

- 1 Read in the data
- 2 Stop Paralell Processing
- 3 Print the best set of hyperparameters selected
- 4 Create a confusion matrix for the Xgboost model
- 5 Comment:

Code ▾

1 Read in the data

The following data set will be used to predict

`loan_default'`. Our goal in this analysis is to create a model able to accurately detect `loan_default`. We have a data set containing 7555

Hide

```
train=readRDS("group5AA_Black-Boopathy_train.rds")
holdout=readRDS( "holdout_df_RF.rds")
```

We have set up for parallel processing

Hide

```
cores=parallel::detectCores()
cl <- parallel::makeCluster(cores-1) # Set CPU cores for parallel execution
registerDoParallel(cl) # Register parallel backend
```

Also, we set up 5-fold cross validation

Hide

```
cv_control <- trainControl(
  method = "cv",
  number = 5,
  allowParallel = TRUE # Enables parallel computation during training
)
```

Now we are in the process of Training XGBoost model using `tuneLength=5` and `set.seed(123)` for reproducibility

Hide

```
set.seed(123)
xgb_model <- train(
  loan_default ~.,
  data = train,
  method = "xgbTree",
  trControl = cv_control, # Apply 5-fold cross-validation
  tuneLength = 5 # caret picks up to 5 values for each of 7 parms and fits all combos up to 5^7
)
```

2 Stop Paralell Processing

Hide

```
stopCluster(cl) # Shut down parallel cluster
registerDoSEQ() # Reset to sequential processing
```

3 Print the best set of hyperparameters selected

Hide

```
print(xgb_model$bestTune)
```

```
##      nrounds max_depth eta gamma colsample_bytree min_child_weight subsample
## 150      250          3 0.3     0            0.8           1             1
```

3.1 Predict the holdout set using the Random Forest model

We create a data frame called `holdout` that contains the actual target values from the holdout sample. Add the class predictions and the positive-class probabilities from the Xgb model. Label these as `xgb.class`, `xgb.prob`. We make sure the table by Showing the first few lines of the data frame.

Hide

```
holdout <- readRDS( "holdout_df_RF.rds" ) # Load the holdout data
holdout$xgb.class<- predict(xgb_model,
                             newdata = holdout,
                             type="raw")
holdout$xgb.prob <- predict(xgb_model,
                             newdata = holdout,
                             type="prob")[, "Yes"]
```

4 Create a confusion matrix for the Xgboost model

Hide

```
confusionMatrix(holdout$xgb.class,
                holdout$loan_default,
                positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction      No     Yes
##       No 60178 12476
##       Yes   848  2053
##
##             Accuracy : 0.8237
##                 95% CI : (0.8209, 0.8264)
##       No Information Rate : 0.8077
##       P-Value [Acc > NIR] : < 2.2e-16
##
##             Kappa : 0.1833
##
## Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.14130
##             Specificity : 0.98610
##       Pos Pred Value : 0.70769
##       Neg Pred Value : 0.82828
##             Prevalence : 0.19230
##       Detection Rate : 0.02717
## Detection Prevalence : 0.03840
##       Balanced Accuracy : 0.56370
##
##       'Positive' Class : Yes
##
```

5 Comment:

This model detects 12476 false negatives and 2053 true positives with accuracy as 82% and Balanced accuracy as 56%.

Hide

```
saveRDS(holdout, "holdout_df_Boost.rds")
```

Next Stage:

We are approaching into comparing all the models with metrics which gives the best decision making in loan default.