Algorithm Developed:  
**1. For Predicting Import Values:**  
  
# Load the required library (prophet)  
install.packages("prophet")  
library(prophet)

# Create a dataframe for the import values (replace this with your new dataset)  
# for reading the datset  
data <- read.csv("agriculture.csv")  
# Prepare the data in Prophet format  
prophet\_data <- data.frame(ds = as.Date(paste(data$Year, "-01-01", sep = "")), y = data$United\_States)  
# Create a Prophet model  
model <- prophet(yearly.seasonality = TRUE, weekly.seasonality = FALSE, daily.seasonality = FALSE)  
model <- fit.prophet(model, prophet\_data)

# Generate future years for predictions  
future\_years <- 2023:2030

# Create a dataframe for predictions  
predict\_data <- data.frame(ds = as.Date(paste(future\_years, "-01-01", sep = "")))

# Make predictions for each region  
prediction\_table <- data.frame(Year = future\_years)  
regions <- colnames(data)[-1]

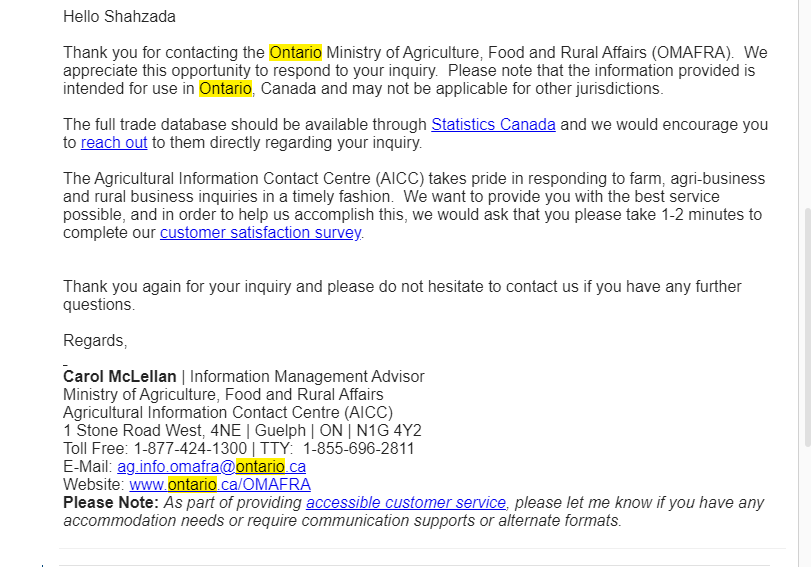
for (region in regions) {  
prophet\_data <- data.frame(ds = as.Date(paste(data$Year, "-01-01", sep = "")), y = data[[region]])  
model <- prophet(yearly.seasonality = TRUE, weekly.seasonality = FALSE, daily.seasonality = FALSE)   
model <- fit.prophet(model, prophet\_data)  
forecast <- predict(model, predict\_data)  
prediction\_table[[region]] <- forecast$yhat  
}  
# Print the prediction table  
print(prediction\_table)  
  
**1.1 For Plotting Import Graph:**  
  
install.packages("ggplot2")  
install.packages("tidyr")  
library(ggplot2)  
library(tidyr)   
# Convert the data to a long format for plotting  
library(tidyr)  
data\_long <- gather(data, key = "Region", value = "Export\_Value", -Year)  
# Plot the graph  
library(ggplot2)  
ggplot(data\_long, aes(x = Year, y = Export\_Value, color = Region)) +  
geom\_line(size = 1) +  
geom\_point(size = 3) +  
labs(x = "Year", y = "Import Values", title = "Import Values by Region (2023 to 2030)") +  
scale\_x\_continuous(breaks = seq(2023, 2030, 1)) +  
theme\_minimal()  
  
**2. For Predicting Export Values:  
  
library(prophet)**

**# Create a dataframe for the export values**

# for reading the datset  
data <- read.csv("agriculture.csv")

**# Prepare the data in Prophet format  
prophet\_data <- data.frame(ds = as.Date(paste(data$Year, "-01-01", sep = "")), y = data$United\_States)  
# Create a Prophet model  
model <- prophet(yearly.seasonality = TRUE, weekly.seasonality = FALSE, daily.seasonality = FALSE)  
model <- fit.prophet(model, prophet\_data)  
# Generate future years for predictions  
future\_years <- 2023:2030  
# Create a dataframe for predictions  
predict\_data <- data.frame(ds = as.Date(paste(future\_years, "-01-01", sep = "")))  
# Make predictions for each region  
prediction\_table <- data.frame(Year = future\_years)  
regions <- colnames(data)[-1]  
for (region in regions) {  
prophet\_data <- data.frame(ds = as.Date(paste(data$Year, "-01-01", sep = "")), y = data[[region]])  
model <- prophet(yearly.seasonality = TRUE, weekly.seasonality = FALSE, daily.seasonality = FALSE)  
model <- fit.prophet(model, prophet\_data)  
forecast <- predict(model, predict\_data)  
prediction\_table[[region]] <- forecast$yhat**

**}  
# Print the prediction table  
print(prediction\_table)  
  
2.1 For Plotting Export Graph:  
install.packages("ggplot2")  
install.packages("tidyr")  
library(ggplot2)  
library(tidyr)  
# Convert the data to a long format for plotting  
library(tidyr)  
data\_long <- gather(data, key = "Region", value = "Export\_Value", -Year)  
# Plot the graph  
library(ggplot2)  
ggplot(data\_long, aes(x = Year, y = Export\_Value, color = Region)) +  
geom\_line(size = 1) +  
geom\_point(size = 3) +  
labs(x = "Year", y = "Export Values", title = "Export Values by Region (2023 to 2030)") +  
scale\_x\_continuous(breaks = seq(2023, 2030, 1)) +  
theme\_minimal()  
  
For Checking the Resulting Factors:  
  
Code for Checking Accuracy for Export in the United States Region:  
  
# Load required libraries  
library(dplyr)  
# Actual values from 2009 to 2022  
actual\_values <- c(7042, 7239, 7679, 8080, 8765, 9241, 10553, 11199, 11272, 11911, 13043, 13803, 15645, 19029)  
# Predicted values from 2023 to 2030  
predicted\_values <- c(16936.98, 17425.34, 18805.15, 19520.78, 20122.65, 20611.01, 21990.81, 22706.45)  
  
# Calculate MAE  
mae <- mean(abs(actual\_values - predicted\_values))  
print(mae)  
# Calculate MSE  
mse <- mean((actual\_values - predicted\_values)^2)  
print(mse)  
# Calculate RMSE  
rmse <- sqrt(mse)  
print(rmse)  
Note: Our model performs well and accurate on MAE and RMSE but not as much as accurate in the MSE.**

**Additional Work:  
**