# Python computational pipeline for predictive machine learning modelling of livestock data

**Dan Tulpan, Associate Professor**

dtulpan@uoguelph.ca

Centre for Genetic Improvement of Livestock (CGIL)

Department of Animal Biosciences

University of Guelph, Ontario, Canada

IMPROVE LIFE.

# Summary

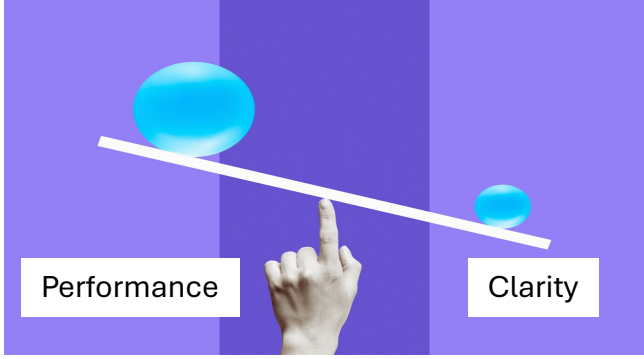


- **What you get from this workshop**
  - Some (hopefully functional) Python code … for regression problems (due to time constraints)
    - The code relies on the Python scikit-learn library

    https://scikit-learn.org/
  - Some information and explanations of what the code does and why
- **Assumptions**
  - You know a bit about machine learning
    - If not, read this: *Greener et al. (2021): A guide to machine learning for biologists*

    (https://www.nature.com/articles/s41580-021-00407-0)
  - You can operate a computer

# Warnings / Disclaimers




- Python code is not optimized or comprehensive
  - It is built to (hopefully) facilitate understanding
  - Sacrificed performance and best programming practices
- Input datasets are assumed to be ready and clean
  - Your job
- The code should only be used for good causes
- If you make money with this code my share is 10% (cash, check or plastic is fine) 😎

# Python Use

- Follow the instructions provided in the "Python_usage_instructions.pdf" file

# Data formatting

- Expectations:
  - Tabular format
  - Last column contains the predictor variable
  - Data was cleaned prior to using the Python script
  - Data includes only numeric values

- Recommended reading:
  - *Browman and Woo (2018) - Data Organization in Spreadsheets* (https://www.tandfonline.com/doi/full/10.1080/00031305.2017.1375989 )

# Data sets (for this workshop)

- 2 subsets of the data from:

Marshall et al. (2023): `A farmer-friendly tool for estimation of weights of pigs kept by smallholder farmers in Uganda`

- **Article:** ttps://link.springer.com/article/10.1007/s11250-023-03561-z
- **Data:** https://data.mel.cgiar.org/dataset.xhtml?persistentId=hdl:20.500.11766.1/FK2/IWXZQH

# MarshallEtAl2023_more_selected_measurements.csv

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10205837/

**Records**

754 pigs from Uganda

**Variables**

- 6 input variables:
  - household_id
  - age_months
  - heartgirth
  - height
  - length
  - body_condition_score
- 1 output variable:
  - exact_weight

MarshallEtAl2023_more_selected_measurements

| household_id | age_months | heartgirth | height | length | body_condition_score | exact_weight |
|---|---|---|---|---|---|---|
| PBM-KML-113 | 34 | 140 | 901 | 141 | 4 | 205 |
| PBM-MSK-138 | 24 | 0 | 0 | 0 | 4 | 200 |
| PBM-MSK-107 | 15 | 130 | 80 | 138 | 4 | 193.2 |
| PBM-MSK-106 | 41 | 140 | 76 | 141 | 4 | 177.2 |
| PBM-WKS-401 | 27 | 128 | 85 | 140 | 4 | 170 |
| PBM-KML-106 | 30 | 121 | 72 | 140 | 4 | 160 |
| PBM-MSK-137 | 19 | 124 | 76 | 142 | 4 | 148 |
| PBM-WKS-401 | 24 | 122 | 81 | 136 | 3 | 137.7 |
| PBM-MSK-139 | 18 | 134 | 89 | 147 | 3 | 134 |
| PBM-MSK-102 | 20 | 117 | 81 | 149 | 4 | 132.9 |
| PBM-MSK-142 | 13 | 121 | 80 | 140 | 4 | 131.5 |
| PBM-WKS-416 | 43 | 120 | 72 | 145 | 3 | 131.1 |
| PBM-HMA-240 | 12 | 113 | 90 | 137 | 3 | 129.5 |
| PBM-MSK-107 | 12 | 112 | 78 | 136 | 4 | 127.3 |
| PBM-MSK-102 | 20 | 122 | 77 | 135 | 4 | 126.5 |

# KabululuEtAl2023_selection.csv

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0295433

**Records**

400 pigs from Tanzania

**Variables**

- 6 input variables (all numeric):
  - Heart_girth
  - Body_length
  - Age_months
  - Sex
  - District
  - Village
- 1 output variable:
  - Weight_kg_by_scale

KabululuEtAl2023_selection

| HEART_GIRTH | BODY_LENGTH | AGE_MONTHS | SEX | DISTRICT | VILLAGE | WEIGHT_KG_BY_SCALE |
|---|---|---|---|---|---|---|
| 118 | 137 | 24 | Female | Mbeya | Mshewe | 91 |
| 69 | 95 | 6 | Female | Mbeya | Njelenje | 33 |
| 65 | 97 | 8 | Female | Mbeya | Njelenje | 26 |
| 66 | 71 | 9 | Female | Mbeya | Njelenje | 20 |
| 72 | 90 | 12 | Male | Mbeya | Njelenje | 37 |
| 67 | 93 | 17 | Female | Mbeya | Njelenje | 34 |
| 78 | 89 | 12 | Female | Mbeya | Njelenje | 42.5 |
| 80 | 90 | 8 | Male | Mbeya | Mjele | 37 |
| 61 | 74 | 7 | Female | Mbeya | Mjele | 23 |
| 60 | 80 | 8 | Female | Mbeya | Mjele | 25 |
| 86 | 115 | 14 | Female | Mbeya | Mjele | 59 |
| 80 | 94 | 9 | Male | Mbeya | Mjele | 42 |
| 72 | 91 | 8 | Female | Mbeya | Mjele | 29 |
| 70 | 90 | 12 | Female | Mbeya | Mjele | 28 |
| 55 | 69 | 8 | Female | Mbeya | Mjele | 17 |
| 76 | 86 | 8 | Female | Mbeya | Mjele | 31 |
| 71 | 79 | 8 | Female | Mbeya | Mjele | 39 |
| 63 | 78 | 8 | Female | Mbeya | Mjele | 20 |
| 57 | 62 | 8 | Female | Mbeya | Mjele | 18 |
| 68 | 80 | 8 | Female | Mbeya | Mjele | 26 |
| 52 | 70 | 6 | Female | Mbeya | Mjele | 14 |
| 63.5 | 78 | 10 | Male | Mbeya | Mjele | 25 |

# RemusEtAl2020_lysine_requirements.csv

https://www.sciencedirect.com/science/article/pii/S1751731119002660

**Records**

40 pigs from Canada

**Variables**

- 8 input variables:
  - Pig_identification
  - Methionine_intake
  - Met+Cys_intake
  - Cys_intake
  - ADG (avg. daily gain)
  - ADFI (avg. daily feed intake)
  - BW_initial
  - BW_final
- 1 output variable:
  - Lys_intake

RemusEtAl2020_lysine_requirements

| | PIG_identificati | Methionine_ inta | Met_Cys_intal | Cys_intak | ADG | ADFI | BWinitia | BWfina | Lysine_intak |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 5.56 | 8.92 | 3.36 | 0.85 | 1.65 | 25.10 | 45.60 | 18.53 |
| 3 | 2 | 9.51 | 13.93 | 4.42 | 0.96 | 2.15 | 27.50 | 50.45 | 24.38 |
| 4 | 4 | 3.06 | 5.47 | 2.41 | 0.45 | 1.30 | 24.85 | 35.65 | 14.59 |
| 5 | 7 | 6.97 | 11.18 | 4.21 | 1.01 | 2.08 | 27.70 | 51.95 | 23.22 |
| 6 | 8 | 4.52 | 8.08 | 3.56 | 0.91 | 1.92 | 25.55 | 47.46 | 21.54 |
| 7 | 9 | 5.45 | 8.75 | 3.30 | 1.02 | 2.09 | 25.00 | 49.45 | 18.17 |
| 8 | 14 | 8.85 | 12.97 | 4.12 | 1.08 | 2.04 | 25.10 | 50.90 | 22.70 |
| 9 | 15 | 5.62 | 10.03 | 4.41 | 1.10 | 2.36 | 25.20 | 51.60 | 26.74 |
| 10 | 16 | 6.00 | 8.79 | 2.79 | 0.69 | 1.36 | 22.20 | 38.70 | 15.39 |
| 11 | 17 | 6.18 | 9.92 | 3.74 | 0.85 | 1.81 | 22.95 | 43.45 | 20.61 |
| 12 | 18 | 7.71 | 11.30 | 3.58 | 1.01 | 1.77 | 22.30 | 46.65 | 19.77 |
| 13 | 19 | 4.46 | 7.97 | 3.51 | 1.03 | 1.88 | 25.85 | 50.45 | 21.24 |
| 14 | 21 | 6.40 | 10.28 | 3.87 | 1.01 | 1.90 | 25.95 | 50.25 | 21.35 |
| 15 | 23 | 6.40 | 9.48 | 3.07 | 0.99 | 2.21 | 25.80 | 45.05 | 16.42 |
| 16 | 25 | 4.48 | 7.20 | 2.71 | 0.79 | 1.35 | 23.75 | 42.60 | 14.95 |
| 17 | 26 | 5.86 | 8.70 | 2.84 | 0.94 | 2.15 | 30.15 | 45.75 | 15.02 |
| 18 | 27 | 5.10 | 9.11 | 4.01 | 1.08 | 2.15 | 27.35 | 53.25 | 24.28 |
| 19 | 29 | 8.83 | 12.94 | 4.11 | 1.04 | 2.01 | 24.80 | 49.80 | 22.65 |
| 20 | 31 | 6.25 | 10.02 | 3.78 | 0.88 | 1.84 | 27.40 | 48.50 | 20.82 |
| 21 | 33 | 4.37 | 7.80 | 3.43 | 0.90 | 1.86 | 24.60 | 46.10 | 20.80 |

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Regression pipeline

1. **Data cleaning**
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Data cleaning


- Remove rows with missing values
- Remove duplicate rows
- Remove duplicate columns
- Remove single value columns
- Find and remove outliers (Z-score method)
- Change categorical columns to numeric
- Save cleaned dataset

# Regression pipeline

1. Data cleaning
2. **Data summarization**
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Overall look at the data

- Check the size of the dataset
  - Number of records (rows)
  - Number of variables/features (columns)
- Look at the first few records
- Look at descriptive statistics
  - Check for obvious outliers or extreme values

# Regression pipeline

1. Data cleaning
2. Data summarization
3. **Data visualization**
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Explore the data visually first

- If feasible/applicable
- Check the distribution of the variables
  - Histograms
  - Scatter-plots
- Check correlations among variables/features

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. **Data splitting**
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Prepare data for modelling

- Separate data into training (80%) and testing (20%)
  - The percentages depend on data size, available time, goals

- Training set:
  - Model construction
  - Model validation
  - Hyper-parameter optimization

- Testing set:
  - Testing the final models

**Golden Rule of Machine Learning**

**NEVER EVER** use the testing set during the construction/validation/optimization stage of a model.

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. **Data scaling**
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Scaling your data

- How
  - Transform data to a standardized range
  - StandardScaler, MinMaxScaler, RobustScaler

- Why
  - Reduces the impact of extreme values

    … for algorithms sensitive to outliers or for those relying on normality assumptions

  - Reduces differences in value scales among variables
    - Speeds up convergence and provides equal opportunities for features to influence the outcome variable

  - Helps making more robust models

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. **Model initialization (default params)**
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# ML Models

- Select models from different categories
  - Tree-based: Decision Tree, AdaBoost, Random Forest
  - Artificial Neural Networks: Multi Layer Perceptron
  - Lazy estimators: K-Nearest Neighbour
  - Linear: Linear Model, LASSO, Ridge
  - Gradient-based: Gradient Boost

- Select more than 2 models
  - Different strengths and weaknesses
  - Different data representations

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. **Preliminary model evaluation**
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Model evaluation strategies

- K-fold cross-validation
  - Choose K as a function of data size and computing time
    - High K values: small-medium datasets
    - Low  K values: large datasets

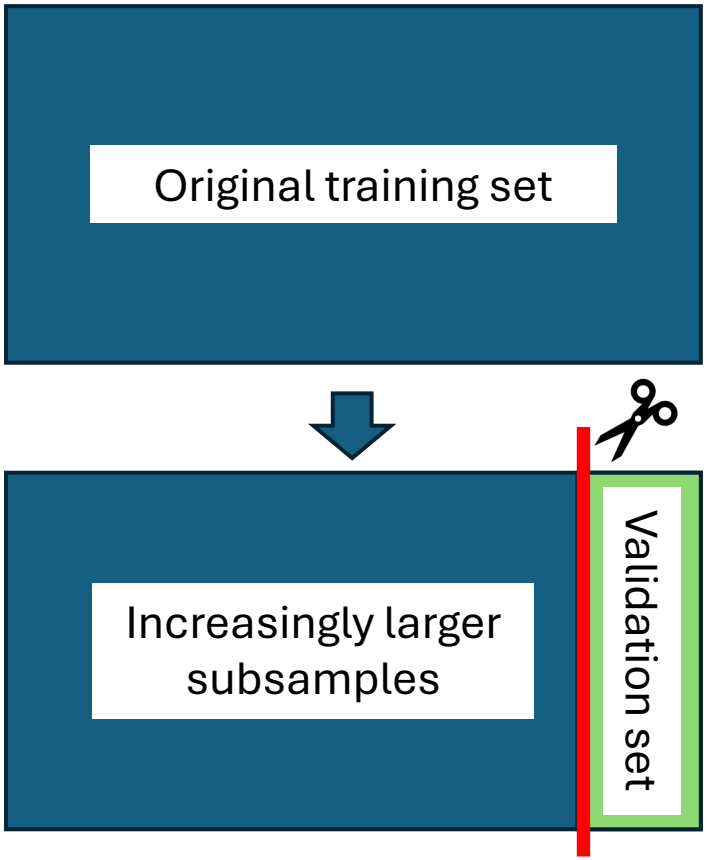
- Choose your measures/"metrics"
  - Regression
    - Errors: MAE, MSE, RMSE, MAPE, …
    - Correlation coefficients: Pearson, Spearman, Kendal, Concordance (CCC)
    - $R^2$
  - Classification
    - Confusion matrix-based: F1-score, precision, recall (TPR, sensitivity), accuracy, … `[NOT USED IN THE CURRENT CODE -- NA]`

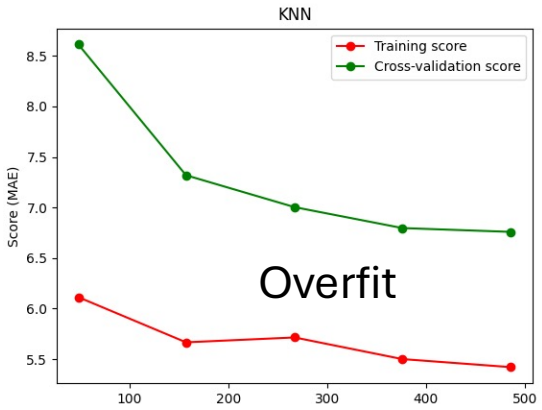
# Regression pipeline

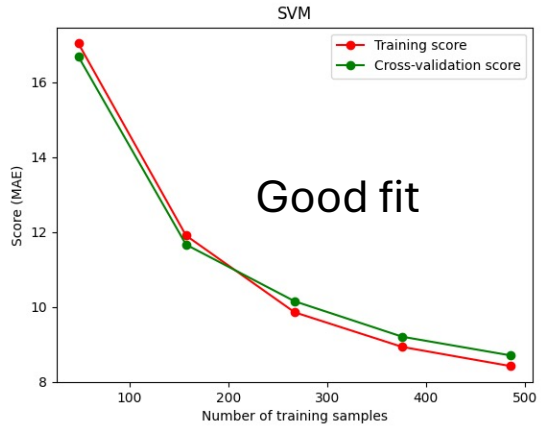
1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. **Overfitting analysis of default models**
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Overfitting analysis

- Use learning curves
  - training vs. validation scores for increasing training set sizes



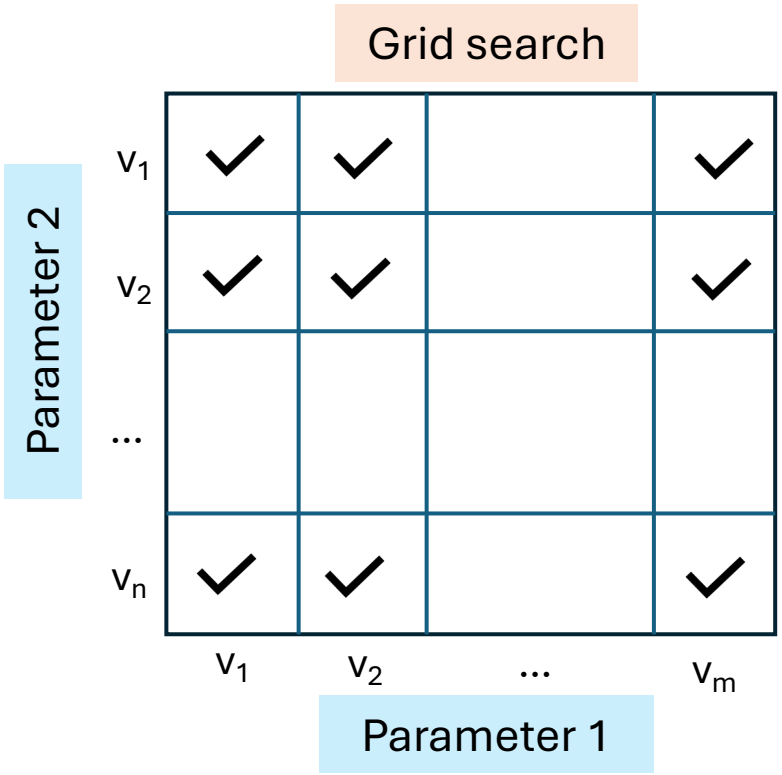



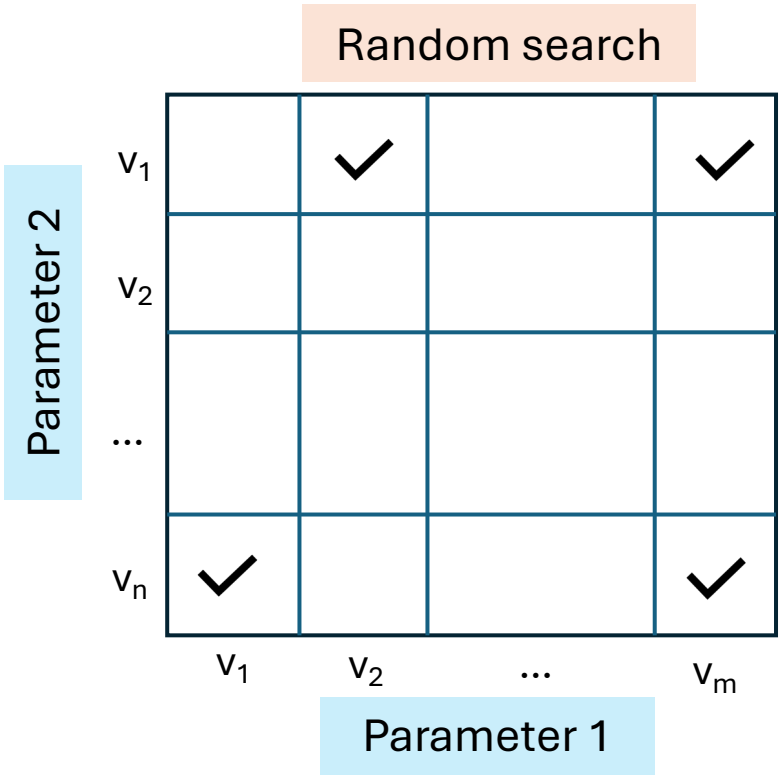
# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. **Hyper-parameter optimization**
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Hyper-parameter optimization

- Hyper-parameter = user-tunable parameter

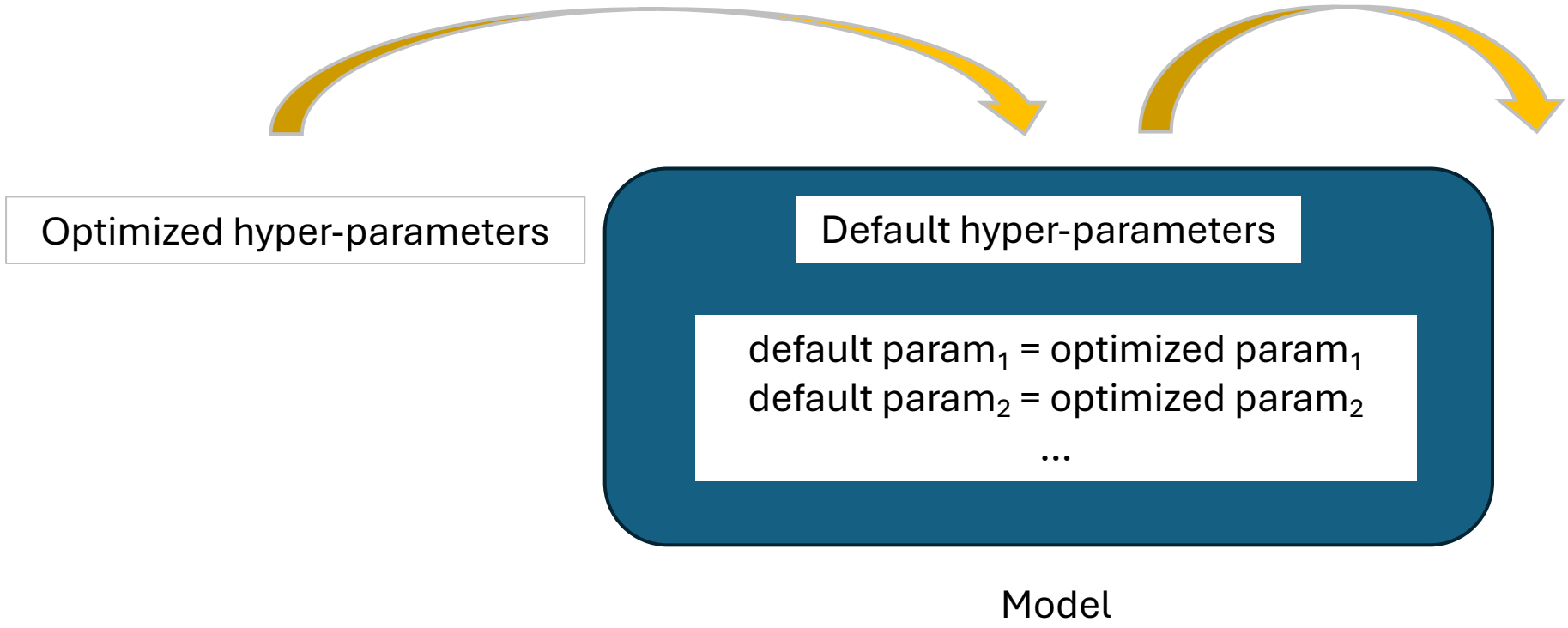


# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. **Update model hyper-parameters**
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Hyper-parameters' update



# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. **Evaluate optimized models**
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Model evaluation (same as for 7)

- K-fold cross-validation
  - Choose K as a function of data size and computing time
    - High K values: small-medium datasets
    - Low  K values: large datasets

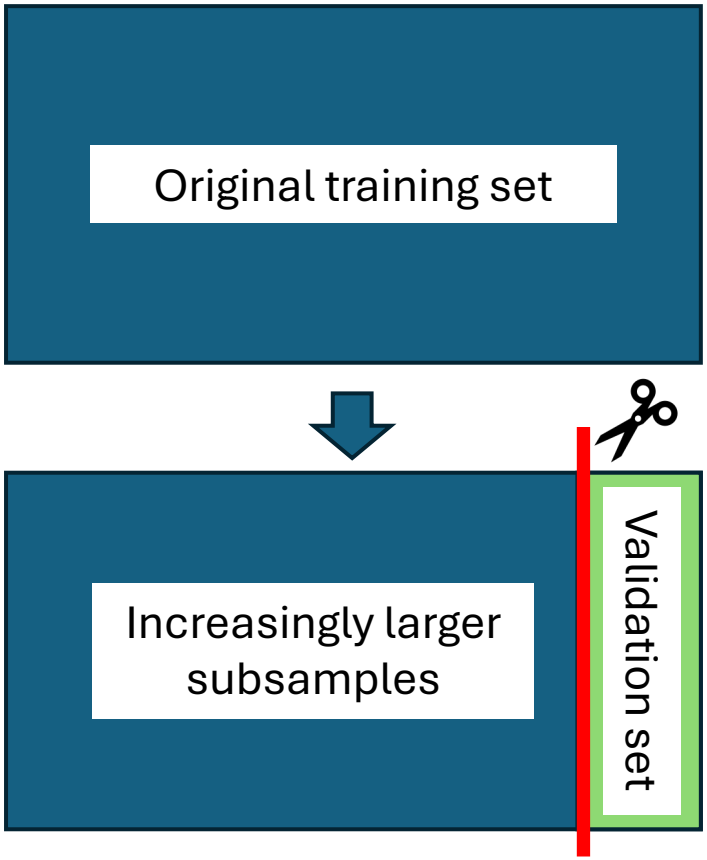
- Choose your measures/"metrics"
  - Regression
    - Errors: MAE, MSE, RMSE, MAPE, …
    - Correlation coefficients: Pearson, Spearman, Kendal, Concordance (CCC)
    - $R^2$
  - Classification
    - Confusion matrix-based: F1-score, precision, recall (TPR, sensitivity), accuracy, … `[NOT USED IN THE CURRENT CODE -- NA]`

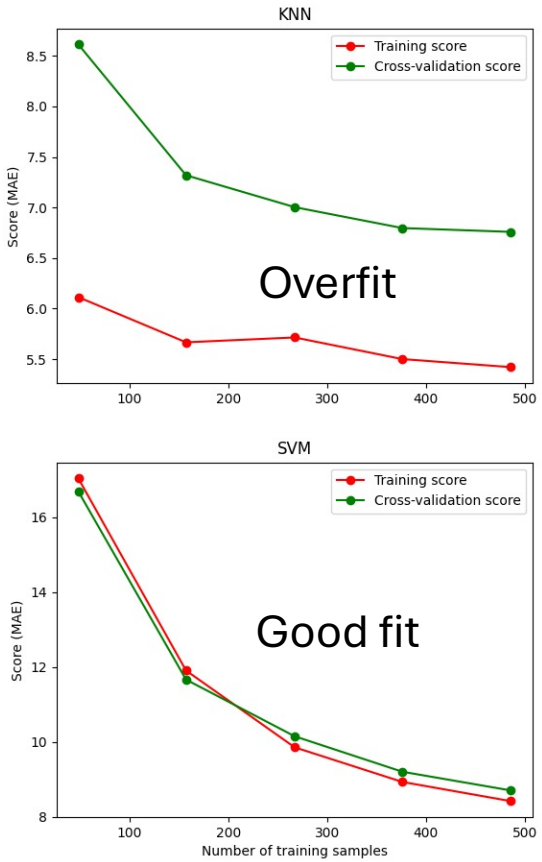
# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. **Overfitting analysis of optimized models**
13. Save optimized models
14. Investigate feature importance
15. Evaluate models on test sets

# Overfitting analysis (same as for 8)

- Use learning curves
  - training vs. validation scores for increasing training set sizes



# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. **Save optimized models**
14. Investigate feature importance
15. Evaluate models on test sets

# Saving models

- Backup all optimized models
- Can be used later for deployment
- Save time on re-training and re-optimizing hyper-parameters

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. **Investigate feature importance**
15. Evaluate models on test sets

# Feature importance

- Use a model-agnostic process

- Permutation Feature Importance (PFI)
  - Shuffle one variable at a time
  - Evaluate each algorithm
  - Idea: if an important variable is shuffled it would hurt the model significantly (poor predictions)

- Other options: SHAPley values

# Regression pipeline

1. Data cleaning
2. Data summarization
3. Data visualization
4. Data splitting
5. Data scaling
6. Model initialization (default params)
7. Preliminary model evaluation
8. Overfitting analysis of default models
9. Hyper-parameter optimization
10. Update model hyper-parameters
11. Evaluate optimized models
12. Overfitting analysis of optimized models
13. Save optimized models
14. Investigate feature importance
15. **Evaluate models on test sets**

# Model evaluation on test sets

- Use various evaluation measures
  - Error-based: MAE, MSE, RMSE, MAPE
  - Correlations: Pearson Product-Moment, Concordance, Spearman
  - (Adjusted) Coefficient of determination

  Note: no single evaluation measure captures everything

- Use visual analysis, too
  - Scatter plots (predicted versus true values)
  - QQ plots for prediction errors

# Thank you