

R Notebook

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```

# Load the CSV (no strings as factors)
annotations <- read.csv("ProjectData_ForClass_v2-3/Annotation.csv", stringsAsFactors
= FALSE)

# Rename columns for consistency
names(annotations) <- c("Index", "Panel", "Tree", "Animal", "Mythology", "Comments")

# Generate matching image names
annotations$image <- paste0("s", seq_len(nrow(annotations)))

# TREE labels: only keep rows with valid 0 or 1
tree_labels <- annotations %>%
  filter(Tree %in% c(0, 1)) %>%
  select(image, label = Tree)

# ANIMAL labels
animal_labels <- annotations %>%
  filter(Animal %in% c(0, 1)) %>%
  select(image, label = Animal)

# MYTH labels
myth_labels <- annotations %>%
  filter(Mythology %in% c(0, 1)) %>%
  select(image, label = Mythology)

# Save output
if (!dir.exists("Output")) dir.create("Output")
save(tree_labels, animal_labels, myth_labels, file = "Output/labels.RData")

# Confirmation
cat("✅ Labels saved:\n")

```

```
## ✅ Labels saved:
```

```
cat("   Tree       :", nrow(tree_labels), "images\n")
```

```
##   Tree       : 34 images
```

```
cat("   Animal     :", nrow(animal_labels), "images\n")
```

```
##   Animal     : 34 images
```

```
cat("    Mythology :", nrow(myth_labels), "images\n")
```

```
##    Mythology : 34 images
```

```
load("Output/features.RData")  
cat("✅ Features loaded:", nrow(features_df), "images\n")
```

```
## ✅ Features loaded: 126 images
```

```
head(features_df$image)
```

```
## [1] "s1" "s2" "s3" "s1" "s10" "s2"
```

```
library(dplyr)  
  
# Load feature + label files  
load("Output/features.RData")  
load("Output/labels.RData")  
  
# Join function: only keep rows with matching images  
prepare_dataset <- function(labels_df, features_df) {  
  inner_join(features_df, labels_df, by = "image")  
}  
  
# Join all datasets  
tree_data <- prepare_dataset(tree_labels, features_df)  
animal_data <- prepare_dataset(animal_labels, features_df)  
myth_data <- prepare_dataset(myth_labels, features_df)  
  
# Save for training  
if (!dir.exists("Output")) dir.create("Output")  
save(tree_data, animal_data, myth_data, file = "Output/datasets.RData")  
  
# Confirm  
cat("✅ Datasets prepared:\n")
```

```
## ✅ Datasets prepared:
```

```
cat("    Tree      :", nrow(tree_data), "rows\n")
```

```
##      Tree      : 98 rows
```

```
cat("      Animal      :", nrow(animal_data), "rows\n")
```

```
##      Animal      : 98 rows
```

```
cat("      Mythology ::", nrow(myth_data), "rows\n")
```

```
##      Mythology : 98 rows
```

```
# Custom SGD function for logistic & exponential loss
run_sgd <- function(X, y, loss_type = "logistic", lr = 0.01, n_iter = 1000) {
  n <- nrow(X)
  d <- ncol(X)
  w <- rep(0, d)
  loss_history <- numeric(n_iter)

  for (iter in 1:n_iter) {
    i <- sample(1:n, 1)
    xi <- X[i, ]
    yi <- y[i]
    margin <- yi * sum(w * xi)

    if (loss_type == "logistic") {
      loss <- log1p(exp(-margin))
      grad <- (-yi * xi) / (1 + exp(margin))
    } else if (loss_type == "exponential") {
      margin_clamped <- max(-100, min(100, -margin))
      loss <- exp(margin_clamped)
      grad <- -yi * xi * exp(margin_clamped)
    } else {
      stop("Unsupported loss type.")
    }

    if (is.nan(loss) || is.infinite(loss)) loss <- 0
    w <- w - lr * grad
    loss_history[iter] <- loss
  }

  return(list(w = w, loss = loss_history))
}

# Load prepared datasets
```

```
load("Output/datasets.RData")

# Helper to prep data
prepare_inputs <- function(data) {
  clean <- data %>% dplyr::filter(complete.cases(.))
  X <- as.matrix(clean[, !(names(clean) %in% c("label", "image"))])
  y <- ifelse(clean$label == 1, 1, -1)
  return(list(X = X, y = y))
}

# Build training input sets
tree_input <- prepare_inputs(tree_data)
animal_input <- prepare_inputs(animal_data)
myth_input <- prepare_inputs(myth_data)

# Train logistic loss models
tree_logistic <- run_sgd(tree_input$X, tree_input$y, "logistic")
animal_logistic <- run_sgd(animal_input$X, animal_input$y, "logistic")
myth_logistic <- run_sgd(myth_input$X, myth_input$y, "logistic")

# Train exponential loss models
tree_expo <- run_sgd(tree_input$X, tree_input$y, "exponential")
animal_expo <- run_sgd(animal_input$X, animal_input$y, "exponential")
myth_expo <- run_sgd(myth_input$X, myth_input$y, "exponential")

# Save models and loss curves
if (!dir.exists("Output/models")) dir.create("Output/models", recursive = TRUE)
save(tree_logistic, animal_logistic, myth_logistic, file = "Output/models/logistic_models.RData")
save(tree_expo, animal_expo, myth_expo, file = "Output/models/exponential_models.RData")

cat("✅ Models trained and saved.\n")
```

```
## ✅ Models trained and saved.
```

```

library(ggplot2)

# Load saved models
load("Output/models/logistic_models.RData")
load("Output/models/exponential_models.RData")

# Plotting function
plot_loss <- function(loss_vector, title, output_file) {
  df <- data.frame(Iteration = 1:length(loss_vector), Loss = loss_vector)
  p <- ggplot(df, aes(x = Iteration, y = Loss)) +
    geom_line(color = "steelblue") +
    ggtitle(title) +
    xlab("Iteration") + ylab("Loss") +
    theme_minimal()

  ggsave(output_file, plot = p, width = 6, height = 4)
  cat("✅ Saved:", output_file, "\n")
}

# Output folder
if (!dir.exists("Output/plots")) dir.create("Output/plots", recursive = TRUE)

# Logistic Loss Plots
plot_loss(tree_logistic$loss, "Tree - Logistic Loss", "Output/plots/tree_logistic_loss.png")

```

```
## ✅ Saved: Output/plots/tree_logistic_loss.png
```

```
plot_loss(animal_logistic$loss, "Animal - Logistic Loss", "Output/plots/animal_logistic_loss.png")
```

```
## ✅ Saved: Output/plots/animal_logistic_loss.png
```


```
plot_loss(myth_logistic$loss, "Mythology - Logistic Loss", "Output/plots/myth_logistic_loss.png")
```

```
## ✅ Saved: Output/plots/myth_logistic_loss.png
```


```

# Exponential Loss Plots
plot_loss(tree_expo$loss, "Tree - Exponential Loss", "Output/plots/tree_expo_loss.png")


```

```
##  Saved: Output/plots/tree_expo_loss.png
```

```
plot_loss(animal_expo$loss, "Animal - Exponential Loss", "Output/plots/animal_expo_loss.png")
```

```
##  Saved: Output/plots/animal_expo_loss.png
```

```
plot_loss(myth_expo$loss, "Mythology - Exponential Loss", "Output/plots/myth_expo_loss.png")
```

```
##  Saved: Output/plots/myth_expo_loss.png
```

```
cat("🎉 All loss plots generated and saved.\n")
```

```
## 🎉 All loss plots generated and saved.
```