Baysian Data Analysis - Predicting Heart Disease

Joakim Juhava, Vesa Ranta-aho
Eino Miettinen

February 2021

Contents

1 Introduction

This project is part of the Aalto University Bayesian Data Analysis 2020 course.

According to WHO, cardiovascular diseases are the number one reason for death in the world. Heart disease prediction is therefore an important topic and could have rather significant effects in overall human health. We chose heart disease prediction as our topic because of its importance, along with fact that based on a quick search, no Bayesian analysis has previously been done on the data we chose, other than with simple naive Bayes classifiers.

Our goal is to use Bayesian data analysis methods to predict if a person has heart disease or not as accurately as possible from data related to heart diseases using R and RStan. We will try a non-hierarchical logistic regression model and a hierarchical logistic regression model and perform feature selection, different amounts of data preprocessing, and then choose the best methods and Stan models using convergence diagnostics, model comparisons, predictive performance assessments and sensitivity analysis.

The report consists of the following sections: introduction, data, models, results, conclusions and discussion, and appendix. In the data section we describe the data, analysis problem and existing analyses, as well as perform variable selection and data preprocessing. In the models section we describe the models we use and justify the likelihood and prior choices. In the results section we perform convergence analysis, posterior predictive checks, predictive performance assessment, model selection and prior sensitivity analysis. The appendix contains all the full Stan models. The Stan models, as well as R code, can also be found at https://github.com/Eficnbo/BDA-proju.

2 Data

2.1 Data Description

The dataset used is a part of an older Kaggle competition - heart disease prediction (https://www.kaggle.com/ronitf/heart-disease-uci). It consists of 303 rows and 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 variables are presented in table 1.

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.

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. 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

Table 1: Variables

name	type	description
Age	numerical (discrete)	subject's age
Sex	categorical (binary)	subject's sex (1 = male, 0 = female)
CP	categorical (4)	chest pain type
Trestbps	numerical (discrete)	resting blood pressure
Chol	numerical (discrete)	serum cholestoral in mg/dl
FBS	categorical (binary)	fasting blood sugar greater than 120 mg/dl (1 = true, 0 = false)
Restecg	categorical (binary)	resting electrocardiographic results
Thalach	numerical (discrete)	maximum heartrate achieved
Exang	categorical (binary)	exercise induced angina (1 = yes, 0 = no)
Oldpeak	numerical (discrete)	ST depression induced by exercise relative to rest
Slope	categorical (3)	slope of the peak exercise ST segment
CA	categorical (4)	number of major vessels colored by flouroscopy
Thal	categorical (4)	type of defect
Target	categorical (binary)	subject has heart disease (1 = true, 0 = false)

diseases.

2.2 Analysis goal and existing analyses

The goal of the analysis problem is to build a model that can classify a subject to having or not having a heart disease from the explanatory variables as accurately as possible. For us, this means choosing the most relevant variables and finding the best coefficients for them in the non-hierarchical and hierarchical regression models so that the model can predict if a subject has a disease or not. For testing the predictive accuracy, the data set will be divided into a training set and a testing set.

After some research on how other people have tried to solve this problem, none seemed to take on the full Bayesian modeling approach, but a handful of people used naive Bayes classifiers. Logistic regression is a popular choice for the problem, but our work differs from the work that others have made by doing it the Bayesian way.

2.3 Variable Selection and Data Preprocessing

Categorical variables with more than two classes are not suitable for logistic regression unless there is some logic in the order of the classes, or unless each category is represented with a separate binary variable. Since the data already contains relatively many input variables compared to the amount of data, removing some of them is likely to make the model better. To not increase the data dimensionality further, we choose to remove the categorical variables with more than two classes, with the exception of chest pain type. As it seems important according to the initial histogram analysis, it will be used as categories in the hierarchical model and is also included in the non-hierarchical model. It is incorporated in the model by giving each chest pain type i its

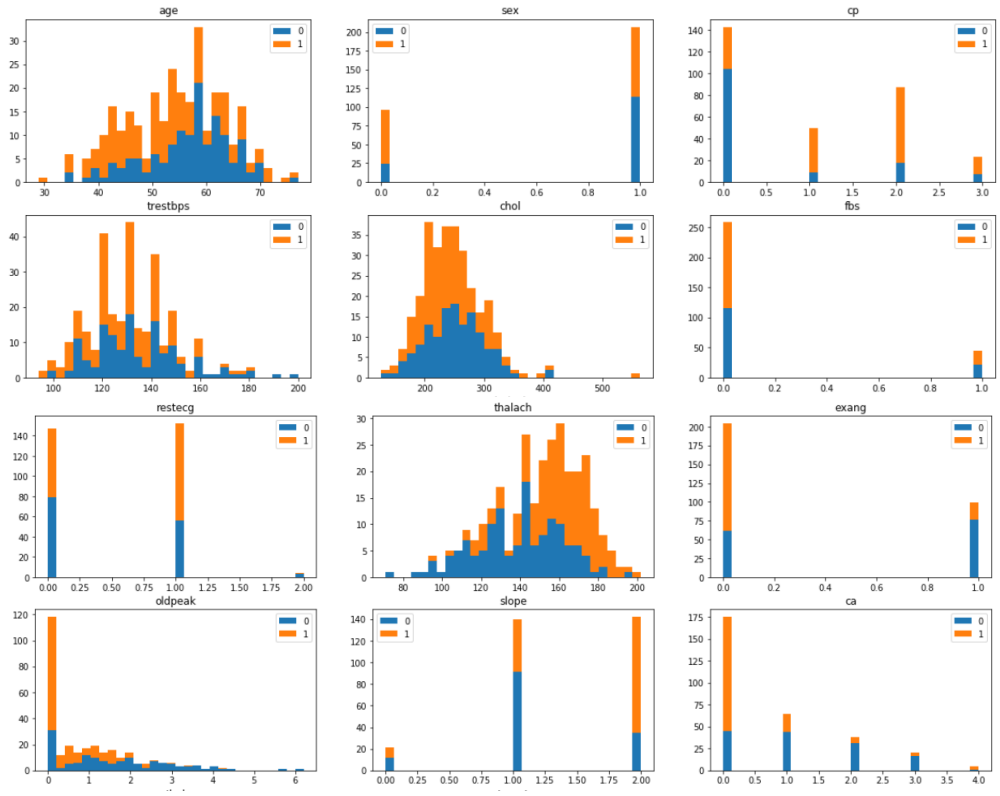


Figure 1: Histograms of the variables. Orange = has a heart disease. Blue = no heart disease.

own $\beta_{0,i}$ coefficient, and the correct $\beta_{0,i}$ is used for every observation.

The data dimensionality is reduced further by removing the remaining input variables that do not correlate with the target variable. Input variables that correlate highly with other input variables can also cause problems, such as unstable parameter values and large confidence intervals in logistic regression models, so only one variable from each highly correlating variable group will be chosen. To assess which variables we should remove based on these reasons, we compute the correlation matrix, which can be seen in figure 2

The correlation matrix (and the histograms) show that CP, Thalach, Exang and Oldpeak are 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, we choose it. Unlike with the categorical variables, 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 in section 3. The final selected variables are Age, Sex, CP and Oldpeak.

The input variables also have different scales, so we standardize them, i.e. transform each input variable to have mean 0 and standard deviation 1. Standardizing the input variables makes it possible to use the same prior distribution for each variable so that the effect of the prior is the same, and it helps a lot when making inferences about the model because comparing the relative

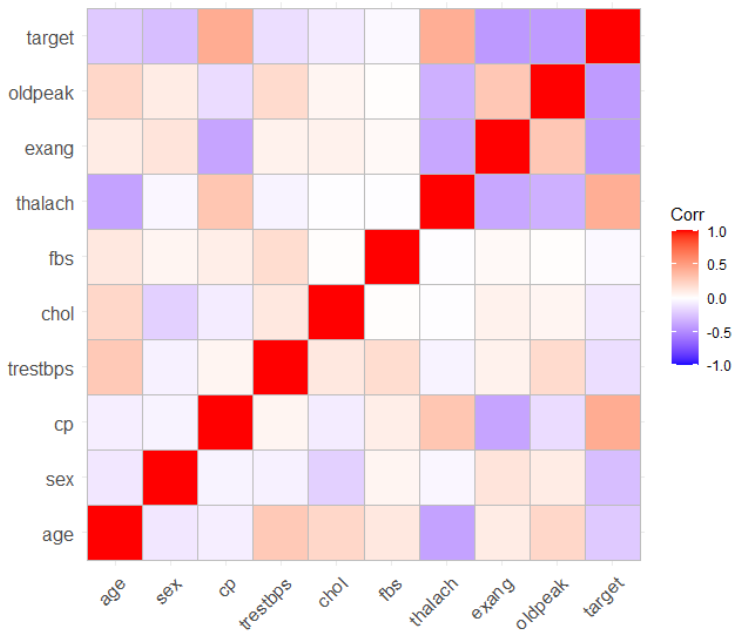


Figure 2: Correlation matrix of the data

magnitudes of the regression coefficients tells what input variables increase the risk of heart disease the most according to the model.

Principal component analysis and including more of the categorcial variables were also tried out, but they yielded bad results in the initial runs and were not analyzed further.

3 Models

3.1 Logistic Regression Model

The structure of the logistic regression model is presented in figure 3, and the details about the model, likelihood and priors are explained in sections 3.1.1 and 3.1.2.

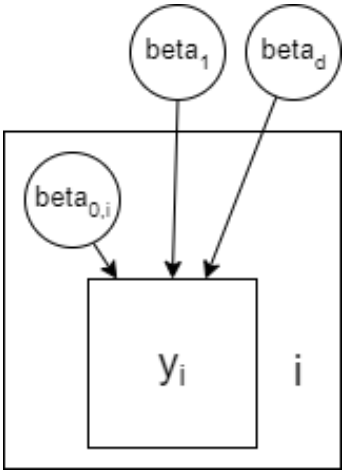


Figure 3: Logistic regression model structure

3.1.1 Likelihood

The binary target variable y for a subject follows a Bernoulli distribution

$$y \sim Bernoulli(p),$$

where p is the probability that the subject has a heart disease and always gets values in [0,1]. Since the target variable is binary, a simple linear regression model can not be used. Instead, we use a logistic regression model, where we assume a linear relationship between the input variable and the log-odds. With d input variables β_j 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:

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

$log(\frac{p}{1-p})$ is the logit transformation of p, and p is used as the parameter in the Bernoulli distribution of the target variable, so by using the inverse logit transformation we get a likelihood for y in terms of the input variables:

$$y \sim Bernoulli(p)$$
$$y \sim Bernoulli(logit^{-1}(logit(p))$$
$$y \sim Bernoulli(logit^{-1}(\beta_0 + \beta_1 x_1 + ... + \beta_d x_d))$$

We model each chest pain category i to have a separate $\beta_{0,i}$ coefficient, but common β_j coefficients for $j > 0$, which is illustrated in figure 3.

The Stan code is in the appendix but we explain the most important parts here. In Stan code we define the likelihood for y_n as

```
y[n] ~ bernoulli_logit(beta_0[cp[n]] + beta * x[n]);
```

where `beta_0[cp[n]]` is the $\beta_{0,i}$ coefficient for the chest pain type (i) of the observation number n, `beta` is a row vector that has all the other β_j coefficients and `x[n]` is a vector that contains the input variables of the observation n.

3.1.2 Priors

For the included variables, we use weakly informative priors for the β_j coefficients because we do not have any substantial knowledge about the medical field or information from other studies that we wanted to include in the model. Since the data is standardized, we can simply use the same prior with the same effect for each β_j:

$$\beta_{0,i} \sim \mathcal{N}(0, 100^2) \ \forall \ i$$

and

$$\beta_j \sim \mathcal{N}(0, 100^2) \ \forall \ j > 0.$$

In the Stan code, this is defined with

```
for (i in 1:4)
   beta_0[i] ~ normal(0, 100);
```

for all of the four chest pain categories i, and

```
beta[j] ~ normal(0, 100);
```

for all selected variables j.

The normal prior has an effect of favouring smaller values for β_j 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, and a negative value should be used if we knew that the variable decreases the risk. The standard deviation parameter controls the informativeness of the distribution. 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,

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

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.

In the Stan code, the informative priors are defined with

```
beta[k] ~ normal(0, 0.01);
```

for all variables k that are not selected.

3.2 Hierarchical Logistic Regression Model

The hierarchical logistic regression model works mostly like the previously described model, but it has multiple levels. The data is divided into four categories with the variable chest pain type, and each category has their own beta coefficients for all input variables. However, the parameters of these coefficients' priors also have their own prior distributions, or so called hyperpriors, which are the same for all of the categories, and are used to draw the parameter values for the coefficient priors. We had no prior information to distinguish between the different chest pain types, which means that the chest pain types can be treated as exchangeable and the hierarchical model is valid. The structure of the hierarchical logistic regression model is presented in figure 4.

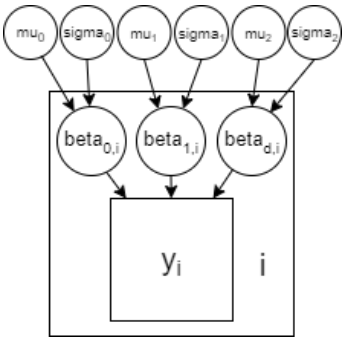


Figure 4: Hierarchical logistic regression model structure

3.2.1 Likelihood

The underlying relation of the target variable and input variables is exactly the same as with the non-hierarchical model, so the likelihood is also the same, with the only difference being that each data category (based on the chest pain type) has their own $\beta_{j,i}$ coefficients for all variables, instead of having common β_j distributions for $j > 0$. The likelihood is therefore

$$y_n \sim Bernoulli(logit^{-1}(\beta_{0,i} + \beta_{1,i}x_1 + ... + \beta_{d,i}x_d)),$$

where j is the category of observation n. In Stan this is defined with

```
y[n] ~ bernoulli_logit(beta_0[cp[n]] + beta[cp[n]] * x[n]);
```

where `cp[n]` is the chest pain type or category of subject n, `beta` is a matrix containing the $\beta_{j,i}$ coefficients, and `beta_0` and `x[n]` are as in the non-hierarchical model.

3.2.2 Priors

The prior choosing logic is the same as with the previous model: we do not have knowledge about the field or information from other studies, so we choose weakly informative priors for all the "active" input variables in the hierarchical model. However this time the $\beta_{j,i}$ priors utilize hyperpriors: the mean of the normal distribution of beta has a wide normal distribution prior, and the variance has a wide non-negative inverse-chi-squared prior. Therefore the priors are:

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

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

$$\beta_{j,i} \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

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:

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

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

$$\beta_{j,i} \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

In Stan, the priors and hyperpriors for β_0 are defined with

```
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);
```

and for all the selected variables j, the hyperpriors are defined with

```
mu[j] ~ normal(0, 100);
sigma[j] ~ inv_chi_square(0.1);
```

and for all the non-selected variables k, the hyperpriors are defined with

```
mu[k] ~ normal(0, 0.01);
sigma[k] ~ inv_chi_square(100);
```

Finally, the priors for all $\beta_{j,i}$ are defined using the hyperpriors with

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

3.3 Running the models

The models were run with the following commands:

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

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

where standata is a list that contains the data vectors and parameters that the Stan model uses. The options used for Stan were 4 chains, 2000 iterations with 1000 warmup iterations, and **adapt_delta** = 0.999. **adapt_delta** is the only option that was changed from the default options. It was increased from 0.8 to 0.999 due to divergent transitions after the warmup, as is discussed in section 4.1.

4 Results

4.1 Convergence diagnostics

Using the default Stan options (4 chains, 2000 iterations, 1000 warmup iterations, adapt_delta = 0.8) both of the models had some divergent transitions after the warmup, and convergence was therefore not good initially. This was due to too large step sizes in the sampler, and was corrected by increasing the adapt_delta -parameter to 0.999.

With the new settings, we achieved good convergence. For both models, there were no more divergent transitions, and all R-Hats were really close to 1 (under 1.005), which means that the chains have most likely converged and the estimates are reliable. The maximum tree depths were not exceeded. The effective sample sizes were way larger than the suggested minimum of 100 times the amount of chains, and about the same magnitude as the number of draws, which means that the within-chain autocorrelation does not increase the uncertainty of the estimates. The estimates and diagnostic values for β_0 = CP, the selected variables β_1 = Age, β_2 = Sex and β_8 = Oldpeak, and one of the non-selected variables β_3 = Trestbps are presented in table 2 for the non-hierarchical model and table 3 for the hierarchical model for chest pain category 1. All of the Rhat, Bulk ESS and Tail ESS values are plotted in figures 5, 6a and 6b, respectively.

Table 2: Logistic Regression Model convergence diagnostics

	mean	mean MCSE	sd	sd MCSE	R-hat	Bulk ESS	Tail ESS
$\beta_{0,1}$	0.27	9.505542e-03	0.34	6.722931e-03	1.0039932	1322	2109
$\beta_{0,2}$	2.2	1.307897e-02	0.56	9.645218e-03	1.0008069	1898	2676
$\beta_{0,3}$	2.6	1.184722e-02	0.46	8.594848e-03	1.0030909	1528	2253
$\beta_{0,4}$	2.6	1.554311e-02	0.7	1.111951e-02	1.0023656	1934	2615
β_1	-0.44	3.122194e-03	0.18	2.353657e-03	1.0010305	3299	2632
β_2	-1.7	1.206472e-02	0.39	8.730644e-03	1.0038906	1083	1666
β_8	-0.10	2.987844e-03	0.19	2.157243e-03	1.0001320	4210	3106
β_3	-0.001	1.446791e-04	0.010	1.642104e-04	1.0008201	4879	2867

Table 3: Hierarchical Logistic Regression Model convergence diagnostics

	mean	mean MCSE	sd	sd MCSE	R-hat	Bulk ESS	Tail ESS
$\beta_{0,1}$	0.26	7.654690e-03	0.39	5.413303e-03	1.0030036	2543	3305
$\beta_{0,2}$	2.0	1.485986e-02	0.7	1.102780e-02	1.0009131	2399	2008
$\beta_{0,3}$	2.9	2.383852e-02	0.8	1.813283e-02	1.0021522	1310	1167
$\beta_{0,4}$	2.3	2.816077e-02	1.0	2.432340e-02	1.0025346	1688	1149
$\beta_{1,1}$	-0.30	3.473952e-03	0.23	8.249372e-03	1.0029528	2429	1894
$\beta_{2,1}$	-1.60	9.262687e-03	0.4	1.750979e-02	1.0025252	1591	1413
$\beta_{8,1}$	-1.14	4.175458e-03	0.3	1.220688e-02	1.0017452	2418	1907
$\beta_{3,1}$	-0.001	3.008925e-04	0.015	1.672952e-04	1.0004782	1851	2531

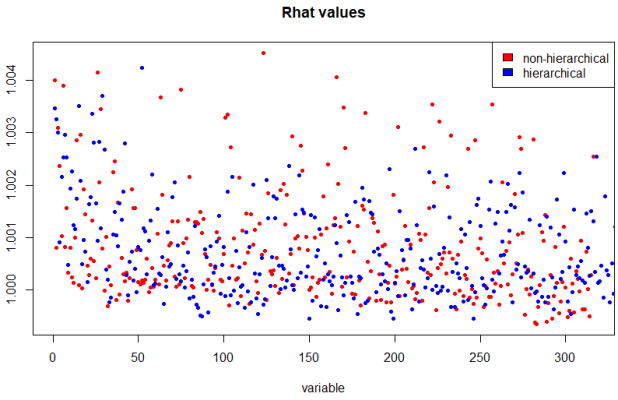
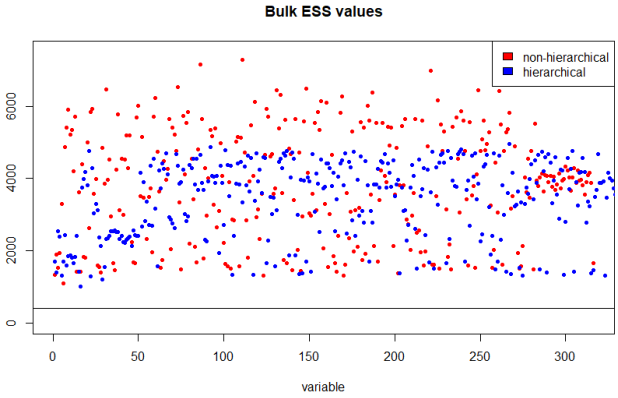
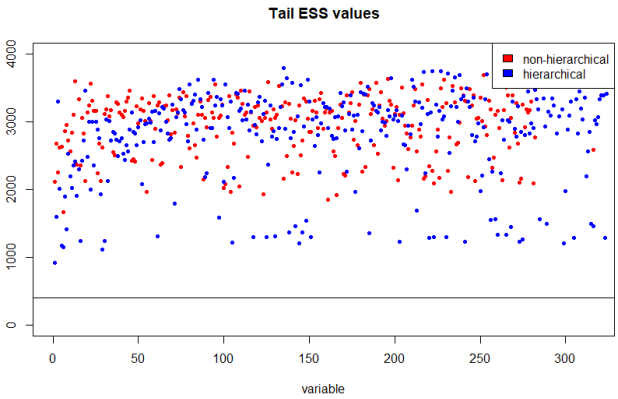


Figure 5: Rhat values for both models. Values close to 1 indicate convergence.



(a) Bulk ESS values



(b) Tail ESS values

Figure 6: Effective sample sizes for both models. The horizontal lines represent the minimal good ESS treshold of 100 per chain.

4.2 Posterior predictive checks

Posterior predictive checks were performed by comparing the distribution of the observed heart diseases and the posterior predictive distributions. The observed heart diseases and posterior predictive draws are in figure 7.

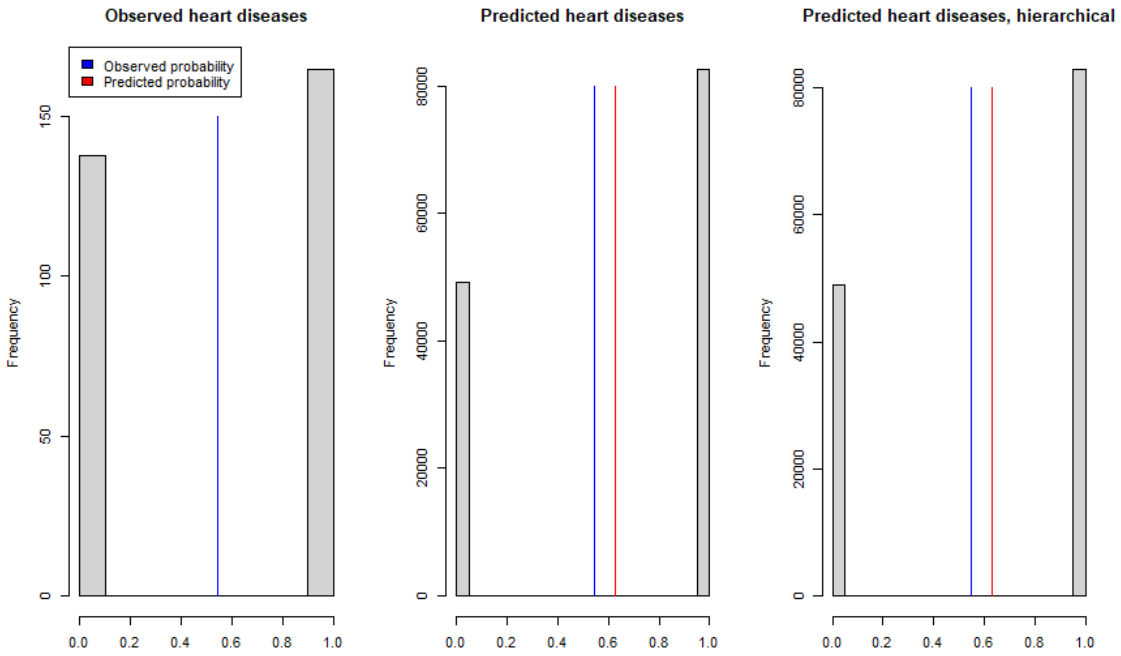


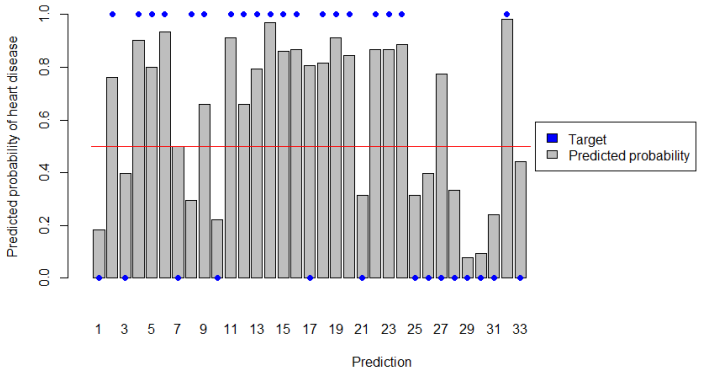
Figure 7: Observed heart diseases and posterior predictive draws

The models predict slightly too many heart diseases, but overall the observed and predicted distributions are pretty close, so the models are reasonable. There is almost no difference between the hierarchical and non-hierarchical models here.

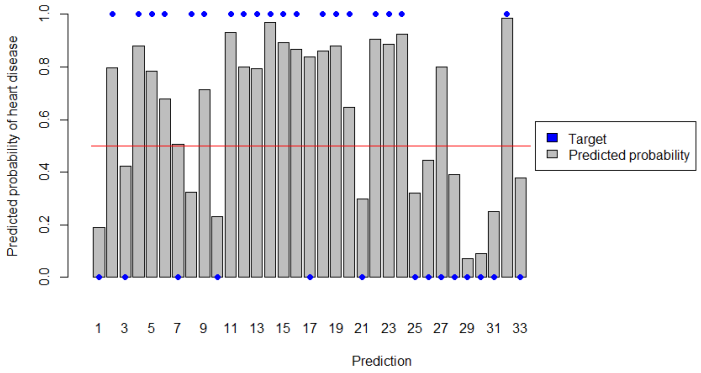
4.3 Predictive performance

The predictive performance of the models was checked by dividing the 303 row data set into a 270 row training set and a 33 row test set, and making predictions for the target variable from the posterior predictive distribution using the test set, and comparing the real and predicted target values. Figures 8a and 8b contain the real target values and means of the predicted target values for each test item. They can be interpreted as the predicted probabilities of a heart disease for the test subjects.

If we classify all predictions where the mean is equal to or over 0.5 to have a heart disease and the rest to not have one, for the non-hierarchical model 30 of the 33 predictions are correct resulting in a 91% predictive accuracy, and for the hierarchical model 29 of the 33 predictions are correct, resulting in a 88% predictive accuracy. The models perform similarly well, and the only difference in classification comes from the seventh prediction, where the probability of heart disease is about



(a) Non-hierarchical model predictions



(b) Hierarchical model predictions

Figure 8: Predicted probabilities of the subjects having a heart disease

50% for both models, so the difference is minimal.

The predictions were made multiple times with different randomly divided test sets to make sure the result is not due to chance. All of the runs yielded approximately similar results, but there was some variation, with most runs having 0-2 predictions less correct. It is possible that we got lucky with the small data set and test set sizes, and that the real classification accuracy is slightly lower. To verify this, proper cross-validation should be done with the test sets, but unfortunately RStudio crashing every time the models were run more than just a couple of times during the same session prevented us from doing this efficiently, so we were left with only the few manual cross-validation runs.

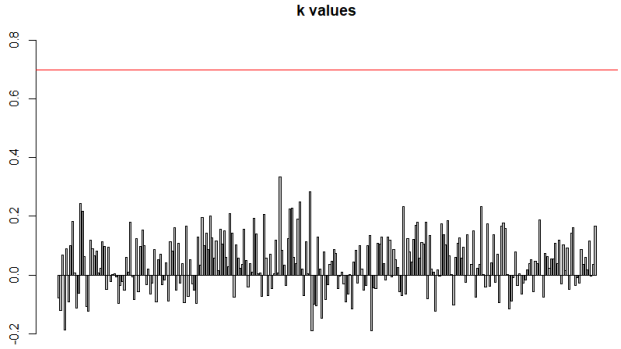
With the real classification accuracy probably being between 80% and 90%, the accuracy is good, but not nearly good enough to make a diagnosis based on the prediction alone in a real situation, for example.

4.4 Model comparison with LOO-CV

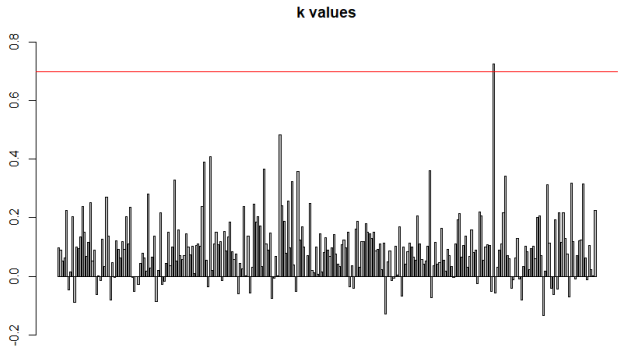
To compare the models, leave-one-out cross-validation was performed. The elpd_loo and p_loo values, and the amount of pareto-k values that are over 0.7 are in table 4, and all of the k-values are plotted in figure 9.

Table 4: LOO-CV diagnostics

model	elpd_loo	p_loo	$\|k > 0.7\|$
non-hierarchical	-127.1614	7.506396	0
hierarchical	-128.7524	12.74573	1



(a) Non-hierarchical model k-values



(b) Hierarchical model k-values

Figure 9: Pareto k values

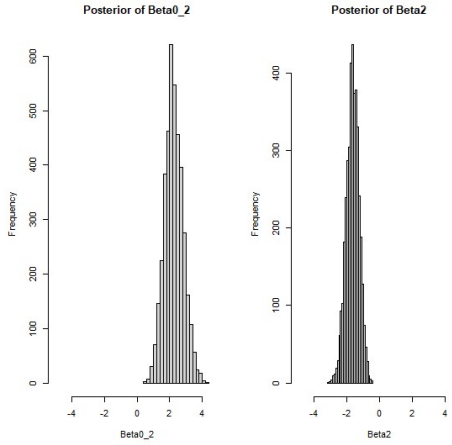
The elpd_loo values of the non-hierarchical model is slightly larger than of the hierarchical model. Additionally, one k-value of the hierarchical model is over 0.7, which indicates that its elpd_loo estimate might be too optimistic. Therefore, based on LOO-CV, the non-hierarchical model is slightly better in this case.

4.5 Prior sensitivity analysis

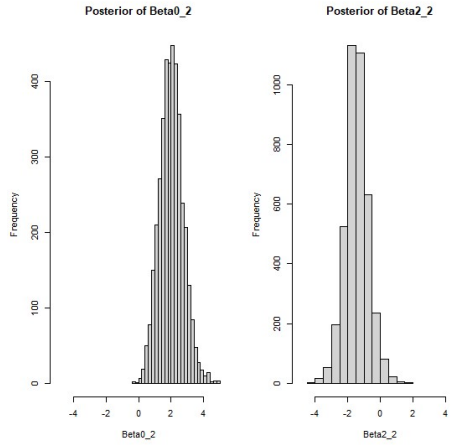
We performed prior sensitivity analysis for the model by running it with two sets of alternative priors for the selected variables. Our original prior for the non-hierarchical model was $\beta_j \sim \mathcal{N}(0, 100^2)$, and the alternative priors were a slightly more informative $\beta_j \sim \mathcal{N}(0, 1^2)$ prior and a shifted $\beta_j \sim \mathcal{N}(2, 10^2)$ prior. For the hierarchical model, the original priors were $\mu_j \sim \mathcal{N}(0, 100^2)$ and $\sigma_j \sim inv - \chi^2(0.1)$, and the alternative ones were again more informative $\mu_j \sim \mathcal{N}(0, 1^2)$ and $\sigma_j \sim inv - \chi^2(1)$ priors, and shifted $\mu_j \sim \mathcal{N}(2, 10^2)$ and $\sigma_j \sim inv - \chi^2(1)$ priors.

The effects of the alternative priors on the posteriors are illustrated in figure 10, where we show the posteriors of the coefficients of the second chest pain category, $\beta_{0,2}$, and sex β_2. The first alternative priors make the posteriors a bit narrower, as expected, and shift the mean towards 0 by about 0.5 for the non-hierarchical model, and about 0.3 for the hierarchical model. The effect of the second alternate priors is much smaller, as expected, because they are not as informative as the first one - they don't cause noticeable effects to the posteriors.

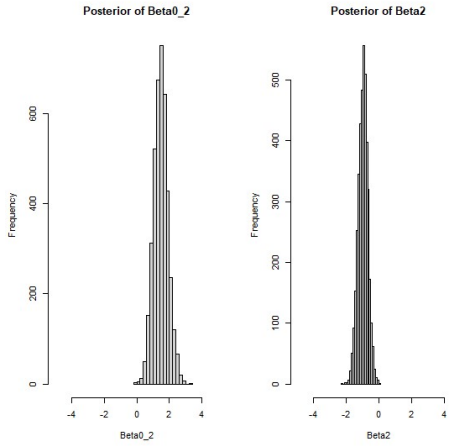
Convergence stays good with the alternative priors, and because the priors are still reasonable and not strongly informative, the predictive capabilities also remain good. Overall, both of the models are robust to small changes in the prior distributions, even though they affect the estimates a bit.



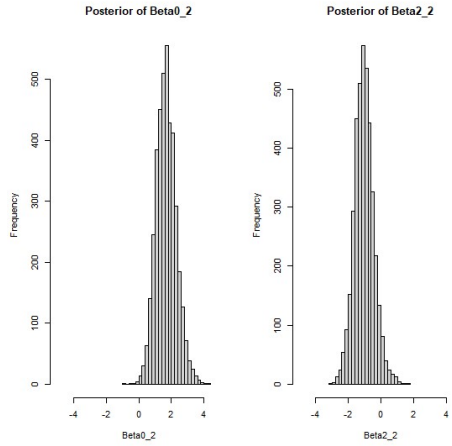
(a) Non-hierarchical model
$\beta_j \sim \mathcal{N}(0, 100^2)$



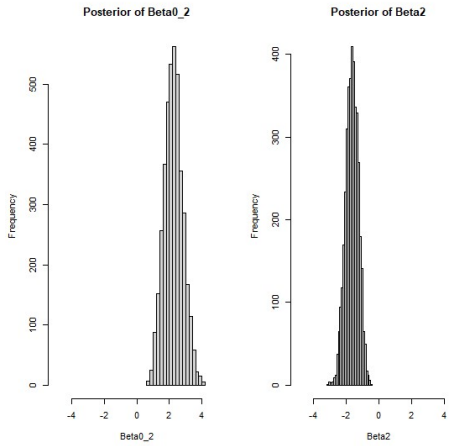
(b) Hierarchical model
$\mu_j \sim \mathcal{N}(0, 100^2)$ and $\sigma_j \sim inv - \chi^2(0.1)$



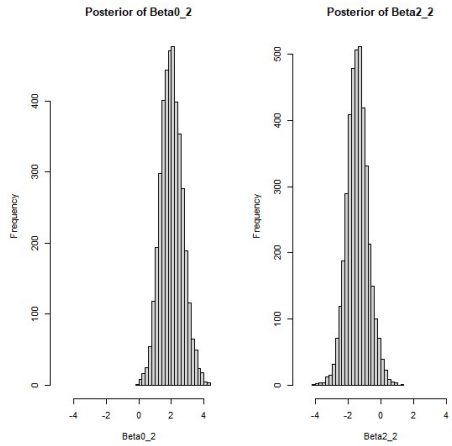
(c) Non-hierarchical model
$\beta_j \sim \mathcal{N}(0, 1^2)$



(d) Hierarchical model
$\mu_j \sim \mathcal{N}(0, 1^2)$ and $\sigma_j \sim inv - \chi^2(1)$



(e) Non-hierarchical model
$\beta_j \sim \mathcal{N}(2, 10^2)$



(f) Hierarchical model
$\mu_j \sim \mathcal{N}(2, 10^2)$ and $\sigma_j \sim inv - \chi^2(1)$

Figure 10: Posterior distributions of $\beta_{0,2}$ and β_2 with alternative priors

5 Conclusion and discussion

The goal of this project was to create two different logistic regression models to predict heart disease. The models predicted heart disease rather well, as we managed to achieve 80% to 90% predictive classification accuracy. The better one of the two models was the non-hierarchical logistic regression model. Even though the model is not accurate enough for serious medical use, we showed that it is possible to apply Bayesian methods to create reasonable models for heart disease prediction.

According to the models, chest pain types 1-3 are good predictors for heart disease, while type 0 is not. Being older actually decreases the probability of having a heart disease, which is counter-intuitive. Since all of the subjects are hospitalised and not from the general population, one reason for this could be that older subjects are more cautious about heart disease symptoms and have more illnesses in general that cause them to go to the hospital more easily, while younger subjects are rarely hospitalised unless they really have a serious illness. Being male and having a higher Oldpeak also decrease the probability of having a heart disease.

Reducing the amount of variables the models used was a key factor in improving the result. This was due to multicollinearity and a relatively small data set that caused us to not have much data from the different combinations of categories and values, which in turn causes uncertainty about the regression coefficients, which can for example be seen in the large standard deviations in tables 2 and 3. With larger data sets, this curse of dimensionality could be battled and uncertainty removed further, or the model could be improved by including more variables. Outlier detection could also improve the results, especially since outliers can have a huge effect in small data sets.

The group did not have prior knowledge on the medical field or heart disease, so the priors used were general and very weakly informative. With assistance from medical professionals and proper research on the field, more informative and useful priors could be used to improve the model.

Overall, even though the models' prediction accuracies are already good, they are quite simple and general models, meaning that there is lots of room for improvement. This is only a good thing, since it gives motivation for more sophisticated Bayesian models for heart disease prediction.

6 Self-reflection

The group learned a lot during the project, especially about planning and executing a Bayesian data analysis project from scratch. The skills learned during the weekly assignments acted as a good basis, but putting them to use in the project helped to really internalize the theory and methods. The group also learned about new topics, such as multivariate models and shrinkage priors. Finding good data, preprocessing, analyzing, building a model, validating the model and making predictions is an important workflow that we've learned and which will surely be useful in the future even outside Bayesian modeling. Of course, the project also strengthened our skills in the Bayesian methods and topics, such as convergence analysis, internal and external model validation, model comparison and prior sensitivity analysis.

A Appendix

A.1 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]);
}
```

A.2 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]);
}
```