Regression as GP recipe [1,2]

- Choose appropriate kernel for data
- Determine prior distribution $\mathcal{N}(0, \Sigma_{YY})$ with training data Y
- Determine posterior distribution with conditioning X

$$\mathcal{N}(\Sigma_{XY}^T \Sigma_{YY}^{-1}\, Y, \Sigma_{XX} - \Sigma_{YX}^T\, \Sigma_{YY}^{-1}\, \Sigma_{\mathrm{YX}})$$

- If needed add noise of data σ_{Data}

$$\mathcal{N}(\Sigma_{XY}^T(\Sigma_{YY}^{-1} + \sigma^2 I)Y, \Sigma_{XX} - \Sigma_{YX}^T(\Sigma_{YY}^{-1} + \sigma^2 I)\Sigma_{\mathrm{YX}})$$

- With marginalization extract any μ_i / σ_i with $\sigma_i^2 = \Sigma_{ii}$
  - Got variance of prediction/ confidence of prediction

[1] K. Weinberger, Machine Learning Lecture: Gaussian Process (https://www.cs .cornell.edu/courses/cs4780/2018fa/lectures/lecturenote15.html)
[2] C. Fonnesbeck, Fitting Gaussian Process Models in Python (https://domino. ai/blog/fitting-gaussian-process-models-python)