GAUSSIAN PROCESS MODEL

A Project-II Report

Submitted in partial fulfilment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Aadi Surana EN18CS301002
Abhinav Joshi EN18CS301005
Alfin Abraham EN18CS301024

Under the Guidance of
Dr. Saurabh Das (Professor IIT Indore)
Mr. Ashish Kumawat (Professor Medicaps University)

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331
MAY 2022

GAUSSIAN PROCESS MODEL

A Project-II Report

Submitted in partial fulfilment of requirement of the

Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

Aadi Surana EN18CS301002
Abhinav Joshi EN18CS301005
Alfin Abraham EN18CS301024

Under the Guidance of
Dr. Saurabh Das (Professor IIT Indore)
Mr. Ashish Kumawat (Professor Medicaps University)

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331
MAY 2022

Report Approval

The project work **"Gaussian Process Model"** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner

Name: Mr. Ashish Kumawat

Designation: Professor

Affiliation: Computer Science Engineering, Medicaps University, Indore

External Examiner

Name:

Designation:

Declaration

We hereby declare that the project entitled **"Gaussian Process Model"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of **Mr. Ashish Kumawat, Professor, Department of Computer Science and Engineering,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, we declare that the content of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Aadi Surana

Abhinav Joshi

Alfin Abraham

May 10, 2022

Certificate

I **Ashish Kumawat** certify that the project entitled **"Gaussian Process Model"** submitted in partial fulfilment for the award of the degree of Bachelor of Technology by **Aadi Surana (EN18CS301002), Abhinav Joshi (EN18CS301005), Alfin Abraham (EN18CS301024)** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

Mr. Ashish Kumawat

Department of Computer Science and Engineering

Medi-Caps University, Indore

Dr. Saurabh Das

Department of Astronomy, Astrophysics and Space Engineering

Indian Institute of Technology Indore

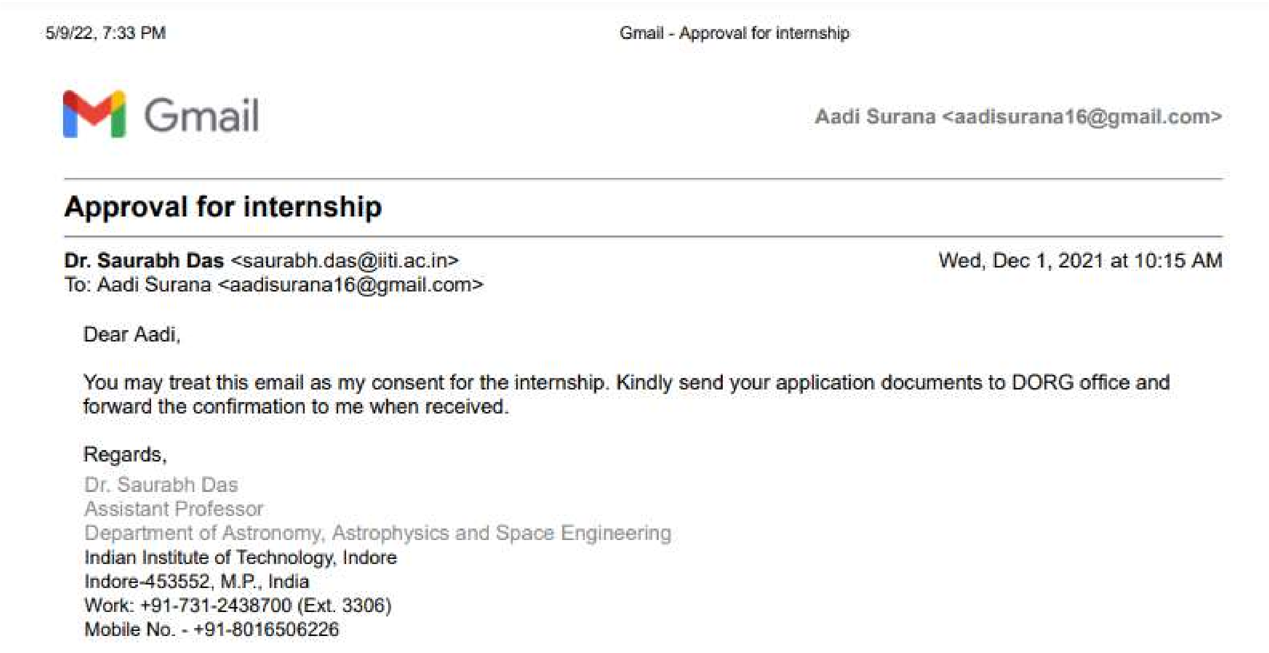
Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Offer Letter of the Project work-II/Internship



5/9/22, 7:33 PM Gmail - Approval for internship

Aadi Surana <aadisurana16@gmail.com>

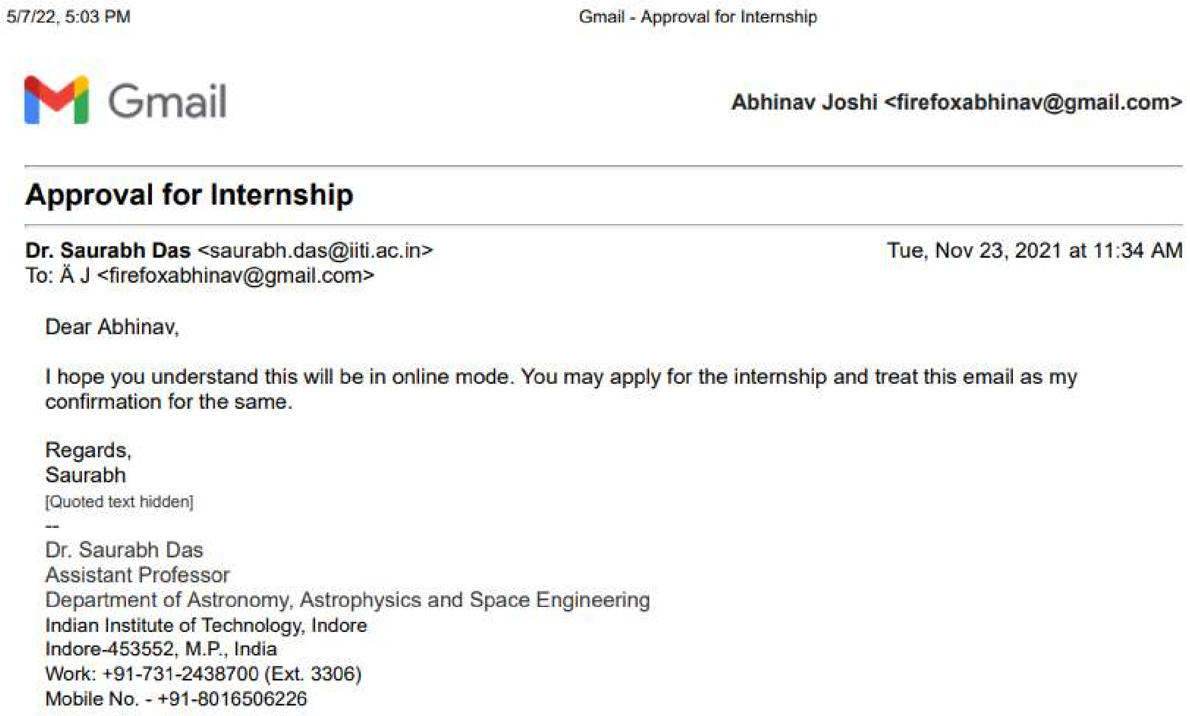
Approval for internship

Dr. Saurabh Das <saurabh.das@iiti.ac.in> Wed, Dec 1, 2021 at 10:15 AM
To: Aadi Surana <aadisurana16@gmail.com>

Dear Aadi,

You may treat this email as my consent for the internship. Kindly send your application documents to DORG office and forward the confirmation to me when received.

Regards,
Dr. Saurabh Das
Assistant Professor
Department of Astronomy, Astrophysics and Space Engineering
Indian Institute of Technology, Indore
Indore-453552, M.P., India
Work: +91-731-2438700 (Ext. 3306)
Mobile No. - +91-8016506226



5/7/22, 5:03 PM Gmail - Approval for Internship

Gmail

Abhinav Joshi <firefoxabhinav@gmail.com>

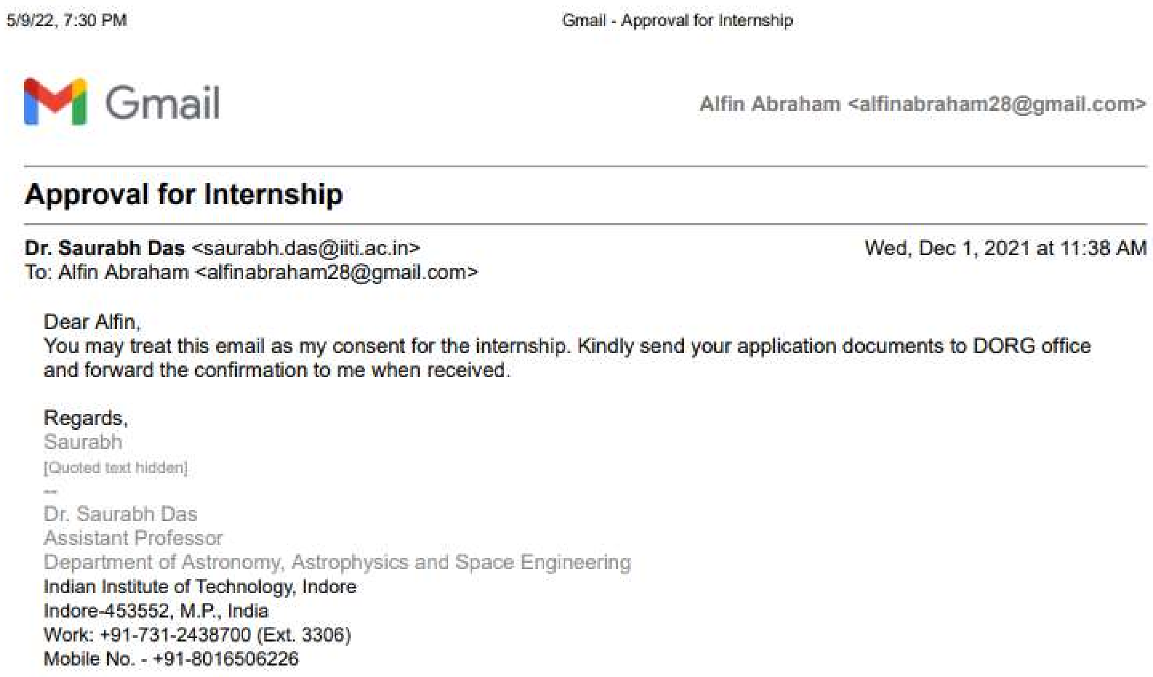
Approval for Internship

Dr. Saurabh Das <saurabh.das@iiti.ac.in> Tue, Nov 23, 2021 at 11:34 AM
To: Ä J <firefoxabhinav@gmail.com>

Dear Abhinav,

I hope you understand this will be in online mode. You may apply for the internship and treat this email as my confirmation for the same.

Regards,
Saurabh
[Quoted text hidden]
--
Dr. Saurabh Das
Assistant Professor
Department of Astronomy, Astrophysics and Space Engineering
Indian Institute of Technology, Indore
Indore-453552, M.P., India
Work: +91-731-2438700 (Ext. 3306)
Mobile No. - +91-8016506226



5/9/22, 7:30 PM Gmail - Approval for Internship

Gmail Alfin Abraham <alfinabraham28@gmail.com>

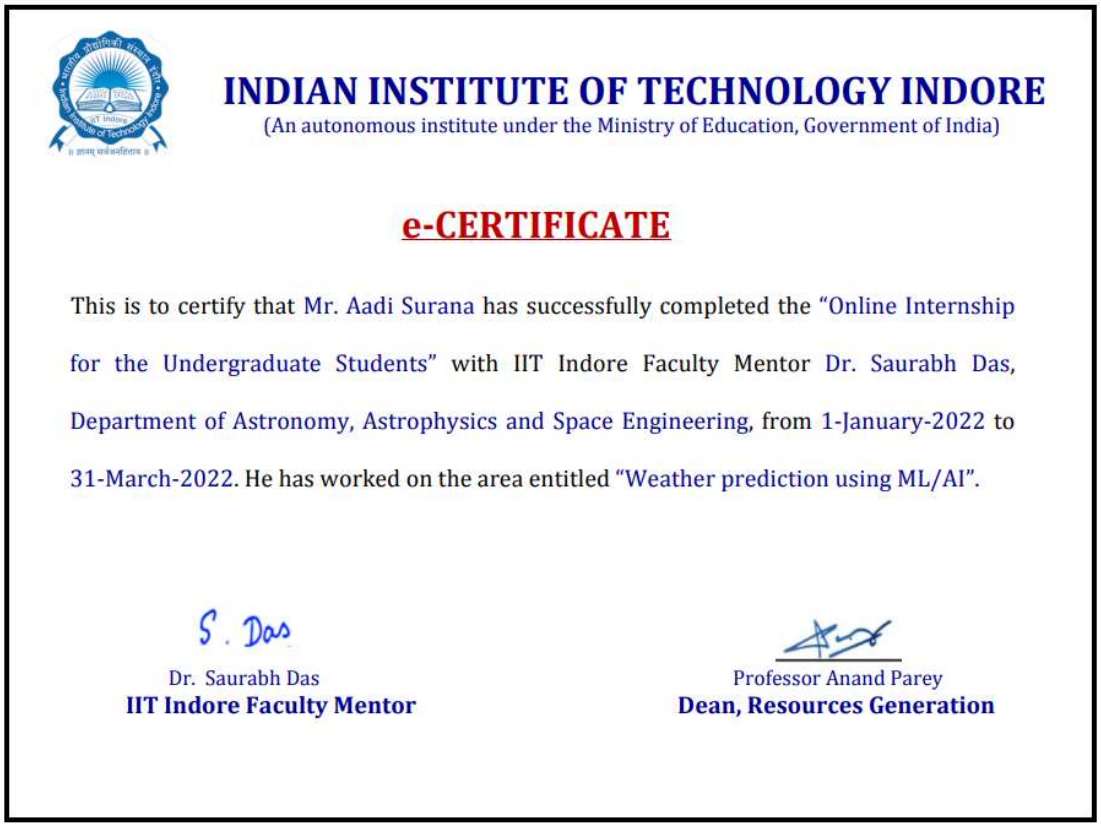
Approval for Internship

Dr. Saurabh Das <saurabh.das@iiti.ac.in> Wed, Dec 1, 2021 at 11:38 AM
To: Alfin Abraham <alfinabraham28@gmail.com>

Dear Alfin,
You may treat this email as my consent for the internship. Kindly send your application documents to DORG office and forward the confirmation to me when received.

Regards,
Saurabh
[Quoted text hidden]
--
Dr. Saurabh Das
Assistant Professor
Department of Astronomy, Astrophysics and Space Engineering
Indian Institute of Technology, Indore
Indore-453552, M.P., India
Work: +91-731-2438700 (Ext. 3306)
Mobile No. - +91-8016506226

Completion certificate/Letter



INDIAN INSTITUTE OF TECHNOLOGY INDORE

(An autonomous institute under the Ministry of Education, Government of India)

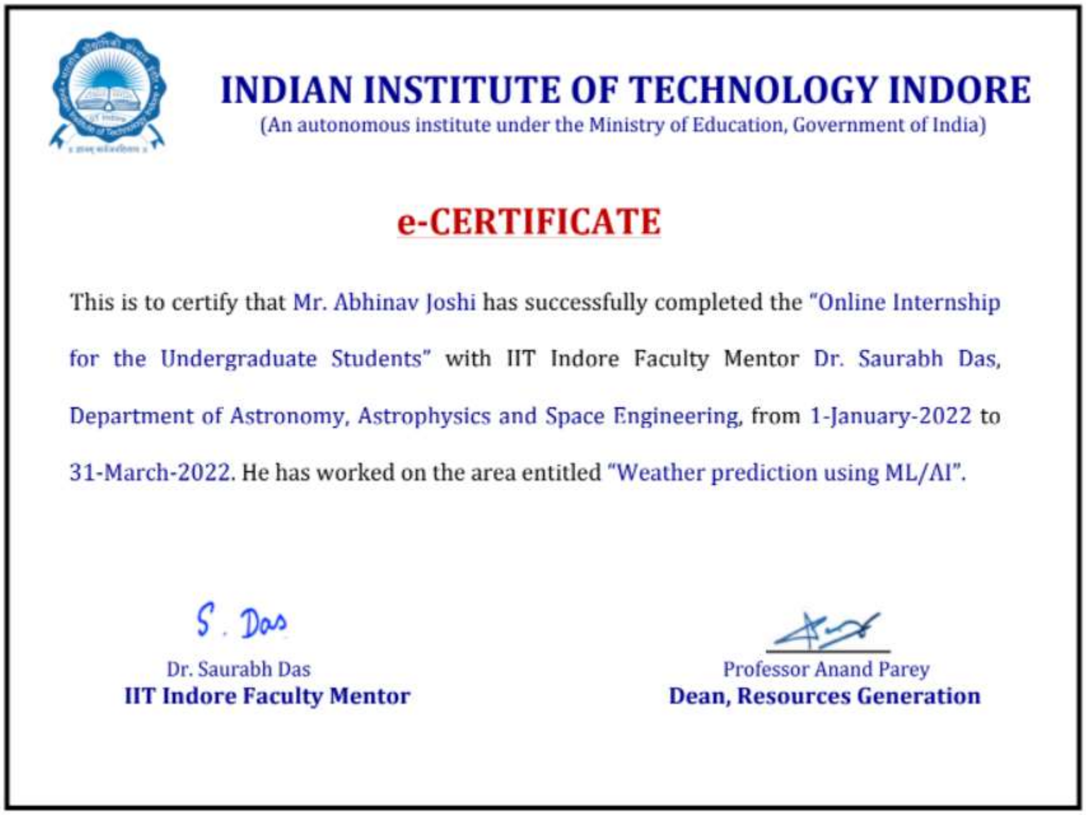
e-CERTIFICATE

This is to certify that Mr. Aadi Surana has successfully completed the "Online Internship for the Undergraduate Students" with IIT Indore Faculty Mentor Dr. Saurabh Das, Department of Astronomy, Astrophysics and Space Engineering, from 1-January-2022 to 31-March-2022. He has worked on the area entitled "Weather prediction using ML/AI".

S. Das

Dr. Saurabh Das
IIT Indore Faculty Mentor

Professor Anand Parey
Dean, Resources Generation



INDIAN INSTITUTE OF TECHNOLOGY INDORE

(An autonomous institute under the Ministry of Education, Government of India)

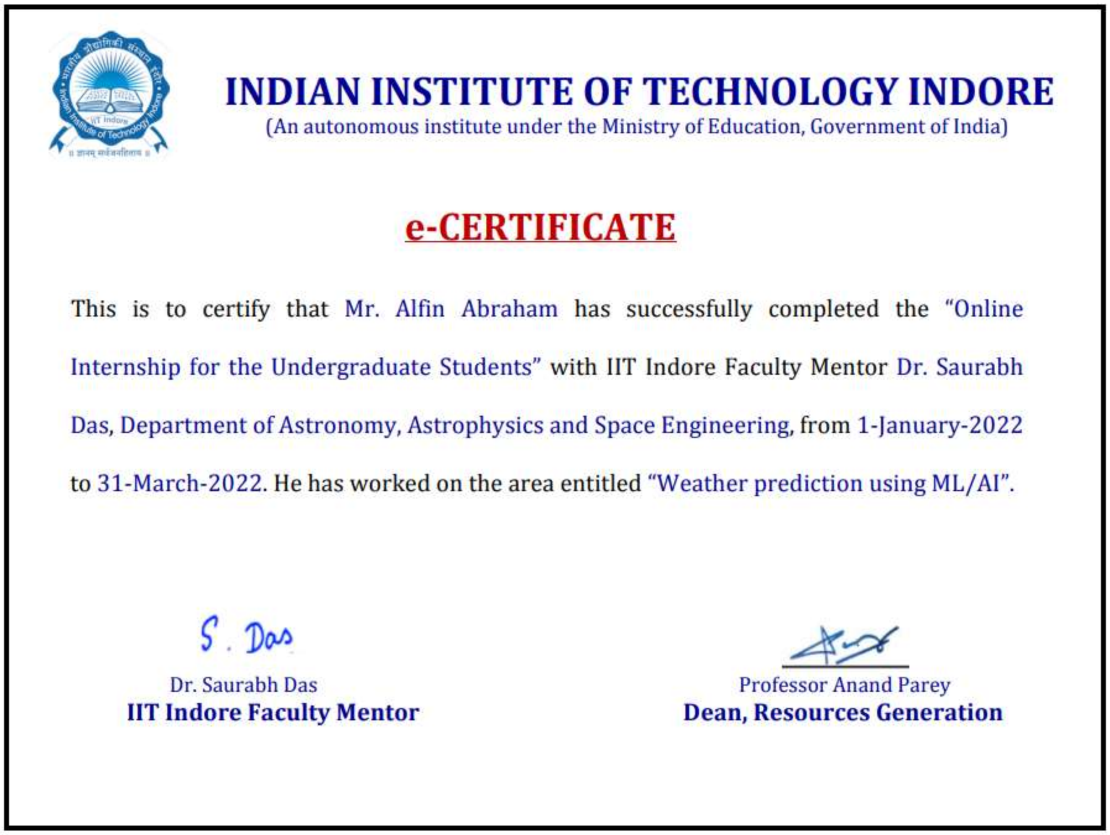
e-CERTIFICATE

This is to certify that Mr. Abhinav Joshi has successfully completed the "Online Internship for the Undergraduate Students" with IIT Indore Faculty Mentor Dr. Saurabh Das, Department of Astronomy, Astrophysics and Space Engineering, from 1-January-2022 to 31-March-2022. He has worked on the area entitled "Weather prediction using ML/AI".

S. Das

Dr. Saurabh Das
IIT Indore Faculty Mentor

Professor Anand Parey
Dean, Resources Generation



INDIAN INSTITUTE OF TECHNOLOGY INDORE

(An autonomous institute under the Ministry of Education, Government of India)

e-CERTIFICATE

This is to certify that Mr. Alfin Abraham has successfully completed the "Online Internship for the Undergraduate Students" with IIT Indore Faculty Mentor Dr. Saurabh Das, Department of Astronomy, Astrophysics and Space Engineering, from 1-January-2022 to 31-March-2022. He has worked on the area entitled "Weather prediction using ML/AI".

S. Das

Dr. Saurabh Das
IIT Indore Faculty Mentor

Professor Anand Parey
Dean, Resources Generation

Acknowledgements

We would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided us with every facility to successfully carry out this project and my profound indebtedness to **Prof. (Dr.) Dileep K Patnayak,** Vice-Chancellor, Medi-Caps University, whose unfailing support and enthusiasm have always boosted our morale. We also thank **Prof. (Dr.) D K Panda,** Pro-Vice-Chancellor and **Dr. Suresh Jain,** Dean Faculty of Engineering, Medi-Caps University, for giving us a chance to work on this project. We would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for the betterment of the project.

We express our heartfelt gratitude to our **External Guide, Dr. Saurabh Das**, Department of Astronomy, Astrophysics and Space Engineering, Indian Institute of Technology Indore as well as to our Internal Guide, Mr. Ashish Kumawat**,** Professor, Department of Computer Science and Engineering, Medi-Caps University, without whose continuous help and support, this project would ever have reached to the completion.

We would also like to thank to my team at IIT Indore who extended their kind support and help towards the completion of this project.

It is their help and support, due to which we became able to complete the design and technical report. Without their support, this report would not have been possible.

Aadi Surana
Abhinav Joshi
Alfin Abraham
B.Tech. IV Year
Department of Computer Science & Engineering
Faculty of Engineering
Medi-Caps University, Indore

Abstract

Social systems produce complex and nonlinear relationships in the indicator variables that describe them. Traditional statistical regression techniques are commonly used in the social sciences to study such systems. These techniques, such as standard linear regression, can prevent the discovery of the complex underlying mechanisms and rely too much on the expertise and prior beliefs of the data analyst. In this thesis, we present two methodologies that are designed to allow the data to inform us about these complex relations and provide us with interpretable models of the dynamics. A first methodology is a Bayesian approach to analyzing the relationship between indicator variables by finding the parametric functions that best describe their interactions. The parametric functions with the highest model evidence are found by fitting a large number of potential models to the data using Bayesian linear regression and comparing their respective model evidence. The methodology is computationally fast due to the use of conjugate priors, and this allows for inference on large sets of models. The second methodology is based on a Gaussian processes framework and is designed to overcome the limitations of the first modelling approach. This approach balances the interpretability of more traditional parametric statistical methods with the predictability and flexibility of non-parametric Gaussian processes

Table of Contents

List of Figures

List of Tables

Abbreviations

The following abbreviations are used:

Abbreviations	Meaning
air_temp	Air temperature
BP	Barometric pressure
DHI	Diffuse horizontal irradiance
DNI	Direct normal irradiance
EU	European Union
GHI	Global horizontal irradiance
IIT	Indian Institute of Technology
NWP	Numerical weather prediction
Per	Periodic kernel
RBF	Radial basis function
RH	Relative humidity
RQ	Rational quadratic kernel
SVR	Support vector machine
UVA	Long-wave ultraviolet radiation
UVB	Short-wave ultraviolet radiation
WD	Wind direction
WD_SD	Standard deviation in wind direction
WS	Wind speed

Notations and Symbols

The following Notation and Symbol are used:

Symbol	Meaning
$y\|x$ and $p(y\|x)$	Conditional random variable y given x and its probability density
$\mathbf{N}(\mu, \Sigma)$ or $\mathbf{N}(x\|\mu, \Sigma)$	(the variable x has a) Gaussian (Normal) distribution with mean vector μ and covariance matrix
$\mathbf{N}(x)$	Short for unit Gaussian $x \sim \mathbf{N}(0, I)$
$\pi(x)$	The sigmoid of the latent value: $\pi(x) = \sigma(f(x))$ (stochastic if $f(x)$ is stochastic
μ	Mean value
σ^2	Standard deviation
l	Length scale
α	Scale mixture

Chapter 1

1. Introduction

The Gaussian processes model is a probabilistic supervised machine learning framework that has been widely used for regression and classification tasks. A Gaussian processes regression (GPR) model can make predictions incorporating prior knowledge (kernels) and provide uncertainty measures over predictions. Gaussian processes model is a supervised learning method developed by computer science and statistics communities. Researchers with engineering backgrounds often find it difficult to gain a clear understanding of it. To understand GPR, even only the basics needs to have knowledge of multivariate normal distribution, kernels, non-parametric model, and joint and conditional probability.

In the Gaussian process modelling approach, one computes predictive distributions whose means serve as output estimates. Gaussian processes (GPs) for regression have historically been first introduced by O'Hagen but started being a popular non-parametric modelling approach after the publication. It is shown that GPs can achieve a predictive performance comparable to (if not better than) other modelling approaches like neural networks or local learning methods.

1.1 Introduction to Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics.

1.1.1 How machine learning works:

1. **A Decision Process:** In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithm will produce an estimate of a pattern in the data.

2. **An Error Function:** An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.

3. **A Model Optimization Process:** If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluation and optimize the process, updating weights autonomously until a threshold of accuracy has been met.

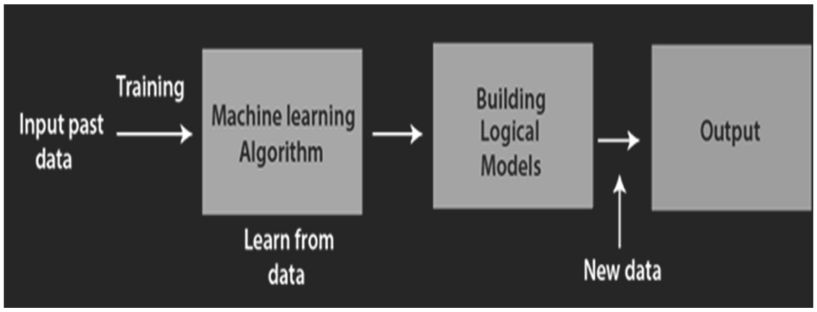


Figure 1.1 Machine Learning Flow

Example:

When you’re building a movie recommender, your algorithm’s decision process might look at how similar a given movie is to other movies you’ve watched and come up with a weighting system for different features.

During the training process, the algorithm goes through the movies you have watched and weights different properties. *Is it a sci-fi movie? Is it funny?* The algorithm then tests out whether it ends up recommending movies that you (or people like you) actually watched. If it gets it right, the weights used, stay the same, if it gets a movie wrong, the weights that led to the wrong decision get turned down so it doesn’t make that kind of mistake again.

Since a machine learning algorithm updates autonomously, the analytical accuracy improves with each run as it teaches itself from the data it analyzes. This iterative nature of learning is both unique and valuable because it occurs without human intervention — providing the ability to uncover hidden insights without being specifically programmed to do so.

1.1.2 Real-world machine learning use cases:

1. **Speech recognition:** It is also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, and it is a capability that uses natural language processing (NLP) to process human speech into a written format. For example, Google Assistant and Apple Siri.

2. **Customer service:** Online chatbots are replacing human agents along the customer journey. They answer frequently asked questions (FAQs) around topics, like shipping, or provide personalized advice, cross-selling products or suggesting sizes for users, changing the way we think about customer engagement across websites and social media platforms. For example, the Zomato virtual assistant, Zia, on the Zomato food delivery service's mobile application assists you in tracking your food order and addresses any issues with food quality, quantity or delivery related issues.

3. **Automated stock trading:** Designed to optimize stock portfolios, AI-driven high-frequency trading platforms make thousands or even millions of trades per day without human intervention.

4. **Computer vision:** This AI technology enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and based on those inputs, it can take action. This ability to provide recommendations distinguishes it from image recognition tasks. Powered by convolutional neural networks, computer vision has applications in photo tagging in social media, radiology imaging in healthcare, and self-driving cars within the automotive industry.

1.1.3 Types of Machine Learning Methods

Many machine learning models are defined by the presence or absence of human influence on raw data — whether a reward is offered, specific feedback is given or labels are used. Following are the major machine learning methods:

1. **Supervised learning:** The dataset being used has been pre-labelled and classified by users to allow the algorithm to see how accurate its performance is.

2. **Unsupervised learning:** The raw dataset being used is unlabeled and an algorithm identifies patterns and relationships within the data without help from users.

3. **Semi-supervised learning:** The dataset contains structured and unstructured data, which guide the algorithm on its way to making independent conclusions. The combination of the two data types in one training dataset allows machine learning algorithms to learn to label unlabeled data.

4. **Reinforcement learning:** The dataset uses a "rewards/punishments" system, offering feedback to the algorithm to learn from its own experiences by trial and error.

1.1.4 Supervised Machine Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing sample data to check whether it is predicting the exact output or not.Some very practical applications of supervised learning algorithms in real life, include: Face Detection, Signature recognition, Spam detection and Weather forecasting.

The working of Supervised learning can be easily understood by the below example and diagram:

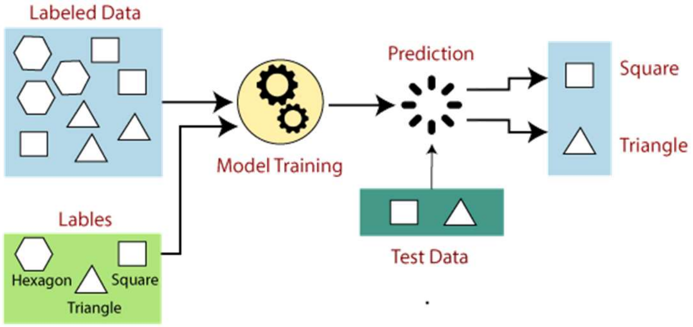


Figure 1.2 Supervised Learning

Types of supervised machine learning algorithms:

1. **Classification Models:** Classification models are used for problems where the output variable can be categorized, such as “Yes” or “No”, or “Pass” or “Fail.” Classification Models are used to predict the category of the data.

2. **Regression Models:** Regression models are used for problems where the output variable is a real value such as a unique number, dollars, salary, weight or pressure, for example. It is most often used to predict numerical values based on previous data observations.

1.2 Literature Review

1.2.1 Regression

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as temperature, age, salary, price, etc.

1. **Linear Regression:** Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. A linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

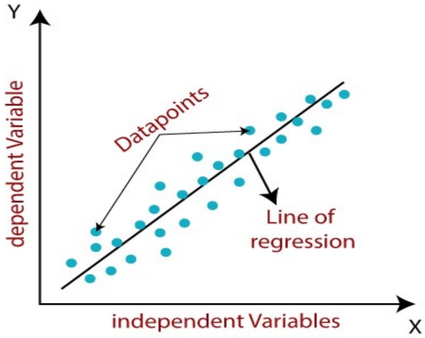


Figure 1.3 Linear Regression

2. **Logistic Regression:** Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either "Yes" or "No", "0" or "1" or a boolean value. But, instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. It is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic

function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something. For example, whether the cells are cancerous or not, whether a mouse is obese or not based on its weight and diabetes prediction.

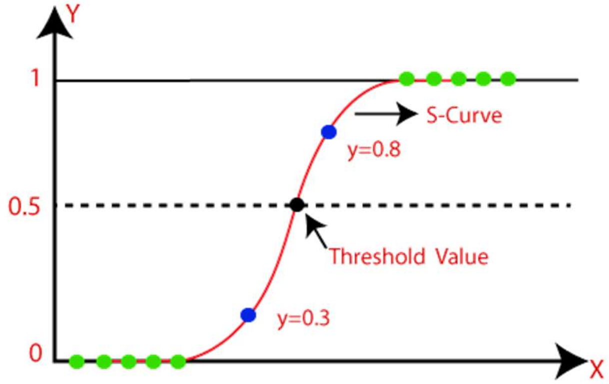


Figure 1.4 Logistic Regression

3. **Bayesian Regression:** It is a very powerful method because they provide us with the entire distribution over regression parameters. In order to calculate inadequate data or unequal distributed data, Bayesian Linear Regression provides a natural mechanism. We use probability distribution instead of point estimates to devise linear regression.

Posterior= (Likelihood*Prior)/ Normalization

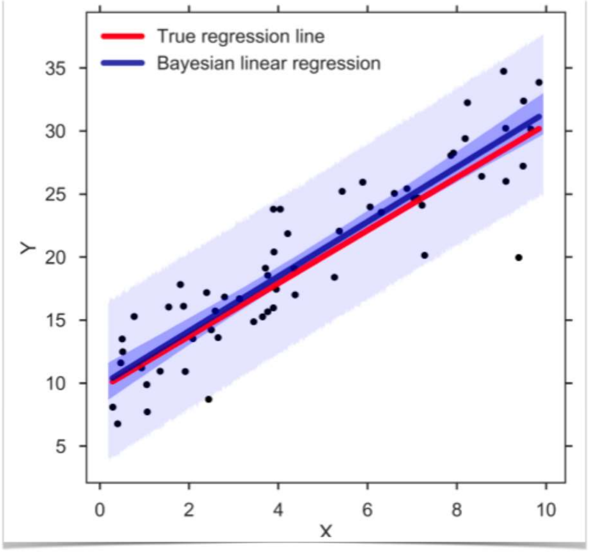


Figure 1.5 Bayesian Regression

1.2.2 Random Variable

A random variable (also known as a stochastic variable) is a real-valued function, whose domain is the entire sample space of an experiment.

Think of the domain as the set of all possible values that can go into a function. A function takes the domain/input, processes it, and renders an output/range. Similarly, a random variable takes its domain (sample space of an experiment), processes it, and assigns every event/outcome a real value. This set of real values obtained from the random variable is called its range.

Types of random variables:

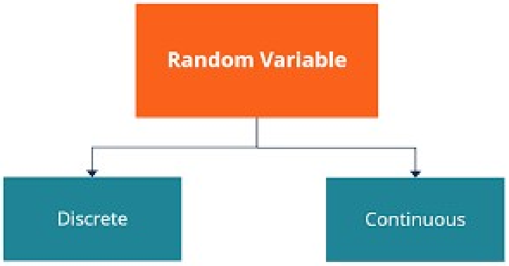


Figure 1.6 Random Variable

1. **Discrete Random Variables:** A discrete random variable is one that may take on only a countable number of distinct values such as 0,1,2,3,4,........ Discrete random variables are usually (but not necessarily) counts. If a random variable can take only a finite number of distinct values, then it must be discrete. Examples of discrete random variables include the number of children in a family, the Friday night attendance at a cinema, the number of patients in a doctor's surgery, and the number of defective light bulbs in a box of ten bulbs. The probability distribution of a discrete random variable is a list of probabilities associated with each of its possible values. It is also sometimes called the probability function or the probability mass function.

2. **Continuous Random Variables**: A continuous random variable is one that takes an infinite number of possible values. Continuous random variables are usually measurements. Examples include height, weight, the amount of sugar in orange, and the time required to run a mile. A continuous random variable is not defined at specific values. Instead, it is

defined over an interval of values and is represented by the area under a curve (in advanced mathematics, this is known as an integral). The probability of observing any single value is equal to 0 since the number of values that may be assumed by the random variable is infinite.

1.2.3 PDF (Probability Density Function):

PDF is a statistical term that describes the probability distribution of the continuous random variable. PDF most commonly follows the Gaussian Distribution. If the features / random variables are Gaussian distributed then PDF also follows Gaussian Distribution. On a PDF graph, the probability of a single outcome is always zero, this happens because the single point represents the line which doesn't cover the area under the curve.

1.2.4 PMF (Probability Mass Function):

PMF is a statistical term that describes the probability distribution of the discrete random variable. The PDF is applicable for continuous random variables while PMF is applicable for discrete random variables For e.g, Throwing a dice (You can only select 1 to 6 numbers (countable))

1.2.5 CDF (Cumulative Distribution Function):

PMF is a way to describe distribution but it is only applicable for discrete random variables and not for continuous random variables. The **cumulative distribution function** is applicable for describing the distribution of random variables whether it is continuous or discrete

1.3 Objectives

Gaussian process is a flexible class of non-parametric machine learning A common application of GPs is regression. For example, given incomplete geographical weather data, such as temperature or humidity, how can one recover values at unobserved locations? If one has good reason to believe the data is normally distributed, then using a GP model could be a judicious choice.

The machinery of probabilistic inference brings to the field of time-series analysis and monitoring robust, stable, computationally practical and principled approaches that naturally accommodate these real-world challenges.

1.4 Significance

Gaussian process regression (GPR) is a nonparametric, Bayesian approach to regression that is making waves in the area of machine learning. GPR has several benefits, including working well on small datasets and having the ability to provide uncertainty measurements on the predictions.Gaussian process regression is nonparametric (*i.e.* not limited by a functional form), so rather than calculating the probability distribution of parameters of a specific function, GPR calculates the probability distribution over all admissible functions that fit the data.

1.5 Research Diagram

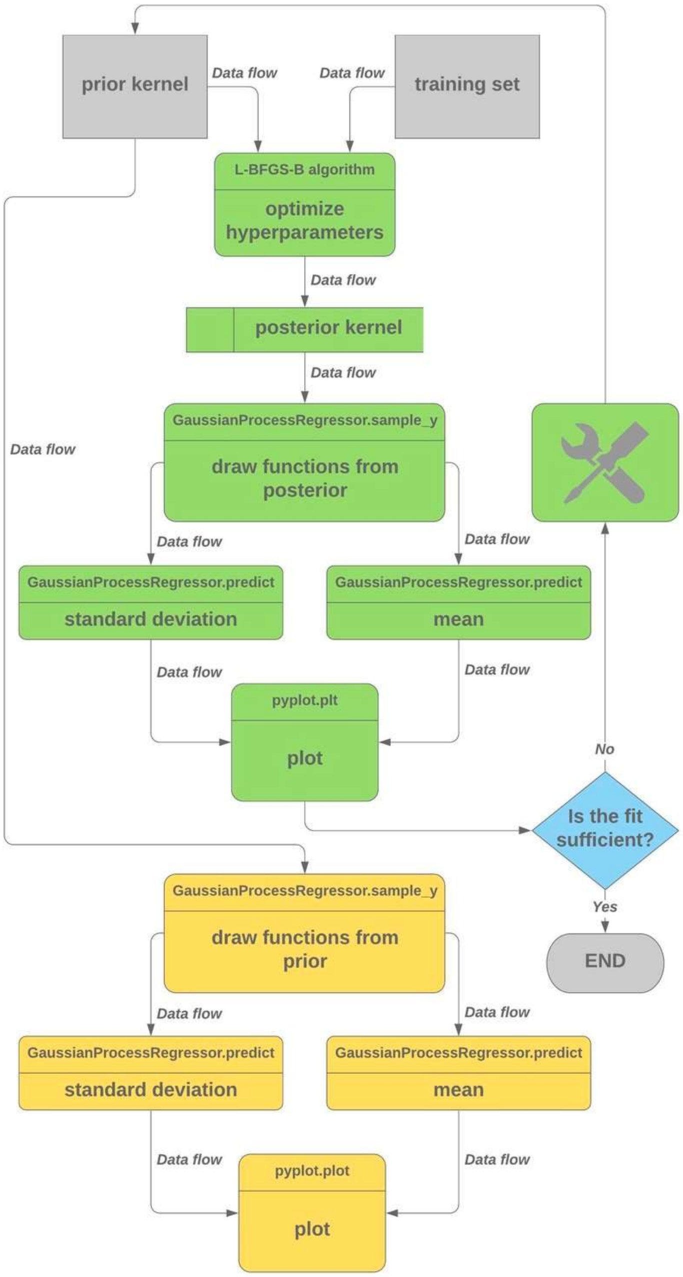


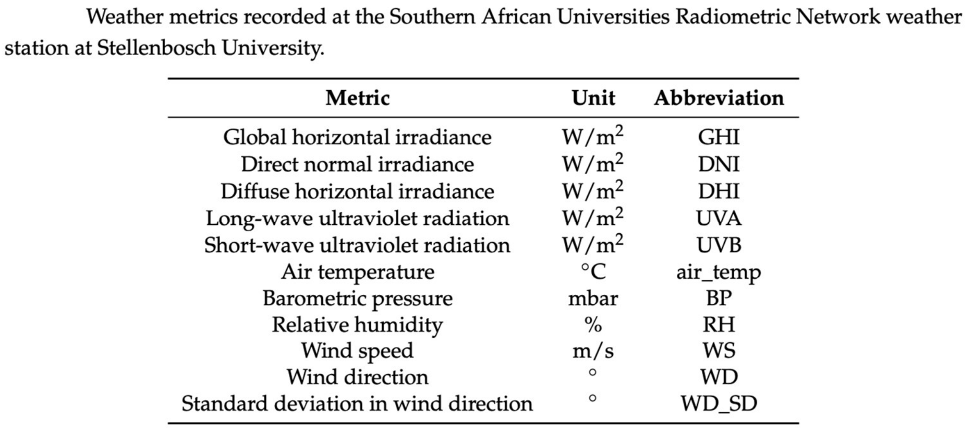
Figure 1.7 Gaussian Process Flow Chart

1.6 Source Of Data

Weather data for the week 1 February 2015 to 8 February 2015 was acquired from the Stellenbosch weather station of the Southern African Universities Radiometric Network. To illustrate the effect of sampling intervals on data characteristics, wind speed data from the Stellenbosch weather station was sampled at different intervals, ranging from 1 h to 12 h.

Gaussian Process Regression on GHI Data

The Gaussian process algorithm will use metrics to construct a covariance matrix that represents the degree of correlation between the various weather metrics. The following metrics are expected to be strongly correlated: GHI, DNI, DHI, UVA, UVB and air temperature. The other metrics may exhibit weak correlation—whether this is the case will be determined by the Gaussian process regression algorithm. The stochastic behaviour of the parameters calls for Gaussian process regression.



Weather metrics recorded at the Southern African Universities Radiometric Network weather station at Stellenbosch University.

Metric	Unit	Abbreviation
Global horizontal irradiance	W/m^2	GHI
Direct normal irradiance	W/m^2	DNI
Diffuse horizontal irradiance	W/m^2	DHI
Long-wave ultraviolet radiation	W/m^2	UVA
Short-wave ultraviolet radiation	W/m^2	UVB
Air temperature	°C	air_temp
Barometric pressure	mbar	BP
Relative humidity	%	RH
Wind speed	m/s	WS
Wind direction	°	WD
Standard deviation in wind direction	°	WD_SD

Table 1.1 Weather metrics

The weather data for the week 1 February 2015 to 8 February 2015 was used to train a Gaussian process regression algorithm. A standard Gaussian process regression was employed, with a multi-in-single-out structure. This means that the Gaussian process regressor was trained on a multi-dimensional array of input data, but that interpolation and prediction were only done for a single output, namely GHI.

Chapter -2

2.1 Experimental Setup

2.1.1 Normal Distribution:

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. In graph form, the normal distribution will appear as a bell curve.

The Normal Distribution has:

1. mean = median = mode
2. symmetry about the centre
3. 50% of values less than the mean and 50% greater than the mean

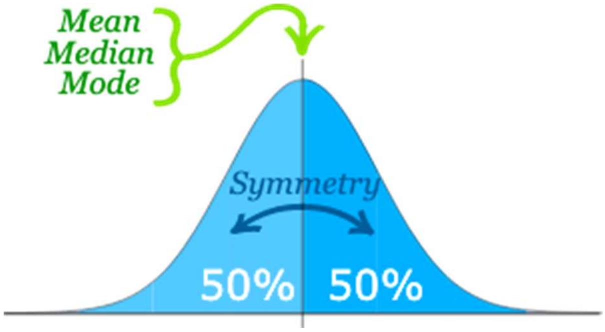


Figure 2.1 Normal Distribution Symmetry

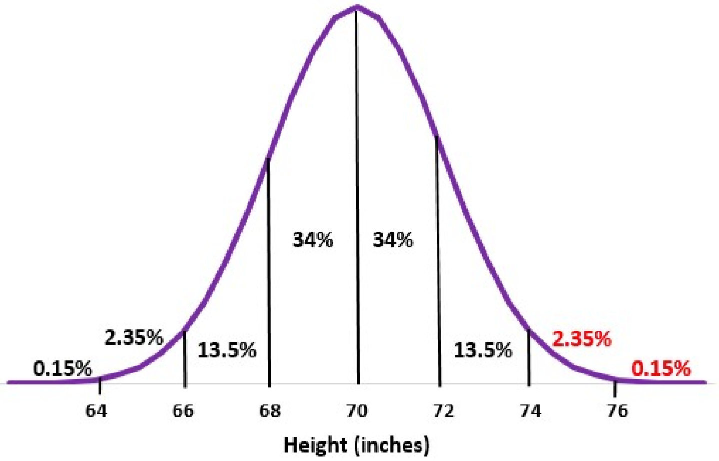


Figure 2.2 Normal Distribution Regions

2.1.2 Univariate Distribution:

The normal distribution, also known as Gaussian distribution, is defined by two parameters, mean μ, which is expected value of the distribution and standard deviation σ which corresponds to the expected squared deviation from the mean. Mean, μ controls the Gaussian's centre position and the standard deviation controls the shape of the distribution. The square of standard deviation is typically referred to as the variance. We denote this distribution as $N(\mu, \sigma^2)$.

Given the mean and variance, one can calculate the probability distribution function of a normal distribution with a normalised Gaussian function for a value x, the density is:

$$P(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

We call this distribution univariate because it consists of one random variable. of one random variable.

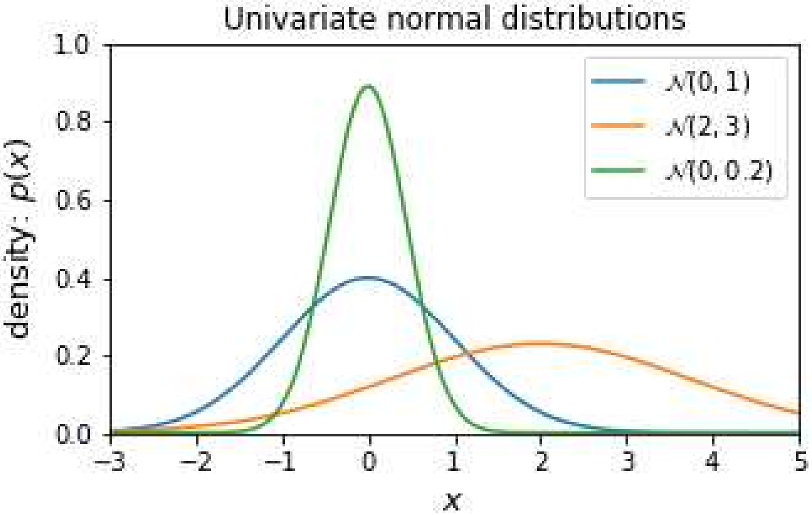


Figure 2.3 Univariate Distribution

2.1.3 Multivariate Distribution:

The multivariate normal distribution is a multidimensional generalisation of the one dimensional normal distribution. It represents the distribution of a multivariate random variable, that is made up of multiple random variables which can be correlated with each other.

Like the univariate normal distribution, the multivariate normal is defined by sets of parameters: the mean vector μ. which is the expected value of the distribution and the variance-covariance matrix Σ, which measures how two random variables depend on each other and how they change together.

We denote the covariance between variables X and Y as *Cov(X,Y).*

The multivariate normal with dimensionality d has a joint probability density given by:

$$P(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{\sqrt{2(\pi)^d|\Sigma|}} exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

where x is a random vector of size d, μ is d×1 mean vector and Σ is the (symmetric and positive definite) covariance matrix of size d×d and |Σ| is the determinant. We denote this multivariate normal distribution as N(μ,Σ).

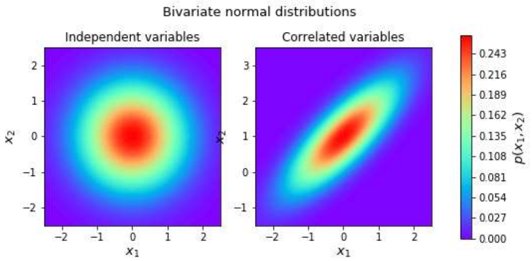


Figure 2.4 Multivariate Distribution

2.1.4 Kernels:

A kernel (or covariance function) describes the covariance of the Gaussian process random variables. Together with the mean function the kernel completely defines a Gaussian process.

Types of Kernels

1. **Exponentiated quadratic kernel:**

   The exponentiated quadratic kernel (also known as the squared exponential kernel, Gaussian kernel or radial basis function kernel) is one of the most popular kernels used in Gaussian process modeling. It can be computed as

$$k(x_a, x_b) = \sigma^2 \exp\left(-\frac{||x_a - x_b||^2}{2\ell^2}\right)$$

2. **Rational quadratic kernel:**

   Similar to the exponentiated quadratic, the rational quadratic kernel will result in a somewhat smooth prior on functions sampled from the Gaussian process. The rational quadratic can be interpreted as an infinite sum of different exponentiated quadratic kernels with different length scales with α determining the weighting between different length scales. When α→∞ the rational quadratic kernel converges into the exponentiated quadratic kernel.

$$k(x_a, x_b) = \sigma^2 \left(1 + \frac{||x_a - x_b||^2}{2\alpha\ell^2}\right)^{-\alpha}$$

3. **Periodic kernel:**

   The periodic kernel allows one to model functions which repeat themselves exactly. Its parameters are easily interpretable: The period p simply determines the distance between repetitions of the function.

$$k(x_a, x_b) = \sigma^2 \exp\left(-\frac{2}{\ell^2}\sin^2\left(\pi\frac{|x_a - x_b|}{p}\right)\right)$$

2.2 Procedure Adopted

Gaussian process regression (GPR) is a nonparametric, Bayesian approach to regression that is making waves in the area of machine learning. GPR has several benefits, including working well on small datasets and having the ability to provide uncertainty measurements on the predictions.

Gaussian process regression is nonparametric (i.e. not limited by a functional form), so rather than calculating the probability distribution of parameters of a specific function, GPR calculates the probability distribution over all admissible functions that fit the data. However, similar to the above, we specify a prior (on the function space), calculate the posterior using training data, and compute the predictive posterior distribution on our points of interest

2.2.1 Combining kernels by multiplication

Kernels can be combined by multiplying them together. Multiplying kernels is an elementwise multiplication of their corresponding covariance matrices. This means that the covariances of the two multiplied kernels will only have a high value if both covariances have a high value. The multiply operation can thus be interpreted as an AND operation.

2.2.2 Combining kernels by addition

Kernels can be combined by adding them together. Adding kernels is an elementwise addition of their corresponding covariance matrices. This means that the covariances of the two added kernels will only have a low value if both of the covariances have a low value. The addition operation can thus be interpreted as an OR operation.

Chapter 3

3.1 Implementation

3.1.1 Gaussian Process Model

In probability theory and statistics, a Gaussian process is a stochastic process (a collection of random variables indexed by time or space), such that every finite collection of those random variables has a multivariate normal distribution, i.e. every finite linear combination of them is normally distributed. The distribution of a Gaussian process is the joint distribution of all those (infinitely many) random variables, and as such, it is a distribution over functions with a continuous domain, e.g. time or space.

Gaussian processes are useful in statistical modelling, benefiting from properties inherited from the normal distribution. For example, if a random process is modelled as a Gaussian process, the distributions of various derived quantities can be obtained explicitly.

Such quantities include the average value of the process over a range of times and the error in estimating the average using sample values at a small set of times. While exact models often scale poorly as the amount of data increases, multiple approximation methods have been developed which often retain good accuracy while drastically reducing computation time.

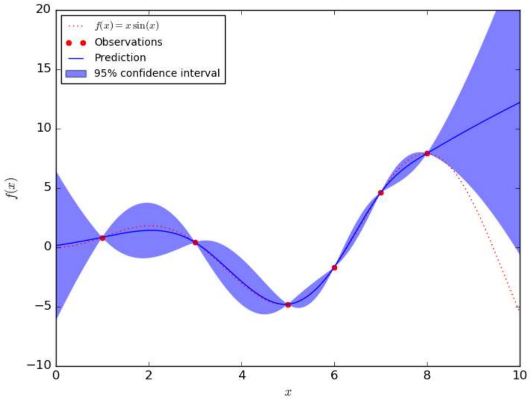


Figure 3.1 Gaussian Process Model

A compound kernel is created. The choice of the kernel was arrived at after testing different kernel configurations.

In order to test the ability of the algorithm to bridge gaps in the data, meter failure was simulated. The resulting training data set consisted of 137 hourly data points between $t0 = 0$ hours and $tend =$ 177 hours. Each data point was thirteen-dimensional, [1 × 13], since it contained, at each time step, readings for GHI, DNI, DHI, DHI_shadowband, UVA, UVB, air_temp, RH, WS, WD, WD_SD and BP

The resulting Gaussian process model was used to interpolate global horizontal irradiance (GHI), as this parameter is generally used to calculate the available solar power at a given site on the earth’s surface.

Chapter -4

Result and Discussion

The interpolation of GHI-data after training the multi-in-single-out Gaussian process regression model on the one-hourly averaged weather data set (with no meter failure), using a four kernel. The inlay shows how the predictions align with measured values. The root-mean-squared error for the interpolation with meter failure, when compared to the measured GHI-values, was found to be

Kernels	**GHI root means Square Error**
Constant Kernel*Rational Quadratic Kernel	291.02 W/m^2
Constant Kernel *Exponent Sine Squared	294.59 W/m^2

Table 4.1 RMS Error

The process of interpolation of GHI data through Gaussian process regression is not trivial and clear answers on the reasons for certain kernels performing better than others are not apparent.

It was found that a compound kernel consisting of a periodic kernel component and a rational quadratic kernel component provided the best interpolation and prediction results for solar radiation data.

Chapter -5

Conclusion and Discussion

This project aimed to illustrate the potential of Gaussian process regression for the interpolation and forecasting of GHI data. Reasonably good interpolation and prediction of solar radiation data were achieved by employing a multi-in-single-out Gaussian process regression with different kernel to a one-hourly averaged weather data set. The effect of sampling interval on the effectiveness of the Gaussian process regression model to capture the characteristics of a weather data set, was also briefly investigated. In this regard, it can be concluded that the sampling interval has a significant effect on the ability of the Gaussian process regression model to accurately capture the structure of a wind speed data set, as measured by the Weibull parameters. This gives rise to the question of whether a Gaussian process regression model of GHI data will be similarly affected by interval deficiency.The reasonably successful forecasting of one-hourly averaged solar radiation data using Gaussian process regression points to the possibility of effectively integrating multi-in-single-out Gaussian process regression into a renewable energy management system. It is recommended that future work focuses on finding a rules-based method for the construction of kernels for GHI data modeling, as well as applying Gaussian process regression to larger datasets. The impact of sampling interval in Gaussian process regression models of GHI data could also make for interesting future research.

Chapter-6

Future Scope

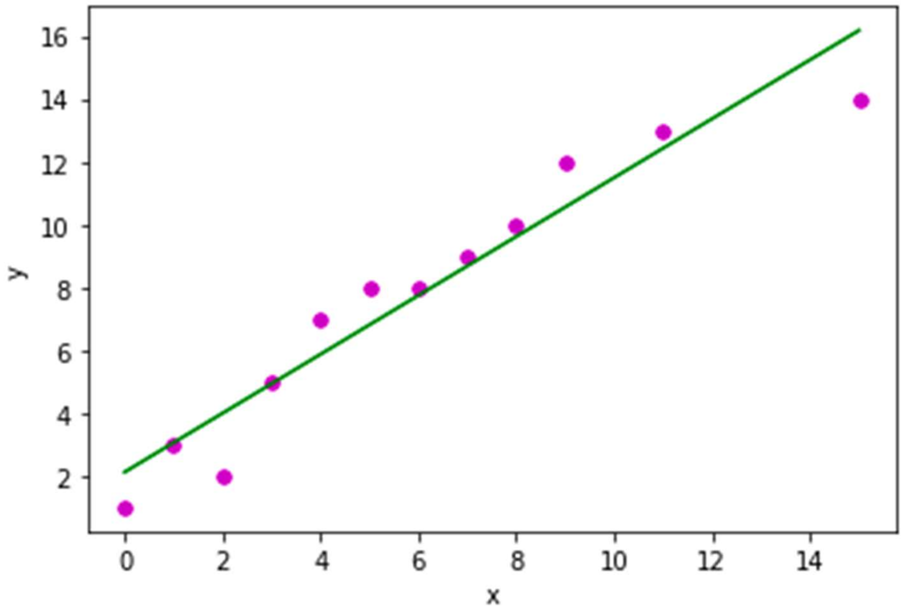
The variability of renewable energy resources, such as solar and wind, poses a challenge to the stability of the electricity grid.These results, achieved in modeling solar radiation data using Gaussian process regression, could open new avenues in the development of probabilistic renewable energy management systems. Such systems could aid smart grid operators and support energy trading platforms, by allowing for better-informed decisions that incorporate the inherent uncertainty of stochastic power systems.Future work will focus on some of the best-performing kernels and assess their performance using larger GHI datasets.

Appendix

I.I Linear Regression:

```
import numpy as np

import matplotlib.pyplot as plt

 def estimate_coef(x, y):

   # number of observations/points

   n = np.size(x)

    # mean of x and y vector

   m_x, m_y = np.mean(x), np.mean(y)

    # calculating cross-deviation and deviation about x

   SS_xy = np.sum(y*x) - n*m_y*m_x

   SS_xx = np.sum(x*x) - n*m_x*m_x

    # calculating regression coefficients

   b_1 = SS_xy / SS_xx      #m

   b_0 = m_y - b_1*m_x      #c

    return(b_0, b_1)

 def plot_regression_line(x, y, b):

   # plotting the actual points as scatter plot

   plt.scatter(x, y, color = "m",

              marker = "o", s = 30)
    # predicted response vector

   y_pred = b[0] + b[1]*x     #y=c+mx

    # plotting the regression line

   plt.plot(x, y_pred, color = "g")

    # putting labels

   plt.xlabel('x')

   plt.ylabel('y')

    # function to show plot

   plt.show()

 def main():

   # observations

   x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9,11,15])

   y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12,13,14])

    # estimating coefficients

   b = estimate_coef(x, y)

   print("Estimated coefficients:\nb_0 = {}  \ \nb_1 = {}".format(b[0], b[1]))

    # plotting regression line

   plot_regression_line(x, y, b)



main()
```

Output:

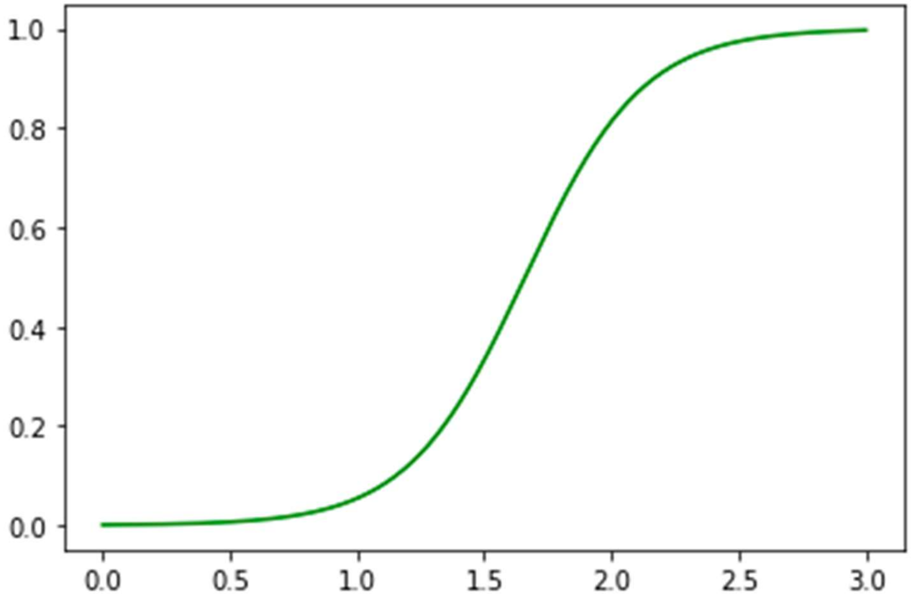


I.II Logistic Regression:

```
# Train a logistic regression classifier to predict whether a flower is iris
virginica or not

from sklearn import datasets

from sklearn.linear_model import LogisticRegression

import numpy as np

import matplotlib.pyplot as plt

iris = datasets.load_iris()

X = iris["data"][:, 3:]

y = (iris["target"] == 2).astype(np.int)

# Train a logistic regression classifier

clf = LogisticRegression()

clf.fit(X,y)
```

```
# Using matplotlib to plot the visualisation

X_new = np.linspace(0,3,1000).reshape(-1,1)

y_prob = clf.predict_proba(X_new)

plt.plot(X_new, y_prob[:,1], "g-", label="virginica")

plt.show()
```

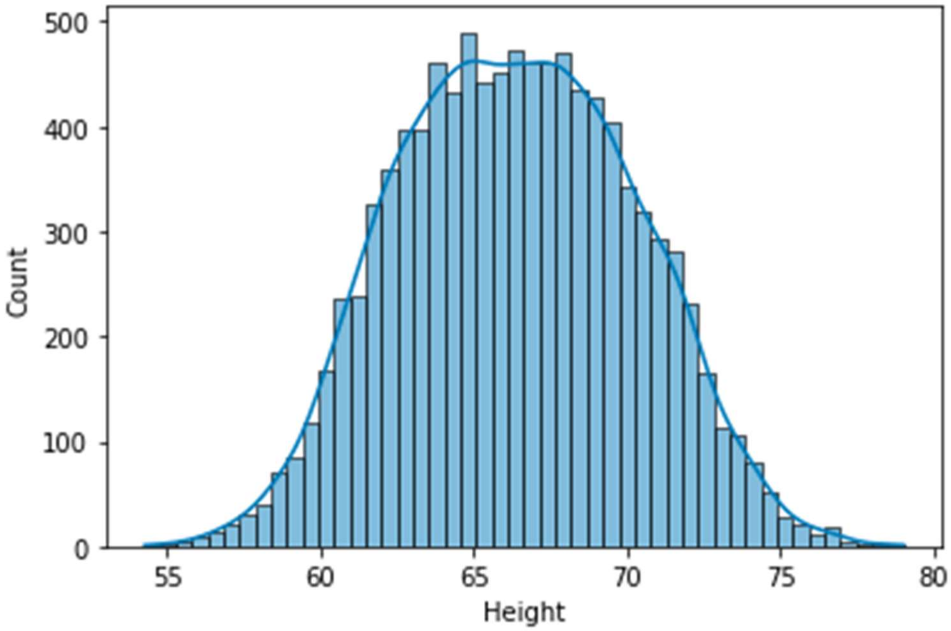
Output:



I.III Bell Curve:

```
import pandas as pd

import seaborn as sn

df=pd.read_csv('/content/sample_data/weight-height.csv')

df.head()

sn.histplot(df.Height, kde=True)
```

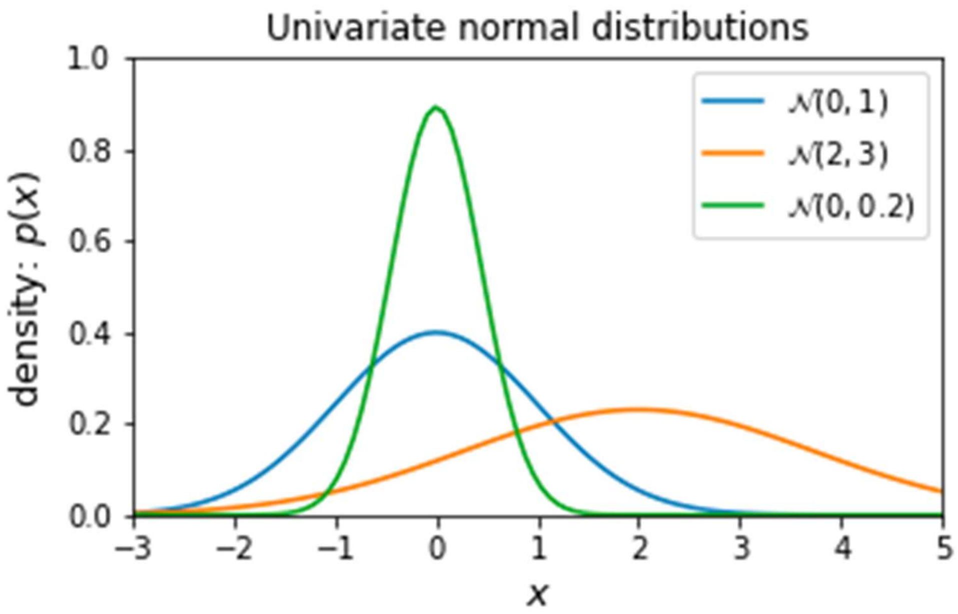
Output:



I.IV Univariate Distribution:

```
import numpy as np

import matplotlib

import matplotlib.pyplot as plt

from matplotlib import cm # Colormaps

import matplotlib.gridspec as gridspec

from mpl_toolkits.axes_grid1 import make_axes_locatable

import seaborn as sns




def univariate_normal(x, mean, variance):

   """pdf of the univariate normal distribution."""

   return ((1. / np.sqrt(2 * np.pi * variance)) *

           np.exp(-(x - mean)**2 / (2 * variance)))



x = np.linspace(-3, 5, num=100)

fig = plt.figure(figsize=(5, 3))

plt.plot(x, univariate_normal(x, mean=0, variance=1),
label="$\mathcal{N}(0, 1)$")

plt.plot(x, univariate_normal(x, mean=2, variance=3),
label="$\mathcal{N}(2, 3)$")

plt.plot( x, univariate_normal(x, mean=0, variance=0.2),
label="$\mathcal{N}(0, 0.2)$")

plt.xlabel('$x$', fontsize=13)
plt.ylabel('density: $p(x)$', fontsize=13)

plt.title('Univariate normal distributions')

plt.ylim([0, 1])

plt.xlim([-3, 5])

plt.legend(loc=1)

fig.subplots_adjust(bottom=0.15)

plt.show()
```

Output:



I.V Gaussian Process Regression:

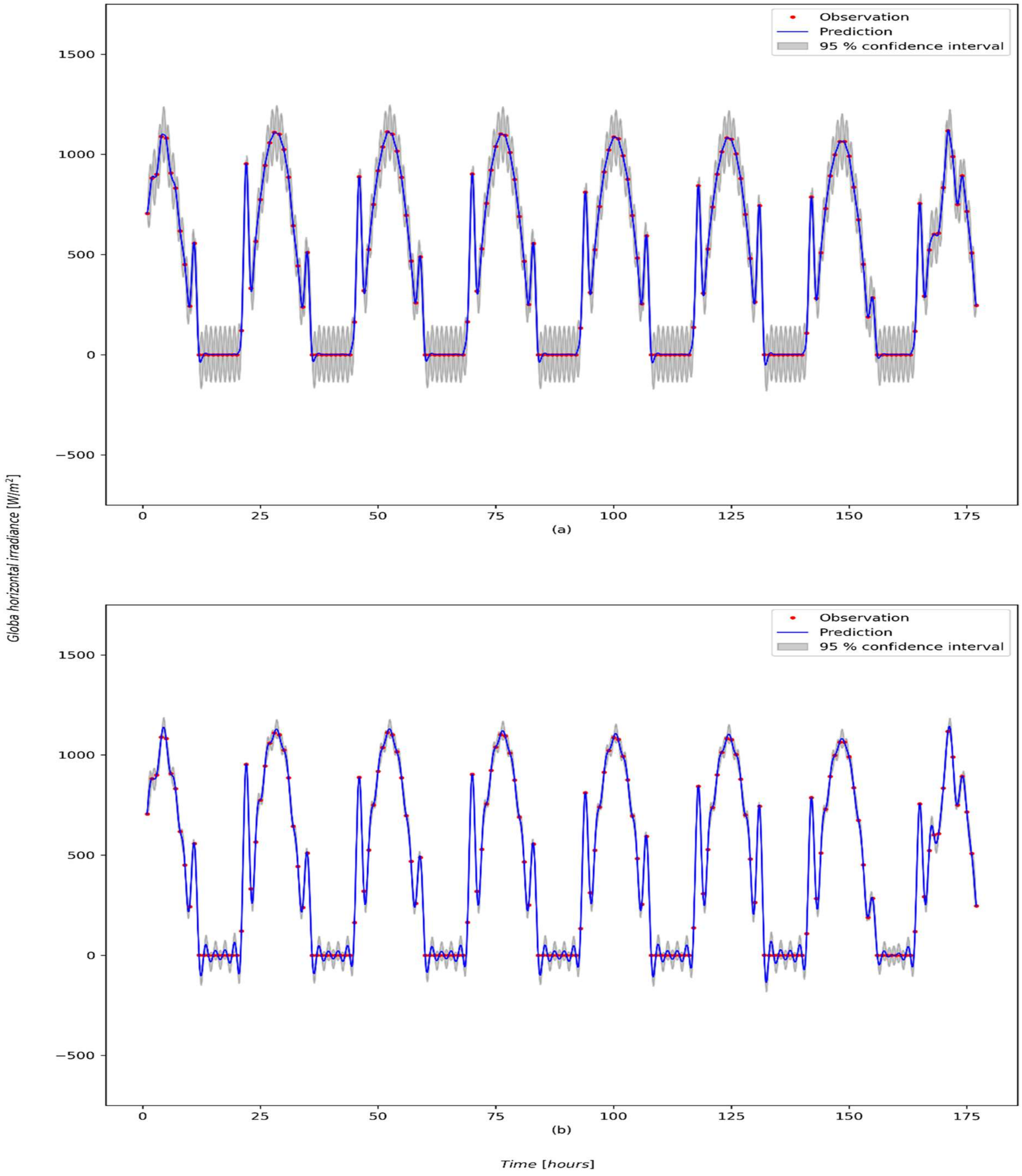
```
import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

from sklearn.gaussian_process import GaussianProcessRegressor

from sklearn.gaussian_process.kernels import RBF, ConstantKernel as C, RationalQuadratic as RQ, WhiteKernel, ExpSineSquared as Exp, DotProduct as Lin



np.random.seed(1)



df = pd.read_csv('/content/sample_data/weather_data.csv', sep=';')



df_array = np.asarray(df)



date = df_array[0:177, 0]

rec_num = df_array[0:177, 1]

ghi = df_array[0:177, 2]

DNI = df_array[0:177, 3]

DHI = df_array[0:177, 4]

DHI_shadowband = df_array[0:177, 5]

UVA = df_array[0:177, 6]

UVB = df_array[0:177, 7]

air_temp = df_array[0:177, 8]
```

```
BP = df_array[0:177, 9]

RH = df_array[0:177, 10]

WS = df_array[0:177, 11]

WD = df_array[0:177, 12]

WD_SD = df_array[0:177, 13]



y = np.asarray(

   [ghi, DNI, DHI, DHI_shadowband, UVA, UVB, air_temp, BP, RH, WS, WD,

    WD_SD]).T



X = np.atleast_2d([

   1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11., 12., 13., 14., 15., 16.,
17.,18., 19., 20., 21., 22., 23., 24., 25., 26., 27., 28., 29., 30., 31.,
32.,33., 34., 35., 36., 37., 38., 39., 40., 41., 42., 43., 44., 45., 46.,
47.,48., 49., 50., 51., 52., 53., 54., 55., 56., 57., 58., 59., 60., 61.,
62.,63., 64., 65., 66., 67., 68., 69., 70., 71., 72., 73., 74., 75., 76.,
77.,78., 79., 80., 81., 82., 83., 84., 85., 86., 87., 88., 89., 90., 91.,
92.,93.,  94.,  95.,  96.,  97.,  98.,  99.,  100.,  101.,  102.,  103.,  104.,
105.,106., 107., 108., 109., 110., 111., 112., 113., 114., 115., 116.,
117.,118., 119., 120., 121., 122., 123., 124., 125., 126., 127., 128.,
129.,130., 131., 132., 133., 134., 135., 136., 137., 138., 139., 140.,
141.,142., 143., 144., 145., 146., 147., 148., 149., 150., 151., 152.,
153.,154., 155., 156., 157., 158., 159., 160., 161., 162., 163., 164.,
165.,166., 167., 168., 169., 170., 171., 172., 173., 174., 175., 176., 177

]).T



#--------------------------------
```

```
x = np.atleast_2d(np.linspace(
   1, 177, 10739)).T


#--


kernel=C()*RQ(length_scale=24,alpha=1)


gp = GaussianProcessRegressor(kernel=kernel,n_restarts_optimizer=4)


gp.fit(X, y)
y_pred_1, sigma_1 = gp.predict(x, return_std=True)


#---


kernel=C()*Exp(length_scale=24,periodicity=1)


gp=GaussianProcessRegressor(kernel=kernel,n_restarts_optimizer=100)


gp.fit(X, y)
y_pred_2, sigma_2 = gp.predict(x, return_std=True)


#----------------------------------
```

```
#Plot figure

fig = plt.figure(num=1, figsize=(11, 0.8), dpi=300, facecolor='w',
edgecolor='k')

fig.text(0.5, -1, '$Time\ [hours]$', ha='center')

fig.text(0.04,10,'$Globa\ horizontal\ irradiance\
[W/m^2]$',va='center',rotation='vertical')

plt.subplot(2, 1, 1)

plt.plot(X, y[:, 0], 'r.', markersize=5, label=u'Observation')

plt.plot(x, y_pred_1[:, 0], 'b-', linewidth=1, label=u'Prediction')

plt.fill_between( x[:, 0],y_pred_1[:, 0] - 1.96 * sigma_1,y_pred_1[:, 0] +
1.96 * sigma_1,alpha=0.2,color='k',label=u'95 % confidence interval')

plt.xlabel('(a)')

plt.legend(loc='upper right', fontsize=10)

plt.ylim(-750, 1750)

plt.subplot(2, 1, 2)

plt.plot(X, y[:, 0], 'r.', markersize=5, label=u'Observation')

plt.plot(x, y_pred_2[:, 0], 'b-', linewidth=1, label=u'Prediction')

plt.fill_between(x[:, 0],y_pred_2[:, 0] - 1.96 * sigma_2,y_pred_2[:, 0] +
1.96 * sigma_2,alpha=0.2,color='k',label=u'95 % confidence interval')

plt.xlabel('(b)')

plt.legend(loc='upper right', fontsize=10)

plt.ylim(-750, 1750)

plt.subplots_adjust(top=20)

plt.savefig('all_in.png', bbox_inches='tight')

#--------------------------------
```

Output:


Observation
Prediction
95 % confidence interval
1500
1000
500
0
−500
0
25
50
75
100
125
150
175
(a)
Globa horizontal irradiance [W/m²]
Observation
Prediction
95 % confidence interval
1500
1000
500
0
−500
0
25
50
75
100
125
150
175
(b)
Time [hours]

Bibliography

[1] Rasmussen, C. E., & Williams, C. K. I., Gaussian processes for machine learning (2016), The MIT Press.

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et. al., Scikit-learn: Machine learning in python (2011), Journal of Machine Learning Research.

[3] Tolba, H.; Dkhili, N.; Nou, J.; Eynard, J.; Thil, S.; Grieu, S. Multi-Horizon Forecasting Using Gaussian Process Regression: A Kernel Study. *Energies* 2020, *13*, 4184.

[4] Csat´o, L. (2002). Gaussian Processes—Iterative Sparse Approximations. PhD thesis, Aston University, UK.

[5] Chu, W., Sindhwani, V., Ghahramani, Z., & Keerthi, S. (2006). Relational learning with gaussian processes. Canada: Vancouver.