Classifying Mushrooms using Random Forest and Logistic Regression

Shrey Patel

11/19/2021

1 Abstract

In this study, I have classified mushrooms as either edible or poisonous using Random Forest (RF) and Logistic Regression (LR) classifiers while considering four features of a gilled mushroom: the odor, the shape of the mushroom cap, the color of the mushroom cap, and the type of gill attachment. This study was done to test and compare the performance of the random forest and logistic regression classifiers on the mushroom data set. Post-analyses, I can conclude that logistic regression classifier is almost seven times faster than random forest, and both of these models have high sensitivity for edible mushrooms.

2 Introduction

Logistic Regression and Random Forest classification algorithms are widely used for supervised learning. However, the performance of a classification algorithm depends on many factors like the train-test split, the sample size, the underlying distribution of the data set, the combination of parameters chosen for the algorithms, and the structure of the dependent variable. Hence, one of the ways to discover a perfectly suited model is through experimentation and comparison. In this study, I have classified mushrooms as edible or poisonous to test and compare the random forest and logistic regression performance metrics.

3 Data

The data set used in this study is publicly available on the UCI Machine Learning Repository. The data consists of information regarding the hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family. (1981) There are a total of 8,124 observations, 22 features and one target variable in the data set. In this study, we will only be working with a subset of 4 features - the odor, the shape of the mushroom cap, the color of the mushroom cap, and the type of gill attachment. The target variable 'type' takes two values 'edible' and 'poisonous.' The labels are well-balanced in the data set as there are a total of 4208 edible and 3916 poisonous mushrooms.

4 Analyses

4.1 Pre-processing the Data

For the ease of analyses, the dependent variable 'type' was replaced with a binarized target variable, 'edible,' such that the label '0' in the edible column represents poisonous mushrooms and the label '1' represents the edible mushrooms. Since all features are of character-type, all columns were then converted to factors. Additionally, no missing data was found. The data set was then split 75/25:train/test, with the training set and the testing set holding 6093 observations and 2031 observations, respectively.

4.2 Random Forest

Random Forest implements multiple decision tree classifiers under the hood. The idea is that each decision tree predicts a class an observation might belong to and the class chosen by the *majority* of decision trees is selected as the predicted class of the random forest classifier. One of the key assumptions is that the individual trees must have *low* correlation. Low correlation is achieved by using a bootstrapped sample instead of the complete data set at the time of building individual decision trees. Moreover, a new optimal train/test split is found every time a decision tree is built. The below used randomForest function in R implements Brelman's random forest algorithm. Let's look at the performance of the RF classifier.

Execution time:

```
tic("Random Forest")
rf <- randomForest(train$edible~., data=train[1:4])
rf_pred <- predict(rf, test[1:4])
toc()
```

```
## Random Forest: 3.67 sec elapsed
```

Performance metrics:

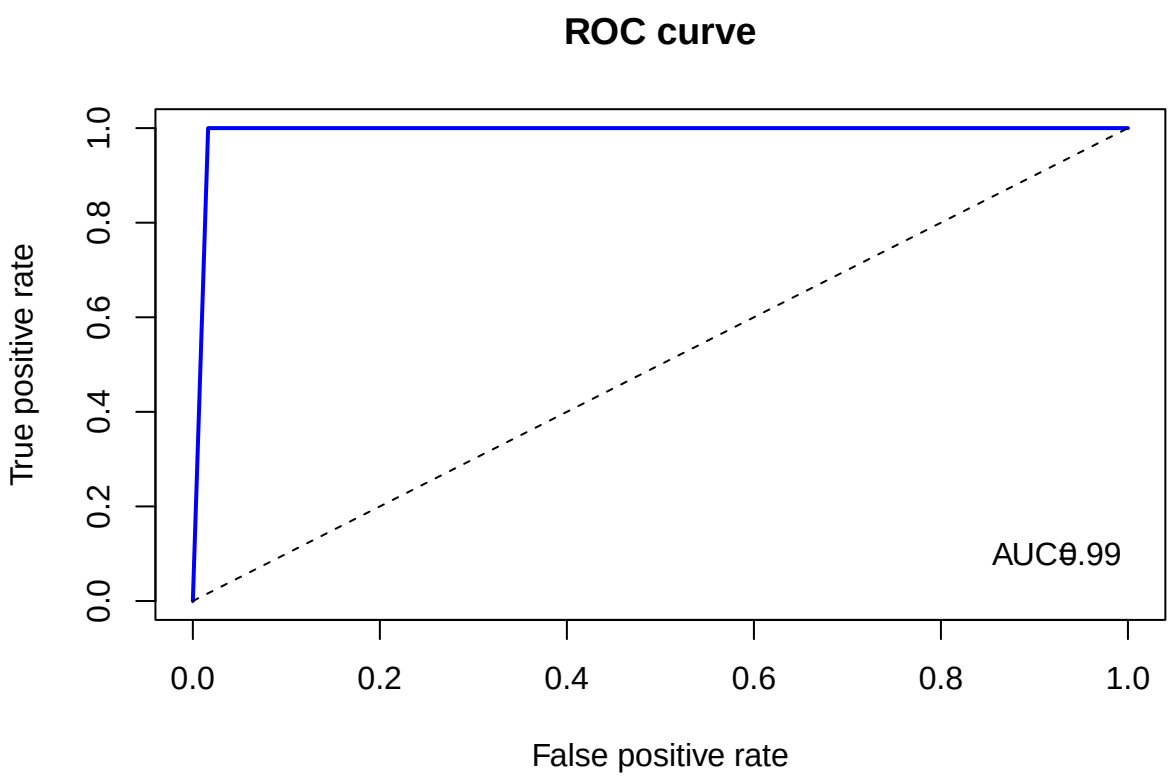
```
confusionMatrix(table(rf_pred,
                      test$edible), positive="1")
```

```
## Confusion Matrix and Statistics
##
##
## rf_pred    0    1
##       0  963    0
##       1   16 1052
##
##                Accuracy : 0.9921
##                  95% CI : (0.9872, 0.9955)
##     No Information Rate : 0.518
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9842
##
##  Mcnemar's Test P-Value : 0.0001768
##
##             Sensitivity : 1.0000
##             Specificity : 0.9837
##          Pos Pred Value : 0.9850
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5180
##          Detection Rate : 0.5180
##    Detection Prevalence : 0.5258
##       Balanced Accuracy : 0.9918
##
##        'Positive' Class : 1
##
```

ROC curve:

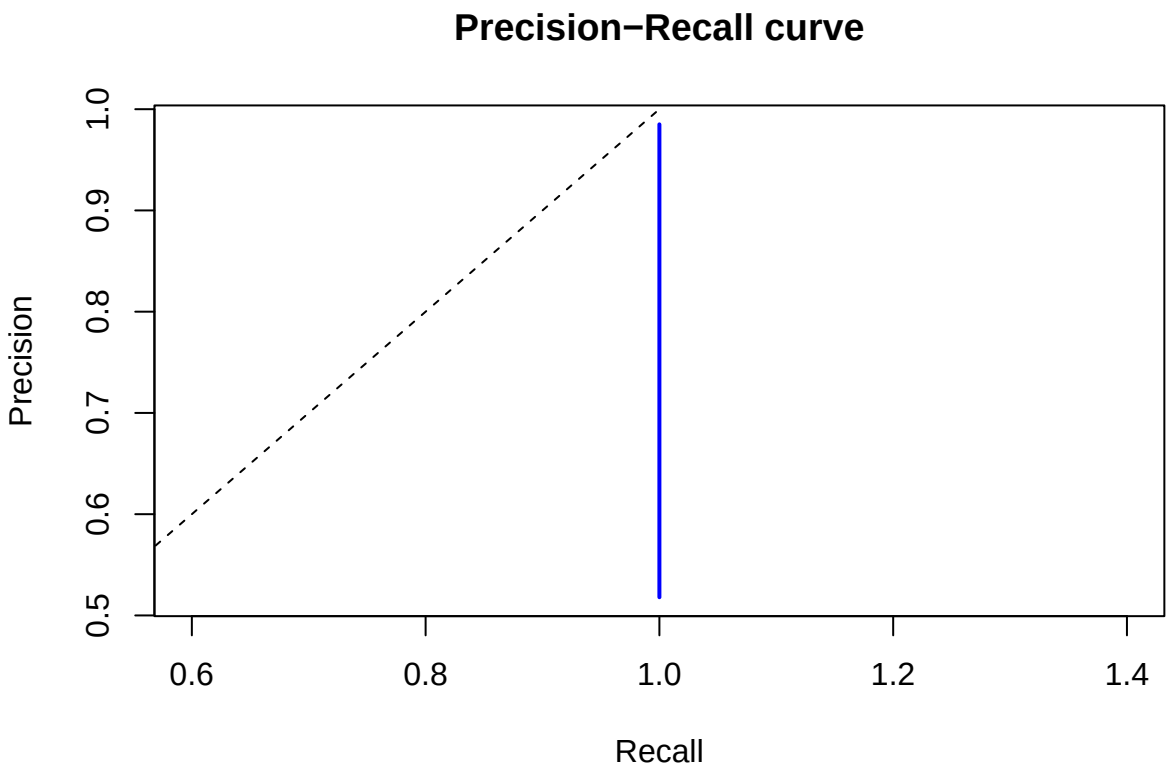
```
# ROC curve
library(ROCR)

pred_rf <- prediction(as.numeric(rf_pred), test$edible)
```

```
roc_rf <- performance(pred_rf, "tpr", "fpr")
plot(roc_rf, colorize = FALSE, lwd = 2, col = "blue", main = "ROC curve")
lines(c(0:100)/100, c(0:100)/100, lty = 2)
auc_rf <- performance(pred_rf, measure = "auc")
auc_rf <- auc_rf@y.values[[1]]
text(0.9, 0.1, "AUC=", cex = 1)
text(0.96, 0.1, round(auc_rf, 2), cex = 1)
```



Precision-Recall curve:

```
# Precision-Recall curve
prc_rf <- performance(pred_rf, "prec", "rec")
plot(prc_rf, colorize = FALSE, lwd = 2, col = "blue", main = "Precision-Recall curve")
lines(c(0:100)/100, c(0:100)/100, lty = 2)
```



4.3 Logistic Regression

Regression analysis is a set of statistical processes used to model and analyse the relationship between the dependent and the independent variables.(2021) Binary logistic regression is used to measure the probability that a certain binary event has occurred given the input. This probability is estimated using a logistic function. Hence, the output of a logistic regression classifier is bounded between 0 and 1, unlike linear regression.

To fit a logistic regression model, the family parameter of glm.fit() function is set to 'binomial.' Let's look at the performance of the RF classifier.

Execution time:

```
tic("Logistic Regression")
lr <- glm(train$edible ~., family="binomial", data=train[1:4])
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#summary(lr)
lr_pred_prob <- predict(lr, test[1:4], type="response")
toc()
```

```
## Logistic Regression: 0.5 sec elapsed
```

Performance metrics:

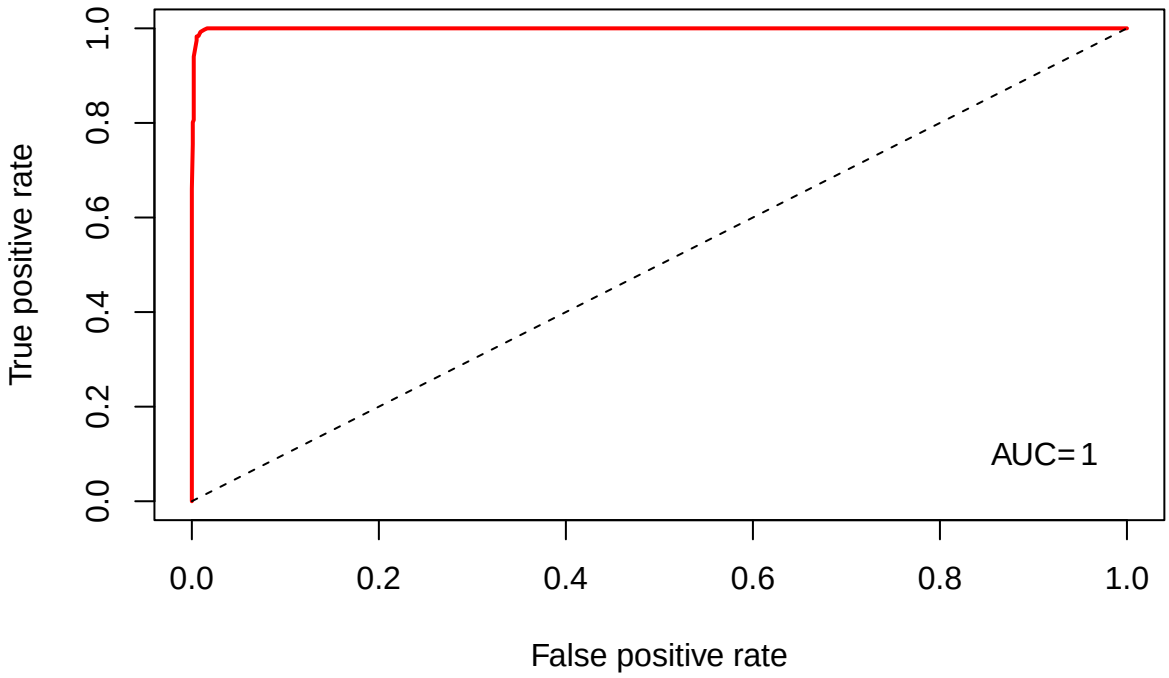
```
# confusion matrix
confusionMatrix(table(lr_pred,
                      test$edible), positive="1")
```

```
## Confusion Matrix and Statistics
```

```
##
##
## lr_pred    0    1
##       0  963    0
##       1   16 1052
##
##                Accuracy : 0.9921
##                  95% CI : (0.9872, 0.9955)
##     No Information Rate : 0.518
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9842
##
##  Mcnemar's Test P-Value : 0.0001768
##
##             Sensitivity : 1.0000
##             Specificity : 0.9837
##          Pos Pred Value : 0.9850
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5180
##          Detection Rate : 0.5180
##    Detection Prevalence : 0.5258
##       Balanced Accuracy : 0.9918
##
##        'Positive' Class : 1
##
```

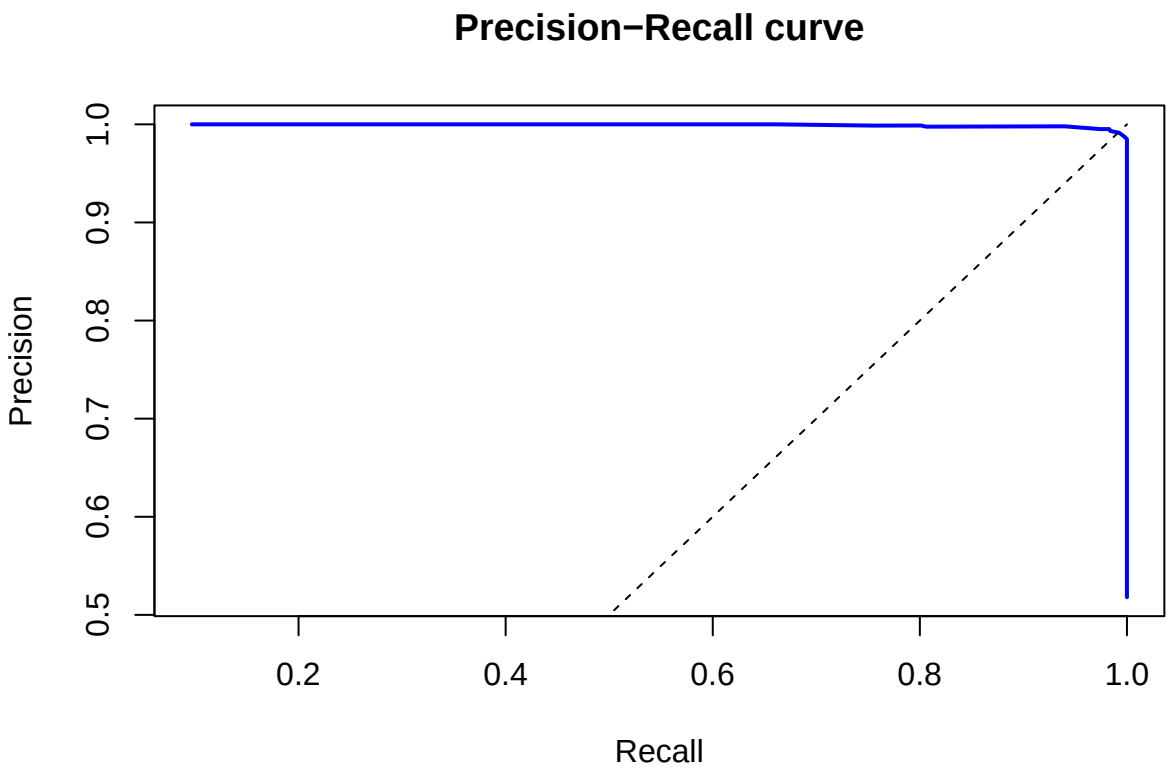
ROC curve:

```r
# ROC curve
pred_lr <- prediction(lr_pred_prob, test$edible)

roc_lr <- performance(pred_lr,"tpr","fpr")
plot(roc_lr,colorize=FALSE, lwd=2, col="red")
lines(c(0:100)/100,c(0:100)/100, lty=2)
#lines(rep(0,101),c(0:100)/100, lty=2, lwd=2, col="blue")
#lines(c(0:100)/100, rep(1,101), lty=2, lwd=2, col="blue")
auc_lr <- performance(pred_lr, measure = "auc")
auc_lr <- auc_lr@y.values[[1]]
text(0.9, 0.1, "AUC=", cex = 1)
text(0.96, 0.1, round(auc_lr, 2), cex = 1)
```



Precision-Recall curve:

```
# Precision-Recall curve
prc_lr <- performance(pred_lr, "prec", "rec")
plot(prc_lr, colorize = FALSE, lwd = 2, col = "blue", main = "Precision-Recall curve")
lines(c(0:100)/100, c(0:100)/100, lty = 2)
```



5 Conclusion

Comparing the results from both classifiers, we can draw the following conclusions:

i. Interesting so see how both the classifiers have identical results. However, logistic regression classifier was almost 7 times faster than random forest.

ii. LR resulted in a perfect AUC score, while random forest scored 0.99.

iii. Looking at the confusion matrix, we can say that the models have high sensitivity for edible mushrooms. All edible samples were detected by both classifiers as edible.

iv. Moreover, all the predicted poisonous samples were indeed poisonous. However, the models could not catch *all* the poisonous samples - leaving behind 18 samples that were classified as edible. Hence, these models could be quite risky in practice.

References

Jeff Schlimmer. 1981. *Mushroom Data Set*. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/mushroom.

"Regression Analysis." 2021. Wikimedia Foundation. https://en.wikipedia.org/wiki/Regression_analysis.