The machine-learning perspective

Predicting time series, performance metrics

Nelson Nazzicari

(bioinformatician & ML engineer) **CREA**, Lodi (Italy)

Machine learning in your head

- The first edition of this course gets 10 students
- The second edition gets 20 students
- The third edition gets 40 students
- The fourth edition gets 80 students
- How many students in the sixth edition?

Machine learning in your head

TRAINING DATA

- The first edition of this course gets 10 students
- The second edition gets 20 students
- The third edition gets 40 students
- The fourth edition gets 80 students
- How many students in the sixth edition?

NEW, UNKNOWN DATA

STUDENTS IN SIXTH EDITION = 320

STUD = 10 x 2 exp (YEAR - 1)

PREDICTION

MATHEMATICAL MODEL

A change in perspective

Traditional statistics

1. Assumptions about the data
2. Apply the model
3. How well does the pre-specified relationship fit the data?

Machine learning

1. Very little assumptions about the data
2. Apply the model
3. Obtain predictions (?)

A change in perspective

1. Decide a performance metric
2. Split the data: training set, validation set
3. Tune the model on the training set
4. Predict the validation set, assess the performances

A change of perspective

Topic	Traditional statistics	Machine Learning
Most common usage	Assessing whether relationships can be generalized from the sample to the population	Accurately predicting or classifying future observations
Main goal	Testing whether pre-specified relationships exist in the data	Identifying patterns in the data without pre-conception
Ideally suited for	Deductive hypothesis testing	Abductive hypothesis generation (see Peirce, 1903, as cited in the Peirce Edition Project, 1998, and examples in Sheetal et al., 2020)
Shape of relationships between variables	Fits data onto predefined shapes specified in the statistical model	Learn the true shape of the relationship between variables
Mathematical proofs	The regression line is proven to be the best linear fit	The results are suggestive and cannot be proven to be optimal (Reyzin, 2019)
How to trust the analysis	Standard robustness tests	Test model on unseen (i.e. new) data. Test the generalizability via secondary analysis
Communicating the results to target audience	Standard equations, beta values	Shapley values (e.g. see Mokhtari et al., 2019)

Table from Sheetal et al., 2022 - DOI: 10.1111/apps.12435

A change of perspective

Topic	Traditional statistics	Machine Learning
Researcher skills needed	Training in statistics	Training in data science and programming
Researcher's experience needed	Experience in statistical models	Experience in analyzing diverse datasets
Computational power needed	Generally most modern laptops can do the analysis	Requires high-end computing environment
Model reuse	Need to build different models for each objective	One algorithm can be reused for different objectives
Number of predictors	Limited by multicollinearity. Adding more predictors to the model might break the model	Limited by computational power. Adding more predictors does not break model
Number of observations	Limited by availability; needing to adjust alpha based on number of observations (Maier & Lakens, 2022)	Limited by availability and computational power; more data is generally better
General pattern of results	Low predictability, high explainability (London, 2019)	High predictability, low explainability. Even though advances are continually happening to explain ML models, explainability is limited

Table from Sheetal et al., 2022 - DOI: 10.1111/apps.12435

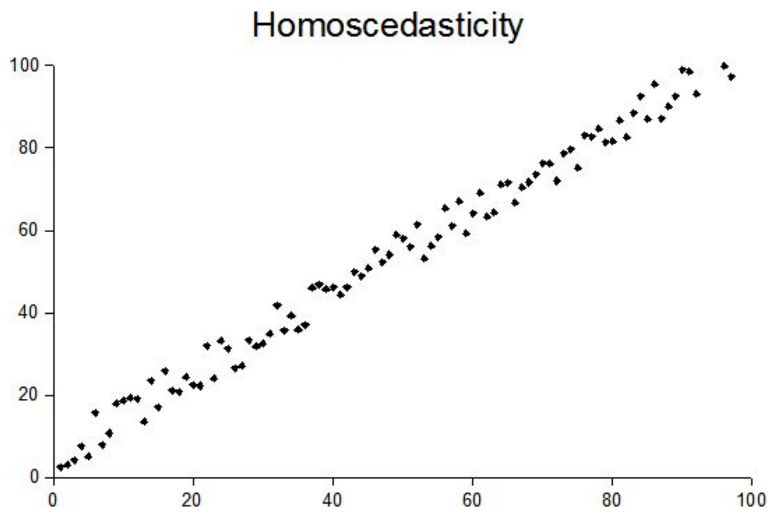
Testing assumptions

Common assumptions on data

- Homoscedasticity
- Multicollinearity (absence of)
- Outliers (absence of)
- Normal (or known) distribution
- Population homogeneity

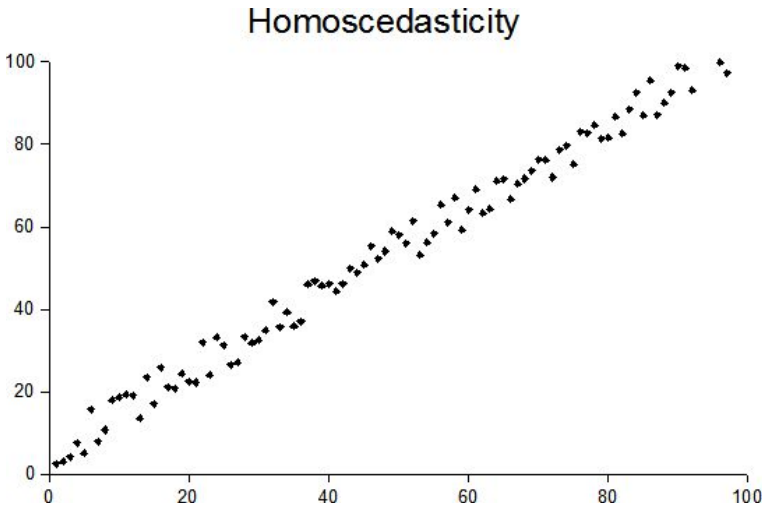
…but always check your model!

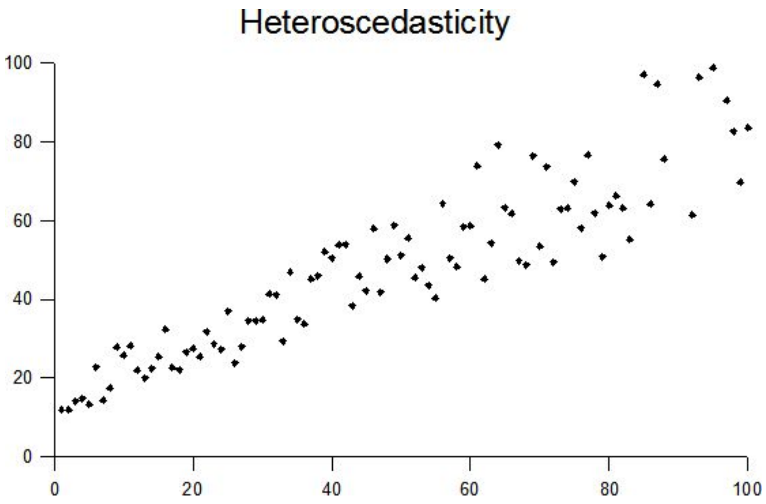
Assumption: Homoscedasticity



$$Y = X + N(0, \sigma^2)$$

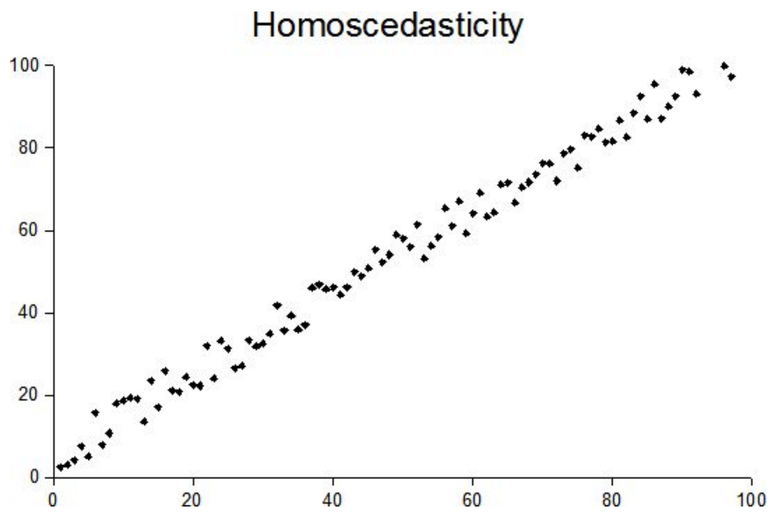
Assumption: Homoscedasticity

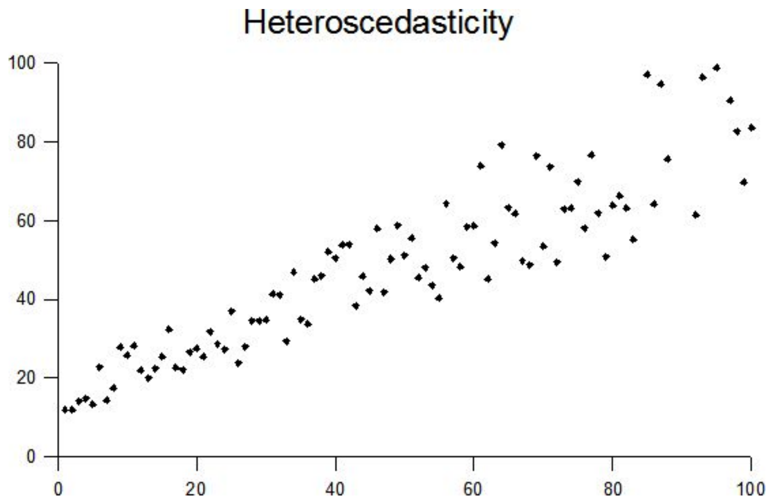




$$Y = X + N(0, \sigma^2)$$

Assumption: Homoscedasticity





$$Y = X + N(0, \sigma^2)$$

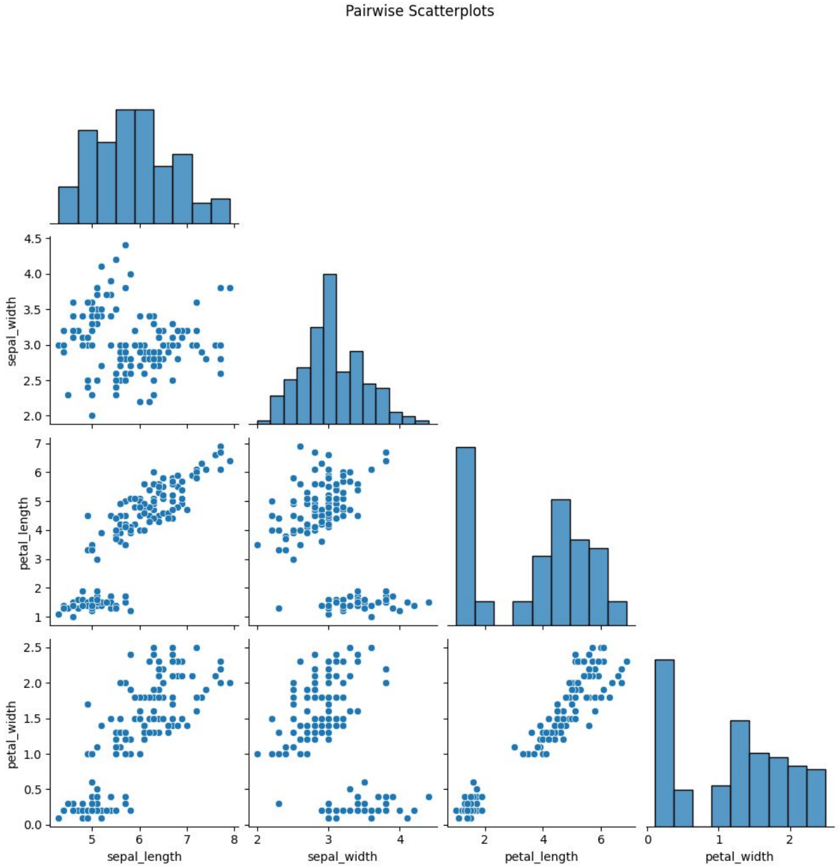
$$Y = X + N(0, f(x))$$

Assumption: no multicollinearity

Def: presence of high correlation between two (or more) independent variables

Assumption: no multicollinearity

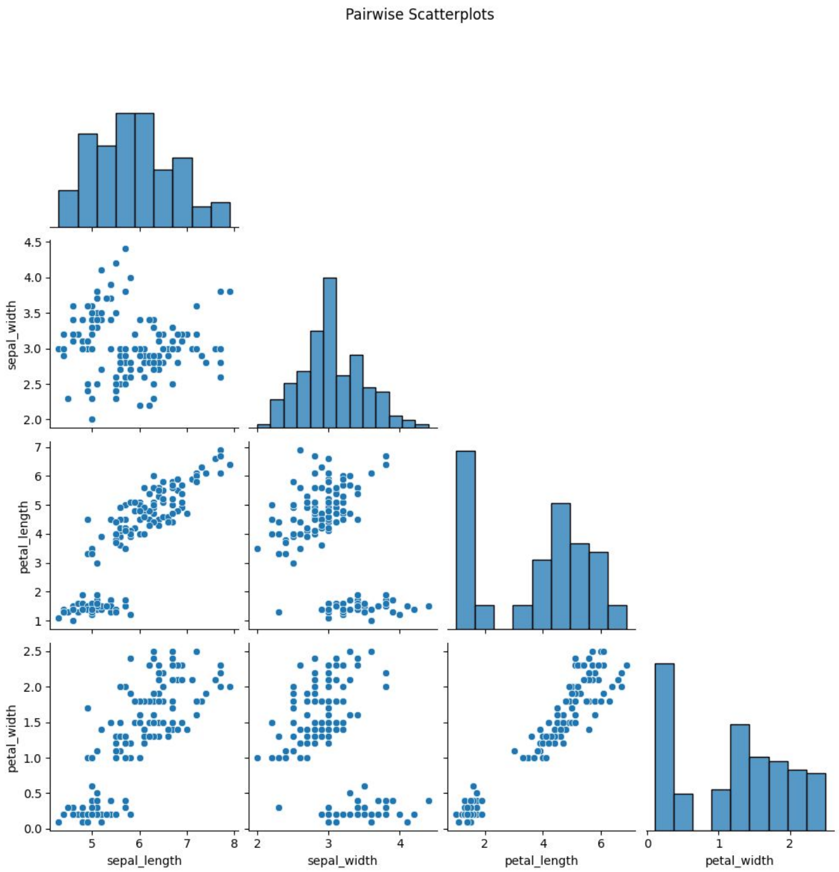
Def: presence of high correlation between two (or more) independent variables



Assumption: no multicollinearity

Def: presence of high correlation between two (or more) independent variables

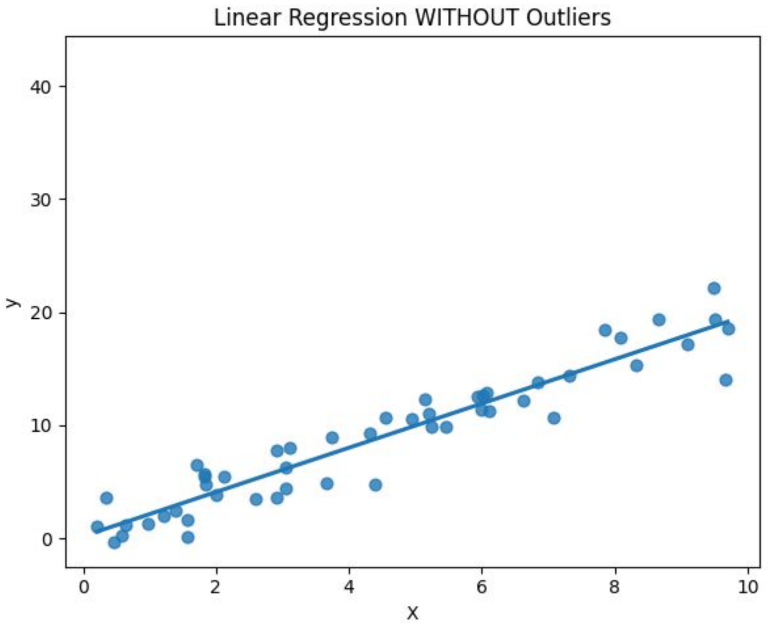
- What variables are truly important?
- Splitting explanatory power among several variables → unstable estimates of regression coefficients
- Large standard errors

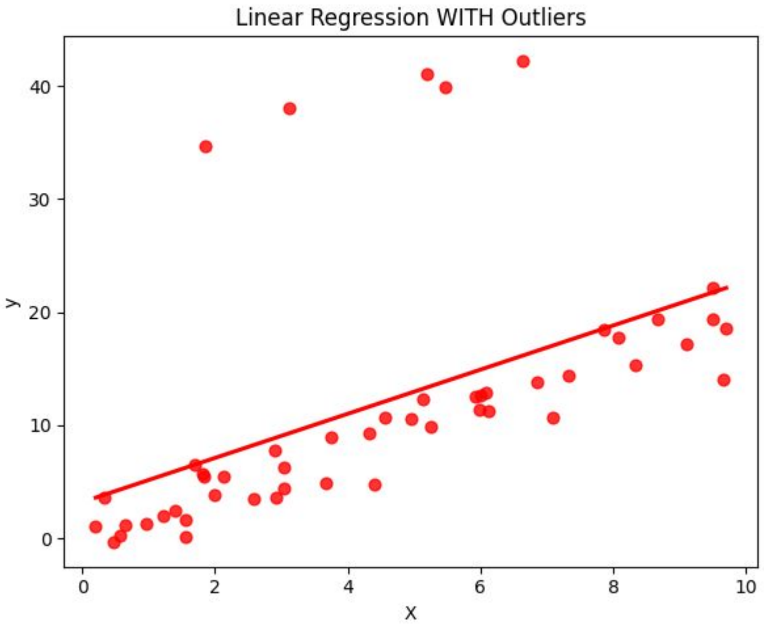


Assumption: no outliers

- Data points that are “very different” from the others
- Errors or the natural shape of the data?
    - (is the sample size big enough?)

- Skew the regression
- Inflate error measure
- Unstable coefficients





Testing assumptions

Common assumptions on data

- Homoscedasticity
  - Breush-Pagan test
- Multicollinearity (absence of)
  - Correlation matrix
  - Variance Inflation Factor (VIF) values (VIF > 5 or 10 is a warning sign)
  - Visual inspection, pair-plot
- Outliers (absence of)
  - Z-score (univariate), Mahalanobis (multivariate)

Testing assumptions

	Assumption about the data	Reference
Linear regression	Homoskedasticity, lack of multi-collinearity, and independently, identically, and normally distributed errors	Ezekiel (1925)
Logistic regression	Homoskedasticity, lack of multi-collinearity, no outliers, linear relationships in the logit metric	Stoltzfus (2011)
K-nearest neighbor	Independence of observations; similar observations are closer to each other in a measurable distance space	Mack and Rosenblatt (1979)
Support vector machines	Clear boundaries between groups, relatively small datasets	Tong et al. (2009)
Decision trees	Continuous variables can be discretized into meaningful buckets	Brodley and Utgoff (1995)
Naive Bayes	Independence of predictors	Lewis (1998)
LASSO	Sparsity (only a few predictors are relevant), irrelevant and relevant predictors are uncorrelated	Tibshirani (1996)
Random forest	No missing data, requires hyperparameter search	Breiman (2001)
Neural networks	No missing data, requires hyperparameter search	LeCun et al. (2015)
Gradient boosting	No formal assumptions, requires hyperparameter search	Friedman (2002)

Table from Sheetal et al., 2022 - DOI: 10.1111/apps.12435

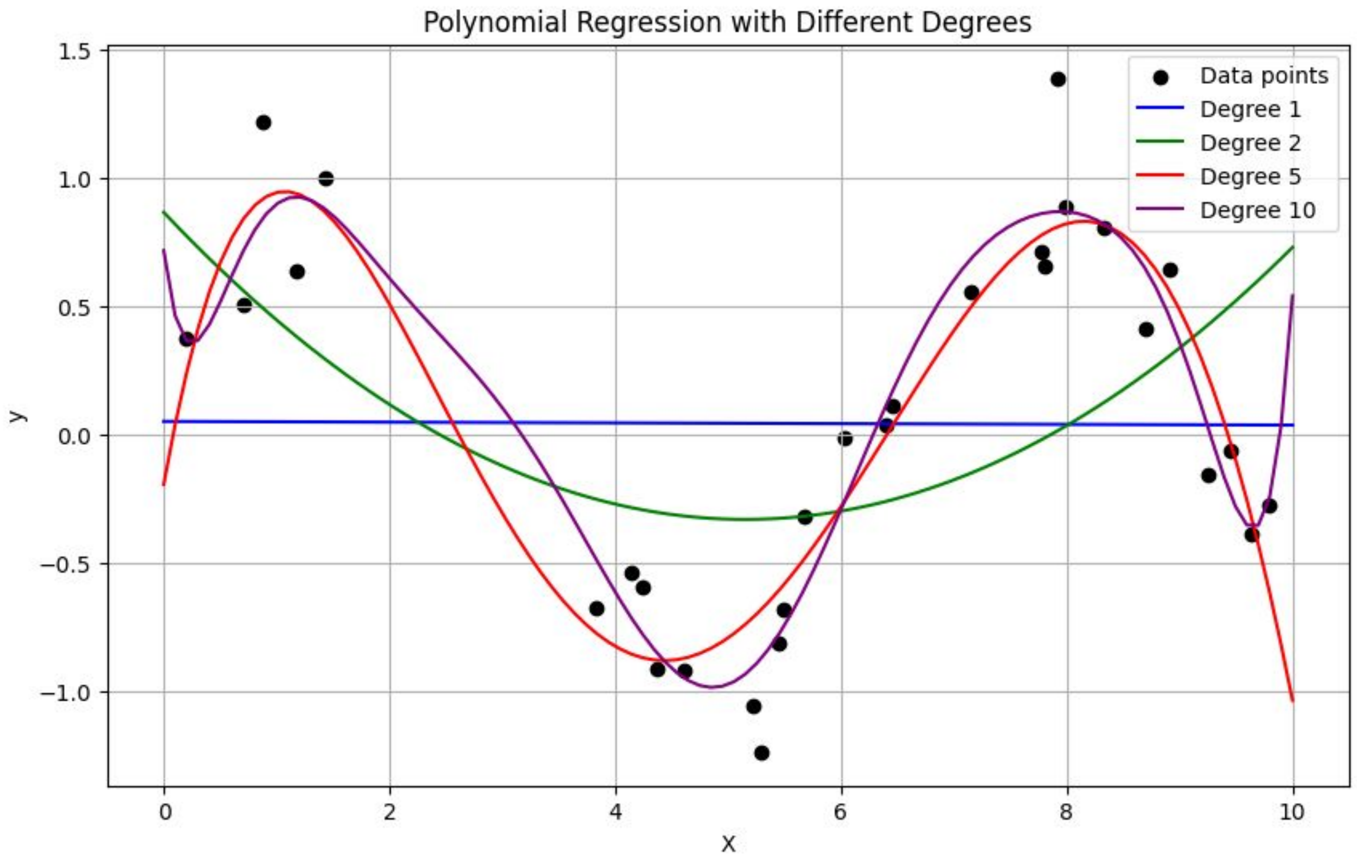
Notebook on assumptions

Parameters vs. Hyperparameters

- Machine learning models (often) ask for hyperparameters selection
- “A priori” choice
- Once fixed, the model will optimize
- Often linked to model “power”

Similar choice can be required with traditional models, too…

Parameters vs. Hyperparameters



Parameters vs. Hyperparameters

- Neural networks
  - Number of layers, number of nodes for each layer, dropout rate, presence (and placement) of convolutionary nodes, residual nodes, …,
- Random forest
  - Number of trees, features and samples per split, tree depth and number of leaves…

A change in perspective, reprise

1. Decide a performance metric
2. Split the data: training set, validation set, test set
3. Decide a list of hyperparameter combinations to try
4. Tune the models on the training set
5. Predict the validation set, assess the performances, find the best hyperparameter values
6. Predict the test set, assess real world performances

Measuring the Performances

Regression

- RMSE
- Pearson’s correlation
- Spearman's correlation

Classification

- Accuracy
- Confusion matrix
- F1 score

Measuring the Performances

Regression

- RMSE (same scale as the data)
- Pearson’s correlation (you don’t care for exact values)
- Spearman's correlation (you care about ranking)

Classification

- Accuracy (easy to understand, problematic with unbalanced classes)
- Confusion matrix (not a true metric, useful for the general feeling)
- F1 score (balancing classes, native for binary classification, adaptable to multiclass)

Measuring the Performances

Metric	MSE		Norm. MSE		MAE		R^2	
	train	test	train	test	train	test	train	test
LRR	2.345	2.307	0.086	0.084	1.002	1.014	0.914	0.916
KRR*	1.462	1.635	0.053	0.059	0.671	0.745	0.947	0.941
FFNN	1.433	1.496	0.052	0.054	0.653	0.672	0.948	0.946
ESN	0.265	0.259	0.009	0.009	0.172	0.173	0.991	0.991
LSTM	0.083	0.086	0.003	0.003	0.092	0.097	0.997	0.997

Fig. 6 shows plots of the true and predicted conversion rates of the differ

https://arxiv.org/abs/2002.01768

Cornell University

arXiv > cs > arXiv:2002.01768

Computer Science > Machine Learning

[Submitted on 5 Feb 2020 (v1), last revised 20 Oct 2020 (this version, v2)]

Forecasting Industrial Aging Processes with Machine Learning Methods

Mihail Bogojeski, Simeon Sauer, Franziska Horn, Klaus-Robert Müller

Accurately predicting industrial aging processes makes it possible to schedule maintenance events further in advance, ensuring a cost-efficient a
mechanistic or simple empirical prediction models. In this paper, we evaluate a wider range of data-driven models, comparing some traditional st
recurrent neural networks (echo state networks and LSTMs). We first examine how much historical data is needed to train each of the models on

Notebook on performances

Measuring the Performances

- You get RMSE = 2.7
- Is it good? Bad?
- Is your job done?

Comparing the Performances

- Top tier: Bayes error rate
  - The best possible error a model could achieve
  - By definition, you'll do worse. By how much?
  - Always unknowable

Comparing the Performances

- Top tier: Bayes error rate
  - The best possible error a model could achieve
  - By definition, you’ll do worse. By how much?
  - Always unknowable
- Mid tier: Panel of experts
  - The best practical approximation of Bayes error rate
  - Expensive

Comparing the Performances

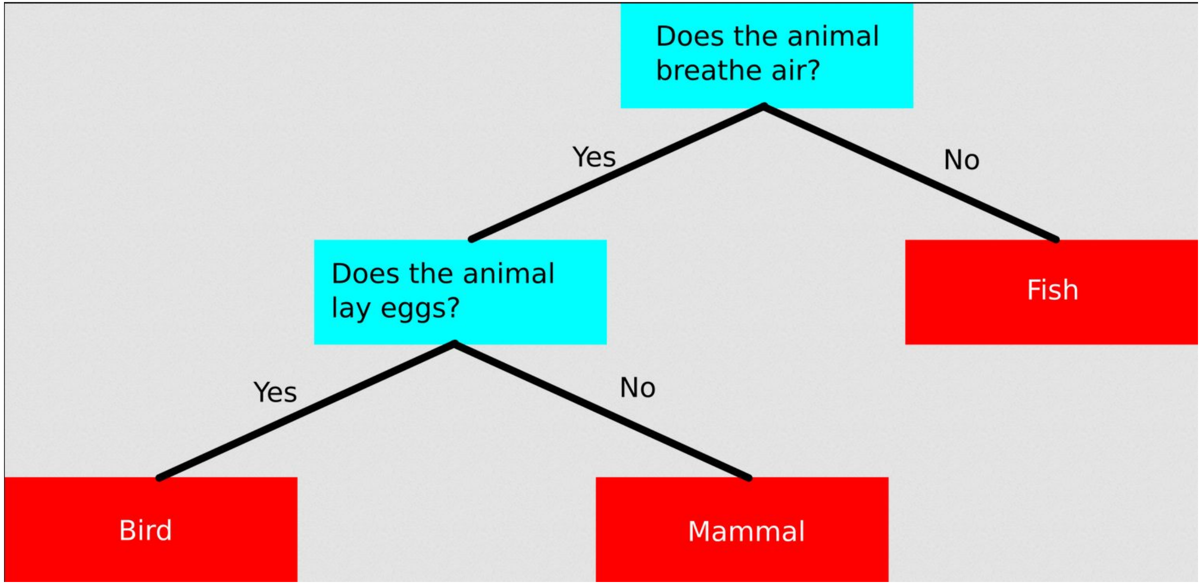
- Top tier: Bayes error rate
  - The best possible error a model could achieve
  - By definition, you'll do worse. By how much?
  - Always unknowable
- Mid tier: Panel of experts
  - The best practical approximation of Bayes error rate
  - Expensive
- Realistic tier: state of the art + naive model
  - Compare your results to other papers
  - Compare your results to a simple model, show the improvement

A change in perspective, reprise #2

1. Decide a performance metric and a target for comparison
2. Split the data: training set, validation set, test set
3. Decide a list of hyperparameter combinations to try
4. Tune the models on the training set
5. Predict the validation set, assess the performances, find the best hyperparameter values
6. Predict the test set, assess real world performances
7. Compare with the declared target

Case study: Adapting RF to longitudinal data

Case study: Adapting RF to longitudinal data



- Single decision tree:
  - very sensible to parameters (e.g. tree depth)
  - Can easily overfit

Case study: Adapting RF to longitudinal data

What is Random Forest?

- ensemble of decision trees
- wisdom of the crowds
- diversity is desired

Case study: Adapting RF to longitudinal data

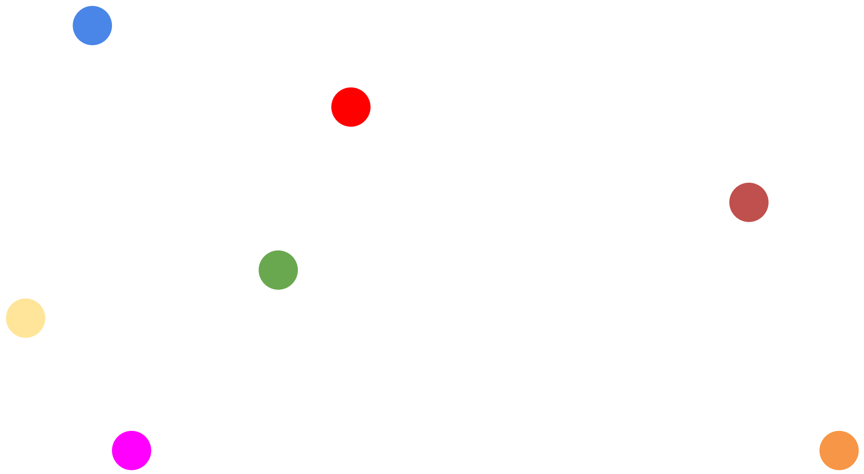
What is Random Forest?

- ensemble of decision trees
- wisdom of the crowds
- diversity is desired

At a 1906 country fair in Plymouth, 800 people participated in a contest to estimate the weight of a slaughtered and dressed ox. Statistician Francis Galton observed that the median guess, 1207 pounds, was accurate within 1% of the true weight of 1198 pounds.

Case study: Adapting RF to longitudinal data

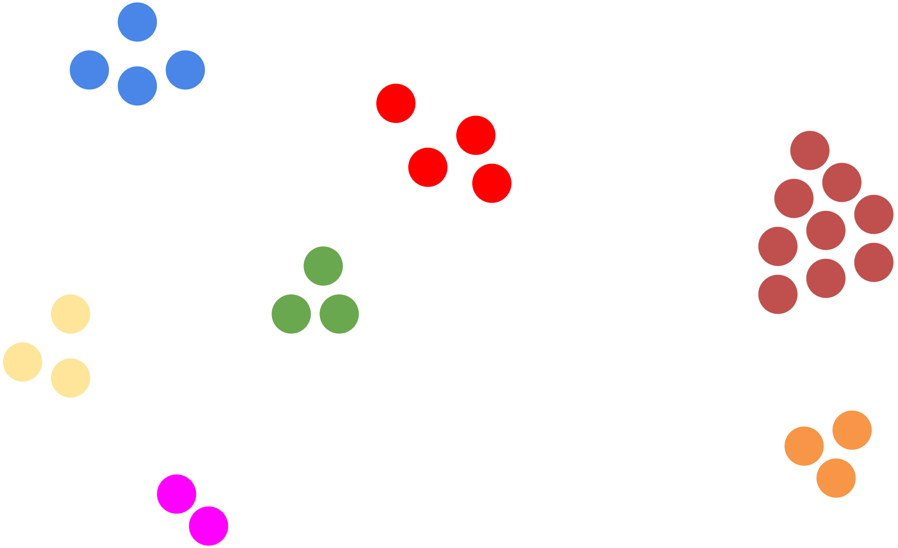
Bootstrap, naive case



Hyperparameters
Trees: 4
Samples per tree: 3

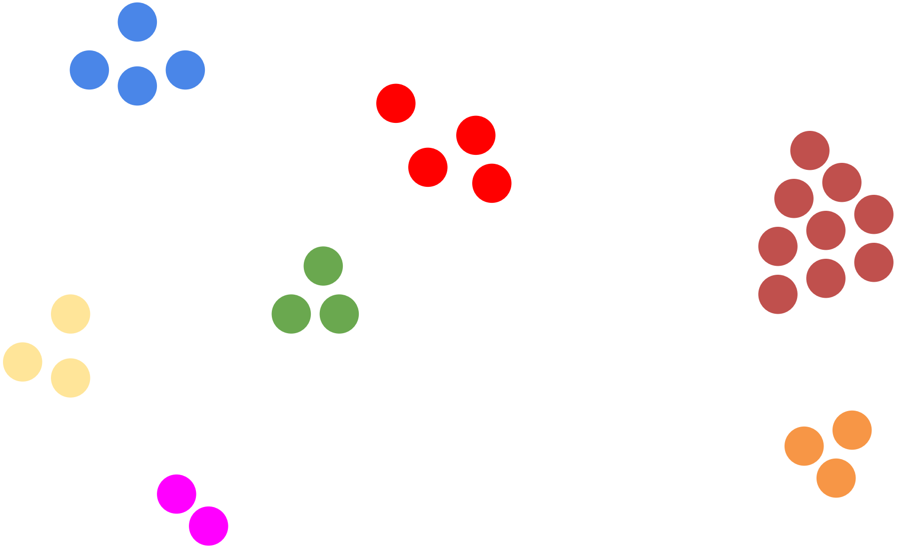
Case study: Adapting RF to longitudinal data

Complication: multiple measures per subject (no time, yet)



Case study: Adapting RF to longitudinal data

Complication: multiple measures per subject (no time, yet)

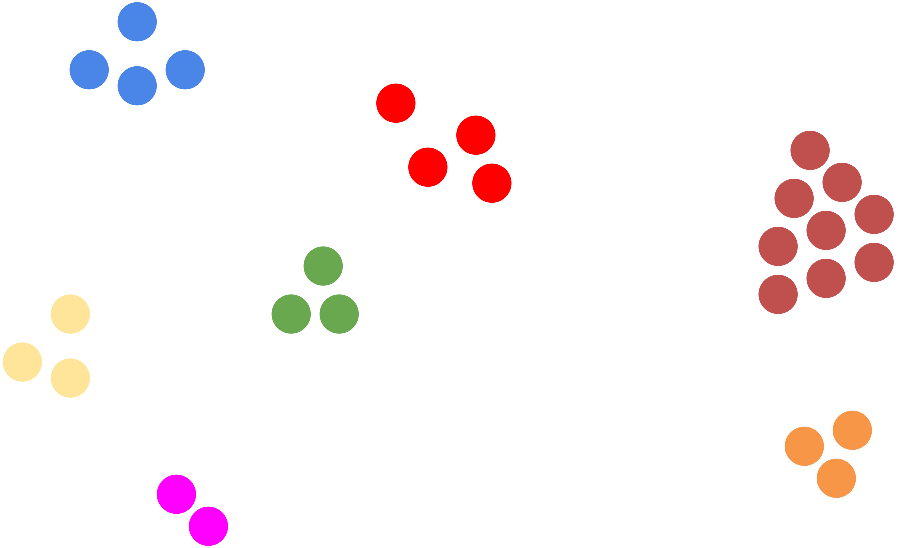


Solution 1: average over measures

- Simple, immediate
- Loss of information
- Masks unbalancedness

Case study: Adapting RF to longitudinal data

Complication: multiple measures per subject (no time, yet)



Solution 2: adapt bootstrap

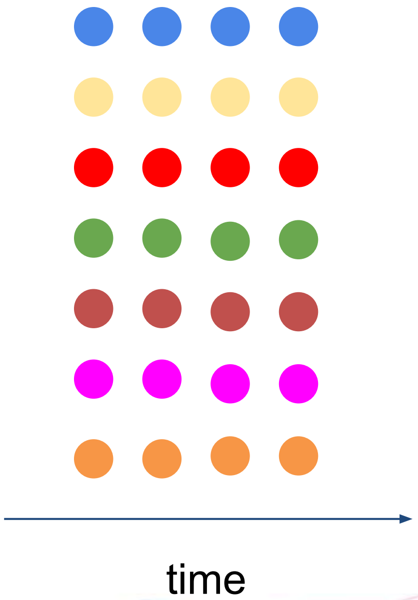
- RF++
- Shown to improve on standard RF

Case study: Adapting RF to longitudinal data

Notebook: RF++

Case study: Adapting RF to longitudinal data

Complications: multiple measures per subject, time is relevant



Case study: Adapting RF to longitudinal data

Notebook: Historical RF

Case study: Adapting RF to longitudinal data

- MERF: Mixed Effects Random Forest
  - Random Forest to model fixed effects (the global part)
  - linear random effect model for subject-specific deviations (random intercepts and possibly slopes)

$$y_{ij}=f(X_{ij})+Z_{ij}b_i+\epsilon_{ij}$$

y_{ij}: observation for subject i at time j

X_{ij}: covariates for the random forest

Z_{ij}: covariates for the random effects (often just a 1 for random intercepts)

b_i~N(0,D): random effects (subject-specific)

ϵ_{ij}~N(0,σ^2): noise

f(·): is the (unknown) function approximated by a Random Forest

Initialize b
While not convergent:
    Estimate RF with current b, predict $f^*(X_{ij})$
    Fit the linear model, obtain new b

Various links

- Random survival forests for dynamic predictions of a time-to-event outcome using a longitudinal biomarker https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/s12874-021-01375-x
- FRET forecasting algorithm
- Time-series forecasting through recurrent topology https://www.nature.com/articles/s44172-023-00142-8
- Machine and deep learning for longitudinal biomedical data: a review of methods and applications https://link.springer.com/article/10.1007/s10462-023-10561-w
- Statistical Learning Methods for Longitudinal High-dimensional Data https://pmc.ncbi.nlm.nih.gov/articles/PMC4181610/
- Using machine learning to analyze longitudinal data: A tutorial guide and best-practice recommendations for social science researchers https://iaap-journals.onlinelibrary.wiley.com/doi/10.1111/apps.12435
- A review on longitudinal data analysis with random forest https://academic.oup.com/bib/article/24/2/bbad002/6991123?login=false