UNIVERSITÄT
LEIPZIG

Faculty of Mathematics and Computer Science

Probabilistic Machine Learning

16-1FJXXXX_time_series_vegetation

Author Jeremias Fichtner, 3715134

SS 2025

Introduction

Brief description of the dataset and problem

The dataset is remotely sensed time series data consisting of time series for **5 environmental variables** (air temperature, evaporation, precipitation, radiation, kNDVI (Kernel Normalized Difference Vegetation Index, [1])). Data has been collected for approximately **280'000 pixels** over a time span of **21 years** in **intervalls of 8 days**, which makes for a total of **1004 timesteps**. This results in the data having the shape *[280'000, 1004, 5]* corresponding to the dimensions *[pixel, time, variable]*. The data is shuffled so that no spatial information is retained, only temporal. The goal is to make a timeseries prediction on the kNDVI variable using one or multiple models and to to assess its quality. There is also a small subset of the data where the kNDVI was not collected, and for this the missing values could be added, given a sufficient performance of the model. The dataset itself has been provided in another module without stating the source, although the recorded time period and variables point to it being Copernicus ERA5 data [2].

Motivation for the project

My background and work experience lie in remote sensing, primarily in tree-focused classification tasks. This project provides an opportunity to explore a different area: time series forecasting in an environmental monitoring context. From a broader perspective, vegetation indices and their prediction is a highly relevant topic for applications in vegetation and biodiversity monitoring, climate change, as well as agriculture. The kNDVI is a metric used to monitor vegetation using spectral remote sensing, which generalizes the traditional NDVI [3], [4] by using kernel methods. Time Series prediction is very interesting because it does not just exploit relations between different variables, but also draws information from temporal correlations in the form of seasonally reoccurring patterns. If accurate kNDVI predictions can be made from other environmental variables, this would enable reliable estimation of vegetation conditions even in cases where optical satellite observations are unavailable for example, due to persistent cloud cover.

Research question

The research question is aimed at exploring, how effectively the given environmental variables, together with historical kNDVI data, can be used to predict future kNDVI time steps. To this extent, relations between the variables and temporal/seasonal patterns will be investigated and used to configure statistical models. Also, a first probabilistic evaluation of the approach will be included.

Methods

Data Analysis

Seasonality in timeseries

Seasonality in time series refers to recurring patterns at specific intervals, such as annually. In the context of vegetation monitoring, seasonality plays a crucial role, as vegetation greenness typically follows annual cycles driven by climatic factors. For example, (k)NDVI values tend to rise during the growing season and decline in the fall and winter when vegetation loses its foliage or becomes dormant. An approximation using a smoothed median window of size four for can be seen in Figure 1. Investigating seasonal behavior can provide beneficial insights to define reasonable frameworks for prediction tasks.

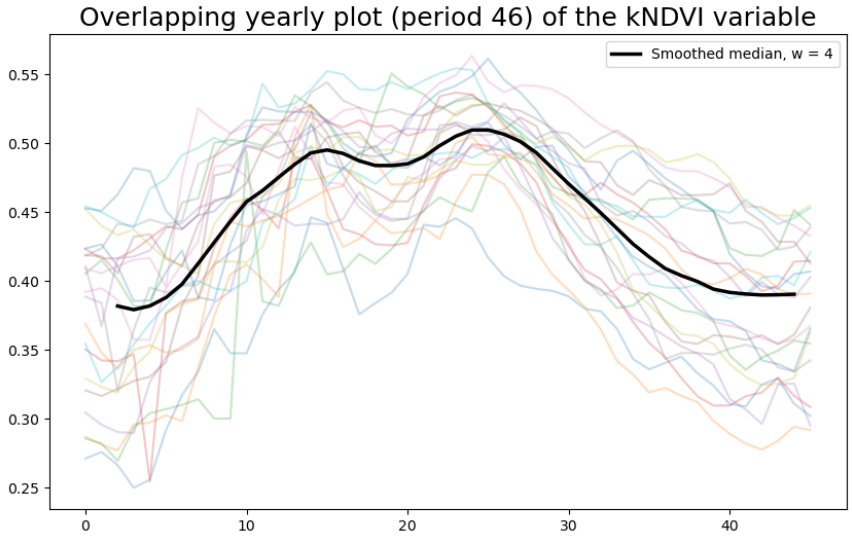

Figure 1 Approximation for yearly kNDVI cycle for a random pixel using a moving window with size four to smooth the median over multiple years.

Autocorrelation and Cross-correlation

Autocorrelation in timeseries is a notion for how much a signal is correlated to lagged versions of itself. As a function of the lag m, t is defined as

$$\Psi_{XX}[m] = \sum_{n=-\infty}^{\infty} x[n] \cdot x^*[n-m] = \frac{cov(x[n], x[n-m])}{\sigma(x[n]) \cdot \sigma(x[n-m])} \tag{1}$$

where x^* is the complex conjugation of x. Autocorrelation can also be measured in space, where it indicates, how much point data is correlated to points at a certain distance. Similar to regular correlation, autocorrelation can take on values between -1 and 1. A value larger than zero indicates a positive correlation, with high values pointing to a strong linear relationship. The same holds true for values smaller than zero, which suggest an inverse relationship between variables, which

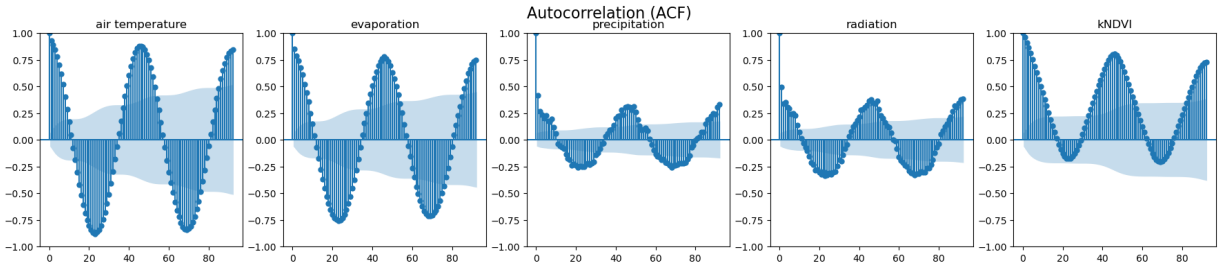

Figure 2 Output of the autocorrelation function for 92 lags for a randomly selected batch from the dataset. High values at lag 46 and 92, as well as low values at lag 23 and 69 indicate presence of a seasonal cycle with period 47.

still can be strong for values closer to -1. Values equal or close to zero can generally be an indicator for a lack of linear dependence of variables.

In regard to seasonality in timeseries, the emergence of a waveform when applying the autocorrelation function (ACF) would be an indicator for its presence, whereas the period between each maximum would be equal to the length of one season. For the evaluation the python package statsmodels.tsa [5] has been used and the results of the ACF applied on every variable of a randomly selected batch from the dataset can be seen in Figure 2.

The strong positive correlations at lag 46 and 92, as well as the strong negative values at lag 23 and 69 identify these timesteps as possibly useful inputs to predict the kNDVI at the current timestep.

Similar to the autocorrelation, which compares timeseries of one single variable, the cross correlation, given as

$$R_{xy}[m] = \sum_{n=-\infty}^{\infty} x[n] \cdot y^*[n-m] = \frac{cov(x[n], y[n-m])}{\sigma(x[n]) \cdot \sigma(y[n-m])} \tag{2}$$

is a function of the lag, that relates timeseries of different variables to each other. In Figure 3, the results of applying the cross correlation function on a random pixel for each variable and 92 lags can be seen. Looking at the kNDVI, the emerging patterns indicate the presence of time dependent relations with the other variables, mostly exhibiting a slight lag, where the kNDVI signal "follows" the others.

Discrete Fourier Transform

While the underlying signal of a timeseries might be continuous, storage in digital systems requires it to be discrete. The Discrete Fourier Transform, defined as

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi \frac{k}{N} n}, 0 < k \leq N \tag{3}$$

is a conversion from the temporal domain into the frequency domain. It takes a timeseries of length N and transforms it into up to k frequency components. The results of the transform applied using SciPy [6] on the kNDVI variable of a random pixel can be seen in Figure 4, where strong peeks at frequency $\frac{1}{46}$, $\frac{2}{46}$ and $\frac{3}{46}$ are further indicators for seasonality.

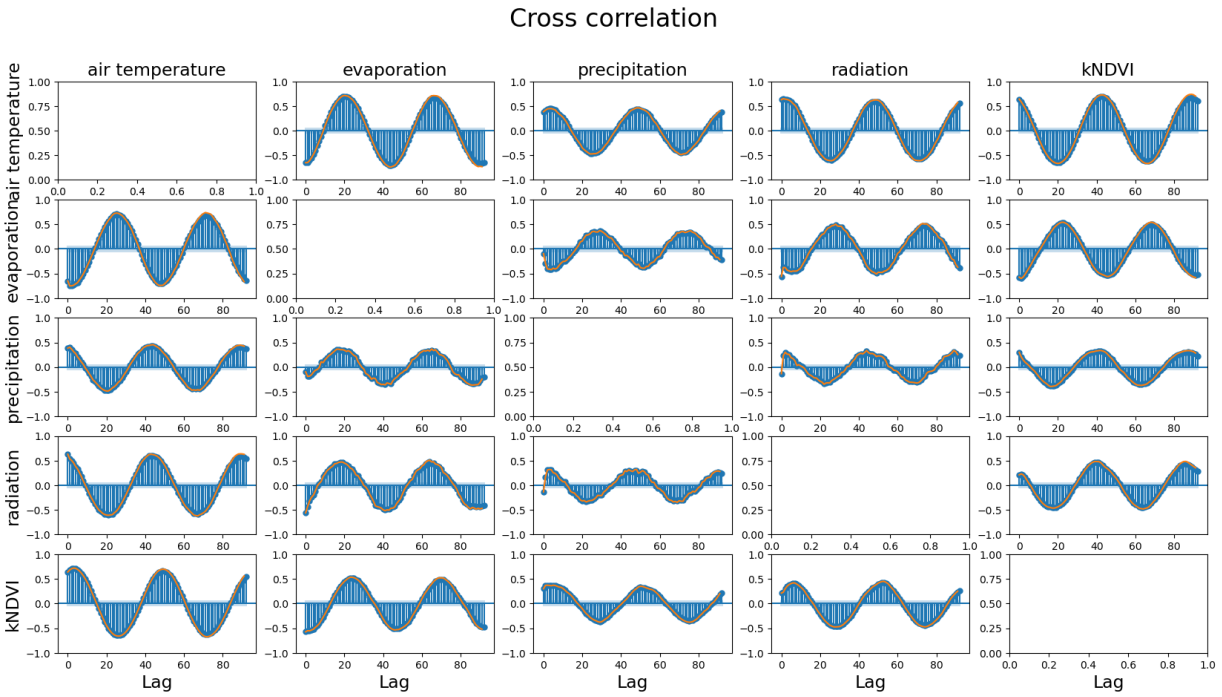

Figure 3 Output of the cross-correlation function for 92 lags for a randomly selected batch from the dataset. Sinusoidal patterns indicate the presence of seasonality and time dependent relations between the variables, where some of the functions exhibit slight shifts.

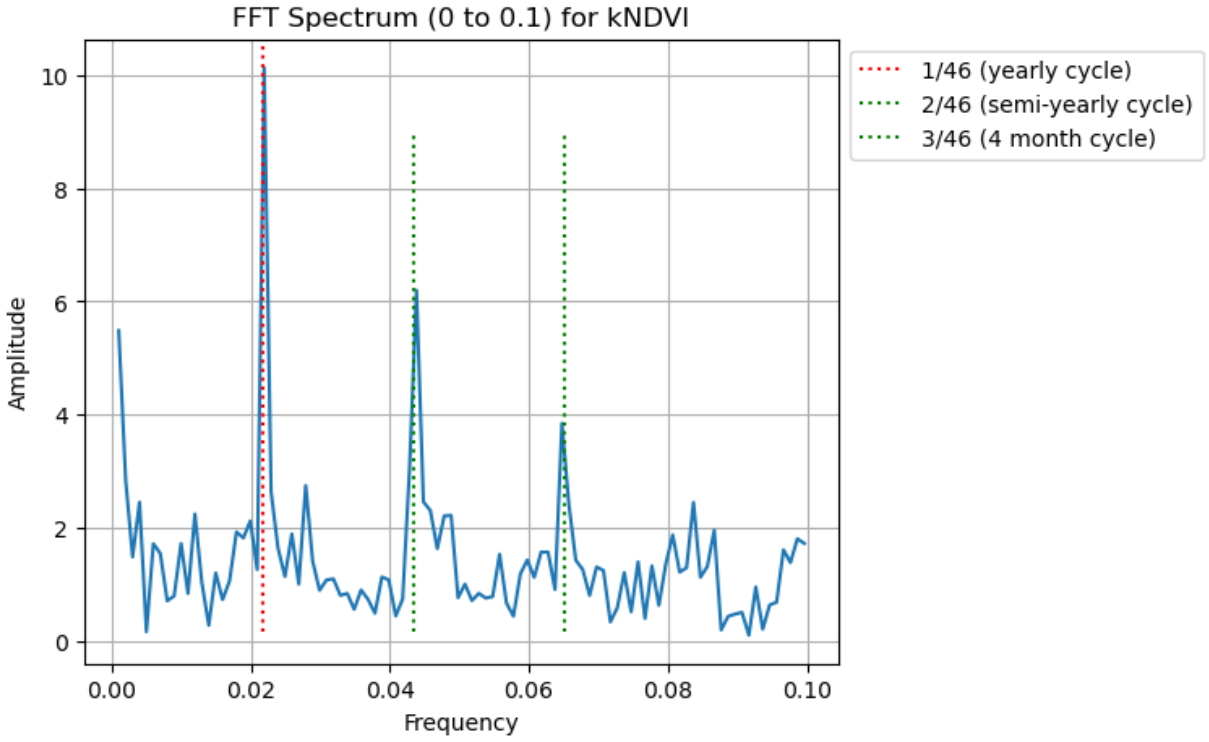

Figure 4 Discrete fourier transformation on the kNDVI variable timeseries of a random pixel, strong peaks indicate seasonality.

Seasonal Trend Decomposition using LOESS

Another widely used method for seasonality analysis is the seasonal trend decomposition using LOESS (STL) [7]. As the name suggest, the signal is decomposed into three distinct constituents: Trend, Seasonal and Residuals. LOESS or LOWESS is a non-parametric approach to fit a smooth curve to data points, meaning that it does not make any assumptions about the underlying distributions in advance. Usually, LOESS approaches are composed of a weighting function composed with a polynomial fit, leading to the estimates only beeing influenced by close points, with more weight on the closest. This could be given as

$$w(x) = (1 - |d|^3)^3 \text{ if } |d| < 1, 0 \text{ otherwise, and } \sum_i w(x_i - x_0) * [y_i - g(x_i)]^2 \tag{4}$$

where g(x) is typically a low-degree polynomial and the second expression minimized. Other approaches fore more robustness against outliers and multivariate scenarios are possible but not discussed here.

Data Transformation and Model Application

Lagged feature creation

Using the high (absolute) correlation values which were identified in the data analysis, the input features have been transformed to include those interactions. For this purpose, for each training feature, the underlying data structure has been changed to incorporate the covariates at the current lag and all variables for the 5 highest auto-correlated and the 2 highest cross-correlated lags per variable: $[1, 2, 3, 4, 28, 29, 45, 46, 74, 75]$. This also transformed the input from the three-dimensional $[batch, time, var]$ to two-dimensional, with kNDVI at lag zero as the target variable and the lag zero covariates, as well as the lagged covariates and kNDVI values as predictors. Integrating the kNDVI value from a year before can be a very good pointer to what the value will be at the current point in time, due to the seasonality. Additionally, the directly preceding values further specify the current value range and give a slight notion on whether its a downward or upward trend. The values from lag 28/29 and 74/75 serve as some sort of inverse predictors providing information on how values are roughly half a year apart. A possible assumption, which would needed to be validated further, could be that they help in the distinction between evergreen in deciduous vegetation by providing a measure for the magnitude of differences between seasons, which could be beneficial additional information for inference.

Stepwise predictions

Following the structure of the transformed inputs, one can see, that kNDVI values from timesteps with up to lag 75 are needed as input variables. Due to the nature of the research question challenge, certain kNDVI (for evaluation purposes e.g. the last 92) values are missing, which leads to the prediction needing to happen stepwise. Starting with the first missing timestep, the model can utilize the up to that point complete information to fill in the value at the first missing spot, making the input information complete up to the new timestep, thus allowing the prediction to

move one step forward and start inferring the next value. This process can be repeated, as long as complete covariate values are provided, as the model is not able to iteratively predict them and would come to a stop, as soon as they are missing from the current lag.

Probabilistic models

PyMC Bayesian Ridge Regression

The first probabilistic approach used a Bayesian linear regression model which was implemented using PyMC [8] with PyTensor [9] for Tensor operations. The zero centered normal distributed prior promotes low coefficient estimates, corresponding to an ridge regression in a frequentist framework. For performance reasons, only the features from the current lag and the most distant lag have been considered. The first assumptions are rather weak, with the coefficients being normally distributed around 0 with a standard deviation of 5:

$$\beta_{0\ldots n} \sim \mathcal{N}(0, 5). \tag{5}$$

For the likelihood a normal distribution is chosen

$$y|\beta_{0\ldots n}, \sigma \sim \mathcal{N}(\beta_0 + \beta_1 * x_1 \ldots \beta_n * x_n, \sigma) \tag{6}$$

with an exponentially distributed sigma

$$\sigma \sim \text{Exp}(1) \tag{7}$$

as a prior for the noise. Posterior inference was performed via MCMC sampling with 500 draws across four parallel cores.

PyMC Bayesian Lasso

The second probabilistic approach used a Bayesian linear regression model with Laplace priors on the coefficients, implemented using PyMC and PyTensor. For computational efficiency, only the first 1000 samples from the scaled training data were used. The Laplace prior promotes sparsity in the coefficient estimates, corresponding to an L_1 regularization in a frequentist framework. This is also why, in this case not the parameters but the samples have been reduced, to better show the possibly resulting sparsity. The scale parameter of the Laplace distribution was given a hierarchical prior via a Half-Cauchy distribution:

$$\lambda \sim \text{HalfCauchy}(1), \tag{8}$$

$$\beta_{1\ldots n} \sim \text{Laplace}(0, \frac{1}{\lambda}). \tag{9}$$

The intercept term was assigned a weakly informative prior:

$$\beta_0 \sim \mathcal{N}(0, 5). \tag{10}$$

The likelihood was modeled as a Normal distribution:

$$y \mid \beta_{0\ldots n}, \sigma \sim \mathcal{N}(\alpha + \beta_1 x_1 + \cdots + \beta_n x_n, \sigma), \tag{11}$$

with the observation noise standard deviation given an Exponential prior:

$$\sigma \sim \text{Exp}(1). \tag{12}$$

Posterior inference was performed via MCMC sampling with 500 draws across four parallel cores.

Scikit-learn Bayesian Ridge Regression

The third model uses the `BayesianRidge` implementation from scikit-learn [10]. This is similar to regular ridge regression, but it uses a Bayesian approach by placing Gaussian priors on the coefficients and the intercept, which corresponds to an L_2 regularization effect. The priors are controlled by two parameters: α for the weights and λ for the noise. In this case, α_{init} was set to 1.0 and λ_{init} to 10^{-3}. After fitting the model on the scaled training data, it produces coefficient and intercept estimates along with uncertainty measures, making it a probabilistic alternative to the standard ridge regression.

Scikit-learn Random Forest Regressor

As a comparison a non-probabilistic model `RandomForestRegressor` from scikit-learn [11] has also been tested. RandomForest is an ensemble method that builds multiple decision trees and averages their predictions to improve accuracy and reduce overfitting. In this setup, the model used 50 trees (`n_estimators = 50`) and the `squared_error` criterion to measure the quality of each split. The model was trained on the scaled training data, though it required noticeably longer computation time compared to the linear models. Random forests can capture complex nonlinear relationships between features and the target, which makes them a useful comparison to the more assumption-driven regression approaches.

Results

Probabilistic Modeling Results

The probabilistic modeling approaches generally showed low uncertainty estimates. Even with the provided weak priors, the calculated posterior distributions for the parameters and standard deviation exhibited low variance as can be seen in Figure 5. The expected sparsity induced by the lasso is not very pronounced which can be seen in Figure 6. While the values mostly seem to be very close to zero, only slightly rounded evaluations provided exact zero values. This may be because the feature selection was based on correlation, meaning only the most highly correlated and thus important features were chosen, making them difficult to remove.

Model performance comparison

The predictive performance of the Random Forest and Bayesian Ridge models was evaluated over multiple forecast horizons, ranging from 1 to 10 years (46 to 460 timesteps). For each horizon, the kNDVI values were removed from the test data to simulate missing observations, and the models predicted these missing values iteratively. The results are depicted in Table 1.

The Random Forest model consistently achieved high predictive accuracy, with R^2 scores starting at 0.959 for 1 year and gradually decreasing to 0.843 over 10 years. Similarly, the Bayesian Ridge regression model exhibited strong performance, with R^2 values beginning at 0.956 and declining to 0.725 across the same period.

While both models maintained robust predictive capabilities, the Random Forest generally outperformed the Bayesian Ridge for longer forecast horizons, especially beyond 5 years. In contrast, Bayesian Ridge showed competitive or slightly superior performance at shorter horizons up to 4 years. These results suggest that the Random Forest model may better capture complex, longer-term dependencies in the data, whereas the Bayesian Ridge excels at shorter timescales.

Table 1 Comparison of R^2 scores for Random Forest and Bayesian Ridge regression models across multiple forecast horizons.

Years	Random Forest R^2	Bayesian Ridge R^2
1	0.959	0.956
2	0.946	0.949
3	0.932	0.935
4	0.922	0.931
5	0.920	0.914
6	0.908	0.907
7	0.906	0.886
8	0.896	0.859
9	0.888	0.805
10	0.843	0.725

Discussion

The results demonstrate that both the Random Forest and Bayesian Ridge models effectively capture the temporal dynamics of the kNDVI time series, exhibiting strong predictive performance across multiple forecast horizons. The Bayesian Ridge model showed slightly better performance for shorter-term predictions, likely due to its ability to leverage the strong linear relationships in the engineered features. Meanwhile, the Random Forest model outperformed for longer horizons, suggesting its strength in modeling more complex, potentially nonlinear interactions over extended periods. The relatively low uncertainty and limited sparsity observed in the probabilistic models imply that the carefully engineered lagged features are highly informative, reducing the need for aggressive regularization or feature elimination.

Limitations of the Approaches

Despite these good outcomes, several limitations are present. The feature engineering relied heavily on autocorrelation and cross-correlation analyses, which, while effective, assume that linear dependencies dominate the relationships between variables. This may overlook subtler nonlinear or non-stationary dynamics. The stepwise iterative prediction procedure, although necessary due to missing data, introduces the risk of error propagation over longer forecast horizons, which is reflected in the gradual decline of performance for both models. The probabilistic models were constrained to subsets of features or data points for computational reasons, potentially limiting their representativeness.

Possible Improvements or Extensions

Future work could explore more flexible feature selection methods that capture nonlinear dependencies, such as kernel methods or embedded feature selection within tree-based models. Incorporating time series specific models, like ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), and state-space models such as Kalman Filters, might better capture complex temporal patterns and dependencies. Lastly, expanding the model evaluation to include more environmental covariates could provide a better understanding of vegetation dynamics.

Conclusion

In the project temporal analysis and predictability of kNDVI using both probabilistic linear models and a non-probabilistic ensemble approach have been examined. Through detailed time series analysis—including seasonality characterization via autocorrelation, cross-correlation, discrete Fourier transforms, and seasonal trend decomposition—key temporal patterns and relevant lagged features were identified to enhance prediction performance.

The incorporation of lagged features reflecting seasonality and variable interactions proved crucial for effective model training and iterative predictions. The findings underscore the importance of seasonality-aware feature construction and the choice of appropriate modeling frameworks in vegetation monitoring as the application context.

In the probabilistic analysis, Bayesian ridge regression and Bayesian Lasso regression have been used to analyze uncertainty in the parameters. The Lasso approach did not substantially increase sparsity, likely because the engineered features were already highly relevant, having been selected based on strong correlations.

Among the evaluated models, the Random Forest regressor demonstrated superior predictive accuracy over longer forecast horizons, capturing complex nonlinear dependencies within the data. In contrast, Bayesian Ridge regression provided competitive performance at shorter time frames while also delivering valuable uncertainty quantification, an asset for informed decision making.

Overall, both approaches were robust, suggesting they are both suited for application after a successful analysis of the data. Future work may explore extended feature engineering, as well as further probabilistic modeling to assess feature properties.

Bibliography

[1] Gustau Camps-Valls et al. “A unified vegetation index for quantifying the terrestrial biosphere”. In: *Science Advances* 7.9 (Feb. 2021), eabc7447. DOI: 10.1126/sciadv.abc7447.

[2] en. URL: https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels?tab=overview.

[3] J. Rouse et al. “Monitoring vegetation systems in the great plains with ERTS”. In: 1973. URL: https://www.semanticscholar.org/paper/Monitoring-vegetation-systems-in-the-great-plains-Rouse-Haas/fb2f60fe0fe2874e5cbf927a2556d719c32eac29.

[4] en. Page Version ID: 1296825487. 2025. URL: https://en.wikipedia.org/w/index.php?title=Normalized_difference_vegetation_index&oldid=1296825487.

[5] Josef Perktold et al. *statsmodels/statsmodels: Release 0.14.2*. Apr. 2024. DOI: 10.5281/ZENODO.593847. URL: https://zenodo.org/doi/10.5281/zenodo.593847.

[6] URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.fft.fft.html.

[7] R. B. Cleveland. “STL : A Seasonal-Trend Decomposition Procedure Based on Loess”. In: 1990. URL: https://api.semanticscholar.org/CorpusID:64570714.

[8] en. URL: https://www.pymc.io/welcome.html.

[9] URL: https://pytensor.readthedocs.io/en/latest/.

[10] en. URL: https://scikit-learn/stable/modules/generated/sklearn.linear_model.BayesianRidge.html.

[11] en. URL: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

Appendix

Anhangsverzeichnis

Probabilistic modeling plots

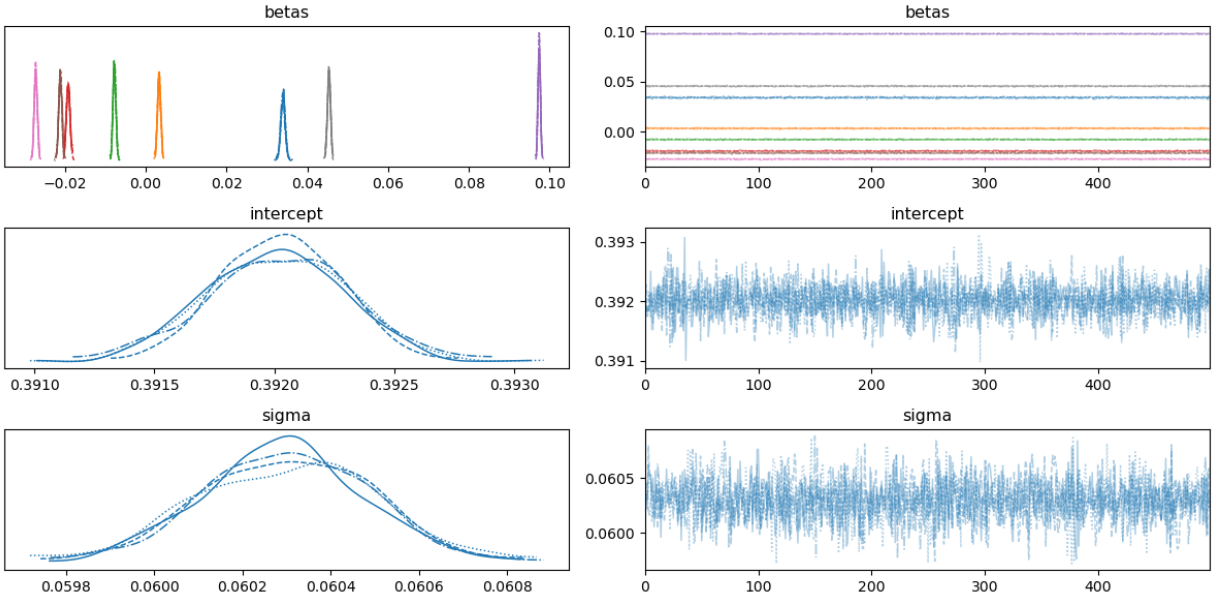

Figure 5 Results of the probabilistic Bayesian linear ridge modeling approach from Section . Low uncertainty for all chosen parameters.

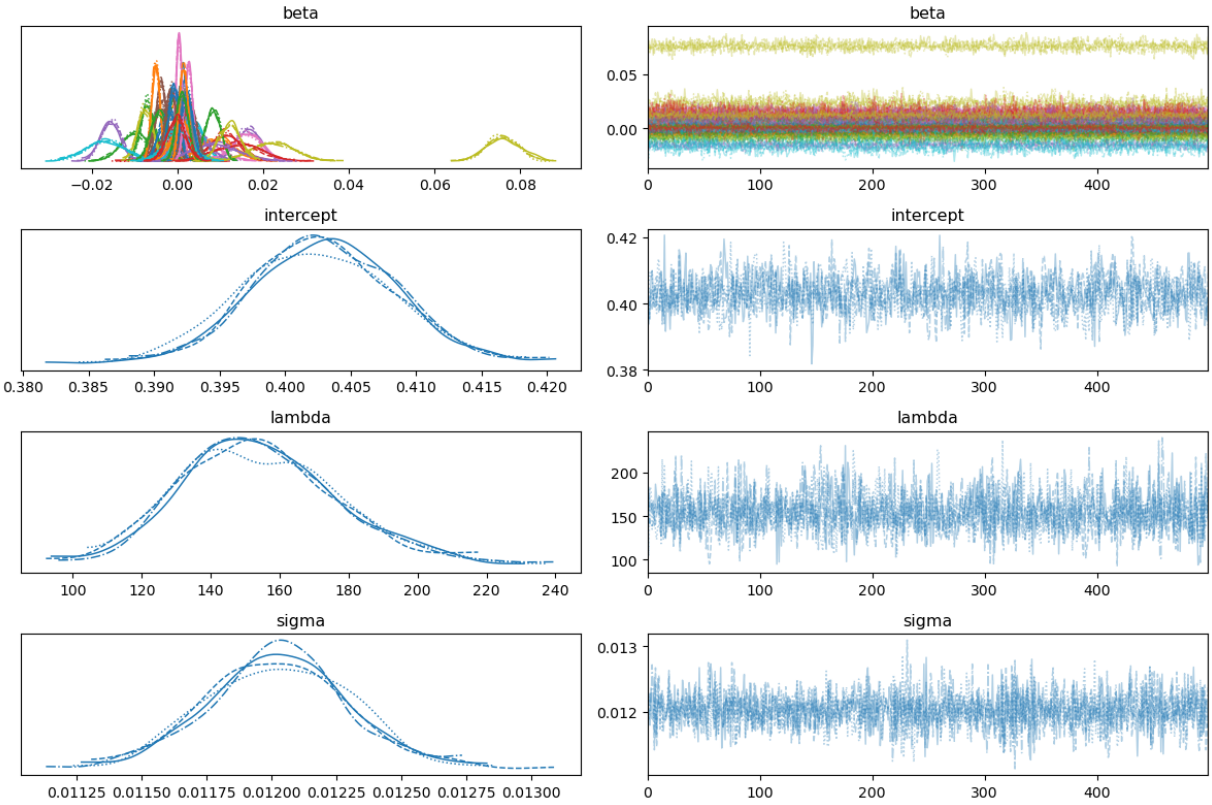

Figure 6 Results of the probabilistic Bayesian lasso modeling approach from Section . Laplace prior places more weight on sparsity.