# Research Project

## Statistical Foundations of Machine Learning

Ward Gauderis & Fabian Denoodt

27/06/2022

Vrije Universiteit Brussel

# Summary

1. stochastic noise and weight decay regularisation
2. Support vector machine kernel comparison
3. Decision tree and k-nearest neighbours regressor forecasting

# stochastic noise and weight decay regularisation

What is the influence of stochastic noise in the dataset on the in- and out-of-sample error of a neural network and how does weight decay regularisation counter this?

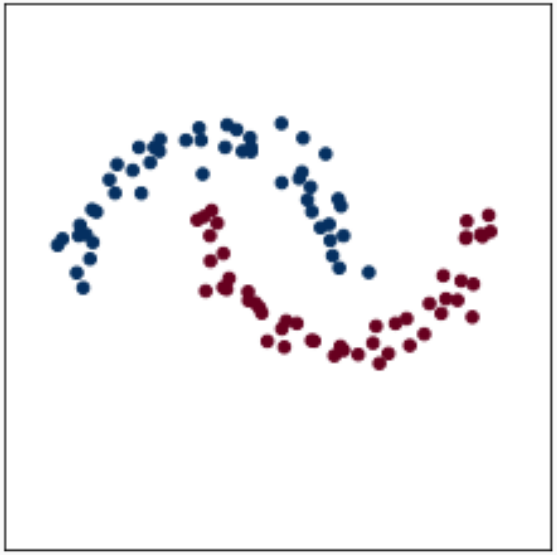
# The datasets

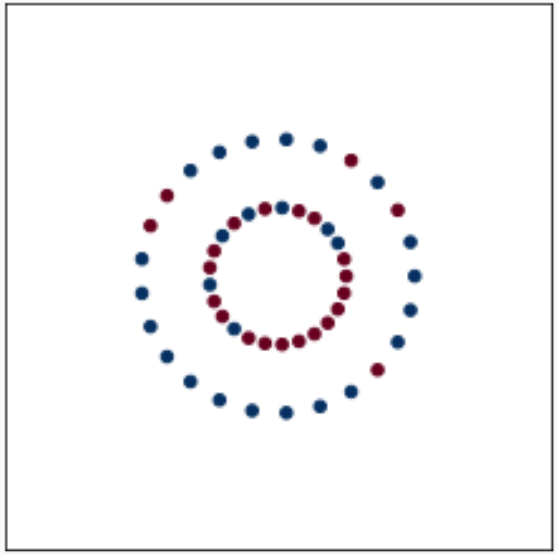
Synthetic normalised $(x, y)$ with $x \in \mathbb{R}^2, y \in \{0, 1\}$

Controllable stochastic noise:

- **Data noise**: $x' = x + \epsilon$ with $\epsilon \sim \mathbb{N}(0, \sigma^2)$
- **Label noise**: swap fraction $\alpha$ of labels $y$

No confounding deterministic noise





# The models

## Neural network

- Predict $h(x) = P(y = 1|x)$
- Linear transformations with non-linear differentiable ReLU activation functions
- Maximise likelihood by minimising the cross-entropy error
- $(20, 20)$ hidden layers & 2000 epochs

Augmented error regularisation:

$$E_{aug}(h, \lambda, \Omega) = E_{in}(h) + \frac{\lambda}{N}\Omega(h)$$

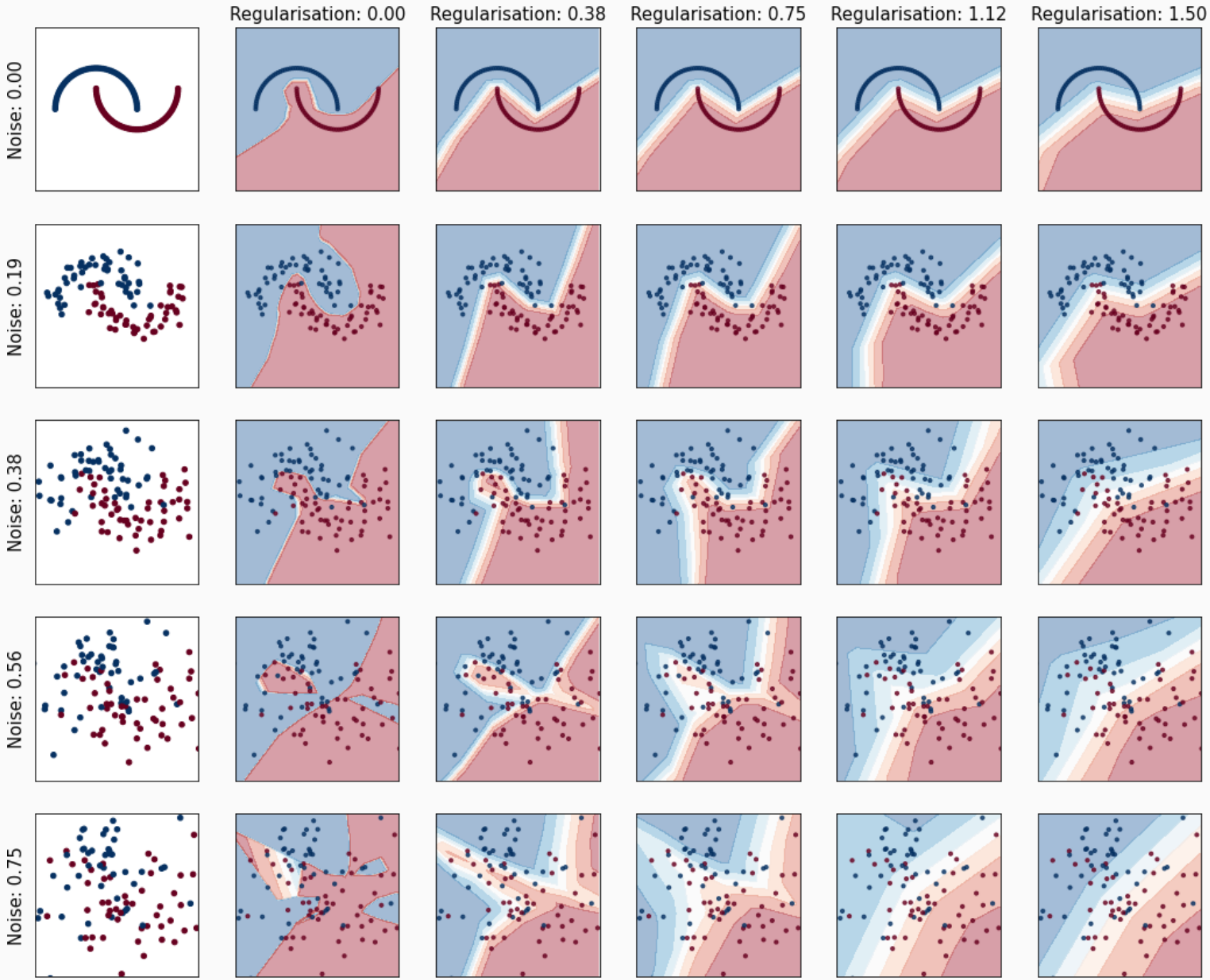
Weight-decay with $L_2$ norm:

$$E_{aug}(w) = E_{in}(w) + \frac{\lambda}{N}||w||^2$$

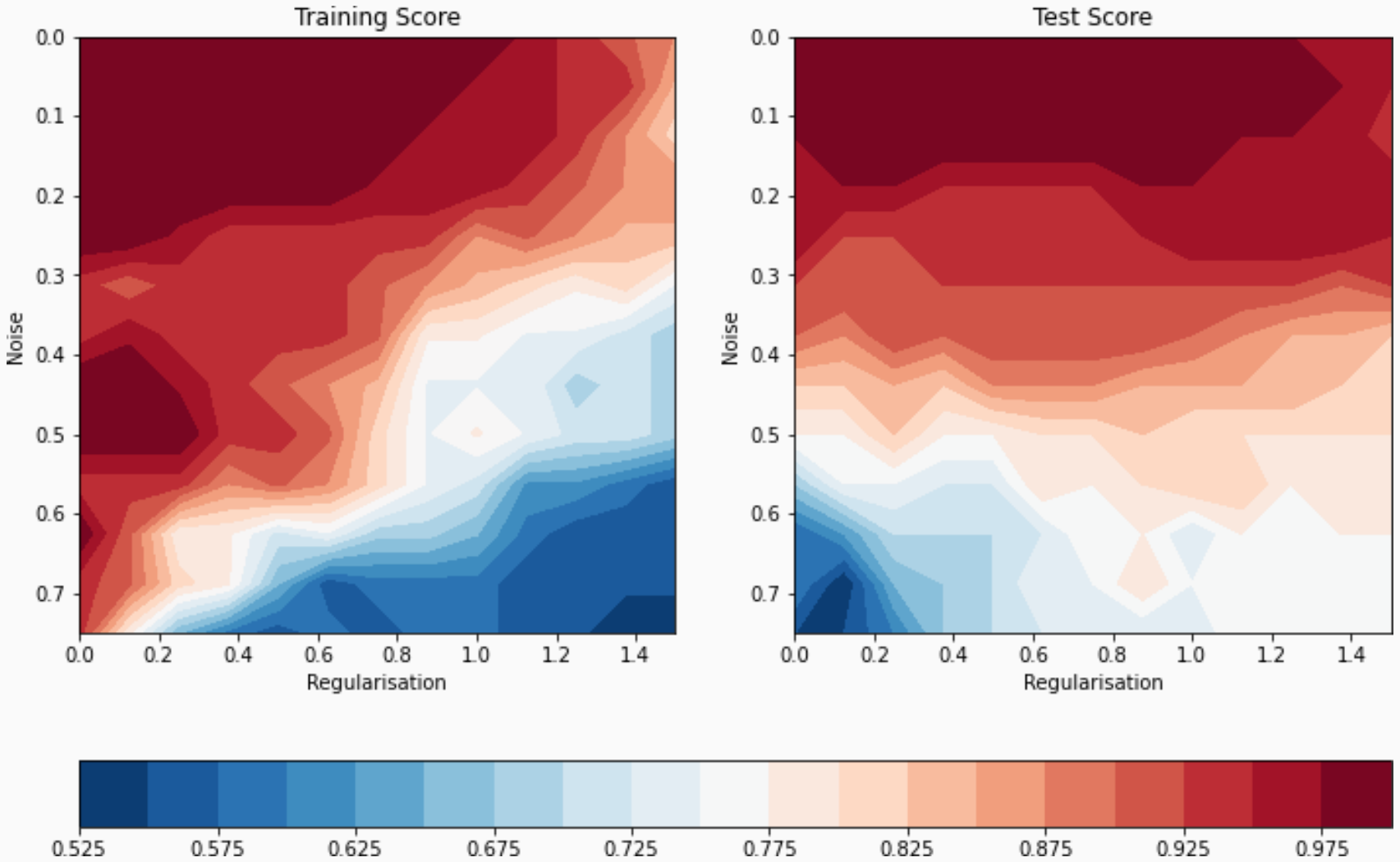
# Experimental setup

Repeat for every combination of dataset type, label and data noise:

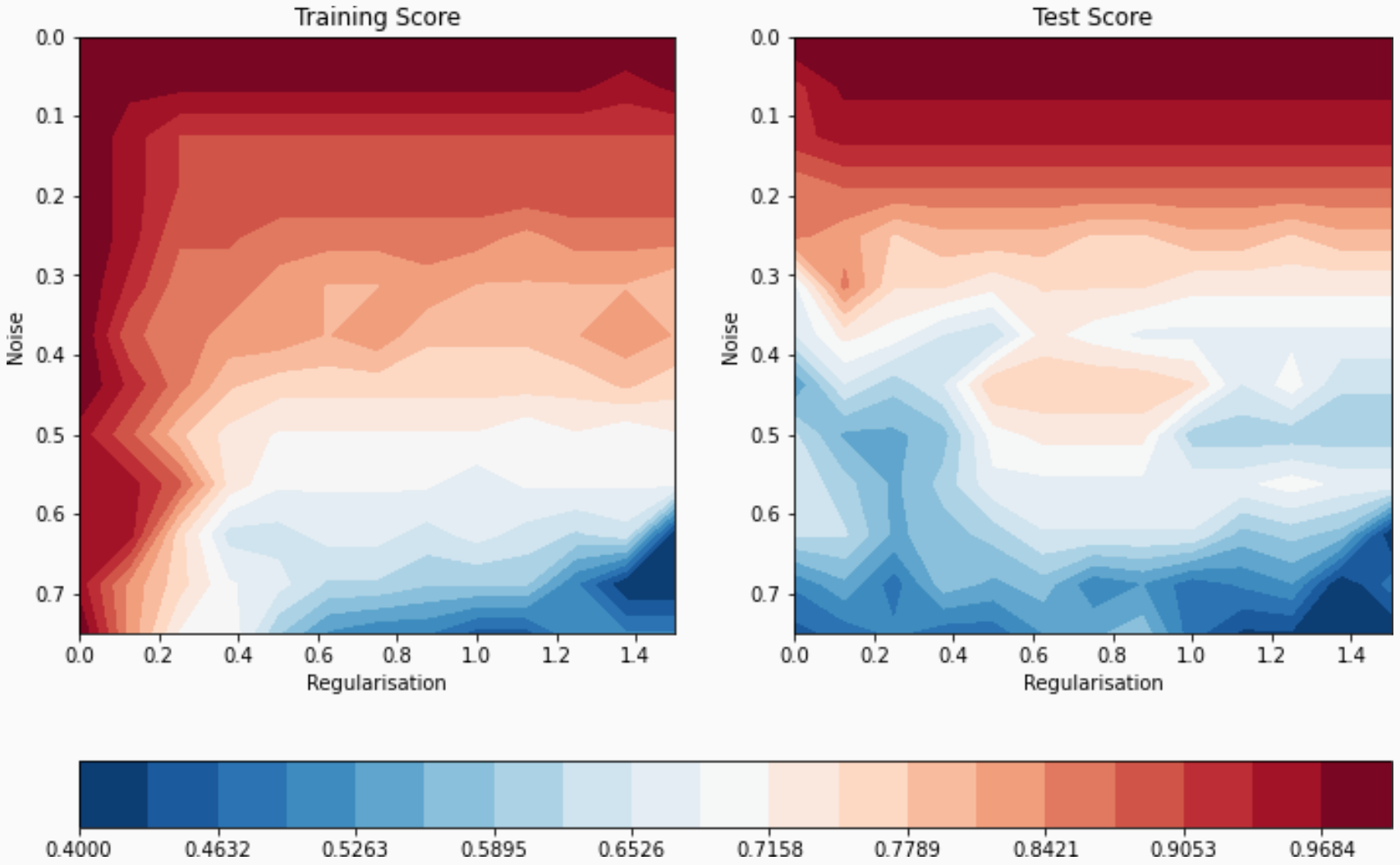
1. Generate 13 datasets of size 100 with noise $\in [0, 0.75]$
2. Create 13 models of with regularisation $\in [0, 1.5]$
3. Train every model on every dataset
4. Compare decision boundaries and training and testing accuracies



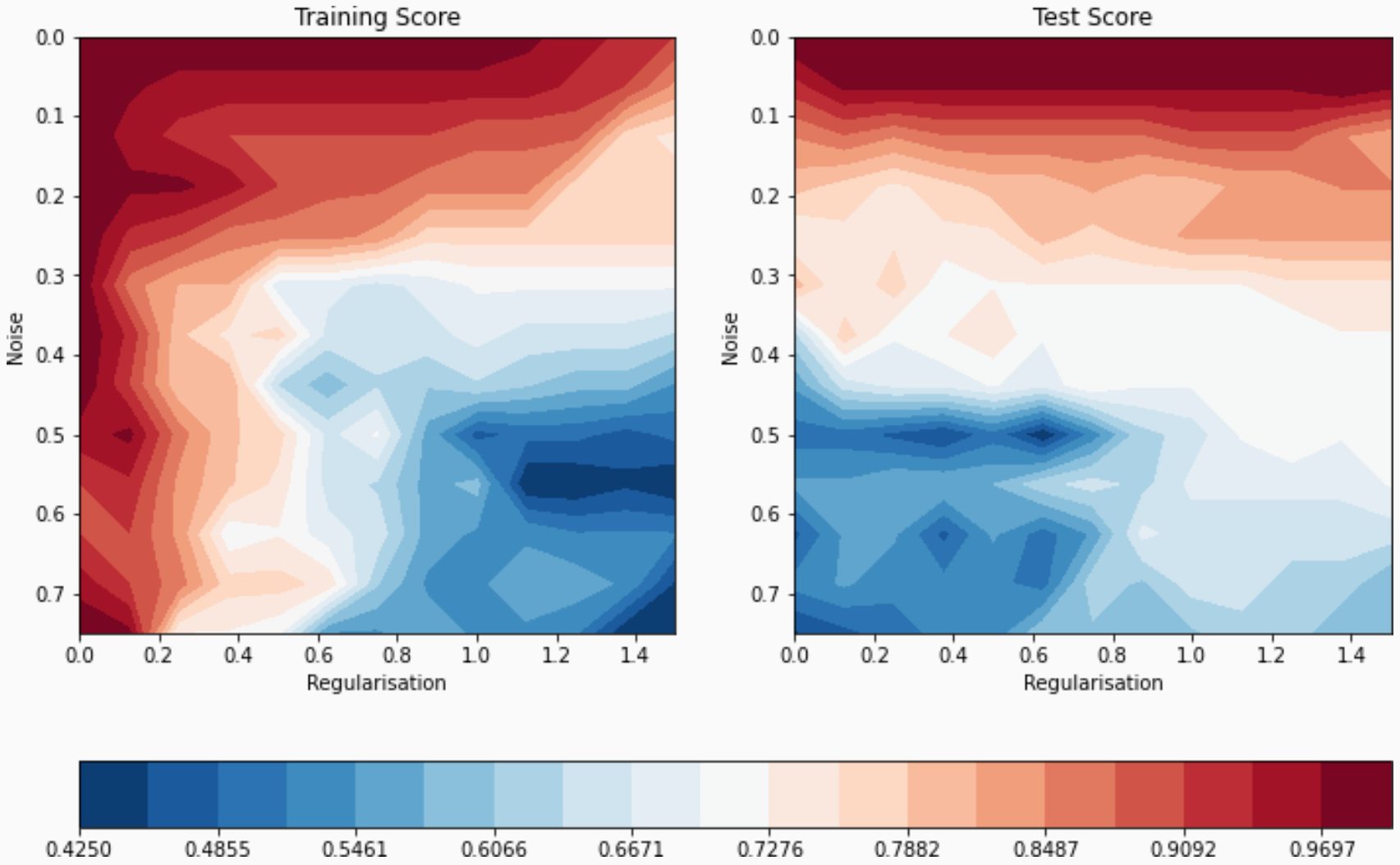

Moon dataset with data noise




Moon dataset with data noise




Circles dataset with label noise




Moon dataset with combined noise

# Conclusions

Neural networks with less effective parameters generalise better on noisier datasets

Label noise versus data noise

- more detrimental to generalisability
- can be combated with less regularisation

Underfitting is less punishing than overfitting

$E_{in}$ becomes less informative about $E_{out}$ with increasing noise and regularisation

Optimal $\lambda$ is hard to know up front without data snooping and should be chosen through model selection

# Support vector machine kernel comparison

How do the linear kernel, polynomial kernel and radial basis function compare to each other, when applied to a synthetic two-dimensional dataset?

# Decision tree and k-nearest neighbours regressor forecasting

How does the decision tree regressor model compare to the k-nearest neighbour regressor model in terms of in- and out-of-sample error for time series forecasting?

# Experimental setup

Root-mean-square error:

$$E = \sqrt{\frac{\sum_{n=0}^{N}(y_n - h(x_n))^2}{N}}$$

Regression score:

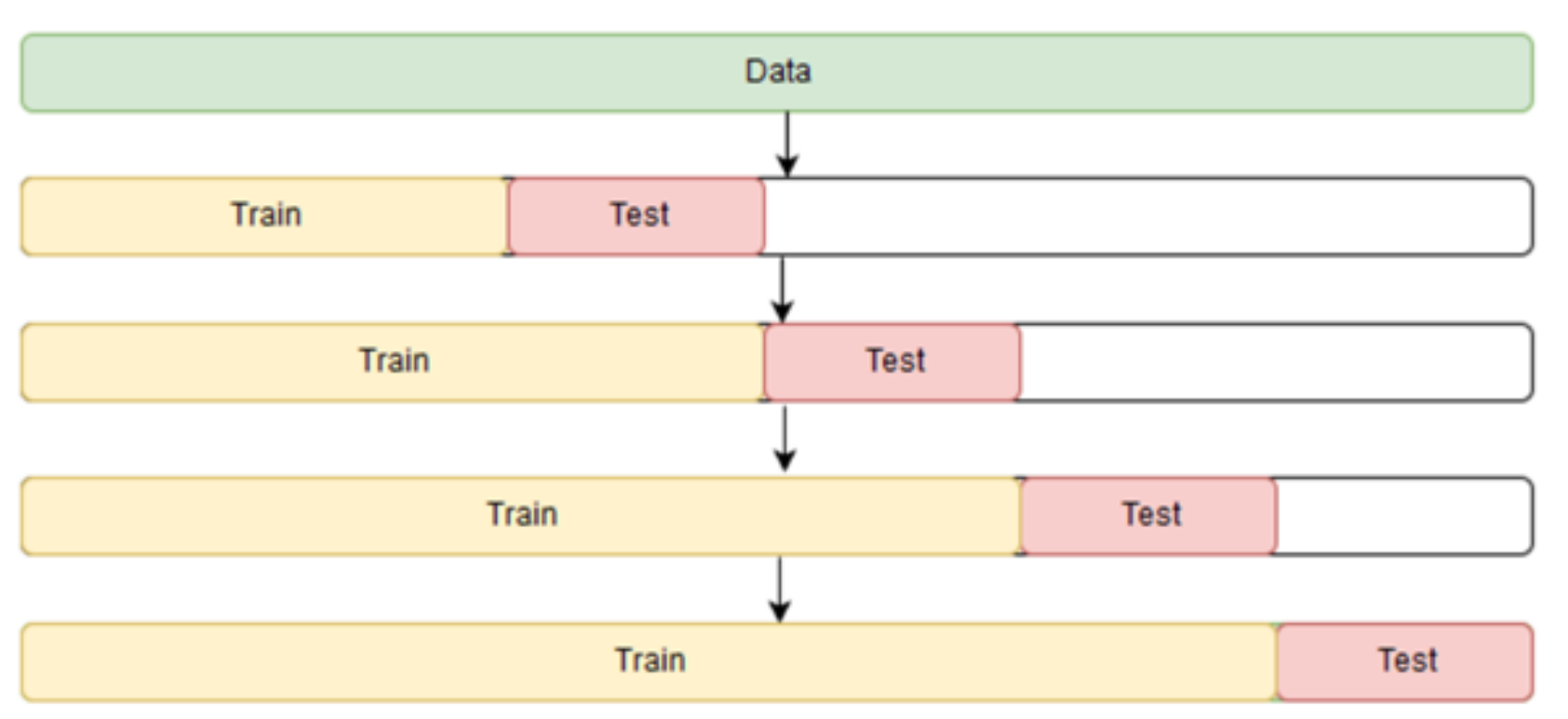
$$1 - \frac{\sum_{n=0}^{N}(y_n - h(x_n))^2}{\sum_{n=0}^{N}(y_n - \bar{y})^2}$$

Data:

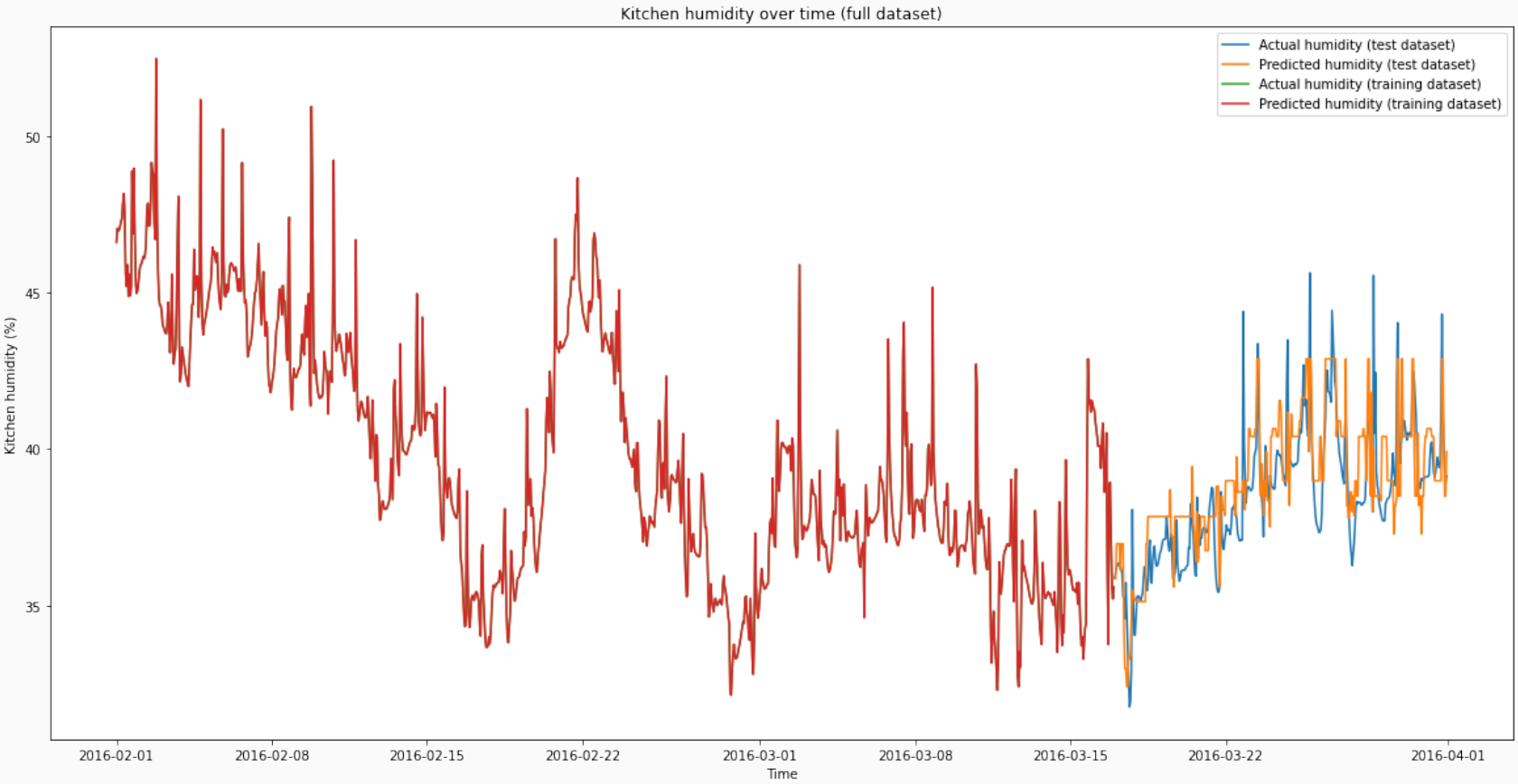
1. Split into subsequent training and testing set
2. Process normalised data

Model comparison:

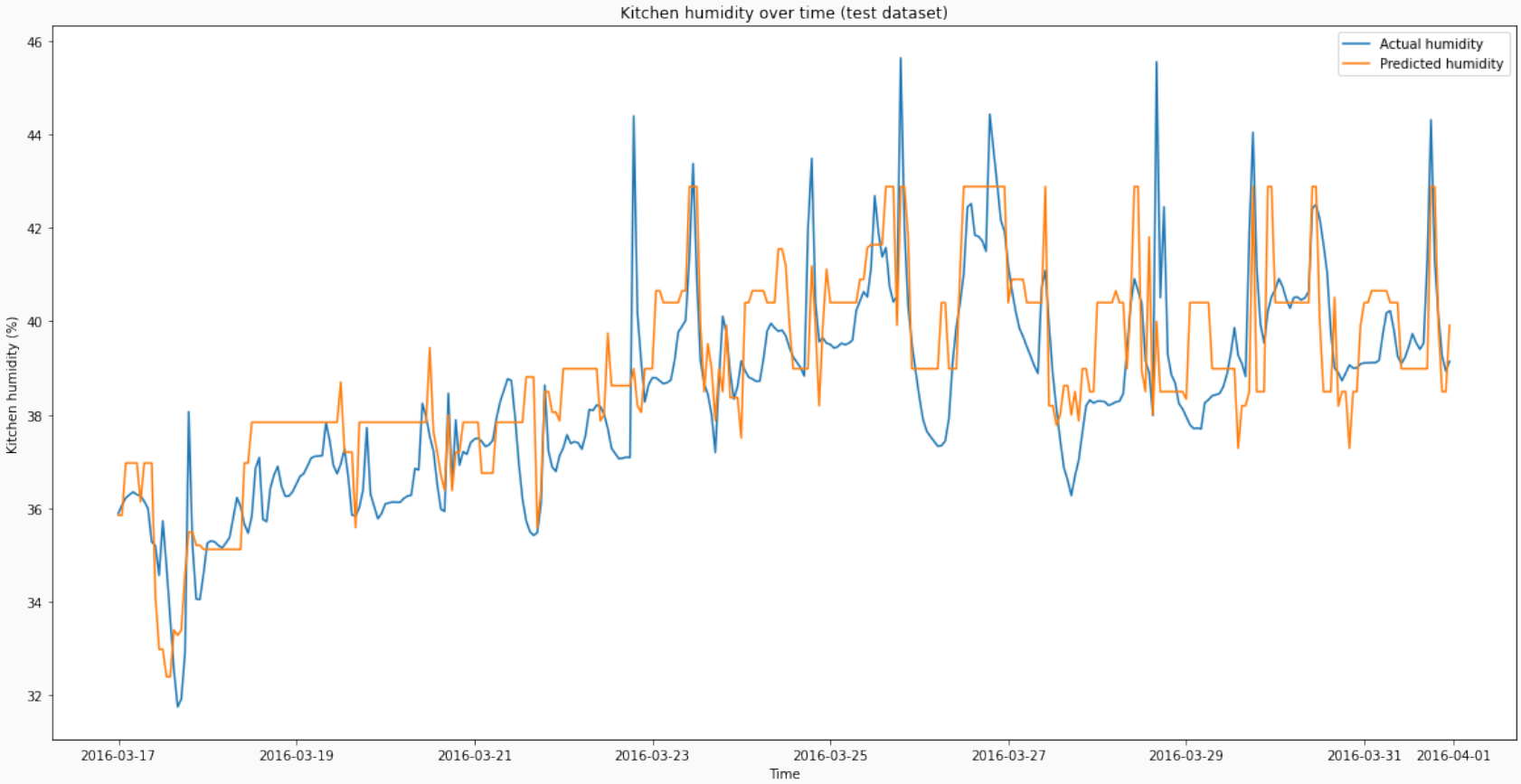
1. Baseline models
2. Equally tuned models through randomised grid search




Time series cross-validation for forecasting

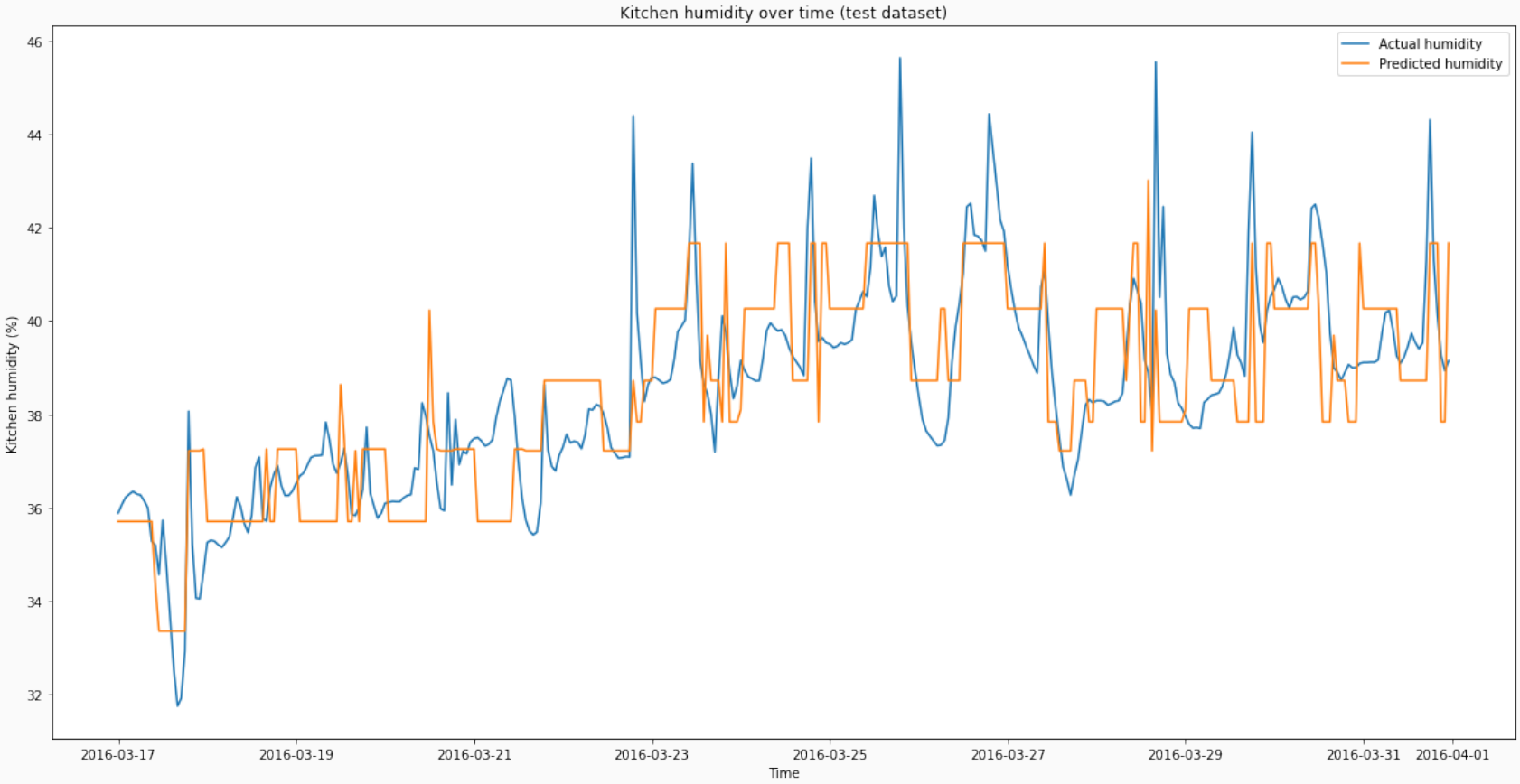



# Baseline decision tree regressor



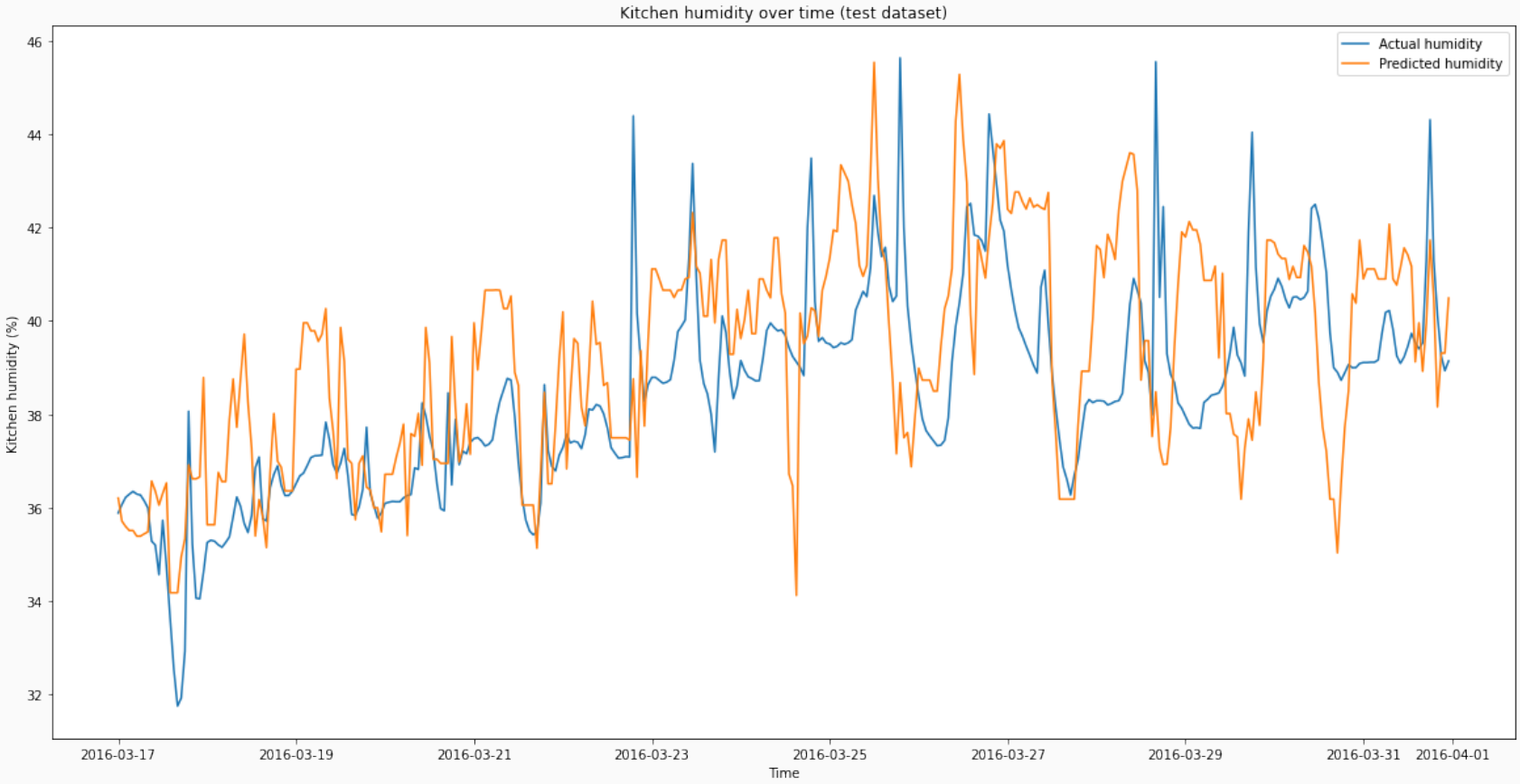

# Baseline decision tree regressor

| | Training | Testing |
|---|---|---|
| Score | 1.000 | 0.479 |
| RMSE | 0.000 | 1.551 |



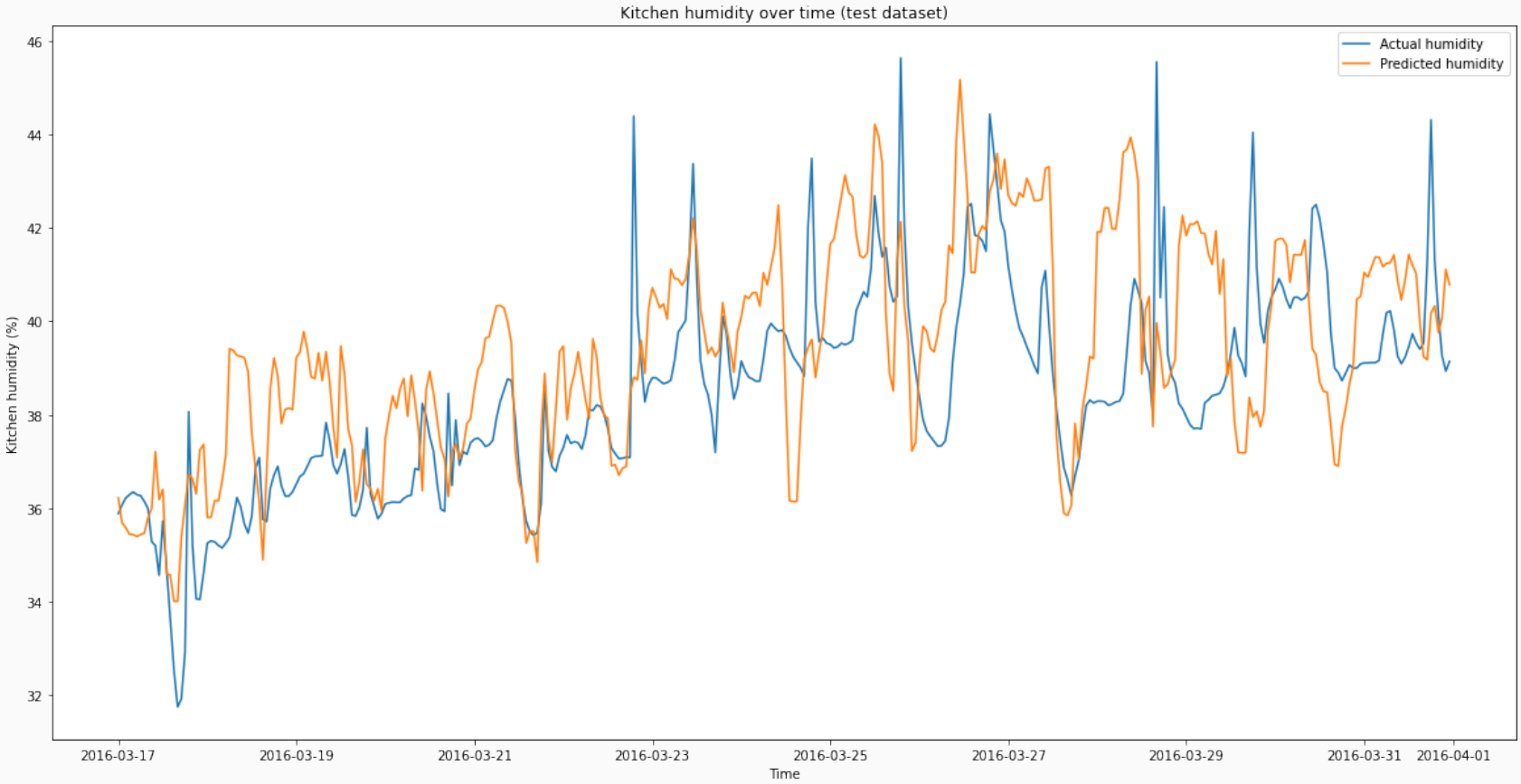

# Tuned decision tree regressor

| | Training | Testing |
|---|---|---|
| Score | 0.949 | 0.554 |
| RMSE | 0.871 | 1.434 |




# Baseline k-nearest neighbours regressor

| | Training | Testing |
|---|---|---|
| Score | 0.945 | 0.040 |
| RMSE | 0.908 | 2.105 |




# Tuned k-nearest neighbours regressor

| | Training | Testing |
|---|---|---|
| Score | 1.000 | 0.089 |
| RMSE | 0.000 | 2.050 |

# Conclusions

Tuned decision tree regressor is most likely to generalise best

Model nature is visible in predictions:

- Decision tree predicts conservative smooth surfaces
- K-nearest neighbours predicts noisy erratic changes resembling the training data

Time-series forecasting is extrapolation

Model selection can improve $E_{out}$ by reducing or increasing model complexity

Data preprocessing and feature selection is important and can be guided also by cross-validation