Instructions for Running the Provided Code

1. Prerequisites

1.1 Required Software

R (version 4.0 or later recommended)

RStudio (optional but recommended)

1.2 Required Libraries

Install the following R libraries if not already installed:

R

code

install.packages("ggplot2")

install.packages("dplyr")

install.packages("caret")

install.packages("randomForest")

install.packages("glmnet")

install.packages("cluster")

1.3 Data Requirements

The dataset file dataset.csv must be located in your current working directory.

Verify the dataset structure by examining its columns and values using the commands:

R

code

str(dataset)

summary(dataset)

2. Running the Code

2.1 Data Loading and Preprocessing

Load the dataset:

R

code

spotify\_data <- read.csv("dataset.csv")

Inspect the dataset: Use str() and summary() to view the structure and summary statistics of the data.

Clean the data:

Remove irrelevant columns (e.g., track\_id, track\_name, album\_name, artists).

Handle missing values by omitting rows with NA values.

R

code

dataset\_clean <- spotify\_data %>%

select(-track\_id, -track\_name, -album\_name, -artists) %>%

na.omit()

2.2 Exploratory Data Analysis (EDA)

Visualize the distributions and relationships between features:

Boxplots for danceability and popularity across genres:

R

code

ggplot(spotify\_data, aes(x = track\_genre, y = danceability, fill = track\_genre)) +

geom\_boxplot() +

theme\_minimal()

Scatter plots to explore feature relationships, e.g., loudness vs. energy:

R

code

ggplot(spotify\_data, aes(x = loudness, y = energy)) +

geom\_point(alpha = 0.7, color = "blue")

Feature distributions: Use histograms to examine the distributions of key features such as danceability and energy.

2.3 Modeling and Evaluation

2.3.1 Ridge Regression

Prepare data for modeling:

Extract the relevant features (danceability, energy, etc.) and the target variable (popularity).

Split the dataset into training and testing sets:

R

code

train\_index <- createDataPartition(dataset\_clean$popularity, p = 0.8, list = FALSE)

train\_data <- dataset\_clean[train\_index, ]

test\_data <- dataset\_clean[-train\_index, ]

Standardize features:

R

code

X\_train <- scale(as.matrix(train\_data[, c("danceability", "energy")]))

X\_test <- scale(as.matrix(test\_data[, c("danceability", "energy")]), center = attr(X\_train, "scaled:center"), scale = attr(X\_train, "scaled:scale"))

Train the ridge regression model:

R

code

ridge\_model <- cv.glmnet(X\_train, train\_data$popularity, alpha = 0, lambda = 10^seq(-3, 3, length.out = 100))

best\_lambda <- ridge\_model$lambda.min

Evaluate the model:

Predict on the test set:

R

code

ridge\_predictions <- predict(ridge\_model, s = best\_lambda, newx = X\_test)

Calculate performance metrics:

R

code

ridge\_mse <- mean((test\_data$popularity - ridge\_predictions)^2)

ridge\_r2 <- 1 - sum((test\_data$popularity - ridge\_predictions)^2) / sum((test\_data$popularity - mean(test\_data$popularity))^2)

Print the results:

R

code

cat("MSE:", round(ridge\_mse, 4), "\nR²:", round(ridge\_r2, 4), "\n")

2.3.2 Random Forest

Train a random forest model to predict track\_genre:

R

code

rf\_model <- randomForest(track\_genre ~ ., data = train\_data, ntree = 100, importance = TRUE)

Evaluate the model:

Generate predictions:

R

code

predictions <- predict(rf\_model, newdata = test\_data)

Compute a confusion matrix:

R

code

conf\_matrix <- confusionMatrix(predictions, test\_data$track\_genre)

print(conf\_matrix)

Analyze feature importance:

R

code

importance\_values <- importance(rf\_model)

importance\_df <- data.frame(Feature = rownames(importance\_values), Importance = importance\_values[, "MeanDecreaseGini"])

print(importance\_df)

2.4 Clustering Analysis

Run K-Means clustering:

Select features and standardize them:

R

code

selected\_features <- spotify\_data %>% select(danceability, energy, valence)

scaled\_features <- scale(selected\_features)

Perform clustering:

R

code

kmeans\_result <- kmeans(scaled\_features, centers = 3, nstart = 25)

spotify\_data$cluster <- as.factor(kmeans\_result$cluster)

Visualize clustering results:

R

code

ggplot(spotify\_data, aes(x = danceability, y = energy, color = cluster)) +

geom\_point(alpha = 0.6) +

labs(title = "Danceability vs Energy by Cluster")

3. Tips for Execution

Ensure the dataset matches the column names referenced in the code (e.g., danceability, energy, track\_genre).

Debug any warnings/errors by verifying the dataset structure and adjusting column names if needed.

Run code sections incrementally to validate each step before proceeding.

4. Expected Output

EDA Visualizations:

Boxplots, scatterplots, and histograms showing feature distributions and relationships.

Ridge Regression Results:

MSE and R² values indicating the model’s accuracy.

Random Forest Results:

Accuracy and feature importance for genre classification.

Clustering Results:

Visual representation of clusters based on music features.