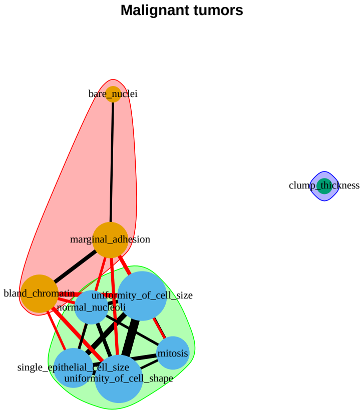
Correlation



Get to know your data

Correlation graphs

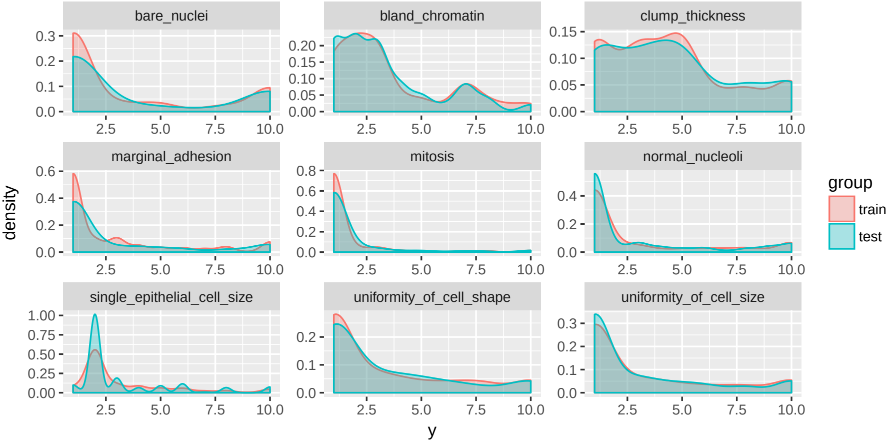




Training, validation and test data

We need to split the data into training and test sets - ideally stratified by response class.

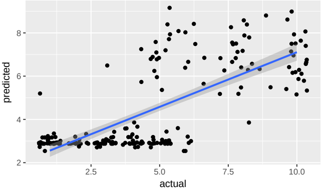
Density distribution



Model examples

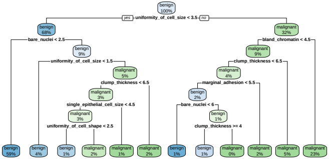
Regression with Linear Models

- e.g. Generalized Linear Models
- with *caret*



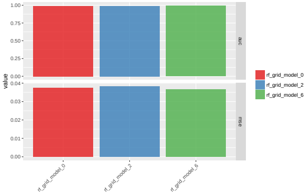
Tree-based classification

- Random Forest or Gradient boosting trees
- with *caret*



Hyper-parameter tuning

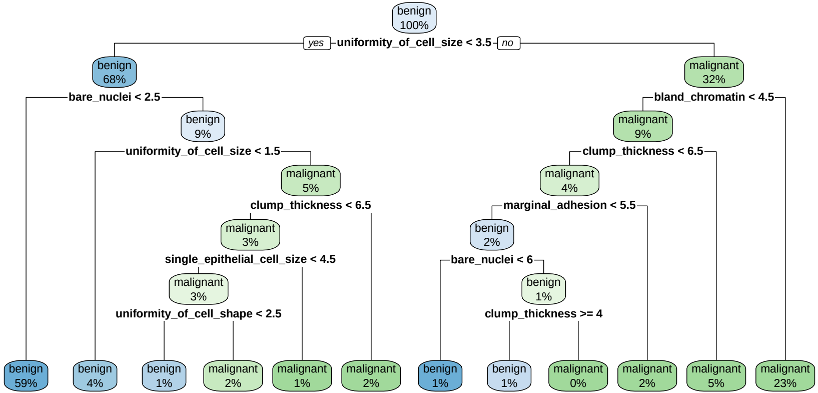
- Grid Search
- with *h2o*



Classification with tree-based models

Decision trees

e.g. Random Forest and gradient boosting trees



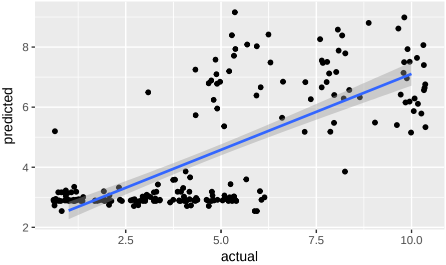
Evaluating model performance

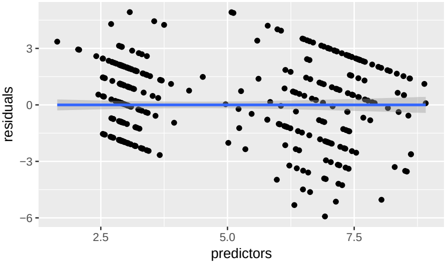
Never use the same data for evaluation that you used for training!

Predictions on test data

Regression

- RMSE: 1.97
- R^2: 0.50

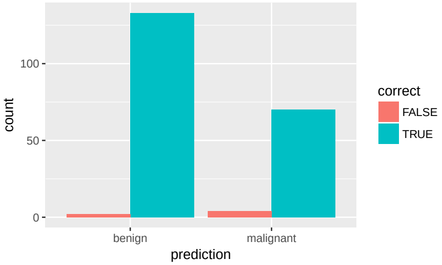


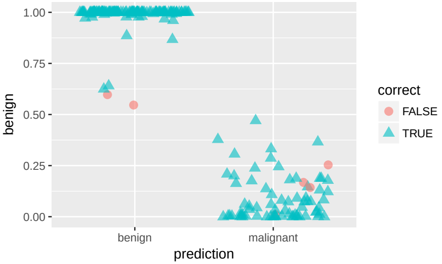


Predictions on test data

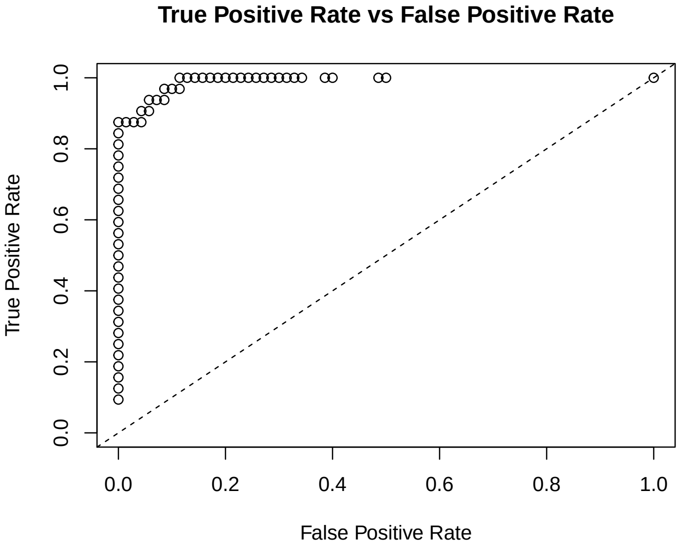
Classification

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction  benign malignant
##   benign       133         2
##   malignant      4        70
##
##                Accuracy : 0.9713
##                  95% CI : (0.9386, 0.9894)
##     No Information Rate : 0.6555
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9369
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9708
##             Specificity : 0.9722
##          Pos Pred Value : 0.9852
##          Neg Pred Value : 0.9459
##              Prevalence : 0.6555
##          Detection Rate : 0.6364
##    Detection Prevalence : 0.6459
##       Balanced Accuracy : 0.9715
##
##        'Positive' Class : benign
##
```





Area Under the Curve (AUC)



Hyper-parameter tuning with grid search

- h2o.grid()
- Random Grid Search (RGS) or Cartesian Grid

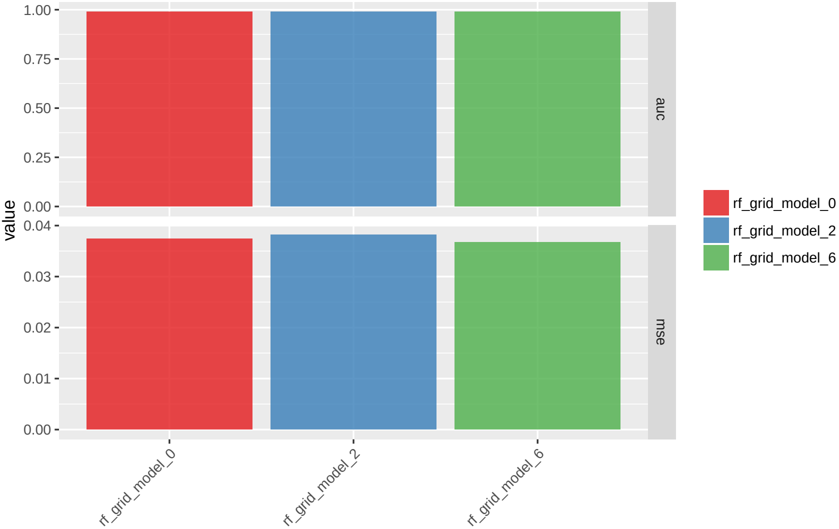
Define a set of hyper-parameters:

- number of trees
- maximum tree depth
- fewest allowed (weighted) observations in a leaf
- etc.

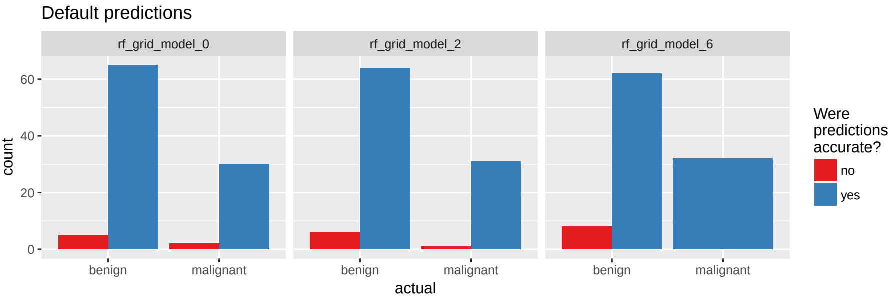
Choose best model from grid:

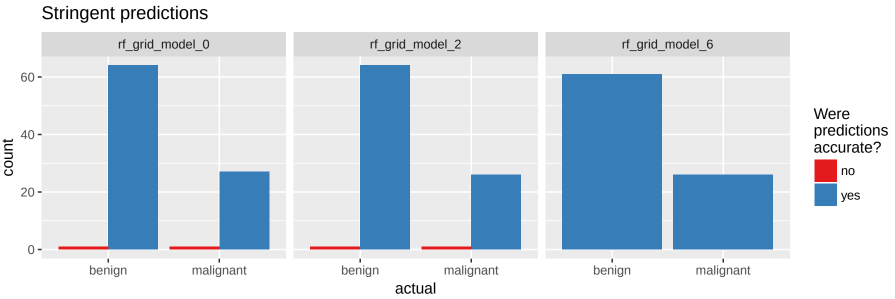
- h2o.getGrid()
- AUC, error, accuracy, etc.

AUC and mean squared error (MSE)



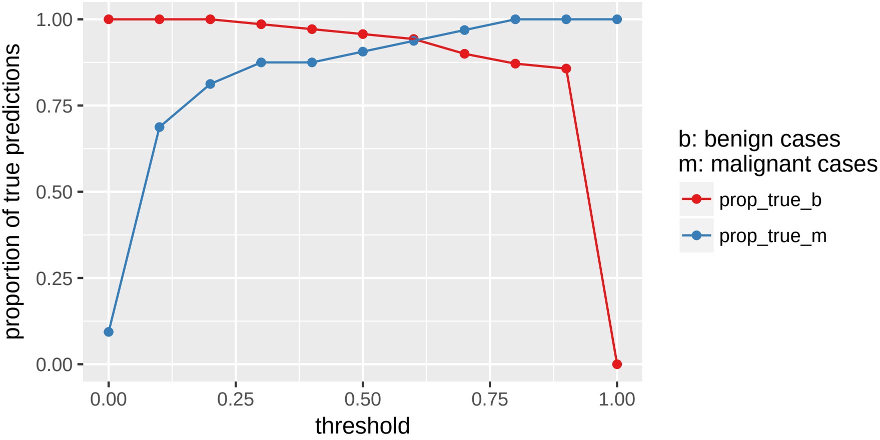
Predictions on test data





Predictions on test data

Choosing a prediction threshold



Take home messages:

- there is no ’one-size-fits-all’ approach to ML

- We want to create meaningful models that we can trust to answer our specific questions!

- know your data well before modeling
- take time to think about pre-processing & features

- test different models & hyper-parameters

- evaluate model performance on independent data
- choose performance measure based on your specific problem
- choose prediction threshold based on your specific problem

Outlook

- ‘Big Data’ needs to be big!
- the more data, the more accurate the models will be

- for really meaningful models, data needs to be shared

- ML could make health care more cost-effective by reducing the energy required for interpretation
- issues: privacy, platform, quality standards

Thank you for your attention!

Questions?

Slides and code will be available on Github:
https://github.com/ShirinG/Webinar_ML_for_disease/share

Code will also be on my website:
https://shiring.github.io

You can contact me via
shirin.glander@wwu.de