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

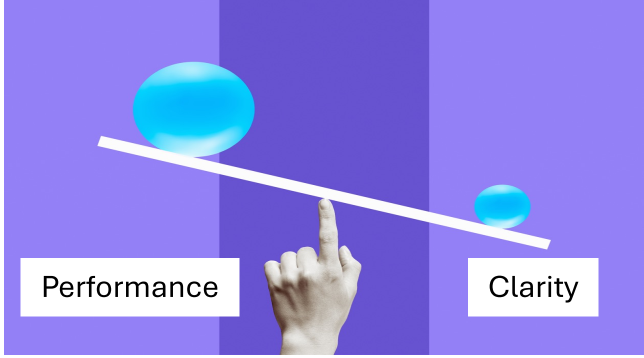
University of Guelph, Ontario, Canada

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_selected_measurements.csv

- 4 input variables (all numeric):
  - heartgirth
  - height
  - length
  - body_condition_score
- 1 output variable:
  - exact_weight

MarshallEtAl2023_selected_measurements

heartgirth	height	length	body_condition_score	exact_weight
81	50.1	95	3	42.7
59	53	64	3	16
59	53	64	3	16
26	17	26	3	1.7
27	17	28	3	1.8
28	21	27.5	3	1.9
99	65	111	3	64.1
62	42	67	3	18.7
34	23	39	2	3.5
37	25	39	3	4.1
97	58	101	3	51.4
93	56	96	4	54.5
92	57	96	3	50.3
89	54	94	4	46.4

MarshallEtAl2023_more_selected_measurements.csv

- 6 input variables (all numeric):
  - household_id
  - age_months
  - heartgirth
  - height
  - length
  - body_condition_score
- 1 output variable:
  - exact_weight

MarshallEtAl2023_more_selected_measurements

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

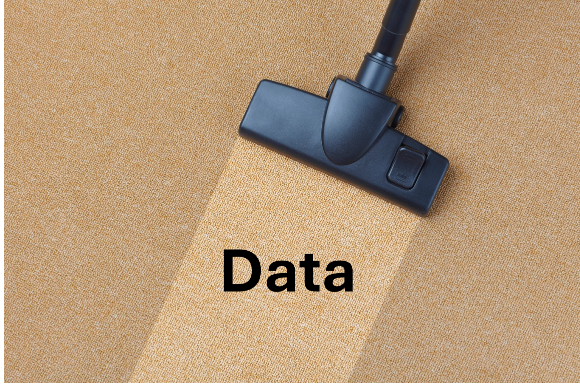
Regression pipeline

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

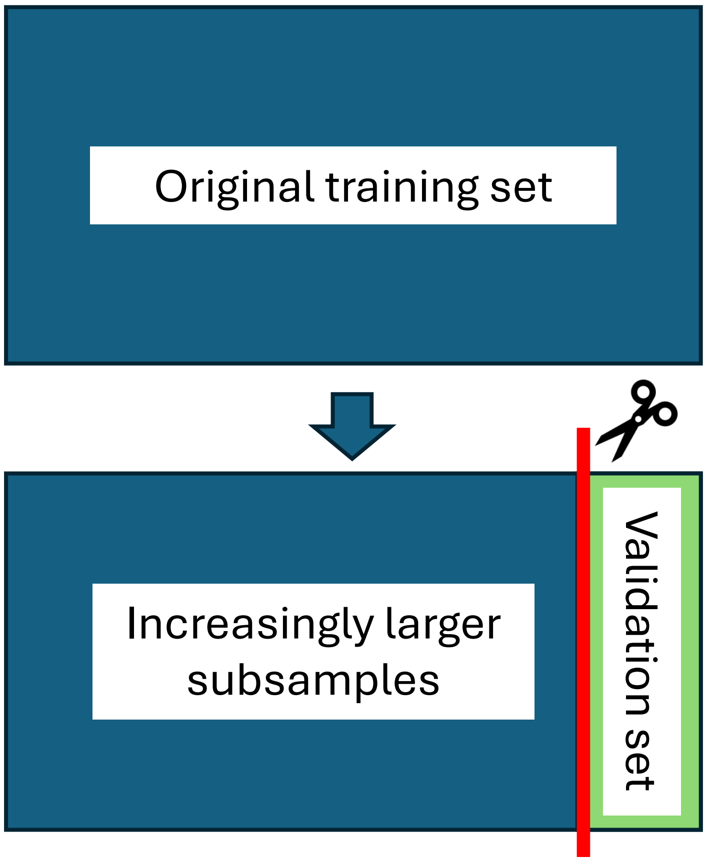
- Choose your measures/”metrics”
  - Regression
    - Errors: MAE, MSE, RMSE, MAPE, …
    - Correlation coefficients: Pearson, Spearman, Kendal, Concordance (CCC)
    - R^2
  - Classification
    - Confusion matrix-based: F1-score, precision, recall (TPR, sensitivity), accuracy, … `[NOT USED IN THE CURRENT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

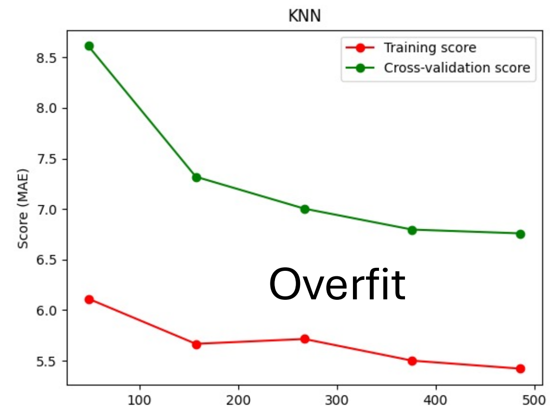
Regression pipeline

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



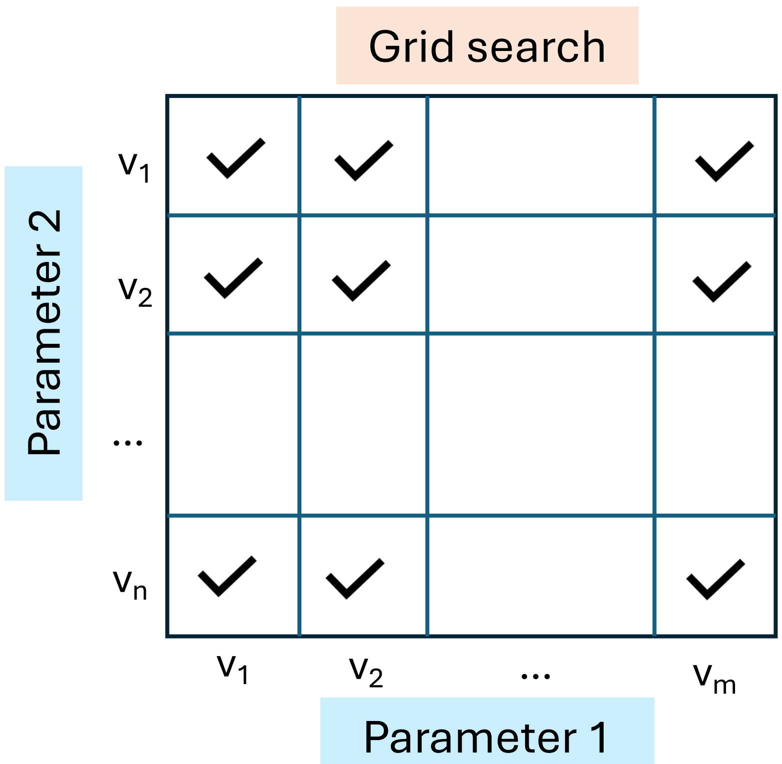


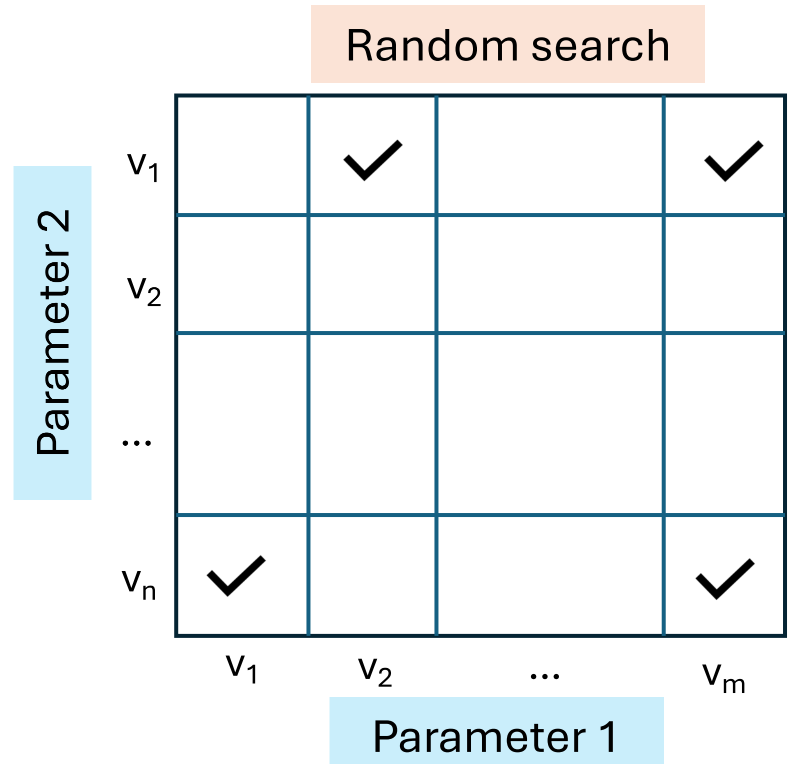
Regression pipeline

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

Optimized hyper-parameters

Default hyper-parameters

default param_1 = optimized param_1
default param_2 = optimized param_2
...

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

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

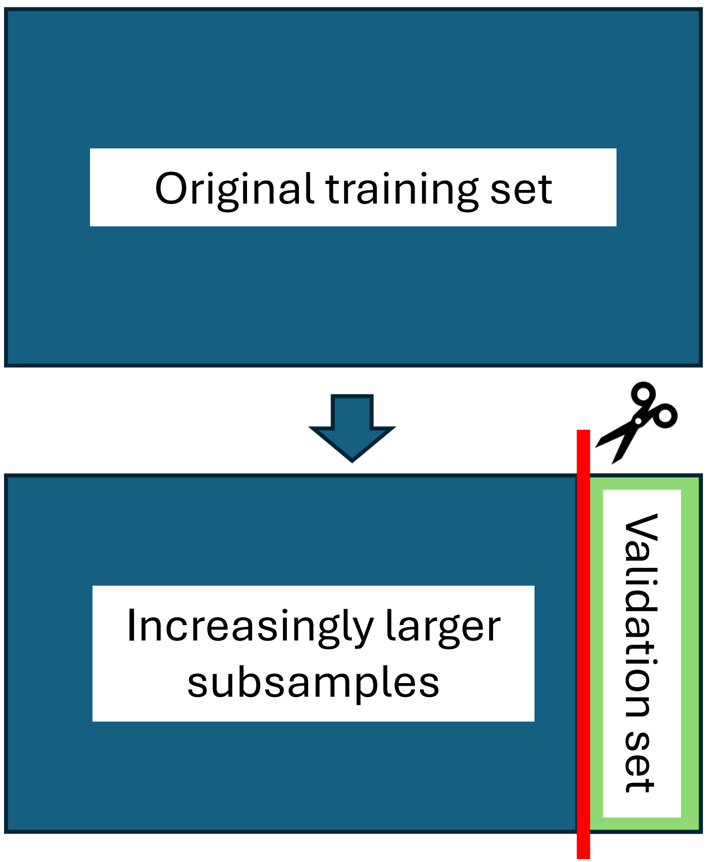
- Choose your measures/”metrics”
  - Regression
    - Errors: MAE, MSE, RMSE, MAPE, …
    - Correlation coefficients: Pearson, Spearman, Kendal, Concordance (CCC)
    - R^2
  - Classification
    - Confusion matrix-based: F1-score, precision, recall (TPR, sensitivity), accuracy, … `[NOT USED IN THE CURRENT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

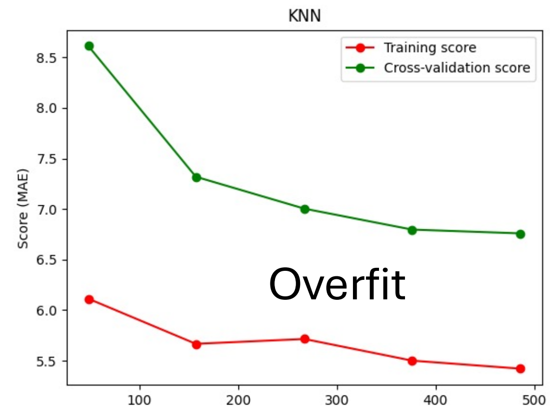
Regression pipeline

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