Baysian data analysis - predicting heart diseases

Joakim Juhava, Vesa Ranta-aho
Eino Miettinen

November 2020

1 Introduction

This project is part of the Bayesian Data Analysis 2020 course. Our goal is to use Bayesian data analysis methods to predict if a person has heart disease as accurately as possible using R and RStan. We will try different models and do different amounts of data preprocessing, and then choose the best method and Stan model using convergence diagnostics, model comparisons, predictive performance assessments and sensitivity analysis.

Our dataset is part of an older Kaggle competition (https://www.kaggle.com/ronitf/heart-disease-uci) - heart disease prediction. It is an important topic and could have rather significant effects in overall human health. After some research on how other people tried to solve this problem, only a few seemed to take the Bayesian approach without any proper results, which is one of the main reasons we wanted to take this dataset along with the importance of the topic. Our work differs completely from the works that others have made about this topic.

We will make two different models: a non-hierarchical logistic regression model and a hierarchical logistic regression model. The report consists of the following parts: introduction, data description, models, results, conclusions and appendix. All the STAN models and convergence diagnostics can be found under the 'Appendix' section.

2 Data description

The dataset consists of 303 rows of 13 numerical and categorical variables that are related to heart diseases, and the target variable, which tells if the subject had a heart disease or not. All subjects were hospitalized. The goal is to choose the most relevant variables and find the best coefficients for them in the non-hierarchical and hierarchical regression models so that given the explanatory variables, the models can predict the target variable as accurately as possible. For testing the predictive accuracy, the dataset will be divided into a training set and a testing set. The variables are as follows:

- Age (numerical)
- Sex (binary)

- CP (chest pain type, 4 categories)
- Trestbps (resting blood pressure, numerical)
- Chol (serum cholestoral in mg/dl, numerical)
- FBS (fasting blood sugar, binary, 1 if greater than 120 mg/dl)
- Restecg (resting electrocardiographic results, binary)
- Thalach (maximum heart rate achieved, numerical)
- Exang (exercise induced angina, binary)
- Oldpeak (ST depression induced by exercise relative to rest, numerical)
- Slope (the slope of the peak exercise ST segment, 3 categories)
- CA (number of major vessels colored by flourosopy, 4 categories)
- Thal (type of defect, 4 categories)
- Target (does subject have a heart disease or not, binary)

Figure 1 contains histograms of the variables separated by the target variable value. It gives initial hints about which variables are the most important ones - if the distribution of the values or categories of a variable differs significantly for subjects with a heart disease compared to subjects with no heart diseases, the variable is likely to be important.

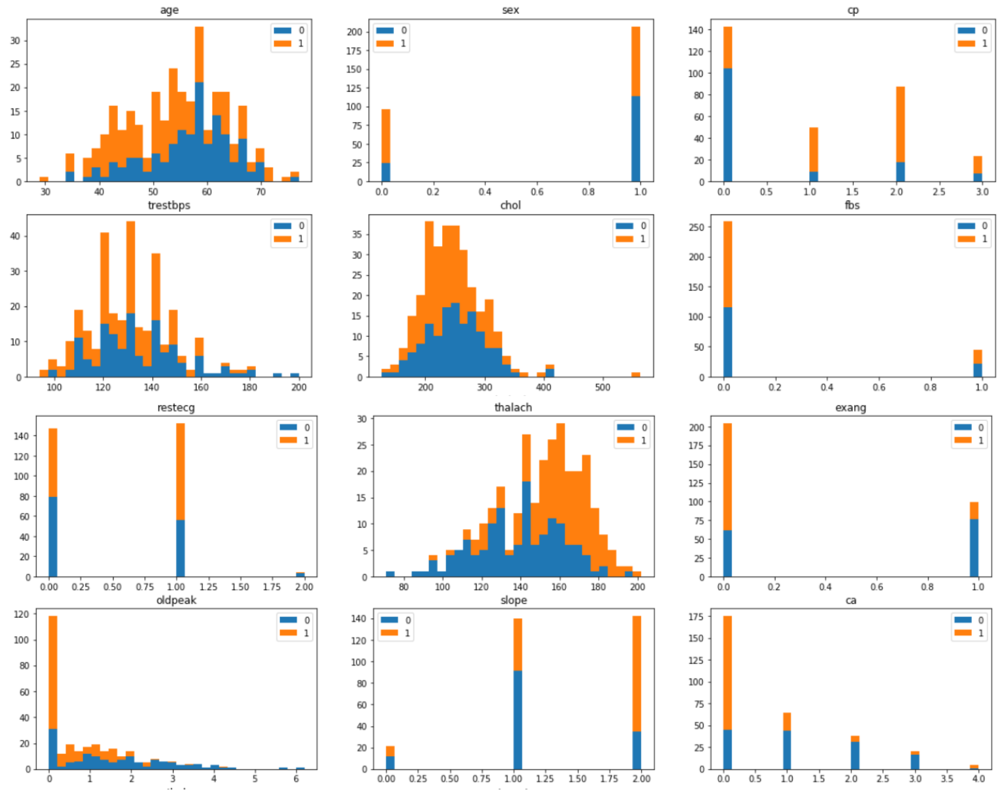

Figure 1: Histograms of the variables. Orange = has a heart disease. Blue = no heart disease.

After a quick observation one can see that the variable CP (chest pain type) is distributed very differently in the case of a heart disease and in the case of no heart disease. Most of the subjects with no heart diseases have no chest pain, while most of the subjects with a heart disease also have chest pain. Therefore we choose to include chest pain, and since it is categorical and divides the subjects nicely into groups, we will use it as one level of our hierarchical logistic regression model. Other variables that seem to have a clearly different distribution depending on the target variable are Thalach, Exang, Oldpeak, Slope and CA. A bit surprisingly serum cholestoral values don't seem to have much difference in the data of heart diseases and in the data of no heart diseases.

3 Models

3.1 Logistic Regression Model

We used logistic regression models because the target variable is binary and we assumed linear relationship between the input variables and the log-odds. With d input variables the relationship can be written in the following mathematical form:

$log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + ... + \beta_d x_d$

We get the odds by exponentiating the log-odds:

$\frac{p}{1-p} = e^{\beta_0 + \beta_1 x_1 + ... + \beta_d x_d}$

The p can be solved:

$p = \frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_d x_d)}}$

The p is probability that person has a heart disease in our model and it always gets values in (0,1). The obtained probability p is used as a parameter of Bernoulli distribution that the target variable y follows. The purpose of a logistic regression model is to find good values for the β_i parameters.

We removed categorical variables with more than two classes from the data because categorical variables are not suitable for logistic regression unless there is some logic in the order of the classes of a categorical variable. However, we kept the categorical variable chest pain type in data because it seemed important and it was used to make the different groups in our hierarchical logistic regression model that we would also implement and it is good to use the same variables when comparing two models. The categorical variable chest pain type is incorporated in the model by giving each chest pain type its own β_0 coefficient, and the correct β_0 would be used for every observation. Because there is quite many input variables in the data, removing some of them are likely to make the model better. Especially highly correlating input variables are a problem in logistic regression models. Therefore we compute the correlation matrix, which can be seen in figure 2

As we noted earlier, CP, Thalach, Exang and Oldpeak are pretty correlated with the target variable. Additionally, Age and Sex have some correlation. However, Thalach, Exang and Oldpeak are also highly correlated with each other, which is likely to cause problems. Therefore we only choose one of these variables, and since Oldpeak seems to be the least correlated with Age and Sex,

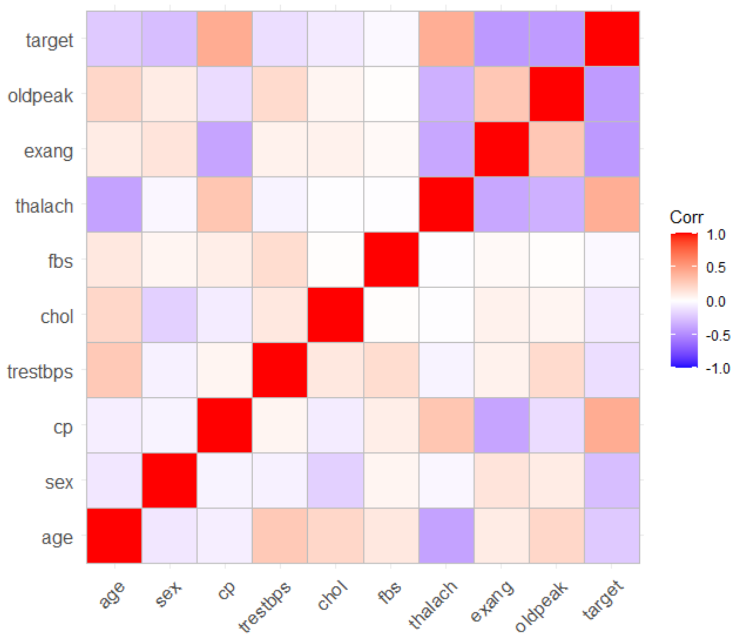

Figure 2: Correlation matrix of the data

we choose it. Unlike with the categorial variables before, we do not actually remove the rest of the variables completely, but use informative shrinkage priors to shrink the coefficients to near-zero values. This will be explained in more detail later.

The STAN code is in the appendix but we explain the most important parts here. In STAN code we used `bernoulli_logit` that transforms $\beta_0 + \beta_1 x_1 + ... + \beta_d x_d$ to p and makes p the parameter of Bernoulli distribution that the target variable y follows. In STAN code the $\beta_0 + \beta_1 x_1 + \beta_1 x_2 + ... + \beta_d x_d$ is written as `beta_0[cp[n]] + beta * x[n]`. The `beta_0[cp[n]]` is the β_0 coefficient for the chest pain type of the observation number n. The beta is a row vector that has all other β_i coefficients and x[n] is a vector that contains the input variables of an observation.

The default chain length was enough to get high enough ESS values, but initially the chains had divergent transitions left after the warmup. To correct this, we lowered the HMC stepsize by increasing the **adapt_delta** -parameter to 0.999. So, the options used for Stan were 4 chains, 2000 iterations with 1000 warmup iterations and adapt_delta = 0.999. The model was run with the following command:

```
separate_model <- stan(file = 'non_hierarchical_model.stan', data = standata,
                       iter = 2000, control = list(adapt_delta = 0.999))
```

For the included variables we used weakly informative priors for the β_i coefficients because we did not have information from other studies that we wanted to include in the model. We wanted to use the same prior distribution for each β_i coefficient but the input variables had different scales so the same prior would have different effects for different input variables. Therefore, we standardized the input variables i.e. transformed each input variable to have mean 0 and standard deviation 1. Standardizing the input variables also helps a lot when making inferences about the model because comparing the relative magnitudes of the β_i coefficients tells what input variables increased the risk of heart disease the most according to the model. The prior we used for each β_i coefficient was:

$\beta_i \sim \mathcal{N}(0, 100^2)$

The normal prior has an effect of favouring smaller values for β_i over big values. The mean 0 means that we do not incorporate knowledge about how an increase in an input variable affects the probability of heart disease for a patient. A positive value for a mean should be used if we knew that an increase in a certain variable increases the risk of having a heart disease. The standard deviation parameter controls how certain how much preference is given to values near to the chosen mean. We used a very high value 100 for the standard deviation so the priors do not affect the results much.

However, for the rest of the variables we used an informative shrinkage prior,

$\beta_i \sim \mathcal{N}(0, 0.01^2)$,

which essentially acts as feature selection, because it shrinks the coefficients of the unwanted variables to near-zero. However it is not as strict as simply removing the variables completely from the data.

3.2 Hierarchical Logistic Regression Model

We also did a hierarchical logistic regression model. The hierarchical logistic regression model works like the previously described model but the model is written in multiple levels. We chose that input variable chest pain type determines the levels in the model, so there are 4 levels. Each chest pain type has their own beta coefficients for all input variables. We had no prior information to distinguish between the different chest pain types which means that the chest pain types could be treated as exchangeable and the hierarchical model is valid.

In hierarchical models there are so called hyperpriors that are used to generate parameters for the actual priors. When a parameter is drawn from a hyperprior the same parameter is used for priors in all levels in the hierarchical model.

The STAN code for the hierarchical model is almost the same as for the previously described basic logistic regression model so the explanation for the STAN model is in the previous section, and again, the STAN code itself is in the appendix. The only exception was that now there are different β_i coefficients for each chest pain type, which use hyperpriors that are the same for all of the chest pain types. The beta variable is therefore a matrix. Also the priors were set differently, which is discussed in the next paragraph. The same STAN options as with the previous model were used, because again the ESS was good with the default chain length, but the chains had divergent transitions. The options used were 4 chains, 2000 iterations with 1000 warmup iterations and adapt_delta $= 0.999$. The model was run with the following command:

```
hierarchical_model <- stan(file = 'hierarchical_model.stan', data = standata,
                           iter = 2000, control = list(adapt_delta = 0.999))
```

The prior choosing logic is the same as with the previous model: we did not have information from other studies so we chose weakly informative priors for all the "active" input variables in the hierarchical model. However this time the β_i priors utilize hyperpriors: the mean of the normal distribution of beta is a wide normal distribution, and the variance has a wide non-negative inverse-chi-squared prior. Therefore the priors are:

$\mu_i \sim \mathcal{N}(0, 100^2)$

$\sigma_i \sim inv - \chi^2(0.1)$

$\beta_{ji} \sim \mathcal{N}(\mu_i, \sigma_i^2)$

For the rest of the parameters we use similar distributions but with different parameters, to again shrink them to near zero to achieve variable selection:

$\mu_i \sim \mathcal{N}(0, 0.01^2)$

$\sigma_i \sim inv - \chi^2(100)$

$\beta_{ji} \sim \mathcal{N}(\mu_i, \sigma_i^2)$

4 Results

4.1 Logistic Regression Model

Using the weakly informative priors for Age, Sex, CP and Oldpeak, and the shrinkage priors for the rest of the non-categorial variables, and default STAN settings, the model achieved R-hat values (<1.01) and the ESS was good, but it had divergent transitions. Convergence was therefore not good initially. This was due to too large step sizes, and was easily corrected by increasing the **adapt_delta** -parameter to 0.999.

With the new settings, we achieved good convergence: the divergences were gone, all R-Hats were really close to 1 (under 1.005), and the ESS's were large. The close to 1 R-hat values mean that the chains have probably converged and the estimates are reliable. The effective sample sizes also match the total number of draws so we can conclude that this model is reliable.

We did posterior predictive checks to test if the predictions of the model match with the actual data. We did the posterior predictive checks by testing the accuracy of the predictions made by the model. This was done by dividing the 303 row data set into a 270 row training set and a 33 row test set, and predictions for the target variables of the test set were made by drawing from the predictive posterior distribution of the target variable. Table 1 contains the predictions and convergence diagnostics of the first five predictions (only five to save time). The results can be interpreted so that the y_pred mean tells the probability that the subject has a heart disease.

If we count all predictions where the mean is equal to or over 0.5 to have a heart disease and the rest to not have one, 29 of the 33 predictions were correct, resulting in a 88% predictive accuracy. The test was run multiple times with randomly chosen test sets to make sure the result is not due to chance. All of the tests yielded similar results, which confirms that the model really has

Table 1: Logistic Regression Model convergence diagnostics

	mean	se mean	sd	R-hat	MCSE sd	Bulk ESS	Tail ESS
y_pred$_1$	0.86	0.0055962007	0.34	0.9995066	0.0039574157	3794	4000
y_pred$_2$	0.11	0.0049871627	0.31	1.0009922	0.0035267118	4031	4000
y_pred$_3$	0.30	0.0073303836	0.46	1.0001374	0.0051837507	3909	4000
y_pred$_4$	0.03	0.0026010866	0.16	1.0004705	0.0018393793	4024	4024
y_pred$_5$	0.70	0.0073942197	0.46	1.0014797	0.0052288984	3856	4000

good predictive accuracy for such a simple model. Still, for real medical use, the model is way too unreliable.

As a part of model selection, we computed the PSIS-LOO elpd values, pareto k diagnostics and p_eff values. They were the following: elpd_loo = -127.8668 p_eff = 7.382036

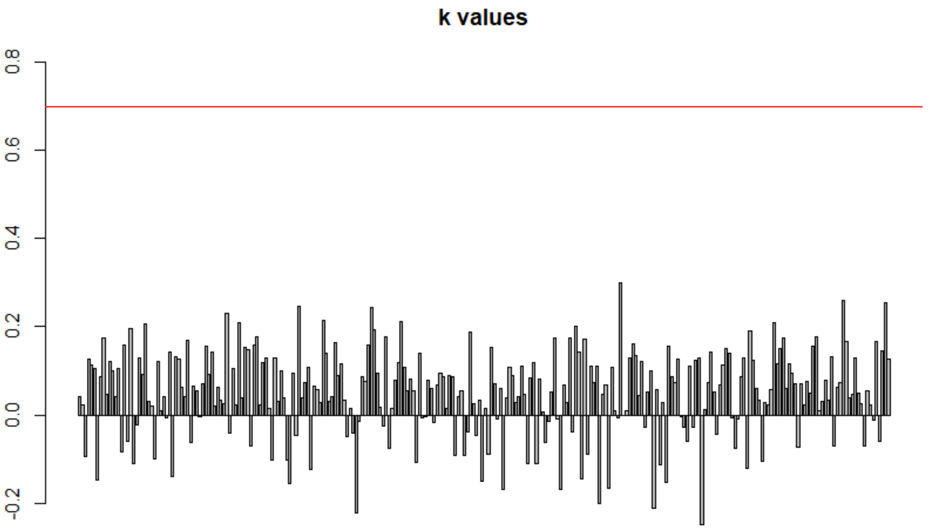

Figure 3: k-values of the separate model

All of the k-values are under 0.7, which indicate that the elpd_loo estimate is reliable.

We did prior sensitivity analysis for the model. Our original prior was $\beta_i \sim \mathcal{N}(0, 100^2)$ and we tried changing it to $\beta_i \sim \mathcal{N}(0, 10^2)$ and $\beta_i \sim \mathcal{N}(10, 100^2)$ to see if the results would change a lot. The change of priors did not significantly affect any of the results. This is partly because even the new priors were weakly informative, so they don't affect the posterior that much.

4.2 Hierarchical model

Using weakly informative hyperpriors for Age, Sex, CP and Oldpeak, and the shrinkage hyperpriors for the rest of the non-categorial variables, and default STAN settings, the model performed in a really similar way to the previous one. It also achieved good R-hat values (<1.01) and the ESS was great, but it had divergent transitions. Convergence was therefore not good initially. This was again due to too large step sizes, and was easily corrected by increasing the **adapt_delta** -parameter to 0.999.

With the new settings, we achieved much better convergence: the divergences were gone, all R-Hats were really close to 1 (under 1.005), and the ESS's were large. The interpretation of these values is the same as with the previous model.

We did posterior predictive checks to test if the predictions of the model match with the actual data. We did the posterior predictive checks by testing the accuracy of the predictions made by the model. This was done by dividing the 303 row data set into a 270 row training set and a 33 row test set, and predictions for the target variables of the test set were made by drawing from the predictive posterior distribution of the target variable. Table 1 contains the predictions and convergence diagnostics of the first five predictions (only five to save time). The results can be interpreted so that the y_pred mean tells the probability that the subject has a heart disease.

Table 2: Hierarchical Logistic Regression Model convergence diagnostics

	mean	se mean	sd	R-hat	MCSE sd	Bulk ESS	Tail ESS
y_pred_1	0.828250000	0.0063319624	0.37721017	1.0012024	0.0044777416	3549	4000
y_pred_2	0.123000000	0.0052513453	0.32847827	0.9995767	0.0037135387	3913	4000
y_pred_3	0.296750000	0.0072950564	0.45688250	0.9994811	0.0051587675	3922	4000
y_pred_4	0.020750000	0.0023865875	0.14256409	0.9997289	0.0016877102	3568	3568
y_pred_5	0.703000000	0.0070455013	0.45699367	1.0006077	0.0049822672	4207	4000

If we count all predictions where the mean is equal to or over 0.5 to have a heart disease and the rest to not have one, 30 of the 33 predictions were correct, resulting in a 91% predictive accuracy. The test was run multiple times with randomly chosen test sets to make sure the result is not due to chance. All of the tests yielded similar results, which confirms that the model really has good predictive accuracy for such a simple model. Still, for real medical use, the model is way too unreliable.

As a part of model selection, we computed the PSIS-LOO elpd values, pareto k diagnostics and p_eff values. They were the following: elpd_loo = -129.2591 p_loo = 12.09546

All of the k-values are under 0.7, which indicate that the elpd_loo estimate is reliable. The elpd_loo value of the first model is slightly higher than with this hierarchical model, so based on it the first model is slightly better.

We did prior sensitivity analysis for the model by changing the prior $\mu_i \sim \mathcal{N}(0, 100^2)$ to $\mu_i \sim \mathcal{N}(0, 10^2)$ and $\mu_i \sim \mathcal{N}(10, 100^2)$. We also tried changing $\sigma_i \sim inv - \chi^2(0.1)$ to $\sigma_i \sim inv - \chi^2(1)$. Again, this had no significant effect on the results.

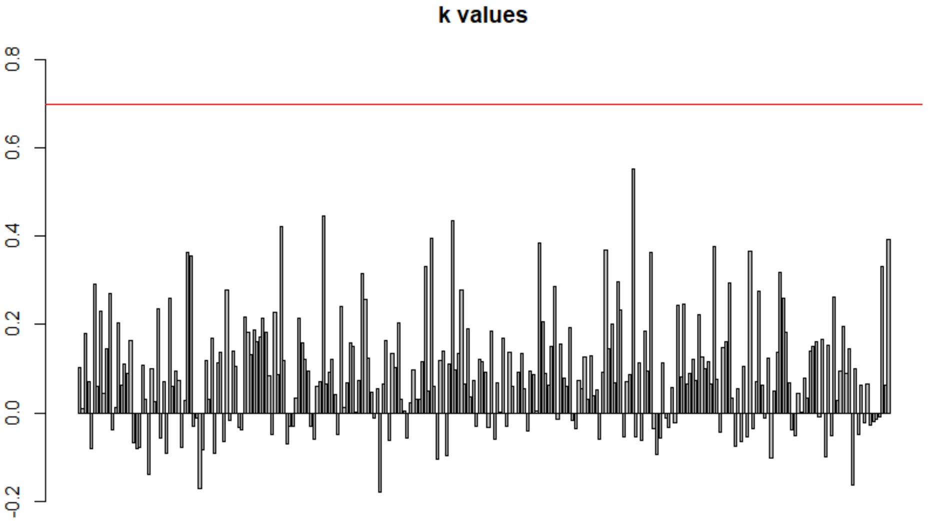

Figure 4: k-values of the hierarchical model

5 Conclusion and discussion

5.1 Conclusion and discussion

The different regression models predicted heart diseases rather well after some preprocessing of the variables. We managed to achieve 91% predictive accuracy. Even though this score won't probably be accurate enough for medical use, it shows that it could be possible to make a model that could predict heart diseases precisely.

For future research and predictions, we would try to improve the model by adding some new variables that weren't in the dataset that could have a significant impact on heart diseases. This could easily improve the predictive accuracy of our models and perhaps even make them reliable enough for medical use. And of course the dataset could have a bit more data, 303 rows of data can have a few outliers that make the models more inaccurate. And obviously our team didn't have much prior knowledge on heart diseases, it could be possible to improve the model with some proper medical assistance. Even so our team managed to create a fairly decent model for predicting heart diseases with the Bayesian approach.

We learned many new things regarding Bayesian data analysis. The most important thing probably was that how to make a fully working Bayesian data analysis project from scratch. This could be very useful in the future because one most likely barely counters the weekly assignment-like problems but problems like this project - a new dataset without any proper knowledge of it or any detailed constructions for it, and the goal is to predict something meaningful from it. More precisely, the team learned especially a lot from variable preprocessing and confirming whether a model is reliable or not.

5.2 Self-reflection

The group learned many new methods while making the project. The group learned especially a lot about organizing a Bayesian data analysis project and using many different skills learned during the weekly-assignments. Starting was difficult because the dataset was completely new, meaning our group didn't have any prior knowledge of it. Besides that our group managed to fulfill our goals and get a proper model to predict heart diseases. The project was very useful in regards of future situations. Now the project members know how to analyze completely new data the Bayesian way and check whether the created models can be reliable. Reliability of the models is very important and the group learned how to prove reliability of models in various ways.

6 Appendix

Logistic regression STAN model:

```
data {
  int<lower=0> N; //number of observations
  int<lower=0> D; //number of dimensions in data excluding chest pain type
  int<lower=0> P; //number of predictions
  int<lower=0,upper=1> y[N]; //targets
  int<lower=1,upper=4> cp[N]; //chest pain types (1,2,3,4)
  int<lower=1,upper=4> pred_cp[P]; //chest pain types in prediction data
  vector[D] x[N]; //data
  vector[D] pred_x[P]; //prediction data
}

parameters {
  real beta_0[4];
  row_vector[D] beta;
}

model {
  //priors
  for (i in 1:4)
      beta_0[i] ~ normal(0, 100);

  //for (d in 1:D)
  //    beta[d] ~ normal(0, 100);
  beta[1] ~ normal(0, 100);
  beta[2] ~ normal(0, 100);
  beta[3] ~ normal(0, 0.01);
  beta[4] ~ normal(0, 0.01);
  beta[5] ~ normal(0, 0.01);
  beta[6] ~ normal(0, 0.01);
  beta[7] ~ normal(0, 0.01);
  beta[8] ~ normal(0, 100);

  //likelihood
  for (n in 1:N)
      y[n] ~ bernoulli_logit(beta_0[cp[n]] + beta * x[n]);
}

generated quantities {
  vector[N] log_lik; //log likelihoods
  int<lower=0,upper=1> y_pred[P]; //predictions

  //predict target in test set
  for (p in 1:P)
      y_pred[p] = bernoulli_logit_rng(beta_0[pred_cp[p]] + beta * pred_x[p]);

  //calculate log likelihoods for model evaluation
  for (n in 1:N)
      log_lik[n] = bernoulli_logit_lpmf(y[n] | beta_0[cp[n]] + beta * x[n]);
}
```

Hierarchical logistic regression STAN model:

```
data {
  int<lower=0> N; //number of observations
  int<lower=0> D; //number of dimensions in data excluding chest pain type
  int<lower=0> P; //number of predictions
  int<lower=0,upper=1> y[N]; //targets
  int<lower=1,upper=4> cp[N]; //chest pain types (1,2,3,4) (hierarchical groups)
  int<lower=1,upper=4> pred_cp[P]; //chest pain types in prediction data
  vector[D] x[N]; //data
  vector[D] pred_x[P]; //prediction data
}

parameters {
  real mu_0;
  real<lower=0> sigma_0;
  real beta_0[4];
  real mu[D];
  real<lower=0> sigma[D];
  row_vector[D] beta[4];
}
```

```
model {
  //priors
  mu_0 ~ normal(0, 100);
  sigma_0 ~ inv_chi_square(0.1);
  for (i in 1:4)
      beta_0[i] ~ normal(mu_0, sigma_0);

  mu[1] ~ normal(0, 100);
  mu[2] ~ normal(0, 100);
  mu[3] ~ normal(0, 0.01);
  mu[4] ~ normal(0, 0.01);
  mu[5] ~ normal(0, 0.01);
  mu[6] ~ normal(0, 0.01);
  mu[7] ~ normal(0, 0.01);
  mu[8] ~ normal(0, 100);
  sigma[1] ~ inv_chi_square(0.1);
  sigma[2] ~ inv_chi_square(0.1);
  sigma[3] ~ inv_chi_square(100);
  sigma[4] ~ inv_chi_square(100);
  sigma[5] ~ inv_chi_square(100);
  sigma[6] ~ inv_chi_square(100);
  sigma[7] ~ inv_chi_square(100);
  sigma[8] ~ inv_chi_square(0.1);

  for (d in 1:D) {
      for (i in 1:4)
          beta[i,d] ~ normal(mu[d], sigma[d]);
  }

  //likelihood
  for (n in 1:N)
      y[n] ~ bernoulli_logit(beta_0[cp[n]] + beta[cp[n]] * x[n]);
}

generated quantities {
  vector[N] log_lik; //log likelihoods
  int<lower=0,upper=1> y_pred[P]; //predictions

  //predict target in test set
  for (p in 1:P)
      y_pred[p] = bernoulli_logit_rng(beta_0[pred_cp[p]] + beta[pred_cp[p]] * pred_x[p]);
  //calculate log likelihoods for model evaluation
  for (n in 1:N)
      log_lik[n] = bernoulli_logit_lpmf(y[n] | beta_0[cp[n]] + beta[cp[n]] * x[n]);
}
```

Table 3: Extra convergence diagnostics for the Logistic Regression Model

	mean	se mean	2.5%	97.5%	R-hat	MCSE Q2.5	MCSE Q97.5	Bulk ESS	Tail ESS
cp_1	0.22	0.0076394235	-0.44373569	0.943674185	1.0010188	0.0183641343	0.0257585815	2054	2526
cp_2	2.2	0.0108691623	1.27598514	3.370809966	1.0006448	0.0238877871	0.0322621115	2441	2426
cp_3	2.5	0.0101792864	1.65821160	3.410594389	1.0011833	0.0261989617	0.0258456855	1945	2166
cp_4	2.7	0.0122887740	1.49236876	4.011424911	1.0009741	0.0322762586	0.0313395420	2678	2856
age	-0.50	0.0027222865	-0.83501670	-0.165149240	1.0009073	0.0091331642	0.0060206502	3953	2899
sex	-1.74	0.0095867035	-2.54184754	-0.991224679	1.0006443	0.0200014246	0.0237015223	1680	2272
trestbps	-0.001	0.0001359591	-0.02006581	0.018926587	1.0000563	0.0003808480	0.0003169722	5367	2672
chol	-0.001	0.0001454460	-0.02110244	0.018381724	1.0011752	0.0005683772	0.0004853234	4814	3001
fbs	-0.000	0.0001315725	-0.01894708	0.018333342	1.0022512	0.0003464041	0.0005751162	5469	2855
thalach	0.002	0.0001389604	-0.01868597	0.021758015	1.0026777	0.0005631372	0.0006807773	5578	2594
exang	-0.001	0.0001331759	-0.02026937	0.018768701	1.0001938	0.0003657261	0.0005356523	5522	3139
oldpeak	-0.90	0.0031935500	-1.30714457	-0.512398956	1.0009550	0.0117684063	0.0092028456	3972	2951