Scatter plot between Petal.Width and Petal.Length by species suggests that setosa is distinguishable. Versicolor and virginica are adjacent but still separable.

I created two dummy variables as below

iris**$**class1\_dummy = **ifelse**(iris**$**Species **==** "setosa", 1, 0) iris**$**class2\_dummy = **ifelse**(iris**$**Species **==** "virginica", 1, 0)

To apply logistic regression, I normalized predictive variables.

iris\_normalized = iris

X = **scale**(iris[1**:**4], center=TRUE, scale=TRUE)

X = **cbind**(X, class1\_dummy = iris**$**class1\_dummy, class2\_dummy = iris**$**class2\_dummy)

I followed the lesson 9 example code as below.

**library**("rjags")

## Loading required package: coda ## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

**set.seed**(725)

mod1\_string = " model { for(i in 1:length(y)) {

y[i] ~ dbern(p[i])

logit(p[i]) = int + b[1] \* Sepal.Length[i] + b[2] \* Sepal.Width[i]

+ b[3] \* Petal.Length[i] + b[4] \* Petal.Width[i]

}

int ~ dnorm(0.0, 1.0/25.0) for(j in 1:4) {

b[j] ~ ddexp(0.0, sqrt(2.0))

}

}

"

data1\_jags = **list**(y = X[, "class1\_dummy"],

Sepal.Length = X[,"Sepal.Length"], Sepal.Width = X[, "Sepal.Width"], Petal.Length = X[, "Petal.Length"], Petal.Width = X[, "Petal.Width"] )

params = **c**("int", "b")

model1 = **jags.model**(**textConnection**(mod1\_string), data=data1\_jags, n.chains = 3)

**update**(model1, 1e3)

mod1\_sim = **coda.samples**(model=model1, variable.names = params, n.iter = 5e3) mod1\_csim = **as.mcmc**(**do.call**(rbind, mod1\_sim))

Gelman and Rubin’s convergence diagnostic shows that model is converged. Even though there is strong autocorrelation, the effective sample size is sufficiently large.

The penalized deviance of DIC is as below.

## Mean deviance: 4.315 ## penalty 2.911

## Penalized deviance: 7.226

The dense plot suggests that the mean of Sepal.Width and Petal.Length is not equal to zero. After fitting the model with Sepal.Width and Petal.Length variables,

the penalized deviance of the DIC is higher than the first model. On the other hand, glm() of R does not converge if I fit the model with all 4 variables. Therefore I use the second prediction model.

## Mean deviance: 4.709

## penalty 2.553

## Penalized deviance: 7.262

The second model shows that 100% accuracy for classifying setosa and non-setosa.

(tab0.5\_setosa = **table**(p\_hat\_setosa **<** 0.5, X[, "class1\_dummy"]))

##

## 0 1

## FALSE 100 0

## TRUE 0 50

## 

I followed the same procedure to get parameters for classifying virginica and non-virginica.

mod4\_string = " model { for(i in 1:length(y)) {

y[i] ~ dbern(p[i])

logit(p[i]) = int + b[1] \* Petal.Length[i] + b[2] \* Petal.Width[i]

}

int ~ dnorm(0.0, 1.0/25.0) for(j in 1:2) {

b[j] ~ ddexp(0.0, sqrt(2.0))

}

}

"

data4\_jags = **list**(y = X[, "class2\_dummy"],

Petal.Length = X[, "Petal.Length"], Petal.Width = X[, "Petal.Width"] )

params = **c**("int", "b")

model4 = **jags.model**(**textConnection**(mod4\_string), data=data4\_jags, n.chains = 3)

**update**(model4, 1e3)

mod4\_sim = **coda.samples**(model=model4, variable.names = params, n.iter = 5e3) mod4\_csim = **as.mcmc**(**do.call**(rbind, mod4\_sim))

The confusion matrix for this model shows 95% accuracy.

##

## 0 1

## FALSE 97 4

## TRUE 3 46

Lastly, I checked the classification performance for virginica data. The accuracy of virginica classification is 100%

versicolor\_df = iris\_normalized[iris\_normalized**$**Species **==** 'versicolor', ]



X\_versicolor = **as.matrix**( versicolor\_df[2**:**3] )

X\_pred\_setosa = coef\_setosa["int"] **+** X\_versicolor **\*** coef\_setosa[1**:**2]

p\_hat\_setosa = 1.0 **/** (1.0 **+ exp**(X\_pred\_setosa)) **table**(p\_hat\_setosa **<** 0.5, versicolor\_df[, "class1\_dummy"])

##

## 0

## FALSE 50

X\_versicolor = **as.matrix**( versicolor\_df[3**:**4] )



X\_pred\_virginica = coef\_virginica["int"] **+** X\_versicolor **\*** coef\_virginica[1**:**2] p\_hat\_virginica = 1.0 **/** (1.0 **+ exp**(X\_pred\_virginica))

**table**(p\_hat\_virginica **>** 0.5, versicolor\_df[, "class2\_dummy"]) ##

## 0

## FALSE 50

## 

glm() and aic suggests different features to fit. For example, setosa and non-setosa classification, glm() suggests Sepal.Length is the most significant feature while our previous models selected Sepal.Width and Petal.Length

**for**(i **in** 1**:**4) {

feature\_name = **colnames**(X)[i] **tryCatch** (

{

model = **glm**(X[, "class1\_dummy"] **~** X[,i], family="binomial")

**print**(**paste**(feature\_name, " Deviance :", model**$**deviance, "AIC", model**$**aic))

},

warning = **function**(w){ **print**(**paste**(feature\_name, ": ", w))

}

)

}

## [1] "Sepal.Length Deviance : 71.8363992272217 AIC 75.8363992272217"

## [1] "Sepal.Width Deviance : 123.828029803175 AIC 127.828029803175"

## [1] "Petal.Length : simpleWarning: glm.fit: algorithm did not converge\n" ## [1] "Petal.Width : simpleWarning: glm.fit: algorithm did not converge\n"

# 