100 Samples: Emily Gill

```r
# Loading in the data set
Pulsar <- read.csv("pulsar.csv")
# Removing the first column
Pulsar1 <- Pulsar[, -1]
# Converting target variable to factor
Pulsar1$v9 <- as.factor(Pulsar1$v9)

# Separating Pulsars and Non-Pulsars
# Non-Pulsars
Pulsar1base <- Pulsar1[Pulsar1$v9 == 0, ]
# Pulsars
Pulsar1true <- Pulsar1[Pulsar1$v9 == 1, ]

# Proportional Sampling Method

# Setting seed for reproducibility
set.seed(123)
# Randomly selecting 100 pulsars
v1 <- sample(nrow(Pulsar1true), 100)
# Randomly selecting 1000 non-pulsars
v2 <- sample(nrow(Pulsar1base), 1000)
# Combining samples
Pulsartry <- rbind(Pulsar1base[v2, ], Pulsar1true[v1, ])

# Fitting Logistic Regression and Random Forest Models
pulsar.logit <- glm(v9 ~ ., data = Pulsartry, family = binomial(link = "logit"))
pulsar.rf <- randomForest(v9 ~ ., data = Pulsartry, ntree = 5000)

# Making Predictions
p1 <- predict(pulsar.logit, type = "response", newdata = Pulsar1)
p2 <- predict(pulsar.rf, newdata = Pulsar1)

# Even Sampling Method

# Setting seed for reproducibility
set.seed(123)
# Randomly selecting 100 pulsars
v1_even <- sample(nrow(Pulsar1true), 100)
# Randomly selecting 100 non-pulsars
v2_even <- sample(nrow(Pulsar1base), 100)
# Combining samples
Pulsar_even <- rbind(Pulsar1base[v2_even, ], Pulsar1true[v1_even, ])
```

```r
# Fitting Logistic Regression and Random Forest Models
pulsar.logit_even <- glm(v9 ~ ., data = Pulsar_even, family = binomial(link = "logit"))
pulsar.rf_even <- randomForest(v9 ~ ., data = Pulsar_even, ntree = 5000)

# Making Predictions
p1_even <- predict(pulsar.logit_even, type = "response", newdata = Pulsar1)
p2_even <- predict(pulsar.rf_even, newdata = Pulsar1)

# Confusion Matrix Calculation

# Creating the Confusion Matrix Function
confusion_matrix <- function(actual, predicted) {
  tp <- sum(actual == 1 & predicted == 1)
  fn <- sum(actual == 1 & predicted == 0)
  fp <- sum(actual == 0 & predicted == 1)
  tn <- sum(actual == 0 & predicted == 0)
  matrix(c(tp, fn, fp, tn), nrow = 2, byrow = TRUE,
         dimnames = list("Actual" = c("Positive", "Negative"),
                         "Predicted" = c("Positive", "Negative")))
}

# Creating the Logistic Regression Confusion Matrices for Proportional and Even Sampling
conf_matrix_logit_prop <- confusion_matrix(as.numeric(Pulsar1$v9) - 1, as.numeric(p1 > 0.5))
conf_matrix_logit_even <- confusion_matrix(as.numeric(Pulsar1$v9) - 1, as.numeric(p1_even > 0.5))

# Creating the Random Forest Confusion Matrices for Proportional and Even Sampling
conf_matrix_rf_prop <- confusion_matrix(as.numeric(Pulsar1$v9) - 1, as.numeric(p2) - 1)
conf_matrix_rf_even <- confusion_matrix(as.numeric(Pulsar1$v9) - 1, as.numeric(p2_even) - 1)

# Adding titles and printing results
cat("Logistic Regression Confusion Matrix - Proportional Sampling:\n")
```

```
## Logistic Regression Confusion Matrix - Proportional Sampling:
```

```r
print(conf_matrix_logit_prop)
```

```
##           Predicted
## Actual     Positive Negative
##   Positive     1338      301
##   Negative       78    16181
```

```r
cat("\nLogistic Regression Confusion Matrix - Even Sampling:\n")
```

```
##
## Logistic Regression Confusion Matrix - Even Sampling:
```

```r
print(conf_matrix_logit_even)
```

```
##           Predicted
## Actual     Positive Negative
##   Positive     1502      137
##   Negative      551    15708
```

```r
cat("\nRandom Forest Confusion Matrix - Proportional Sampling:\n")
```

```
##
```

```
## Random Forest Confusion Matrix - Proportional Sampling:
print(conf_matrix_rf_prop)

##           Predicted
## Actual     Positive Negative
##   Positive     1355      284
##   Negative       86    16173
cat("\nRandom Forest Confusion Matrix - Even Sampling:\n")

##
## Random Forest Confusion Matrix - Even Sampling:
print(conf_matrix_rf_even)

##           Predicted
## Actual     Positive Negative
##   Positive     1494      145
##   Negative      718    15541
```

Analysis of Results:

Random Forest: Proportional Sampling

- Random forest achieves good recall for the positive class (1355/1639 ≈ 82.7%), which indicates it identifies the most positive cases. However, the precision is slightly lower due to a moderate number of false positives (86 out of 16259 negatives). The model's performance is reasonable but not exceptional. While recall is decent, proportional sampling does not significantly improve generalization, as seen from the balance between true and false positives.

Random Forest: Even Sampling

- The recall improves significantly for the positive class (1494/1639 ≈ 91.2%), but this improvement comes at the cost of a large increase in false positives (718). This results in a drop in precision and overall accuracy. Even sampling balances the dataset, leading to better identification of positive cases, but introduces a significant trade-off in terms of false positives. This suggests that random forest struggles to handle the modified distribution effectively.

Logistic Regression: Proportional Sampling

- Logistic regression performs well with proportional sampling, achieving high precision and recall for the positive class (1338/1639 ≈ 81.6%). The number of false positives is minimal (78 out of 16259 negatives), leading to excellent overall accuracy and F1 score. Proportional sampling reflects the real-world class distribution, enabling the model to generalize better. Logistic regression maintains a good balance between precision and recall with this approach.

Logistic Regression: Even Sampling

- Logistic regression improves recall for the positive class (1502/1639 ≈ 91.6%) but at the cost of a higher false positive rate (551 out of 16259 negatives). This slightly lowers precision compared to proportional sampling but maintains strong F1 and accuracy scores. Even sampling allows logistic regression to detect more positive cases, but the trade-off in precision may not always be desirable, especially in scenarios where false positives are costly.

Overall Conclusion:

- Logistic regression with proportional sampling emerges as the best performer overall. It provides a balanced trade-off between precision and recall and achieves high generalization by reflecting real-world class distributions.

Within Sampling Methods Conclusion:

- Proportional Sampling: Logistic regression outperforms random forest due to its ability to balance precision and recall effectively. Random Forest performs reasonably but is not as good.
- Even Sampling: Logistic regression also outshines random forest, as random forest struggles with a high false positive rate under this approach. Logistic regression demonstrates better adaptability to balanced datasets.