Haberman's Survival Prediction

Using Bayesian Logistic Regression

INTRODUCTION

In this project Bayesian logistic regression is applied to find the linear decision boundary that can predict the survival of patients who had undergone surgery for breast cancer. The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

Data is collected from https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival. It contains 306 observations each having 3 predictors of numeric type and one dependent binary variable.

It contains the following attributes:

1. Age - Age of patient at time of operation - predictor
2. Patient's year of operation. - Predictor
3. Number of positive axillary nodes detected - Predictor
4. Survival status (1 is survived 5 year or longer 2 otherwise) - Dependent Variable

DATA CLEANING AND PRE-PROCESSING

```
dat <- read.csv("haberman.data")
dat$X64         <-NULL          names(dat)                    <-
c("Age","PosAuxNodes","Survived") dat$Age   <-(dat   $Age
-mean  (dat$Age))/sd(dat$Age)
dat$PosAuxNodes    <-(dat    $PosAuxNodes    -mean    (dat$PosAuxNodes))/sd(dat$PosAuxNodes)
dat$Survived[dat$Survived == 2] <-0
```

Patient's year of operation may not provide any meaningful information to the model hence can be removed from the feature set. Also, Survival encoding has been changed i.e.. for non-survival coding is replaced from 2 to 0.

BAYERSIAN LOGISTIC MODEL

The linear combination of predictors can form a linear model which is given as the input to the logistic function which gives probability between 0-1. Bernoulli function takes this probability and decides the label for output variable.   This logistic model is represented by the following equation:

$$Y \sim Bernoulli(\text{logistic}(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k)),$$

Figure 1: Bayesian Model

$$\text{logistic}(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k) = \frac{1}{[1 + \exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)]}.$$

Figure 2: Logit

where,

Each coefficient of prediction and including intercept term, the prior distribution of these are assumed to follow normal distribution with mean 0 and standard deviation of 1. As it is assumed none of the predictor is important for this model. Once Bayesian model is fitted it gives the posterior distribution which can be used in determining the importance of corresponding predictor.

The prior distribution is shown using the JAG's figure:

Dependent variable Y follows Bernoulli distribution based on the probability return by logit function. Logit function takes linear combinations of coefficient and predictive variables along with intercept term.

JAG's MODEL

As the prior distribution are defined, now to calculate the posterior distribution. It is determined

```
cat("model { for ( i in 1:Ntotal ) {
        # In JAGS, ilogit is logistic: y[i] ~ dbern( ilogit( zbeta0 + sum( zbeta[1:Nx] *
           zx[i,1:Nx] ) ) )
        }
        # Priors vague on standardized scale: zbeta0
        ~ dnorm( 0 , 1/2^2 ) for ( j in 1:Nx ) { zbeta[j]
        ~ dnorm( 0 , 1/2^2 )
        }
        # Transform to original scale:
        beta[1:Nx] <- zbeta[1:Nx] / xsd[1:Nx]
        beta0 <- zbeta0 - sum( zbeta[1:Nx] * xm[1:Nx] / xsd[1:Nx] ) }",file="jag.model.txt")


x   =dat[,1   :2]   xm      =
apply(x,2,  mean)
xsd  =  apply(x,2,  sd)
df <- list("Ntotal"= nrow(dat),
               "y"=dat $Survived,
               "zx"=x,
               "Nx"=2,
               "xm"=xm,
               "xsd"=xsd)


bayes.mod.params     <-     c("beta0","beta[1]","beta[2]")  bayes.mod.fit  <-
jags(data =df, parameters.to.save =bayes.mod.params,
                              n.chains  =3,model.file  =  'jag.model.txt')
```

using MCMC chain simulation by applying JAG's model.

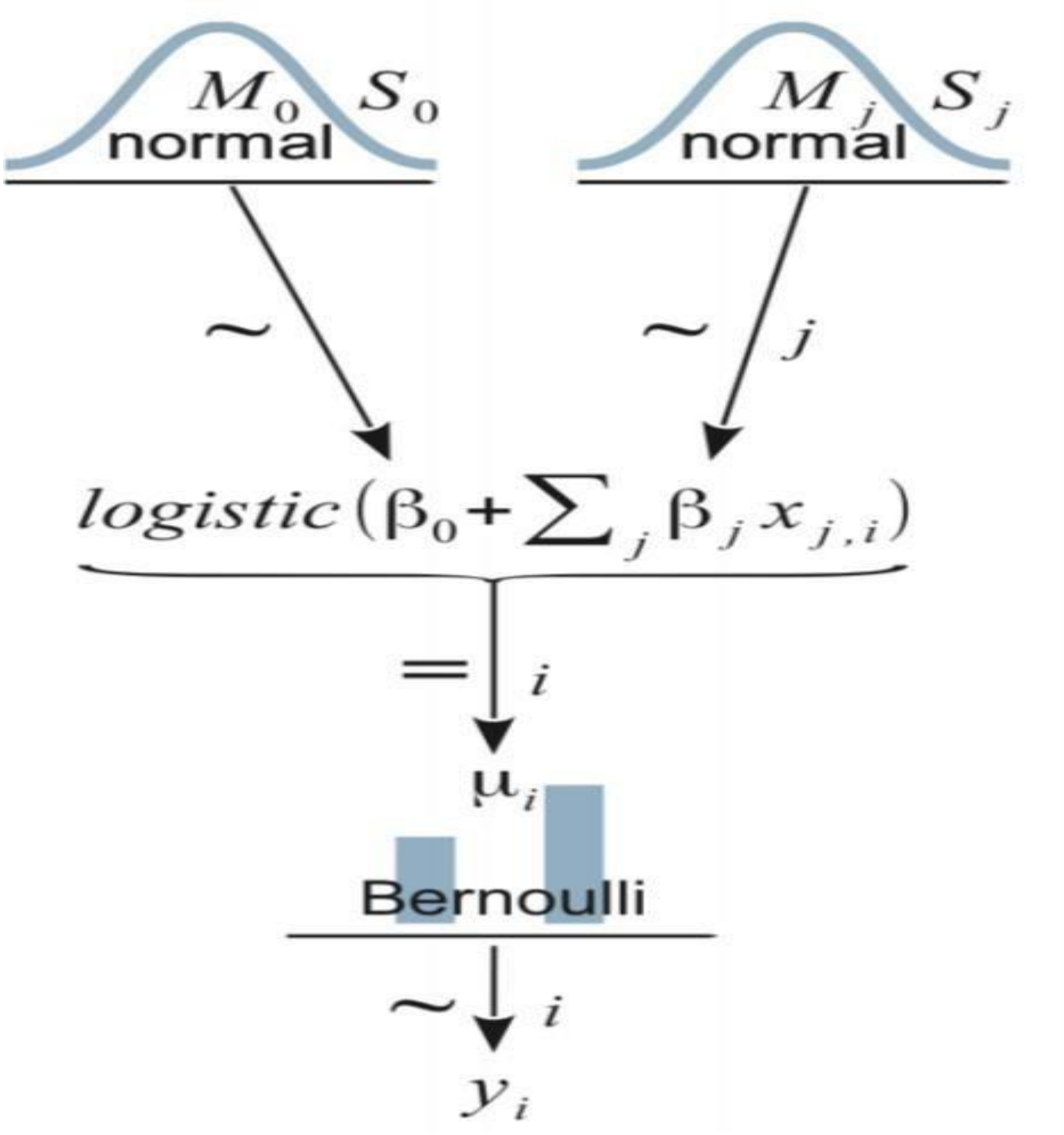
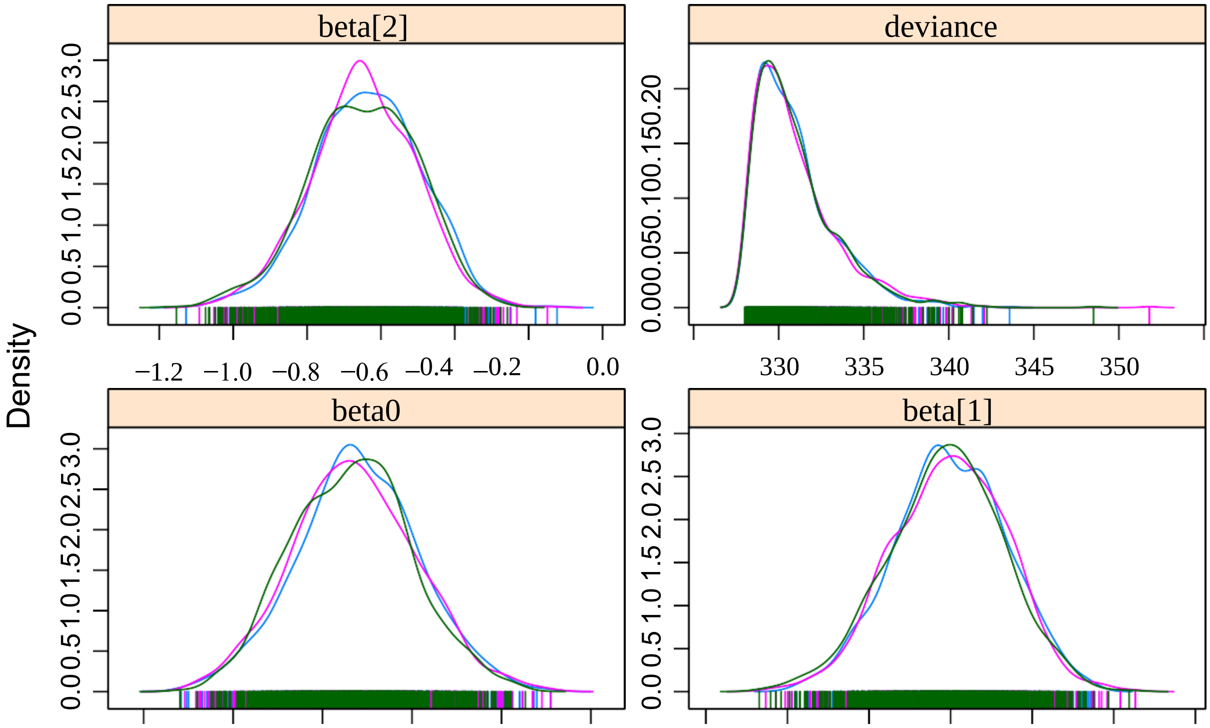


Figure 3: JAG's Model

```
bayes.mod.fit.mcmc <- as.mcmc(bayes.mod.fit)
```

D

```
densityplot(bayes.mod.fit.mcmc,layout= c(2,2),aspect="fill")
```



0.6 0.8 1.0 1.2 1.4 1.6 −0.8 −0.6 −0.4 −0.2 0.0 0.2

From the posterior distribution it is observed that each coefficient mean is not close to 0. It means each predictor is important in determining the survival of the patient.

```
v=posterior.mode(bayes.mod.fit.mcmc)
```

```
## Warning in  posterior.mode(bayes.mod.fit.mcmc):  posterior.mode  expecting
## mcmc object
```

```
v
```

```
##     beta0   beta[1]           beta[2]
        deviance ## 1.0834552 -0.1833558   -
0.6557318 328.7818425
```

Taking mode as the coefficient to find the linear model. It is observed that there is inverse relation with the predictors and survival of the variable.

This linear equation can be used to make predictions when given as input to logistic function

```
sigmoid <- function(z)
{
```

followed by bernoulli function.

```
g  <-1  /(1+exp(-z))
return(g)
}

cost <- function(theta)
{
m <- nrow(X)
g <- sigmoid(X *         theta)
J  <-(1  /m)*sum((-Y*log(g))  -  ((1-Y)*log(1-g))) return(J)
}

w1 =v[1 :3]
x1  =  cbind(rep(1,  nrow(x)),  x) y1  =1  /(1+exp  (-
as.matrix(x1)   *           as.matrix  (w1)))

y11  =  ifelse(y1  >  0.5,1,0)
confusionMatrix(y11, dat$Survived,positive ="1")
```

```
  ## Confusion Matrix and Statistics
  ##
##                        Reference
  ##   Prediction 0 1
##                         0 13 10
##                        1 68 214
  ##
##                                   Accuracy : 0.7443
##                                     95%        CI : (0.6914, 0.7923)
  ##      No  Information  Rate  : 0.7344
  ##     P-Value  [Acc  >  NIR]  : 0.3764
  ##
##                                        Kappa : 0.1502
  ## Mcnemar's Test P-Value : 1.09e-10
  ##

  ##                                Sensitivity : 0.9554
  ##                                Specificity : 0.1605
  ##                          Pos Pred Value : 0.7589
  ##                         Neg Pred Value : 0.5652
  ##                                Prevalence : 0.7344
  ##                           Detection Rate : 0.7016
  ##       Detection Prevalence : 0.9246
  ##                    Balanced Accuracy : 0.5579
  ##
  ##                'Positive' Class : 1
  ##
```

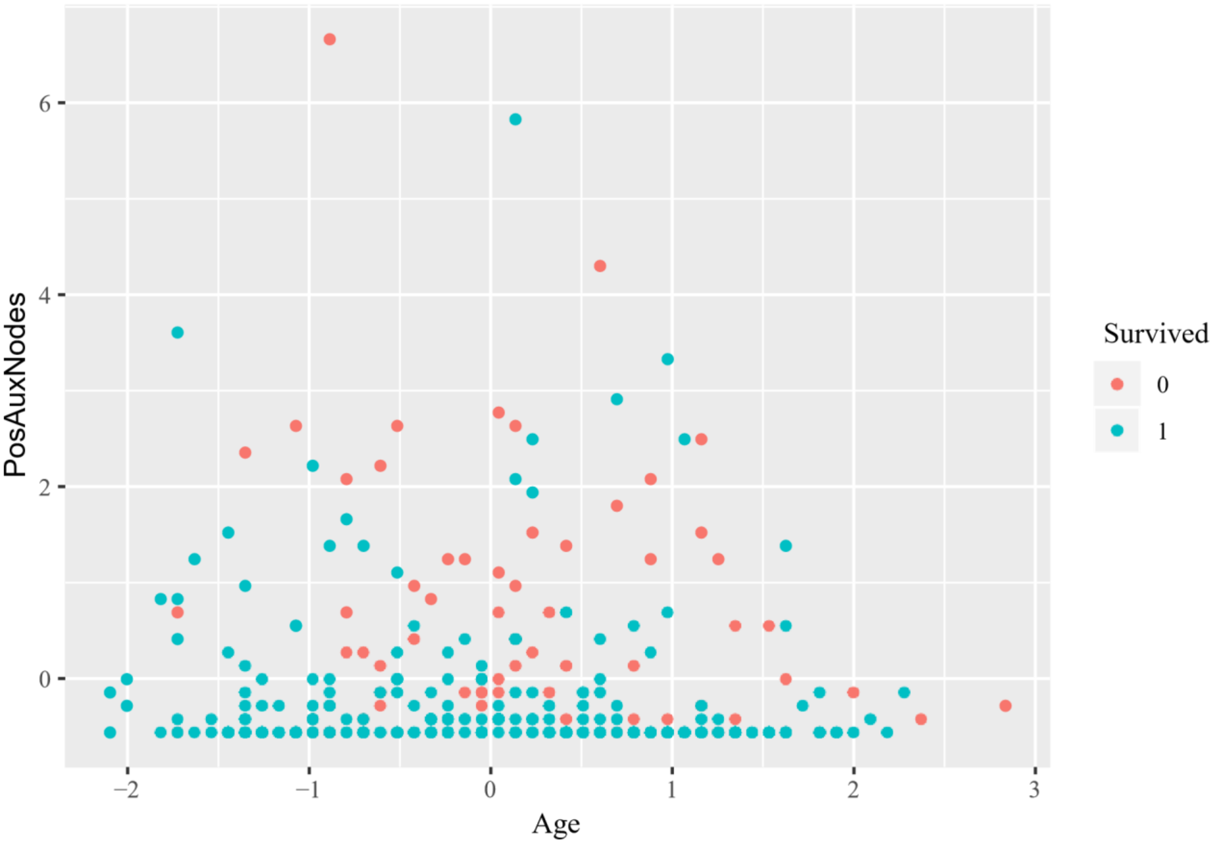
REFITTING THE MODEL WITH GUESS PARAMETER

As bayesian model as uneffected from the outliers in the data, the presence of outlier in the data is

```
dat$Survived <- as.factor(dat$Survived) g  <- ggplot(data =dat, aes(x =Age,y
=PosAuxNodes,col =Survived))
```

explored and model is revised accordingly.

```
g +geom_point()
```



From the plot, it is clear that there are few outlier points. Hence, the model needs to be updated. We have introduced guess term in estimating the label in case when it fails to predict from the model.

Revised JAG’s model is shown below:

```
cat("

model { for ( i in 1:Ntotal ) { # In JAGS, ilogit is logistic: y[i] ~ dbern(guess * (1/2) + (1-guess) * ilogit( zbeta0 +
        sum( zbeta[1:Nx] * zx[i,1:Nx] ) )
            } guess ~ dbeta(1,
        9)
        # Priors vague on standardized scale:
        zbeta0 ~ dnorm( 0 , 1/2^2 ) for ( j in
        1:Nx ) {
        zbeta[j] ~ dnorm( 0 , 1/2^2 )
        }
        # Transform to original scale: beta[1:Nx] <-
        zbeta[1:Nx] / xsd[1:Nx]
        beta0 <- zbeta0 - sum( zbeta[1:Nx] * xm[1:Nx] / xsd[1:Nx] )
        }

        ",file="jag.model.guess.txt")
```

)

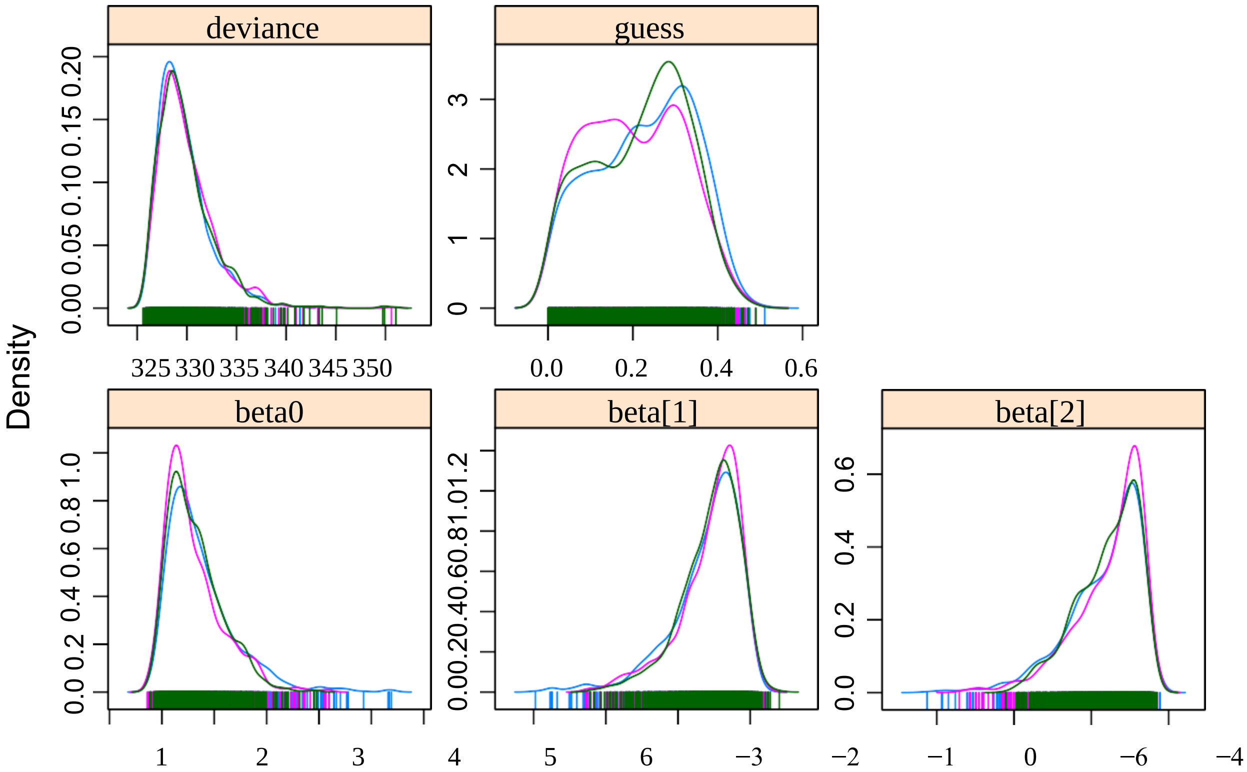
```
bayes.mod.params <- c("beta0","beta[1]","beta[2]","guess") bayes.mod.fit <-
jags(data =df, parameters.to.save =bayes.mod.params,
                                          n.chains =3,model.file = 'jag.model.guess.txt')
```

```
## Compiling model graph
##       Resolving undeclared variables ##
         Allocating
nodes
## Graph information:
## Observed stochastic nodes: 305   ##
Unobserved  stochastic  nodes:  4     ##
Total graph size:
2351
##
## Initializing model
```

```
bayes.mod.fit.mcmc <- as.mcmc(bayes.mod.fit)
```

DIAGNOSTICS

```
densityplot(bayes.mod.fit.mcmc,layout=  c(3,2),aspect="fill")
```



−2 0 *#densplot(bayes.mod.fit.mcmc)*

```
v2 = posterior.mode(bayes.mod.fit.mcmc)
```

```
## Warning in posterior.mode(bayes.mod.fit.mcmc): posterior.mode expecting.## mcmc object
```

From the posterior distribution it is clear that guess plays an important role, it guessed

```
w2 =v2[1 :3]
x1  =  cbind(rep(1,  nrow(x)),  x)
y2  =1  /(1+exp  (-as.matrix(x1)        * as.matrix  (w )2y22
=  ifelse(y2  >  0.5,1,0)
confusionMatrix(y22, dat$Survived)
```

approximately 25% times when it failed to estimate the true label.

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction 0 1
##                0 17 10
##                1 64 214
##
##                         Accuracy : 0.7574
##                         95%□     CI : (0.7053, 0.8044)
##    No  Information  Rate  : 0.7344
##    P-Value  [Acc  >  NIR]  : 0.2005
##
##                            Kappa : 0.2099
## Mcnemar's Test P-Value : 7.223e-10
##

##                          Sensitivity : 0.20988
##                          Specificity : 0.95536
##                       Pos Pred Value : 0.62963
##                       Neg Pred Value : 0.76978
##                           Prevalence : 0.26557
##                       Detection Rate : 0.05574
##    Detection Prevalence : 0.08852
##                    Balanced Accuracy : 0.58262
##
##            'Positive' Class : 0
##
```

From the refined model, both accuracy and kappa statistics have been improved. This shows data has outliers which this model is capable to handle.

REFERENCES

- https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival
- Denver Dash and Gregory F. Cooper. Model Averaging with Discrete Bayesian Network Classifiers. Decision Systems Laboratory Intelligent Systems Program University of Pittsburgh

- Tierney, L. (1994). Markov chains for exploring posterior distributions. The Annals of Statistics 22, 1701-1728
- Silvapulle, M. J. (1981). On the existence of maximum likelihood estimators for the binomial response models. Journal of the Royal Statistical Society, Series B 43, 310-313.