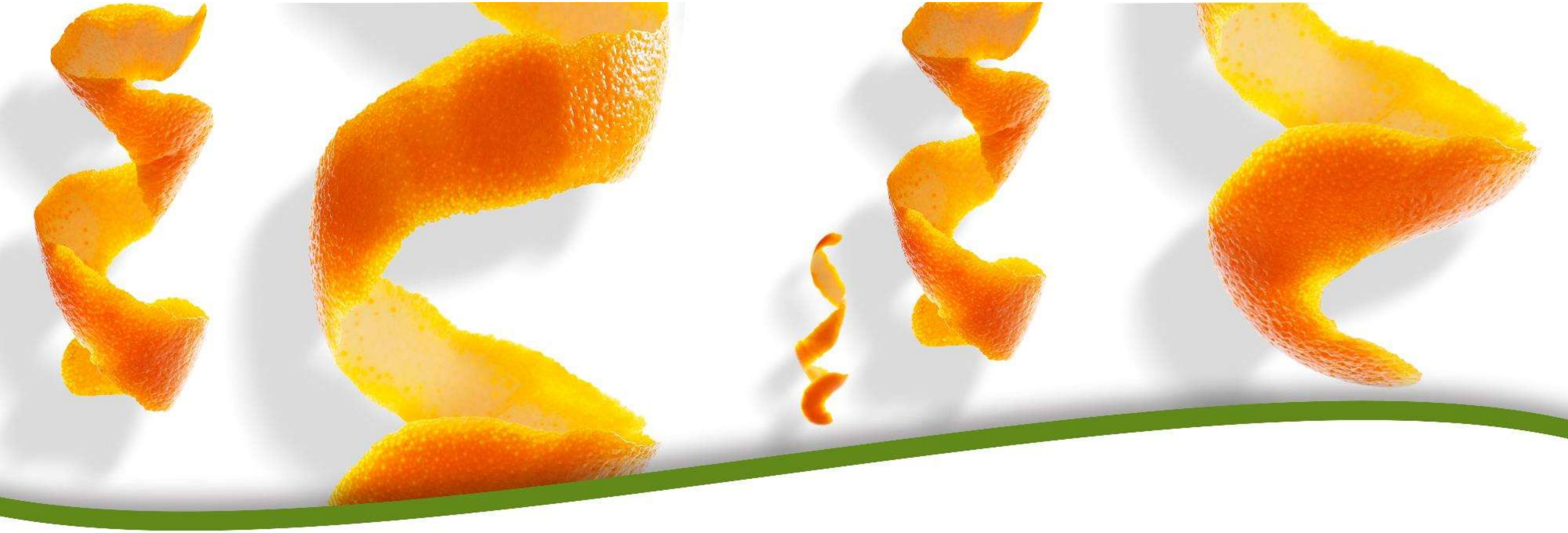


Enhancing Predictions:
A Comparative Study of Machine Learning Models and Classical Methods

dsk.2024, Nov. 8th, Sune Petersen

Unlocking Nature-Powered Success®

A HUBER COMPANY

t: +45 5616 5858
e: sune.petersen@cpkelco.com

Outline

- About CP Kelco
- Raw data used for training
- Machine Learning Apps and models tested
- Data structure and pre-processing
- Regression Learner App results
- PLS Toolbox App results
- What is XGB?
- Learnings and next step
- Questions?

About CP Kelco

CP Kelco is a leading global producer and innovator of **nature-based** specialty ingredients made from:

Microbial Fermentation	Extraction from Land and Sea Plants	Dairy Derived
Xanthan Gum	Pectin	Whey Protein Concentrate
Gellan Gum	Carrageenan	
Diutan Gum	Citrus Fiber	
Fermentation-Derived Cellulose	Locust Bean Gum	

Decades of Industry and Product Experience

Extensive portfolio of versatile, high-performance solutions.

Industry renowned **technical support, quality** and **safety** standards.

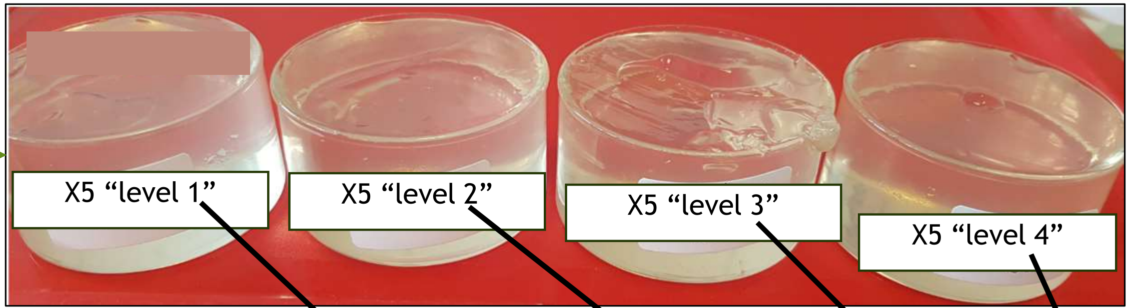
Modeling from QC data (gel application)

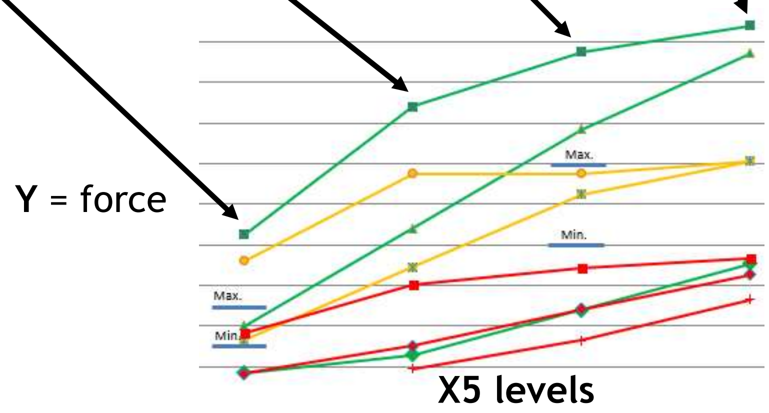
Sample

Test gel

Measurements on sample:
X1, X2 = fast = every day!
X3 = slow = not measured daily

QC - gel test system:
X4 = test system (3 to choose from)
X5 = 4 levels of a gel-component



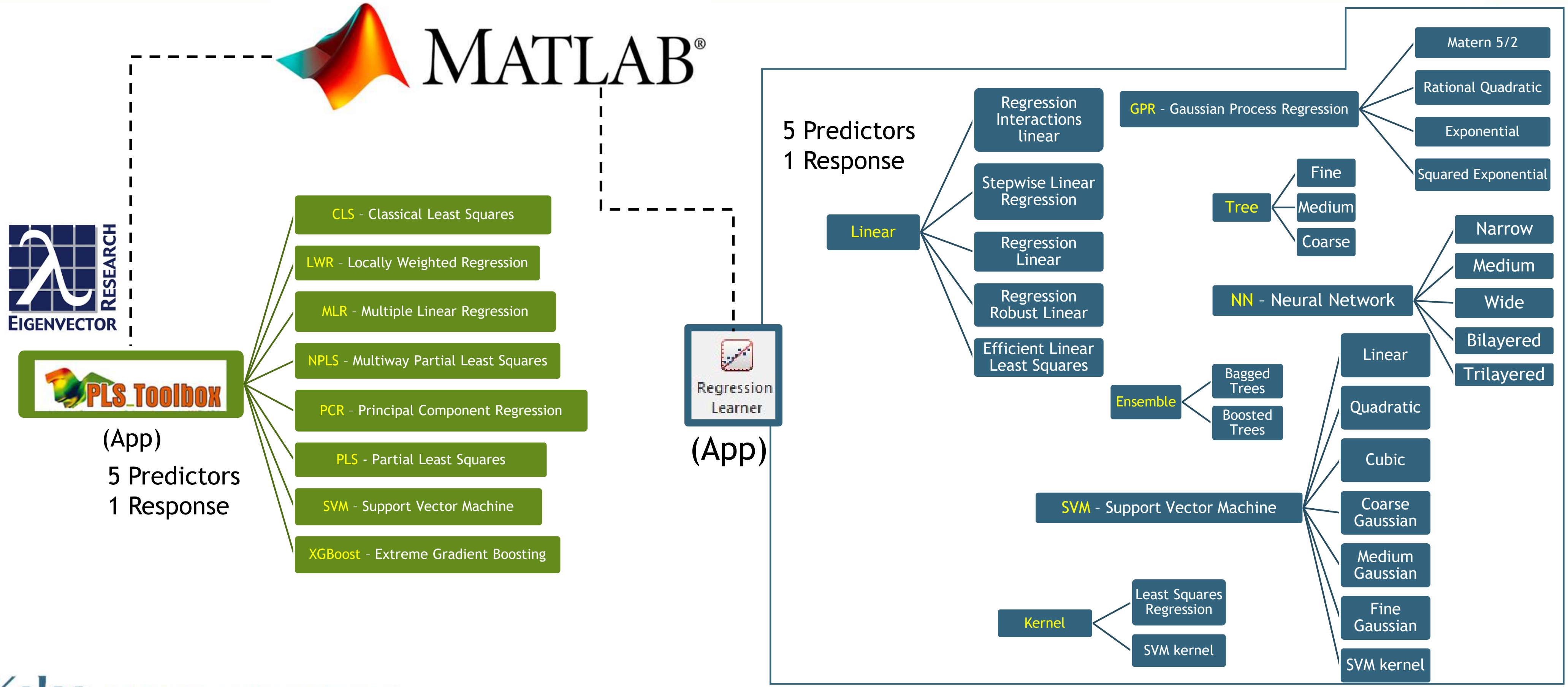


Classical approach: 4 models (4Y)

New approach: 1 model?? (1Y)

X = predictor(s) , Y = response(s)

Machine Learning Apps and regression models tested



CPKelco A HUBER COMPANY Unlocking Nature-Powered Success®

Data structure and pre-processing

Data

- **5 Discrete Predictors** (X1, X2, X3, X4, and X5) + **1 Response** (Force)
- **No "Outliers" removed!**

Pre-processing

- Data was **Autoscaled** prior to training and testing

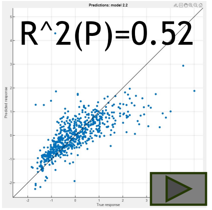
(Value – Mean)/SD

- All **gel-test measurements** (= X and Y data set) were **randomized before splitting** into
  85% for training/cross validation (= **3584** gel-test measurements)
  15% for testing (**633** gel-test measurements)

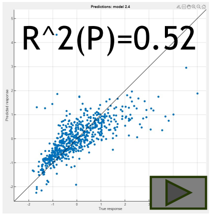
Regression Learner App results

Regression Learner

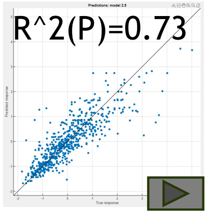
Linear Regression (1805)



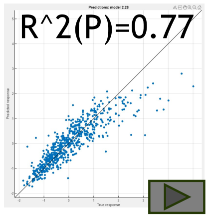
Stepwise Linear (1960)



Tree (1963)

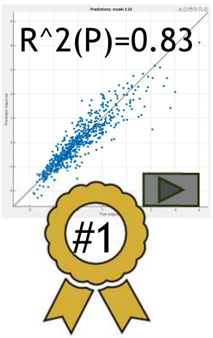


Kernel (1964)



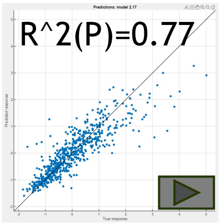
(= Transforming data to higher dimmensional space)

NN ("1980's")



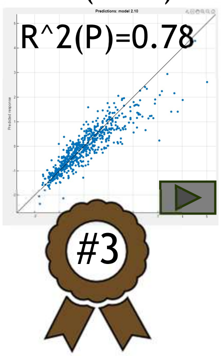
(Neural Network)

Ensemble (1990's)



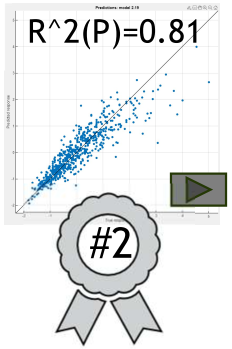
(= Combines multiple individual models)

SVM (1992)



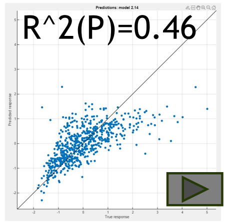
(Support Vector Machine)

GPR (1996)



(Gaussian Process Regression)

Efficient linear (????)

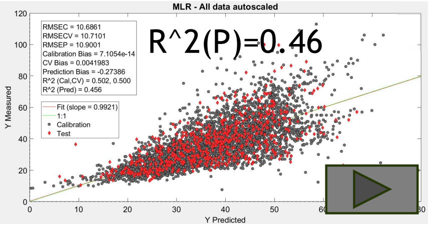


Unlocking Nature-Powered Success®

R^2(P)= "prediction power on new data after model training"

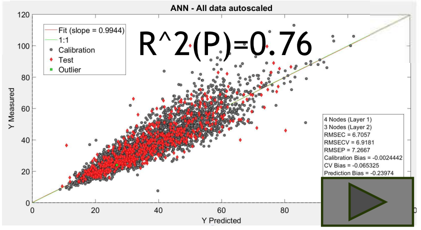
PLS Toolbox App results

MLR (year 1805)



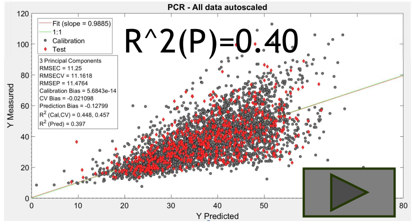
(Multiple Linear Regression)

ANN (year 1943)



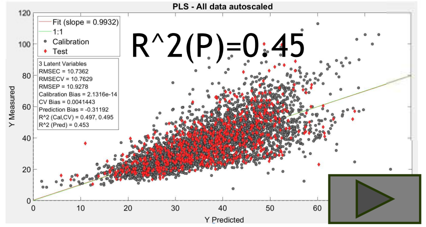
(Artificial Neural Network)

PCR (year 1960)



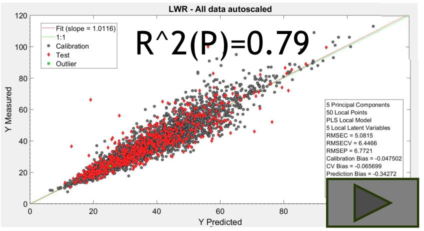
(Principal Component Regression)

PLS (year 1966)



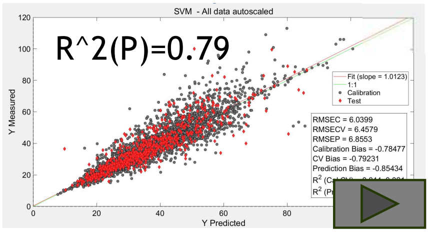
(Partial Least Squares)

LWR (year 1988)



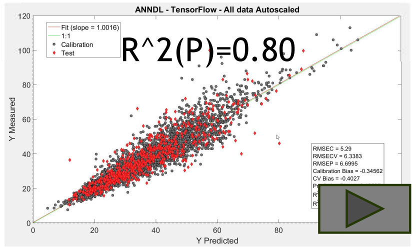
(Locally Weighted Regression)

SVM (year 1992)



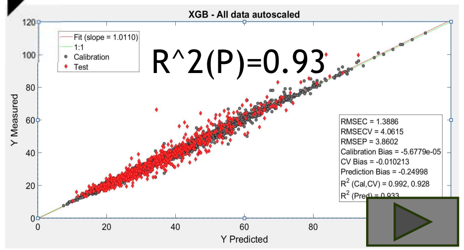
(Support Vector Machine)

ANNDL (year 2006)



(Artificial Neural Network Deep Learning)

XGB (year 2014)



(Extreme Gradient Boosting)

R^2(P)= "prediction power on new data after model training"

About XGB

What is XGB?

Extreme Gradient Boosting, commonly known as **XGBoost** (**XGB**), is a **powerful machine learning algorithm** that has **gained popularity for its efficiency and performance** in predictive modeling tasks.

How XGBoost Works

1. **Boosting Technique:** XGBoost is based on the boosting technique, where **multiple weak learners** (usually **decision trees**) are **combined to form a strong learner**. Each new tree corrects the errors made by the previous trees.
2. **Gradient Descent:** It uses gradient descent to minimize the loss function. The algorithm **iteratively adds trees to the model**, each one aiming **to reduce the residual errors of the previous trees**.
3. **Regularization:** XGBoost includes regularization terms in its objective function to **prevent overfitting**. This makes it **more robust compared to other boosting algorithms**.
4. **Parallel Processing:** It supports parallel processing, which speeds up the training process significantly.
5. **Handling Missing Values:** XGBoost can handle missing values internally, making it more versatile for real-world data.

Pros

- **High Performance:** XGBoost **often outperforms other algorithms** in terms of **accuracy** and **speed**.
- **Flexibility:** It can be used for both **classification** and **regression tasks**.
- **Feature Importance:** Provides **insights into feature importance**, helping in feature selection.
- **Scalability:** Efficiently handles **large datasets** and can be distributed across clusters.
- **Regularization:** Built-in regularization helps in **preventing overfitting**.

Cons

- **Complexity:** The algorithm can be complex to tune due to the large number of hyperparameters.
- **Computationally Intensive:** Despite its efficiency, it can still be computationally intensive, especially for very large datasets.
- **Interpretability:** Models can be less interpretable compared to simpler algorithms like linear regression or decision trees.

Conclusion and next step

- **Be creative with your data and how you use them** (5X+1Y *vs.* 4X+4Y)
- **Test many different models** to increase chances to get Great prediction power!
- **Build fewer but more flexible models if possible** (*vs.* interpolation /maintenance)
- **XGB** (Extreme Gradient Boosting) **is an interesting option for complex data!**

- **Next Step:** Try XGB-prediction from only spectral X-predictors (= “instant predictions”)

Many thanks 🙏🤗
Now time for Questions…

Experiences with XGB?

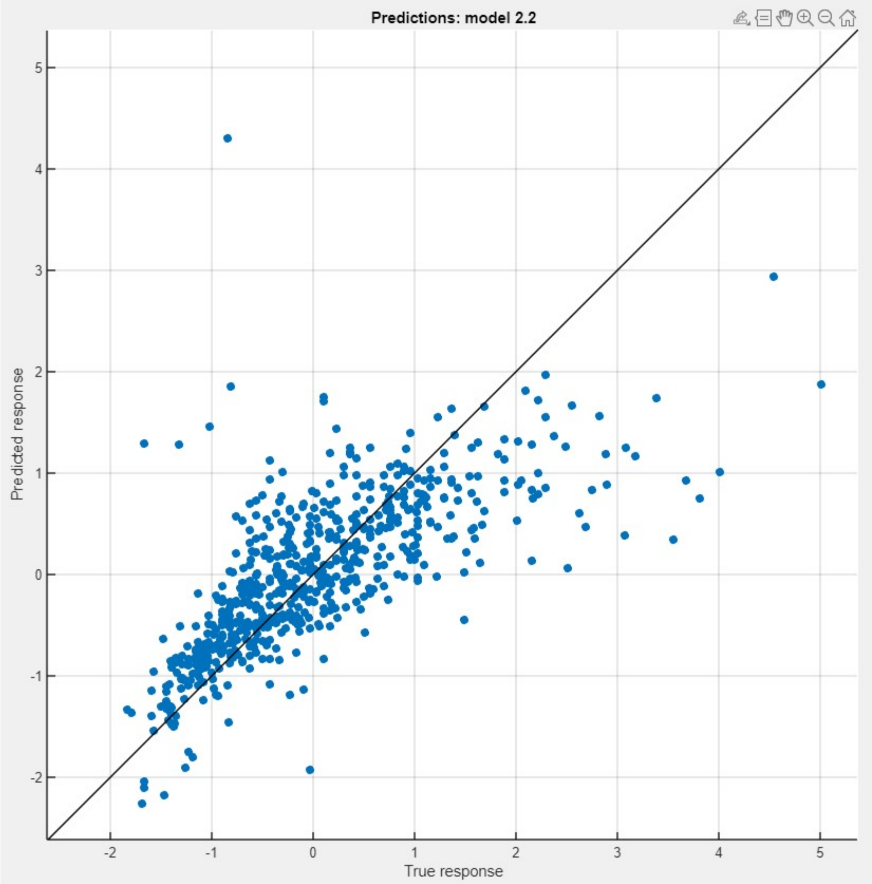
Other ideas/models to try out?

Other input/feedback?

Thanks for your attention! 😃

Linear Regression (1805)

2.2 Linear Regression
Last change: Interactions Linear



Model 2.2: Linear Regression
Status: Tested

Training Results

RMSE (Validation)	0.63596
R-Squared (Validation)	0.59
MSE (Validation)	0.40444
MAE (Validation)	0.45746
MAPE (Validation)	167.9%
Prediction speed	~100000 obs/sec
Training time	9.2049 sec
Model size (Compact)	~12 kB

Test Results

RMSE (Test)	0.72135
R-Squared (Test)	0.52
MSE (Test)	0.52034
MAE (Test)	0.49084
MAPE (Test)	163.6%

Model Hyperparameters

Preset: Interactions Linear
Terms: Interactions
Robust option: Off

Feature Selection: 5/5 individual features selected

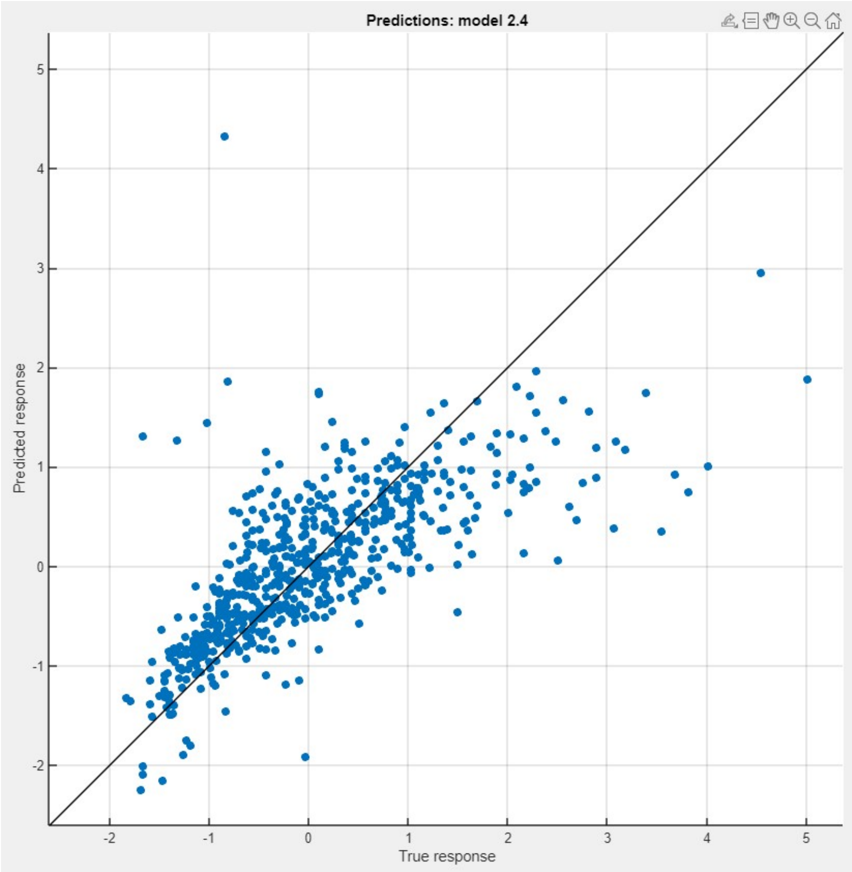
PCA: Disabled

Optimizer: Not applicable

CPKelco A HUBER COMPANY Unlocking Nature-Powered Success®

Stepwise Linear (1960)

2.4 Stepwise Linear Regression
Last change: Stepwise Linear



Model 2.4: Stepwise Linear Regression
Status: Tested

Training Results

RMSE (Validation)	0.63507
R-Squared (Validation)	0.59
MSE (Validation)	0.40331
MAE (Validation)	0.45741
MAPE (Validation)	168.0%
Prediction speed	~100000 obs/sec
Training time	11.004 sec
Model size (Compact)	~12 kB

Test Results

RMSE (Test)	0.72157
R-Squared (Test)	0.52
MSE (Test)	0.52067
MAE (Test)	0.49105
MAPE (Test)	163.5%

Model Hyperparameters

Preset: Stepwise Linear
Initial terms: Linear
Upper bound on terms: Interactions
Maximum number of steps: 1000

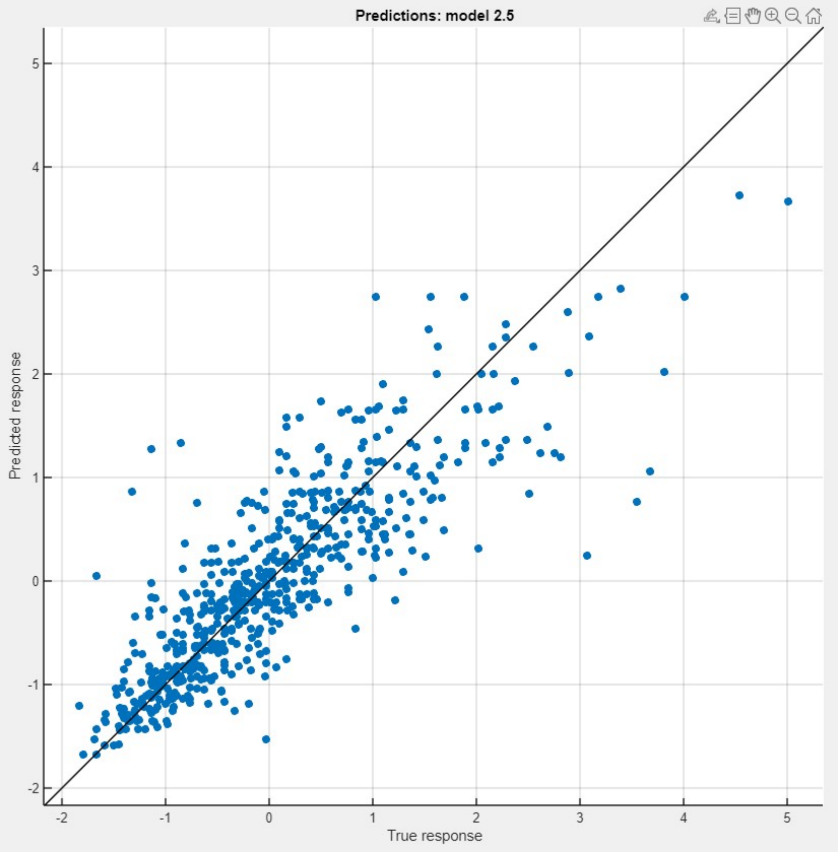
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

Tree (1963)

2.5 Tree
Last change: Fine Tree



Model 2.5: Tree
Status: Tested

Training Results

RMSE (Validation)	0.53499
R-Squared (Validation)	0.71
MSE (Validation)	0.28621
MAE (Validation)	0.37970
MAPE (Validation)	159.8%
Prediction speed	~130000 obs/sec
Training time	12.949 sec
Model size (Compact)	~148 kB

Test Results

RMSE (Test)	0.54773
R-Squared (Test)	0.73
MSE (Test)	0.30001
MAE (Test)	0.37929
MAPE (Test)	137.2%

Model Hyperparameters

Preset: Fine Tree
Minimum leaf size: 4
Surrogate decision splits: Off

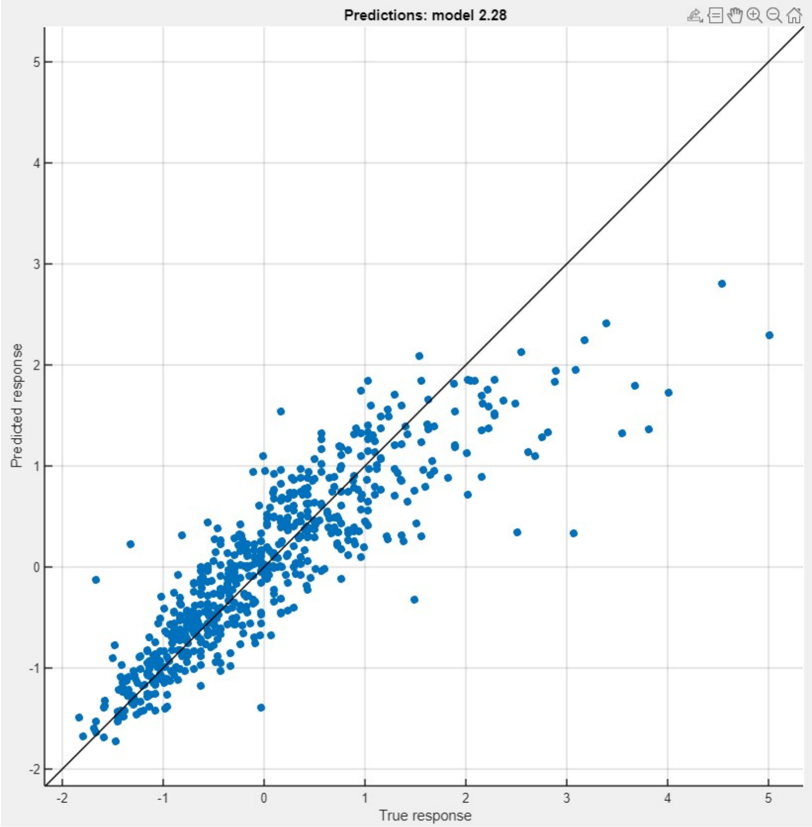
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

Kernel (1964)

2.28 Kernel
Last change: Least Squares Regression Kernel



Model 2.28: Kernel
Status: Tested

Training Results

RMSE (Validation)	0.46964
R-Squared (Validation)	0.78
MSE (Validation)	0.22056
MAE (Validation)	0.33827
MAPE (Validation)	140.4%
Prediction speed	~110000 obs/sec
Training time	667.21 sec
Model size (Compact)	~10 kB

Test Results

RMSE (Test)	0.50224
R-Squared (Test)	0.77
MSE (Test)	0.25224
MAE (Test)	0.35226
MAPE (Test)	138.9%

Model Hyperparameters

Preset: Least Squares Regression Kernel
Learner: Least Squares Kernel
Number of expansion dimensions: Auto
Regularization strength (Lambda): Auto
Kernel scale: Auto
Standardize data: Yes
Iteration limit: 1000

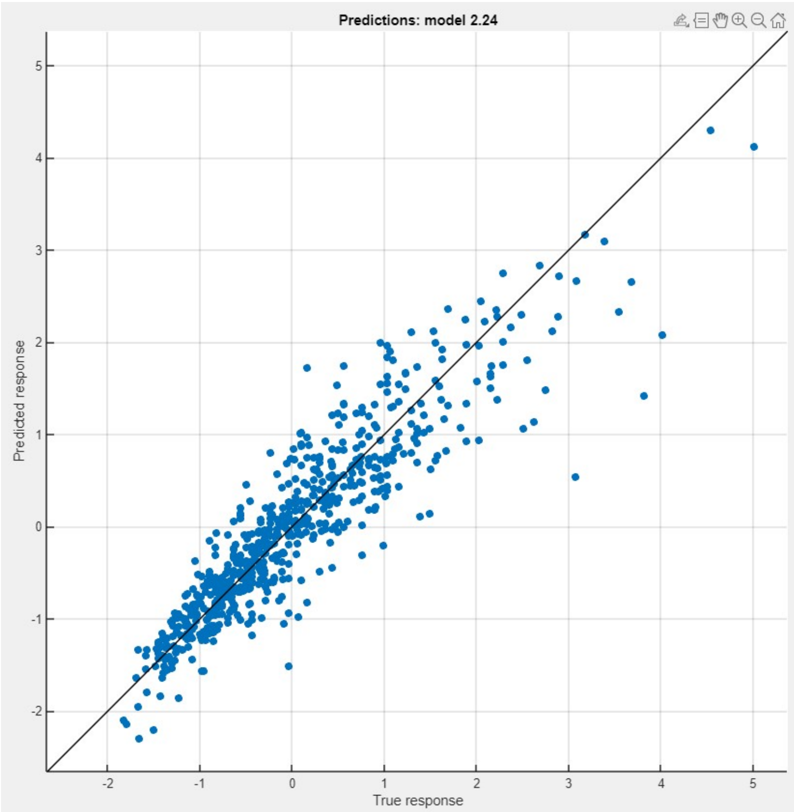
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

NN (”1980’s”)

2.24 Neural Network
Last change: Wide Neural Network



Model 2.24: Neural Network
Status: Tested

Training Results

RMSE (Validation)	0.42570
R-Squared (Validation)	0.82
MSE (Validation)	0.18122
MAE (Validation)	0.30569
MAPE (Validation)	126.2%
Prediction speed	~190000 obs/sec
Training time	646.51 sec
Model size (Compact)	~11 kB

Test Results

RMSE (Test)	0.43306
R-Squared (Test)	0.83
MSE (Test)	0.18754
MAE (Test)	0.30892
MAPE (Test)	131.2%

Model Hyperparameters

Preset: Wide Neural Network
Number of fully connected layers: 1
First layer size: 100
Activation: ReLU
Iteration limit: 1000
Regularization strength (Lambda): 0
Standardize data: Yes

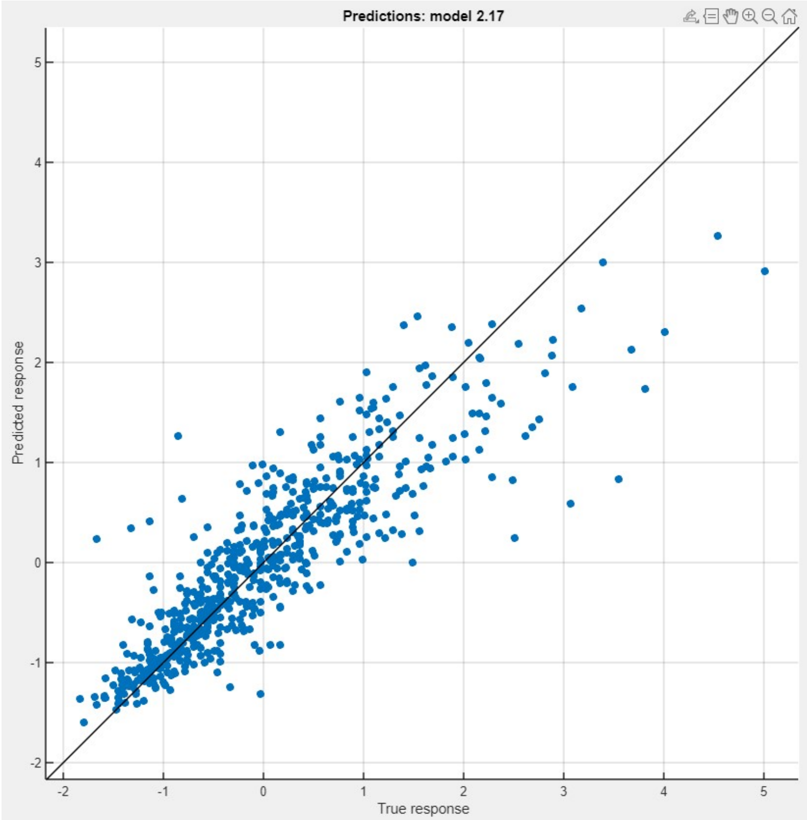
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

Ensemble (1990’s)

2.17 Ensemble
Last change: Bagged Trees



Model 2.17: Ensemble
Status: Tested

Training Results

RMSE (Validation)	0.46659
R-Squared (Validation)	0.78
MSE (Validation)	0.21771
MAE (Validation)	0.33402
MAPE (Validation)	134.9%
Prediction speed	~20000 obs/sec
Training time	53.499 sec
Model size (Compact)	~2 MB

Test Results

RMSE (Test)	0.50089
R-Squared (Test)	0.77
MSE (Test)	0.25089
MAE (Test)	0.34581
MAPE (Test)	139.1%

Model Hyperparameters

Preset: Bagged Trees
Minimum leaf size: 8
Number of learners: 30
Number of predictors to sample: Select All

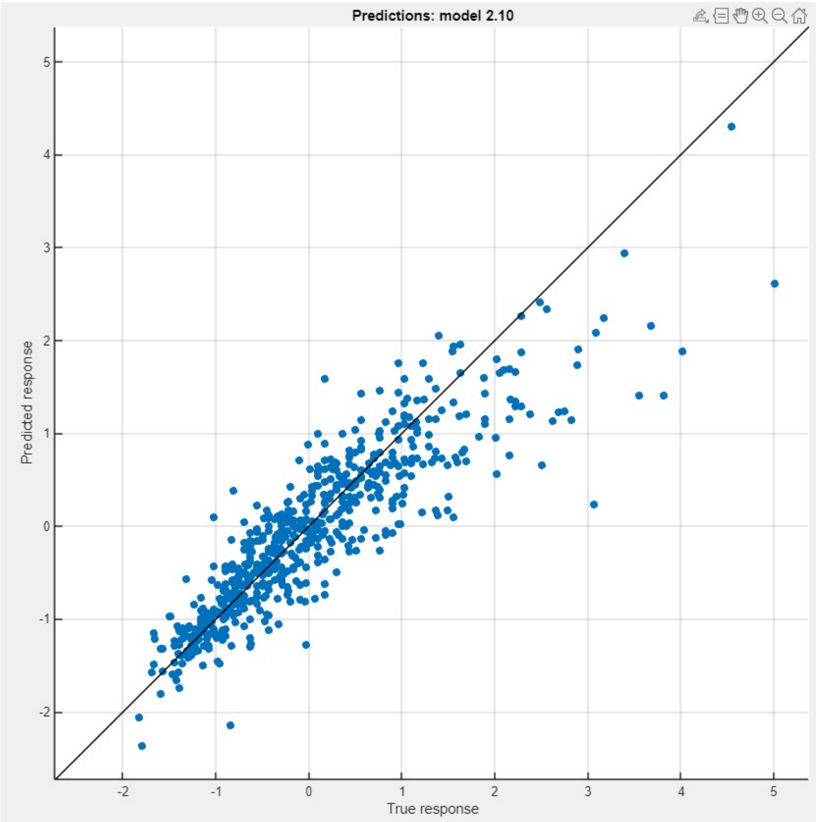
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

SVM (1992)

2.10 SVM
Last change: Cubic SVM



Model 2.10: SVM
Status: Tested

Training Results

RMSE (Validation)	0.46295
R-Squared (Validation)	0.78
MSE (Validation)	0.21432
MAE (Validation)	0.32091
MAPE (Validation)	132.8%
Prediction speed	~89000 obs/sec
Training time	38.906 sec
Model size (Compact)	~133 kB

Test Results

RMSE (Test)	0.48762
R-Squared (Test)	0.78
MSE (Test)	0.23777
MAE (Test)	0.33717
MAPE (Test)	125.4%

Model Hyperparameters

Preset: Cubic SVM
Kernel function: Cubic
Kernel scale: Automatic
Box constraint: Automatic
Epsilon: Auto
Standardize data: Yes

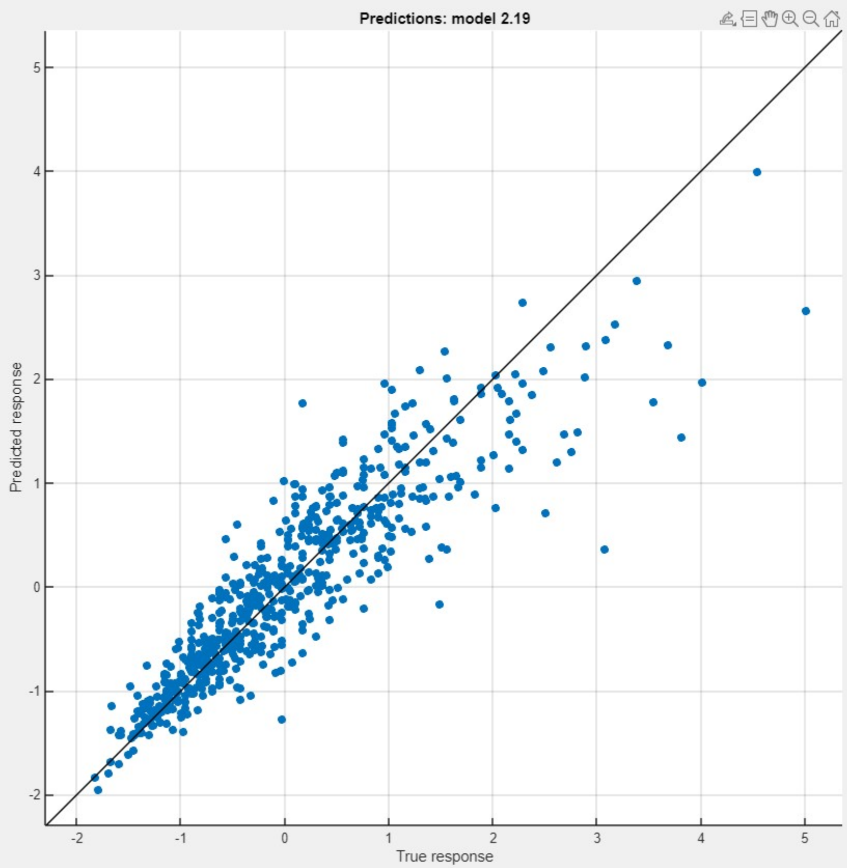
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

GPR (1996)

2.19 Gaussian Process Regression
Last change: Matern 5/2 GPR



Model 2.19: Gaussian Process Regression
Status: Tested

Training Results

RMSE (Validation)	0.42413
R-Squared (Validation)	0.82
MSE (Validation)	0.17989
MAE (Validation)	0.29945
MAPE (Validation)	132.1%
Prediction speed	~18000 obs/sec
Training time	253.48 sec
Model size (Compact)	~181 kB

Test Results

RMSE (Test)	0.45062
R-Squared (Test)	0.81
MSE (Test)	0.20306
MAE (Test)	0.31148
MAPE (Test)	131.4%

Model Hyperparameters

Preset: Matern 5/2 GPR
Basis function: Constant
Kernel function: Matern 5/2
Use isotropic kernel: Yes
Kernel scale: Automatic
Signal standard deviation: Automatic
Sigma: Automatic
Standardize data: Yes
Optimize numeric parameters: Yes

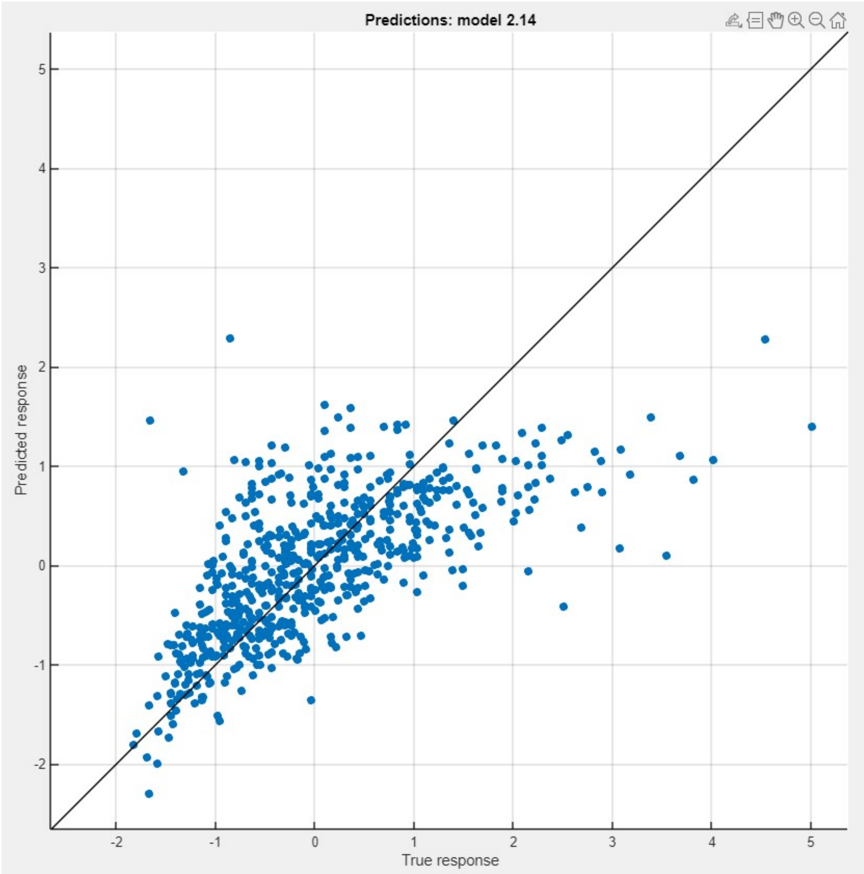
Feature Selection: 5/5 individual features selected

PCA: Disabled

Optimizer: Not applicable

Efficient linear (????)

2.14 Efficient Linear
Last change: Efficient Linear Least Squares



Model 2.14: Efficient Linear
Status: Tested

Training Results

RMSE (Validation)	0.70087
R-Squared (Validation)	0.50
MSE (Validation)	0.49122
MAE (Validation)	0.52808
MAPE (Validation)	179.7%
Prediction speed	~240000 obs/sec
Training time	47.328 sec
Model size (Compact)	~11 kB

Test Results

RMSE (Test)	0.77095
R-Squared (Test)	0.46
MSE (Test)	0.59436
MAE (Test)	0.55712
MAPE (Test)	183.3%

Model Hyperparameters

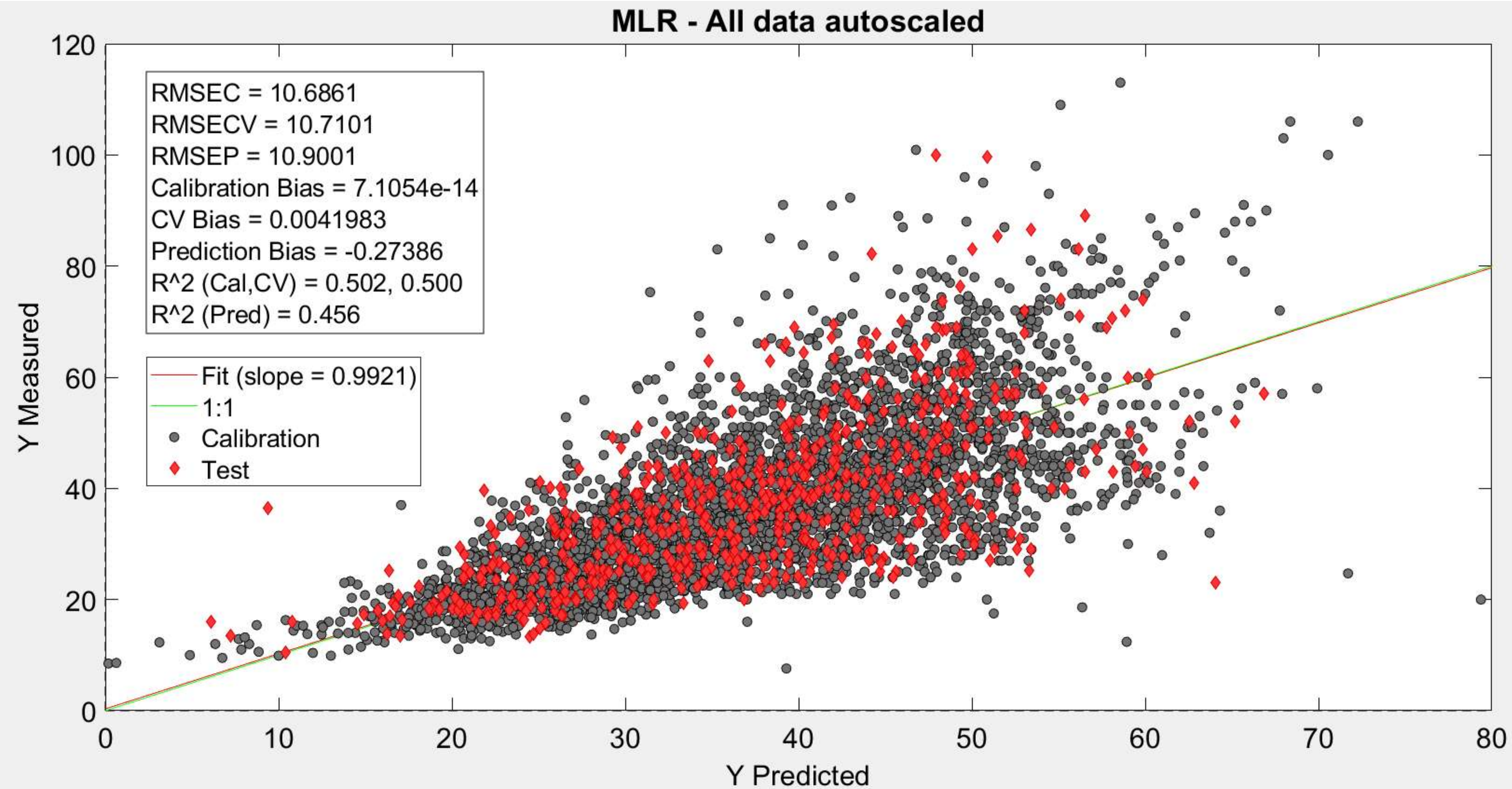
Preset: Efficient Linear Least Squares
Learner: Least squares
Solver: Auto
Regularization: Auto
Regularization strength (Lambda): Auto
Relative coefficient tolerance (Beta tolerance): 0.0001

Feature Selection: 5/5 individual features selected

PCA: Disabled

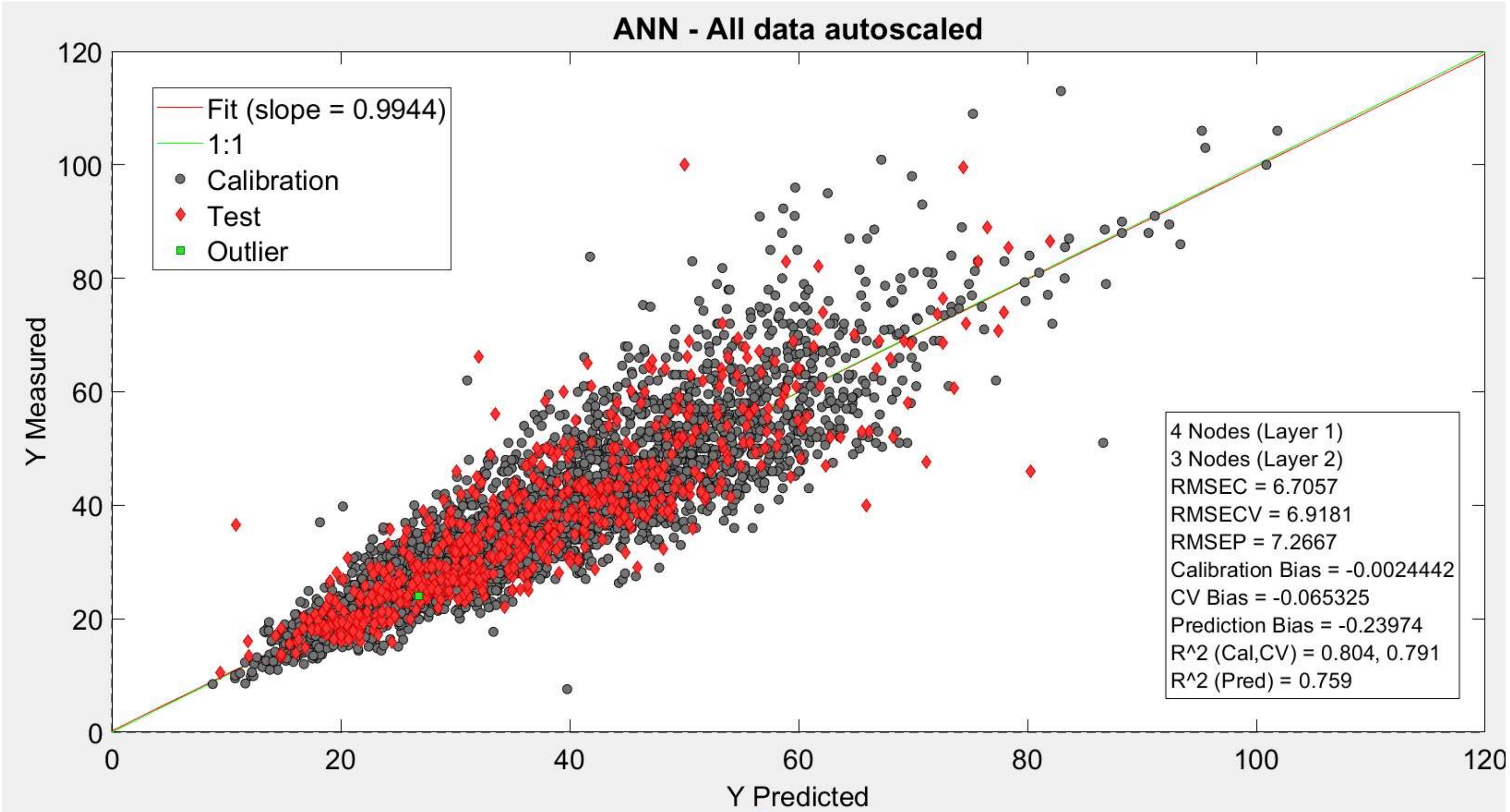
Optimizer: Not applicable

MLR – Multiple Linear Regression

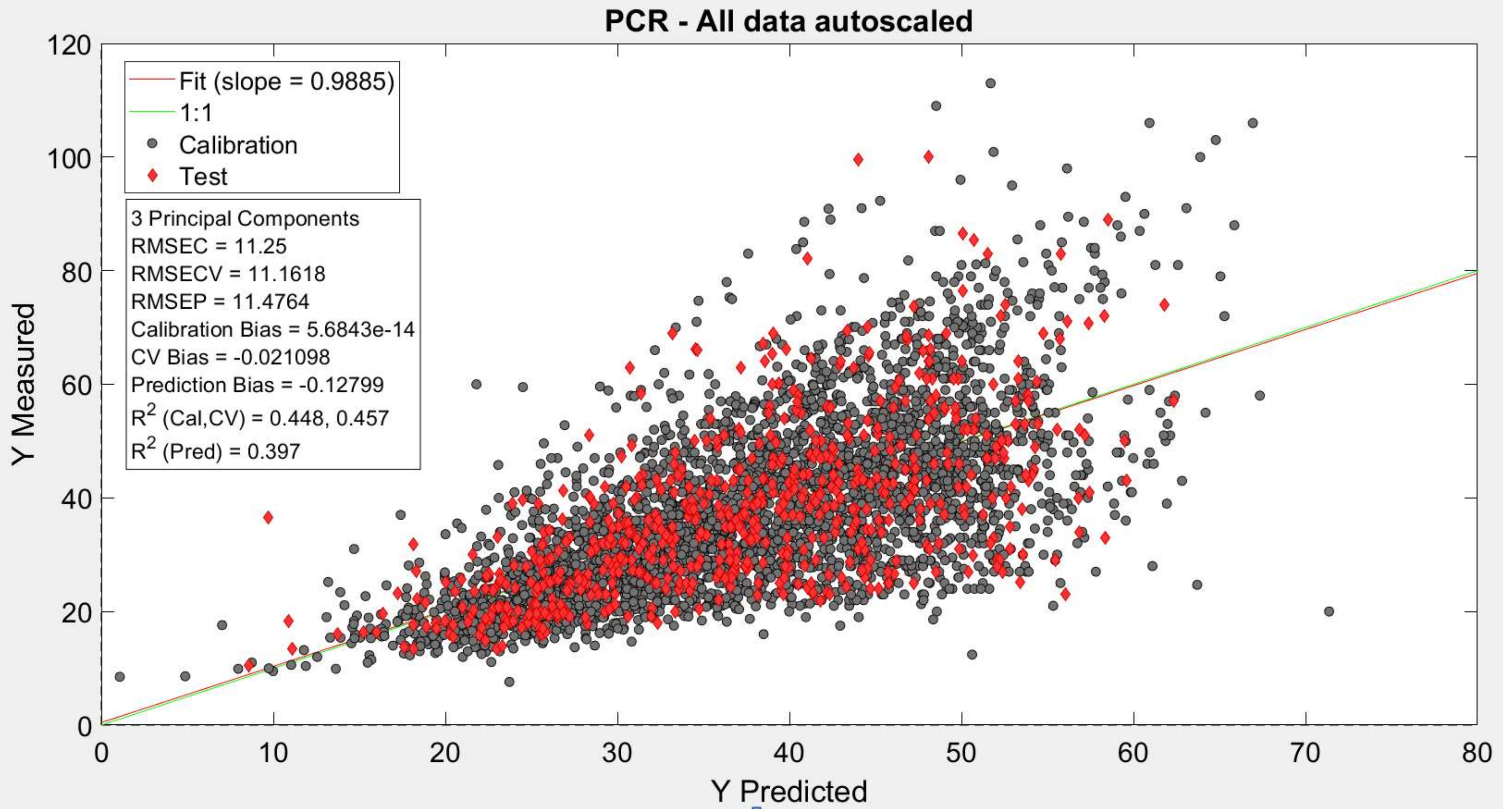


Unlocking Nature-Powered Success®

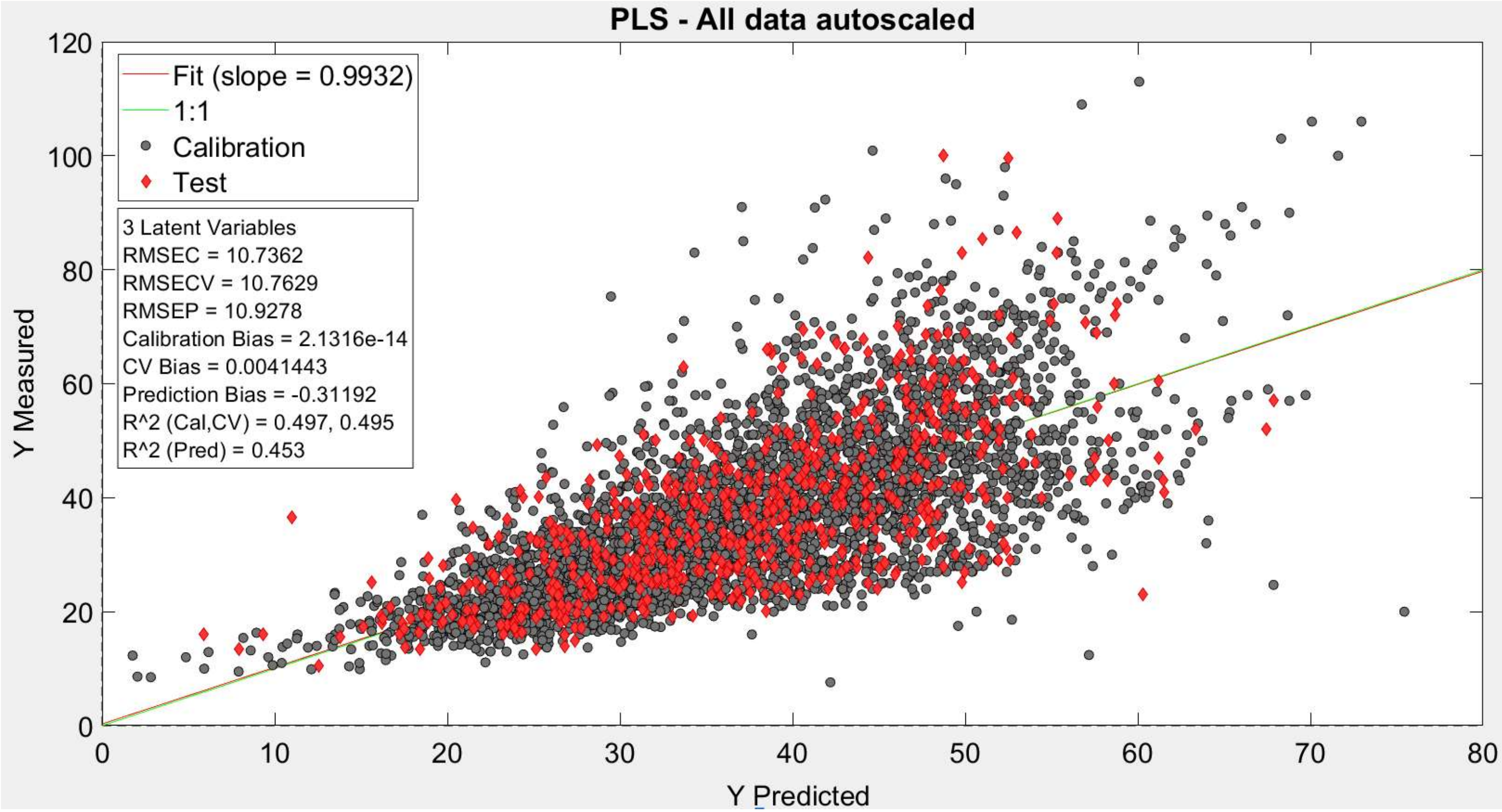
ANN – Artificial Neural Network



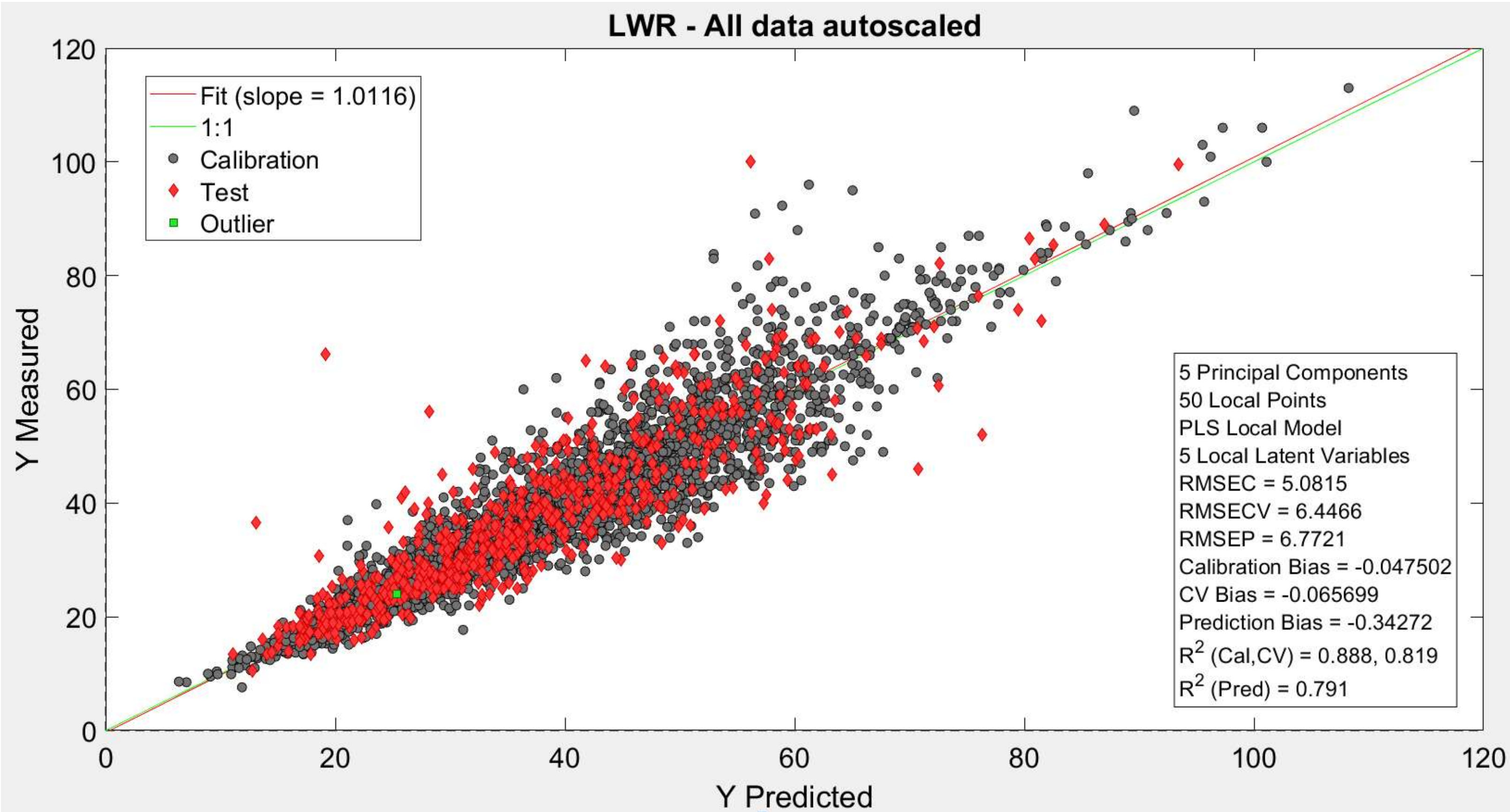
PCR – Principal Component Regression



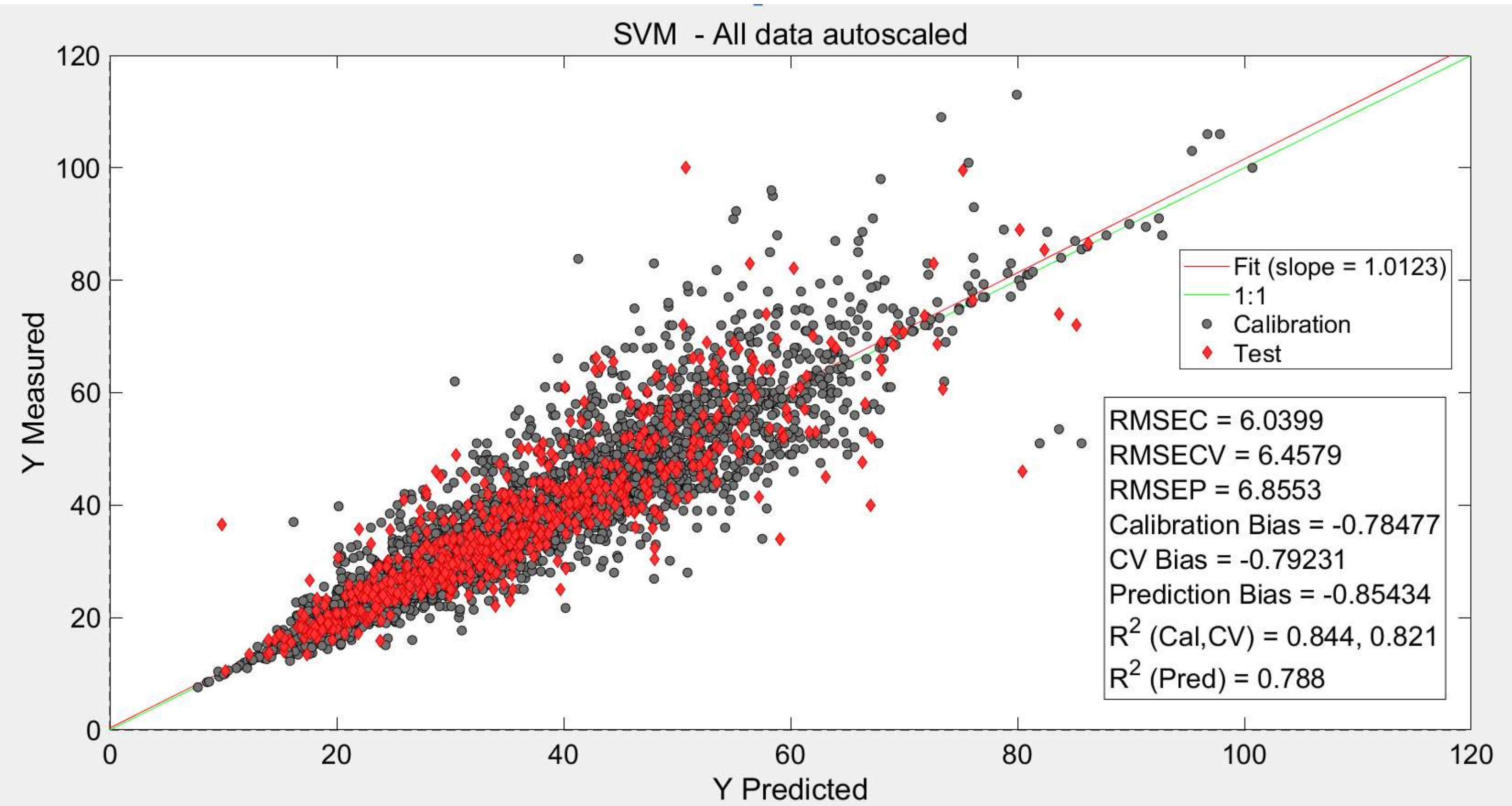
PLS – Partial Least Squares



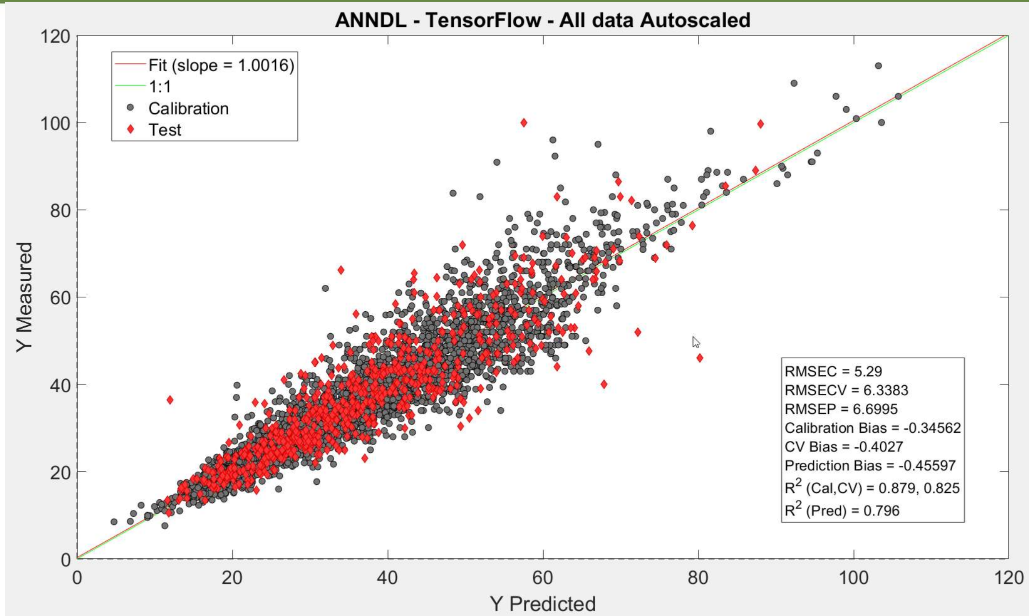
LWR – Locally Weighted Regression



SVM – Support Vector Machine



ANNDL – Artificial Neural Network Deep Learning



Compared Models

	B...	Model Na...	Sklearn Layers	Activation	RMSEC (Cal)	RMSECV (CV)	RMSE Ratio (RMSECV/RMSEC)	R2C (Cal)	R2CV (CV)	Bias	Bias (CV)
1	☑	Model 1	value = { 100 };	relu	6.441	6.699	1.04	0.8191	0.8043	-0.0461	-0.1016
2	☑	Model 2	value = { 100 50 };	relu	6.414	6.416	1	0.822	0.8205	0.5818	0.08236
3	☑	Model 3	value = { 100 50 25 };	relu	6.028	6.532	1.084	0.842	0.8143	-0.3365	-0.1483
4	☑	Model 4	value = { 100 50 25 10 };	relu	5.716	6.52	1.141	0.8578	0.8146	0.2035	0.05477
5	☑	Model 5	value = { 100 50 25 10 5 };	relu	5.806	6.418	1.105	0.8533	0.8204	-0.1505	0.001411
6	☑	Model 6	value = { 100 50 25 10 5 };	relu	5.276	6.504	1.233	0.8793	0.8155	0.3978	-0.04347
7	☑	Model 7	value = { 100 50 25 10 5 };	identity	10.74	10.8	1.006	0.4999	0.4927	-0.7275	0.0521
8	☑	Model 8	value = { 100 50 25 10 5 };	tanh	5.946	6.57	1.105	0.8464	0.8118	-0.3129	0.07948
9	☑	Model 9	value = { 100 50 25 10 5 };	logistic	7.402	7.494	1.012	0.7621	0.7558	0.4931	-0.1123

Model Na...	Tensorflow Layers	Activation	Optimizer	RMSEC (C...	RMSECV (...	RMSE Rat...	R2C (Cal)	R2CV (CV
Model 3	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];value{3}.type = 'Dense';value{3}.units = 10;value{3}.size = [ ];	relu	adam	5.29	6.338	1.198	0.8785	0.8255
Model 2	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];	relu	adam	5.533	6.336	1.145	0.8674	0.825
Model 5	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];value{3}.type = 'Dense';value{3}.units = 10;value{3}.size = [ ];	relu	adamax	5.763	6.469	1.122	0.8563	0.8177
Model 1	value{1}.type = 'Dense';value{1}.units = 100;	relu	adam	6.138	6.547	1.067	0.8364	0.8131
Model 7	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];value{3}.type = 'Dense';value{3}.units = 10;value{3}.size = [ ];	relu	adam	6.325	6.982	1.104	0.8278	0.7888
Model 4	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];value{3}.type = 'Dense';value{3}.units = 10;value{3}.size = [ ];	linear	adam	10.7	10.76	1.006	0.5015	0.4954
Model 6	value{1}.type = 'Dense';value{1}.units = 100;value{2}.type = 'Dense';value{2}.units = 50;value{2}.size = [ ];value{3}.type = 'Dense';value{3}.units = 10;value{3}.size = [ ];	linear	adamax	10.72	10.76	1.003	0.4984	0.4953

XGB – Extreme Gradient Boost