KRIGING USING GAUSSIAN PROCESSES AND HIDDEN MARKOV MODELS

INTRODUCTION

- **Aim** of the project is to **create** a **model** to **predict** the **value** of a **spatially distributed variable at unsampled locations based on observed data**.
- **Weather data** over a **big area** is considered.
- In particular **daily data** of **multiple weather stations** in **Romania** is used, the data available on Kaggle.

KRIGING

- **Method** implemented in this project is known as **kriging**.
- **Given** the **observations** of **temperature** in **multiple station** find a **gaussian process** that **capture** the **spatial dependencies** between them.
- **This** will **allow** to **interpolate** the **temperature** at **any** given **point inside** the **area** considered.
- The **gaussian processes** was **implemented** with **longitude** and **latitude** as **input feature**, and **minimum temperature** as **target feature**.

GAUSSIAN PROCESSES

- A **gaussian process** is a **distribution** over **functions**.
- Any **finite set of points** $\{x_1, ..., x_n\}$ **from** the **input space** will have a **joint Gaussian distribution** when **mapped through** the **GP**.
- A GP is defined as:

$$f(x) \sim GP(\mu(x), k(x, x'))$$

- $\mu(x)$ is the mean function, and k(x,x') is the kernel function.

GAUSSIAN PROCESSES

- The kernel determines the covariance between the function values at any two points x and x’.
- The kernel function encodes assumptions about the properties of the target function.
- The kernels used in this project are the linear:

$$K_{LIN}(x, x') = xx'$$

- And the rational quadratic:

$$K_{RQ}(x, x') = \sigma^2 \left(1 + \frac{(x - x')^2}{2\alpha l^2} \right)^{-\alpha}$$

REGRESSION

- Input X= $\{x_1, ..., x_n\}$, output$\{y_1, ..., y_n\}$, and $y = f(x) + \epsilon$ with $\epsilon \sim N(0, \sigma_n^2)$
- The prior between an observation x and a test point x* is:

$$\begin{bmatrix} y \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,x^*) \\ K(x^*,X) & K(x^*,x^*) \end{bmatrix}\right)$$

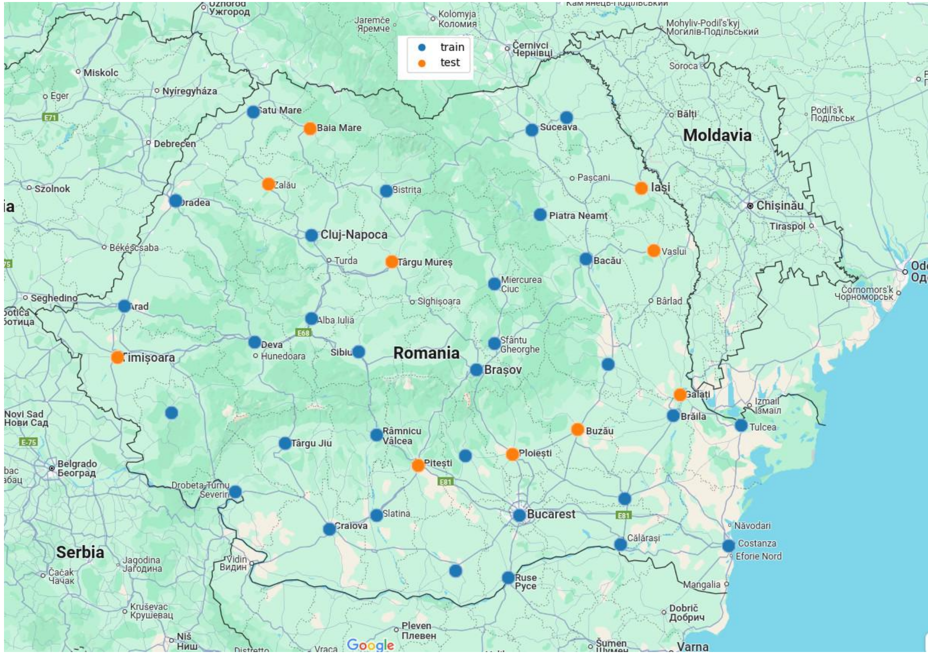
- while the posterior is $f(x^*) \mid X, y, x^* \sim \mathcal{N}\left(\mu^*(x^*), \sigma^{*2}(x^*)\right)$,

  with mean: $\mu^* = K(x^*, X)\left[K(X,X) + \sigma_n^2 I\right]^{-1} y$

  and variance: $\sigma^{*2} = K(x^*, x^*) - K(x^*, X)\left[K(X,X) + \sigma_n^2 I\right]^{-1} K(X, x^*)$
- So the GP return an estimate of the function value at x* given the observed data, as well as the uncertainty about this estimate.

DATA EXPLORATION

- **Dataset used** is composed of **daily observations** of multiple data (min/max temperature, precipitation…) at multiple **weather stations** in Romania.
- Out of the **42 stations** available **32** were randomly selected to **train** the model and the others to test it.
- **Data** spatially **ordered** based on **Longitude** and **latitude**.



IMPLEMENTATION

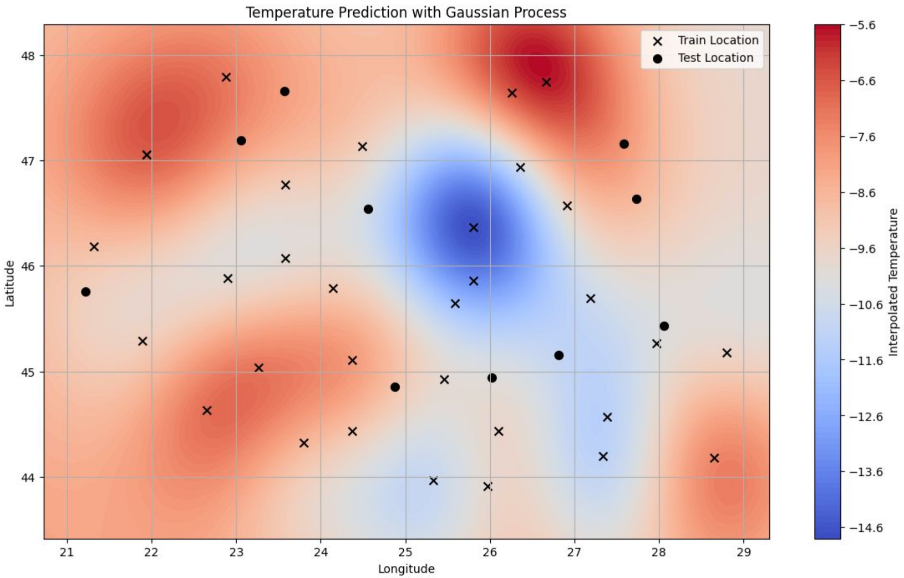
- To implement gaussian processes the package gpflow was used.
- The **kernel selected** is the **sum** of a **linear** and a **rational quadratic** one, plus a noise kernel.
- The **linear** kernel will catch the possible **linear dependencies** based on longitude and latitude.
- The **rational quadratic** kernel will catch **eventual more complex dependencies**.

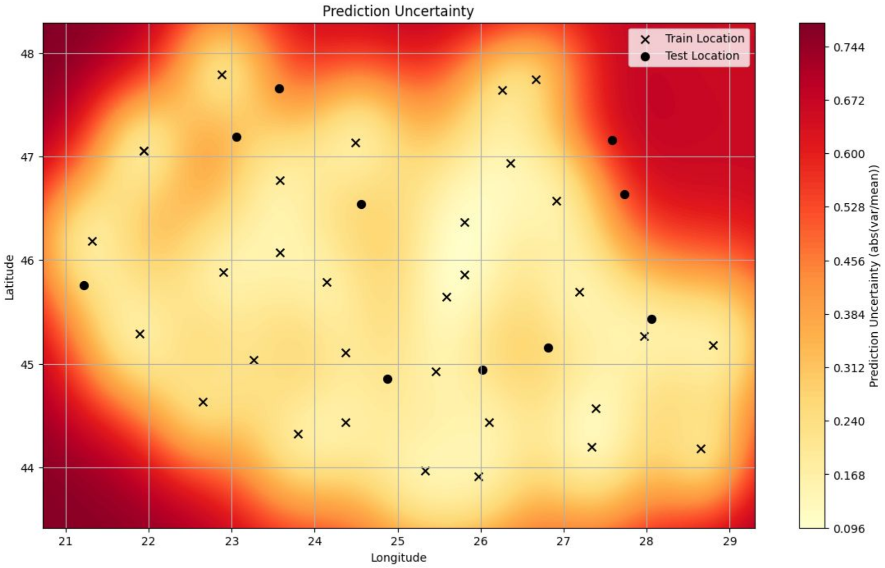
IMPLEMENTATION

- The **kriging** model is **trained on** a **single day**.
- **Given** the **measured minimum temperature** at the **train stations**, the **hyperparameters** of the **kernels** are **optimized to best fit** the observations.
- The **optimization** is done by **minimizing** the **negative log marginal likelihood**.
- Once the **GP** is **trained** it is possible to **interpolate** the **minimum temperature** at **any longitude/latitude couple**.
- The **GP** will also **return** the **variance** of the interpolation.

IMPLEMENTATION

- Application on a grid of longitude/latitude points.





- The farther from train point the higher the variance.

PERFORMANCE EVALUATION

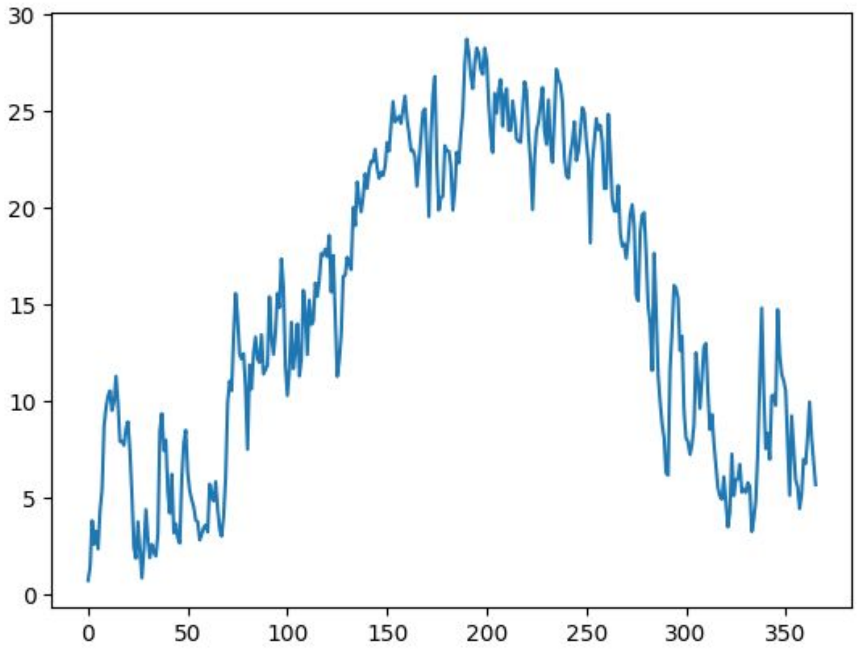
- To **evaluate** the model **performance** the **mean absolute error** on the ten test station was considered.
- for this case it is **0.72**, which is good.

	Latitude	Longitude	True Temperature	Interpolated Temperature	Variance	absolute error
0	47.6572	23.5660	-6.1	-8.032682	3.198266	1.932682
1	45.1511	26.8174	-11.2	-11.043236	2.679019	0.156764
2	45.4337	28.0548	-10.3	-10.183949	1.874984	0.116051
3	47.1594	27.5873	-7.8	-7.841039	4.515648	0.041039
4	44.8515	24.8799	-8.3	-9.077781	1.864657	0.777781
5	44.9407	26.0233	-10.0	-10.136955	2.083504	0.136955
6	46.5426	24.5580	-10.0	-10.599646	2.663611	0.599646
7	45.7534	21.2233	-11.8	-9.400043	2.464442	2.399957
8	46.6385	27.7331	-8.2	-8.694581	4.389910	0.494581
9	47.1878	23.0579	-7.3	-7.850125	2.430556	0.550125

- this is the **performance** on the **single day**, what about other days?

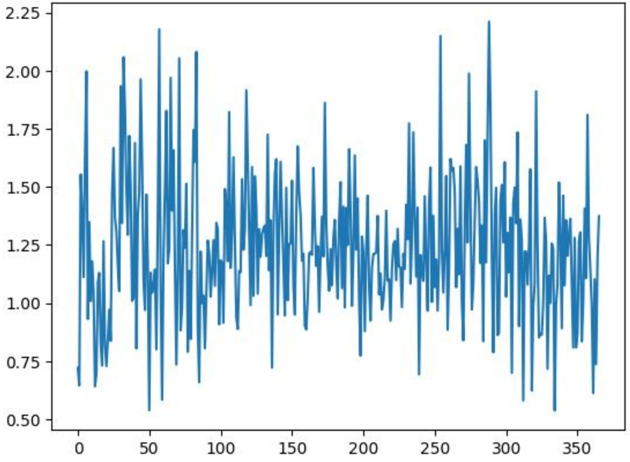
PERFORMANCE EVALUATION

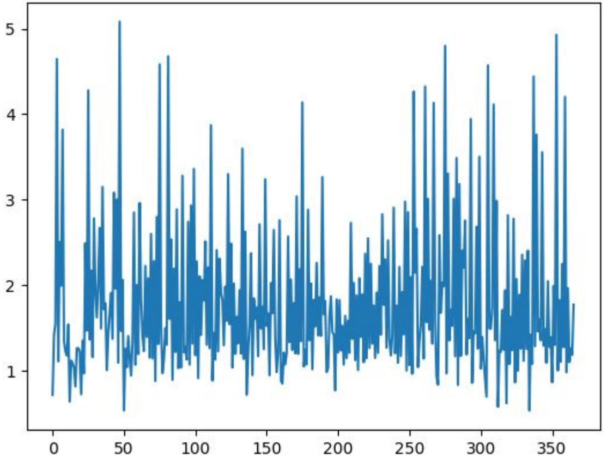
- At **first** the **same model** was **tested** on **multiple days**.
- The **mae** is **acceptable** for a **couple** of **days** then it blows up.
- The **spatial dependencies varies** based on the **day**.



PERFORMANCE EVALUATION

- Then a new **model** was created **daily** and another **bi-daily**.





- **Daily** models perform **good**, mean MAE of 1.25, **bi-daily** is **worse**, similar mean but higher volatility (because of second days).

CONSIDERATIONS

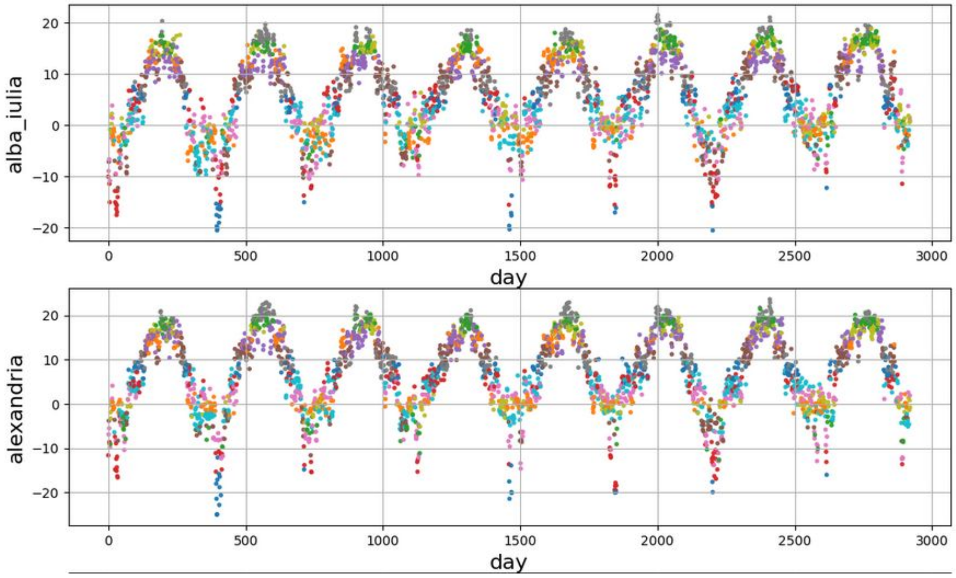
- **Gaussian processes scales cubically** with **data.**
- When dealing with **more stations** and **modeling** a **lot of days**, **or** when **data comes** in **hourly** instead of daily, **training** this **many GPs** is **not feasible**.
- The **spatial dependencies** might **vary based** on the **weather conditions**, so a **hidden markov model** was **trained** to **detect** the **weather state**, and a **different GP** was trained for **each** found **state**.

HIDDEN MARKOV MODELS

- **HMM** is a **statistical markov model** in which the **system** being **modeled** is assumed to be a **Markov process** with **unobserved hidden states**.
- They are **characterized** by the **initial distribution**, the **transition probabilities** and the **emission probabilities**.
- **In this case** the **interests** lies **only** in being able to **assign** the **correct hidden states** to **observations** (most likely explanation).
- HMM implemented with the package hmmlearn.

HIDDEN MARKOV MODELS

- First hmm **trained** only **on minimum temperature** of the train stations, and for eight years of data, the **temperature** is supposed to be relatively **homogenous across** the whole **area.**
- **20** different **states** identified.
- The **hidden states** seems to be **based** on the **mean temperature** across the **stations**.
- They **might have caught** some **other dependencies**.

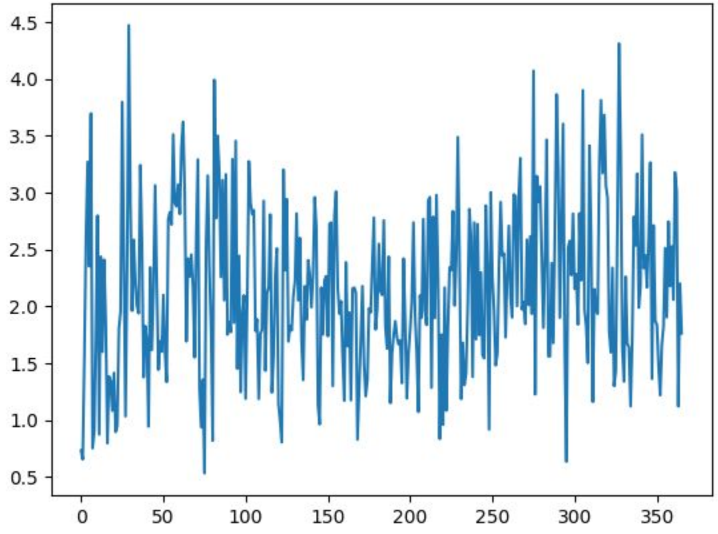


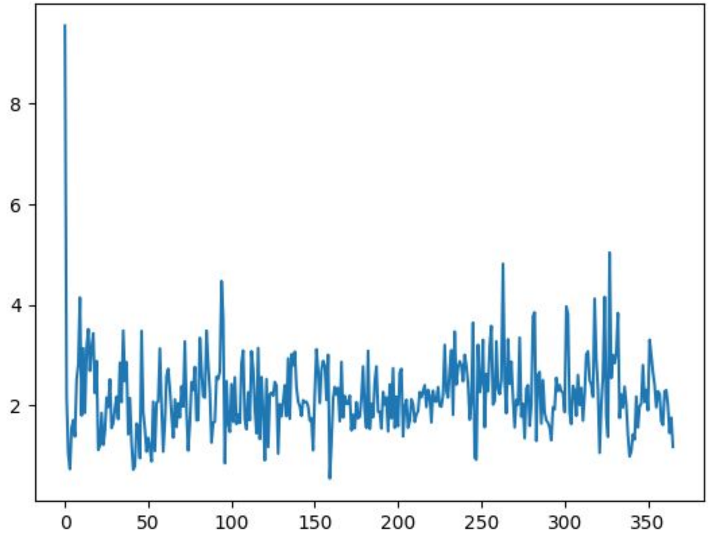
HIDDEN MARKOV MODELS: GP

- **For each state** a **different GP** was **trained**, to train selected the first occurrence of the state.
- The **GPs** were tested on the **first year of data** (same as before), and on the first **year of data unseen** from the **HMM**, so eight years after the other.
- In both cases the **most likely sequence** of **hidden states** is selected according to **Viterbi algorithm**, and the corresponding GP for the interpolation is selected accordingly.

HIDDEN MARKOV MODELS: GP

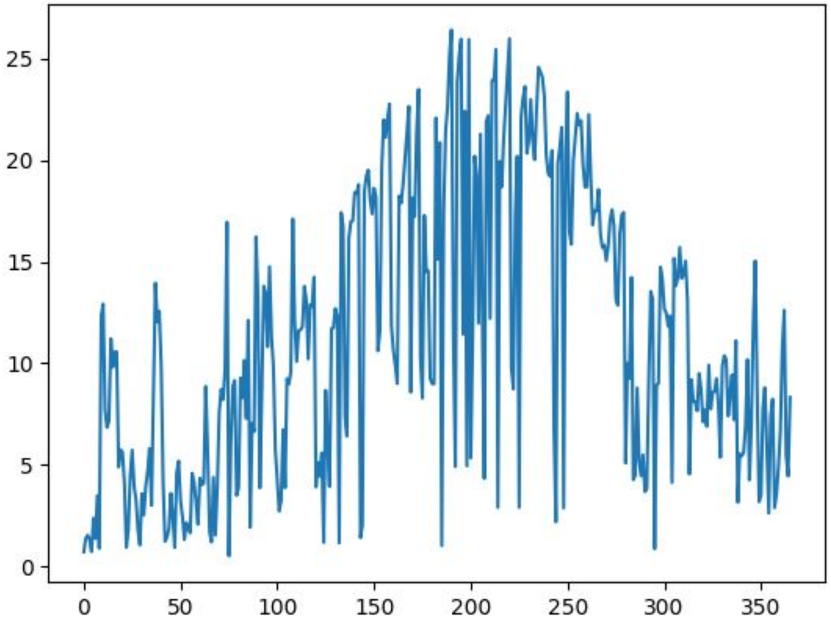
- Performance on first year worse than daily models, but comparable with bi-daily.
- 8 years later performance still comparable.





HIDDEN MARKOV MODELS: GP

- **Same process** followed **training HMM** with **other weather data** (relative humidity, precipitation, cloud cover, wind speed, pressure).
- Very **bad results obtained,** probably because other **variable** are **not homogenous enough** across the area, so a **single weather state** for the whole area is **not enough**.



CONCLUSION

- With this method **possible** to **interpolate temperature across** all **region**.
- It is **possible** to **identify** weather **states that corresponds** to **different spatial dependencies** of temperature.
- A single **GP for each state** will yield **good interpolations across** the **years**.
- GP could be enhanced adding other spatial data such as altitude.
- **Understanding** better the **dependencies of** the **states, better kernels** could be select accordingly.
- Finally it should be possible to implement forecast model to have forecast across all area.