Machine Learning Concepts

1.3

Quiz

Exam Date

Add Exam Info Here

Topics from last week

- Overfitting
- Metrics to measure performance: RMSE, Confusion Matrix
- Different ways to divide training and test set
- Cross-validation:
  - Probabilistic
  - systematic
- Train and Test error
- Complexity
- Bias and variance

Last week summary

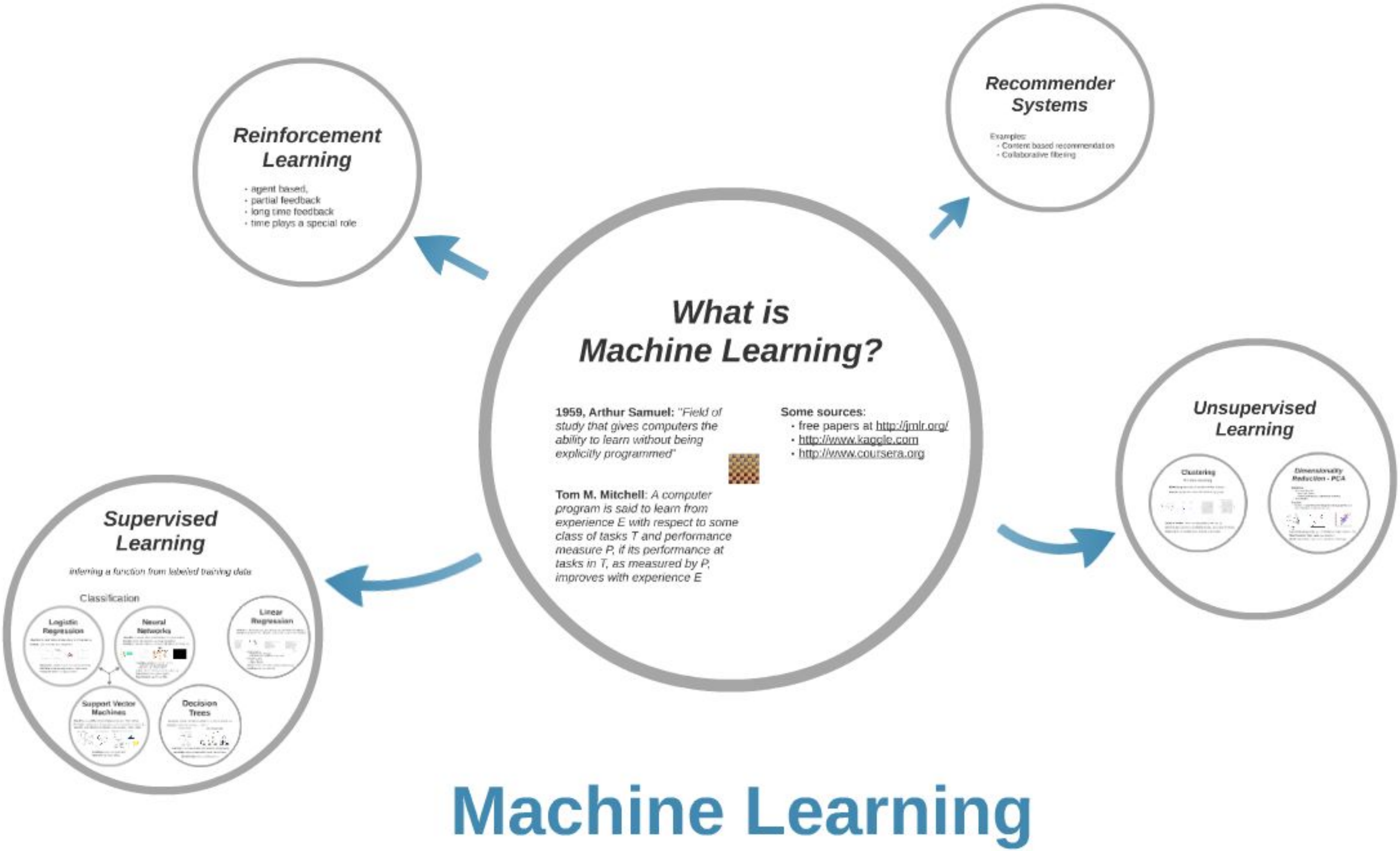
We have reviewed some general concepts in Supervised Learning:

- Measuring **prediction power** of regression and classification models
- Separating **Training** and **Test Error**
- Separating data into **Training**, **Validation** and **Test** sets
- A number of **Cross-validation Techniques**
- Model complexity consequences: **Bias vs. Variance**
- **Model selection** criterias
- **Benchmark** Model selection

We prepared for the discussion of a number of Machine Learning Models

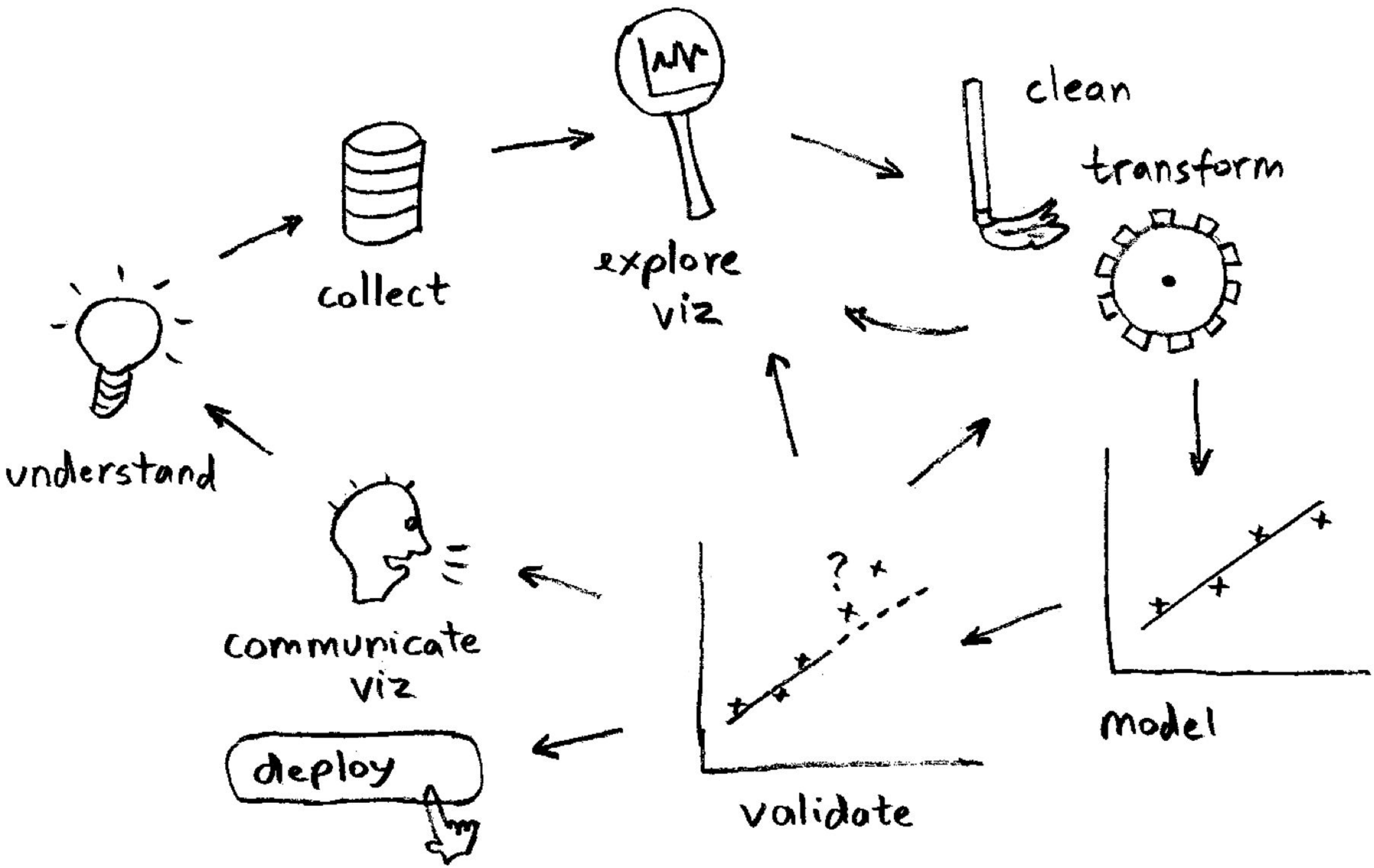
- **Manual Models** to familiarize with the concept of learning
- **Linear Model:** assumptions, advantage, feature transformations
- **Nearest Neighbour:** algorithm, advantages, difficulties, complexity

https://prezi.com/06swcwazd0ai/machine-learning/



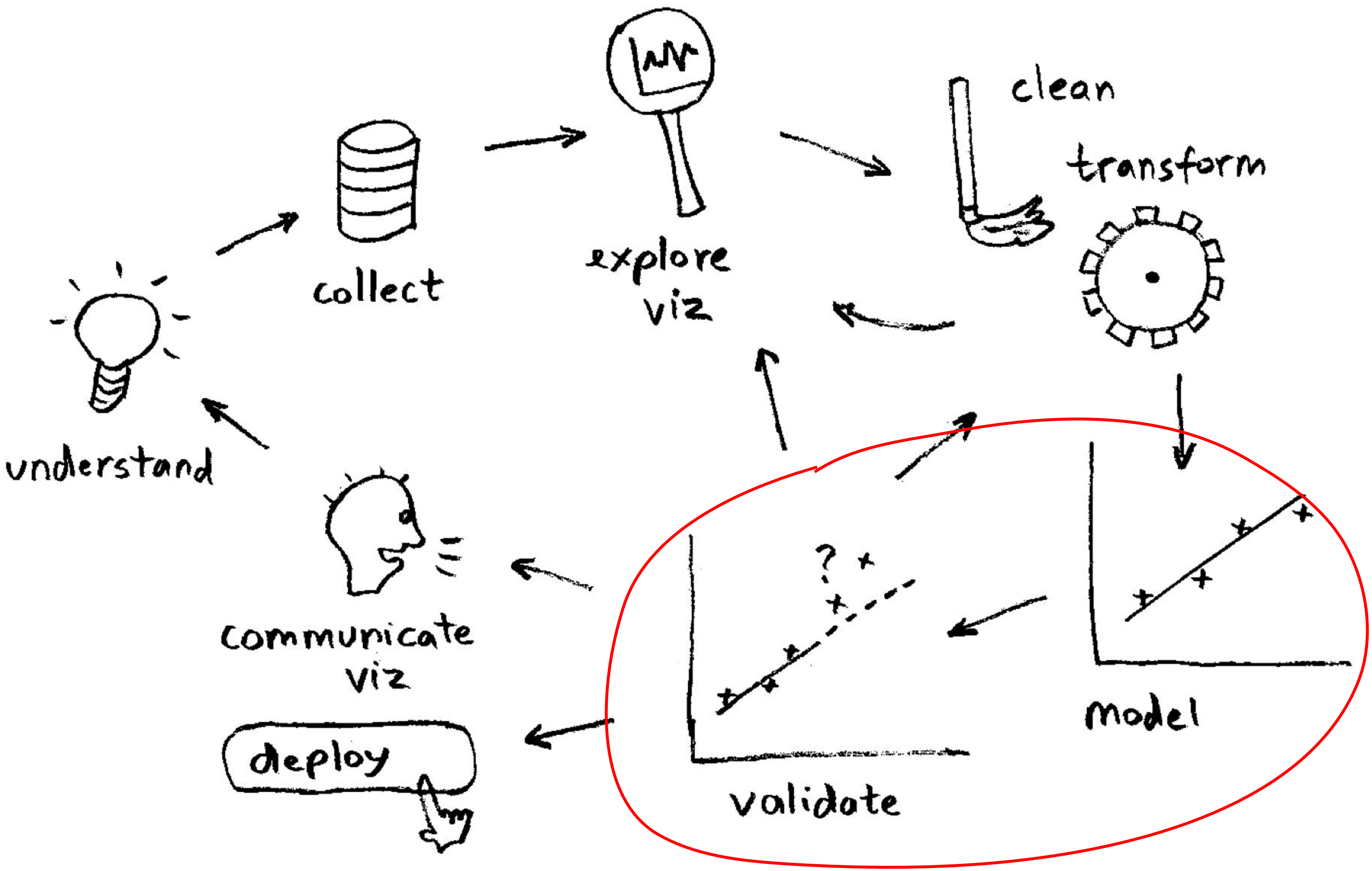
Supervised Learning?

Learning by Example



clean
transform
collect
explore
viz
understand
communicate
viz
deploy
model
validate

Regularized Linear Models



collect
explore viz
clean
transform
understand
communicate viz
deploy
model
validate

Linear Regression Model - output

```
Call:
lm(formula = count ~ season + weekday , data = bikeTrain)

Residuals:
    Min       1Q   Median       3Q      Max
-184.13  -97.84  -28.24   62.80  507.68

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  76.6538     5.6346  13.604    <2e-16 ***
season       27.1605     1.5824  17.164   <2e-16 ***
weekday      -0.1629     0.8768  -0.186     0.853
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 129.9 on 5419 degrees of freedom
Multiple R-squared:  0.05159 ,   Adjusted R-squared:  0.05124
F-statistic: 147.4 on 2 and 5419 DF,  p-value: < 2.2e-16
```

R^2: explained variance

$$R^2 = 1 - \frac{\sum(Y_{actual} - Y_{predicted})^2}{\sum(Y_{actual} - Y_{mean}))^2}$$

$$R^2_{adjusted} = 1 - \frac{(1 - R^2) * (N - 1)}{N - p - 1}$$

p = Number of predictors

N = total sample size

Regularization

$$Y = \theta_1 X_1 + \theta_2 X_2 + \ldots \theta_n X_n$$

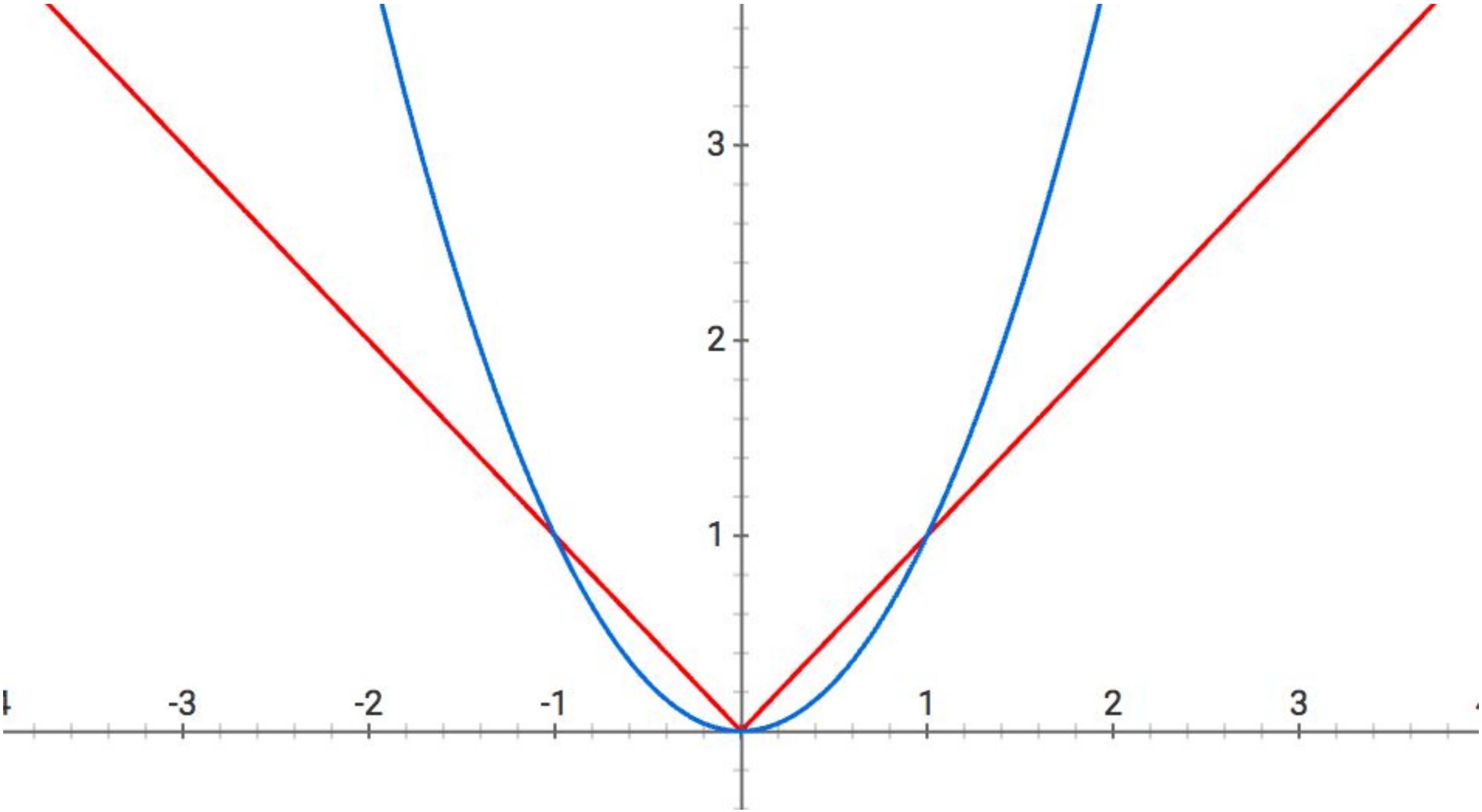
1. Lasso $\min\left(||Y - X\theta||_2^2 + \lambda||\theta||_1\right)$
2. Ridge $\min\left(||Y - X(\theta)||_2^2 + \lambda||\theta||_2^2\right)$
3. Elastic Net $\min\left(||Y - X\theta||_2^2 + \lambda_1||\theta||_1 + \lambda_2||\theta||_2^2\right)$

Lambda_1 = lambda * alpha

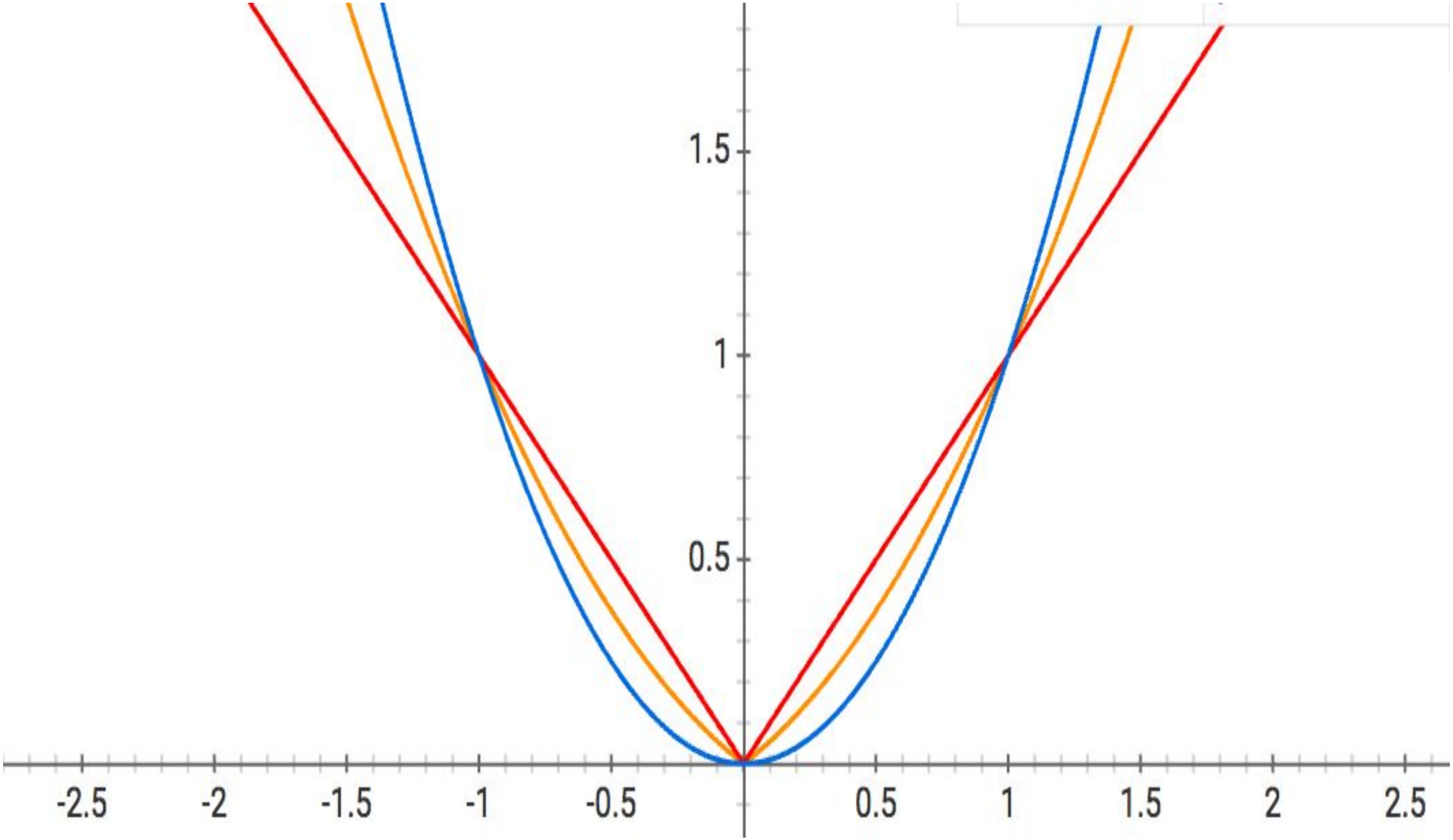
Lambda_2 = lambda * (1 - alpha)

Alpha is between 0 and 1

Lasso vs Ridge Regression



Lasso vs Ridge vs ElasticNet Regression

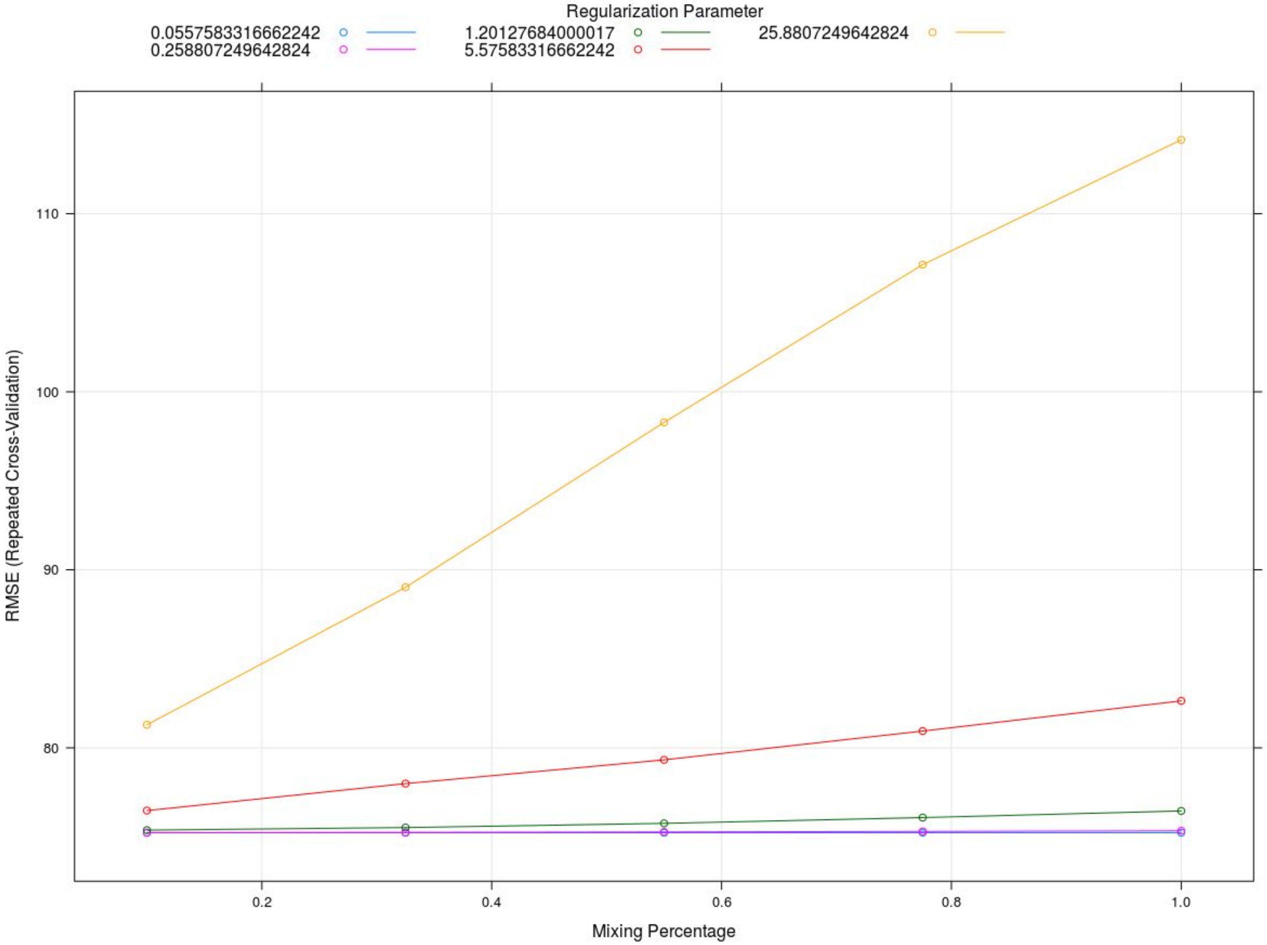


Coding Exercise: Improve lm results

Exercise 1: build a linear model for the Bike Sharing Demand data.
Start with: `ml.1.3/lect/1_regression_tools.R`
Which is better, Plain LM, Lasso, Ridge or ElasticNet?
Does CV help? And RepeatedCV?

Conclusion 0: introducing caret (glmnet)

```
trctrl <- trainControl(method = "cv")
lmElasticNetCaret <- train(
    count~quarter+temp+atemp+weather+hour+holiday+workingday+windspeed,
    data = bikeTrain,
    method = "glmnet",
    trControl=trctrl,
    tuneLength=5
)
```



Further Conclusions

1. Linearly non-separable tasks can benefit from introducing factors
2. Linearly non-separable tasks can benefit from introducing higher level components (poly)
3. Cross validation with glmnet can find the right balance in ElasticNet

Linear Regression Model: assumptions

Assumptions:

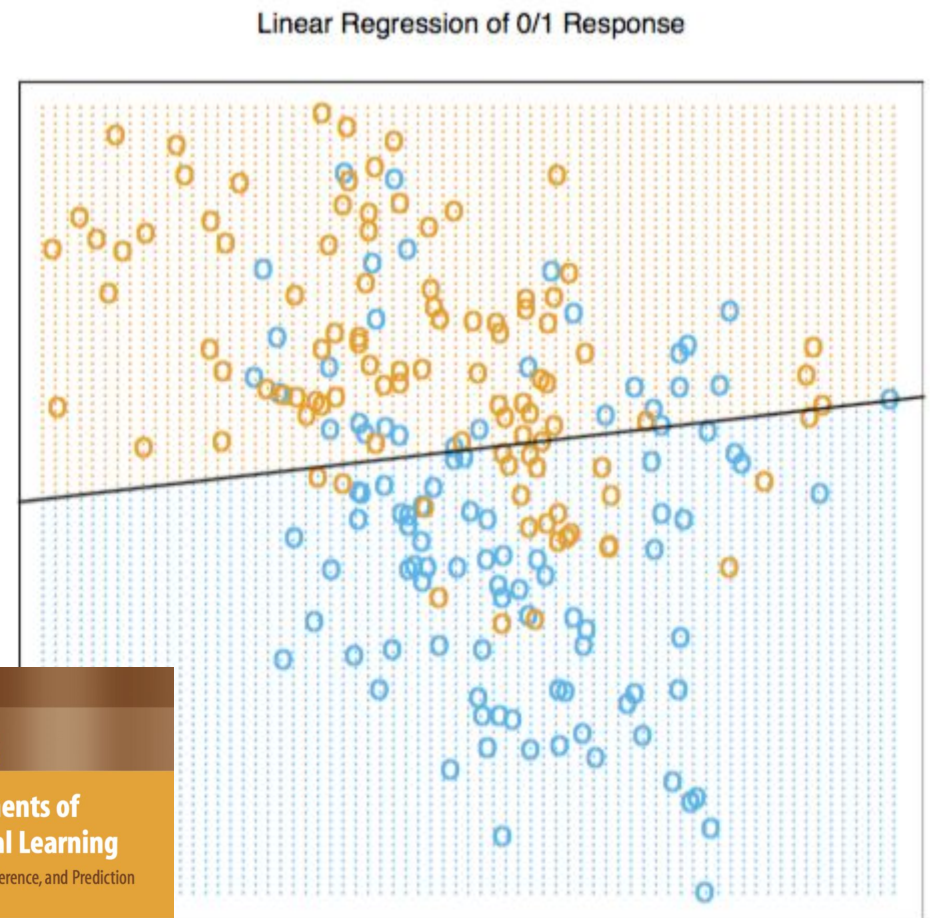
- Linear Separability
- Normal Distribution of errors
- etc.

Workaround:

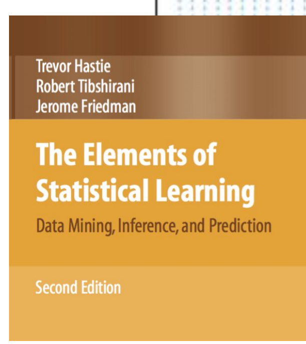
- Feature engineering
- Feature transformation: log, exp, pow
- Poly

Advantage:

- Easy and quick to fit
- Easy to understand



source:



Decision Tree

Decision Trees: How does it work?

Steps:

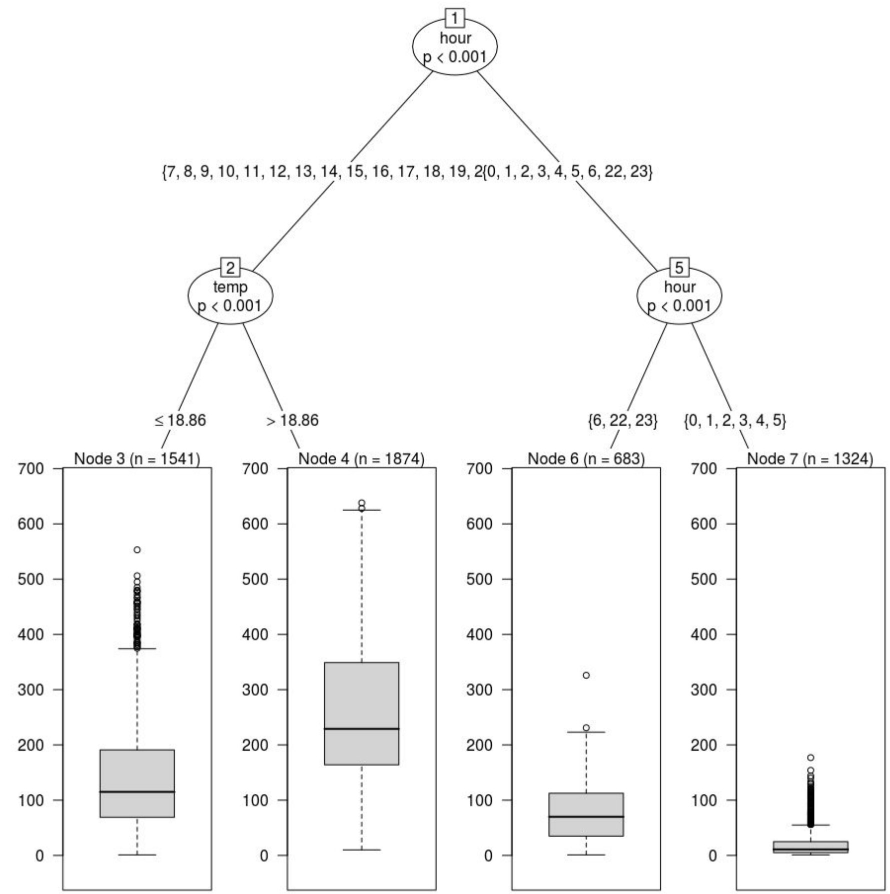
- Take a set of training data
- Look at the available features
- Find the feature and a split defined on this feature that maximizes information gain (reduces the most entropy)
- Repeat this step on the child nodes

Advantage?

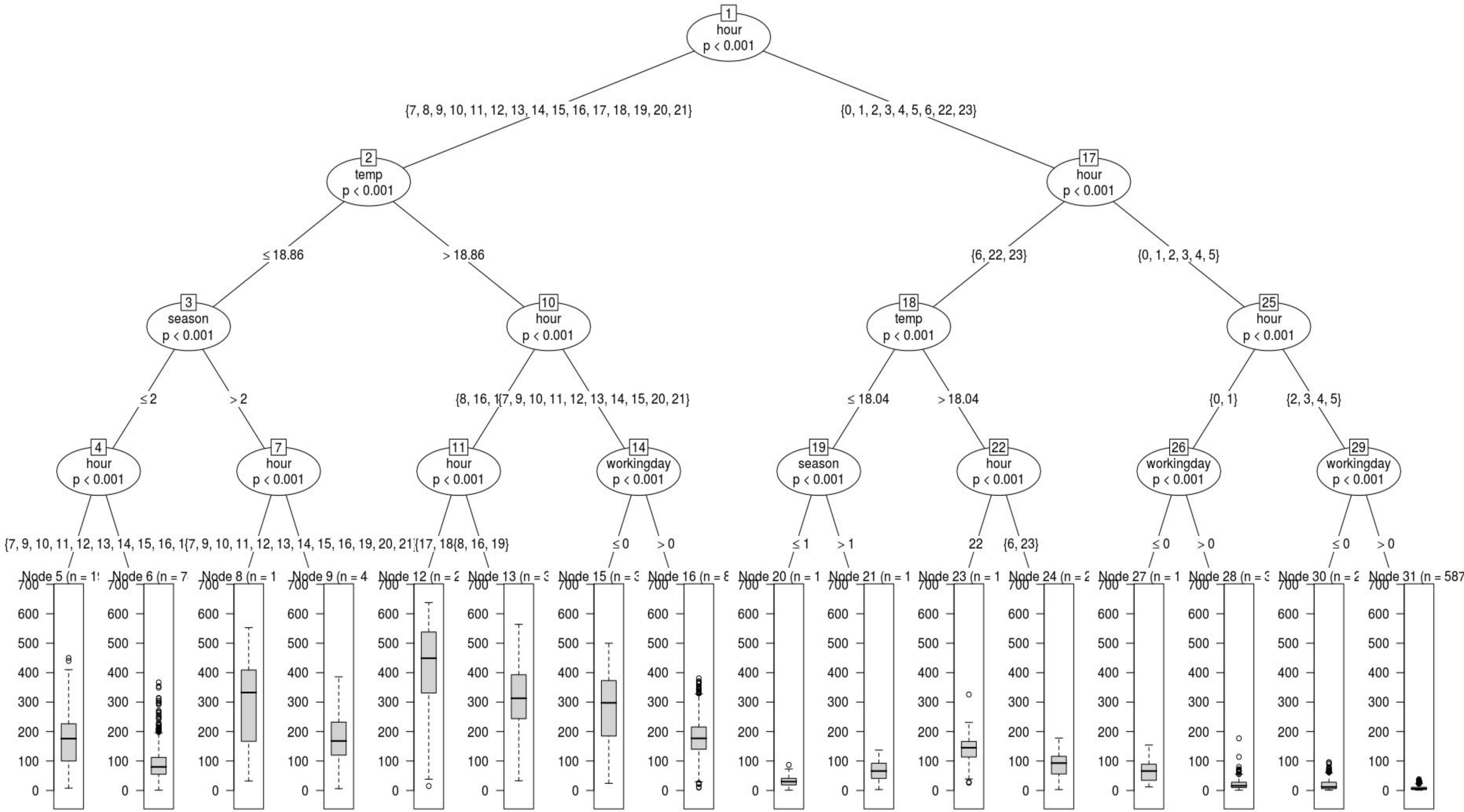
Easy to understand.

Difficulties?

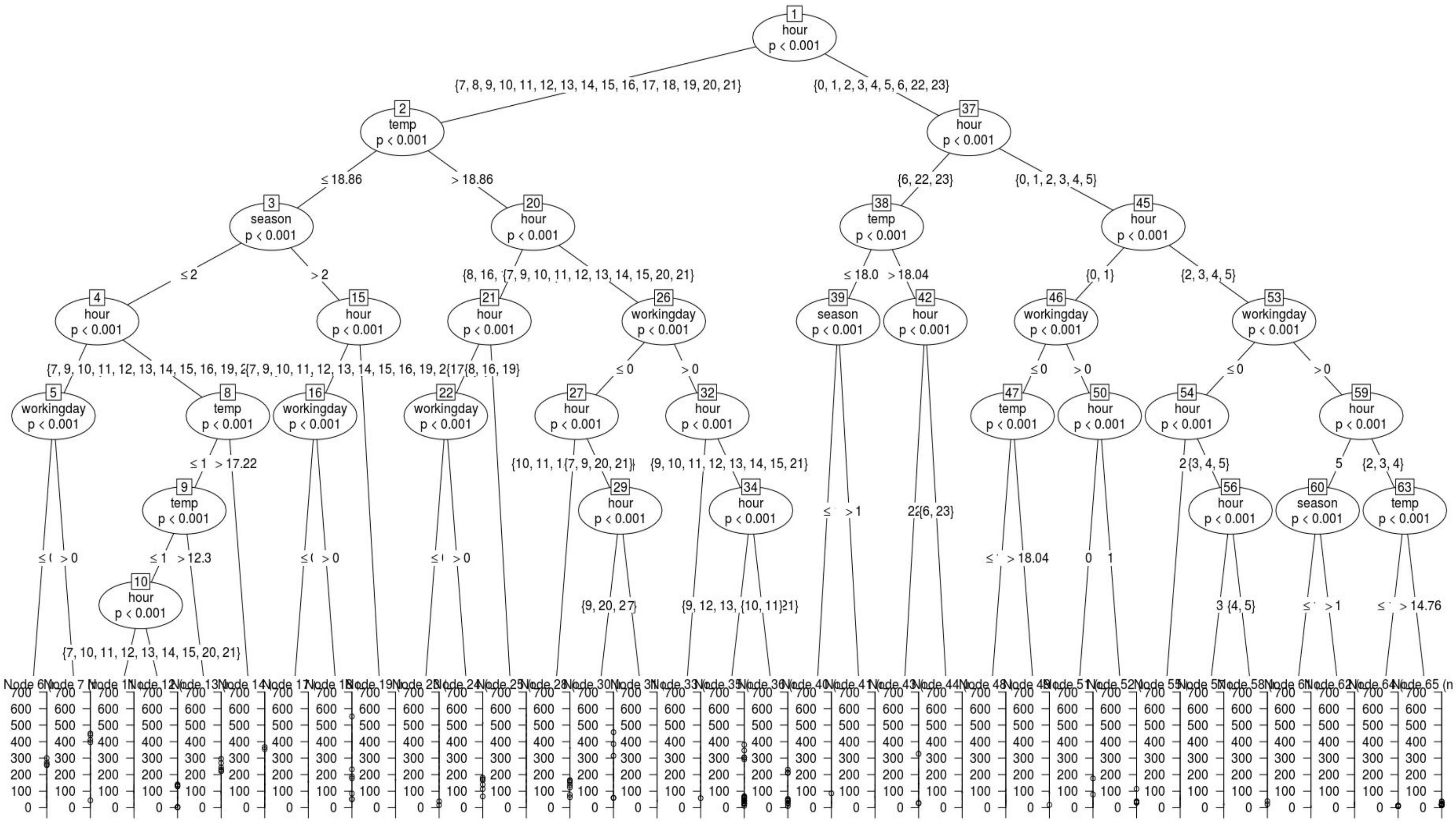
Easy to overfit, needs control parameters.



Decision Tree: how do they work?



Decision Tree: mincriterion=0.99..



Trees: model **complexity control**

A few examples:

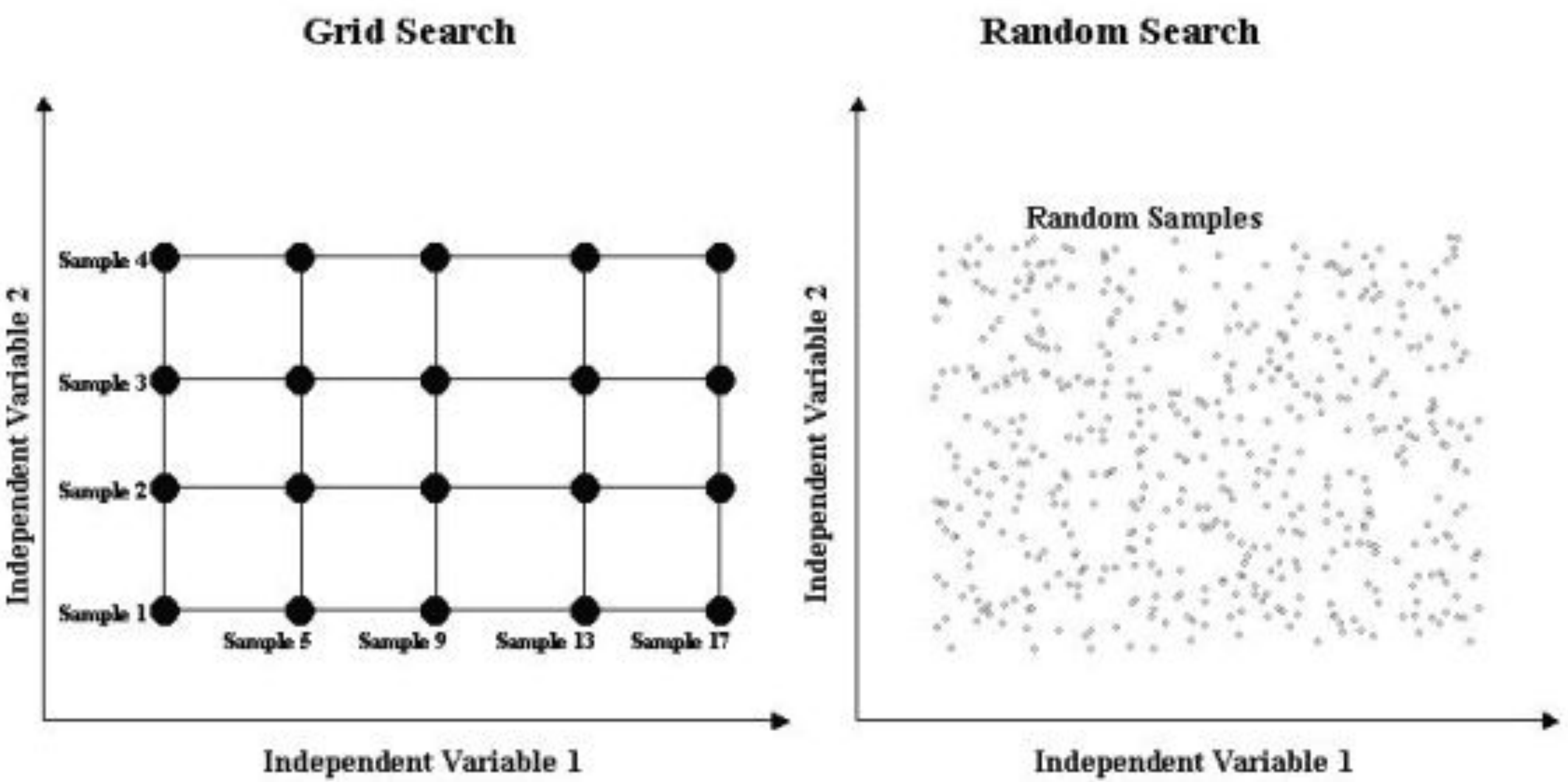
- **Maxdepth:** maximum depth of the tree. The default maxdepth = 0 means that no restrictions are applied to tree sizes.
- **Minsplit:** the minimum sum of weights in a node in order to be considered for splitting.
- **Minbucket:** the minimum sum of weights in a terminal node.
- **Mincriterion**: the value of the test statistic (for testtype == "Teststatistic"), or 1 - p-value (for other values of testtype) that must be exceeded in order to implement a split.
- Etc. (see ctree documentation)

Coding Exercise: Improve Tree results

Exercise 3: build a tree model for the Bike Sharing Demand data.
Start with: `ml.1.3/lect/1_regression_tools.R`
Cross validation helps?

Hyperparameter search

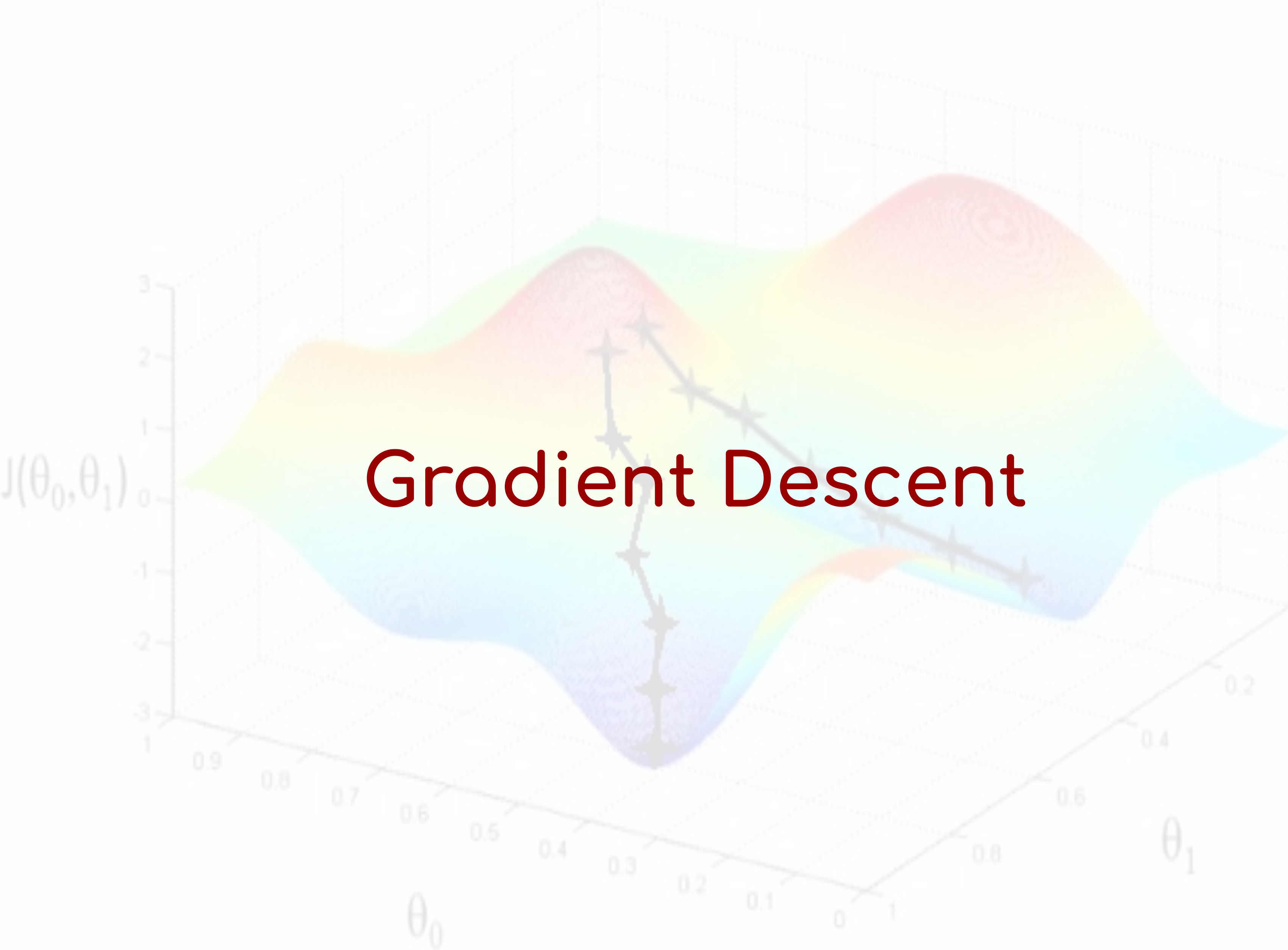
Grid Search vs Random Search



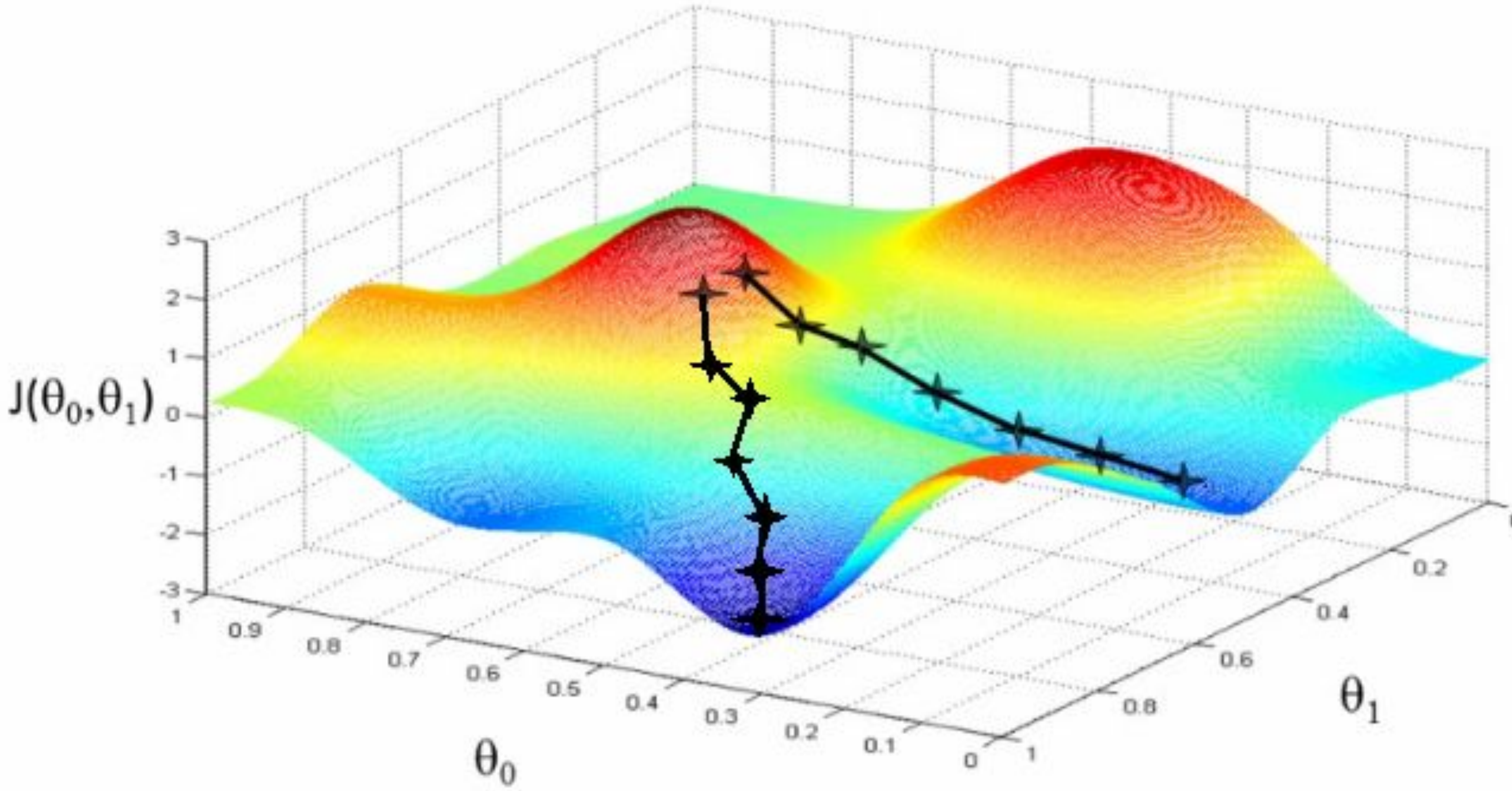
Recap

1. Nn
    a. Complexity
2. Test, Validation, Train (cross validation)
3. Bias and Variance
4. Lasso, Ridge and ElasticNet

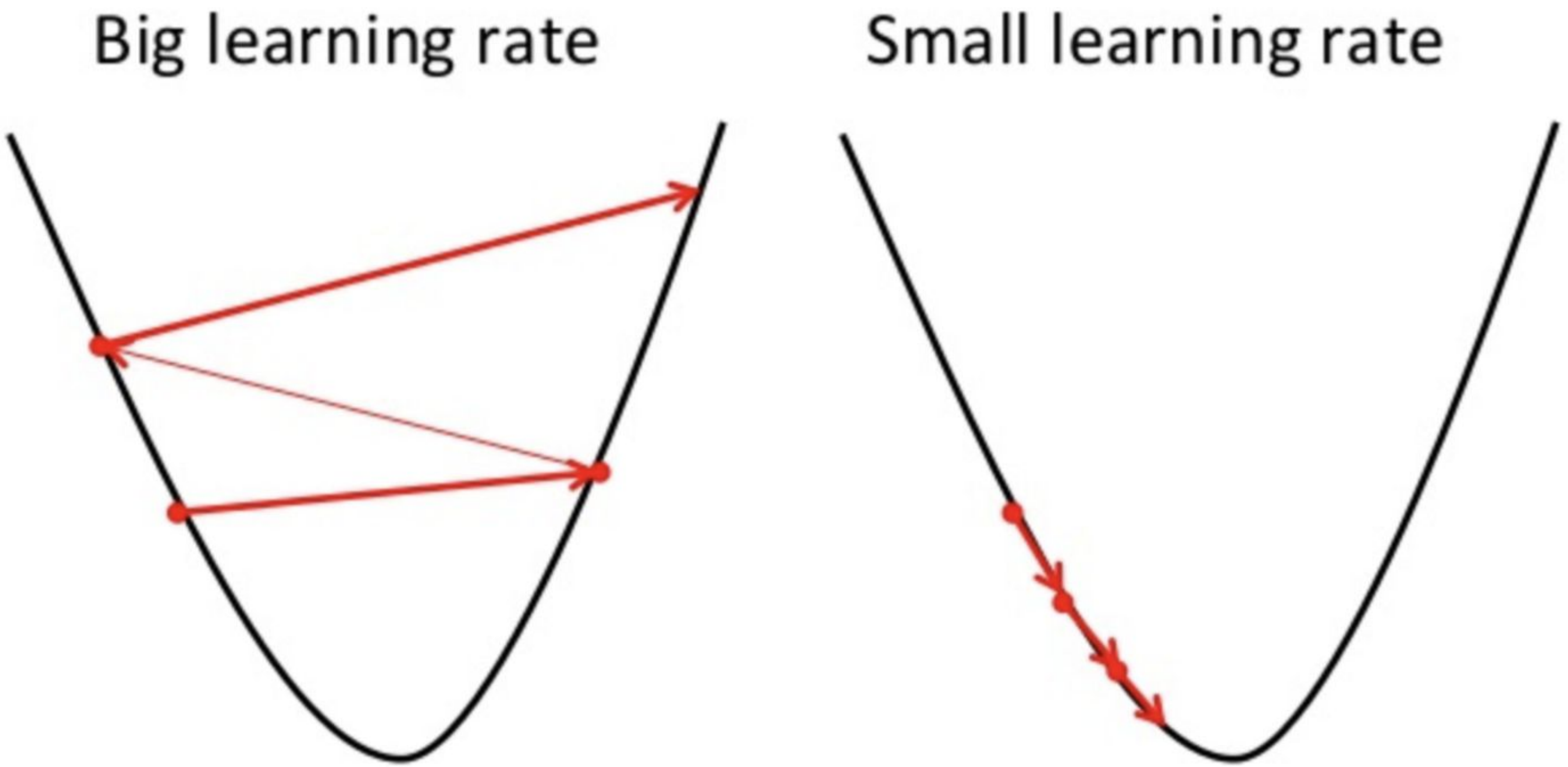
Gradient Descent



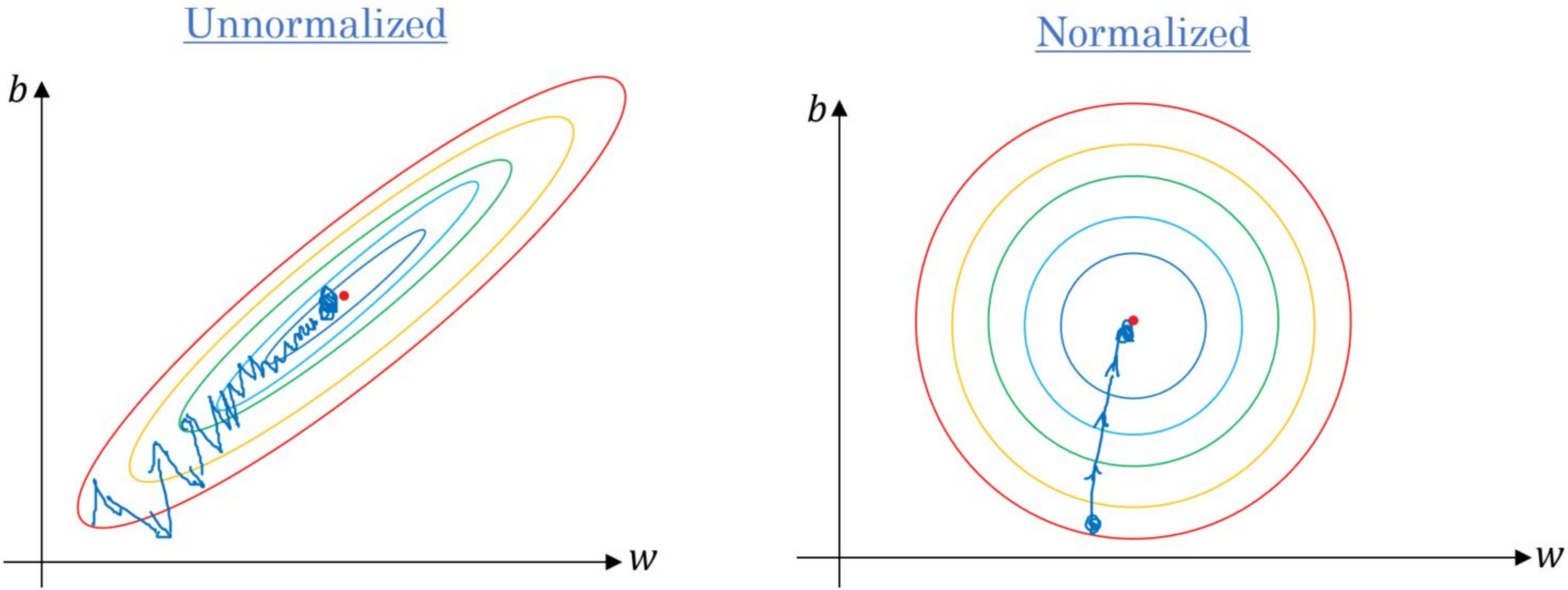
Local minima



Gradient Descent: Learning Rate



Gradient Descent: Impact of Normalization



Gradient Descent with Momentum:

$V_t = \beta V_{t-1} + (1-\beta)\, \delta L(w, x, y)$, where δL is the derivative of the loss function

$W = W - \alpha V_t$, where α is the learning rate, W -parameters, V-gradient with momentum

Data Normalization

Data Normalization

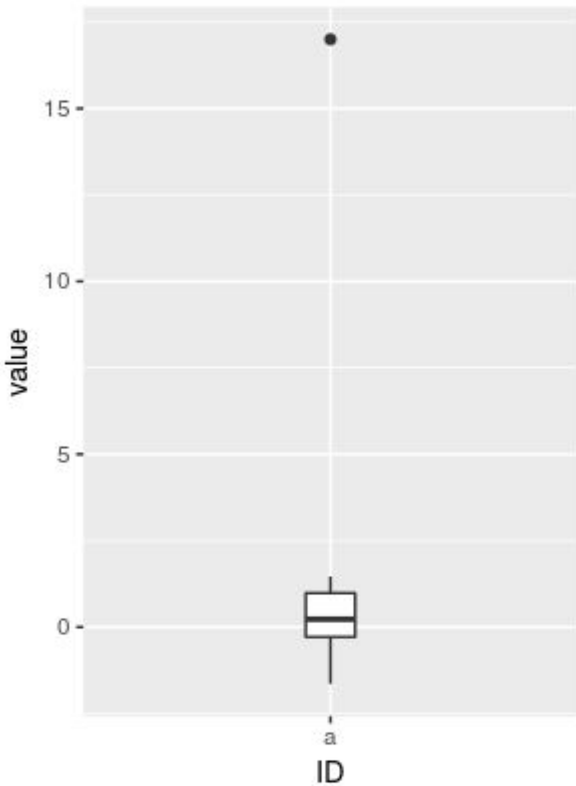
Examples:

1. Fix mean (demean): subtract mean
2. Fix standard deviation: divide by standard deviation
3. Fix distribution, i.e. apply ranking and inverse distribution
4. Remove noise (**Clustering** and **Principal Component Analysis**)

Outliers

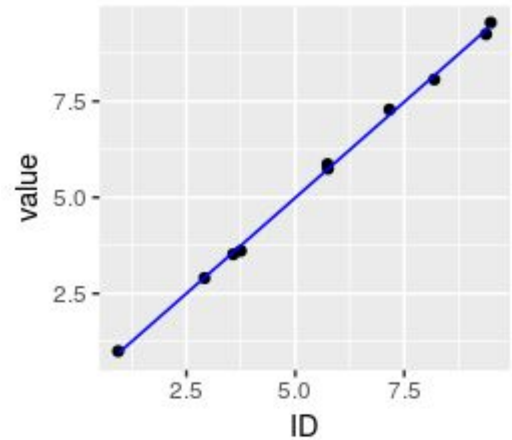
Fixing Outliers

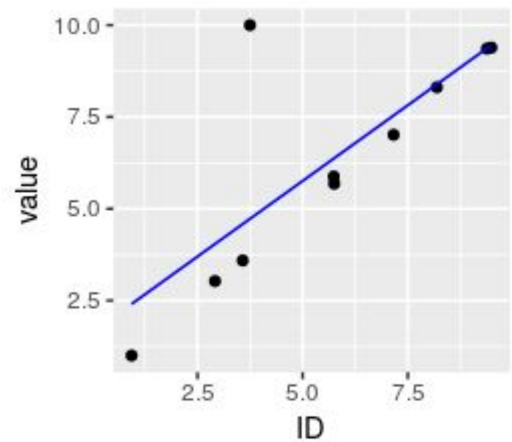
Univariate Method



Winsorizing vs **Trimming**

Multivariate Method





Minkowski Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^{1.5}$$

Example:

```
> 10 ^ 2
[1] 100
> 10 ^ 1.5
[1] 31.62278
```

Summary

We reviewed a number of **Machine Learning Models** and their **R package**:

- **Linear Model:** assumptions, advantage, feature transformations
- **Nearest Neighbour:** algorithm, advantages, difficulties, complexity
- **Trees**: algorithm, advantages, complexity and complexity control
- **SVMs**: algorithm, advantages, disadvantages, complexity control

We reviewed a number of **shrinkage** parameters:

- **Ridge**, **Lasso**, **ElasticNet** for Linear Regression
- Number of neighbours in K Nearest Neighbour
- Trees: max_depth, min_split etc.
- SVMs: margin, kernel

We studied **hyperparameter** calibration:

- Default parameters
- **Grid search**

https://prezi.com/06swcwazd0ai/machine-learning/