**credit-risk-analytics-dashboard/**

│

├── data/

│ └── sample\_dataset.csv # (de-identified sample data for demo)

│

├── scripts/

│ ├── data\_cleaning.R # R script for preprocessing

│ ├── model\_training.R # Logistic regression + confusion matrix

│ └── scoring\_script.R # Generates PD per borrower

│

├── dashboard/

│ └── powerbi\_dashboard.pbix # Power BI dashboard file

│

├── reports/

│ ├── summary\_report.pdf # Executive summary or business insights

│ └── dashboard\_user\_manual.pdf # User guide

│

├── visuals/

│ └── screenshots/ # PNGs of charts, KPI cards, etc.

│

├── LICENSE # MIT or Apache 2.0

├── README.md # Overview and instructions

└── .gitignore

**R-STRUCTURE**

#1.Load Necessary Libraries

# Install (if needed) and load required packages

install.packages(c("tidyverse", "ggplot2", "caret", "corrplot", "ROCR"))

library(tidyverse)

library(caret)

library(corrplot)

library(ROCR)

#2. Import Your Dataset

data <- read.csv("financial\_risk\_dataset.csv", stringsAsFactors = TRUE)

data <- read.csv(file.choose())

glimpse(data)

summary(data)

#3. Clean and Prepare the Data

# Handle missing values (remove or impute)

data <- na.omit(data)

# Convert Default\_Status to factor for classification

data$Default\_Status <- as.factor(data$Default\_Status)

# Optional: scale numeric features

data$Debt\_to\_Income\_Ratio <- data$Existing\_Debt / (data$Applicant\_Income + data$Coapplicant\_Income + 1)

data <- read.csv(file.choose(), stringsAsFactors = TRUE) # or your correct file path

head(data)

nrow(data)

data$Debt\_to\_Income\_Ratio <- data$Existing\_Debt / (data$Applicant\_Income + data$Coapplicant\_Income + 1)

#Exploratory Data Analysis (EDA)

# Histogram of Credit Score

ggplot(data, aes(Credit\_Score)) +

geom\_histogram(fill = "skyblue", bins = 30) +

theme\_minimal() +

labs(title = "Credit Score Distribution")

# Boxplot of Loan Amount by Default Status

ggplot(data, aes(Default\_Status, Loan\_Amount)) +

geom\_boxplot(fill = "orange") +

labs(title = "Loan Amount vs. Default Status")

#Correlation matrix:

numeric\_data <- data %>% select\_if(is.numeric)

corrplot(cor(numeric\_data, use = "complete.obs"), method = "circle")

#Predictive Model\_Logistic Regression

set.seed(123)

trainIndex <- createDataPartition(data$Default\_Status, p = 0.8, list = FALSE)

train <- data[trainIndex, ]

test <- data[-trainIndex, ]

model <- glm(Default\_Status ~ Credit\_Score + Debt\_to\_Income\_Ratio + Payment\_Delays\_6mo + Credit\_Utilization\_Ratio,

data = train, family = binomial)

summary(model)

table(data$Default\_Status)

str(data$Default\_Status)

data$Default\_Status <- as.factor(data$Default\_Status)

table(data$Default\_Status) # Should show at least 2 levels

library(caret)

set.seed(123)

trainIndex <- createDataPartition(data$Default\_Status, p = 0.8, list = FALSE)

train <- data[trainIndex, ]

test <- data[-trainIndex, ]

model <- glm(Default\_Status ~ Credit\_Score + Debt\_to\_Income\_Ratio +

Payment\_Delays\_6mo + Credit\_Utilization\_Ratio,

data = train, family = binomial)

summary(model)

table(train$Default\_Status)

colSums(is.na(train))

train\_clean <- na.omit(train)

apply(train\_clean[, c("Credit\_Score", "Debt\_to\_Income\_Ratio",

"Payment\_Delays\_6mo", "Credit\_Utilization\_Ratio")], 2, var)

model <- glm(Default\_Status ~ Credit\_Score + Payment\_Delays\_6mo,

data = train\_clean, family = binomial)

summary(model)

table(train$Default\_Status)

install.packages("caret") # for confusionMatrix

install.packages("e1071") # required dependency

library(caret)

library(e1071)

# Predict probabilities

pred\_probs <- predict(model, newdata = test, type = "response")

# Classify using a 0.5 threshold

pred\_class <- ifelse(pred\_probs > 0.5, 1, 0)

# Ensure both predicted and actual are factors

pred\_class <- as.factor(pred\_class)

test$Default\_Status <- as.factor(test$Default\_Status)

conf\_matrix <- confusionMatrix(pred\_class, test$Default\_Status, positive = "1")

print(conf\_matrix)

# Create a table

cm\_table <- table(Predicted = pred\_class, Actual = test$Default\_Status)

# Plot it

library(ggplot2)

library(reshape2)

cm\_melt <- melt(cm\_table)

ggplot(cm\_melt, aes(x = Actual, y = Predicted, fill = value)) +

geom\_tile() +

geom\_text(aes(label = value), color = "white", size = 8) +

scale\_fill\_gradient(low = "steelblue", high = "darkred") +

labs(title = "Confusion Matrix", x = "Actual", y = "Predicted") +

theme\_minimal()