**Week 3 - Task 3: Optimizing AI-Generated Code**

### **User-Generated Prompt for AI (Example)**

"I have a dataset containing environmental measurements, including columns for Date, Temperature, Humidity, and Wind Speed. I need a Python script that:\*

* **Loads the dataset from a CSV file.**
* **Performs data cleaning (removes missing values and handles incorrect data types).**
* **Identifies and removes redundant computations or loops.**
* **Optimizes performance using vectorized operations instead of inefficient loops.**
* **Saves the optimized dataset as a new CSV file.** Can you generate an efficient Python script for these tasks?"\*

### **AI-Generated Python Code for Task 3**

import pandas as pd

def optimize\_environmental\_data(file\_path):

"""

Optimizes AI-generated environmental data processing by removing inefficiencies.

"""

try:

# Load dataset

df = pd.read\_csv(file\_path)

# Handle missing values

df.dropna(inplace=True)

# Ensure correct data types

df['Date'] = pd.to\_datetime(df['Date'], errors='coerce')

df['Temperature'] = pd.to\_numeric(df['Temperature'], errors='coerce')

df['Humidity'] = pd.to\_numeric(df['Humidity'], errors='coerce')

df['Wind Speed'] = pd.to\_numeric(df['Wind Speed'], errors='coerce')

# Remove redundant computations (AI might generate repeated calculations)

df = df.drop\_duplicates()

# Optimize computations: Use vectorized operations instead of loops

df['Temperature\_F'] = df['Temperature'] \* 9/5 + 32 # Convert Celsius to Fahrenheit

# Save optimized dataset

optimized\_file\_path = "optimized\_" + file\_path

df.to\_csv(optimized\_file\_path, index=False)

print(f"Optimized dataset saved as {optimized\_file\_path}")

return df

except Exception as e:

print("Error processing the dataset:", e)

return None

# Example usage

file\_path = "Environmental\_Dataset\_example.csv"

processed\_data = optimize\_environmental\_data(file\_path)

**Week 3 - Task 3: Optimizing AI-Generated Code (R Version)**

**User-Generated Prompt for AI (Example)**

*"I have a dataset containing environmental measurements, including columns for Date, Temperature, Humidity, and Wind Speed. I need an R script that:*\*

* **Loads the dataset from a CSV file.**
* **Performs data cleaning (removes missing values and handles incorrect data types).**
* **Identifies and removes redundant computations or loops.**
* **Optimizes performance using vectorized operations instead of inefficient loops.**
* **Saves the optimized dataset as a new CSV file.** Can you generate an efficient R script for these tasks?"\*

**AI-Generated R Code for Task 3**

library(dplyr)

optimize\_environmental\_data <- function(file\_path) {

"""

Optimizes AI-generated environmental data processing by removing inefficiencies.

"""

tryCatch({

# Load dataset

df <- read.csv(file\_path, stringsAsFactors = FALSE)

# Handle missing values

df <- na.omit(df)

# Ensure correct data types

df$Date <- as.Date(df$Date, format="%Y-%m-%d")

df$Temperature <- as.numeric(df$Temperature)

df$Humidity <- as.numeric(df$Humidity)

df$Wind.Speed <- as.numeric(df$Wind.Speed)

# Remove redundant computations (AI might generate repeated calculations)

df <- distinct(df)

# Optimize computations: Use vectorized operations instead of loops

df <- df %>% mutate(Temperature\_F = Temperature \* 9/5 + 32) # Convert Celsius to Fahrenheit

# Save optimized dataset

optimized\_file\_path <- paste0("optimized\_", file\_path)

write.csv(df, optimized\_file\_path, row.names = FALSE)

print(paste("Optimized dataset saved as", optimized\_file\_path))

return(df)

}, error = function(e) {

print(paste("Error processing the dataset:", e))

return(NULL)

})

}

# Example usage

file\_path <- "Environmental\_Dataset\_example.csv"

processed\_data <- optimize\_environmental\_data(file\_path)