**Bayesian Docking-Score Predictor**

Created: 11/15/2023 by Tom Lever

Updated: 11/18/2023 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 in the molar Gibbs free energy of a compound. A docking score is measured in kilocalories per mole. According to Dr. Ryan Weil, is amazing and is okay. See below *Appendix 1: Exploring Change In Molar Gibbs Free Energy*.

**Data**

Our data set is *Data\_Frame\_Of\_SMILESs\_Docking\_Scores\_And\_Other\_Data.csv* (DFSDSAOD). 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. Each row has an Entry ID. There is one Entry ID per ligand. There is one ligand per Entry ID.

**Implementation**

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 construct a feature matrix where each row contains a docking score and 1024 numbers of occurrences of substructures. Our feature matrix has 1,060,613 observations.

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

where is an error. We assume that is normally distributed with mean and standard deviation .

Thus, is normally distributed with mean and standard deviation .

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

In our code we define a matrix of all training values of , a tensor variable representing , a vector of training values of , a tensor variable representing a prior probability density distribution that is half normal with standard deviation , and a tensor variable representing likelihood .

We use pymc to sample testing values of from the posterior predictive probability density distribution

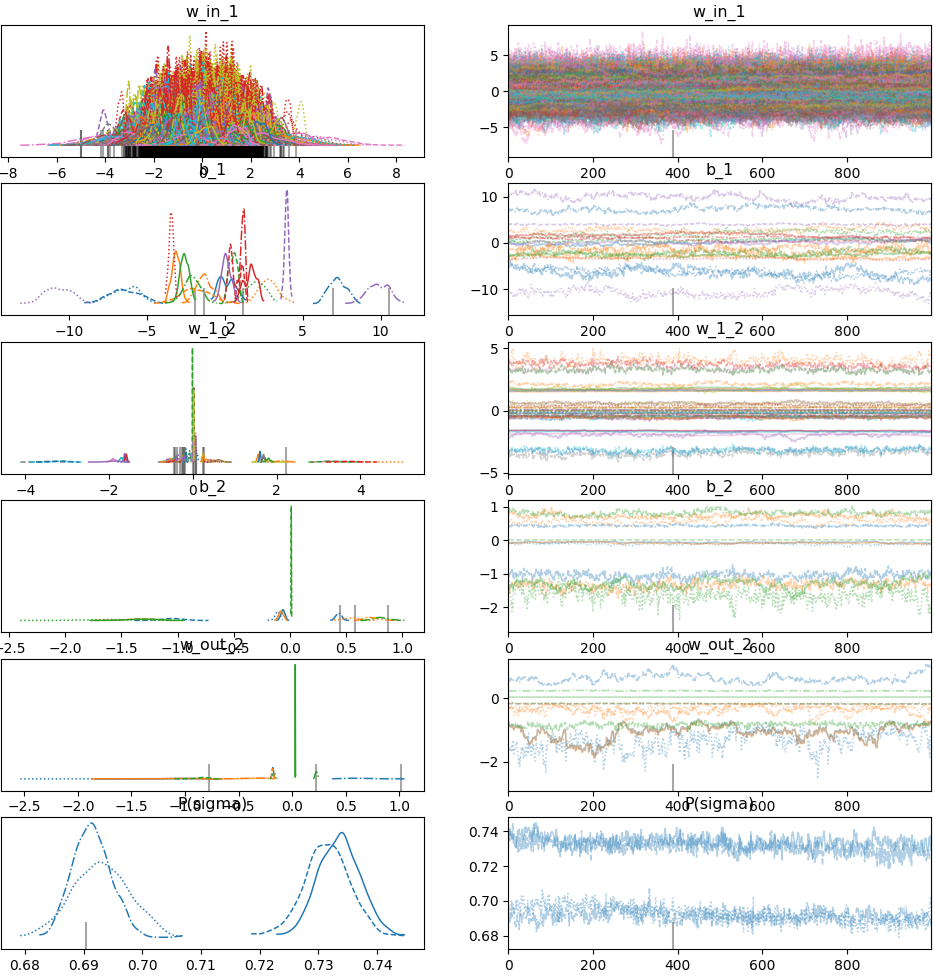
The number of testing values of sampled is the product of a number of chains (e.g., ), a number of samples per chain (e.g., ), and a number of testing observations (e.g., ). 4 chains were sampled by default. We find vectors of averages and standard deviations of testing values of sampled. Each vector has length equal to the number of testing observations.

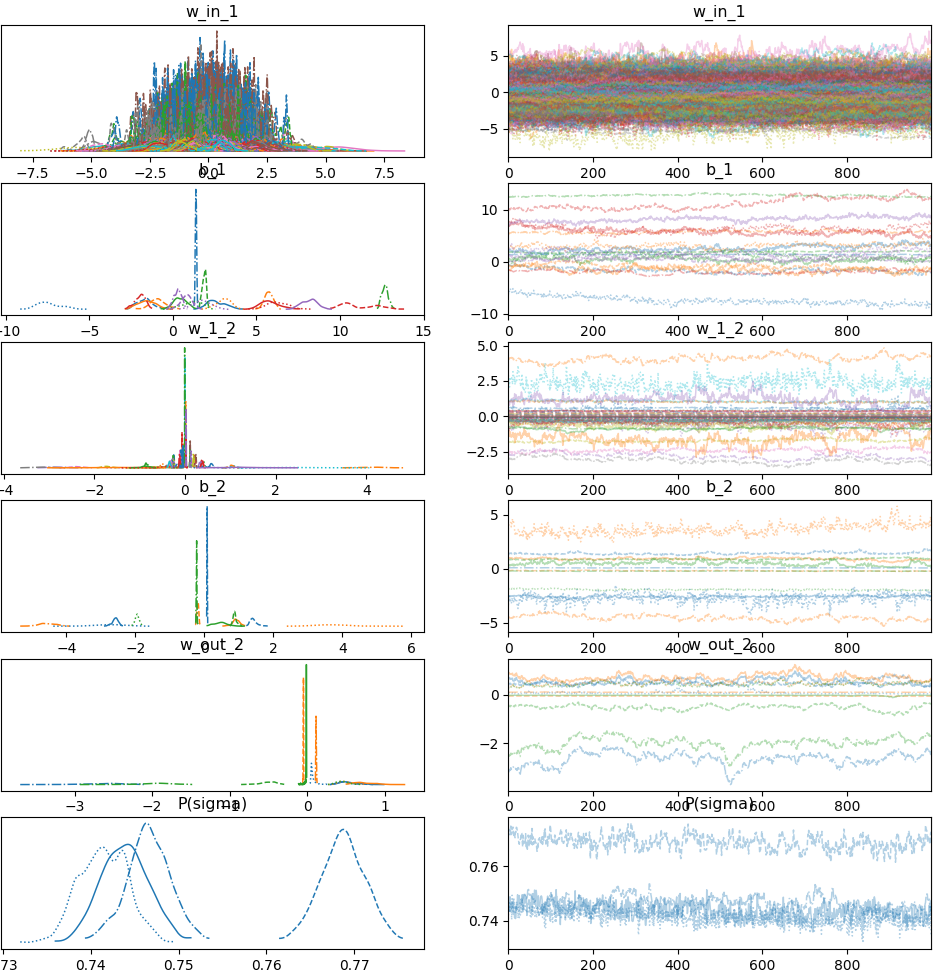
As a side note, in the original paper *BART: Bayesian Additive Regression Trees*, “BART’s many features are illustrated with a bake-off against competing methods of different datasets, with a simulation experiment and on a drug discovery classification problem.” According to *Statistical Learning: 8.6 Bayesian Additive Regression Trees*, “It turns out that the 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 posterior 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] and [number of trees per iteration] , and a moderate value for [number of burn-in iterations] : for instance, , , and are reasonable choices. BART has been shown to have impressive out-of-box performance – that is, it performs well with minimal tuning.”

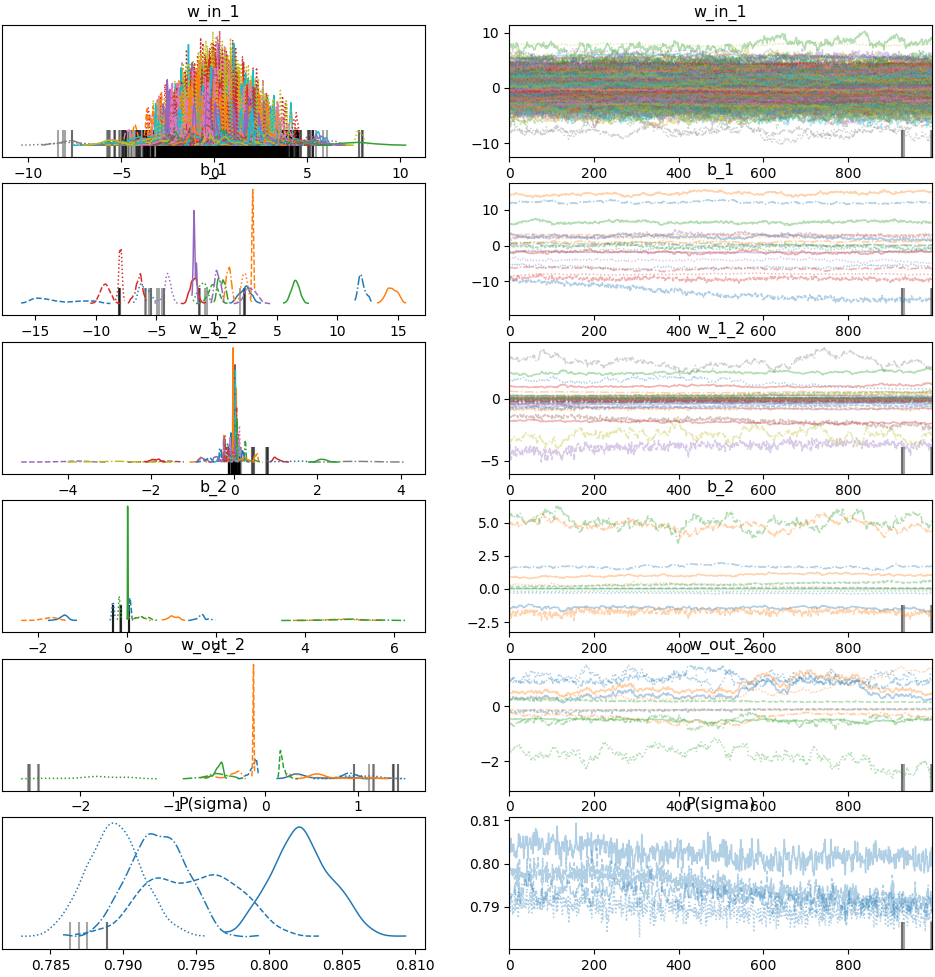
We also train and predict with a Bayesian Neural Network (BNN). In this model, a neural network is trained to map a vector of random variables to a response variable . is multiplied by the range of observed response values . Our BNN has:

1. An input layer through which the BNN receives an input matrix
2. A first hidden layer with:
   1. An initial weights matrix of random numbers sampled from the standard normal distribution
   2. A normal distribution from which weight matrices are sampled. The normal distribution has shape , mean , and standard deviation .
   3. An initial bias vector of random numbers sampled from the standard normal distribution. Biases are added to each row of .
   4. A normal distribution from which weight matrices are sampled. The normal distribution has shape , mean , and standard deviation .
   5. A hyperbolic-tangent activation function
3. A second hidden layer with:
   1. An initial weights matrix of random numbers sampled from the standard normal distribution
   2. A normal distribution from which weight matrices are sampled. The normal distribution has shape , mean , and standard deviation .
   3. An initial bias vector of random numbers sampled from the standard normal distribution
   4. A normal distribution from which weight matrices are sampled. The normal distribution has shape , mean , and standard deviation .
   5. A hyperbolic-tangent activation function
4. An output layer with:
   1. An initial weights matrix of random numbers sampled from the standard normal distribution
   2. A normal distribution from which weight matrices are sampled. The normal distribution has shape , mean , and standard deviation .

When we estimate a joint posterior probability density distribution for training data, we may generate a Trace Plot of distributions each of values of a parameter in a parameter vector estimated as No U-Turn Sampling (NUTS) progress. Below we visualize such distributions 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.

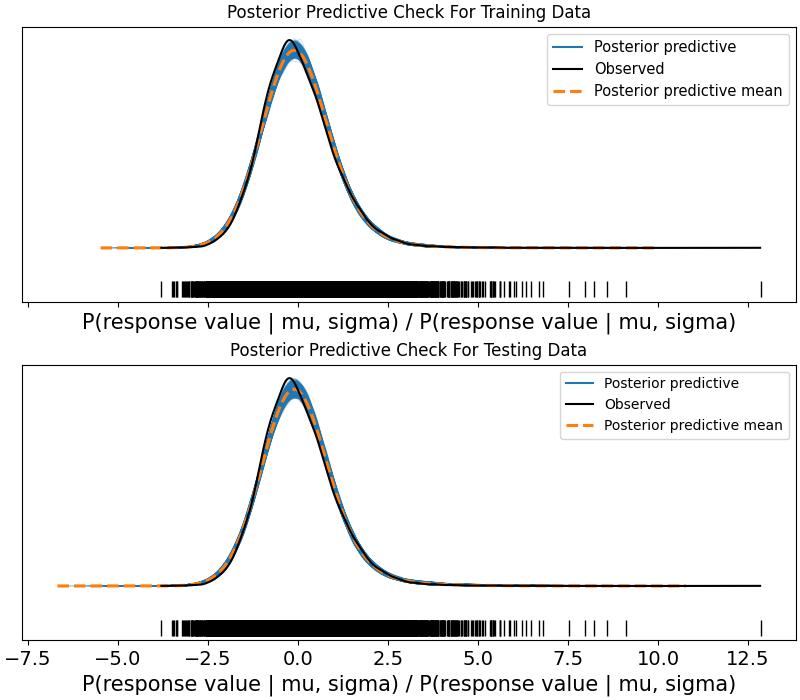


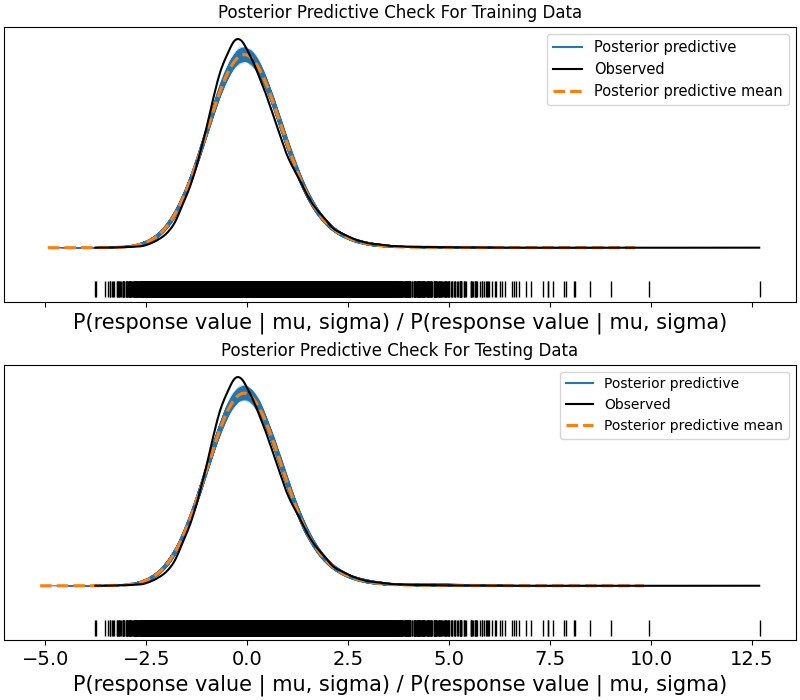


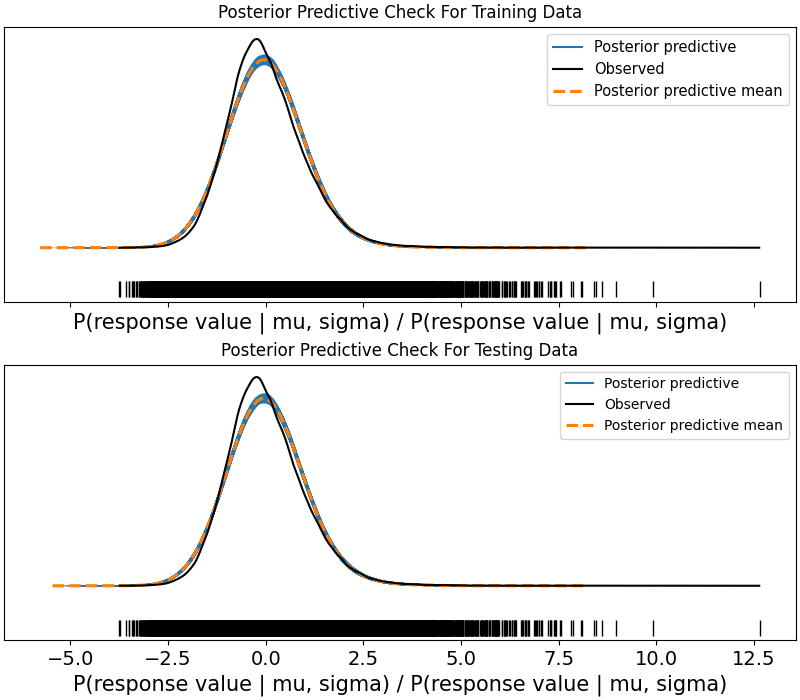


When we sample training and testing values of from posterior predictive probability density distributions for our BNN, we may generate a Plot Of Posterior Predictive Check. A Plot Of Posterior Predictive Check consists of a Rug Plot of observed docking scores along the horizontal axis, a black probability density distribution of observed docking scores, one blue probability density distribution of predicted docking scores for each chain, and an orange average probability density distribution of predicted docking scores. The Plots Of Posterior Predictive Check displayed below corresponds to training and testing our BNN on 33,145, 66,289, and 135,577 observations. The average probability density distribution of predicted docking scores approximates the probability density distribution of observed docking scores.

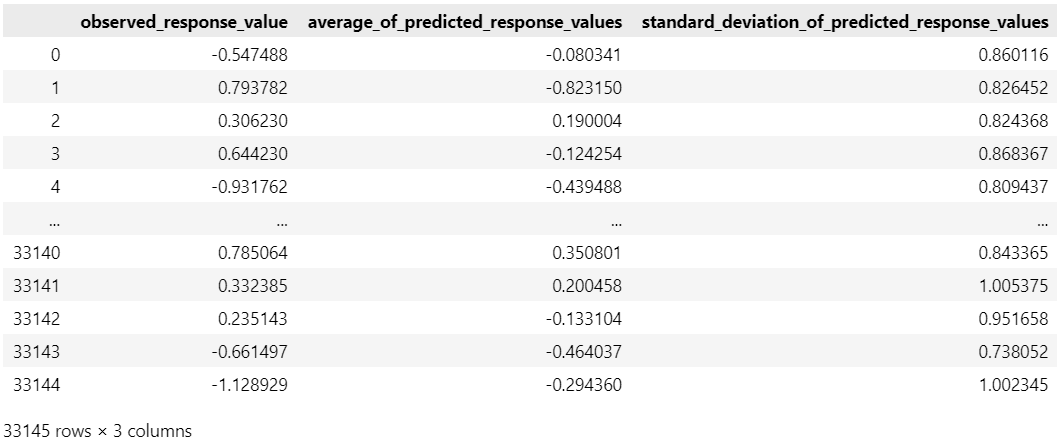
In accordance with our Trace Plots, the average probability density distribution of predicted docking scores approximates the probability density distribution of observed docking scores less well when our BNN is trained and tested on 132,577 observations than when our BNN is trained and tested on 66,289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

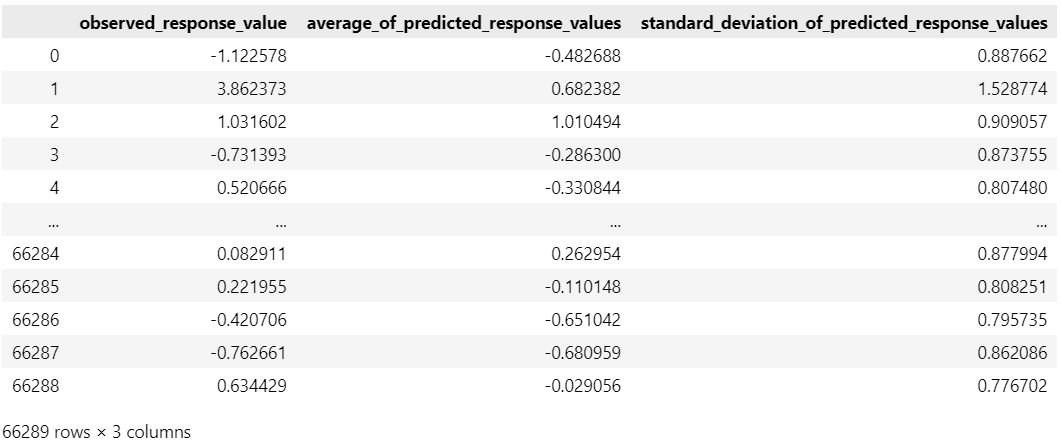


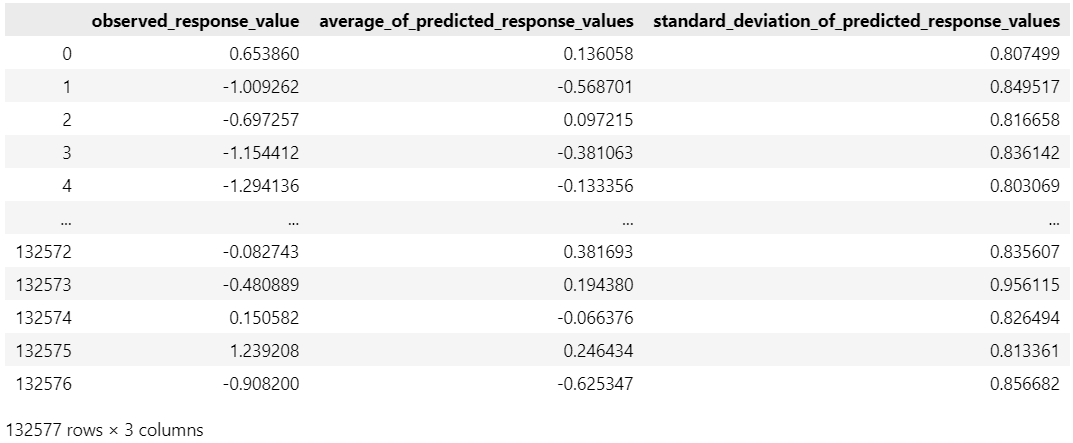




After we sample testing values of from posterior predictive probability density distributions for our BNN, we may generate a data frame of observed docking scores and averages and standard deviations of docking scores predicted by the BNN based on numbers of occurrences of substructures. Such data frames generated after training and testing our BNN on 33,145, 66,289, and 132,577 observations are displayed below.







After we generate a data frame of observed docking scores and averages of docking scores predicted by our BNN based on numbers of occurrences of substructures, we may generate a Plot Of Gains Curves / 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

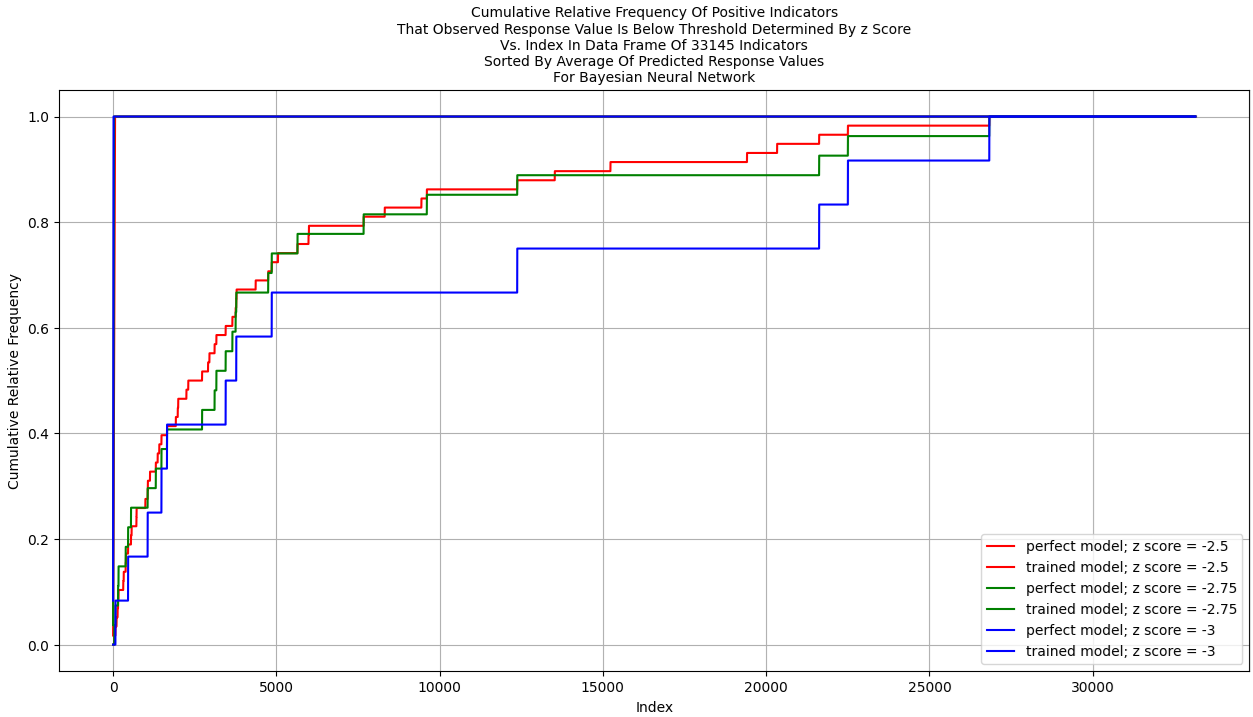
vs. index in a data frame of indicators

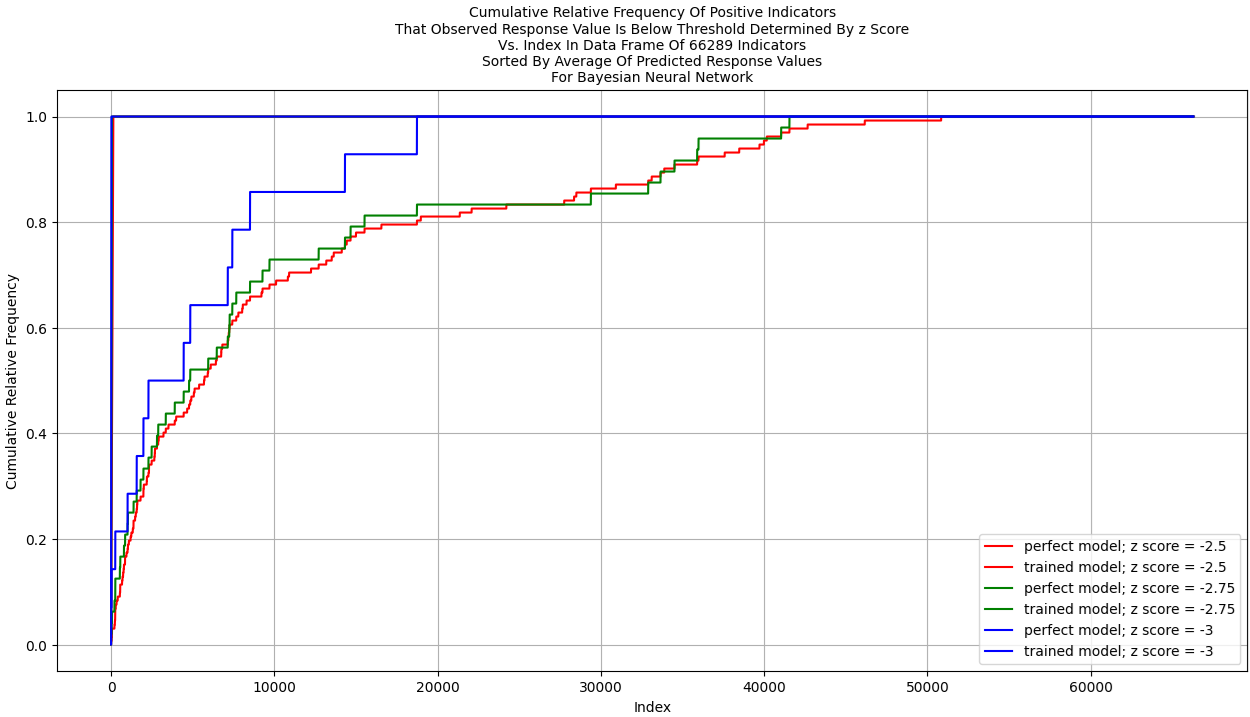
sorted by a column of averages of predicted response values

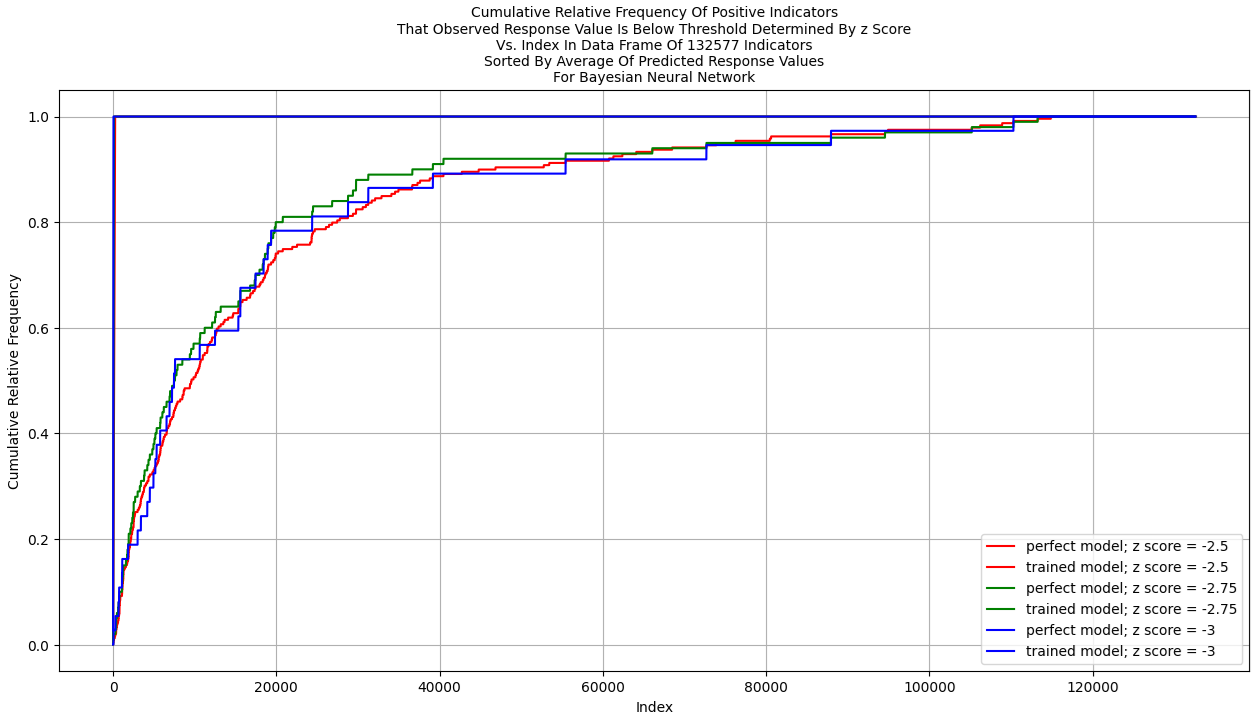
for a model.

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





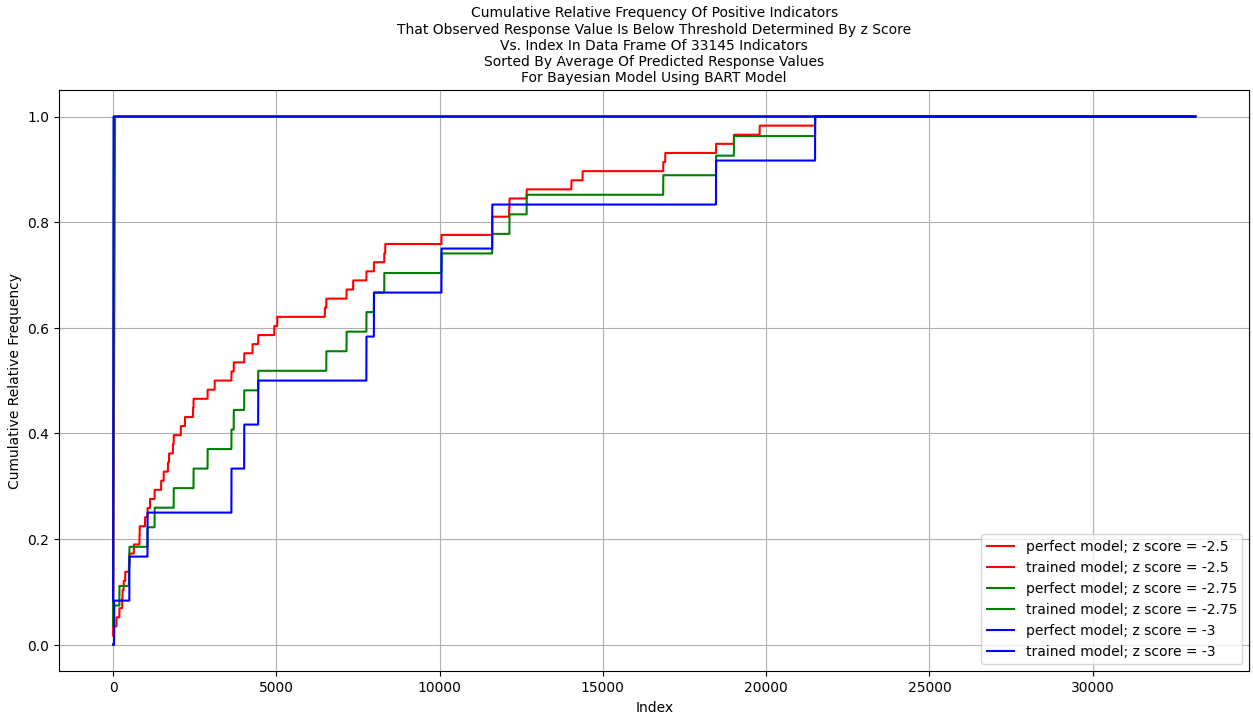


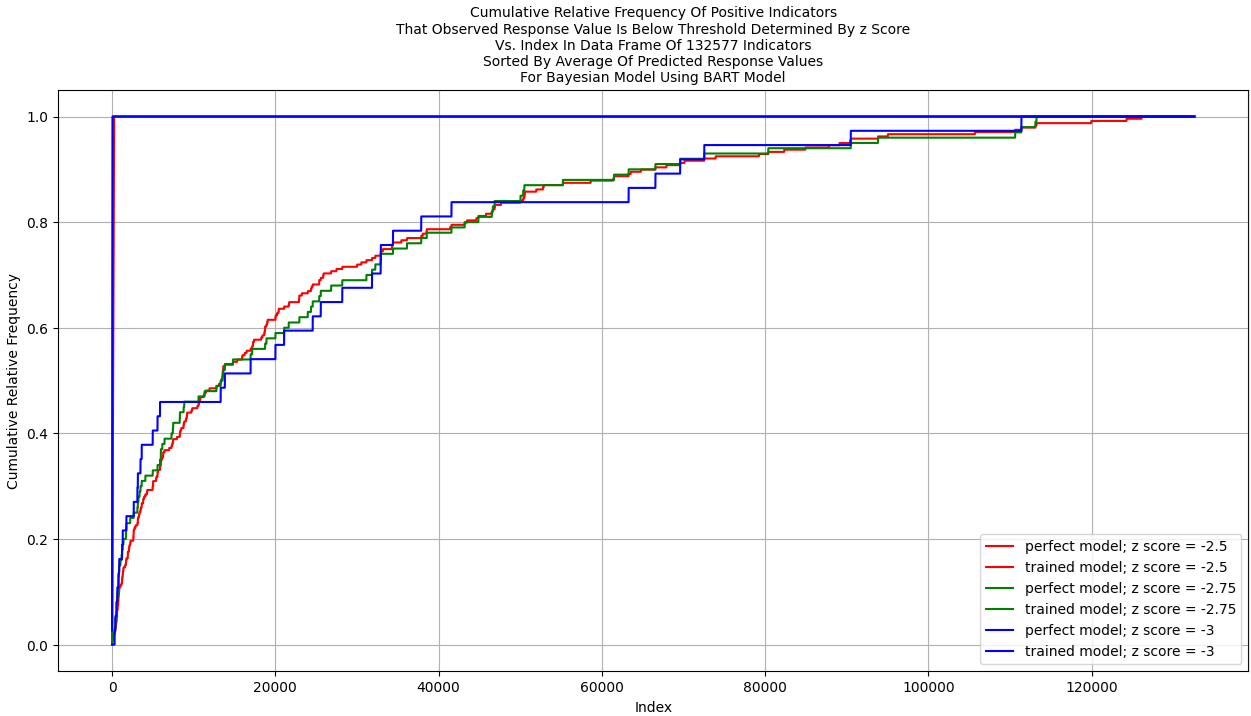
The Gains Curves for 33,145, 66,289, and 132,577 observations and our *Bayesian Model Using BART Model* are presented below. For a user-defined *z* score of or , our BNN performs better than our *Bayesian Model Using BART Model* when each is trained and tested on 33,145 observations. For a z score of , our *Bayesian Model Using BART Model* performs better.

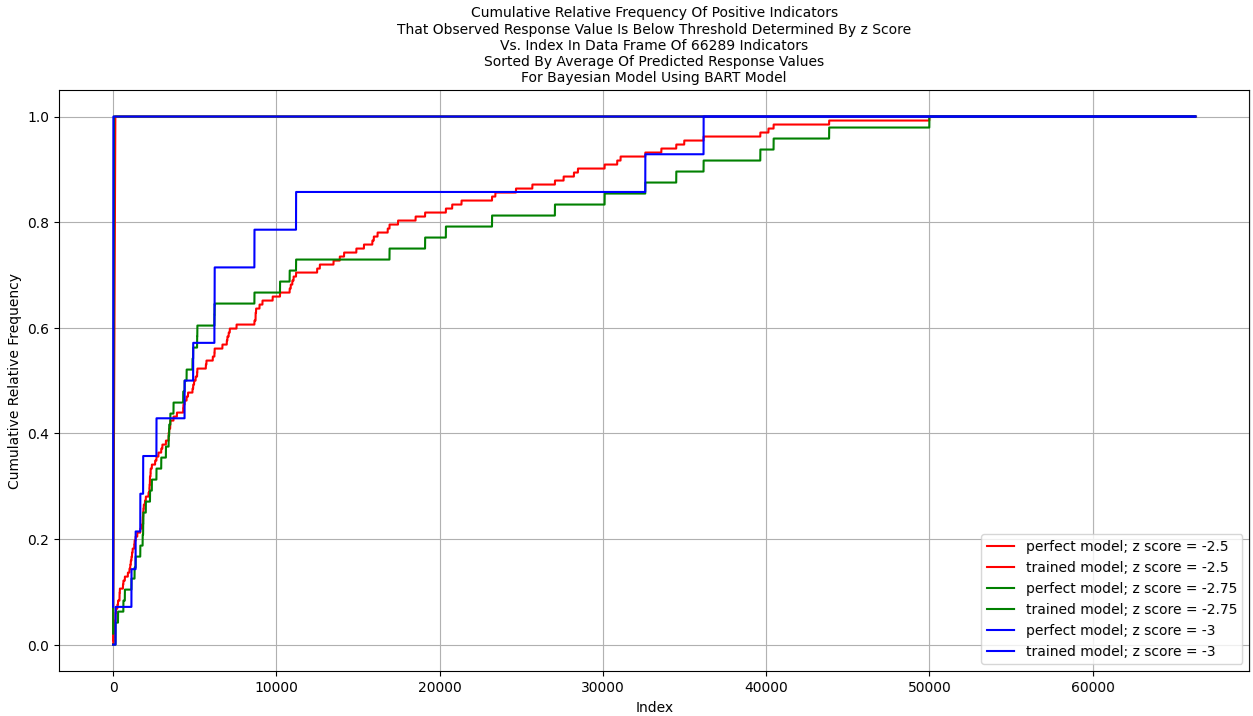
For a user-defined z score of or , our BNN performs better than our *Bayesian Model Using BART Model* when each is trained and tested on 66,289 observations. For a *z* score of , our *Bayesian Model Using BART Model* performs better.

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.

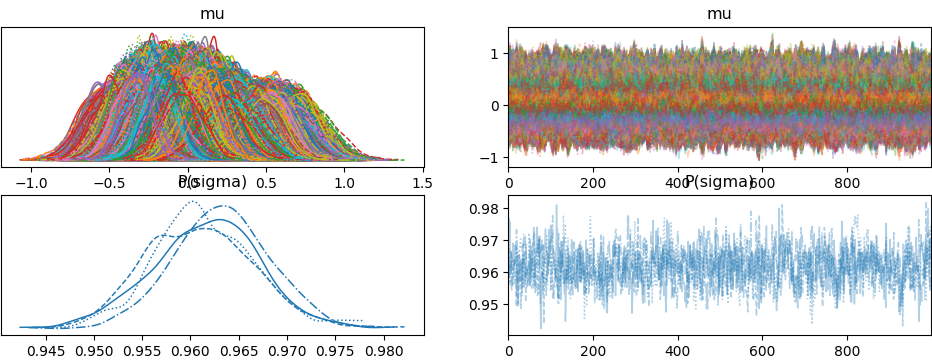
Both our BNN and our *Bayesian Model Using BART Model* may be expanded and tuned.

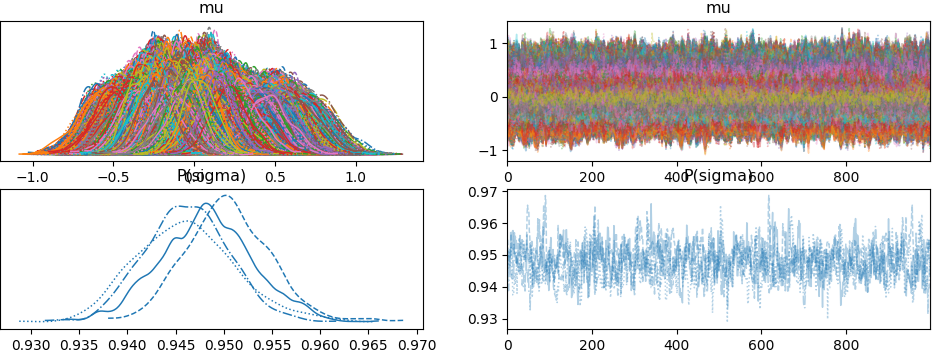


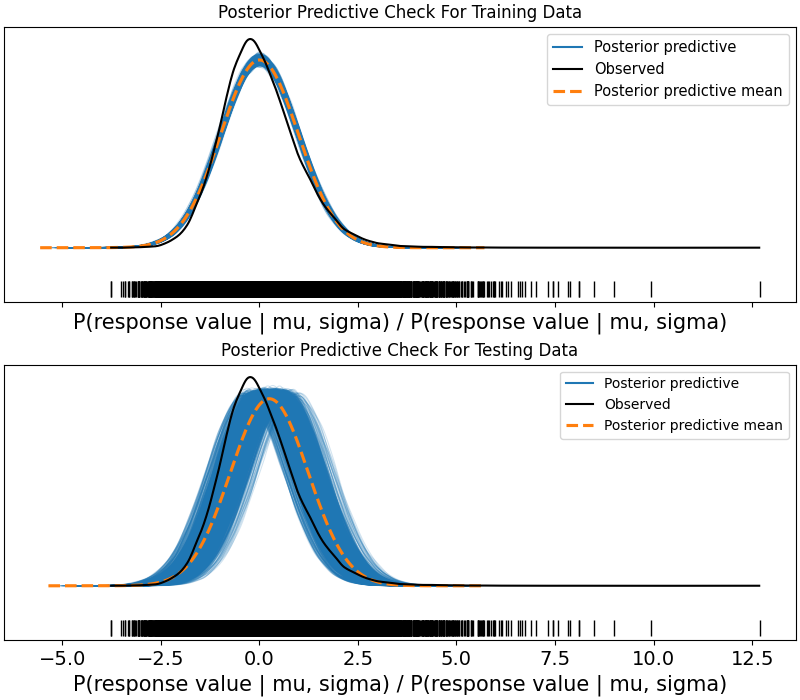


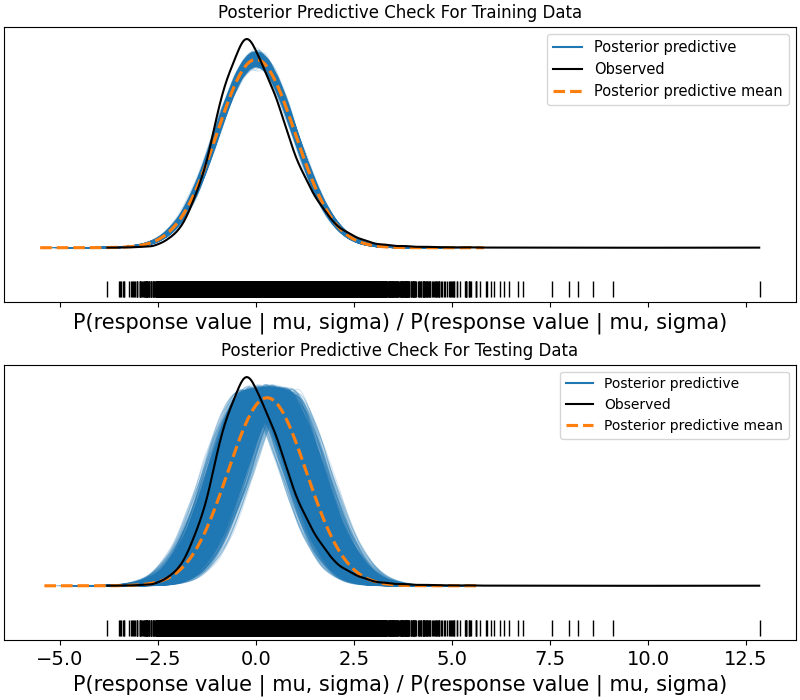


Below are the Trace Plots and Plot Of Posterior Predictive Checks for our *Bayesian Model Using BART Model* when the model is trained and tested on 33,145 and 66,289 observations. Perhaps our *Bayesian Model Using BART Model* performs better when the model is trained and tested on 66,289 observations than when the model is trained on 33,145 observations. It seems that the divergence of the average posterior predictive probability density distribution for our *Bayesian Model Using BART Model* when the model is trained and tested on 66,289 observations from the observed probability density distribution is on par with the divergence of the average posterior predictive probability density distribution for our BNN when the model is trained 132,577 observations from the observed probability density distribution. However, the variances of the traces used to find the average posterior predictive probability density distributions for our *Bayesian Model Using BART Model* when the model is tested is far greater than those for our BNN.









**Applications**

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.

Bayesian Docking-Score Predictor could be trained on (e.g., ) pairs of SMILES and docking score and used to predict docking scores of ligands based on many (e.g., ) other SMILES’s. Rows of data representing docking of ligands could be sorted in ascending order by predicted docking score. The first rows of data corresponding to the lowest docking scores could be added to the training data set.

Bayesian Docking-Score Predictor may be used to calculate Bayesian predictions, estimate uncertainty, or in optimization of hyperparameters of Bayesian models.

**Appendix 1: Exploring Change In Molar Gibbs Free Energy**

The enthalpy of a compound is the sum of the compound’s internal energy and the pressure energy 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 on the compound by its surroundings and the volume of the compound.

The molar internal energy of a compound is the internal energy of the compound per mole.

The molar volume is the volume of the compound per mole.

The molar pressure energy of a compound is the pressure energy of the compound per mole.

The molar enthalpy 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.

The change in enthalpy 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.

The change in molar enthalpy 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.

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.

is the probability that the compound is in the th state according to the Boltzmann distribution.

is the number of microstates whose energy equals the compound’s energy.

For an isolated system, .

The entropy of a compound is a quantity that satisfies “an infinitesimal change in entropy is equal to the ratio of an infinitesimal quantity of heat in a reversible reaction and the temperature of the compound”.

The molar entropy of a compound is the entropy of the compound per mole.

The change in entropy 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.

The change in molar entropy 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.

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

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

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.

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.