Explaining Heatwaves with Machine Learning

Sebastian Buschow[1], Jan Keller[2,3] and Sabrina Wahl[1,2]

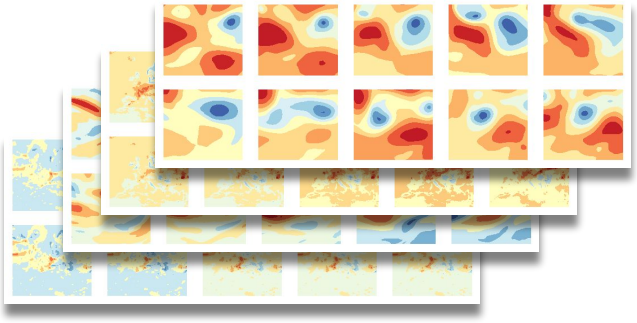
[1] University of Bonn, Institute of Geosciences, Meteorology department
[2] Hans-Ertel-Zentrum für Wetterforschung
[3] Deutscher Wetterdienst

Bundesministerium für Bildung und Forschung

FONA Research for sustainability

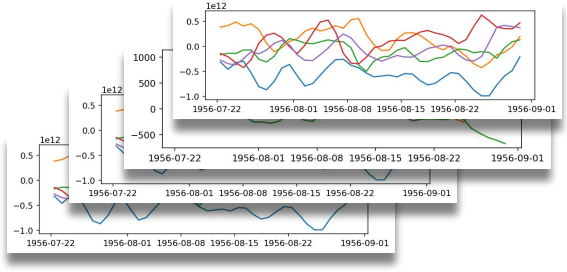
CLIMXTREME Climate Change and Extreme Events

Fields of geopotential, moisture, …



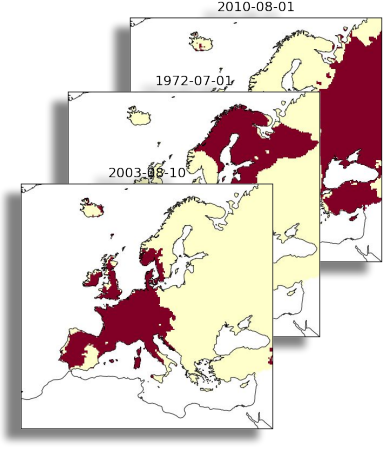
PCA

Principal components



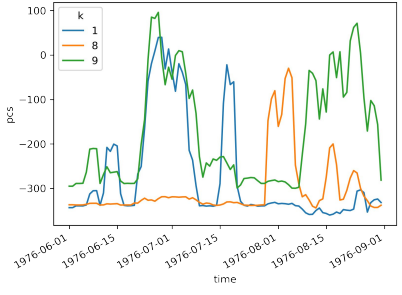
predictors

Binary heatwaves:
3 or more *hot days in a row**



Logistic PCA

Principal HW components



predictand

Statistical learner

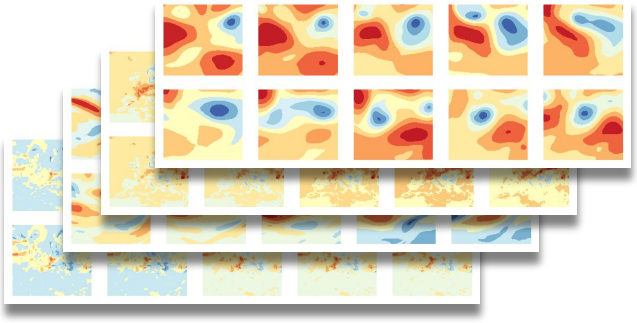
- Simple neural net
- lm for comparison

Explanations

***hot day*: daily T_{max} > 90% quantile for the current calendar day**

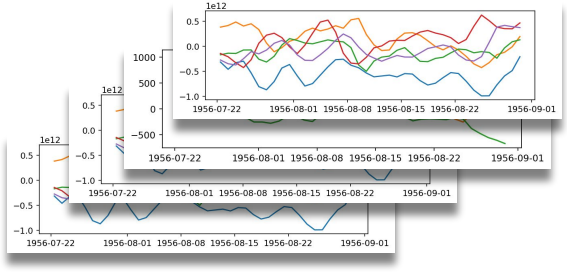
Data: ERA5 JJA, 1950-2020, EURO-CORDEX domain

Fields of geopotential, moisture, …



PCA

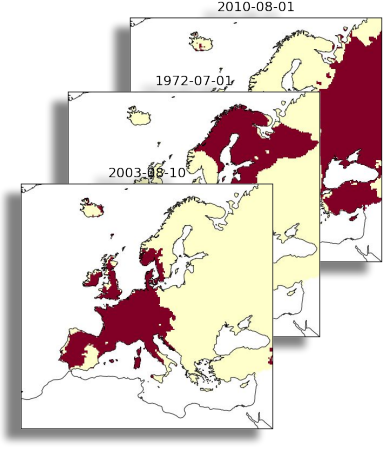
Principal components



predictors

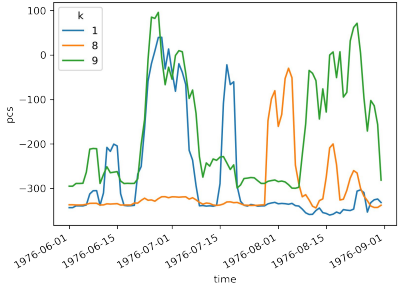
Neural networks everywhere !

Binary heatwaves:
3 or more *hot days in a row**



Logistic PCA

Principal HW components



predictand

Statistical learner

- Simple neural net
- lm for comparison

Explanations

***hot day*: daily T_{max} > 90% quantile for the current calendar day**

Data: ERA5 JJA, 1950-2020, EURO-CORDEX domain

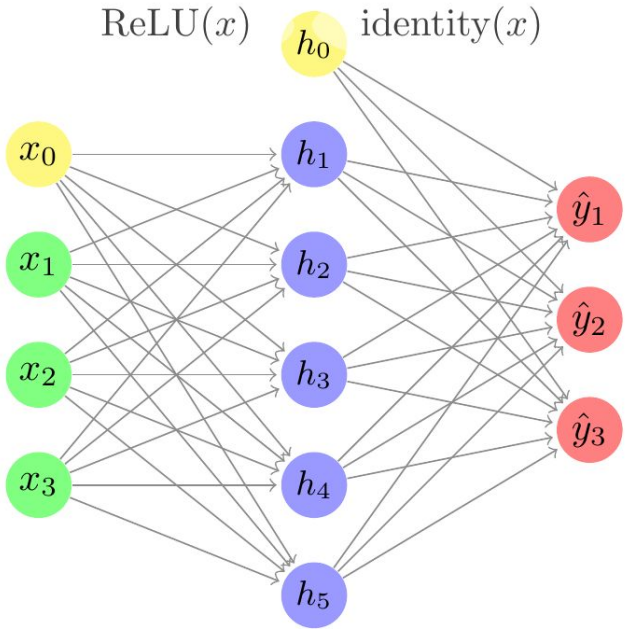
Everything is a neural net

$$\hat{y} = M_2 \cdot \mathrm{ReLU}(M_1 \cdot x + b_1) + b_2$$

A simple neural net: input x ⟶ output ŷ

$h_1 = \Phi_1(M_1 * x + b_1),$
$h_2 = \Phi_2(M_2 * h_1 + b_2),$
…
$y = \Phi_n(M_n * h_{n-1} + b_n)$

Find weights that optimize cost(M_i , b_i |x, y) !



Nonlinear regression

ReLU(x) = max(x, 0)

Everything is a neural net

A simple neural net: input x ⟶ output ŷ

$h_1 = \Phi_1(M_1 * x + b_1),$

$h_2 = \Phi_2(M_2 * h_1 + b_2),$

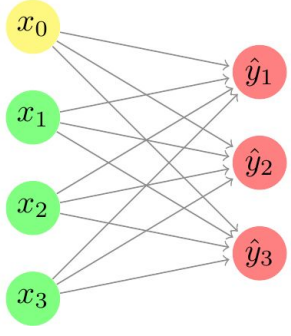
…

$y = \Phi_n(M_n * h_{n-1} + b_n)$

Find weights that optimize cost(M_i , b_i |x, y) !

$$\hat{y} = M \cdot x + b$$

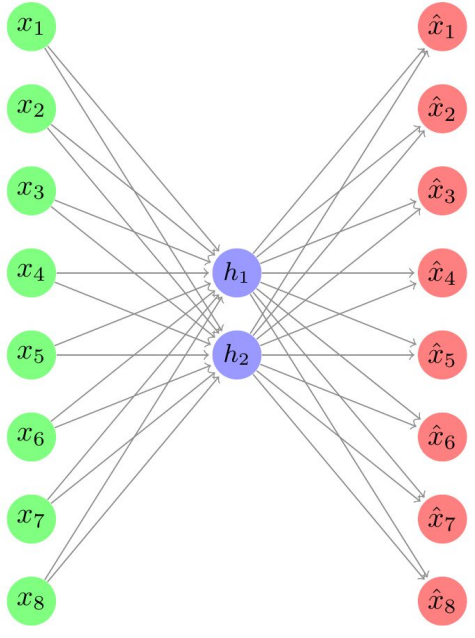
identity(x)



Linear regression
cost = MSE(y, ŷ(x))

$$\hat{x} = M_2 \cdot (M_1 \cdot x)$$

identity(x) identity(x)



Principal component analysis
cost = MSE(x, x̂(x))

Everything is a neural net

$$\hat{x} = \sigma(M_2 \cdot (M_1 \cdot x))$$

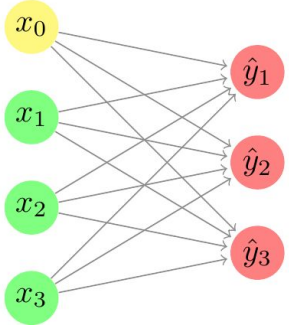
A simple neural net: input x ⟶ output ŷ

$h_1 = \Phi_1(M_1 * x + b_1),$
$h_2 = \Phi_2(M_2 * h_1 + b_2),$
…
$y = \Phi_n(M_n * h_{n-1} + b_n)$

Find weights that optimize cost(M_i , b_i |x, y) !

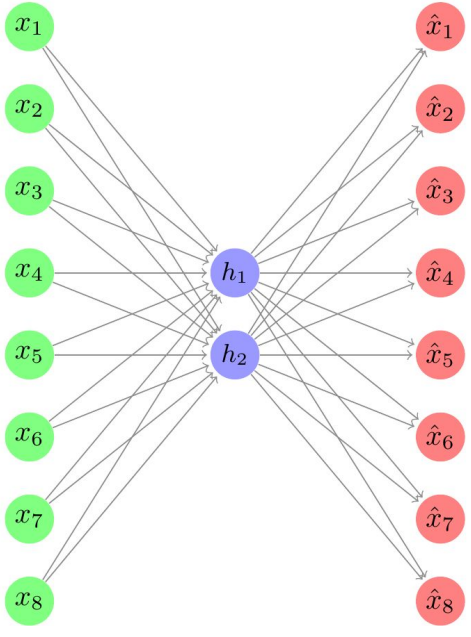
$$\hat{y} = \sigma(M \cdot x + b)$$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



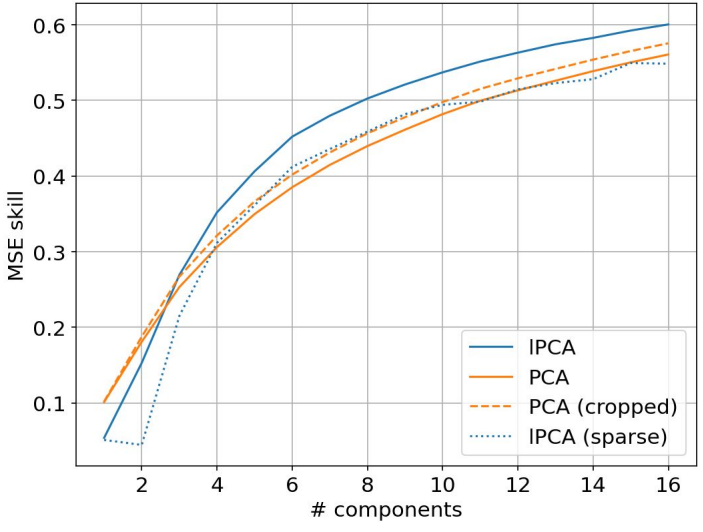
***Logistic** regression*
cost = BCE(y, ŷ(x))

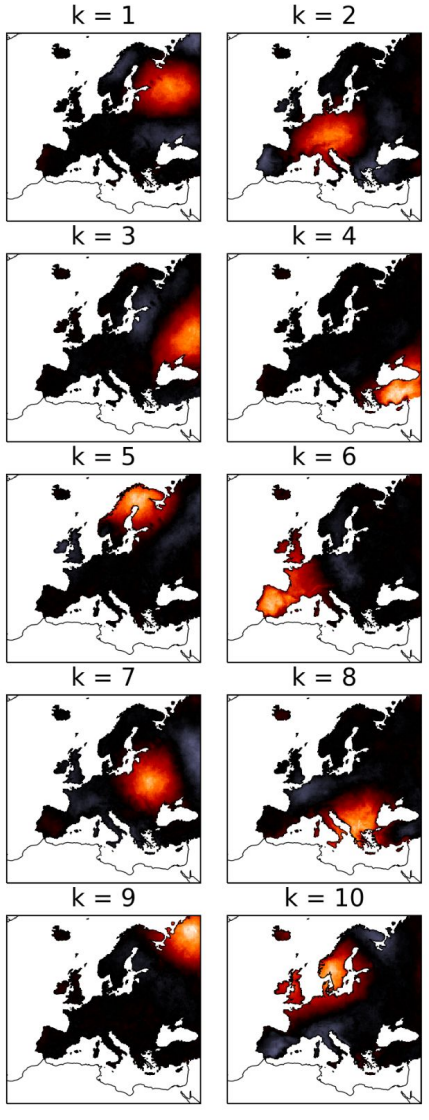
$$\text{identity}(x) \qquad \sigma(x)$$



***Logistic** PCA*
cost = BCE(x, x̂(x))

Logistic Principal Component Analysis





⟶ optimizes Bernoulli likelihood (“BCE”)
⟶ can be better than PCA also in terms of MSE (!)
⟶ sparse logistic EOFs via additional L_1 penalty

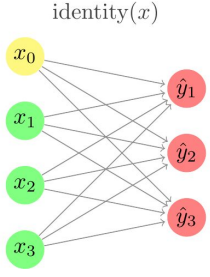
Landgraf, A. J., & Lee, Y. (2020). Dimensionality reduction for binary data through the projection of natural parameters. *Journal of Multivariate Analysis*

Modelling heatwaves in the reduced space

Predictand: 10 rotated logistic PCs of heatwaves
Predictors: 20 PCs of soil moisture (1,2) and geopotential (1000,500), u (200), precip, net solar radiation, T-advection (1000), evaporation

Multivariate linear regression

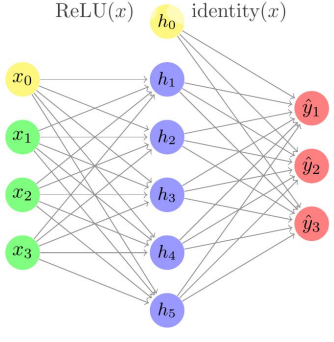
- 10 + 20 × 9 x 10 = 1810 parameters
- Least squares fit
- **R^2=0.43**

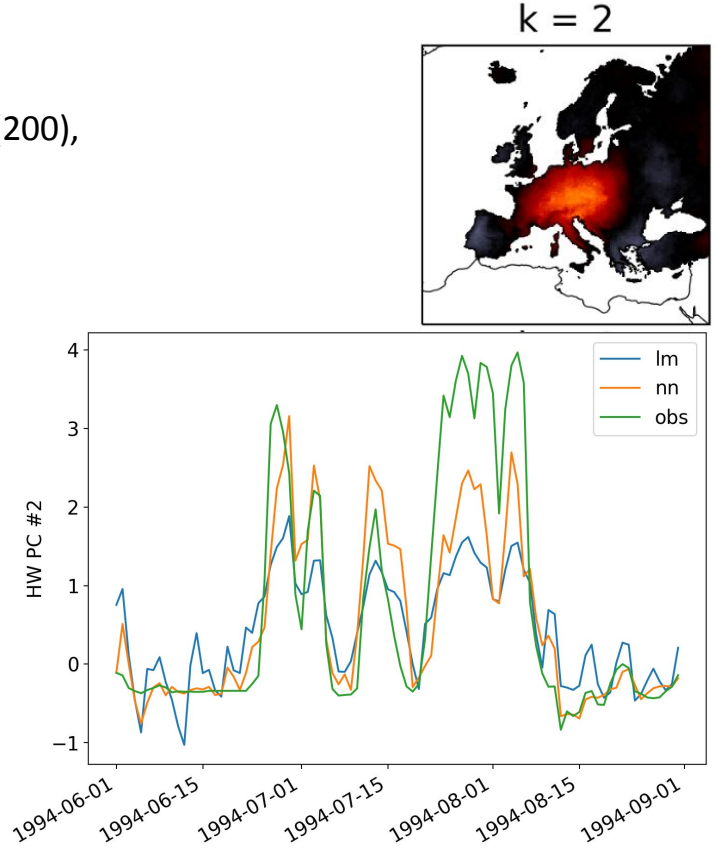


vs.

Simple feed forward neural net

- One hidden layer with 40 nodes
  ⟶ (20 x 9 + 1) × 40 + (40+1) × 10
  = 7650 parameters
- ReLu activation, 20% dropout
- Optimized with *Adam*
- **R^2=0.70**





Observed and modelled heatwave PC # 2 in summer 1994

How much R^2 is due to each predictor?

Neural network parameters not generally interpretable, predictors are correlated

→ idea: predictor importance given by change in R^2 when this variable is added !

Random decomposition of R^2

compute $\varphi_0 = R^2_0$ (no predictors)

for k=1,...,m:

randomly **select** a predictor that wasn't chosen yet

train a model with all predictors selected so far

set $\varphi_k = R^2_k - R^2_{k-1}$

Example:

$R^2(-) = 0$ → $\varphi_0 = 0$

R^2(Z1000) = 0.14 → φ(Z1000) = 0.14

R^2(Z1000, soil moisture) = 0.41 → φ(soil moisture) = 0.41 - 0.14 = 0.27

R^2(Z1000, soil moisture, Z500) = 0.71 → φ(z500) = 0.71 - 0.41 = 0.30

How much R^2 is due to each predictor?

Neural network parameters not generally interpretable, predictors are correlated

→ idea: predictor importance given by change in R^2 when this variable is added !

Random decomposition of R^2

compute $\varphi_0 = R^2_0$ (no predictors)

for k=1,...,m:

randomly **select** a predictor that wasn't chosen yet

train a model with all predictors selected so far

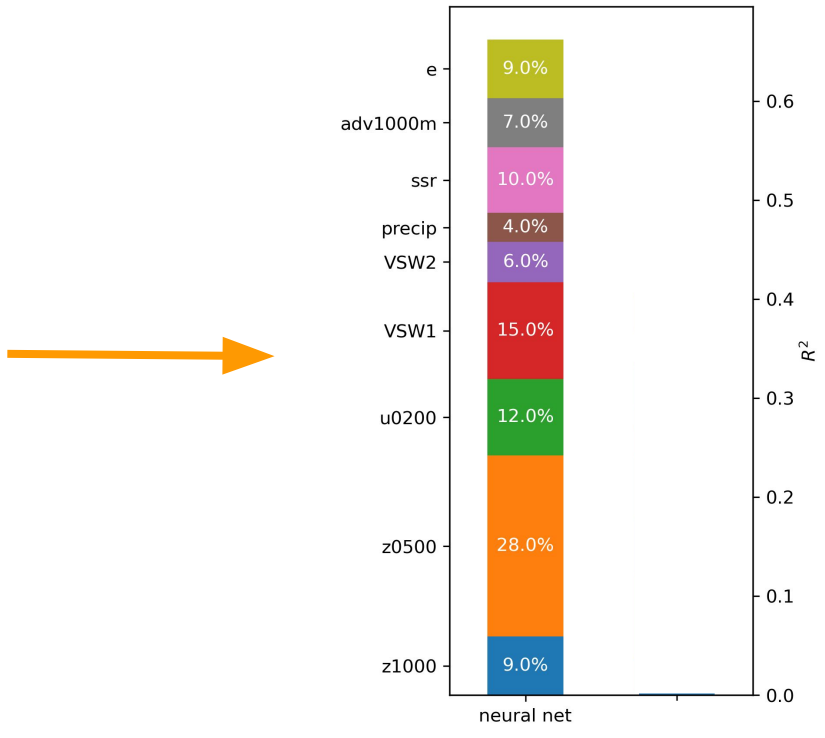
set $\varphi_k = R^2_k - R^2_{k-1}$

$R^2 = \varphi_0 + \varphi_1 + \ldots + \varphi_m$, values depend on the random order in which variables were added

use **Expectation**(φ_i) over all random orders → **Shapley values** (Shapley 1952, Nobel Prize 2012)

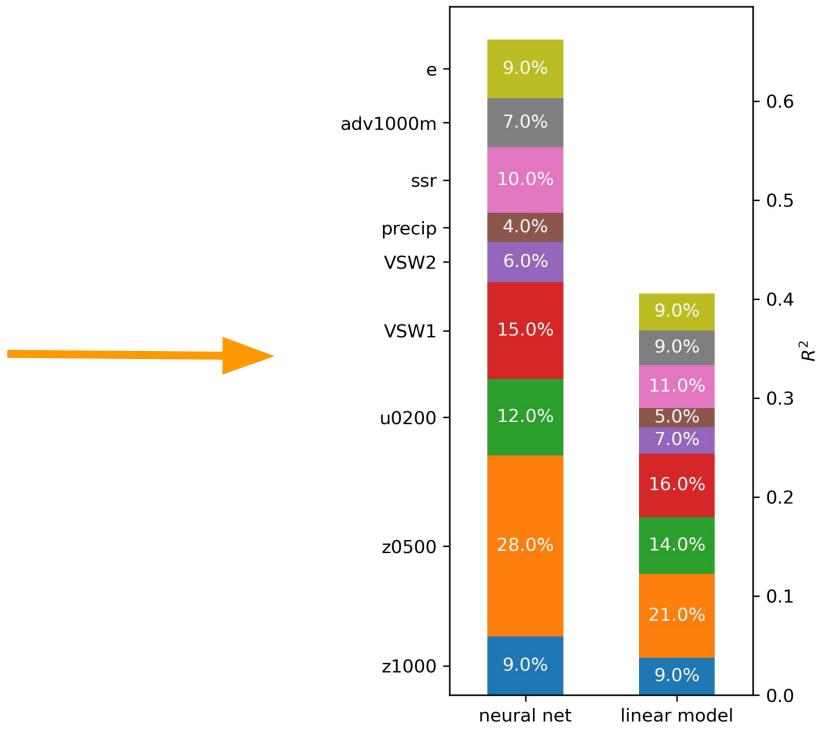
Decomposition of R^2 by Shapley values

predictors	R^2
z1000-u0200-VSW2-precip-ssr-e	57.8
z1000-z0500-u0200-VSW2-precip	67.2
z0500-ssr-e	56.4
z1000-z0500-VSW2-precip-e	68.9
z1000-precip-ssr	38.8
z0500-u0200-precip	48.6
…	…
z1000-u0200-VSW1-VSW2-adv1000m-e	61.0
VSW1-adv1000m	45.4
z1000-z0500-VSW1-VSW2-precip-ssr-adv1000m	69.8
z1000-z0500-u0200-precip-ssr-e	66.1



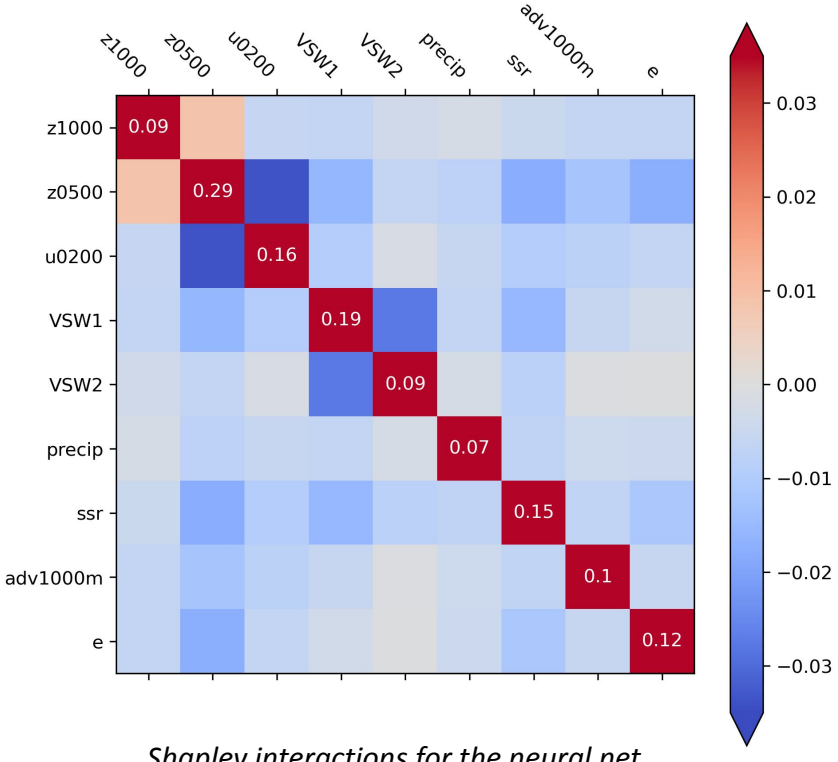
Decomposition of R^2 by Shapley values

predictors	R^2
z1000-u0200-VSW2-precip-ssr-e	57.8
z1000-z0500-u0200-VSW2-precip	67.2
z0500-ssr-e	56.4
z1000-z0500-VSW2-precip-e	68.9
z1000-precip-ssr	38.8
z0500-u0200-precip	48.6
…	…
z1000-u0200-VSW1-VSW2-adv1000m-e	61.0
VSW1-adv1000m	45.4
z1000-z0500-VSW1-VSW2-precip-ssr-adv1000m	69.8
z1000-z0500-u0200-precip-ssr-e	66.1



Shapley interactions

Hausken, K., & Mohr, M. (2001): how does the contribution of predictor A change if B is also present?
→ *Shapley interaction values*

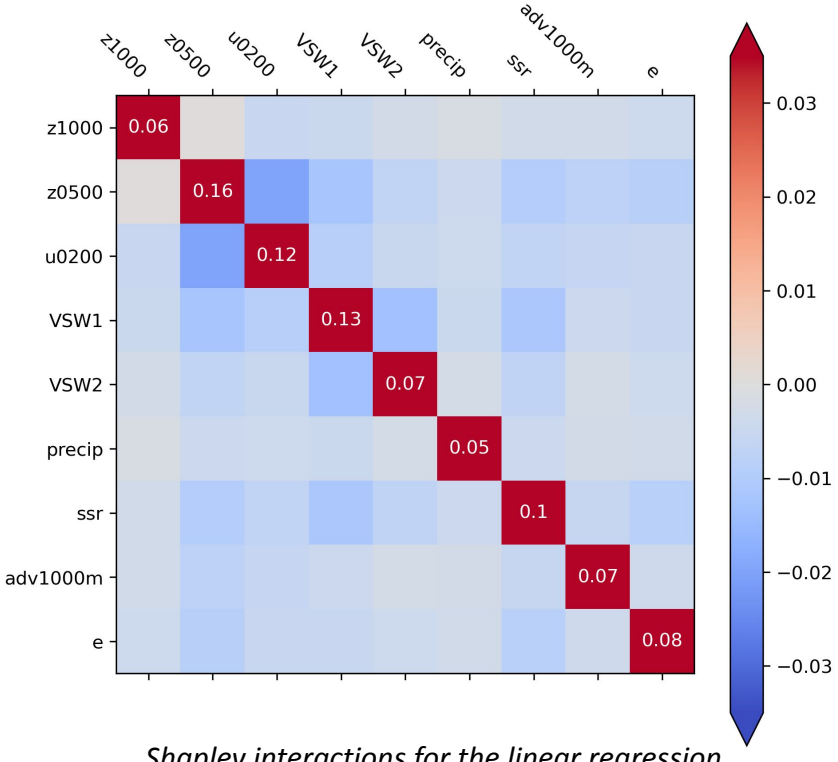


Shapley interactions for the neural net

- Rows and columns add up to Shapley values
- Negative values quantify overlap
- Positive synergy between z1000 and z0500, despite strong correlation
  → *thermal lows, hydrostatic equation*

Shapley interactions

Hausken, K., & Mohr, M. (2001): how does the contribution of predictor A change if B is also present?
→ *Shapley interaction values*



Shapley interactions for the linear regression

- Rows and columns add up to Shapley values
- Negative values quantify overlap
- Positive **synergy** between z1000 and z0500, despite strong correlation
  → *thermal lows, hydrostatic equation*

Take away points

- There is a **PCA for binary variables**, it can efficiently be implemented as a neural net
- **Shapley values** and **interactions** reveal how much *can* be learned from each predictor, and how the predictors interact
- Two predictors can be highly correlated and still **complement** each other
- A non-linear model can be superior while still learning the same connections as its linear counterpart

---

References

Hausken, K., & Mohr, M. (2001). The value of a player in n-person games. *Social Choice and Welfare*, *18*(3), 465-483

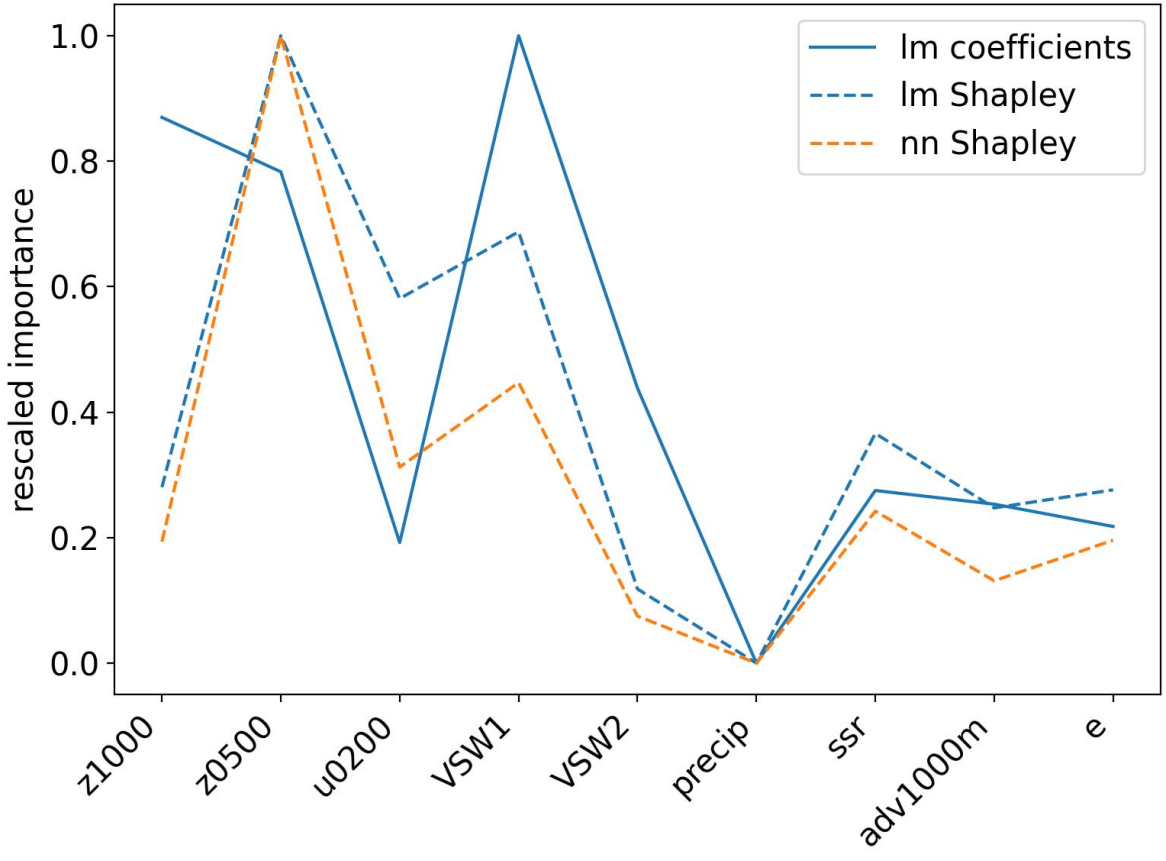
Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... & Thépaut, J. N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999-2049.

Landgraf, A. J., & Lee, Y. (2020). Dimensionality reduction for binary data through the projection of natural parameters. *Journal of Multivariate Analysis*, *180*, 104668.

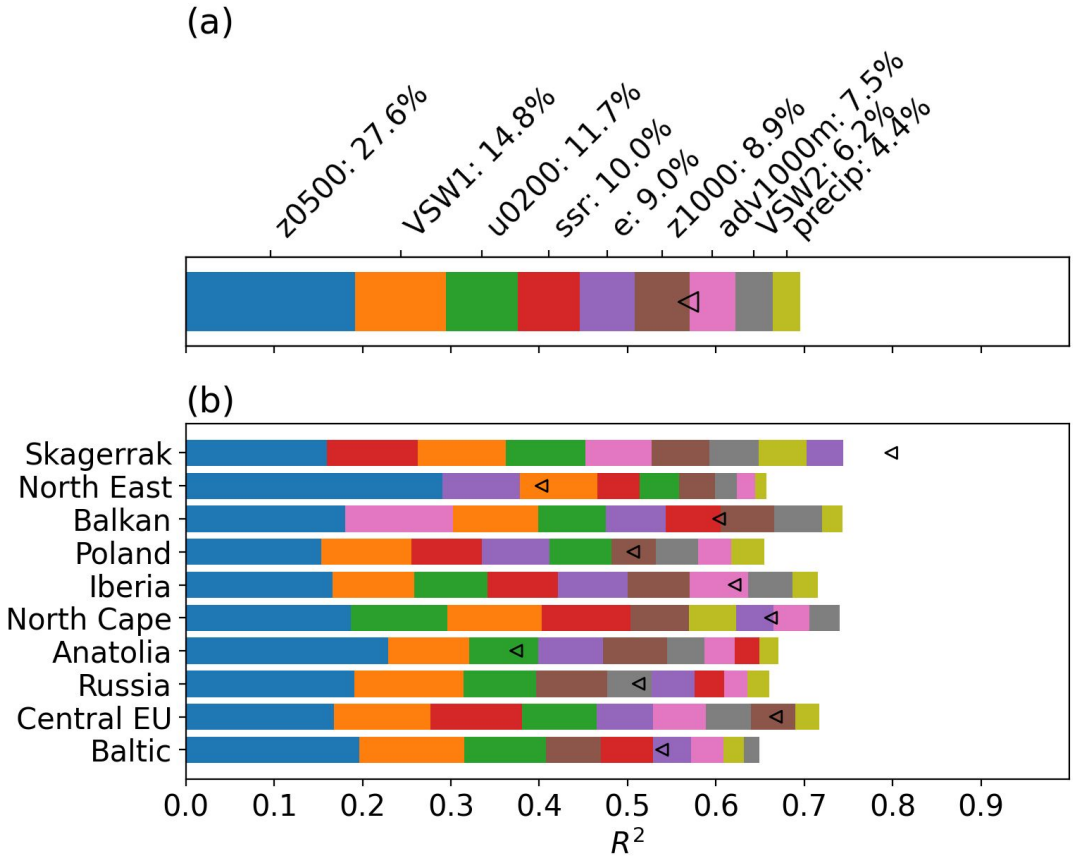
Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319-330.

Shapley, L. S. (1997). A value for n-person games. *Classics in game theory*, *69*.

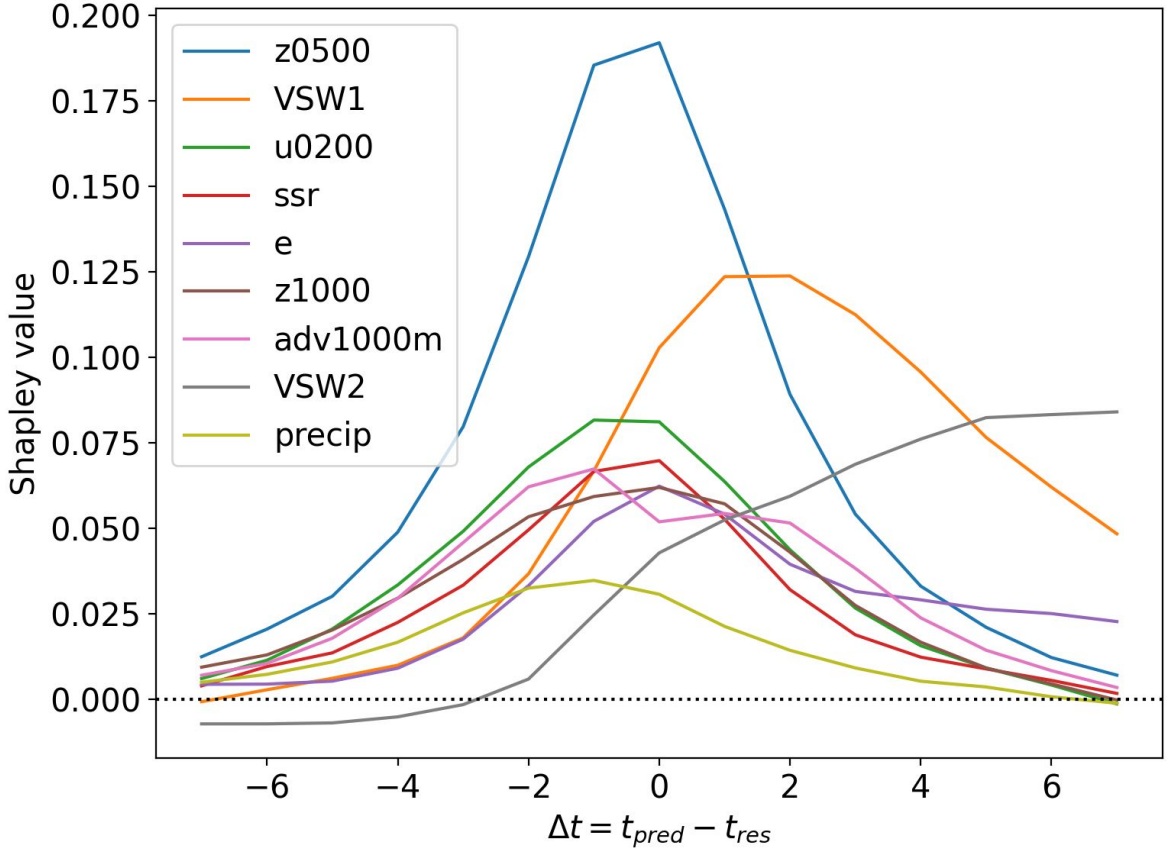
Shapley values vs. regression coefficients



Regional Shapley values



Shapley values for time shifted predictors



Consider a function f depending on its inputs $S \subseteq M = \{X_1, ..., X_m\}$. We want to decompose

$$f(M) = \varphi_0 + \sum_{i=1}^{m} \varphi_i$$

Shapley (1952):

$$Prob(S \text{ is drawn before } X_i) := \gamma(s, m) = \frac{1}{m} \binom{m-1}{s}^{-1},$$

$$\varphi_i(f, M) = \sum_{\text{all } S \subset M} \gamma(s, m) \left[f(S \cup X_i) - f(S) \right],$$

Hausken and Moor (2001):

$$\begin{aligned} \varphi_{i,j}(f, M) &:= \varphi_i(\varphi_j(f, \cdot), M) \\ &= \sum_{\text{all } S \subset M} \gamma(s, m) \left[\varphi_j(f, S \cup X_i) - \varphi_j(f, S) \right]. \end{aligned}$$

1. $f(X_1, ..., X_n) = \sum_{i=1}^{n} \varphi_i$
2. if $f(S \cup X_i) = f(S)$ for all S, then $\varphi_i = 0$
3. if $f(S \cup X_i) = f(S \cup X_j)$ for all $S \cap \{X_i, X_j\} = \{\}$, then $\varphi_i = \varphi_j$
4. if $f'(S \cup X_i) - f'(S) \geq f(S \cup X_i) - f(S)$ for all S, then $\varphi_i(f') \geq \varphi_i(f)$

Shapley (1952), Young (1985): Shapley values are **the only scheme** with these four properties!