Bayesian Docking-Score Predictor

Created: 11/15/2023 by Tom Lever
Updated: 12/07/2023 by Tom Lever

Introduction

According to Dr. Bill Basener, “the goal of the project is for you to apply Bayesian machine learning to a real dataset in an advanced way. This means that the project should show you can apply probabilistic reasoning to a nontrivial problem of your choosing.”

Naomi Ohashi and Tom Lever have developed Bayesian Docking-Score Predictor. Bayesian Docking-Score Predictor receives the Simplified Molecular Input Line Entry System (SMILES) of a ligand and provides a docking score of how well that ligand binds to a protein.

Our predictor receives the SMILES of a ligand. A SMILES is a string representing the chemical structure of a ligand. A ligand is a molecule that binds to another molecule that is usually larger. There is one SMILES per ligand. There is one ligand per SMILES.

A docking score is a measure of how well a ligand binds to another molecule. A docking score is a change Δg in the molar Gibbs free energy g of a compound. A docking score is measured in kilocalories per mole. According to Dr. Ryan Weil, $-15\ \frac{kilocalories}{mol}$ is amazing and $-9\ \frac{kilocalories}{mol}$ is okay. See below *Appendix 1: Exploring Change In Molar Gibbs Free Energy*.

Data

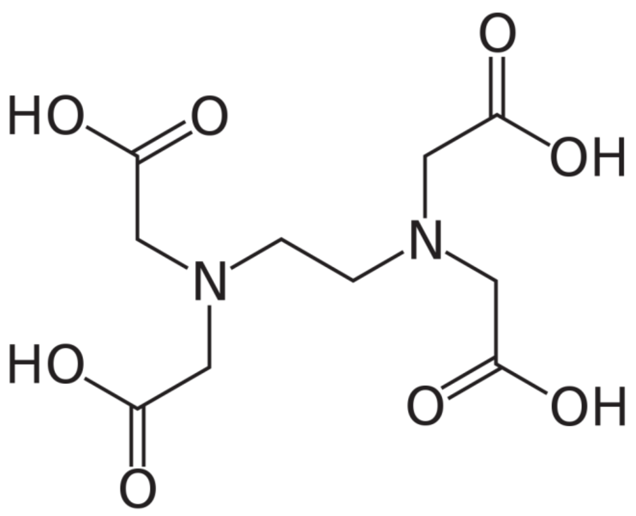
Our raw data set is a CSV file called *Data Frame Of SMILESs Docking Scores And Other Data*. Our data set is associated with one protein and with one site of that protein at which ligands bind. Each row in our data set encapsulates data relating to a ligand docking to the protein. There is one ligand per row. There is one row per ligand.

We use the above data set to a construct a CSV file called *Data Frame Of Docking Scores And SMILESs*. This is our primary data set. There are 2,121,226 observations, each with a docking score and a SMILES.

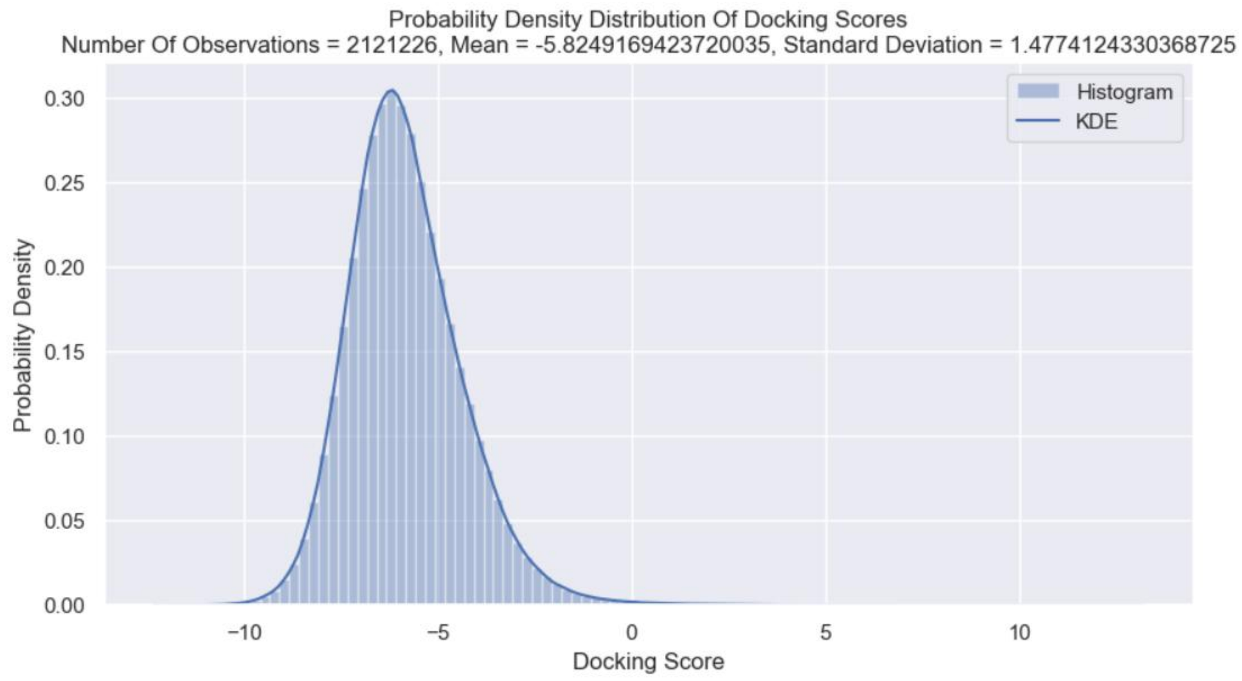
According to the National Institute of Health, Ethylenediaminetetraacetic acid (EDTA) is a medication used in the management and treatment of heavy metal toxicity. Its SMILES is

OC(=O)CN(CCN(CC(O)=O)CC(O)=O)CC(O)=O

According to Wikipedia, its chemical structure is



See below probability distribution of docking scores in our data set.



Methods

A SMILES of a ligand is converted into a numerical representation of the chemical structure of the ligand. The numerical representation is a vector of the number of occurrences of certain common substructures of molecules. One substructure is a structure of carbon atoms. Each vector of numbers of occurrences of substructures is folded into a vector of 1024 numbers of occurrences of substructures. We also experimented with converting a SMILES into a vector of values of molecular descriptors relating to shape, lipophilicity, polarity, and propensity to form hydrogen bonds.

We construct a feature matrix containing training and testing observations. An observation is a vector containing and docking score and 1024 numbers of occurrences of substructures. Our feature matrix has up to 2,121,226 observations.

According to YouTube video Statistical Learning: 8.6 Additive Regression Trees, "It turns out that the [Bayesian Additive Regression Trees (BART)] method can be viewed as a Bayesian approach to fitting an ensemble of trees: each time we randomly perturb a tree in order to fit the residuals, we are in fact drawing a new tree from a posterior [probability] distribution. Furthermore, the BART algorithm can be viewed as a Markov Chain Monte Carlo procedure for fitting the BART model. We typically choose large values for [number of iterations] B and [number of trees] K, and a moderate value for [number of burn-in iterations] L: for instance, K = 200, B = 1000, and L = 100 are reasonable choices. BART has been shown to have impressive out-of-box performance - that is, it performs well with minimal tuning."

Interestingly, per the paper BART: Bayesian Additive Regression Trees, "BART's many features are illustrated with a bake-off against competing methods of 42 different datasets, with a simulation experiment and on a drug discovery classification problem."

We train and predict with a *Bayesian Model Using A Bayesian Additive Regression Trees (BART) Model*. In this model, a BART model with 50 trees is trained to map a vector $\vec{x}$ of random variables x_i to a response variable $\hat{y}$. Each random variable x_i represents a number of occurrences of substructures. Response variable $\hat{y}$ represents a predicted docking score. $\hat{y}$ approximates a random variable y that represents an observed docking score.

$$y = \hat{y} + \epsilon = BART(\vec{x}) + \epsilon$$

where ϵ is an error. We assume that ϵ is normally distributed with mean $\mu = 0$ and standard deviation σ.

$$\epsilon \sim N(\mu = 0, \sigma)$$

Thus, y is normally distributed with mean $\mu = BART(\vec{x})$ and standard deviation σ.

$$y \sim N[\mu = BART(\vec{x}), \sigma]$$

In this model, we use Python packages `pymc` and `pymc-bart` to estimate a joint posterior probability distribution for training data

$$f(\mu, \sigma \mid y) = \frac{f(y \mid \mu, \sigma) f(\mu, \sigma)}{\int_{\mu=-\infty}^{\infty} \int_{\sigma=-\infty}^{\infty} f(y \mid \mu, \sigma) f(\mu, \sigma)\, d\sigma\, d\mu} = \frac{f(y \mid \mu, \sigma) f(\mu) f(\sigma)}{\int_{\mu=-\infty}^{\infty} \int_{\sigma=-\infty}^{\infty} f(y \mid \mu, \sigma) f(\mu) f(\sigma)\, d\sigma\, d\mu}$$

In our code, we define a matrix $\boldsymbol{X}$ of all training values of $\vec{x}$, a tensor variable representing μ, a vector of training values of y, a tensor variable representing a prior probability distribution $f(\sigma)$ that is half normal with standard deviation 100, and a tensor variable representing likelihood $f(y \mid \mu, \sigma)$.

We use `pymc` to sample testing predicted docking scores / testing values of $\hat{y}$ from the posterior predictive probability distribution

$$f(\hat{y} \mid y) = \int_{\mu=-\infty}^{\infty} \int_{\sigma=-\infty}^{\infty} f(\hat{y}_{test} \mid \mu, \sigma)\, f(y_{test} \mid \mu, \sigma)\, f(\mu, \sigma)\, d\sigma\, d\mu$$

The number of testing values of $\hat{y}$ sampled is the product of a number of chains (e.g., 4), a number of samples per chain (e.g., 1000), and a number of testing observations (e.g., 1,060,613). 4 chains were sampled by default. We find vectors of averages and standard deviations of testing values of $\hat{y}$ sampled. Each vector has length equal to the number of testing observations.

We also train and predict with a Bayesian Neural Network (BNN). In this model, a neural network is trained to map a vector vector-x of random variables x-subscript-i to a response variable y-hat. y-hat is multiplied by the range of observed response values y to encourage our BNN to predict docking scores across the range of observed docking scores. Unfortunately, our BNN predicts docking scores across an interval of docking scores centered at 0 that is narrow when compared to the range of observed docking scores. See our density plot below. This may be due to our training docking scores being roughly normally distributed, our BNN being shallow, or our BNN being improperly trained. Solutions may include training on uniformly distributed docking scores, deepening our BNN, or tuning our BNN.

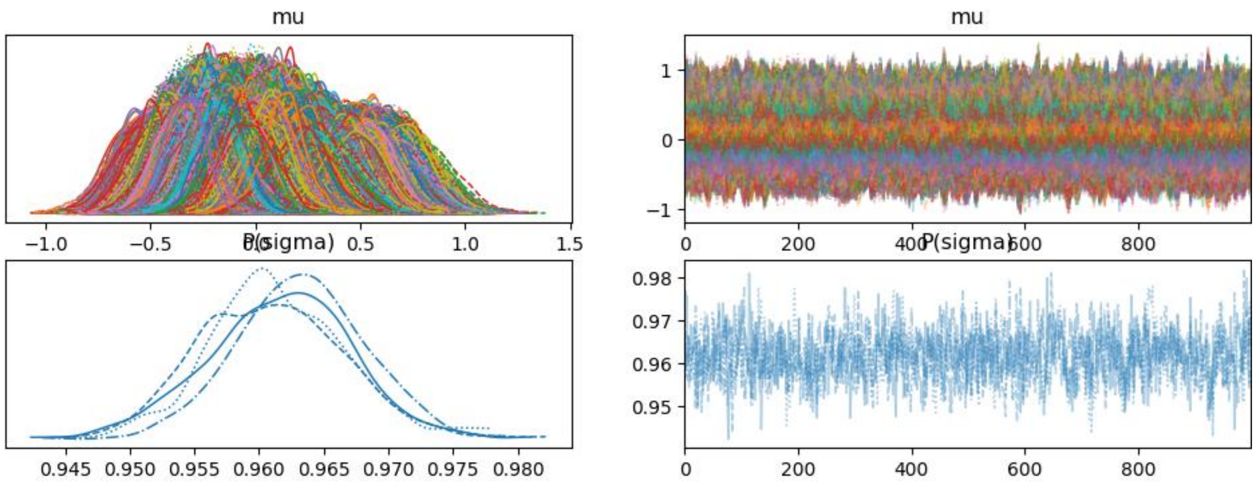
Our BNN has:
An input layer through which the BNN receives an input matrix X belonging to the set of all m × 1024 matrices, where m is the number of training and testing observations. As noted at the beginning of this article, I request help allowing the numbers of training and testing observations to be different.
A first hidden layer with an initial weights matrix W[h(1), 0] belonging to the set of all 1024 × 5 matrices. The weights matrix consists of random numbers sampled from the standard normal probability distribution.
A normal probability distribution from which weight matrices W[h(1)] are sampled. The normal probability distribution has shape (1024, 5), mean 0I, and standard deviation 1I.
A normal bias vector b[h(1), 0] belonging to the set of all 1 × 5 matrices. The bias vector consists of random numbers sampled from the standard normal probability distribution. Biases are added to each row of XW[h(1)].
A normal probability distribution from which bias matrices b[h(1)] are sampled. The normal probability distribution has shape (1, 5), mean 0I, and standard deviation 1I.
A hyperbolic tangent activation function.
A second hidden layer with:
An initial weights matrix W[h(2), 0] belonging to the set of 5 × 3 matrices. The weights matrix consists of random numbers sampled from the standard normal probability distribution.
A normal distribution from which weight matrices W[h(2)] are sampled. The normal distribution has shape (5, 3), mean 0I, and standard deviation 1I.
An initial bias vector b[h(2), 0] belonging to the set of all 1 × 3 matrices. The bias vector consists of random numbers sampled from the standard normal probability distribution.

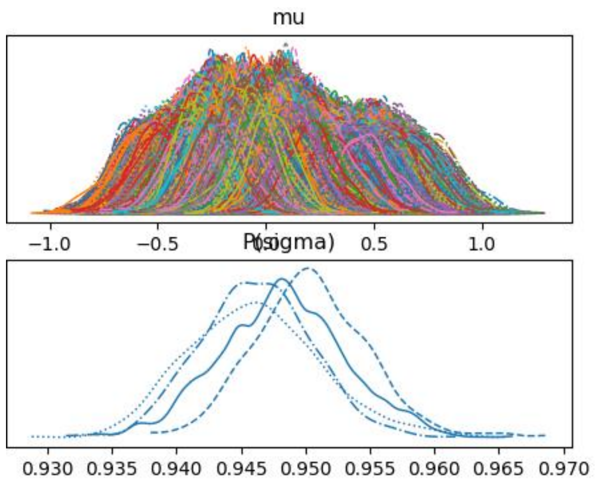
A normal distribution from which bias matrices b[h(2)] are sampled. The normal probability distribution has shape (1, 5), mean 0I, and standard deviation 1I.
A hyperbolic tangent activation function.
An output layer with:
An initial weights matrix W[o, 0] belonging to the set of all 3 × 1 matrices. The weights matrix consists of random numbers sampled from the standard normal probability distribution.
A normal distribution from which weight matrices W[o] are sampled. The normal distribution has shape (3, 1), mean 0I, and standard deviation 1I.

<u>Results</u>

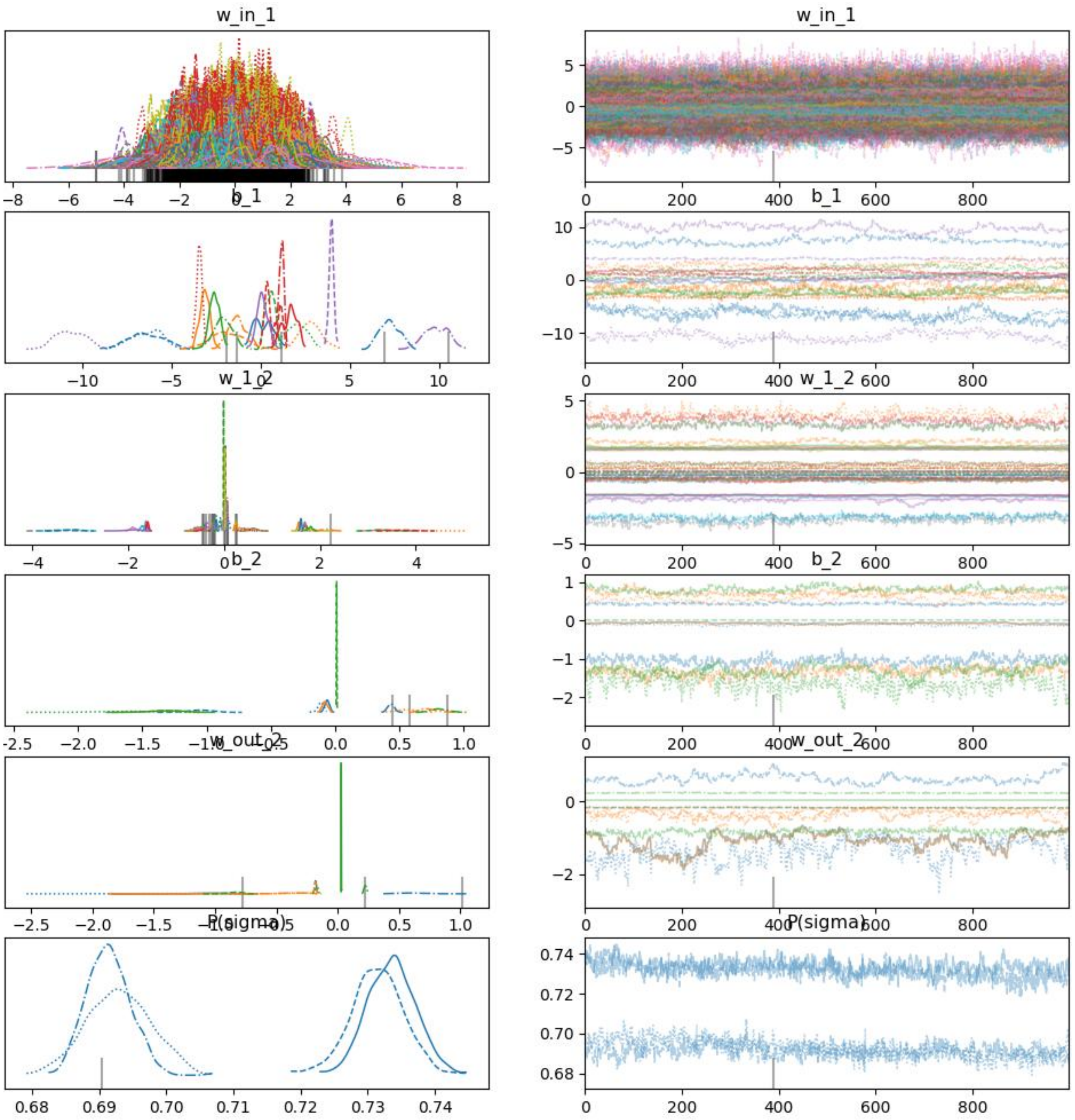
Trace Plots

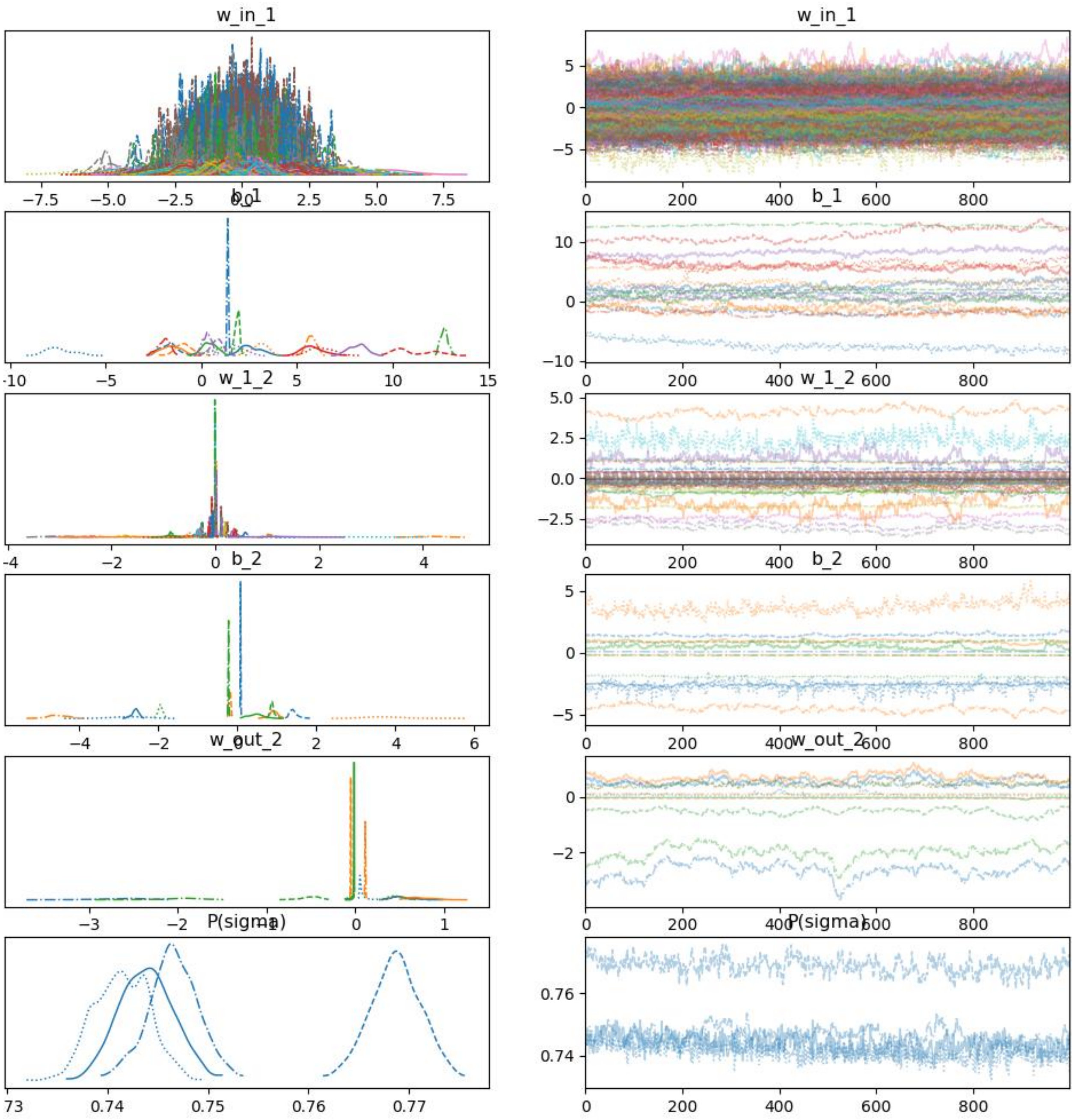
When we estimate a joint posterior probability distribution for training data, we may generate a Trace Plot of probability distributions. Each distribution is a probability distribution of values of a parameter in a parameter vector estimated as No U-Turn Sampling (NUTS) progresses. Below we visualize such distributions for our Bayesian Model Using BART Model when the model is trained with 33,145 and 66,289 observations. There is one distribution for each parameter and chain. 4 chains were sampled by default. The Trace Plot is less noisy when our model is trained with 66,289 observations.

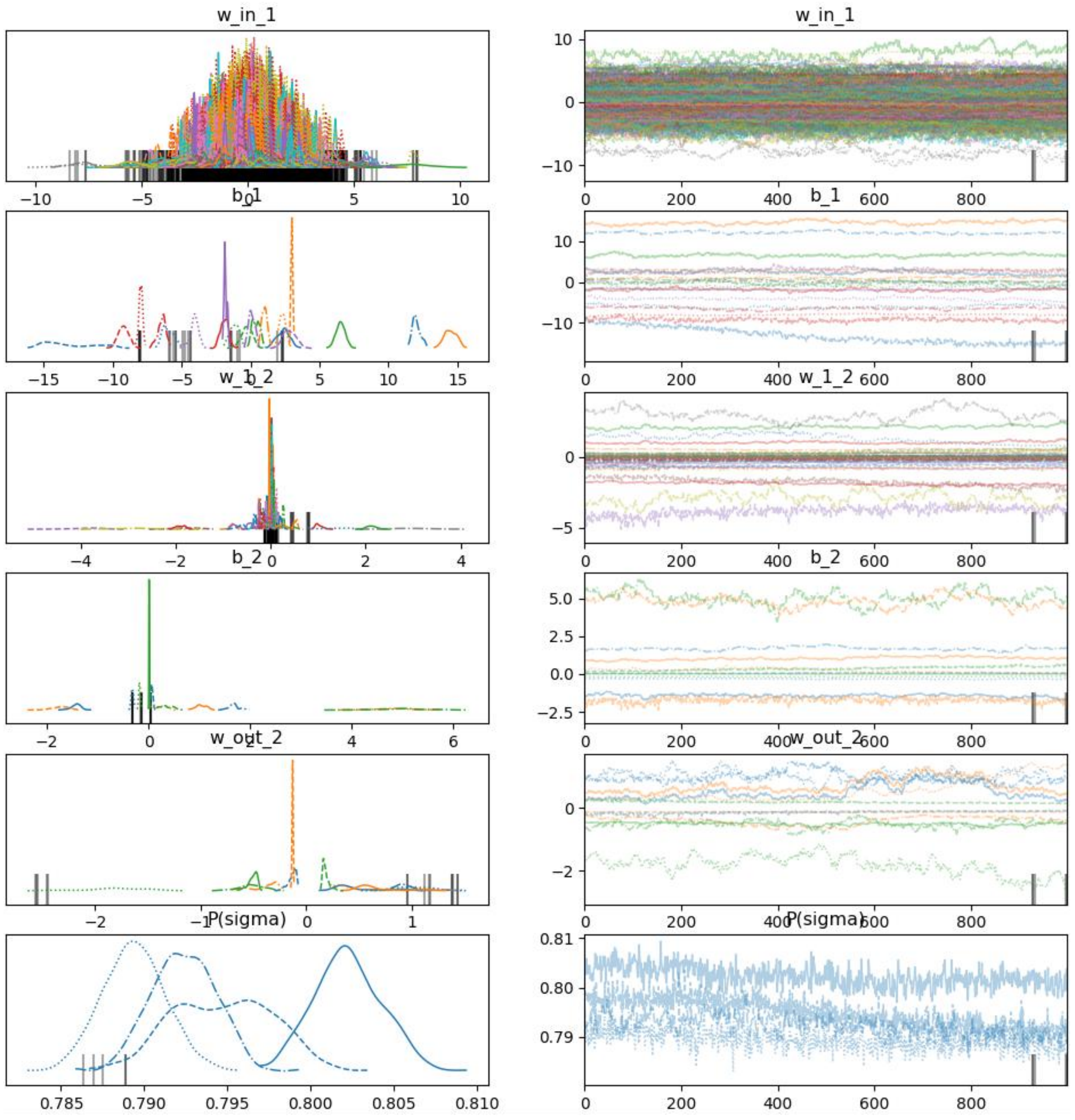




Below we visualize Trace Plots for our Bayesian Neural Network when the BNN is trained with 33,145, 66,289, and 132,577 observations. There is one distribution for each parameter and chain. 4 chains were sampled by default. The Trace Plot is least noisy when our BNN is trained with 66,289 observations.


w_in_1
w_in_1
b_1
b_1
w_1_2
w_1_2
b_2
b_2
w_out_2
w_out_2
P(sigma)
P(sigma)


w_in_1
b_1
w_1_2
b_2
w_out_2
P(sigma)

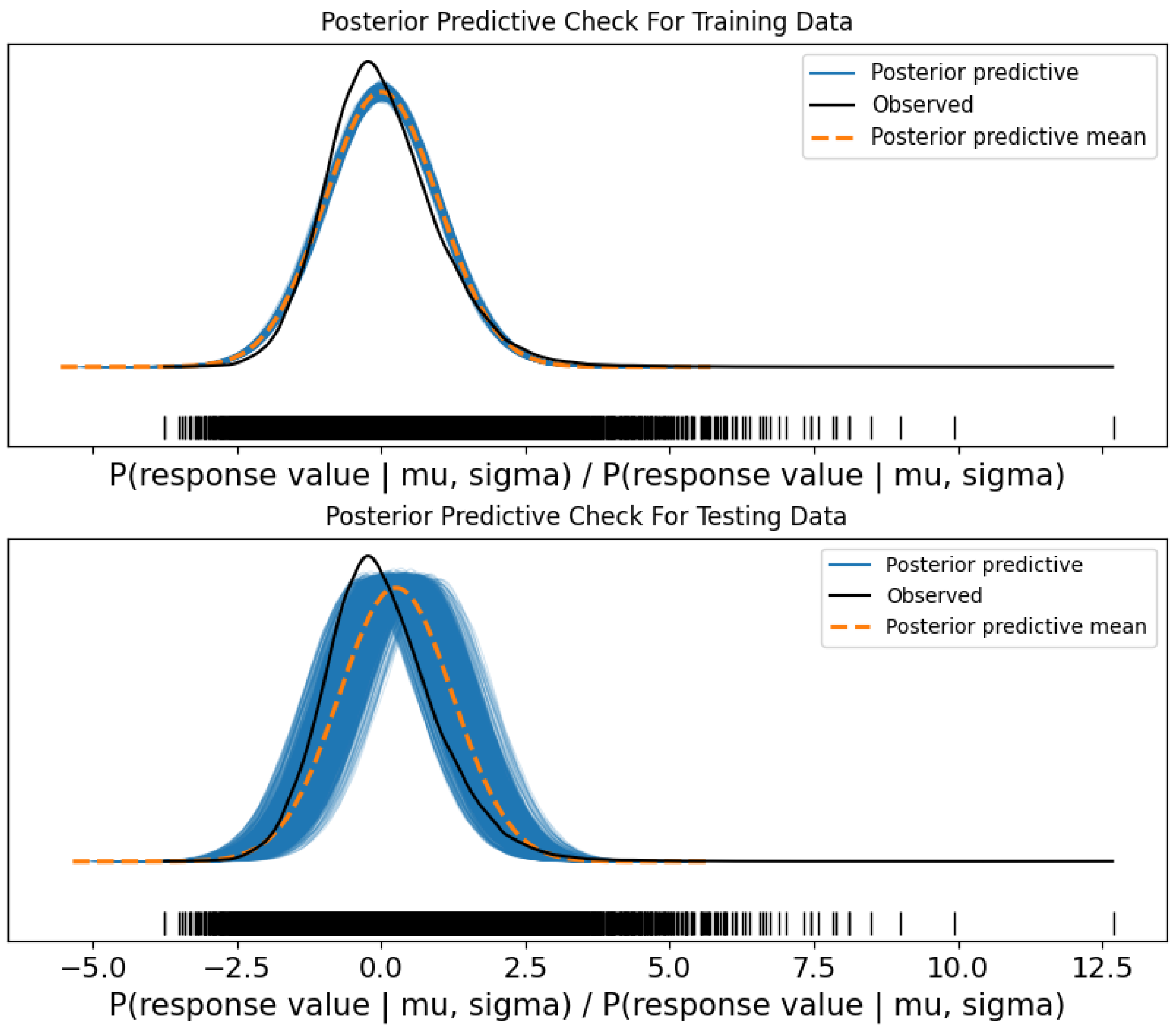


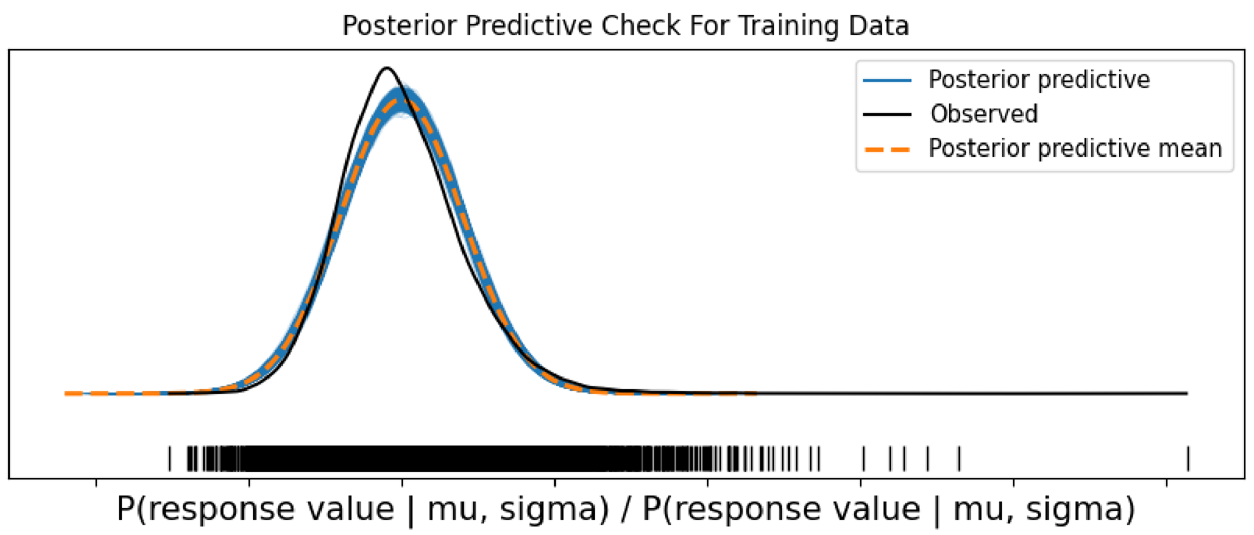
Plots of Posterior Predictive Check

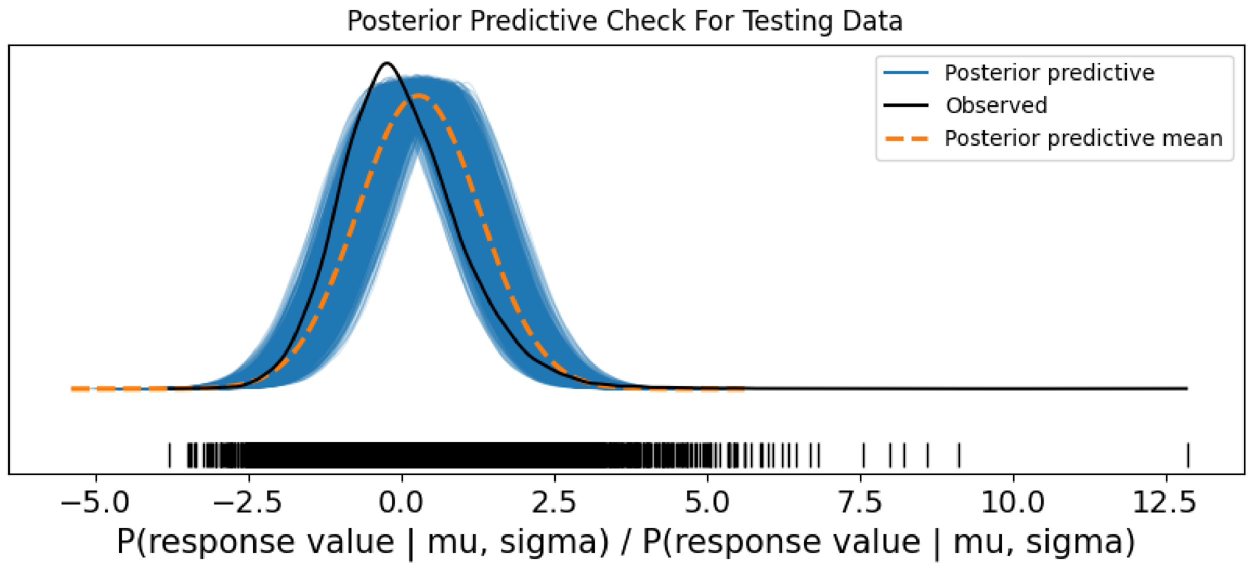
When we sample training and testing values of predicted docking score y-hat from posterior predictive probability distributions, we may generate a Plot Of Posterior Predictive Check. A Plot Of Posterior Predictive Check consists of a Rug Plot of observed docking scores along the horizontal axis, a black probability distribution of observed docking scores, one blue posterior predictive probability distribution of predicted docking scores for each chain, and an orange average posterior predictive probability distribution of predicted docking scores. The Plots Of Posterior Predictive Check displayed below correspond to training and testing our Bayesian

Model Using BART Model on 33,145 and 66,289 observations. The average posterior predictive probability distribution of predicted docking scores approximates the probability distribution of observed docking scores.

Perhaps our Bayesian Model Using BART Model performs better when the model is trained and tested on 66,289 observations than when the model is trained on 33,145 observations. The variances of the traces used to find the average posterior predictive probability distributions for our Bayesian Model Using BART Model when the model is tested are much higher than the variances for our Bayesian Neural Network (see below).

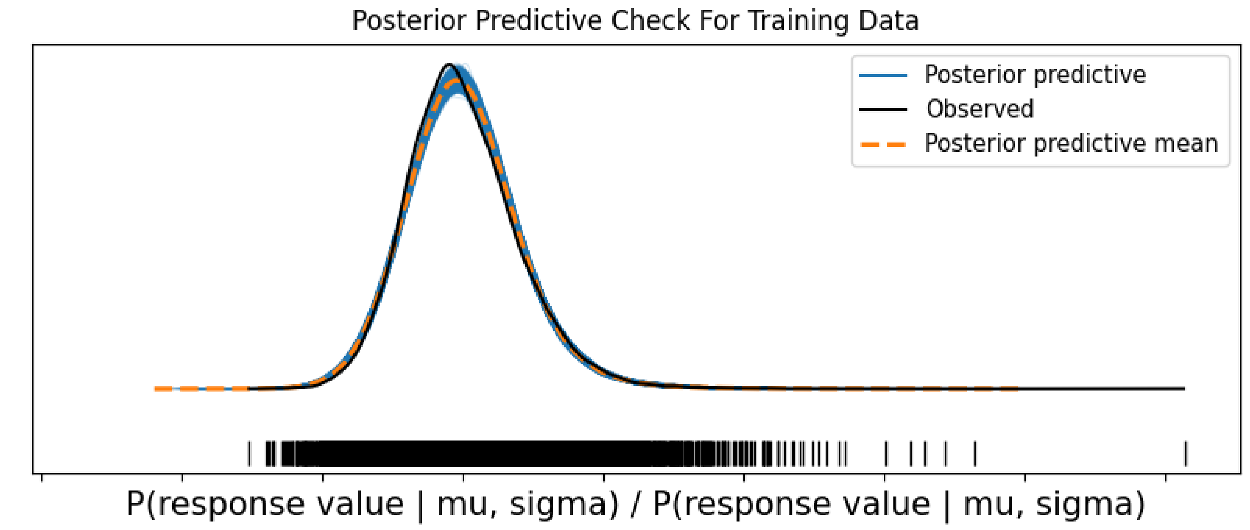


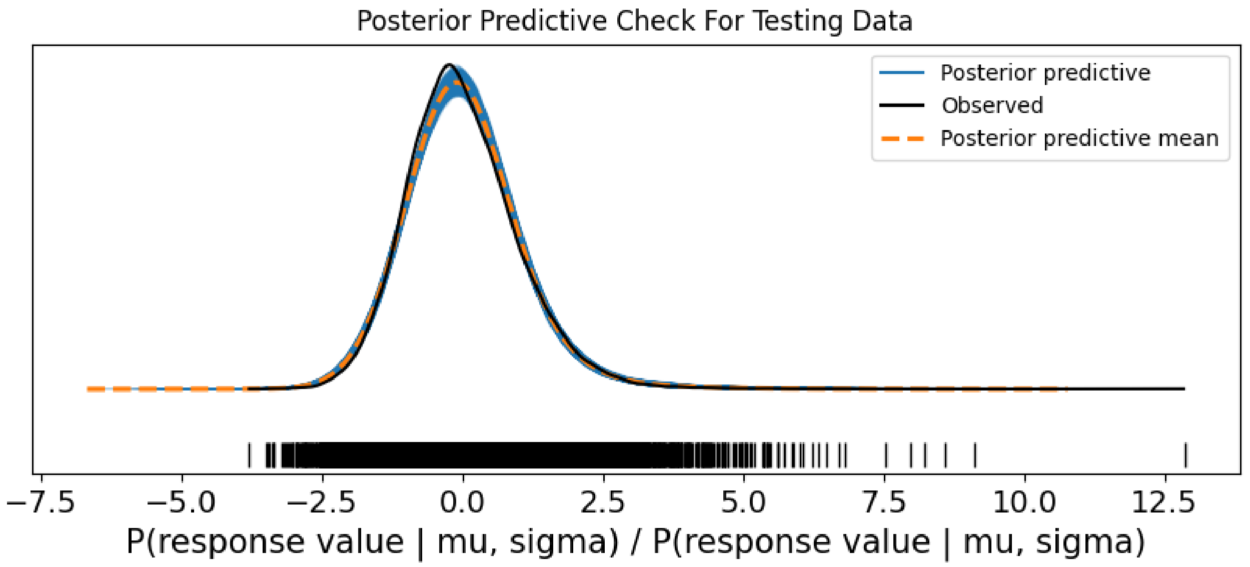


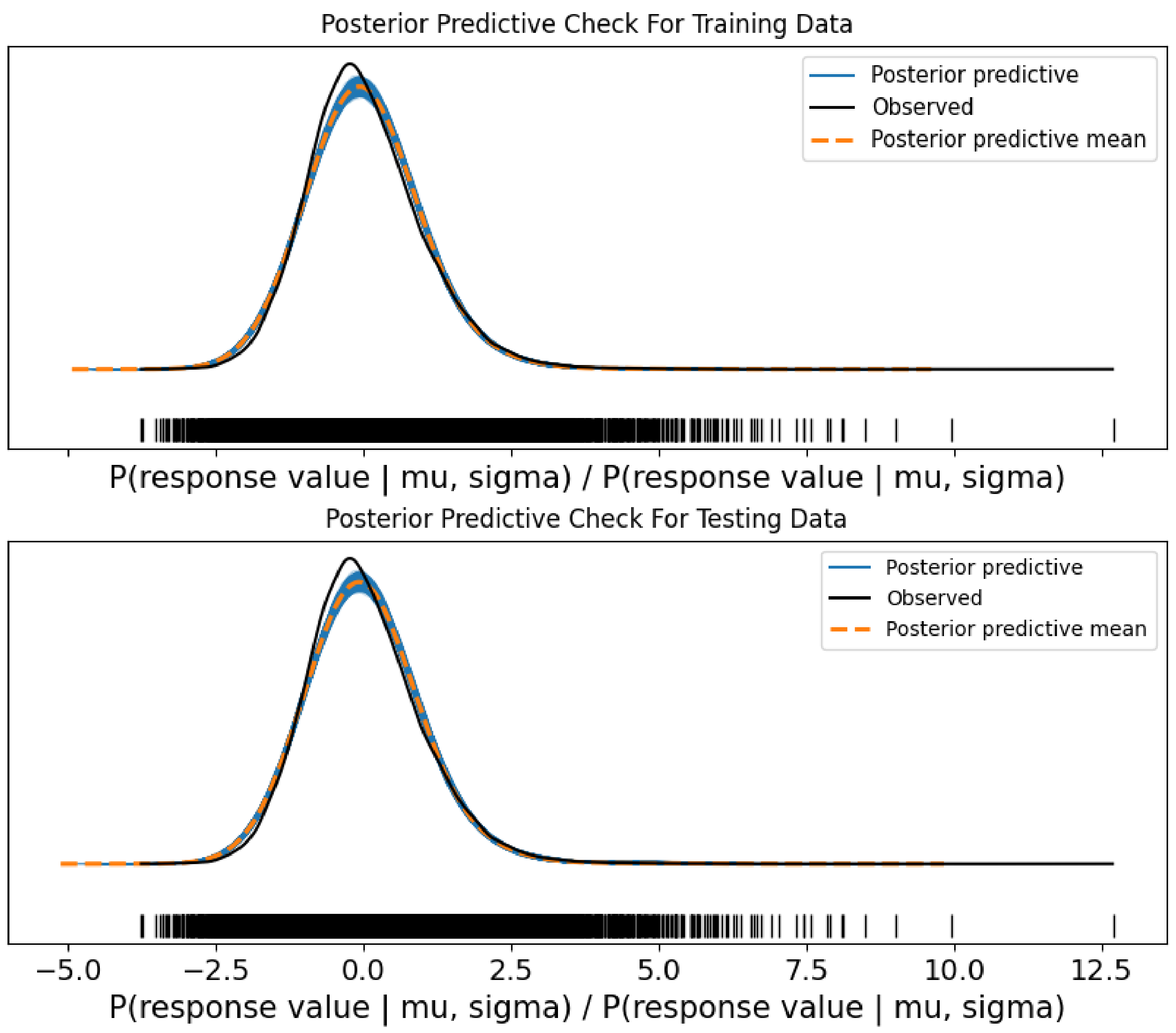


The Plots Of Posterior Predictive Check displayed below correspond to training and testing our BNN on 33,145, 66,289, and 135,577 observations. The average posterior predictive probability distribution of predicted docking scores approximates the probability distribution of observed docking scores.

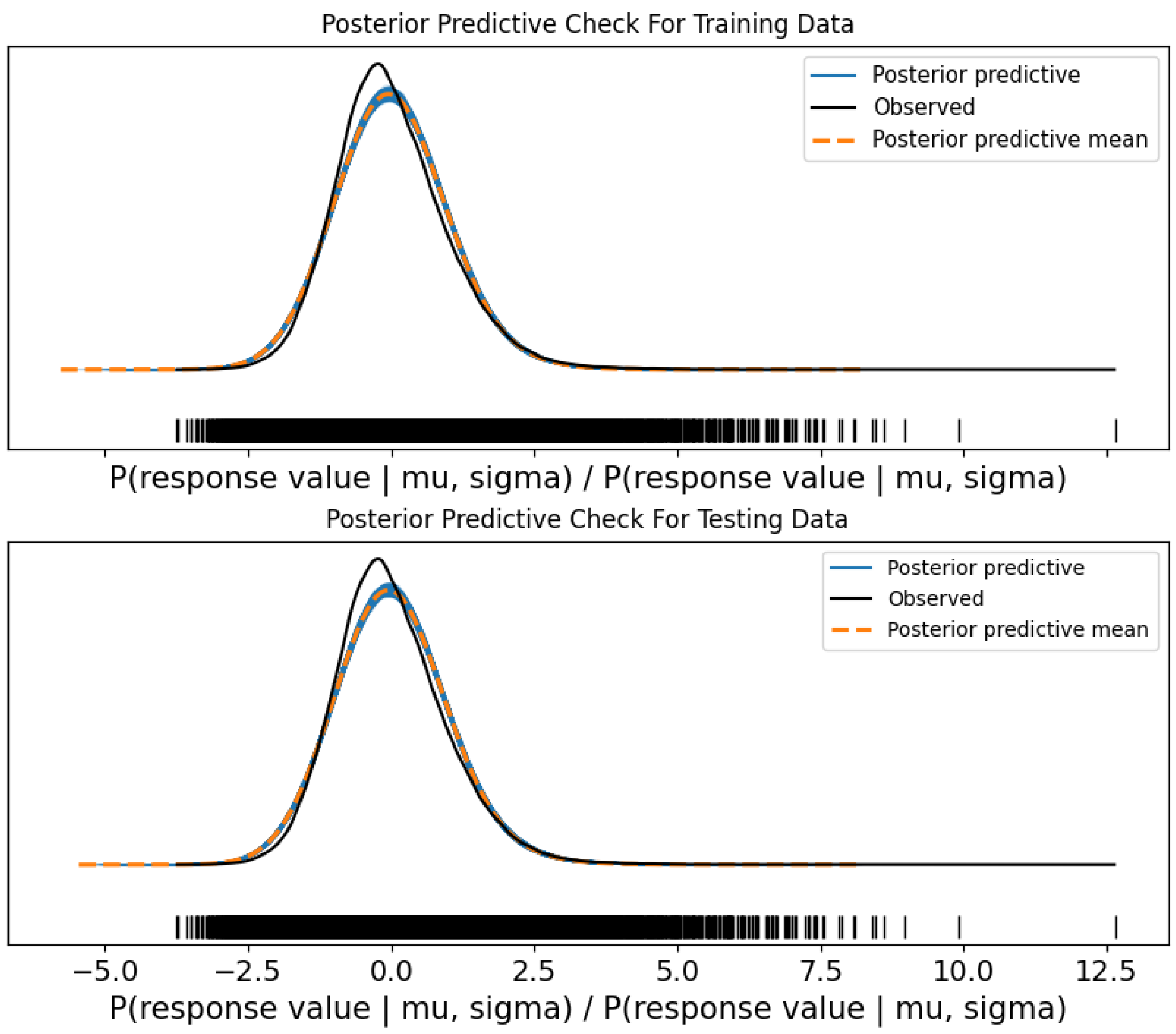
In accordance with our Trace Plots, the average posterior predictive probability distribution of predicted docking scores approximates the probability distribution of observed docking scores less well when our BNN is trained and tested on 132,577 observations than when our BNN is trained and tested on 66,289 observations. Our BNN performs best when the BNN is trained and tested on 33,145 observations.


Posterior Predictive Check For Training Data
Posterior predictive
Observed
Posterior predictive mean
P(response value | mu, sigma) / P(response value | mu, sigma)


Posterior Predictive Check For Testing Data
Posterior predictive
Observed
Posterior predictive mean
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
P(response value | mu, sigma) / P(response value | mu, sigma)



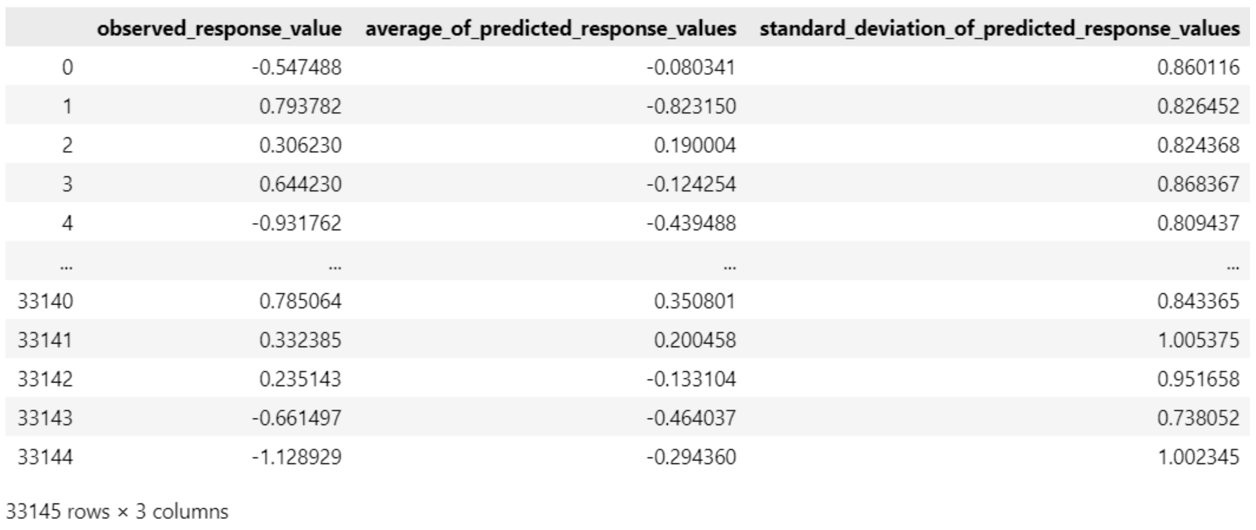
Posterior Predictive Check For Training Data
Posterior predictive
Observed
Posterior predictive mean
P(response value | mu, sigma) / P(response value | mu, sigma)
Posterior Predictive Check For Testing Data
Posterior predictive
Observed
Posterior predictive mean
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
P(response value | mu, sigma) / P(response value | mu, sigma)



It seems that the divergence of the average posterior predictive probability distribution for our Bayesian Model Using BART Model when the model is trained and tested on 66,289 observations from the observed probability distribution is on par with the divergence for our BNN when the model is trained on 132,577 observations. However, the variances of the traces used to find the average posterior predictive probability distributions for our Bayesian Model Using BART Model when the model is tested is far greater than those for our BNN.

Data Frames Of Observed Docking Scores And Averages And Standard Deviations Of Predicted Docking Scores

After we sample testing values of predicted docking scores y-hat from posterior predictive probability distributions, we may generate a data frame of observed docking scores and averages and standard deviations of docking scores predicted by a model based on numbers of occurrences of substructures. Such a data frame generated after training and testing our BNN on 33,145 observations is displayed below.



	observed_response_value	average_of_predicted_response_values	standard_deviation_of_predicted_response_values
0	-0.547488	-0.080341	0.860116
1	0.793782	-0.823150	0.826452
2	0.306230	0.190004	0.824368
3	0.644230	-0.124254	0.868367
4	-0.931762	-0.439488	0.809437
...	...	...	...
33140	0.785064	0.350801	0.843365
33141	0.332385	0.200458	1.005375
33142	0.235143	-0.133104	0.951658
33143	-0.661497	-0.464037	0.738052
33144	-1.128929	-0.294360	1.002345

33145 rows × 3 columns

Gains Curves

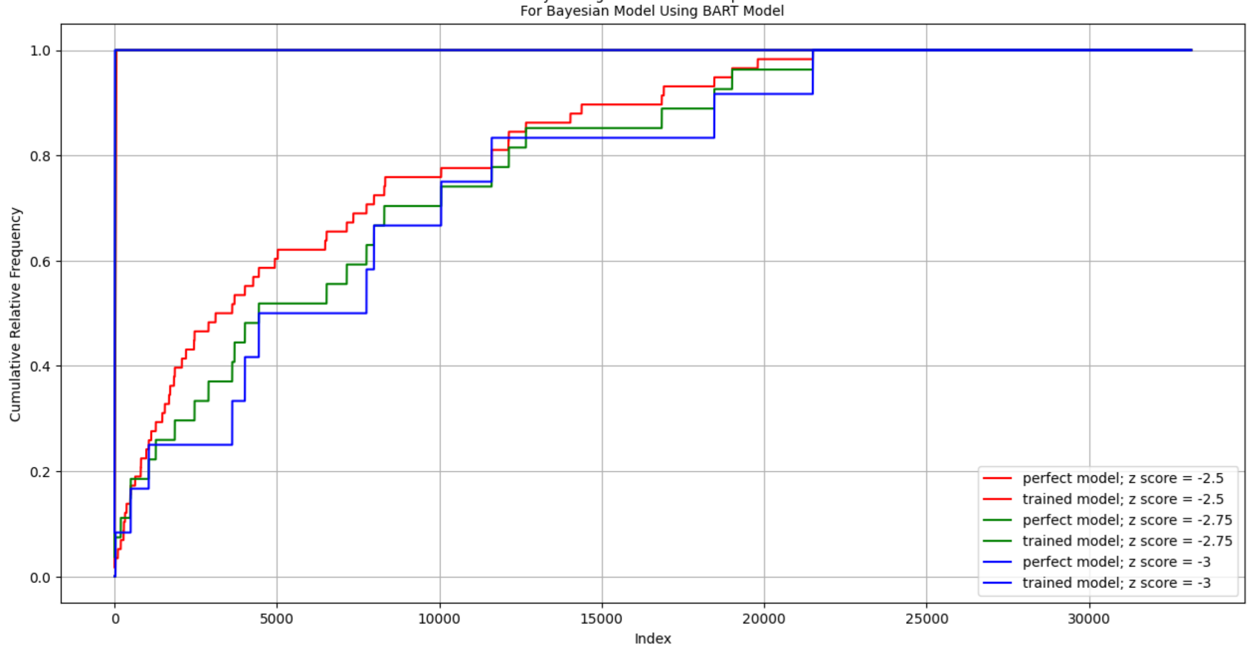
After we generate a data frame of observed docking scores and averages and standard deviations of docking scores predicted based on numbers of occurrences of substructures, we may generate a Plot Of Gains Curves. A Plot Of Gains Curves is also called an Enrichment-Factor Plot. A Gains Curve is a plot of

cumulative relative frequency of positive indicators
that an observed docking score is below a threshold determined by a user-defined *z* score
vs. index in a data frame of indicators
sorted by a column of averages of predicted response values
for a model.

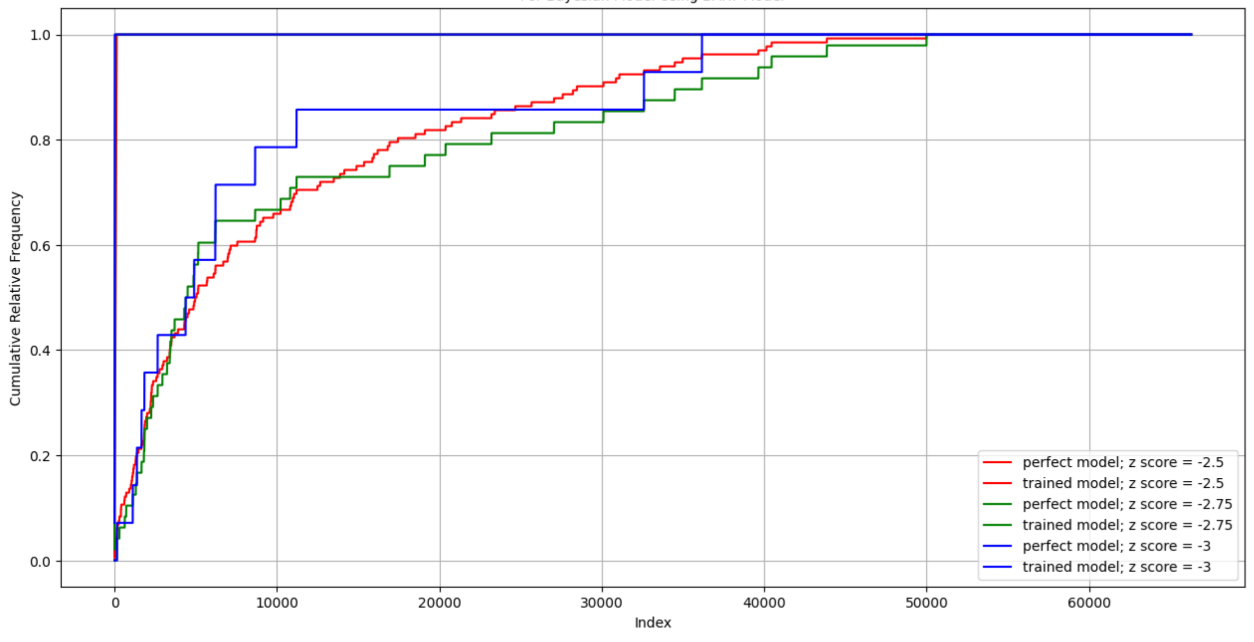
For a small proportion of positive indicators, a Gains Curve resembles a Receiver Operator Characteristic (ROC) Curve. For a large proportion of positive indicators, the upper left corner of a Gains Curve for a perfect model is shifted right. Models may be compared by comparing their Gains Curves in a manner similar to comparing models according to their ROC Curves.

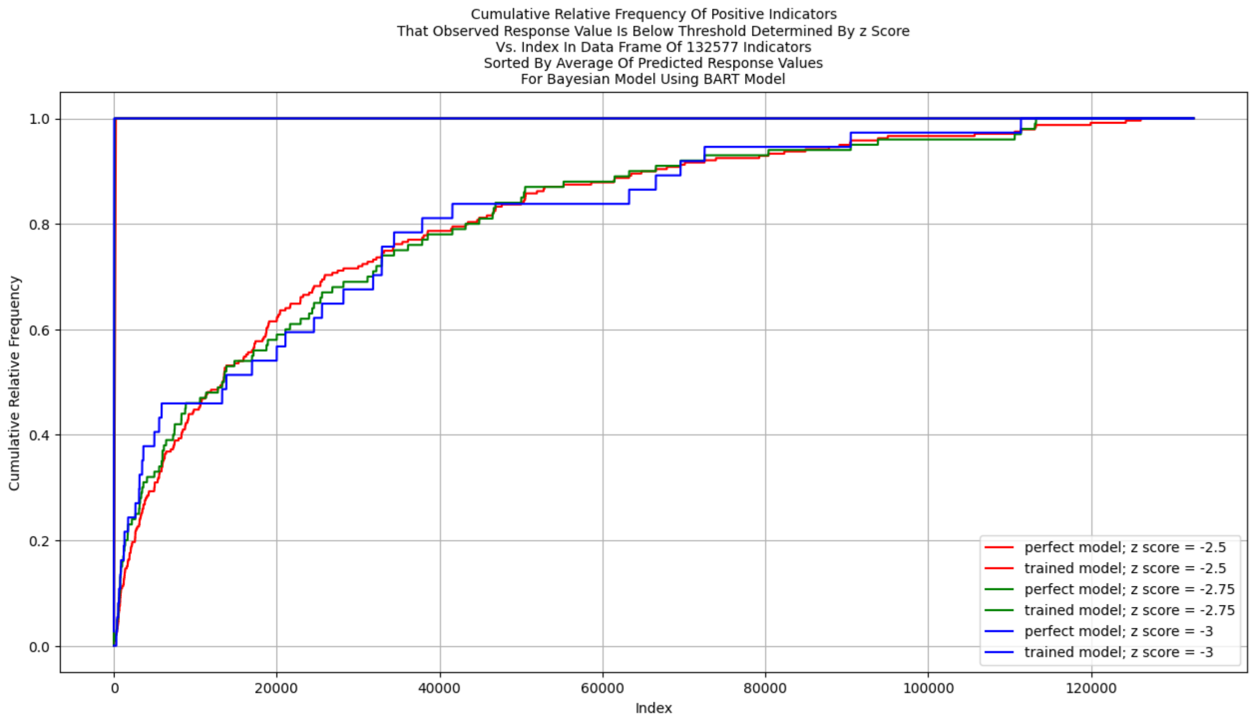
The Gains Curves for 33,145, 66,289, and 132,577 observations and our *Bayesian Model Using BART Model* are presented below. Our Bayesian Model Using BART Model with 66,289 observations performs best.

Cumulative Relative Frequency Of Positive Indicators
That Observed Response Value Is Below Threshold Determined By z Score
Vs. Index In Data Frame Of 33145 Indicators
Sorted By Average Of Predicted Response Values
For Bayesian Model Using BART Model

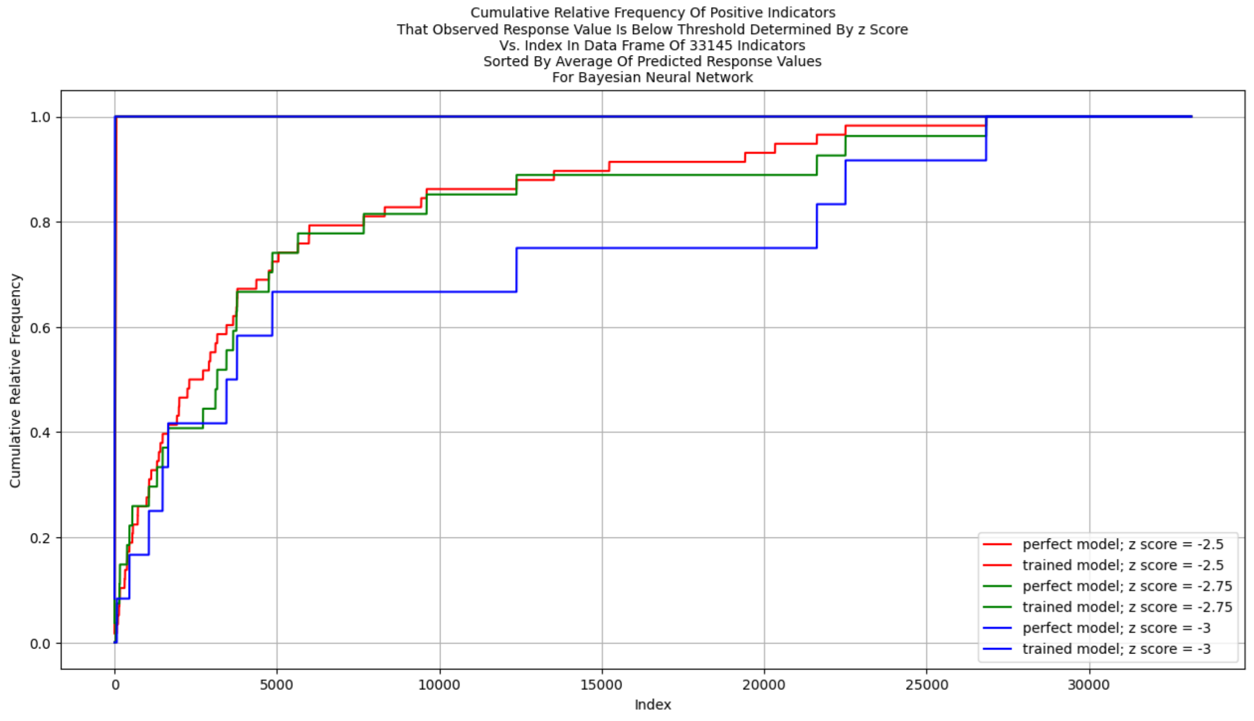


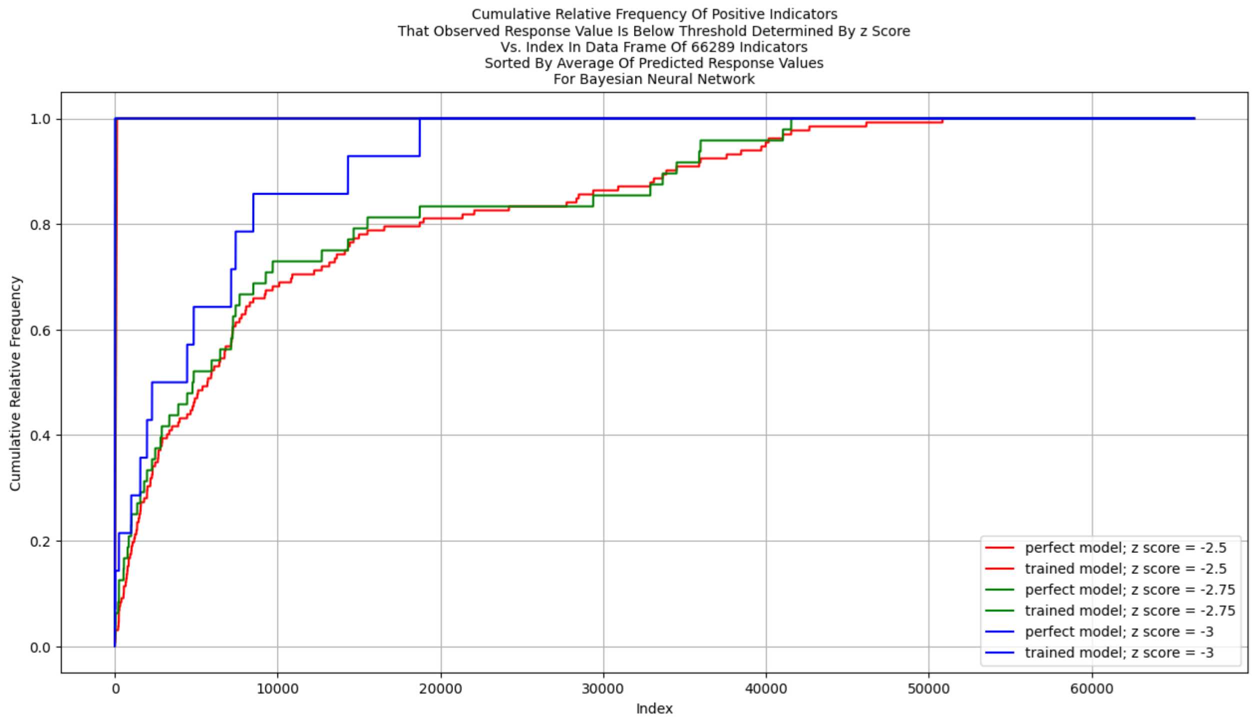
Cumulative Relative Frequency Of Positive Indicators
That Observed Response Value Is Below Threshold Determined By z Score
Vs. Index In Data Frame Of 66289 Indicators
Sorted By Average Of Predicted Response Values
For Bayesian Model Using BART Model

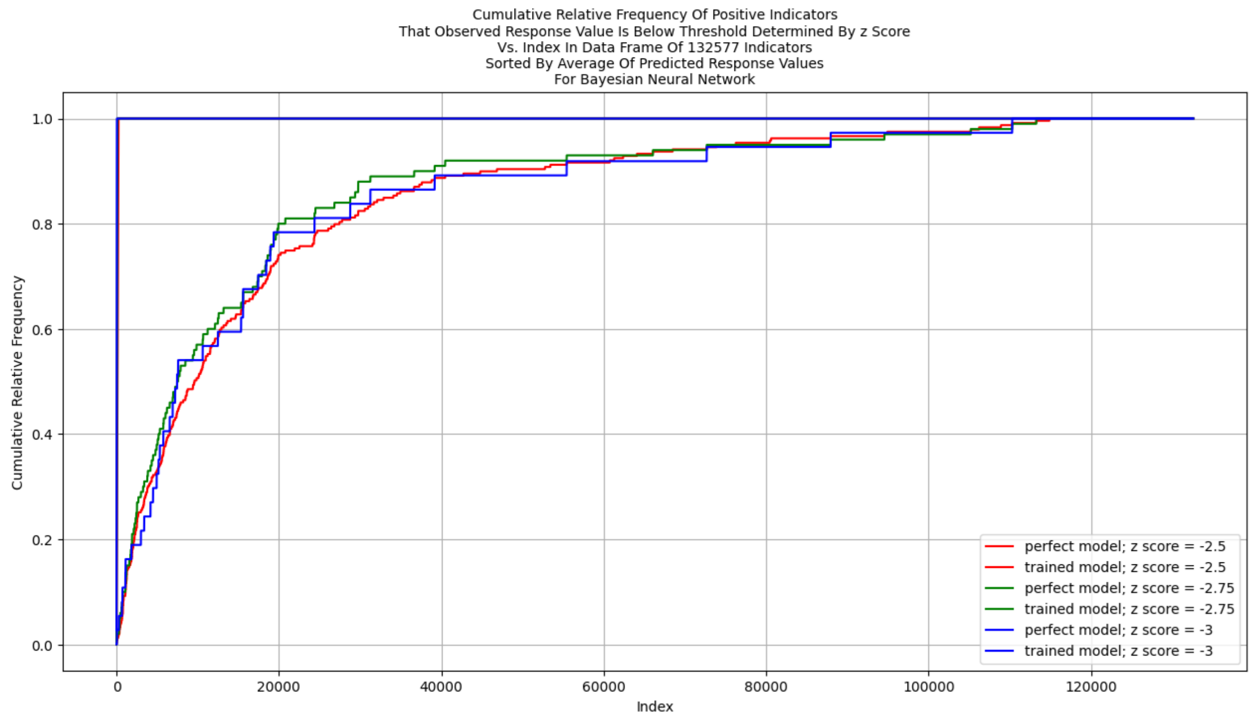




The Gains Curves for 33,145, 66,289, and 132,577 observations and our BNN are presented below. In accordance with our Trace Plots and Plots Of Posterior Predictive Check, for a user-defined *z* score of -3, our BNN performs best when trained and tested on 66,289 observations. For a user-defined z score of -2.5 or -2.75, our BNN performs best when trained and tested on 132,577 observations.







For a user-defined z score of -2.5 or -2.75, our BNN performs better than our *Bayesian Model Using BART Model* when each is trained and tested on 33,145 observations. For a z score of -3, our *Bayesian Model Using BART Model* performs better.

For a user-defined z score of -2.75 or -3, our BNN performs better than our *Bayesian Model Using BART Model* when each is trained and tested on 66,289 observations. For a z score of -2.5, our *Bayesian Model Using BART Model* performs better.

Considering the upper left points of the Gains Curves for our BNN and *Bayesian Model Using BART Model* when each is trained and tested on 132,577 observations, our BNN performs better.

It seems our BNN performs better than our Bayesian Model Using BART Model most of the time. Both models perform better when trained on more observations. Both our BNN and our Bayesian Model Using BART Model may be expanded and tuned.

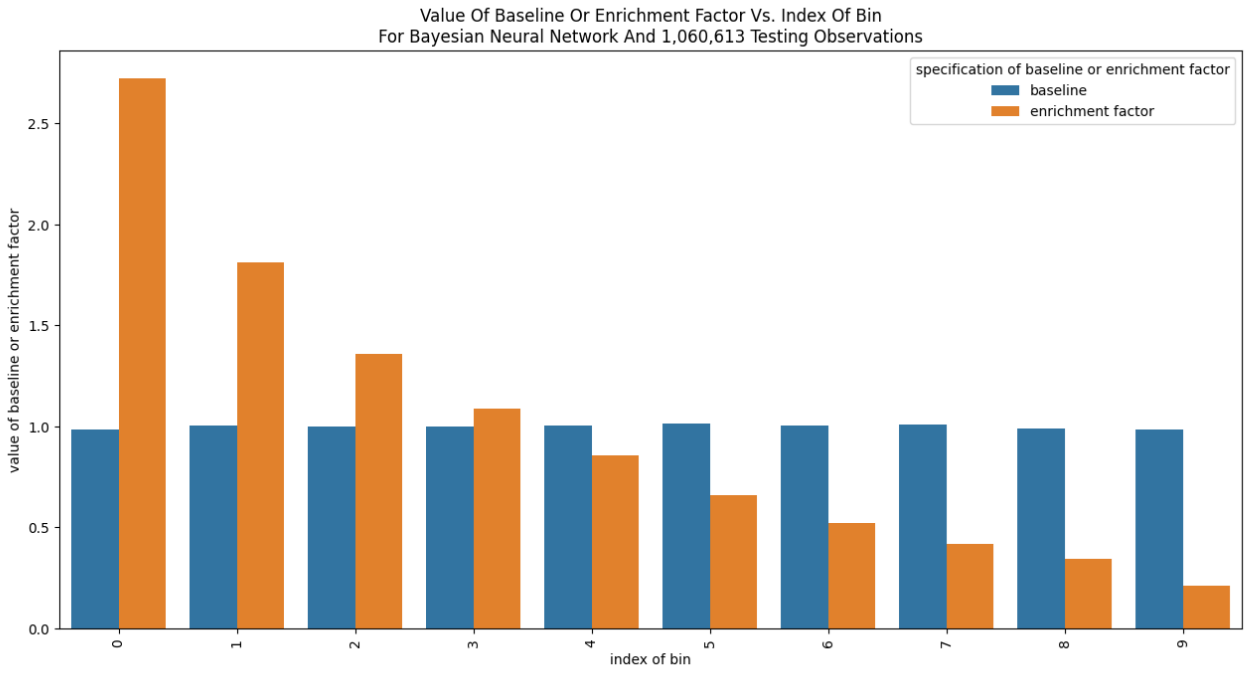
Decile-Wise Lift Charts

Before developing Gains Curves, we considered Decile-Wise Lift Charts. Below is *Value Of Baseline Or Enrichment Factor Vs. Index Of Bin For Bayesian Neural Network And 1,060,613 Testing Observations*. A Decile-Wise Lift Chart may be thought of as the derivative of a Gains Curve. *Value Of Baseline Vs. Index Of Bin* represents the rate of change of the Gains Curve for a model that predicts randomly. *Enrichment Factor Vs. Index Of Bin* represents the rate of change of the Gains Curve for our BNN when the BNN is trained on 1,060,613 observations.

The below Decile-Wise Lift Chart was created based on *Data Frame Of 1,060,613 Observed Docking Scores And Averages And Standard Deviations Of Docking Scores Predicted By BNN Based On Numbers Of Occurrences Of Substructures*. A bin is a group of adjacent observed docking scores in the above data frame. A value of baseline or an enrichment factor is computed for each of 10 adjacent bins as

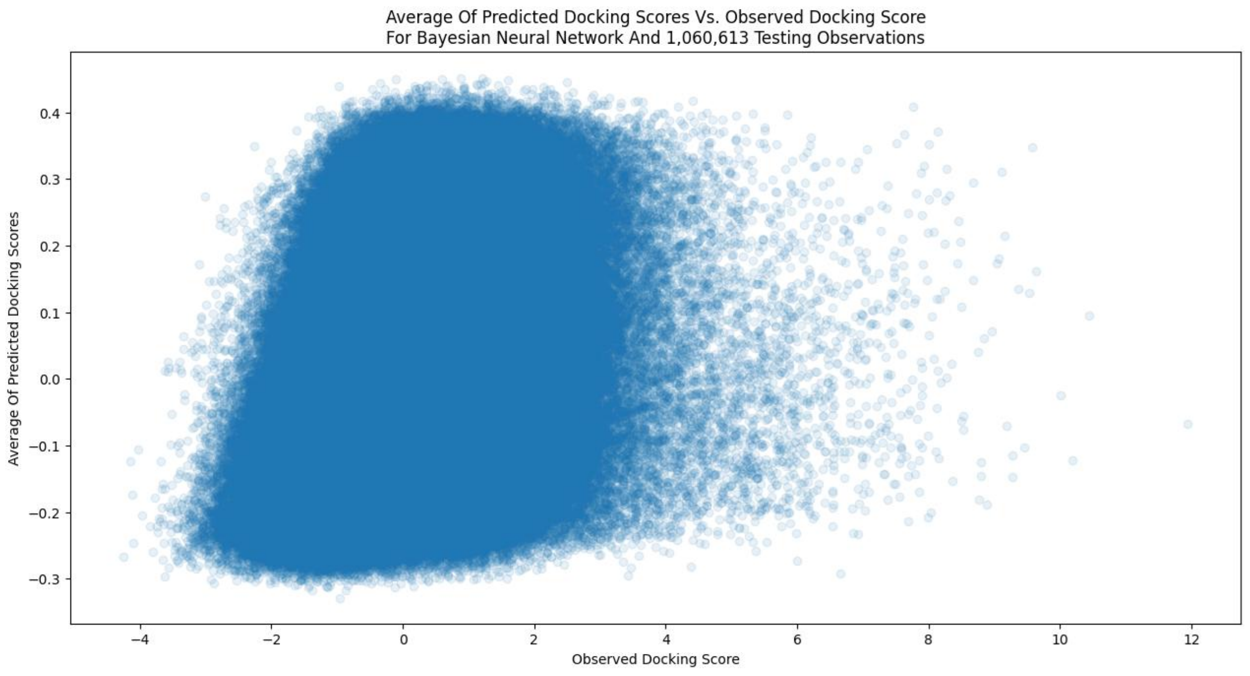
$$\frac{proportion\ of\ observed\ docking\ scores\ in\ bin\ in\ lowest\ 10\ percent}{proportion\ of\ observed\ docking\ scores\ in\ bin}$$

In computing a value of baseline, the rows in the above data frame are randomized. In computing an enrichment factor, the rows in the above data frame are sorted in ascending order by average of predicted docking scores.



Density Plots

In tandem with considering Decile-Wise Lift Charts, we considered Density Plots. Below is *Average Of Predicted Docking Scores Vs. Observed Docking Score For Bayesian Neural Network And 1,060,613 Testing Observations*. There is a low correlation between average of predicted docking scores and observed docking score.



Conclusion

Key Findings

Because our mean posterior predictive probability density distributions of docking scores approach our observed distribution of docking scores, and our Gains curves approach the upper-left corners of their plots, our models are useful in predicting docking scores based on numbers of occurrences of substructures.

Our Bayesian Neural Network may perform best with more observations. Both our *Bayesian Model Using BART Model* and our BNN may be expanded and tuned. All that being said, it seems that there is a low correlation between average of predicted docking scores and observed docking score, and that the range of averages is small compared with the range of observed docking scores.

Improvements

Our models may be improved if they are trained and tested on a group of observations where the docking scores of that group are uniformly distributed.

Our Bayesian Docking-Score Predictor may be improved through iterative screening. Bayesian Docking-Score Predictor could be trained on n (e.g., 100,000) pairs of SMILES and docking score and used to predict docking scores of ligands based on many (e.g., 1,900,000) other SMILES's. Rows of data representing docking of ligands could be sorted in ascending order by predicted docking score. The first n rows of data corresponding to the lowest predicted docking scores could be added to the training dataset.

Applications

Bayesian Docking-Score Predictor may be used to predict, to estimate uncertainty, or in optimization of hyperparameters of other models. A trained Bayesian Docking-Score Predictor could be used for developing a training data set of SMILES's and docking scores of ligands for another predictor.

Appendix 1: Exploring Change In Molar Gibbs Free Energy

The enthalpy H of a compound is the sum of the compound's internal energy U and the pressure energy E_p of the compound. The pressure energy is the energy required to establish the compound's physical dimensions. The pressure energy is the product of the pressure P on the compound by its surroundings and the volume V of the compound.

$$H = U + E_\mathrm{p}$$

The molar internal energy u of a compound is the internal energy of the compound per mole.
The molar volume v is the volume of the compound per mole.
The molar pressure energy e_p of a compound is the pressure energy of the compound per mole.

The molar enthalpy h of a compound is the enthalpy of the compound per mole. The molar enthalpy of a compound is the sum of the compound's molar internal energy and molar pressure energy.

$$h = u + e_{\mathrm{p}}$$

The change in enthalpy ΔH of a reaction is the difference between the sum of the enthalpies of the products of the reaction and the sum of the enthalpies of the reactants.

$$\Delta H = \sum_{i=1}^{number\ of\ products} [H_i] - \sum_{i=1}^{number\ of\ reactants} [H_i]$$

The change in molar enthalpy Δh of a reaction is the difference between the sum of the molar enthalpies of the products and the sum of the molar enthalpies of the reactants.

$$\Delta h = \sum_{i=1}^{number\ of\ products} [h_i] - \sum_{i=1}^{number\ of\ reactants} [h_i]$$

The entropy of a compound is a measure of uncertainty, disorder, or mixedupness of the compound. The entropy measures the degree to which the probability of the compound being in a particular microstate is spread out over different microstates. A microstate specifies all molecular details about the system including the position and velocity of every molecule. The more such states are available to the compound with appreciable probability, the greater the entropy.

$$S = -k_B \sum_{i=1}^{number\ of\ possible\ microstates} [p_i ln(p_i)] = k_B ln(\Omega)$$

p_i is the probability that the compound is in the ith state according to the Boltzmann distribution.
Ω is the number of microstates whose energy equals the compound's energy.
For an isolated system, $p_i = \frac{1}{\Omega}$.
The entropy S of a compound is a quantity that satisfies "an infinitesimal change in entropy dS is equal to the ratio of an infinitesimal quantity of heat in a reversible reaction and the temperature of the compound".
The molar entropy s of a compound is the entropy S of the compound per mole.

The change in entropy ΔS of a reaction is the difference between the sum of the entropies of the products of the reaction and the sum of the entropies of the reactants.

$$\Delta S = \sum_{i=1}^{number\ of\ products} [S_i] - \sum_{i=1}^{number\ of\ reactants} [S_i]$$

The change in molar entropy Δs of a reaction is the difference between the sum of the molar entropies of the products of the reaction and the sum of the molar entropies of the reactants.

$$\Delta s = \sum_{i=1}^{number\ of\ products} [s_i] - \sum_{i=1}^{number\ of\ reactants} [s_i]$$

The Gibbs free energy of a compound at some time is the difference between the enthalpy H of the compound and the product of the temperature T and entropy S of the compound.

$$G = H - TS$$

The molar Gibbs free energy is the difference between the molar enthalpy of the compound and the product of the temperature T and the molar entropy of the compound.

$$g = h - Ts$$

The change in Gibbs free energy of a reaction is the maximum amount of free and useful energy available to do non-volume-expansion work that can be extracted from the reactants at fixed temperature and pressure, which can be attained only in a completely reversible process. The change in Gibbs free energy of a reaction is the difference between the change in enthalpy of the reaction and the change in the product of the temperature and entropy of the compound.

$$\Delta G = \Delta(H - TS) = \Delta H - \Delta(TS)$$

The change in molar Gibbs free energy of a compound is the difference between the change in molar enthalpy of the compound and the change in the product of the temperature and molar entropy of the compound.

$$\Delta g = \Delta(h - Ts) = \Delta h - \Delta(Ts)$$