Applied Geodata Science I

Session 11

Prof. Dr. Benjamin Stocker

Spring semester 2023

GECO

UNIVERSITÄT BERN

Supervised machine learning

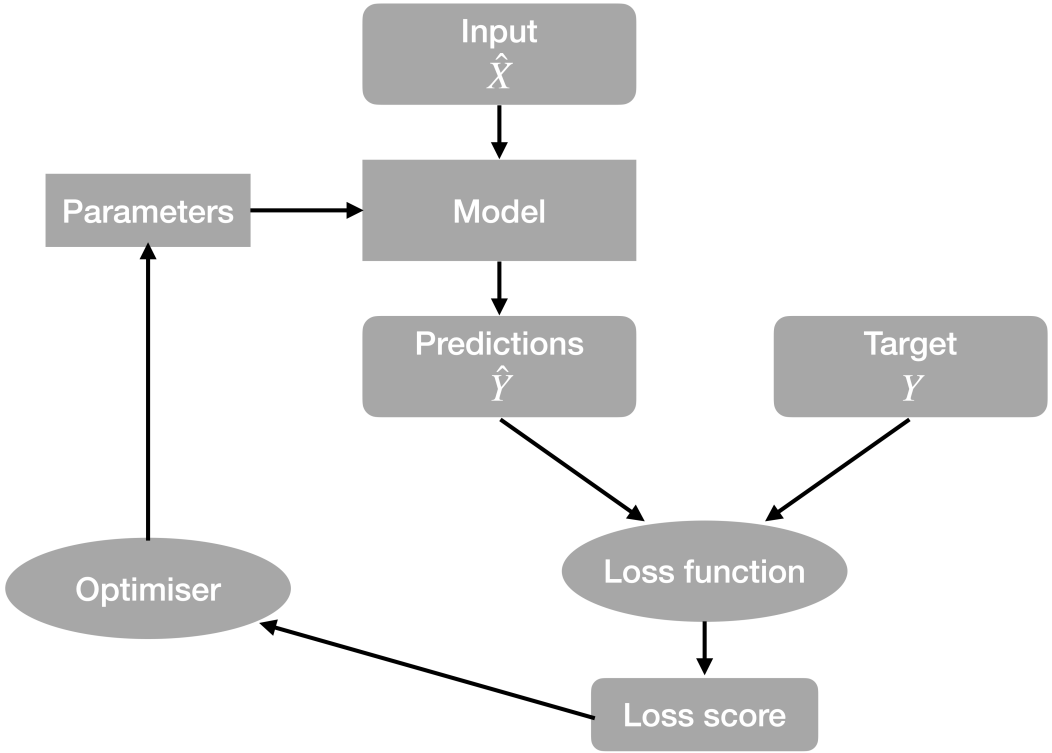
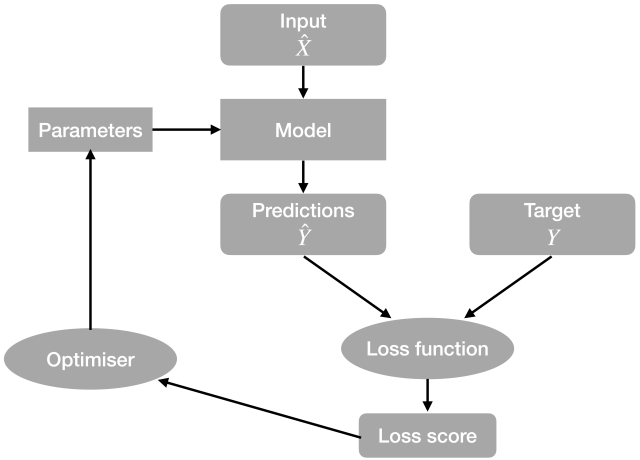
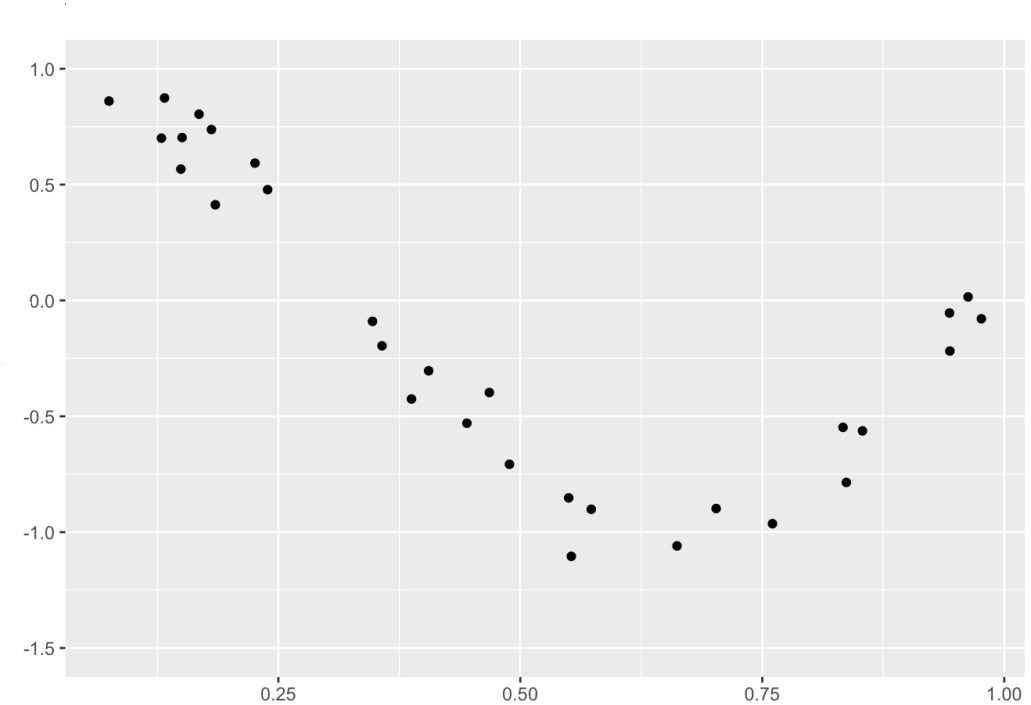
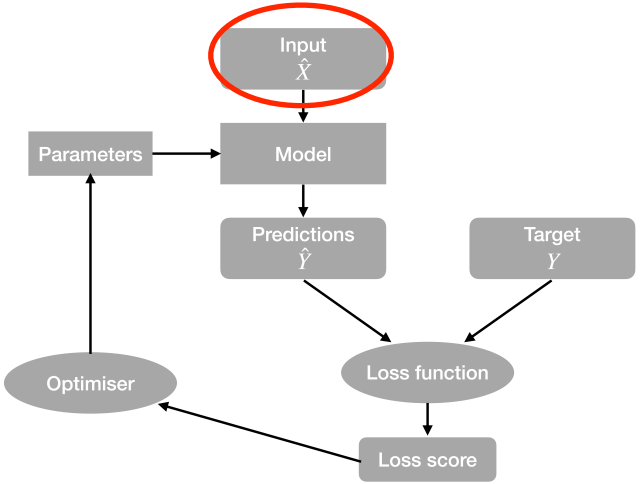
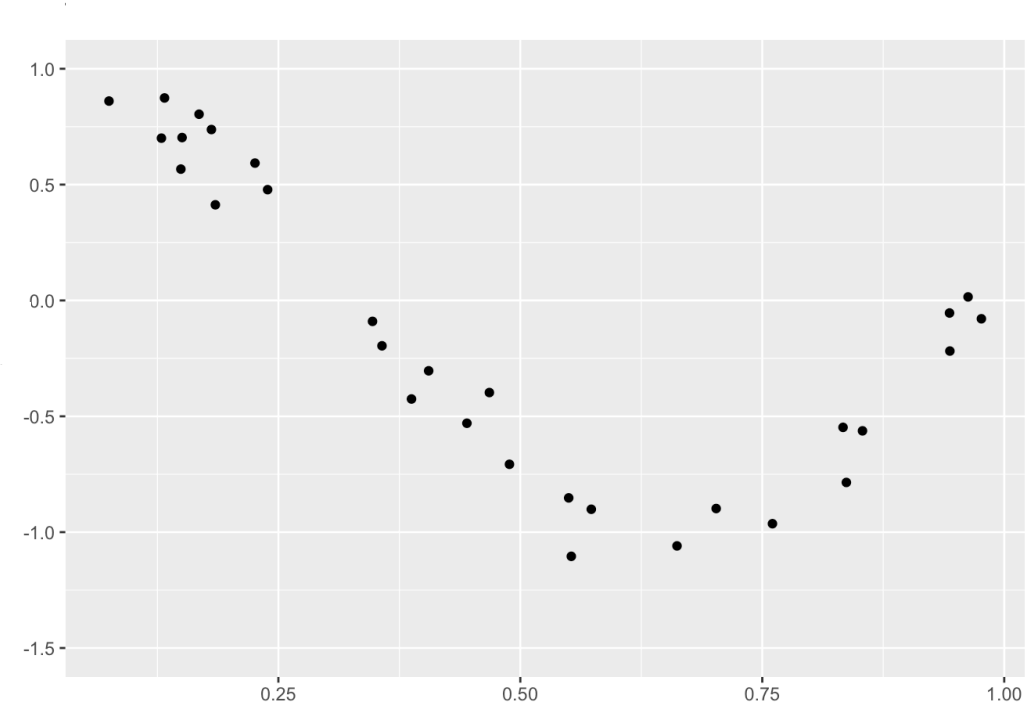


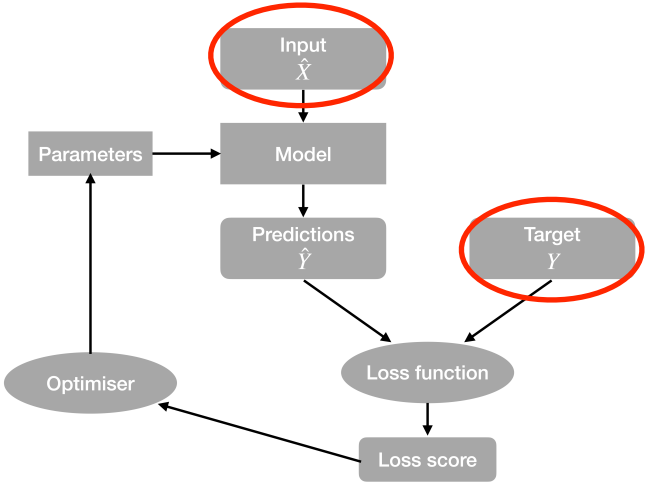
Figure adapted from Chollet & Allaire (2018)

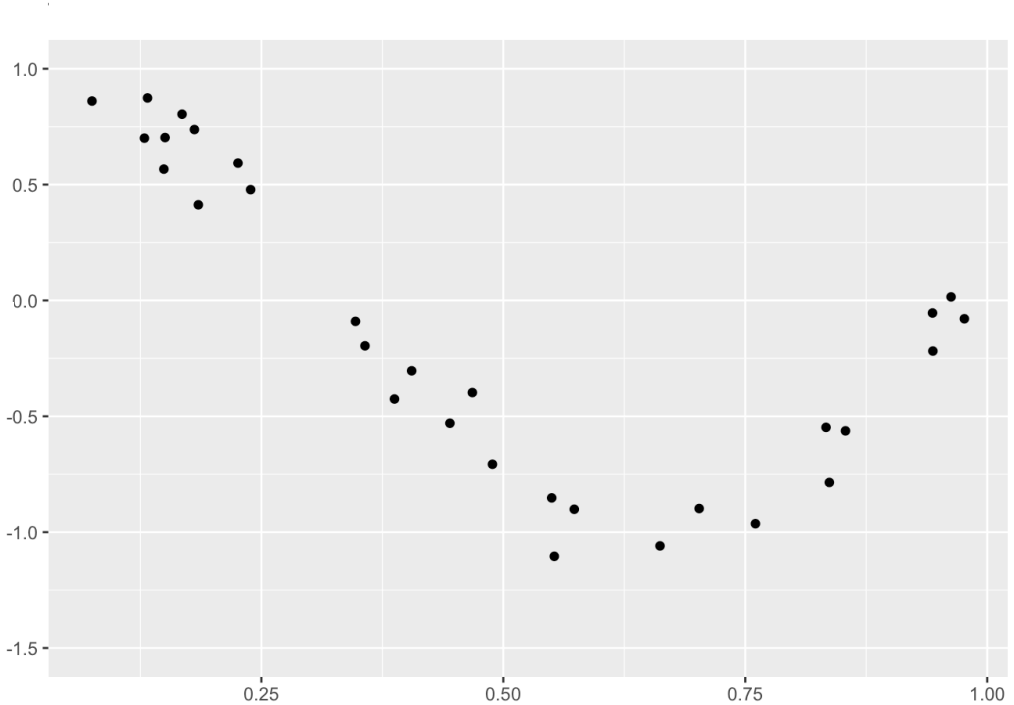

Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score

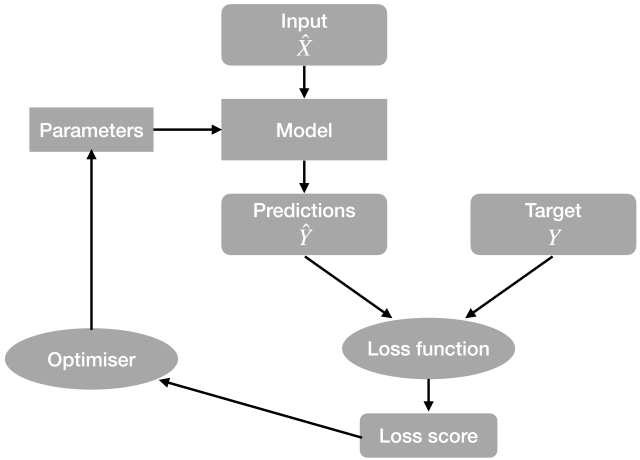

1.0
0.5
0.0
-0.5
-1.0
-1.5
0.25
0.50
0.75
1.00


Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score

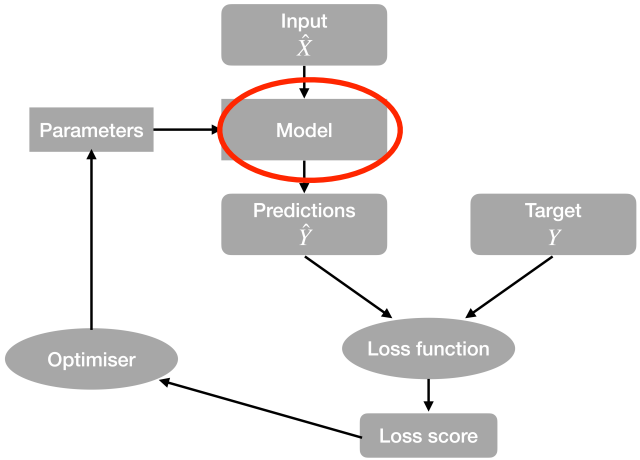

1.0
0.5
0.0
-0.5
-1.0
-1.5
0.25
0.50
0.75
1.00


Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score

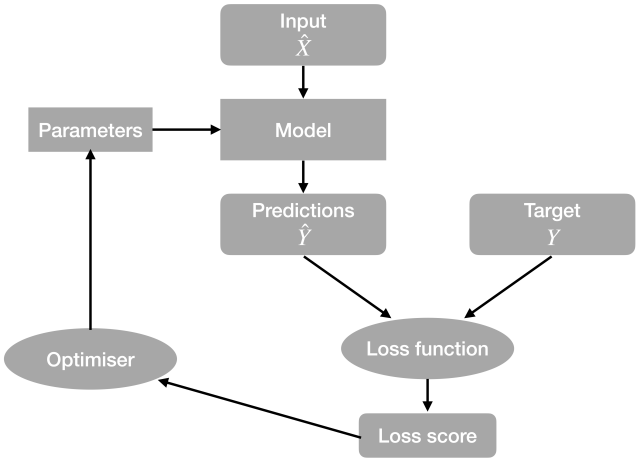

1.0
0.5
0.0
-0.5
-1.0
-1.5
0.25
0.50
0.75
1.00

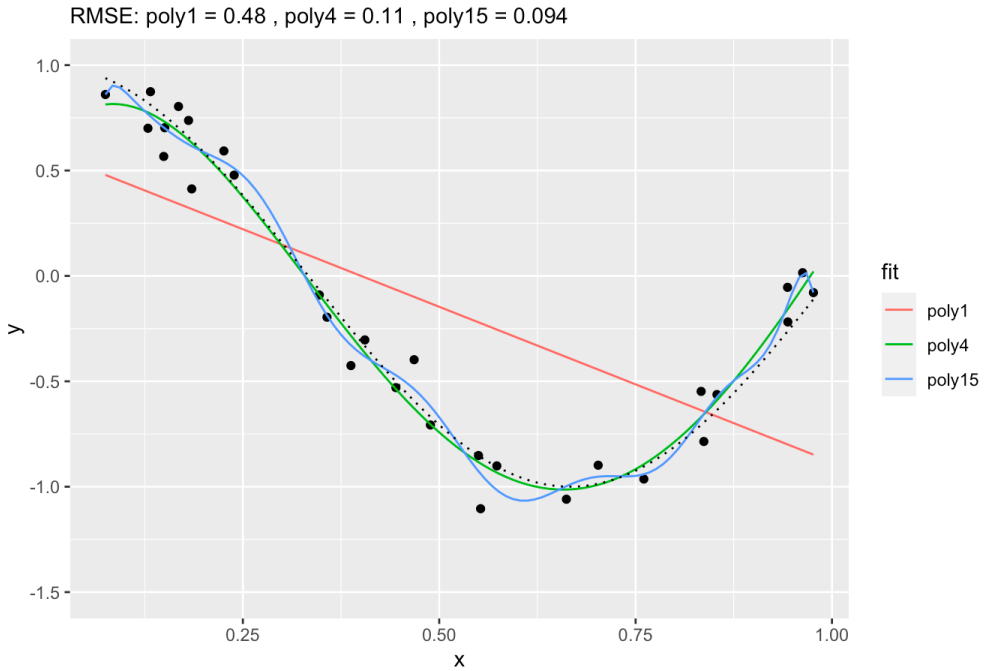


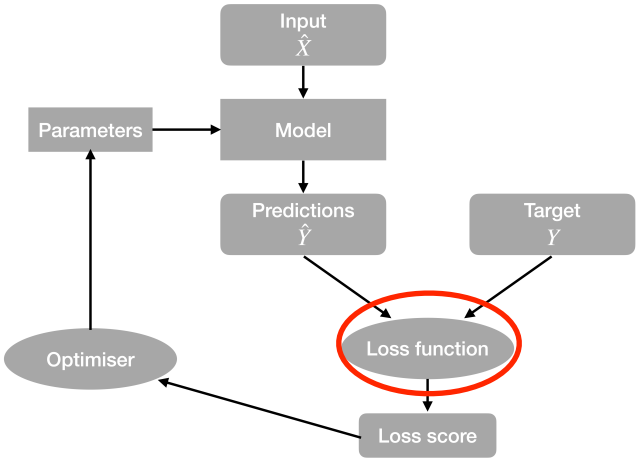
$$y = \sum_{n=0}^{N} a_n x^n$$

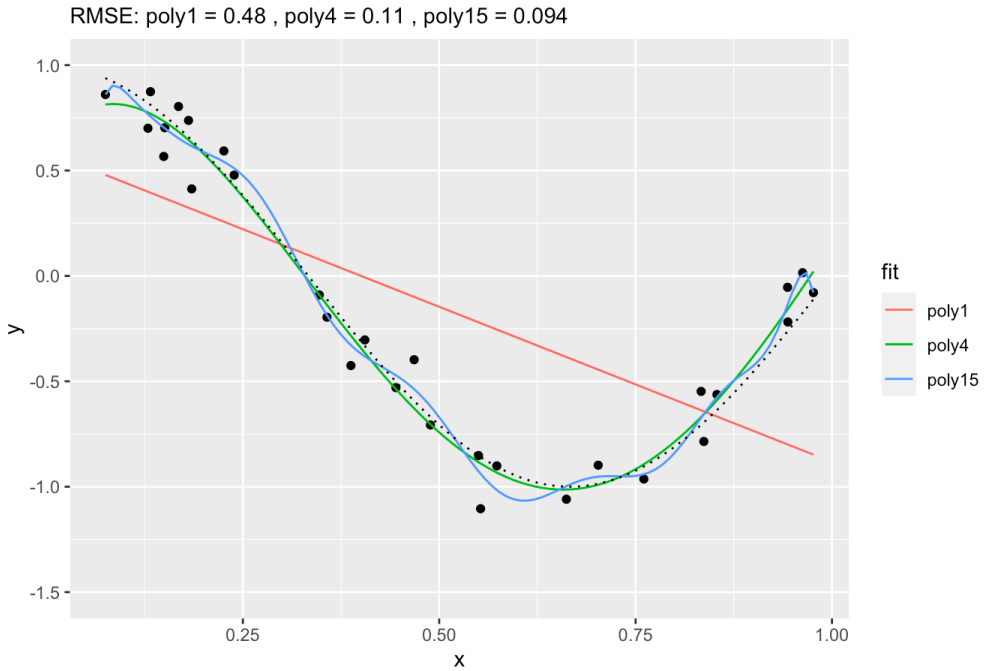


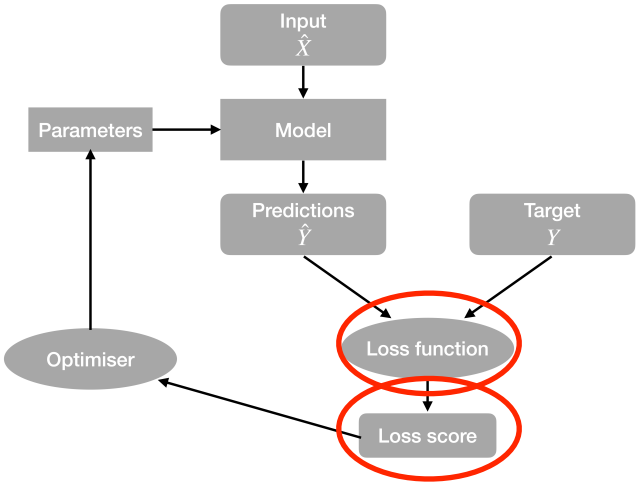
$$y = \sum_{n=0}^{N} a_n x^n$$

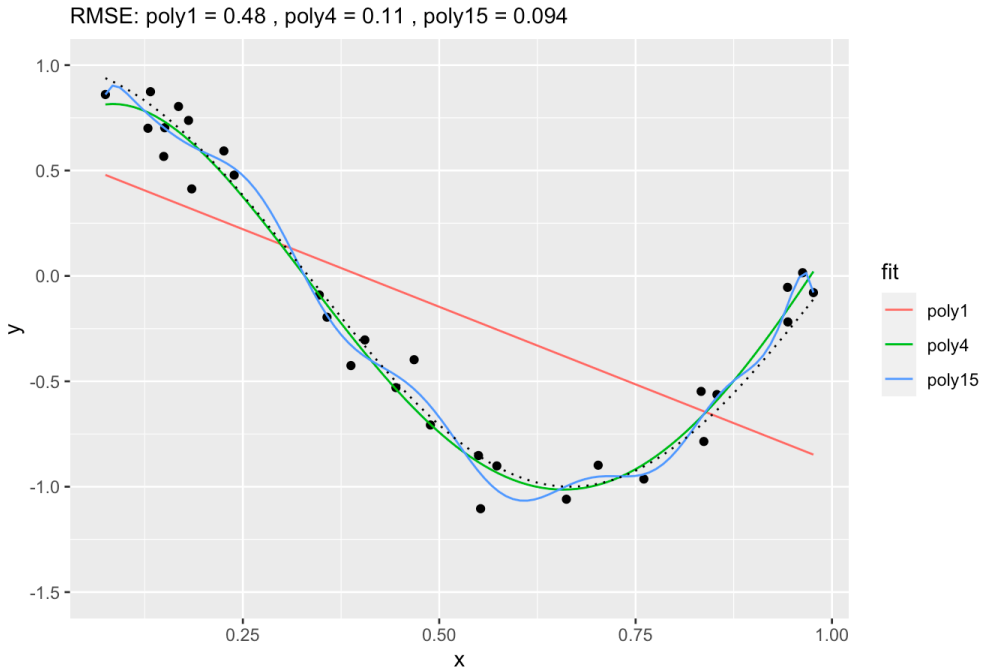

Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score

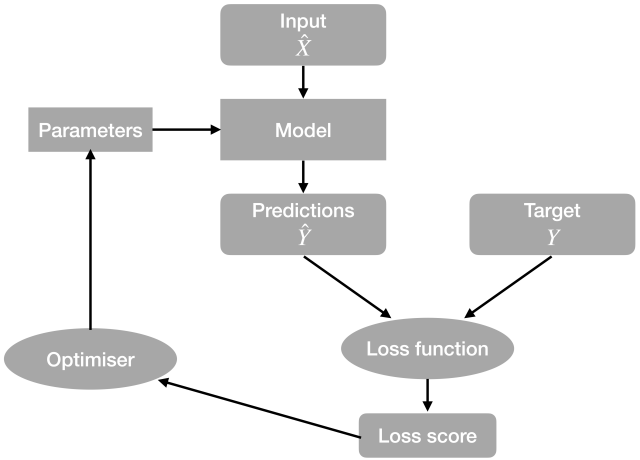

RMSE: poly1 = 0.48 , poly4 = 0.11 , poly15 = 0.094
1.0
0.5
0.0
-0.5
-1.0
-1.5
y
0.25
0.50
0.75
1.00
x
fit
poly1
poly4
poly15


Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score


RMSE: poly1 = 0.48 , poly4 = 0.11 , poly15 = 0.094
1.0
0.5
0.0
-0.5
-1.0
-1.5
y
0.25
0.50
0.75
1.00
x
fit
poly1
poly4
poly15


Input
$\hat{X}$
Parameters
Model
Predictions
$\hat{Y}$
Target
Y
Optimiser
Loss function
Loss score

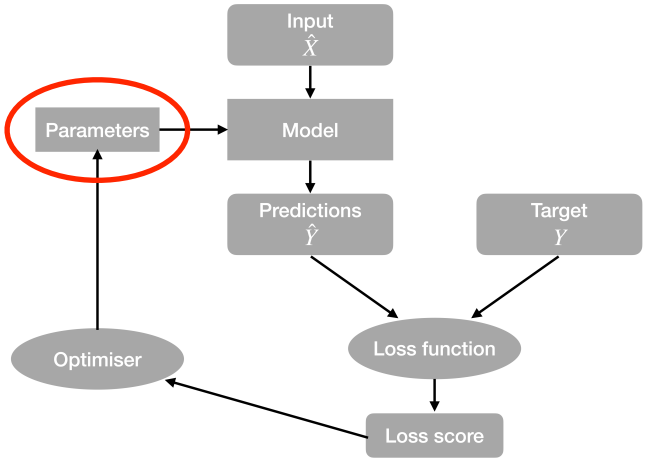

RMSE: poly1 = 0.48 , poly4 = 0.11 , poly15 = 0.094
1.0
0.5
0.0
-0.5
-1.0
-1.5
y
0.25
0.50
0.75
1.00
x
fit
poly1
poly4
poly15




```
> polyfit_4

Call:
lm(formula = y ~ poly(x, 4), data = df_train)

Coefficients:
(Intercept)  poly(x, 4)1  poly(x, 4)2  poly(x, 4)3  poly(x, 4)4
    -0.1310      -2.3504       2.4388       0.7023      -0.3058
```

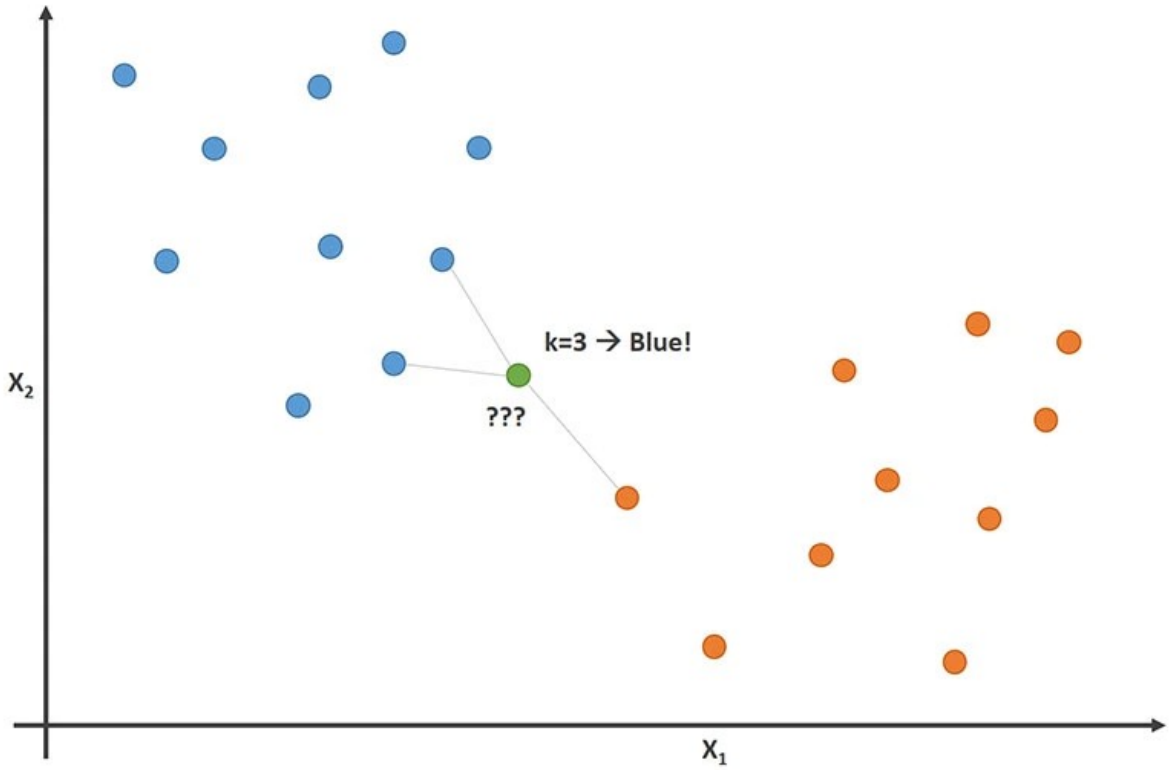



```
> polyfit_4

Call:
lm(formula = y ~ poly(x, 4), data = df_train)

Coefficients:
(Intercept)  poly(x, 4)1  poly(x, 4)2  poly(x, 4)3  poly(x, 4)4
    -0.1310      -2.3504       2.4388       0.7023      -0.3058
```

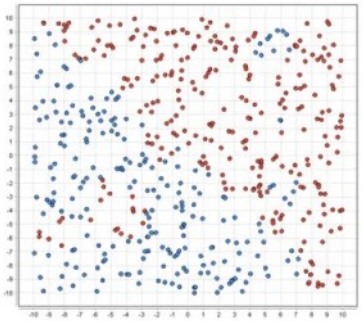
k-Nearest Neighbours



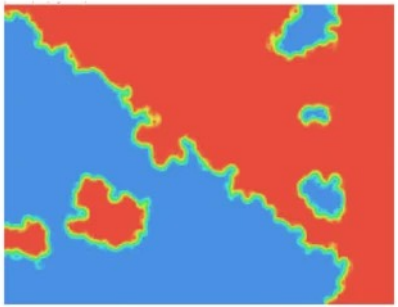
https://rapidminer.com/blog/k-nearest-neighbors-laziest-machine-learning-technique/

k-Nearest Neighbours

Data

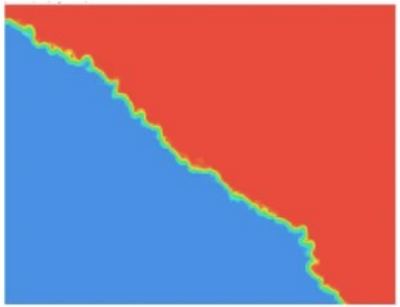


k=1



k=10

k=50



https://rapidminer.com/blog/k-nearest-neighbors-laziest-machine-learning-technique/

KNN as a demo algorithm

KNN as a demo algorithm

- Fast and “lazy”: no calculations during training

KNN as a demo algorithm

- Fast and “lazy”: no calculations during training
- Intuitive meaning of the single hyperparameter: *k*

KNN as a demo algorithm

- Fast and “lazy”: no calculations during training
- Intuitive meaning of the single hyperparameter: *k*
- Illustrative for the need of data standardisation

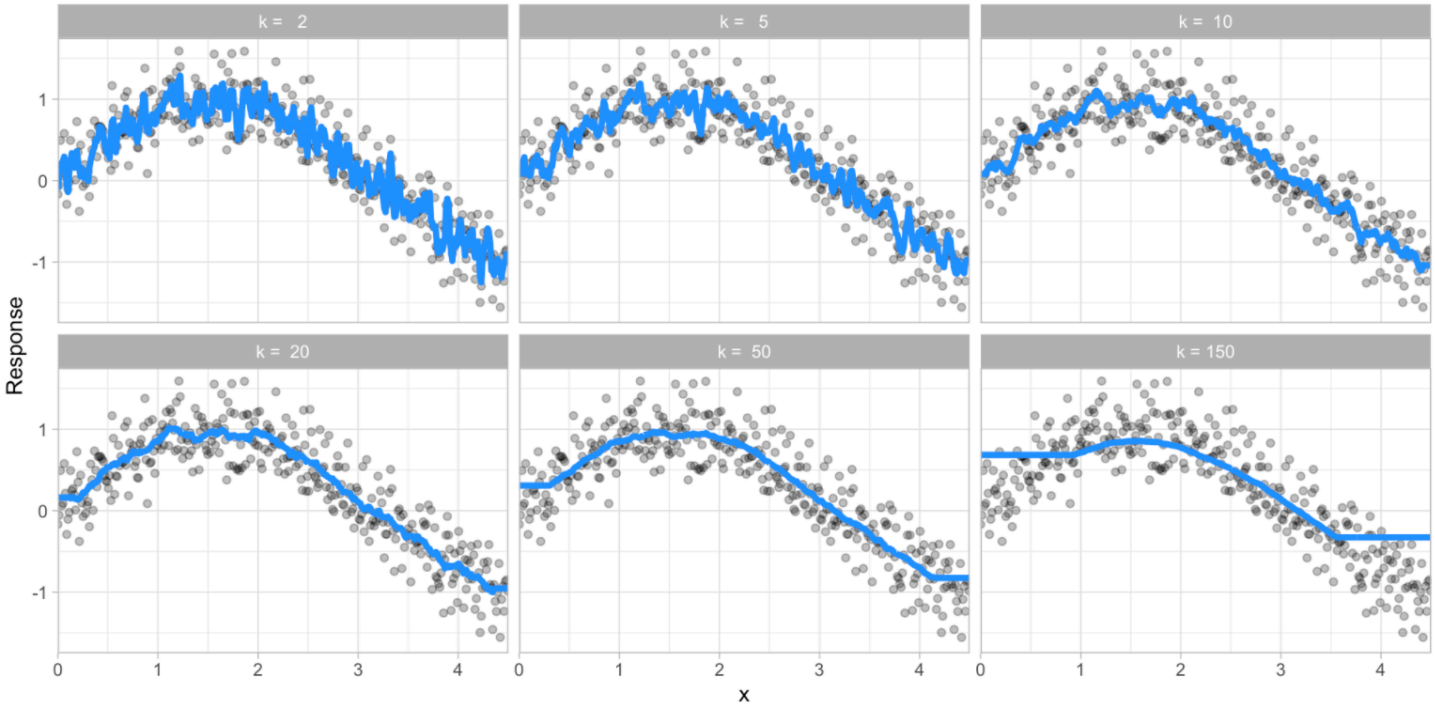
KNN as a demo algorithm

- Fast and “lazy”: no calculations during training
- Intuitive meaning of the single hyperparameter: *k*
- Illustrative for the need of data standardisation
- Widely used, e.g., for data imputation

KNN as a demo algorithm

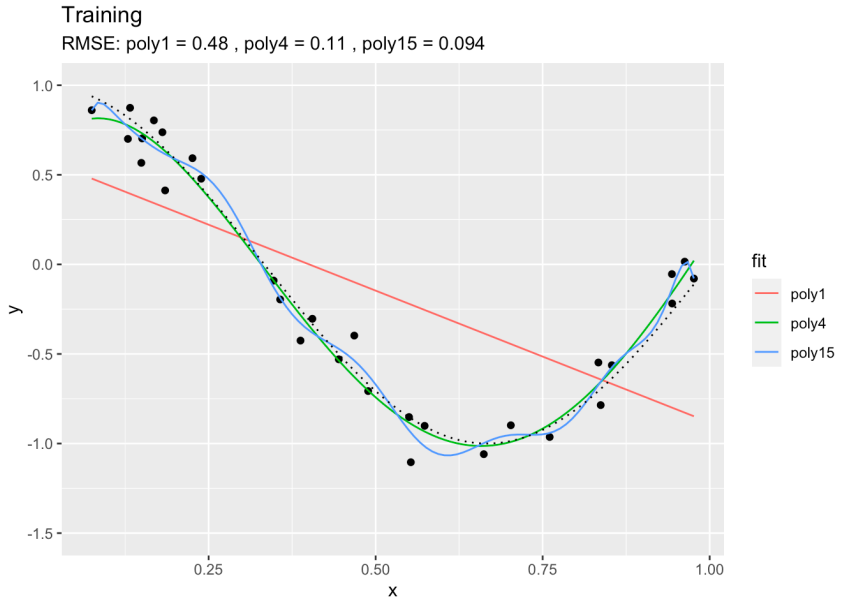
- Fast and “lazy”: no calculations during training
- Intuitive meaning of the single hyperparameter: *k*
- Illustrative for the need of data standardisation
- Widely used, e.g., for data imputation
- Note: needs to read all data into memory

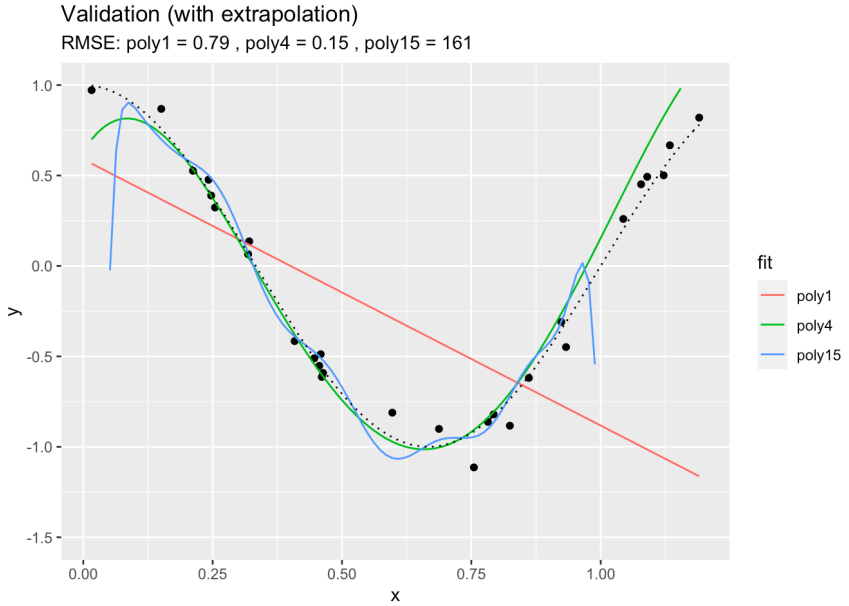
k-Nearest Neighbours



Bradley & Boehmke *Hands On Machine Learning in R*

Validation (with extrapolation)

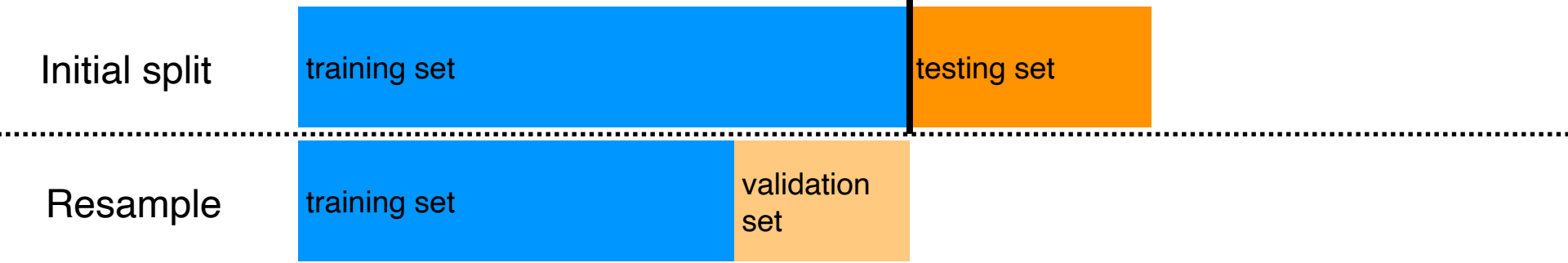




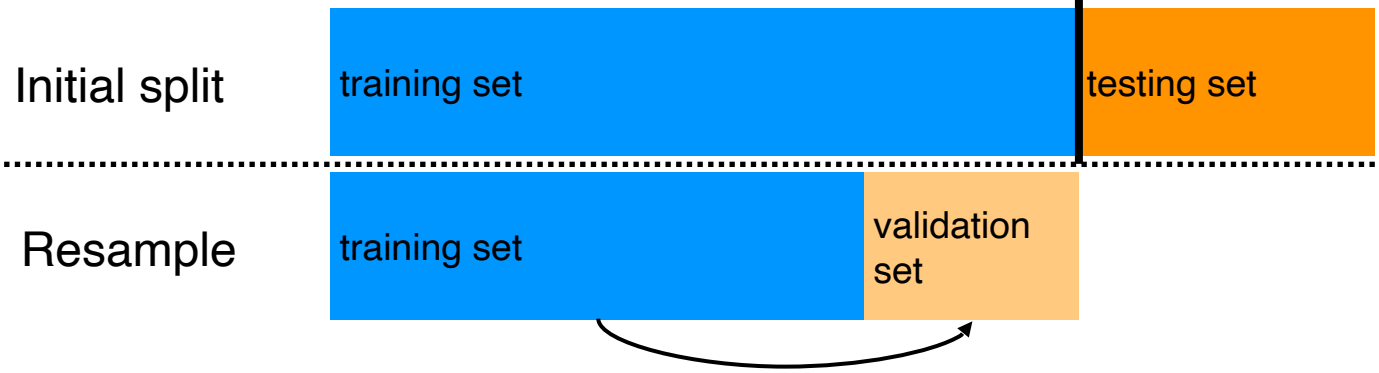
Testing vs. validation set

Initial split

Testing vs. validation set

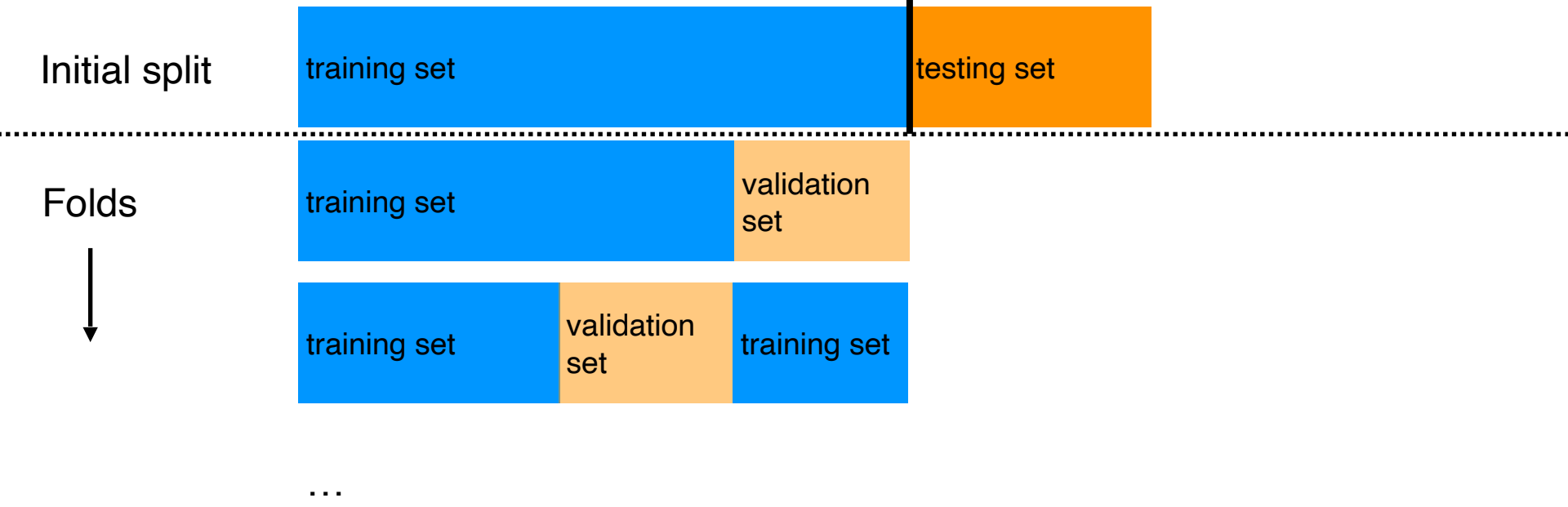


Testing vs. validation set

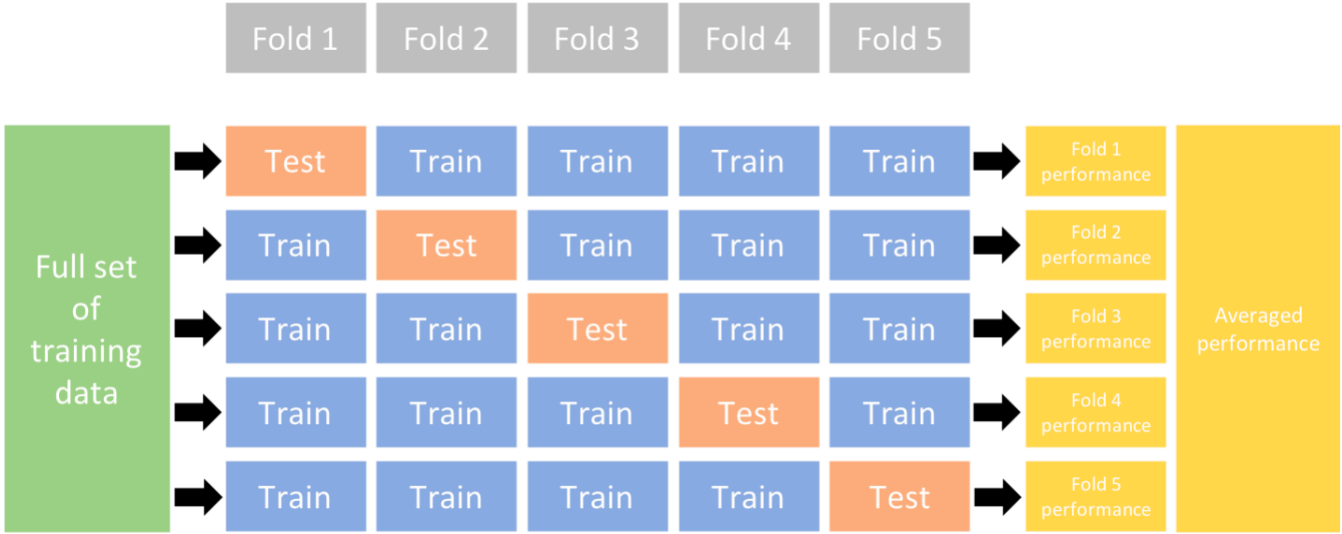


- Hyperparameter tuning
- Feature selection

K-fold cross validation



Resampling



Boehmke & Greenwell (2019) *Hands On Machine Learning in R*

Workflow of model training and testing

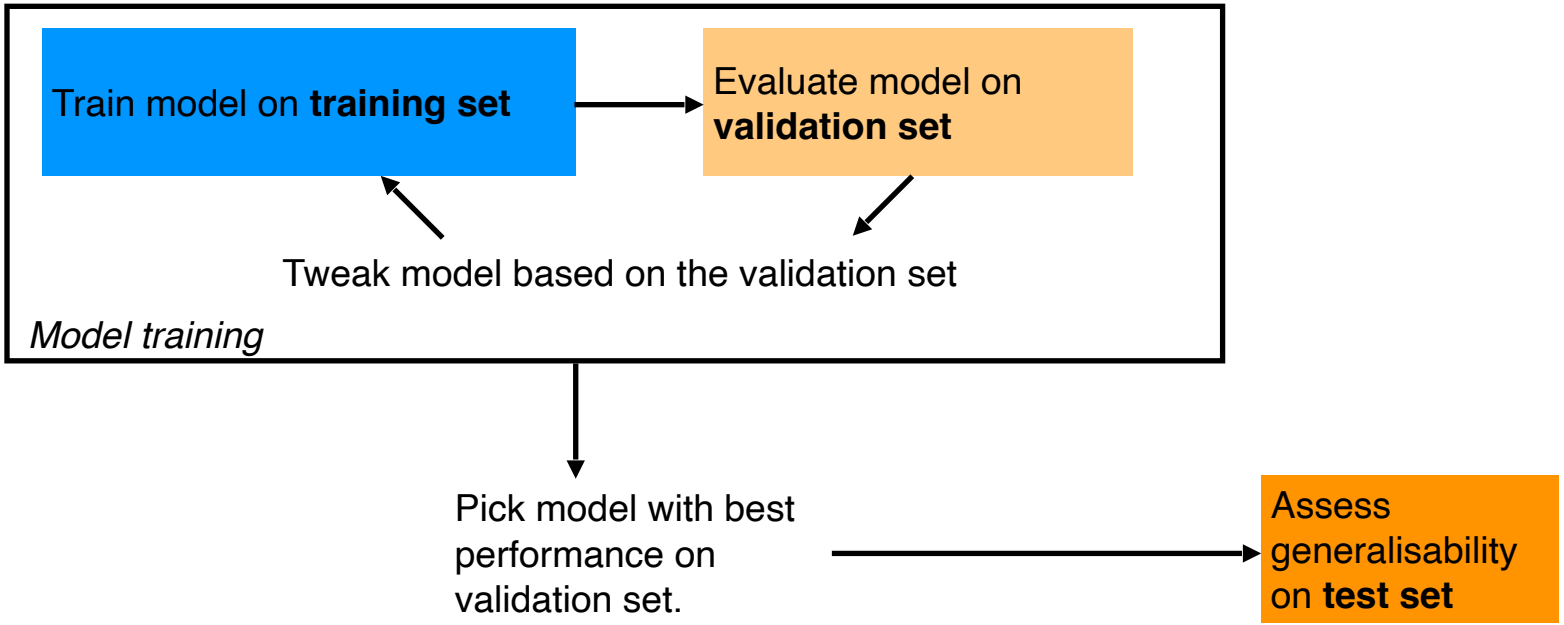


Figure adopted form [Google Machine Learning Crash Course]

Workflow of model training and testing

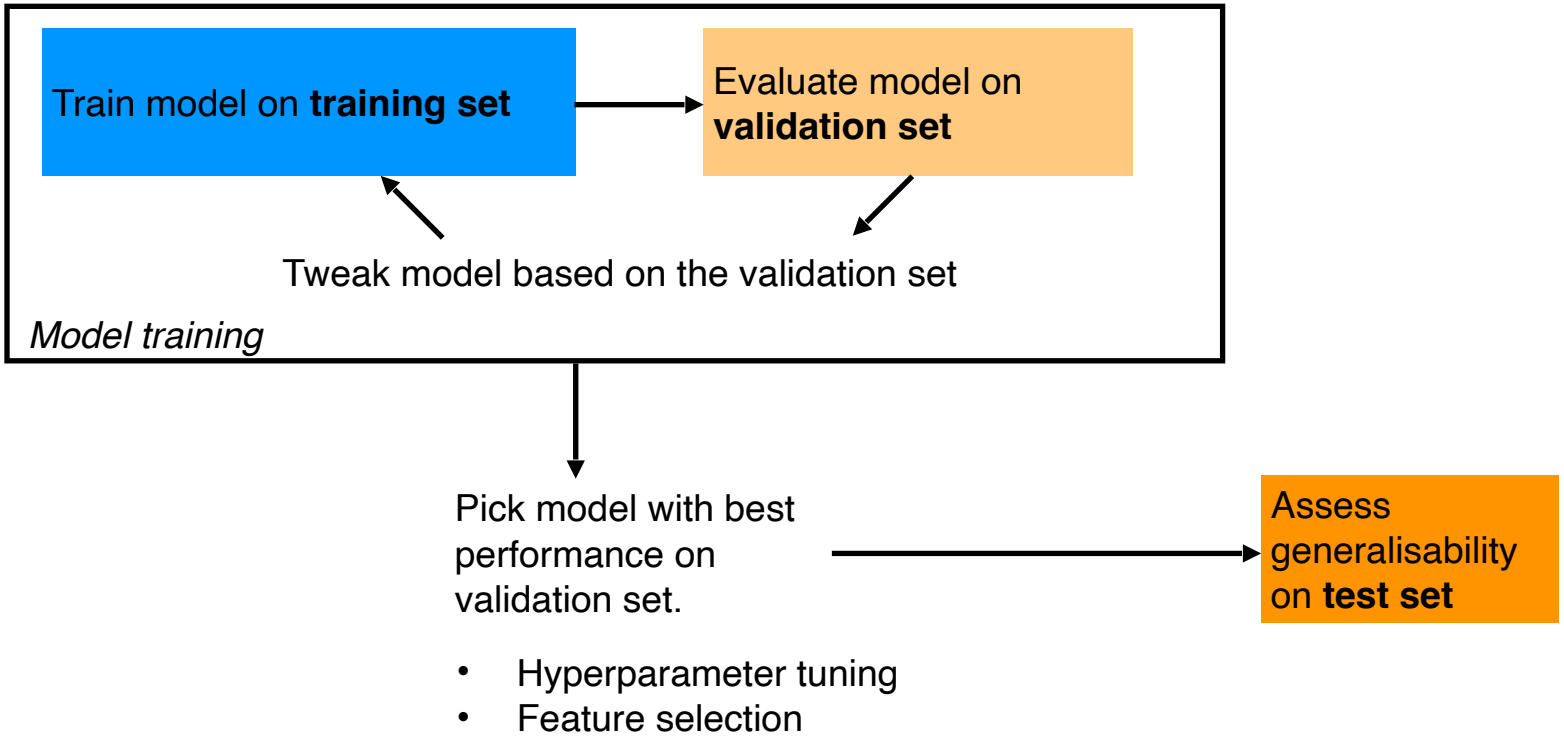


Figure adopted form [Google Machine Learning Crash Course]