xgboost_caret

LAC

2022-11-13

Load libraries

```
library(xgboost)
library(caret)
library(Matrix)
library(dplyr)
library(ggplot2)
```

Load data

```
iris <- read.csv("./iris.csv")
```

Perform stratified random split of the data set with the caret package

```
train_index <- caret::createDataPartition(iris$Species, p=0.8, list = FALSE)
train_set <- iris[train_index,] # Training Set
test_set <- iris[-train_index,] # Test Set

write.csv(train_set, "trainset.csv")
write.csv(test_set, "testset.csv")
```

```
train_set <- read.csv("trainset.csv", header = TRUE)
head(train_set)
```

```
##   X Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 1          5.1         3.5          1.4         0.2  setosa
## 2 2          4.9         3.0          1.4         0.2  setosa
## 3 3          4.7         3.2          1.3         0.2  setosa
## 4 4          4.6         3.1          1.5         0.2  setosa
## 5 6          5.4         3.9          1.7         0.4  setosa
## 6 7          4.6         3.4          1.4         0.3  setosa
```

```
train_set <- train_set[,-1]
train_set$Species <- factor(train_set$Species)
```

```
test_set  <- read.csv("testset.csv", header = TRUE)
test_set <- test_set[,-1]
test_set$Species <- factor(test_set$Species)
```

Build XGBoost model

```
trainset_labels <- train_set$Species
trainset_labels_num <- as.integer(train_set$Species) - 1
trainset_mat <- Matrix(as.matrix(train_set[, -length(train_set)]), sparse = TRUE)
dim(trainset_mat)
```

```
## [1] 120   4
```

```
testset_labels <- test_set$Species
testset_labels_num <- as.integer(test_set$Species) - 1
testset_mat <- Matrix(as.matrix(test_set[, -length(test_set)]), sparse = TRUE)
dim(testset_mat)
```

```
## [1] 30  4
```

```
model <- xgboost( data = trainset_mat, label=trainset_labels_num,
                  max_depth=2, eta=1, nthread=2, nrounds=20,
                  num_class = 3, objective="multi:softprob", eval_metric="mlogloss")
```

```
## [1]  train-mlogloss:0.283157
## [2]  train-mlogloss:0.144284
## [3]  train-mlogloss:0.091283
## [4]  train-mlogloss:0.069587
## [5]  train-mlogloss:0.056696
## [6]  train-mlogloss:0.043419
## [7]  train-mlogloss:0.038659
## [8]  train-mlogloss:0.035747
## [9]  train-mlogloss:0.033639
## [10] train-mlogloss:0.031815
## [11] train-mlogloss:0.030467
## [12] train-mlogloss:0.028576
## [13] train-mlogloss:0.027565
## [14] train-mlogloss:0.026323
## [15] train-mlogloss:0.025366
## [16] train-mlogloss:0.024426
## [17] train-mlogloss:0.023188
## [18] train-mlogloss:0.022438
## [19] train-mlogloss:0.021831
## [20] train-mlogloss:0.021148
```

Predict on test data

```
pred <- predict(model, testset_mat)
pred <- matrix(pred, nrow=30, byrow = TRUE)
pred
```

```
##               [,1]         [,2]         [,3]
##  [1,] 0.9962483048 0.0033289369 0.0004227230
##  [2,] 0.9962483048 0.0033289369 0.0004227230
##  [3,] 0.9962483048 0.0033289369 0.0004227230
##  [4,] 0.9822852612 0.0172978919 0.0004167983
##  [5,] 0.9929209352 0.0066578100 0.0004213112
##  [6,] 0.9962483048 0.0033289369 0.0004227230
##  [7,] 0.9929209352 0.0066578100 0.0004213112
##  [8,] 0.9929209352 0.0066578100 0.0004213112
##  [9,] 0.9962483048 0.0033289369 0.0004227230
## [10,] 0.9962483048 0.0033289369 0.0004227230
## [11,] 0.0018869573 0.9974201918 0.0006928329
## [12,] 0.0050970279 0.9869404435 0.0079624886
## [13,] 0.0018869573 0.9974201918 0.0006928329
## [14,] 0.0018869573 0.9974201918 0.0006928329
## [15,] 0.0017674405 0.9977953434 0.0004372906
## [16,] 0.0045328001 0.9877915978 0.0076755825
## [17,] 0.0020709548 0.9972078204 0.0007212556
## [18,] 0.0029383516 0.9893737435 0.0076878760
## [19,] 0.0022884202 0.9967066646 0.0010049996
## [20,] 0.0148131996 0.9464296103 0.0387571231
## [21,] 0.0002934058 0.0003272922 0.9993792772
## [22,] 0.0002934058 0.0003272922 0.9993792772
## [23,] 0.0002199028 0.0003536509 0.9994264841
## [24,] 0.0005202268 0.0006924004 0.9987873435
## [25,] 0.0002934058 0.0003272922 0.9993792772
## [26,] 0.0005202268 0.0006924004 0.9987873435
## [27,] 0.0005202268 0.0006924004 0.9987873435
## [28,] 0.0141157657 0.6252365708 0.3606477082
## [29,] 0.0005202268 0.0006924004 0.9987873435
## [30,] 0.0004014343 0.0024884001 0.9971101880
```

```
# or
# dim(pred) <- c(3, 30)
# pred <- t(pred)
```

```
xgbpred <- as.data.frame(ifelse(pred > 0.5, 1, -1))
xgbpred$pred <- if_else(xgbpred$V1==1, 0, -1)
xgbpred$pred2 <- if_else(xgbpred$V2==1, 1, xgbpred$pred)
xgbpred$pred3 <- if_else(xgbpred$V3==1, 2, xgbpred$pred2)
```

Confusion Matrix

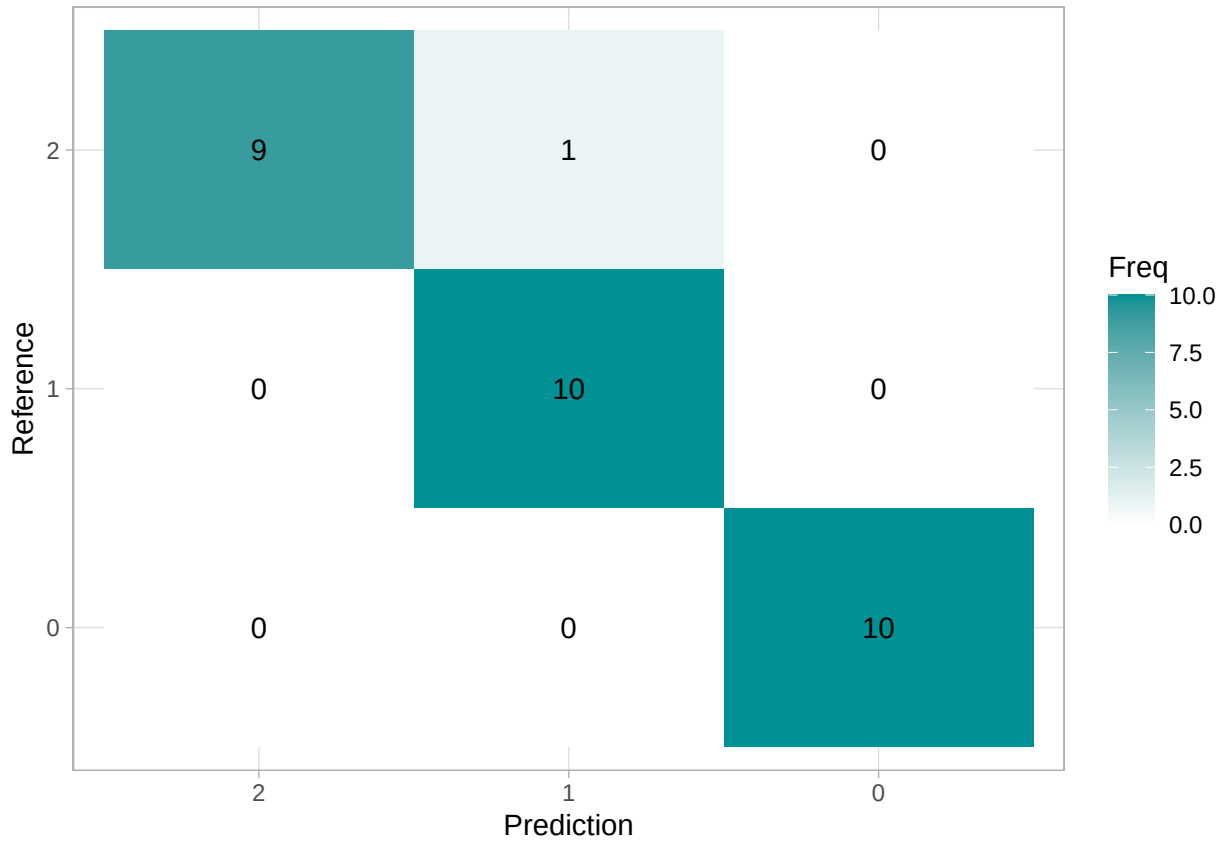
```
caret::confusionMatrix(factor(xgbpred$pred3), factor(testset_labels_num))
```

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

```
##
##           Reference
## Prediction  0  1  2
##          0 10  0  0
##          1  0 10  1
##          2  0  0  9
##
## Overall Statistics
##
##                Accuracy : 0.9667
##                  95% CI : (0.8278, 0.9992)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 2.963e-13
##
##                   Kappa : 0.95
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2
## Sensitivity            1.0000   1.0000   0.9000
## Specificity            1.0000   0.9500   1.0000
## Pos Pred Value         1.0000   0.9091   1.0000
## Neg Pred Value         1.0000   1.0000   0.9524
## Prevalence             0.3333   0.3333   0.3333
## Detection Rate         0.3333   0.3333   0.3000
## Detection Prevalence   0.3333   0.3667   0.3000
## Balanced Accuracy      1.0000   0.9750   0.9500
```

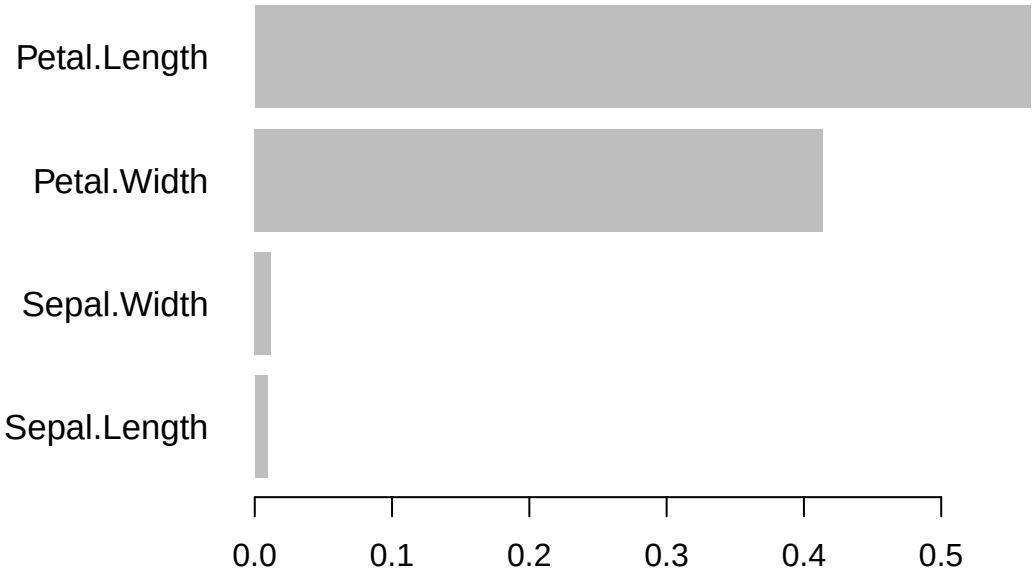
```
cm <- caret::confusionMatrix(factor(xgbpred$pred3), factor(testset_labels_num))
plt <- as.data.frame(cm$table)
plt$Prediction <- factor(plt$Prediction, levels=rev(levels(plt$Prediction)))

ggplot(plt, aes(Prediction, Reference, fill= Freq)) +
  geom_tile() +
  geom_text(aes(label=Freq)) +
  scale_fill_gradient(low="white", high="#009194") +
  theme_light()
```



Feature Importance

```
importance_mat <- xgb.importance(feature_names = colnames(trainset_mat), model = model)
xgb.plot.importance(importance_matrix = importance_mat)
```



Save the model to RDS file

```
saveRDS(model, "xgboost_model.rds")
```

```
sessionInfo()
```

```
## R version 4.2.2 Patched (2022-11-10 r83330)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.1 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.20.so
##
## locale:
##  [1] LC_CTYPE=en_CA.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_CA.UTF-8        LC_COLLATE=en_CA.UTF-8
##  [5] LC_MONETARY=en_CA.UTF-8    LC_MESSAGES=en_CA.UTF-8
##  [7] LC_PAPER=en_CA.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_1.0.10    Matrix_1.5-1    caret_6.0-93    lattice_0.20-45
## [5] ggplot2_3.4.0   xgboost_1.6.0.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.9           lubridate_1.9.0      listenv_0.8.0
##  [4] class_7.3-20         digest_0.6.30        ipred_0.9-13
##  [7] foreach_1.5.2        utf8_1.2.2           parallelly_1.32.1
## [10] R6_2.5.1             plyr_1.8.8           stats4_4.2.2
## [13] hardhat_1.2.0        e1071_1.7-12         evaluate_0.18
## [16] highr_0.9            pillar_1.8.1         rlang_1.0.6
## [19] data.table_1.14.6    rpart_4.1.19         rmarkdown_2.18
## [22] labeling_0.4.2       splines_4.2.2        gower_1.0.0
## [25] stringr_1.4.1        munsell_0.5.0        proxy_0.4-27
## [28] compiler_4.2.2       xfun_0.35            pkgconfig_2.0.3
## [31] globals_0.16.2       htmltools_0.5.3      nnet_7.3-18
## [34] tidyselect_1.2.0     tibble_3.1.8         prodlim_2019.11.13
## [37] codetools_0.2-18     fansi_1.0.3          future_1.29.0
## [40] withr_2.5.0          ModelMetrics_1.2.2.2 MASS_7.3-58
## [43] recipes_1.0.3        grid_4.2.2           nlme_3.1-160
## [46] jsonlite_1.8.3       gtable_0.3.1         lifecycle_1.0.3
## [49] magrittr_2.0.3       pROC_1.18.0          scales_1.2.1
## [52] future.apply_1.10.0  cli_3.4.1            stringi_1.7.8
## [55] farver_2.1.1         reshape2_1.4.4       timeDate_4021.106
## [58] generics_0.1.3       vctrs_0.5.1          lava_1.7.0
## [61] iterators_1.0.14     tools_4.2.2          glue_1.6.2
## [64] purrr_0.3.5          parallel_4.2.2       fastmap_1.1.0
## [67] survival_3.4-0       yaml_2.3.6           timechange_0.1.1
## [70] colorspace_2.0-3     knitr_1.41
```