**Lab 2**

**Machine Learning Lab 2: Interpretation**

Sishira Ithigani

MSBA, Northwood University

113681-MGT-665-NW2 Solving Problems with Machine Learning.

Dr. Mohammad Nasim

In order to interpret the machine learning model, we have Accuracy, Precision and Recall, F1 score, ROC curve, Residual analysis, Prediction intervals.

Apart from accuracy lets discuss about the remaining metrics.

**Precision and Recall**: Precision is the ratio of true positive predictions to the sum of true positives and false positives. It gives us the ratio of students predicted to pass the course who actually passed. This gives us a better understanding on the accuracy of the positive predictions.

Recall is ratio of true positive predictions to the sum of true positives and false negatives. It tells us the ratio of students who passed the course that were correctly identified by the model. It gives us a better outlook on how many of the actual positive instances the model was able to capture.

**ROC curve:** I used ROC to visualize the tradeoff between true positive rate and false positive rate. In this instance it helped us in understanding model’s performance at different classification thresholds. A higher AUC-ROC indicates better performance.

**Confusion Matrix**: I used confusion matrix for tabular representation of model’s predictions. It gives true positives, false positives, true negatives, false negatives. From this we also derived accuracy, recall and false positive rate.

**Sensitivity**: It means the True Positive Rate and Recall. I used this to measure the ratio of actual positive cases that were correctly identified by the model. It tells us how well the model identifies positive instances.

**Specificity:** It gives us the True negative rate. It is used to measure the ratio of actual negative cases that were correctly identified by the model. It tells us how well the model identifies negative instances.

**Prevalence**: I used prevalence to calculate the ratio of positive cases. It is used to represent the frequency of the positive class in the dataset.

**Detection Prevalence:**  I used this to calculate the ratio of predicted positive cases. It is used to know the frequency of predicted positive cases in the dataset.

**Balanced Accuracy:** I used Balanced Accuracy to calculate the mean of sensitivity and specificity.

**F1** **score**: It is the mean of precision and recall. It gives a single metric that balances both i.e., precision and recall. Here we used it to compare models that have different tradeoffs between precision and recall. This is used in cases where classes are imbalanced. It is used to know the overall performance by considering both false positives and false negatives.

**AUC-PR**: It helps in measuring the area under the precision-recall curve. It is useful when we have imbalanced datasets. A higher AUC-PR indicates better performance.

Apart from accuracy, we have to look beyond in order to understand the performance across various metrics. In addition, other models like decision trees, SVM’s can be used to compare the models to choose the best performing model for a given problem statement.

Overall, these metrics helps us in measuring the performance in classifying the students into different categories i.e., pass and fail based on the various parameters.

**Source code:**

# Load the required libraries

library(caret)

library(dplyr)

library(randomForest)

library(ggplot2)

# Read the data

data <- read.csv("https://raw.githubusercontent.com/nasimm48/machine-learning/main/lab-2/data/oulad-assessments.csv")

# Remove rows where there are missing values

data <- na.omit(data)

# Convert categorical variables to factors

data <- data %>%

mutate(across(c(code\_module, code\_presentation, assessment\_type), as.factor))

# Binning the score variable

data$score <- cut(data$score, breaks=quantile(data$score, probs=0:3/3, na.rm=TRUE), include.lowest=TRUE, labels=c("Low", "Medium", "High"))

data$score <- as.factor(data$score)

# reproducibility

set.seed(123)

# Split the data into training and testing sets

data\_set\_size <- floor(nrow(data) \* 0.80)

index <- sample(1:nrow(data), size = data\_set\_size)

training <- data[index, ]

testing <- data[-index, ]

# Fit random forest model for classification

rf <- randomForest(score ~ code\_module + code\_presentation + assessment\_type, data = training)

# Prediction and Result

predictions <- predict(rf, newdata = testing, type = "response")

result <- data.frame(Actual = testing$score, Predicted = predictions)

# Generate confusion matrix

conf\_matrix <- confusionMatrix(data = result$Predicted, reference = result$Actual)

# Plot the confusion matrix

confusion\_plot <- ggplot(as.data.frame(conf\_matrix$table), aes(x = Reference, y = Prediction, fill = Freq)) +

geom\_tile(color = "black") +

geom\_text(aes(label = Freq), color = "white", size = 3) +

scale\_fill\_gradient(low = "skyblue", high = "darkblue") +

ggtitle("Confusion Matrix") +

theme\_bw() +

xlab("Actual Class") +

ylab("Predicted Class") +

theme(axis.text = element\_text(size = 10),

axis.title = element\_text(size = 12),

plot.title = element\_text(size = 14, hjust = 0.5))

print(confusion\_plot)

conf\_matrix

Top of Form



