Hurdle Testing Associational Model

```
# Environment Cleaning: Clearing workspace
rm(list = ls())

# load packages
library(rpart)
library(dplyr)
library(caret)
library(data.table)
library(mlflow)
library(reticulate)
library(Metrics)
library(here)
```

Inputs

Testing data

```
# data for testing
# reading the test datasets.
# Reading base_test data
base_test <- read.csv(here("data", "base_test.csv"))

# Redaing trunc_test data
truncated_test <- read.csv(here("data", "truncated_test.csv"))

df_test <- bind_rows(
  base_test,
  truncated_test
)

nrow(df_test)
```

```
## [1] 1896
```

Trained models

```
# Read the .rds models
base_reg <- readRDS(here("associational XGBOOST", "damage_fit_reg_base.rds"))
trunc_reg <- readRDS(here("associational XGBOOST", "trunc_damage_fit_reg.rds"))
clas_model  <- readRDS(here("associational XGBOOST", "ass_XGBOOST_class.rds"))
```

Binned Testing

```
source(here("R", "ass_hurdle_function.R"))

# setting threshold for classification step
threshold = 0.3

preds <- ass_hurdle_function(df = df_test, ass_clas_model = clas_model,
  ass_base_model = base_reg, ass_trunc_model = trunc_reg ,threshold = threshold)
```

```
# Define bin edges
# Define bin edges
bins <- c(0, 0.00009, 1, 10, 50, 100)

# Assign data to bins
bin_labels <- cut(df_test$damage_perc, breaks = bins, include.lowest = TRUE, right = TRUE)

# Create a data frame with actual, predicted, and bin labels
data <- data.frame(
  actual = df_test$damage_perc,
  predicted = preds,
  bin = bin_labels
)

# Calculate RMSE per bin
unique_bins <- levels(data$bin) # Get unique bin labels
rmse_by_bin <- data.frame(bin = unique_bins, rmse = NA, count = NA) # Initialize results data frame

for (i in seq_along(unique_bins)) {
  bin_data <- data[data$bin == unique_bins[i], ] # Filter data for the current bin
  rmse_by_bin$rmse[i] <- sqrt(mean((bin_data$actual - bin_data$predicted)^2, na.rm = TRUE)) # Calculate
  rmse_by_bin$count[i] <- nrow(bin_data) # Count observations in the bin
}

# Calculate weighted average RMSE
total_count <- sum(rmse_by_bin$count, na.rm = TRUE)
w_avg  <- sum(rmse_by_bin$rmse * rmse_by_bin$count)/total_count

# Display RMSE by bin
print(rmse_by_bin)
```

```
##         bin      rmse count
## 1 [0,9e-05]  1.051770  1011
## 2 (9e-05,1]  3.985899   454
## 3    (1,10] 10.688192   231
## 4   (10,50] 12.692468   174
## 5  (50,100] 23.077279    26
```

```
w_avg
```

```
## [1] 4.298739
```