Bayesian Machine Learning Report

Konstantinos Azas

17th May 2022

Individual Contribution	
Coding	40%
Theoretical/mathematical derivation	40%
Useful thoughts/opinions	40%

Contents

1 Introduction

This report illustrates various Bayesian modelling techniques used to construct real multivariate regression models. The aim is find good predictors that fit with the "energy efficiency" data that are provided. In Bayesian modelling, this is done by observing the unknown structure of the posterior distribution that the data follow. The methods used to approximate from the posterior would be Bayesian linear regression mentioned in section 3, Hamiltonian Monte Carlo (HMC) mentioned in 4 and Gaussian processes mentioned in 5.

The "energy efficiency" data are composed of 768 examples, each having 9 features. One feature comprises 1 constant bias and the rest of the input variables give measurements for basic architectural parameters for buildings (e.g. "Roof Area" and "Glazing area"). The target variable in the dataset is "Heating Load". Performing some pre-processing on the data, they are being split equally into training and test data. In many occasions, the data inputs might have measurements based on different scales, which may result in a bias while a model is fitted [3]. And to avoid any issues with different scales for different inputs, it is optimal to standardise the data, also ensuring the assumption that they follow a Gaussian distribution. Standardisation is applied with mean 0 and variance 1.

2 Exploratory Analysis

Initially, an exploratory analysis for the data variables is undertaken to observe which of them are considered more "relevant" for the prediction of the target variable. Further comments can also be included to describe their linearity or otherwise. To prove this, several plots showing correlations were included for visual observations as well as Ordinary Least Squares for numerical observations.

2.1 Correlations

The data appear to be non-Gaussian and for this the Spearman's correlation coefficient has been used as a metric to determine how relevant each input is with the target. Plots shown in figure 1 demonstrate the association between each of the input variables with "Heating Load".

From the correlation plots, it is observed that the inputs that are the most "irrelevant" are "Orientation" and "Glazing Area Distribution" as their correlation coefficient is near to 0. The "Glazing Area" input might also be considered irrelevant with a correlation of 0.26. Hence, these can be excluded to the model fitted for the prediction of the target. Another visualisation is illustrated with a heatmap as well shown in figure 2.

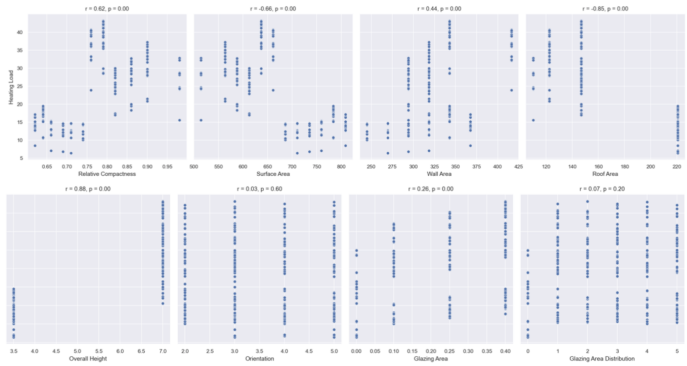

Figure 1: Correlations

Figure 2: Correlation heatmap

2.2 Ordinary Least Squares

This is a regression method used to estimates the unknown weights of a linear model. Each input variable has its own weight which is to be estimated and used for further predictions. The data will be fitted using least squares and metrics such as MAE and RMSE are also evaluated to observe the performance of the model. From table 3, by looking at the p-values there is evidence that "Glazing Area" might be considered relevant to be included in the model.

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.910
Model:	OLS	**Adj. R-squared:**	0.909
Method:	Least Squares	**F-statistic:**	544.8
Date:	Sat, 14 May 2022	**Prob (F-statistic):**	1.72e-192
Time:	20:18:49	**Log-Likelihood:**	-968.22
No. Observations:	384	**AIC:**	1952.
Df Residuals:	376	**BIC:**	1984.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	22.9207	0.155	147.581	0.000	22.615	23.226
x1	-7.2346	1.654	-4.373	0.000	-10.488	-3.982
x2	-3.9422	1.206	-3.269	0.001	-6.314	-1.571
x3	0.7560	0.312	2.420	0.016	0.142	1.370
x4	-4.2319	1.091	-3.880	0.000	-6.376	-2.087
x5	7.2040	0.858	8.401	0.000	5.518	8.890
x6	-0.1252	0.156	-0.803	0.423	-0.432	0.182
x7	2.7702	0.162	17.129	0.000	2.452	3.088
x8	0.2041	0.161	1.267	0.206	-0.113	0.521

Omnibus:	7.903	**Durbin-Watson:**	1.890
Prob(Omnibus):	0.019	**Jarque-Bera (JB):**	11.852
Skew:	0.109	**Prob(JB):**	0.00267
Kurtosis:	3.832	**Cond. No.**	8.07e+15

Figure 3: OLS Estimation

Metric	**Train Error**
RMSE	3.0115517876503617
MAE	2.1306794414069143
Metric	**Test Error**
RMSE	2.8435880167333694
MAE	2.069010093808354

Table 1: The RMSE and MAE of the train and test sets for OLS

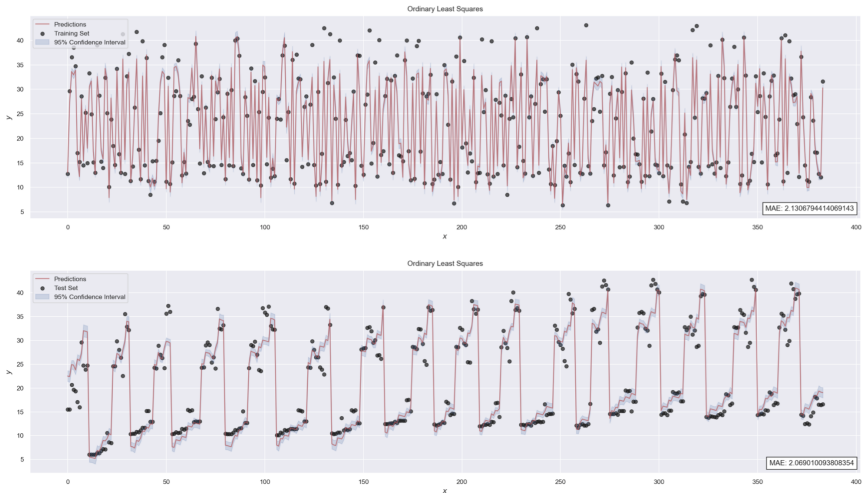

Figure 4: OLS predictions for training and test set

3 Bayesian Linear Regression

In this section, a standard linear regression model is performed with unknown coefficient set $\mathbf{w}$. It is formulated as:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon \tag{1}$$

where $\mathbf{y}$ is "Heating Load", $\mathbf{X}$ is the data for all examples and fetaures and ϵ is the Gaussian noise. Several assumptions are considered. These are:

- $\mathbf{w}$ is assumed to have a Gaussian prior, such as $\mathbf{w} \sim \mathcal{N}(0, \sigma^2_{\mathbf{w}})$, where $\alpha = \frac{1}{\sigma^2_{\mathbf{w}}}$ is the precision of prior.
- Gaussian noise is modelled as $\mathcal{N}(0, \sigma^2_\epsilon)$, where $\beta = \frac{1}{\sigma^2_\epsilon}$ is the precision of noise.
- Now we have the unknown hyper-parameter set $\theta = (\sigma^2_\epsilon, \sigma^2_{\mathbf{w}}) = (\alpha, \beta)$.
- If we denote the observation data as D, the posterior we want to estimate can be written as $p(\mathbf{w}, \theta|D)$.

3.1 Type-II Maximum Likelihood

The expression $p(\mathbf{w}, \theta|D)$ in this scenario will be formulated as $p(\mathbf{w}, \alpha, \beta|\mathbf{y})$ and using the product rule it turns to be:

$$p(\mathbf{w}, \alpha, \beta|\mathbf{y}) = p(\mathbf{w}|\alpha, \beta, \mathbf{y})p(\alpha, \beta|\mathbf{y}) \tag{2}$$

The second term is equal to:

$$p(\alpha, \beta|\mathbf{y}) = \frac{p(\mathbf{y}|\alpha, \beta)p(\alpha)p(\beta)}{p(\mathbf{y})} \tag{3}$$

Hence, the final equation is:

$$p(\mathbf{w}, \alpha, \beta|\mathbf{y}) = \frac{p(\mathbf{w}|\alpha, \beta, \mathbf{y})p(\mathbf{y}|\alpha, \beta)p(\alpha)p(\beta)}{p(\mathbf{y})} \tag{4}$$

Assuming that $p(\alpha)$ and $p(\beta)$ are uninformative priors, only the marginal likelihood is maximised. It is formulated as:

$$\alpha, \beta = \underset{\alpha,\beta}{\operatorname{argmax}} \log p(\mathbf{y}|\alpha, \beta) \tag{5}$$

To determine what $p(\mathbf{y}|\alpha, \beta)$ is, we need to determine the distribution of $\mathbf{y}$ which is formulated as $\mathcal{N}(\mu, \mathbf{\Sigma})$. The parameters will be derived from the initial linear regression model in 1.

Proof.

$$\begin{aligned}
\mathbb{E}(\mathbf{y}) &= \mathbb{E}(\mathbf{X}\mathbf{w} + \epsilon) \\
&= \mathbf{X}\,\mathbb{E}(\mathbf{w}) + \mathbb{E}(\epsilon) \\
&= \mathbf{0}
\end{aligned}$$

and for the variance,

$$\begin{aligned}
Var(\mathbf{y}) &= Var(\mathbf{X}\mathbf{w} + \epsilon) \\
&= \mathbf{X}Var(\mathbf{w})\mathbf{X^T} + Var(\epsilon) \\
&= \frac{\mathbf{X}\mathbf{X^T}}{\alpha} + \frac{1}{\beta}\mathbf{I}
\end{aligned}$$

□

Hence, the log-marginal likelihood is formulated as:

$$\log\left(\frac{1}{(2\pi\mathbf{\Sigma})^{\frac{n}{2}}}\exp\left(\frac{1}{2}(\mathbf{y^T}\mathbf{\Sigma^{-1}}\mathbf{y})\right)\right) \tag{6}$$

To determine the most probable α and β, random samples are generated uniformly from -5 to 0 for both hyperparameters. The combination of the ones that achieve the highest log-marginal likelihood are the maximum likelihood estimates. A visualisation is given below in 5. Then, those are used to compute the posterior distribution to determine predictions for new data points. A table with the metrics and the most probable values is illustrated below in 3.

Priors	
α	0.01174362845702136
β	0.10836802322189586
$\log(\alpha)$	-4.444444444444445
$\log(\beta)$	-2.2222222222222223
Log-Likelihood	-295.71283172405134

Table 2: Type-II most probable values

Metric	**Train Error**
RMSE	3.011551809679529
MAE	2.130668537391788
Metric	**Test Error**
RMSE	2.8063010631019174
MAE	1.9907650699570147

Table 3: The RMSE and MAE of the train and test sets for Type-II

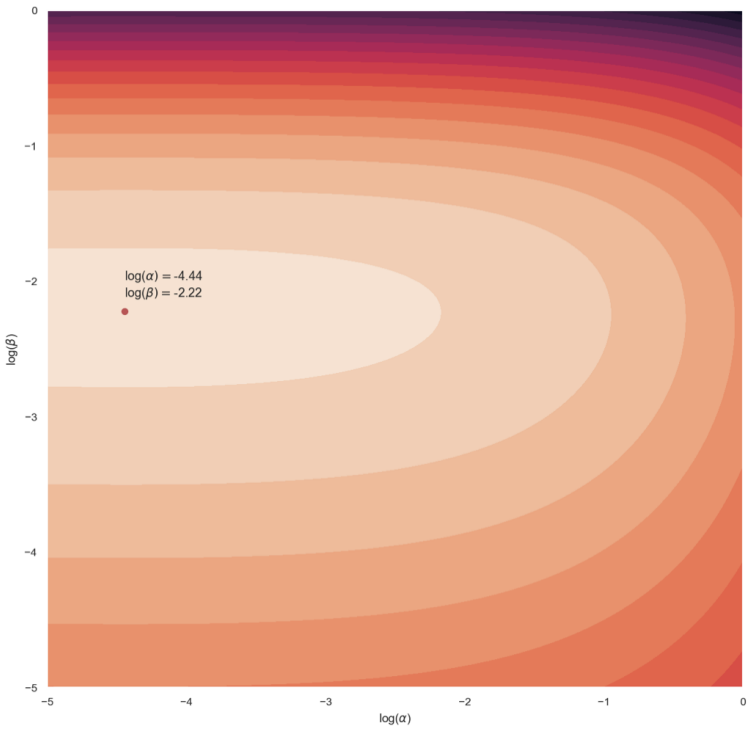

Figure 5: Type-II contour plot

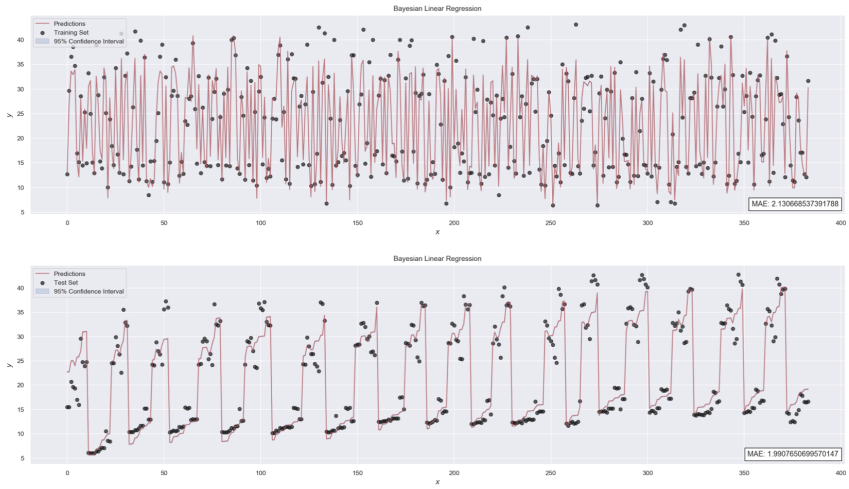

Figure 6: Type-II maximum likelihood predictions for training and test set

3.2 Variational Inference

Variational inference is a deterministic approximation method used to sample parameters from the unknown posterior distribution $P(\theta)$, previously shown in equation 4. This is done, by choosing a "proposal" distribution $Q(\theta)$ as an initial guess of the posterior. The aim is to minimise some measure of "difference" between $Q(\theta)$ and $P(\theta)$. In this scenario, using "Mean-Field Theory" factorisation the proposal is formulated as:

$$\mathcal{Q}(\mathbf{w}, \alpha, \beta) = \mathcal{Q}(\mathbf{w})\mathcal{Q}(\beta)\mathcal{Q}(\alpha) \tag{7}$$

where:

$$\begin{aligned}\mathcal{Q}(\mathbf{w}) &= \mathcal{N}(\mathbf{w}|0, (\alpha\beta)^{-1}) \\ \mathcal{Q}(\beta) &= \mathcal{G}(\alpha|a, b) \\ \mathcal{Q}(\alpha) &= \mathcal{G}(\beta|c, d)\end{aligned}$$

The posterior mean and covariance are formulated as:

$$\begin{aligned}\mathbf{\Sigma} &= \left(\frac{c_N}{d_N}\mathbf{X^T X} + \frac{a_N}{b_N}\mathbf{I}\right)^{-1} \\ \mu &= \frac{c_N}{d_N}\mathbf{\Sigma X^T y}\end{aligned}$$

where:

$$b_N = b_0 + \frac{1}{2}(\mu^{\mathbf{T}}\mu + tr(\mathbf{\Sigma}))$$
$$d_N = d_0 + \frac{1}{2}(\mathbf{y} - \mathbf{X}\mu)^T(\mathbf{y} - \mathbf{X}\mu)$$
$$a_N = a_0 + \frac{M}{2}$$
$$c_N = c_0 + \frac{N}{2}$$

where M is the number of features of $\mathbf{X}$ and N is the number of examples of $\mathbf{X}$. The above hyperparameters are used to compute α and β which are formulated as:

$$\alpha = \frac{a_N}{b_N}$$
$$\beta = \frac{c_N}{d_N}$$

which basically it is the expected value of a Gamma distribution. In the algorithm, a_N and c_N are kept fixed. Only b_N and d_N vary. The contour plot and metrics are shown below in 7 and 5.

Priors	
α	0.011916753508509775
β	0.11026038962261975
$\log(\alpha)$	-4.42981001110939
$\log(\beta)$	-2.2049105321553415
Log-Likelihood	-295.7527268132292

Table 4: Variational Inference most probable values

Metric	**Train Error**
RMSE	3.0115518111318993
MAE	2.1306681834494188
Metric	**Test Error**
RMSE	2.8435839397385267
MAE	2.0689828677240825

Table 5: The RMSE and MAE of the train and test sets for Variational Inference

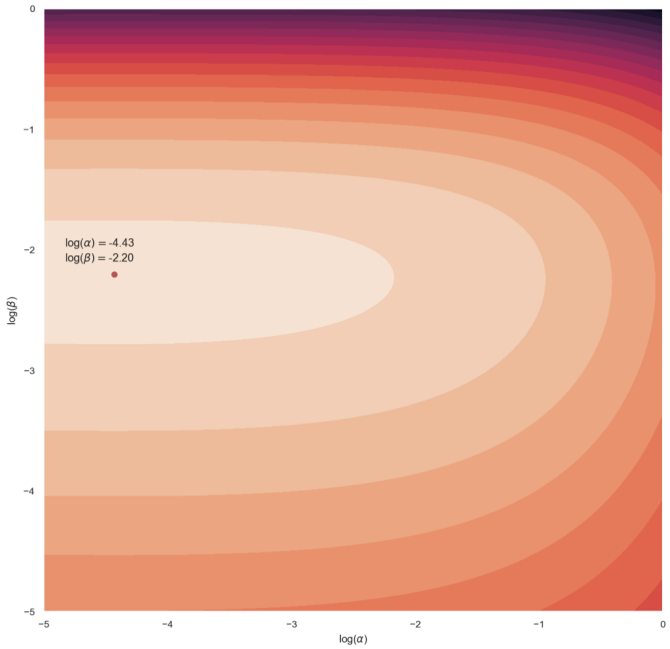

Figure 7: Variational Inference contour plot

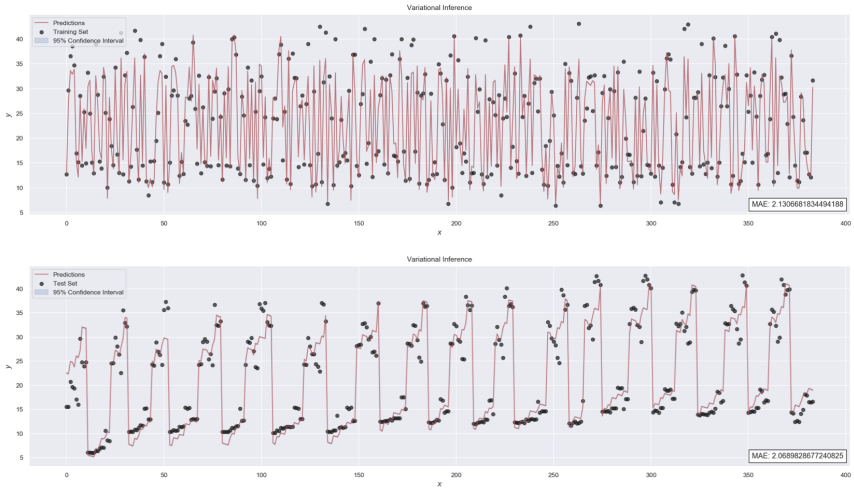

Figure 8: Variational inference predictions for training and test set

4 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo is a stochastic approximation method which aims to explore the state space using the "Hamiltonian Dynamics". To achieve this we introduce the "energy" function which is formulated as:

$$H(\theta, \rho) = -\log P(\theta) + \frac{1}{2}||\rho^2|| \tag{8}$$

where the dynamics are defined as:

$$\frac{\partial \theta_i}{\partial t} = \frac{\partial H}{\partial \rho_i}$$
$$\frac{\partial \rho_i}{\partial t} = -\frac{\partial H}{\partial \theta_i}$$

The acceptance ratio for proposing a new state θ' is formulated as:

$$\min\left(1, \exp(H(\theta, \rho) - H(\theta', \rho'))\right) \tag{9}$$

HMC will be implemented on a 2D Gaussian example and on the linear regression model.

4.1 2D Gaussian example

In this example, a 2D Gaussian is chosen for the energy. So the negative log-likelihood would be:

$$\frac{1}{2}\log(2\pi\mathbf{\Sigma}) + \frac{1}{2}\mathbf{x^T\Sigma^{-1}x} \tag{10}$$

where $\mathbf{x} = (x_0, x_1)$. The derivation of the gradients is given below.

Proof. Suppose $\mathbf{\Sigma^{-1}} = \begin{pmatrix} \sigma_{00}^2 & \sigma_{10}^2 \\ \sigma_{01}^2 & \sigma_{11}^2 \end{pmatrix}$

$$\begin{aligned} \frac{1}{2}\mathbf{x^T\Sigma^{-1}x} &= \frac{1}{2}(x_0, x_1)\begin{pmatrix} \sigma_{00}^2 & \sigma_{10}^2 \\ \sigma_{01}^2 & \sigma_{11}^2 \end{pmatrix}\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \\ &= \sigma_{00}^2 x_0^2 + 2\sigma_{10}^2 x_0 x_1 + \sigma_{11}^2 x_1^2 \end{aligned}$$

Then,

$$\begin{aligned} \frac{\partial(-\log P(\theta))}{\partial x_0} &= \frac{1}{2}(2\sigma_{00}^2 x_0 + \sigma_{10}^2 x_1) \\ &= \sigma_{00}^2 x_0 + \sigma_{10}^2 x_1 \\ \frac{\partial(-\log P(\theta))}{\partial x_1} &= \frac{1}{2}(2\sigma_{10}^2 x_0 + \sigma_{11}^2 x_1) \\ &= \sigma_{10}^2 x_0 + \sigma_{11}^2 x_1 \end{aligned}$$

□

Hyperparameters	
R	5000
L	20
eps	0.36

Table 6: Hyperparameters chosed for 2D Gaussian example

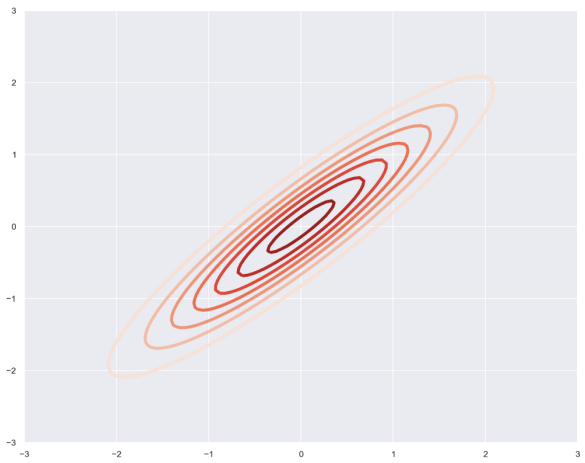

Figure 9: Centred 2D Gaussian

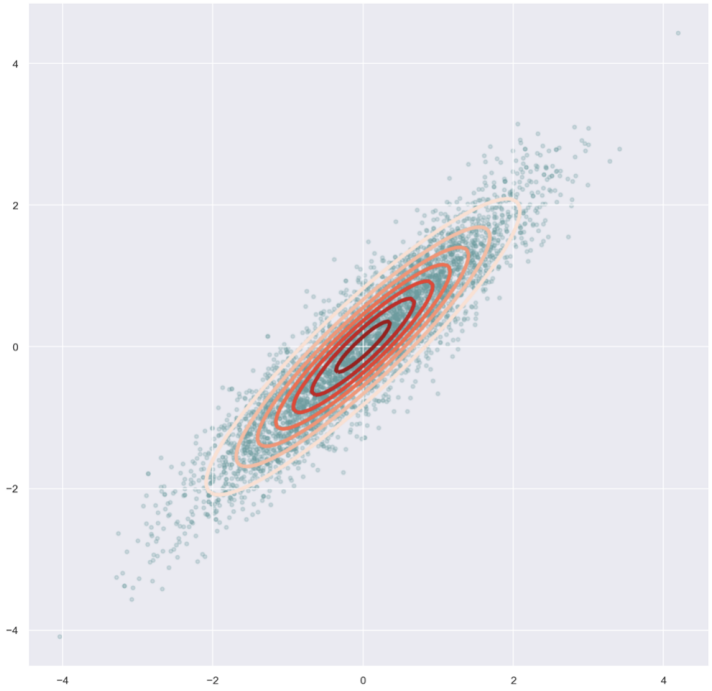

Figure 10: HMC on 2D Gaussian

```
Calc.          Numeric        Delta          Acc.
    -4.87878       -4.87878    1.570930e-10   11
     4.53138        4.53138   -1.379288e-10   11
|----------|  0% accepted [ 3 secs to go ]
|#---------| 92% accepted [ 2 secs to go ]
|##--------| 92% accepted [ 2 secs to go ]
|###-------| 91% accepted [ 2 secs to go ]
|####------| 91% accepted [ 2 secs to go ]
|#####-----| 91% accepted [ 1 secs to go ]
|######----| 90% accepted [ 1 secs to go ]
|#######---| 90% accepted [ 1 secs to go ]
|########--| 90% accepted [ 1 secs to go ]
|#########-| 90% accepted [ 0 secs to go ]
|##########| 90% accepted [ 0 secs to go ]
HMC: R=5000 / L=20 / eps=0.36 / Accept=90.4%
```

Figure 11: Acceptance rate for 2D Gaussian

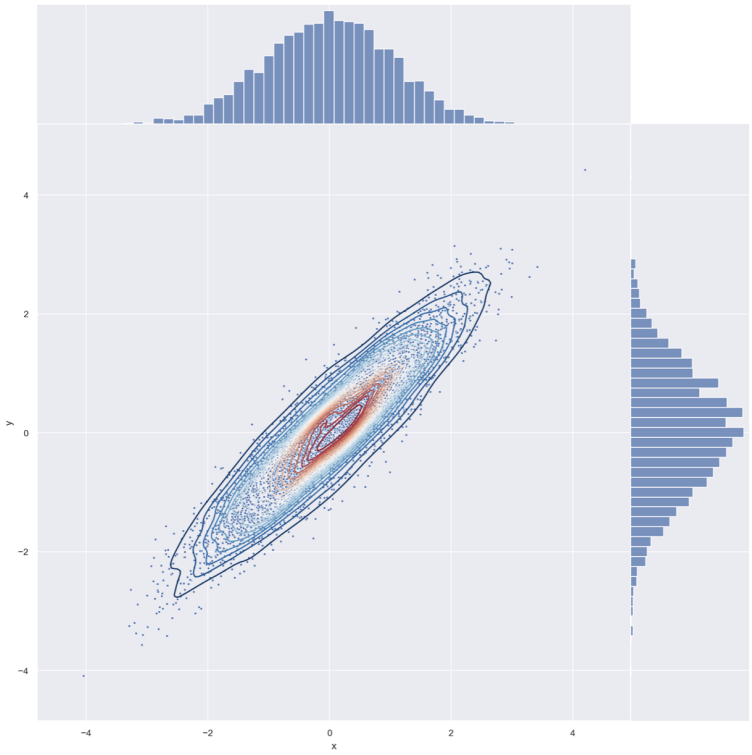

Figure 12: HMC on 2D Gaussian with marginal distributions

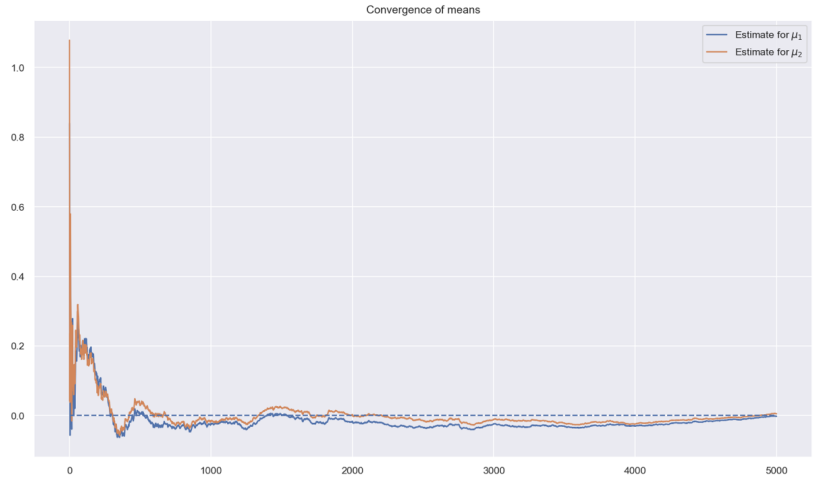

Figure 13: Convergence of means for 2D Gaussian

4.2 Linear regression

The posterior given in 2 is used for the energy function where:

$$p(\mathbf{y}|\mathbf{X},\beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{n}{2}} \exp\left(\frac{-\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})\right) \tag{11}$$

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{m}{2}} \exp\left(\frac{-\alpha}{2}(\mathbf{w}^\mathbf{T}\mathbf{w})\right) \tag{12}$$

Now, the energy function will be the joint distribution of 11 and 12.

Proof.

$$\begin{aligned}
-L &= -\frac{n}{2}\log(\beta) + \frac{n}{2}\log(2\pi) + \frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})... \\
&... - \frac{m}{2}\log(\alpha) + \frac{m}{2}\log(2\pi) + \frac{\alpha}{2}\mathbf{w}^\mathbf{T}\mathbf{w} \\
&\propto -\frac{n}{2}\log(\beta) + \frac{\beta}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) - \frac{m}{2}\log(\alpha) + \frac{\alpha}{2}\mathbf{w}^\mathbf{T}\mathbf{w} \\
\frac{\partial(-L)}{\partial\alpha} &= -\frac{m}{2\alpha} + \frac{1}{2}\mathbf{w}^\mathbf{T}\mathbf{w} \\
\frac{\partial(-L)}{\partial\beta} &= -\frac{n}{2\beta} + \frac{1}{2}(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w}) \\
\frac{\partial(-L)}{\partial\mathbf{w}} &= -\beta(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{X}) + \alpha\mathbf{w}^\mathbf{T}
\end{aligned}$$

□

Hyperparameters	
R	20000
L	100
eps	0.0025

Table 7: Hyperparameters for Linear regression HMC

Optimal Values	
α	0.014306838372461161
β	0.10854886876405839
Bias	0.20568049399730945

Table 8: Optimal values of the unknown terms for HMC Linear Regression

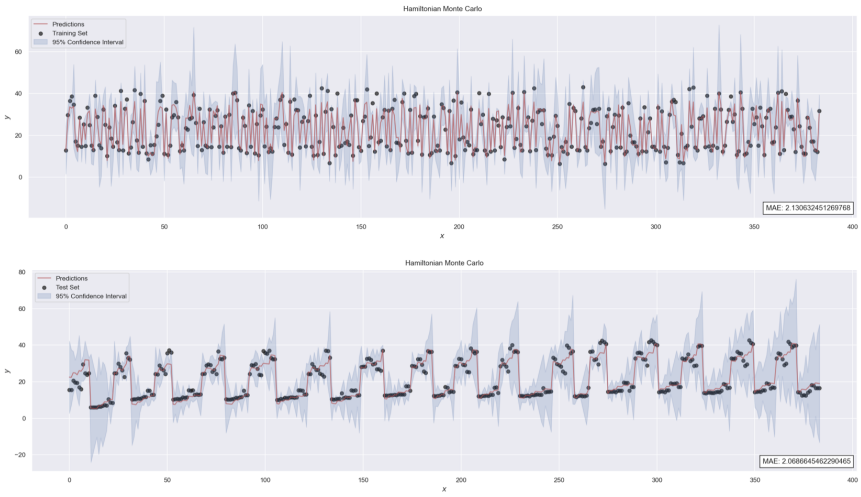

Figure 14: HMC for linear regression predictions for training and test set

```
Calc.          Numeric        Delta          Acc.
     290.633        290.633  -3.982234e-08   10
    -178.667       -178.667  -4.401591e-08   10
     2.28905        2.28905   2.599009e-09    9
   -0.722124      -0.722124  -1.016047e-08    8
    -0.39343       -0.39343  -2.445999e-08    8
   0.0755569       0.075557   2.518834e-08    7
   -0.422401      -0.422401  -1.829638e-08    8
    0.719567       0.719567  -7.127348e-08    8
  -0.0125016     -0.0125017  -6.799340e-08    6
    0.276659       0.276659  -7.174795e-08    7
   0.0203778      0.0203778  -1.893499e-08    7
|----------|  0% accepted [ 63 secs to go ]
|#---------| 90% accepted [ 56 secs to go ]
|##--------| 91% accepted [ 49 secs to go ]
|###-------| 92% accepted [ 43 secs to go ]
|####------| 90% accepted [ 37 secs to go ]
|#####-----| 90% accepted [ 31 secs to go ]
|######----| 90% accepted [ 25 secs to go ]
|#######---| 90% accepted [ 18 secs to go ]
|########--| 89% accepted [ 12 secs to go ]
|#########-| 90% accepted [ 6 secs to go ]
|##########| 90% accepted [ 0 secs to go ]
HMC: R=20000 / L=100 / eps=0.0025 / Accept=89.6%
```

Figure 15: Acceptance rate for linear regression HMC

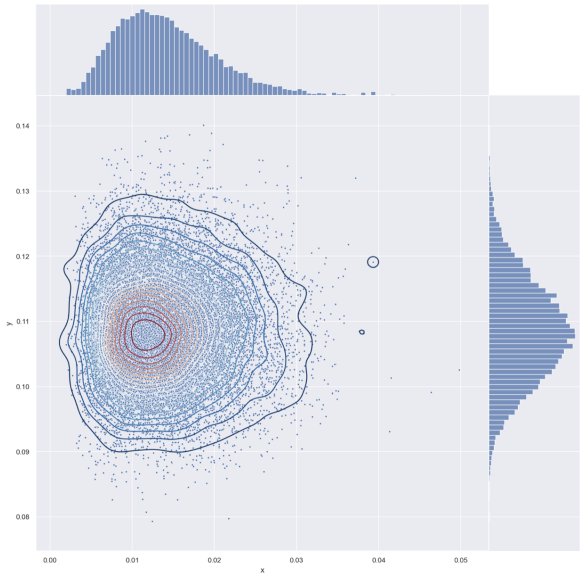

Figure 16: HMC on linear regression data with marginal distributions

Metric	**Train Error**
RMSE	3.0115605945713653
MAE	2.130632451269768
Metric	**Test Error**
RMSE	2.843311886408616
MAE	2.0686645462290465

Table 9: The RMSE and MAE of the train and test sets for HMC Linear regression

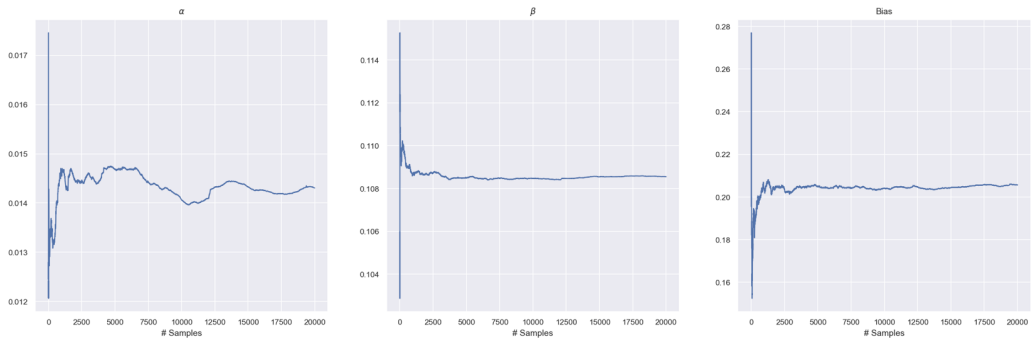

Figure 17: HMC convergence of hyperparameters

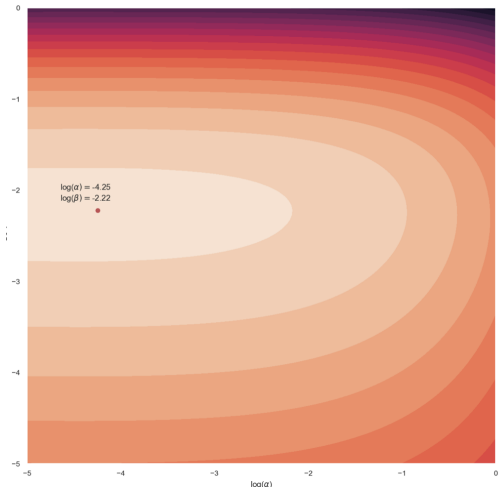

Figure 18: HMC contour plot

5 Gaussian Processes

A Gaussian Process is a collection of random variables, all following a normal distribution. For the rest of the algorithms explained above, the goal was to approximate through the unknown structure of a single posterior by choosing a proposal as an initial guess. Here, we'll try and consider all possible distributions that might represent the posterior and this is done only if the mean and covariance are functions for all the data given [4]. The reason why kernels are used as well. Thus, they can give a more reliable estimate of their own uncertainty [2] and covariance can be expressed in terms of an infinite number of basis functions [1].

5.1 Default Kernel

The hyperparameters of the kernel are optimized during while `GaussianProcessRegressor` is fitted to the data and this is done while maximising the log-marginal-likelihood based on the passed optimiser. The kernel is defined as:

```
ConstantKernel(1.0,constant_value_bounds=
'fixed'*RBF(1.0,length_scale_bounds='fixed')
```

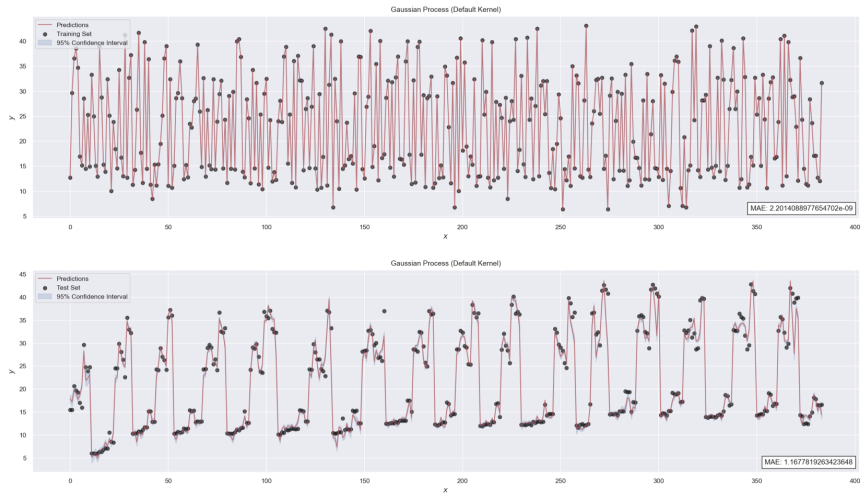

Figure 19: GP with default kernel predictions for training and test set

5.2 Custom RBF Kernel

A slight modification on the parameters of the RBF kernel.

```
kernel = 1.0 * RBF(length_scale=0.5, length_scale_bounds=(1e-3, 1e2))
```

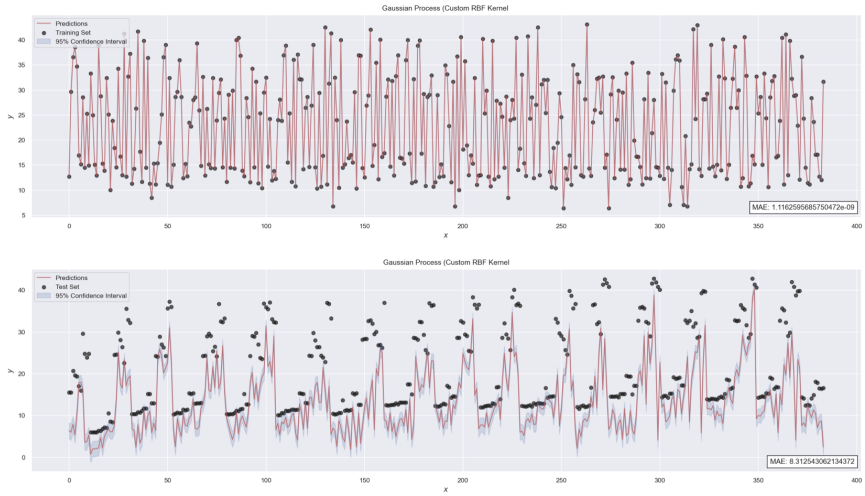

Figure 20: GP with RBF kernel predictions for training and test set

5.3 Custom RBF Kernel + White Noise

Here, we chose to add some noise to the previous RBF kernel. This allows the model to learn the noise level of the data.

```
kernel = 1.0 * RBF(length_scale=0.5, length_scale_bounds=(1e-3,1e2))
   + WhiteKernel(noise_level=1e-4,noise_level_bounds=(1e-22, 1e2))
```

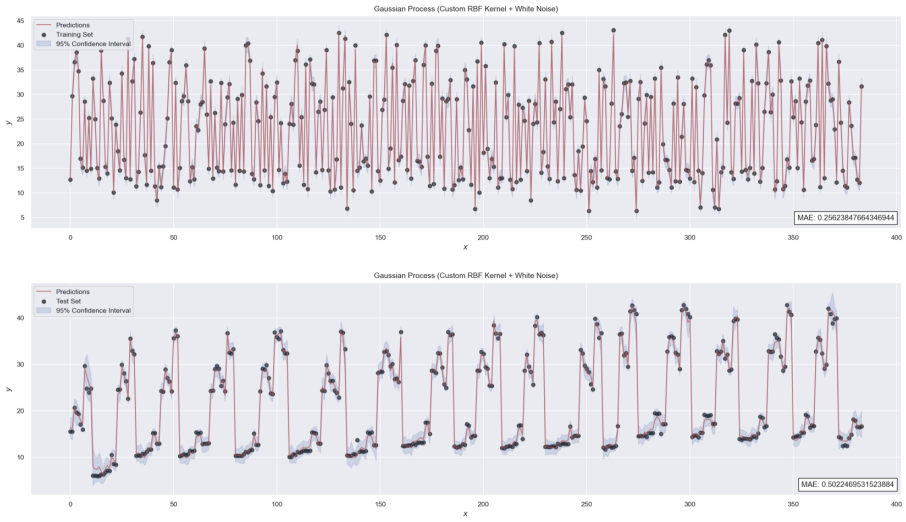

Figure 21: GP with RBF kernel with white noise predictions for training and test set

5.4 Bayesian Neural Network

Bayesian neural networks are alike the rest of the neural networks, but weights are sampled from a probability distribution which is the unknown posterior. Approximating methods, like HMC or Variational inference can be used for sampling from the posterior. We have used `tensorflow-probability` for implementing it. Although the weights that we used were fixed, hence the network performed worse than the previous method in section 5.3.

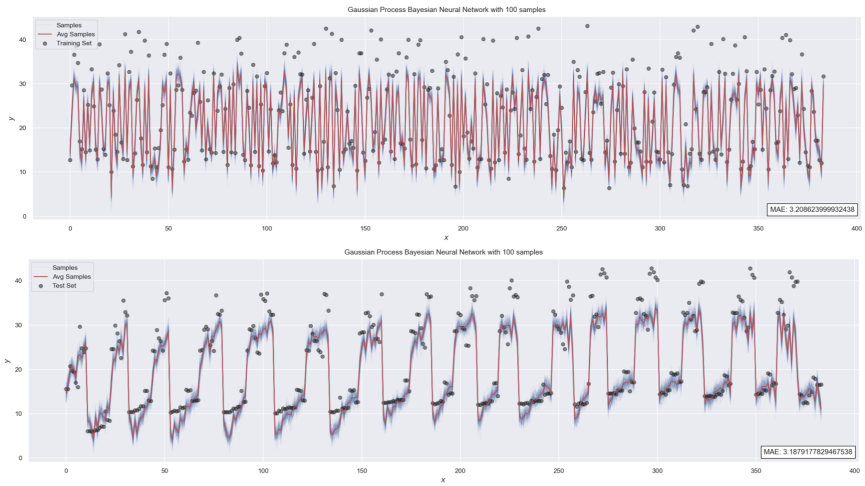

Figure 22: GP with BNN predictions for training and test set

5.5 Comparisons

Here, the metrics for all GP methods are illustrated in the tables below. The initial kernels seemed to overfit slightly as shown in table 10 and 11. To prevent this, adding some noise to the kernel it helps in regularization.

Metric	**Default Kernel**
RMSE	3.826778857177952e-09
MAE	2.2014088977654702e-09
	Custom RBF Kernel
RMSE	1.3779388124263726e-09
MAE	1.1162595685750472e-09
	RBF + White Noise
RMSE	0.32481282810488643
MAE	0.25623847664346944
	BNN
RMSE	13.278580874026702
MAE	3.208623999932438

Table 10: The RMSE and MAE of the training set for GP

Metric	**Default Kernel**
RMSE	1.8900458253349346
MAE	1.1677819263423648
	Custom RBF Kernel
RMSE	10.924232516393255
MAE	8.312543062134372
	RBF + White Noise
RMSE	0.6718571298232691
MAE	0.5022469531523884
	BNN
RMSE	13.458050299098405
MAE	3.1879177829467538

Table 11: The RMSE and MAE of the test set for GP

6 Final Remarks

The reason that metrics for the training and test set were evaluated, is to assess the level of underfitting/overfitting between models. In terms of MAE, the Gaussian process with the RBF + White Noise kernel is the most accurate one. Furthermore, it is illustrated clearly in figure 21 and this due to the ability of customising the kernel giving us the authority to assess what might the unknown structure of the posterior can be.

Although, the Bayesian neural network had similar MAE to the Gaussian process, but had a much higher RMSE which implies that the model is not well optimised and has large outliers.

The second best performance was in HMC, achieving the most optimal and β. Although, there is a slight bias as shown in figure 16 where the x values have a slightly higher skewness than y values. Furthermore, from the posterior plot it is clearly illustrated that the data is non-Gaussian. For the rest of the algorithms performance was similar as in HMC. In practise, stochastic approximation methods like the HMC result in inexact solutions due to the difficulties in obtaining sufficient independent samples in finite time. The main advantage in Variational inference, is that it seeks for solutions that are "somehow" close to the original and this can be dealt in a practical time frame.

It must be noted that the model with the smallest difference between training and test set, for both MAE and RMSE was obtained in OLS. This suggests that it is well generalised with fewer outliers, compared to the rest of the models.

Overall, the best absolute difference achieved for MAE was found in OLS, which is robust to outliers and for the RMSE the Gaussian process which suggests that it had a better spread of the data compared to the rest of the models.

It must be noted that this dataset gave us the authority to also visualise the confidence intervals for each model. The confidence interval for Gaussian process in 21 had a much smaller range implying that predicted values can be more confident.

References

[1] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[2] Oscar Knagg. An intuitive guide to gaussian processes. https://towardsdatascience.com/an-intuitive-guide-to-gaussian-processes-ec2f0b45c71d#:~:text=Gaussian%20processes%20are%20a%20powerful,estimate%20of%20their%20own%20uncertainty., 2019. [Online], [Accessed 13th May].

[3] Serafeim Loukas. How and why to standardize your data:a python tutorial. https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832, 2020. [Online], [Accessed 14th May].

[4] Hilarie Sit. Quick start to gaussian process regression. https://towardsdatascience.com/quick-start-to-gaussian-process-regression-36d838810319, 2019. [Online], [Accessed 13th May].