Distributional Prediction of Greenhouse Gas Emissions Using Machine Learning

Yu-Sheng Chen
MS in NYU Financial Engineering
yc4468@nyu.edu

Industry Advisor: Ruslan Tepelyan

Table of Contents

1 Introduction

As climate change accelerates, investors are now demanding even higher standards of sustainability. It is reported that the companies with strong ESG scores generally attract better talents and have higher retention rate. Nowadays, companies' financial metrics alone are not sufficient in the investment decision-making process; ESG performance should also be taken into consideration for better investment decision.

While big companies are more willing to disclose the measures or indicators of their performance on environmental, social and governance issues for greater reputation, small companies usually don't have enough incentive to do so. As a result, investors often need to infer some specific ESG measures from the other metrics or sources.

In this project, I try to use machine learning techniques to predict one of the most crucial ESG metrics of companies — Greenhouse Gas (GHG) Scope 1 Emission Intensity (GHG Scope 1 divided by Revenue). Instead of making simple point forecasting, the model is designed in the way that can generate probabilistic forecasting. That is, the model will predict the expected distribution of the outcome given a set of inputted data. The kind of prediction allows for creating prediction intervals and thus provides rich information about the uncertainty of the prediction as well as the confidence of the model.

2 Data Extraction and Cleansing

All data are collected through the Bloomberg terminal and the company universe is built by combining Russell 3000 Index and Bloomberg ESG Index. Regarding the features, three categories of fields are fetched — (1) Fundamentals: total asset, market capitalization, revenue, net income, etc. (2) ESG-related: the number of employees, Bloomberg ESG disclosure score, etc. (3) Industry Sector: Bloomberg's Industry Classification (BICS) Level 4.

Due to Bloomberg API request limit, which indicates no more than 5000 unique identifiers per month, I narrowed down the cope of the company universe by only picking up the companies that are listed or domiciled in the United States. The reason why BICS Level 4 was chosen rather than level 3 or level 5 is because all companies have at least four levels; therefore, BICS Level 4 is just like the greatest common factor.

As for the target, GHG Scope 1 intensity is used as the label. GHG Scope 1 value represents a company's direct GHG emissions from sources that are owned or controlled by the reporting entity. If companies don't report GHG Scope 1 figures, estimated ones by the Bloomberg internal model are taken as proxy. If both GHG Scope 1 and estimated GHG Scope 1 are absent, that company would be discharged from the training data universe.

3 Data Exploration & Feature Selection

After filtering data with some criteria, there were approximately 20,000 rows of data left. When it came to handling the missing data, I ended up deciding to directly remove the data points with any missing value for three main reasons. Firstly, the majority of the missing data are fundamental data. If a company doesn't report revenue for a particular year, it is unlikely to disclose net income either. Imputation is apparently not suitable for this kind of situation. Secondly, a large portion of the missing values occurred before the IPO. All well-known techniques, such as mean imputation, zero imputation, regression imputation, forward filling and backward filling, are not sensible solutions to this case because they cannot reflect the rapid growth transition from private to public state. Last but most important, the model is constructed to predict distribution rather than mean. Intuitively speaking, the model should predict a narrower interval if there are more available features; however, it would be much difficult to achieve this goal if imputation is used because an additional feature must be added to indicate whether imputation is applied for each instance and proper loss function must be forged. Besides, imputation complicates the model evaluation as it becomes complex to identify whether errors are from the model itself or imputation.

As for the numerical features, descriptive and correlation analysis were conducted to examine their relationship to the response variable and determine which attributes may be more predictive. Total asset, revenue, prior year's GHG scope 1 figure (if it exists), etc., are among the most relevant features, while net income, free cash flow and current market capitalization bear little linear association with the response variable. (Figure 1)

Figure 1 Correlation Analysis for whole dataset

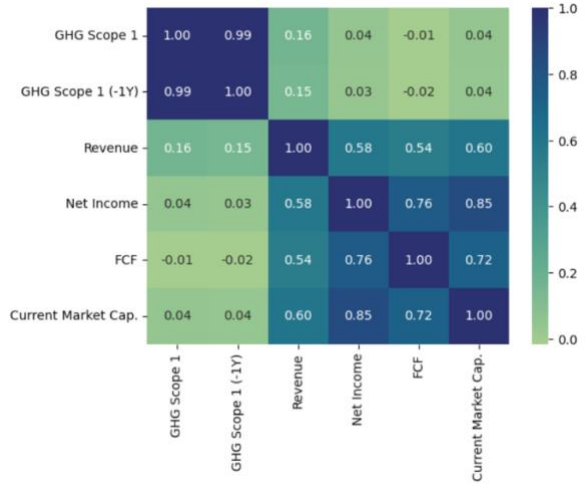
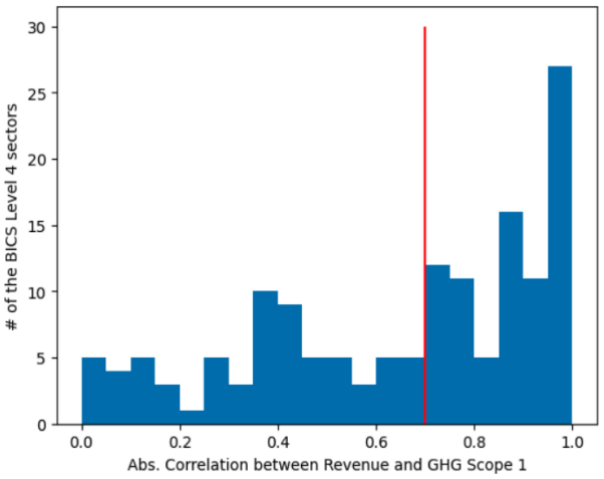


Figure 2 Correlation Analysis within subgroups



After the preliminary screening, data was divided into the subgroups based on the BICS Level 4 field and similar analyses were conducted on each subgroup. Among the total 190 sectors, about half of them show strong correlation (>0.7) between Revenue and GHG scope 1 as the figure 2 Without splitting, the correlation figure is mere 0.16. Therefore, BICS Level 4 serves as the first judging factor to the pipeline of the machine learning model.

4 Model Methodology

There are several methods and frameworks that enable probabilistic forecasting. They can be broadly divided into four categories — (1) Delta method: a well-known procedure used to quantify uncertainty in statistical models; however, it's computationally demanding due to use of the Hessian matrix (2) Bootstrap method: a resampling technique allowing for estimating statistics on a population by iteratively resampling a dataset with replacement (3) Bayesian method: a framework for manipulating uncertainty, learning from noisy data, and for making predictions that maximize likelihood, (4) Quantile regression: a method of predicting the conditioning quantiles of the dependent variable. Delta method and Bootstrap method are primarily applied in deep neural networks.

In this project, I will focus on Bayesian method and Quantile regression to provide prediction intervals. Particularly, Gaussian Process Regression is adopted in this project as it is a Bayesian, non-parametric and probabilistic machine learning algorithm.

4.1 Gaussian Process Regression (GPR)

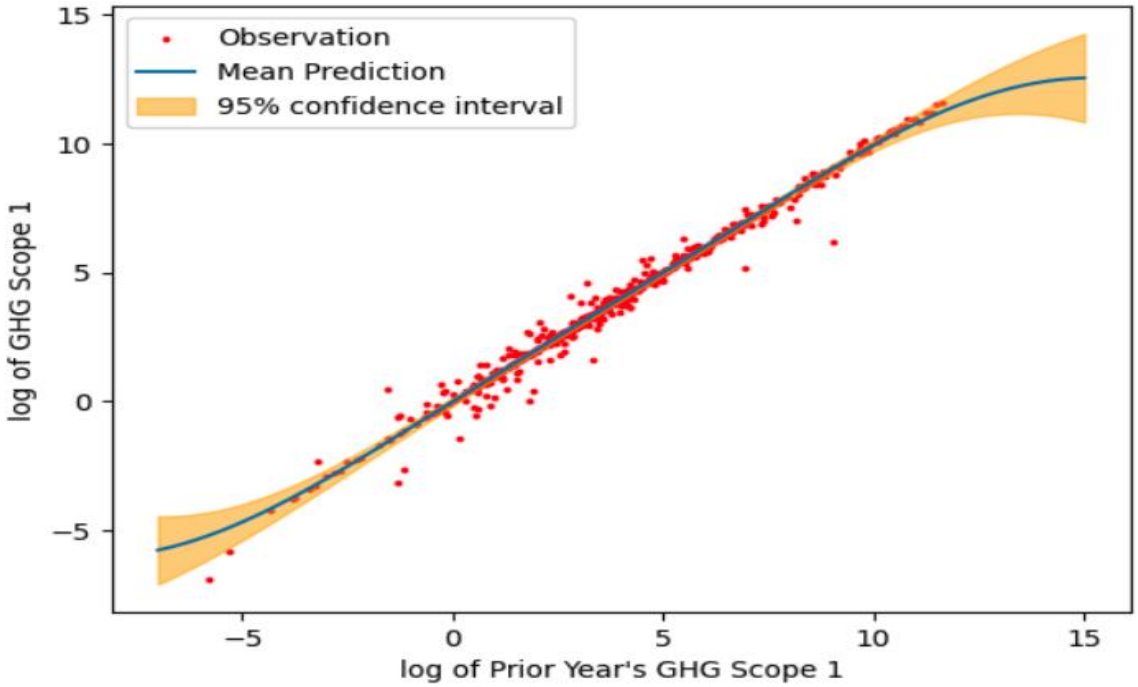
GPR is an extremely powerful machine learning framework as it merely relies on few parameters to make predictions. The core of this method is undoubtedly Gaussian process, which is a collection of random variables such that any finite number of them have a joint Gaussian distribution. In order words, Gaussian processes generalize multivariate Gaussian distributions over finite dimensional vectors to infinite dimensionality and defines a distribution over function space. Given a Gaussian process, a function at the point **x** can be expressed as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Where $m(\cdot)$ is a mean function and $k(\cdot\,,\,\cdot)$ is a covariance function. Covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a symmetric positive definite kernel function, identical to kernels used in Support Vector Machine (SVM) algorithm. Therefore, covariance function is the most crucial part of GPR models as it describes the statistical relationship between two points $\mathbf{x}, \mathbf{x}'$ in the input space and in turn determines the shape of prior and posterior as well as most of the properties of GPR model.

For better visualization, Figure 3 displays what GPR model on one feature looks like. The blue line is mean prediction, just like point-forecasting from the simple regression model, while the yellow area is 95% prediction intervals. If there are more data points in an area, the model would predict a narrower prediction interval because there are more nearby points for the model to get information from.

Figure 3 GPR model on one feature



Instead of merely using one feature, multiple variables are fed into the model. In order to incorporate several features, the kernel must be appropriately chosen or designed to correctly measure the similarity between pairs of points. The most widely used kernel is probably the Radial Basis Function (RBF) Kernel, also called the Squared Exponential Kernel —

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 e^{-\frac{(x-x')^2}{2\ell^2}}$$

This kernel is the de-facto default kernel for Support Vector Machines as well due to its nice property of being infinitely smooth and differentiable. In addition, it is relatively straightforward to integrate the RBF kernel with other functions. There are only two hyper-parameters for:

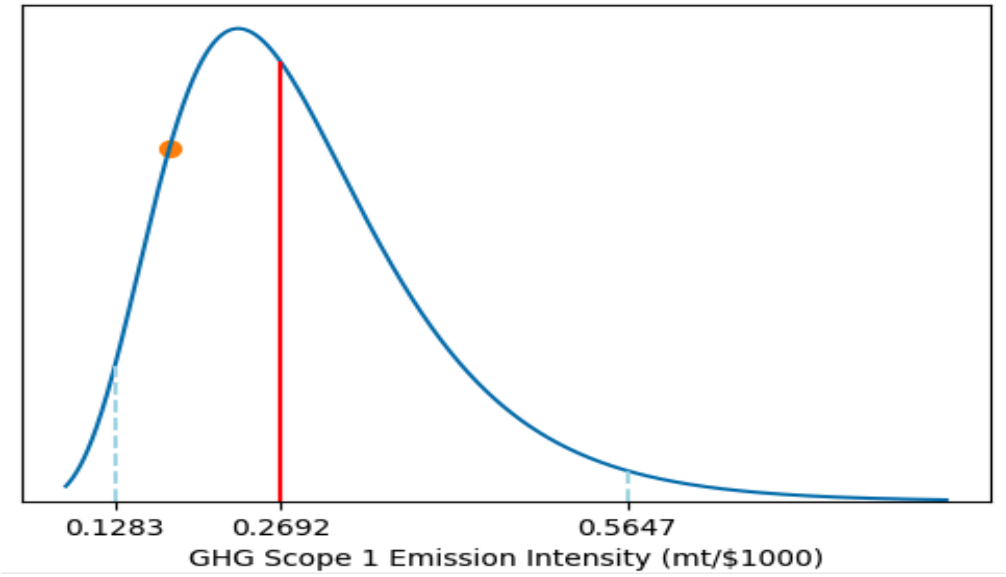
- the length scale ℓ determines how smooth the function is. If the length scale is chosen too small, the observation variance drops to zero rapidly and thus the resulting posterior is significantly wiggly and overfitting.
- the output variance σ^2 is just a scale factor. It decides the average distance of the function from its mean. If the output variance is chosen too small, the function will stay close to the mean. If the output variance is made too large, the function is likely to chase outliers.

There are also many different kernels, such as Linear Kernel, Period Kernel, Rational Quadratic Kernel and so forth. In my model, I put together Squared Exponential Kernel, White Noise Kernel and Constant Kernel by both multiplication and addition to reach the arguably optimal kernel.

Log transformation is applied on most of the numerical features for two reasons. Firstly, the covariance function (kernel) is stationary, but the magnitude of many numerical features can vary a lot. For example, GHG Scope 1 figures can range from single digit to four or five digits. Log transformation scales them down and generally makes the variance more homogeneous. The other reason is that our prediction interval should always be within the positive range because theoretically observed GHG Scope 1 figure cannot be negative. After the log transformation, the resulting prediction intervals are guaranteed to be in positive territory as the log-normal distribution is defined only for positive values.

As a simple illustration of the model output, Figure 4 displays the model prediction for Google's 2021 GHG Scope 1 emission intensity. The blue curve is the predicted distribution, which is positively skewed; the red vertical line is the predicted mean; the orange point is the true value; the area between two dotted light blue lines is 95% prediction interval.

Figure 4 Model Prediction for Google's 2021 GHG Scope 1 Emission Intensity



As stated before, BICS Level 4 is used as the first judging factor in the model. To be specific, distinctive model is trained for different BICS Level 4 sector. Consequently, there are roughly 190 sub-models wrapped in the whole structure. This division not only clusters the similar data points so that the fitting function for each sector is not too bumpy, but also makes the computation much more tractable. If only a single model is developed, as the dataset becomes larger, the training process will soon become computationally infeasible given that the time complexity of GPR is $O(n^3)$ for matrix inversion.

4.2 Quantile Regression (QR)

Quantile regression, considered as an extension of classical ordinary least squares regression, is a supervised technique aimed at estimating the quantiles of the conditional distribution of some response variable. It allows for predicting the conditional quantiles of a response variable distribution in the linear model for different quantile levels, such as 0.10, 0.50 and 0.90. To be specific, QR relaxes the assumption of ordinary least squares regression that relationships between response and explanatory variables are the same across all value and no assumption is made about the distribution of the response variable, so QR is more robust to misspecification of the error term. Besides, QR is relatively insensitive to outliers in comparison to mean estimation using OLS.

Similar to the approach in GPR, separate quantile regression model is trained for different BICS Level 4 sector. In each BICS Level 4 sector, the model is trained for 9 different percentiles, from 10th to 90th with the increment of 10. For illustration, Figure 5 displays the prediction of the linear quantile regression model trained with only one feature (Prior Year's GHG Scope 1) for 20th, 50th and 80th percentile. After the models are trained, prediction intervals can be made for sets of inputs. For example, if the 60% prediction interval is desired, the forecast would be the scope between the prediction by the model trained for the 20th percentile and the model trained for the 80th percentile.

Unlike the data preprocessing in the GPR model, log transformation is not applied on the numerical features and the response variable because QR does not assume a constant variance for the target value, indicating that there is no need to stabilize the variance of heteroscedastic data with a monotonic transformation.

Figure 5 Linear Quantile Regression

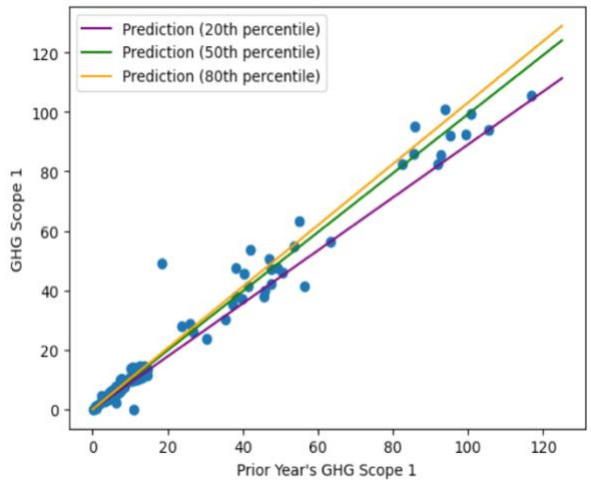
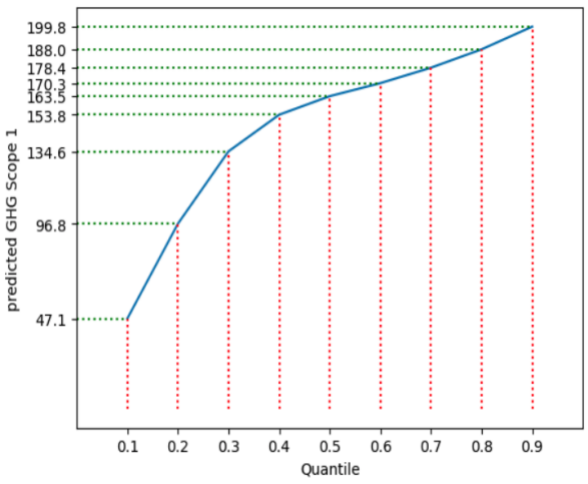


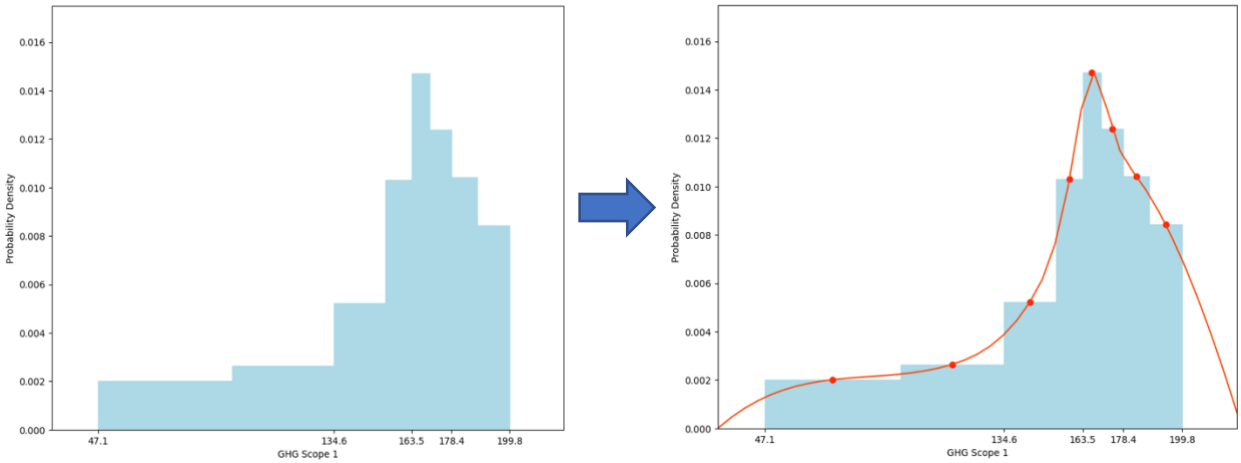
Figure 6 Quantile Function (inverse CDF)



The quantile function is also called inverse cumulative distribution function. To be specific, the quantile function is one way of prescribing a probability distribution and hence can be viewed as the unscaled and unnormalized discrete cumulative distribution function. Given the level of confidence, the corresponding predicted range can be easily pursued as shown in Figure 6.

In order to build up the corresponding probability density function, the following steps are performed as the Figure 7 shows. Firstly, the cumulative density function is approximated with the rectangles by normalizing the area; Secondly, the points in the middle of each quantile are connected using the cubic spline.

Figure 7 Approximation of Cumulative Distribution Function



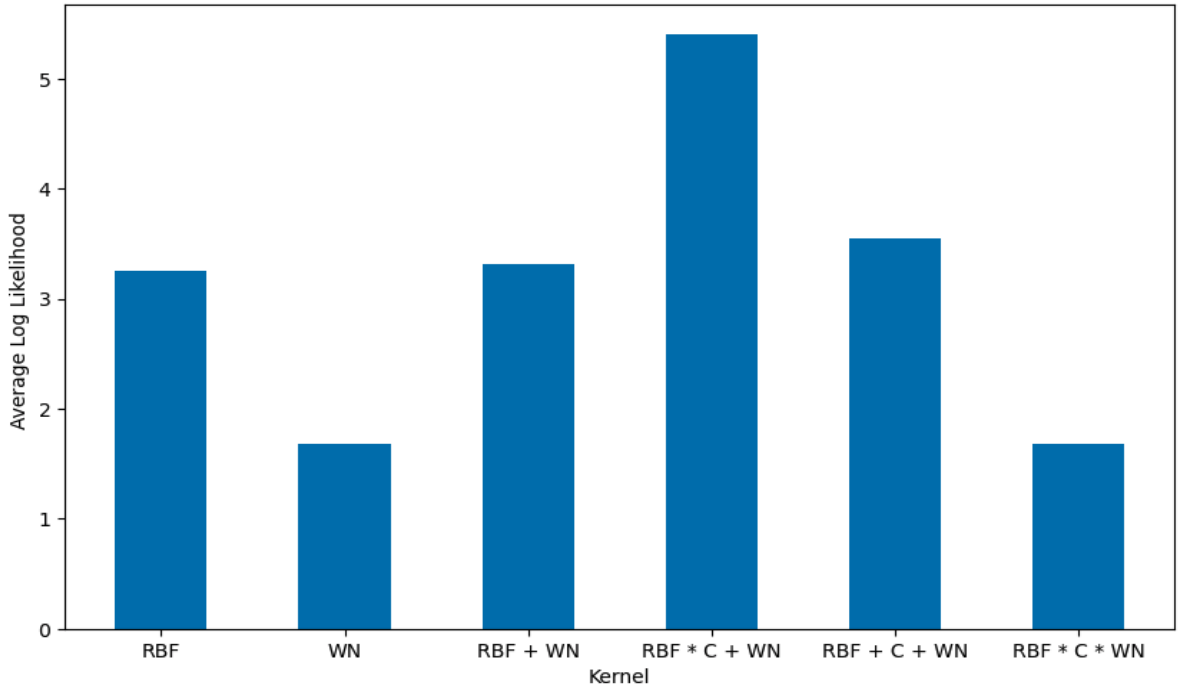
Ideally, the models should be trained with more quantiles so that the inferred probability density curve would be smoother and more accurate. However, data points in some sectors are insufficient to generate distinctive fitting curves for certain quantiles. For instance, if the models of the "Music" sector are trained for 99 different percentiles, from 1st to 99th with the increment of 1, the predictions by the models trained for the 1^{st} to 8^{th} percentiles are exactly the same. In this case, more quantiles instead lead to the difficulty of converting the quantile function to the probability distribution.

Last but not least, in order to ensure that the prediction intervals by the models would always be within the positive range, the fitted line is accepted only if the intercept is non-negative. If the intercept of the trained model is negative, the condition of passing the origin is imposed and the model is retrained.

5 Evaluation Framework

Root Mean Squared Error (RMSE) is used as the metric for evaluation of the model's ability to produce point estimates. For prediction intervals, log likelihood is used for comparison of the models with different kernels or hyper-parameters. As the kernel is the most crucial part of the Gaussian Process Regression model, different kernels are contrasted first. As the Figure 8 displays, the constant kernel multiplied by the RBF kernel plus a white noise kernel achieves the highest likelihood, which is within the expectation given its wide use.

Figure 8 Comparison of Different Kernels in Terms of Likelihood



Another hyper-parameter, ***n_restarts_optimizer***, is also important as it specifies the number of restarts of the optimizer for finding the kernel's parameters which maximize the log-marginal likelihood. Ideally, this hyper-parameter should be set to a reasonably large number so that the model will more likely reach the global maximum instead of plunging into local maxima. However, the Gaussian Process Regression is inherently a computationally intensive algorithm and repetitious training is barely affordable. Fortunately, the Figure 9 shows that the model arrives at the plateau when the number of restarts reaches 4. As a result, ***n_restarts_optimizer*** is set to 5 in my model.

In order to check the consistency of the model, variances are computed for the models using different partitions of the training dataset. As the Figure 10 illustrates, the variability in the model performance is low and hence the model is quite consistent.

Figure 9 Likelihood of different restart #

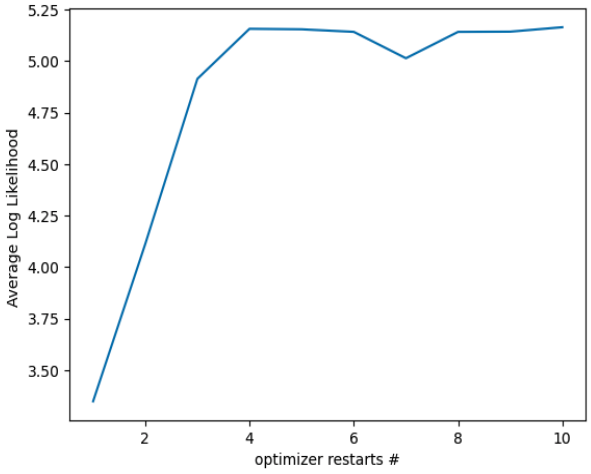
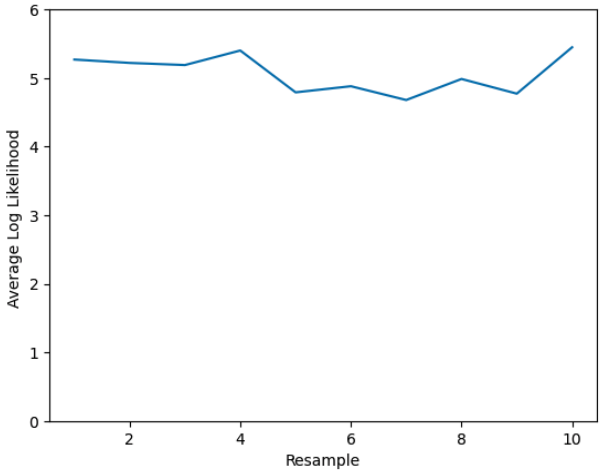
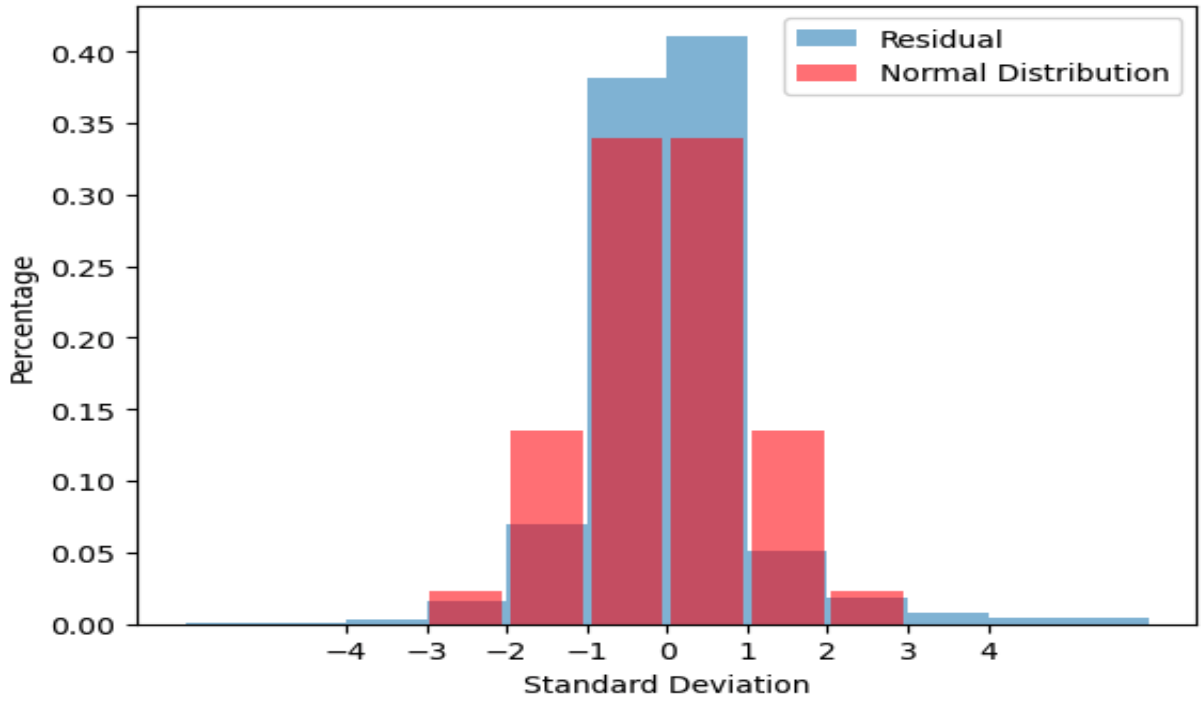


Figure 10 Likelihood of different training sets



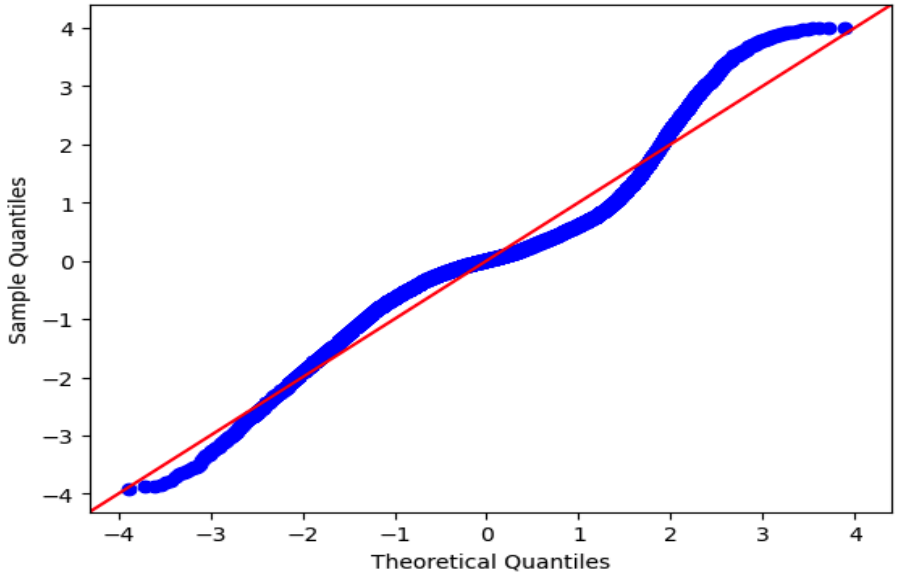
The residuals of the model prediction are looked at to test whether the assumption of normality is correct. The immediate prediction from the model before the exponential transformation is used for analyses. All true values are standardized with the predicted mean and predicted standard deviation. Firstly, the data is visualized using the histogram and compared to the normal distribution. As you can see in the Figure 11, the distribution of the residuals is bell-shaped and symmetric, albeit with the heavier tails and the higher peak.

Figure 11 Distribution of Residuals vs. Normal Distribution



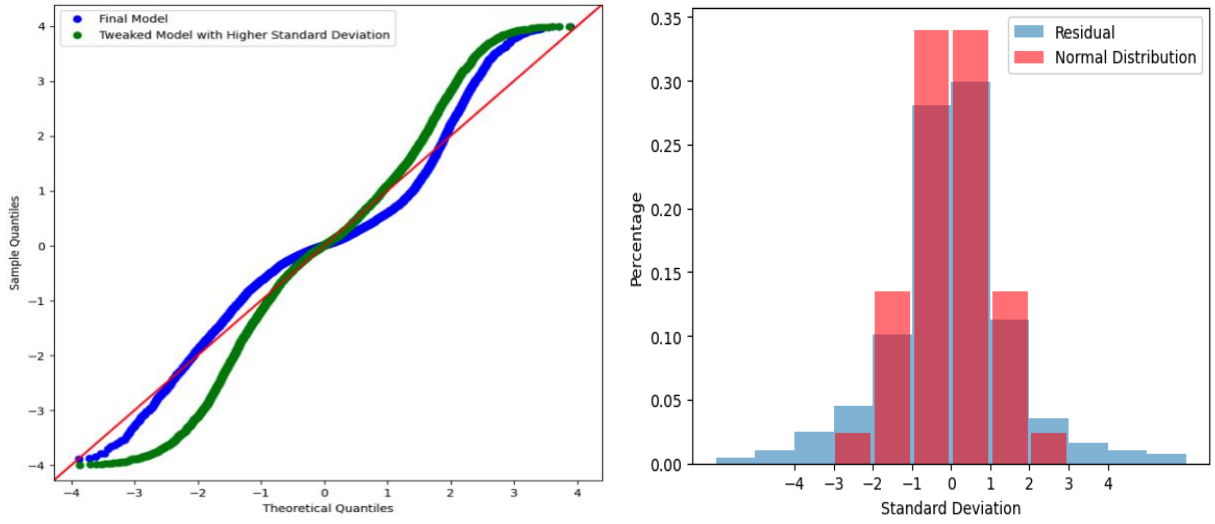
Secondly, Quantile vs Quantile Plot is adopted. Theoretical quantiles are plotted against the actual quantiles of the residuals. The Figure 12 reveals the observations similar to the previous one. There are too many data points in the middle as well as both extremities but fewer instances in the range from one to three standard deviations.

Figure 12 QQ-Plot of the Residuals



This situation cannot be ironed out owning to the limitation of the Gaussian Process Regression model. For example, if the model is tweaked so that the predicted standard deviations become smaller, the middle section of the distribution of the residuals will mirror the normal distribution more closely, but the tails will become even fatter simultaneously, shown in Figure 13. In other words, the assumption of normality wipes out the possibility of fixing both issues at the same time. Student-T Process Regression might be a potential alternative algorithm.

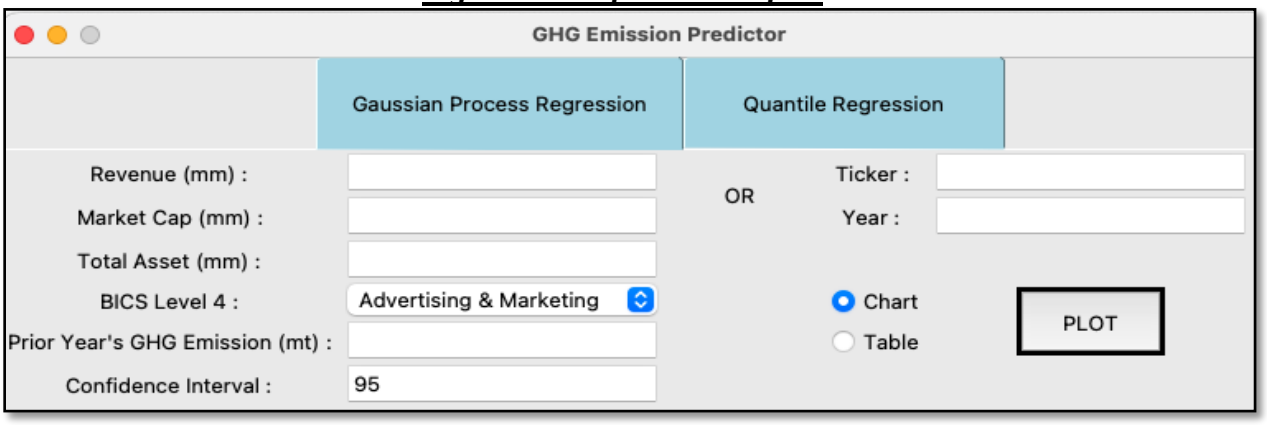
Figure 13 Artificially Lower the Predicted Standard Deviations



6 User Interface

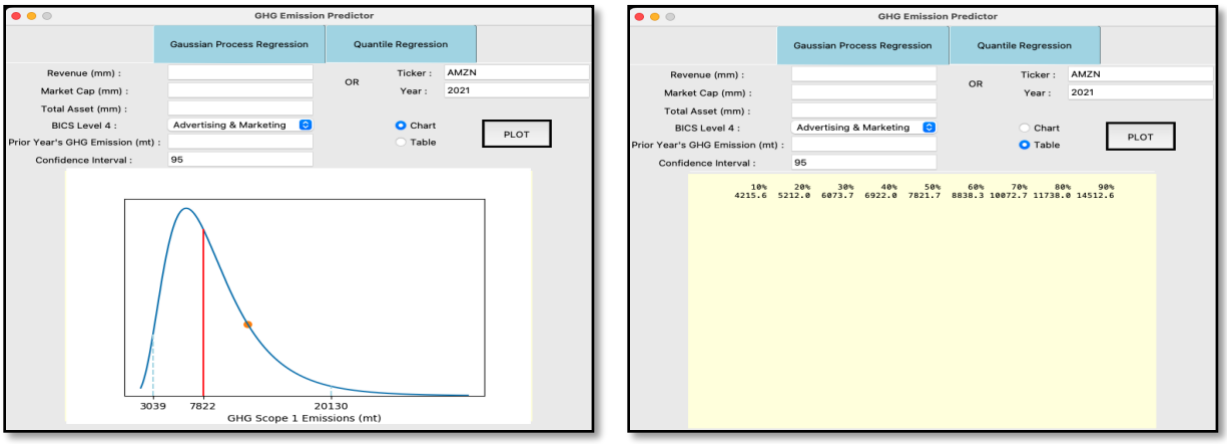
In order to build an interactive and graphical application, Tkinter was used to create the interface. Tkinter is a portable graphical user interface construction library shipped with Python as a standard library module. It is widely used, comprehensively documented, mature, well-supported and simple to implement.

Figure 14 Graphical Interface



The main menu of the application is shown in the Figure 14. Through the blue tabs at the topmost side, you can first choose whether Gaussian Process Regression or Quantile Regression is used to make distributional prediction. Regarding the data input, there are two available ways — one way is to input the required fields such as Revenue, Market Capitalization, Total Asset, Prior Year's GHG Scope 1 Emission (optional) and selecting the most suitable BICS Level 4 sector; the other way is to directly enter the stock ticker as well as the year in which you have interest, and the matched data will be fetched if it exists in the internal dataset (An error will pop up if there is no matched data corresponding the inputted ticker plus the year). At the right bottom side of the panel, you can choose whether the result is displayed in the chart format or tabular format (Figure 15).

Figure 15 Prediction for Amazon's 2021 Emission in Chart and Tabular Format



7 Conclusion

The aim of this project has been to conduct some data analysis on ESG data and develop machine learning models that are able to forecast GHG Scope 1 emission intensity. Instead of merely generating point-prediction, the models were designed particularly so that they can make distributional prediction to quantify the uncertainties, including both the noise inherently in the estimates or the model parameters as well as the uncertainty of out-of-sample measurement. I have decided to build one nonlinear model and one linear model and companies' financial metrics, ESG fields as well as industry segmentation data were used as the features.

As for the non-linear model, Gaussian Process Regression (GPR) was chosen. As a non-parametric and probabilistic machine learning algorithm, the model is able to directly make prediction intervals. While the overall prediction performance of the trained model was relatively stellar, there are some limitations of this algorithm — (1) the response variable is assumed to bear Gaussian noise, which may not be the case. (2) whole dataset is used to perform prediction, which is rather computationally expensive especially in high dimensional space. (3) the prediction on the boundary cases sometimes doesn't make intuitive sense. For example, the predicted GHG Scope 1 emission intensity may go down when the number of employees go up, all other variables held constant.

Regarding the linear model, Quantile Regression was adopted. Unlike the GPR model, Quantile Regression cannot directly make prediction intervals, so I derived the distributional prediction following the steps shown in Section 4-2. Quantile Regression assumes no error distribution and is relatively resistant to outliers, so the trained model is theoretically robust; however, the prediction intervals by the Quantile Regression model tend to be more inaccurate and much wider in comparison the ones predicted by the GPR model.

In this project, the company universe was confined to encompass merely the US companies and the number of features was small due to the limited capacity of downloading data in the Bloomberg Lab. For future works, the models presumably can be ameliorated further with a larger dataset, especially for the Quantile Regression model. On the other hand, the parameters of the GPR model can also be optimized using the machine with a higher level of computing power. Owning to the limitation of the personal laptop, the number of restarts of the optimizer for finding the kernel's parameters was set to five, which likely only led to the local maximum of likelihood, rather than the global one.

To sum up, while the GPR model generates better prediction in terms of the overall performance measurement because it can incorporate non-linear effect between the features, the Quantile Regression model has superior interpretability as it assesses the impact of one feature on a quantile of the output conditional on specific values of the other features and only allows for linear relationships between the covariates. Each model has its own advantages and can be used jointly to assist in making environmentally conscious investment decisions.

8 Reference

Thomas Beckers. 2021. “An Introduction to Gaussian Process Models”

Tim Pearce, Mohamed Zaki, Alexandra Brintrup, Andy Neely. 2018. “High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach”

Bijan J Borah, Anirban Basu. 2013. “Highlighting differences between conditional and unconditional quantile regression approaches through an application to assess medication adherence”

Andrew G. Wilson, Ryan P. Adams. 2013. “Gaussian Process Kernels for Pattern Discovery and Extrapolation”

David K. Duvenaud, Hannes Nickisch, Carl Rasmussen. 2011. “Additive Gaussian Processes”

Arman Melkumyan, Fabio Ramos. 2006. “Multi-Kernel Gaussian Processes”

Peter Sollich, Christopher K. I. Williams. 2004. “Using the Equivalent Kernel to Understand Gaussian Process Regression”