Assignment 8 Code + Analysis

Code with comments for default case:

```
# loading randomForest into R environment
library(randomForest)
# reading in csv file with pulsar data
Pulsar<-read.csv("pulsar.csv")
# creating a data frame of pulsar data with everything except the first column
Pulsar1<-Pulsar[,-1]
# makes Pulsar1 a factor variable so that it could be categorical
Pulsar1$v9<-as.factor(Pulsar1$v9)
# subsetting Pulsar1 to when values in 9th column are equal to 0 and stores it into
Pulsar1base since 0 represents false
Pulsar1base<-Pulsar1[Pulsar1[,9]==0,]
# subsetting Pulsar1 to when values in 9th column are equal to 1 and stores it into
Pulsar1true since 1 represents true
Pulsar1true<-Pulsar1[Pulsar1[,9]==1,]
# generate random sample of 100 values from integers 1 to 1638
v1<-sample(1638,100)
# generate random sample of 1000 values from integers 1 to 16529
v2<-sample(16259,1000)
# binds indices specified in v2 for Pulsar1base with rows of indices in v1 for Pulsar1true
Pulsartry<-rbind(Pulsar1base[v2,],Pulsar1true[v1,])
# creates a logistic regression model for the data in Pulsartry data where v9 is the
response variable and every other variable is a predictor
pulsar.logit<-glm(v9~.,Pulsartry,family=binomial(link="logit"))
# creates a random forest model for the data in Pulsartry data where v9 is the response
variable and every other variable is a predictor
# has 5000 trees in the forest
pulsar.rf<-randomForest(v9~.,data=Pulsartry,ntree=5000)
# generates predictions from logistic regression model
p1<-predict(pulsar.logit,type="response",Pulsar1)
# generates predictions from random forest model
p2<-predict(pulsar.rf,Pulsar1)

#making a 2x2 table on what we are predicting for the logistic model
sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
```

```
#making a 2x2 table representing our predictions for the random forest model
sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))

#what the actual values should be for the prediction
# baselines
sum(as.numeric(Pulsar1$v9)-1)
sum(1-(as.numeric(Pulsar1$v9)-1))
```

Default Case: 100 pulsar vs 1000 nonpulsar

```
Pulsar<-read.csv("pulsar.csv")
Pulsar1<-Pulsar[,-1]
Pulsar1$v9<-as.factor(Pulsar1$v9)
Pulsar1base<-Pulsar1[Pulsar1[,9]==0,]
Pulsar1true<-Pulsar1[Pulsar1[,9]==1,]
v1<-sample(1638,100)
v2<-sample(16259,1000)
Pulsartry<-rbind(Pulsar1base[v2,],Pulsar1true[v1,])
pulsar.logit<-glm(v9~.,Pulsartry,family=binomial(link="logit"))
pulsar.rf<-randomForest(v9~.,data=Pulsartry,ntree=5000)
p1<-predict(pulsar.logit,type="response",Pulsar1)
p2<-predict(pulsar.rf,Pulsar1)
sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
sum(as.numeric(Pulsar1$v9)-1)
sum(1-(as.numeric(Pulsar1$v9)-1))
```

First run (Tanvi):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1297
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 342
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 78
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16181
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1348
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 291
```

```
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 86
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16173
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Second run (Nithya):

```
> #making a 2x2 table on what we are predicting for the logistic model
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1301
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 338
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 71
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16188
```

```
> #making a 2x2 table representing our predictions for the random forest model
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1366
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 273
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 102
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16157
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Third Run (Raashi)

```
> #making a 2x2 table on what we are predicting for the logistic model
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1352
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 287
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 94
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16165
>
> #making a 2x2 table representing our predictions for the random forest model
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1381
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 258
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 108
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16151
>
```

Fourth Run (Shivali)

```
> #making a 2x2 table on what we are predicting for the logistic model
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1478
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 161
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 308
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15951
>
> #making a 2x2 table representing our predictions for the random forest model
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1518
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 121
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 400
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15859
>
> #what the actual values should be for the prediction
> # baselines
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fifth Run (Srimathi)

```
> #making a 2x2 table on what we are predicting for the logistic model
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1364
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 275
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 116
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16143
> #making a 2x2 table representing our predictions for the random forest mode
l
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1375
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 264
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 117
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 16142
> #what the actual values should be for the prediction
> # baselines
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Analysis for Default Case:

Looking at all of the models for logistic regression and random forest where the sample sizes for both pulsar and non-pulsar data are 100 and 1000 respectively, the runs that are closest to the values 1630, 0, 0, and 16259 are what the best model would be. So, we can see that Shivali's model for random forest is the most accurate out of the ten models, where the false negative value is 71, which is close to 0 and the false positive value is 322, which is the closest without the false negative value increasing. While the other models may not be accurate, they can still be somewhat decent analyses of the dataset.

*** ASSIGNMENT CRITERIA BEGINS HERE*****

Case 1: VALUE 200

200 VS 200 (EVEN)

```
Pulsar<-read.csv("pulsar.csv")
Pulsar1<-Pulsar[,-1]
Pulsar1$v9<-as.factor(Pulsar1$v9)
Pulsar1base<-Pulsar1[Pulsar1[,9]==0,]
Pulsar1true<-Pulsar1[Pulsar1[,9]==1,]
v1<-sample(1638,200)
v2<-sample(16259,200)
Pulsartry<-rbind(Pulsar1base[v2,],Pulsar1true[v1,])
pulsar.logit<-glm(v9~.,Pulsartry,family=binomial(link="logit"))
pulsar.rf<-randomForest(v9~.,data=Pulsartry,ntree=5000)
p1<-predict(pulsar.logit,type="response",Pulsar1)
p2<-predict(pulsar.rf,Pulsar1)
sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
sum(as.numeric(Pulsar1$v9)-1)
sum(1-(as.numeric(Pulsar1$v9)-1))
```

First run (Tanvi):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1506
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 133
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 481
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15778
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1500
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 139
```

```
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 507
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15752
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Second Run (Nithya):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1497
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 142
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 513
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15746
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1524
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 115
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 694
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15565
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Third Run (Raashi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1492
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 147
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 409
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15850
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1499
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 140
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 506
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15753
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fourth Run (Shivali)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1508
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 131
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 586
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15673
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1531
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 108
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 607
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15652
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fifth Run (Srimathi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1509
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 130
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 577
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15682
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1501
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 138
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 533
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15726
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Analysis for Case 1: For Case 1, Raashi's run for the logistic regression seems to be the one that is the most accurate to the actual values. The reason for this being is that 409 is the closest to 0 out of all the 2X2 tables generated for this case. However, there are also other models that come close to this in terms of other values. A considerable model to keep in be would be Shivali's model for logistic regression because those predicted values are the closest to the actual values. Another considerable model to keep in mind would be Tanvi's logistic regression model. We can see that most of the logistic regression models seem to be more accurate in predicting the actual values. While the other models may not be accurate, they can still be somewhat decent analyses of the dataset.

Case 2: 300 pulsar vs 300 non pulsar

```
Pulsar<-read.csv("pulsar.csv")
Pulsar1<-Pulsar[,-1]
Pulsar1$v9<-as.factor(Pulsar1$v9)
Pulsar1base<-Pulsar1[Pulsar1[,9]==0,]
Pulsar1true<-Pulsar1[Pulsar1[,9]==1,]
v1<-sample(1638,300)
v2<-sample(16259,300)
Pulsartry<-rbind(Pulsar1base[v2,],Pulsar1true[v1,])
pulsar.logit<-glm(v9~.,Pulsartry,family=binomial(link="logit"))
pulsar.rf<-randomForest(v9~.,data=Pulsartry,ntree=5000)
p1<-predict(pulsar.logit,type="response",Pulsar1)
p2<-predict(pulsar.rf,Pulsar1)
sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
sum(as.numeric(Pulsar1$v9)-1)
sum(1-(as.numeric(Pulsar1$v9)-1))
```

First run (Tanvi):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1491
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 148
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 501
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15758
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1519
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 120
```

```
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 570
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15689
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
>
```

Second run (Nithya):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1506
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 133
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 694
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15565
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1520
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 119
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 607
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15652
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Third Run (Raashi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1487
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 152
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 378
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15881
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1503
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 136
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 504
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15755
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fourth Run (Shivali)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1495
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 144
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 449
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15810
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1533
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 106
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 646
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15613
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fifth Run (Srimathi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1494
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 145
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 483
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15776
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1496
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 143
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 545
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15714
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Case 2 Analysis: Looking at all of the models for logistic regression and random forest where the sample sizes for both pulsar and non-pulsar data are 200 and 200 respectively, the runs that are closest to the values 1639, 0, 0, and 16259 are what the best model would be. So, we can see that Raashi's model for logistics is the most accurate out of the ten models, where predicted values of 152 and 387 are closest to the actual model. Another considerable model which lines up pretty closely is Shivali's logistic regression model.

Case 3: 400 pulsar vs 400 non pulsar

```
Pulsar<-read.csv("pulsar.csv")
Pulsar1<-Pulsar[,-1]
Pulsar1$v9<-as.factor(Pulsar1$v9)
Pulsar1base<-Pulsar1[Pulsar1[,9]==0,]
Pulsar1true<-Pulsar1[Pulsar1[,9]==1,]
v1<-sample(1638,400)
v2<-sample(16259,400)
Pulsartry<-rbind(Pulsar1base[v2,],Pulsar1true[v1,])
pulsar.logit<-glm(v9~.,Pulsartry,family=binomial(link="logit"))
pulsar.rf<-randomForest(v9~.,data=Pulsartry,ntree=5000)
p1<-predict(pulsar.logit,type="response",Pulsar1)
p2<-predict(pulsar.rf,Pulsar1)
```

sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
sum(as.numeric(Pulsar1$v9)-1)
sum(1-(as.numeric(Pulsar1$v9)-1))

First run (Tanvi):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1494
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 145
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 483
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15776
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1529
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 110
```

```
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 431
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15828
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Second run (Nithya):

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1507
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 132
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 449
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15810
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1526
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 113
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 613
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15646
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Third Run (Raashi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1503
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 136
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 589
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15670
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1549
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 90
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 774
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15485
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

We are evaluating 400 pulsar values against 400 non-pulsar values. In logistic regression, out of the total predicted pulsar values, 1503 were correctly identified as pulsars (true positives), while 136 non-pulsar values were misclassified as pulsars (false positives). 589 actual pulsar values were classified wrongly as non-pulsars (false negatives), and 15670 non-pulsar values were accurately identified (true negatives). The random forest model correctly classified 1549 pulsar values (true positives) and misclassified only 90 non-pulsar values as pulsars (false positives). However, it incorrectly labeled 774 actual pulsar values as non-pulsars (false negatives), while

accurately identifying 15485 non-pulsar values (true negatives). These metrics were checked against the total number of pulsar and non-pulsar values in the dataset, which comprised 1639 pulsars and 16259 non-pulsars. A similar analysis applies for the default case, Case 1. Case 2, and Case 3. The numbers change for each case as needed, depending on the output for each case.

Fourth Run (Shivali)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1478
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 161
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 308
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15951
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1518
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 121
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 400
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15859
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Fifth Run (Srimathi)

```
> sum((as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 1511
> sum((1-as.numeric(p1>.5))*(as.numeric(Pulsar1$v9)-1))
[1] 128
> sum((as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 564
> sum((1-as.numeric(p1>.5))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15695
> sum((as.numeric(p2)-1)*(as.numeric(Pulsar1$v9)-1))
[1] 1543
> sum((1-(as.numeric(p2)-1))*(as.numeric(Pulsar1$v9)-1))
[1] 96
> sum((as.numeric(p2)-1)*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 594
> sum((1-(as.numeric(p2)-1))*(1-(as.numeric(Pulsar1$v9)-1)))
[1] 15665
> sum(as.numeric(Pulsar1$v9)-1)
[1] 1639
> sum(1-(as.numeric(Pulsar1$v9)-1))
[1] 16259
```

Case 3 Analysis: For Case 3, Raashi's random forest model seems to generate the closest predicted values to the actual value, thus we can claim that it is one of the most accurate 2X2 tables. However, there are also other models that come close to this. Other 2X2 tables that come close to the accurate values would be Shivali's logistic regression table, and Srimathi's random forest model. The two models seem to be extremely close in terms of the outputted predicted values so either model can be chosen to do the Pulsar analysis for this case, depending on the situation and context. While the other models may not be accurate, they can still be somewhat decent analyses of the dataset.