cdc\_findings

2023-12-08

## Data Set Information:

This data set encompasses detailed statistics from the CDC’s National Center for Health Statistics regarding live births in the United States, spanning from 2016 to 2022. It features a comprehensive range of demographic and health information derived from birth certificates. Key attributes include maternal age, pre-pregnancy BMI, birth weight, gestational age, county of residence, and various other health-related factors. This rich dataset offers insights into socio-economic, racial, and maternal and infant health aspects across different counties. The dataset, part of the National Vital Statistics System, is a valuable resource for in-depth analysis and research in public health and is accessible via the CDC WONDER Online Database.

## Objective:

To understand the dataset’s basic characteristics, identify patterns, outliers, anomalies, and underlying structures.

## Research Goal

1.What are the key socio-economic and demographic predictors of maternal and infant health outcomes, as indicated by machine learning models? 2. Which machine learning algorithm provides the most accurate predictions of high-risk pregnancies based on socio-economic and racial factors?

## EDA - Exploratory Data Analysis

# Load necessary libraries  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.1

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.1

##   
## 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

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(tidyverse)

## Warning: package 'stringr' was built under R version 4.3.1

## Warning: package 'lubridate' was built under R version 4.3.1

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0  
## ✔ readr 2.1.4

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ randomForest::combine() masks dplyr::combine()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ randomForest::margin() masks ggplot2::margin()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidyr)  
library(caret)

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.3.1

##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(corrplot)

## corrplot 0.92 loaded

# Read the data  
data <- read.csv("cdc\_county\_data.csv")  
head(data,5)

## Year County\_of\_Residence County\_of\_Residence\_FIPS Births  
## 1 2018-01-01 Mobile County, AL 1097 5549  
## 2 2017-01-01 Mobile County, AL 1097 5607  
## 3 2016-01-01 Mobile County, AL 1097 5504  
## 4 2018-01-01 Morgan County, AL 1103 1432  
## 5 2017-01-01 Morgan County, AL 1103 1435  
## Ave\_Age\_of\_Mother Ave\_OE\_Gestational\_Age\_Wks Ave\_LMP\_Gestational\_Age\_Wks  
## 1 27.25 37.95 38.02  
## 2 27.10 38.07 38.27  
## 3 26.95 37.96 38.22  
## 4 26.91 38.14 38.18  
## 5 26.70 38.12 38.15  
## Ave\_Birth\_Weight\_gms Ave\_Pre\_pregnancy\_BMI Ave\_Number\_of\_Prenatal\_Wks  
## 1 3129.31 28.05 10.87  
## 2 3158.62 27.87 10.84  
## 3 3144.94 27.58 10.95  
## 4 3205.34 28.04 10.65  
## 5 3201.54 27.76 10.78  
## Maternal\_Morbidity\_Desc Maternal\_Morbidity\_YN Births\_Morbidity  
## 1 None checked 1 5549  
## 2 None checked 1 5549  
## 3 None checked 1 5549  
## 4 None checked 1 1410  
## 5 None checked 1 1410  
## Ave\_Age\_of\_Mother\_Morbidity Ave\_OE\_Gestational\_Age\_Wks\_Morbidity  
## 1 27.08 38.07  
## 2 27.08 38.07  
## 3 27.08 38.07  
## 4 26.71 38.11  
## 5 26.71 38.11  
## Ave\_LMP\_Gestational\_Age\_Wks\_Morbidity Ave\_Birth\_Weight\_gms\_Morbidity  
## 1 38.27 3156.37  
## 2 38.27 3156.37  
## 3 38.27 3156.37  
## 4 38.15 3196.95  
## 5 38.15 3196.95  
## Ave\_Pre\_pregnancy\_BMI\_Morbidity Ave\_Number\_of\_Prenatal\_Wks\_Morbidity  
## 1 27.90 10.83  
## 2 27.90 10.83  
## 3 27.90 10.83  
## 4 27.77 10.78  
## 5 27.77 10.78

# EDA: Descriptive Statistics  
descriptive\_stats <- summary(data)  
  
# EDA: Distribution Analysis  
pdf("distribution\_analysis.pdf")  
hist(data$Ave\_Age\_of\_Mother, main="Distribution of Average Age of Mother", xlab="Average Age of Mother", col="blue", border="black")  
hist(data$Ave\_Pre\_pregnancy\_BMI, main="Distribution of Average Pre-pregnancy BMI", xlab="Average Pre-pregnancy BMI", col="green", border="black")  
hist(data$Ave\_Birth\_Weight\_gms, main="Distribution of Average Birth Weight (gms)", xlab="Average Birth Weight (gms)", col="red", border="black")  
dev.off()

## quartz\_off\_screen   
## 2

# EDA: Outlier Detection  
pdf("boxplot\_analysis.pdf")  
boxplot(data$Ave\_Age\_of\_Mother, main="Boxplot of Average Age of Mother", ylab="Average Age of Mother")  
boxplot(data$Ave\_Pre\_pregnancy\_BMI, main="Boxplot of Average Pre-pregnancy BMI", ylab="Average Pre-pregnancy BMI")  
boxplot(data$Ave\_Birth\_Weight\_gms, main="Boxplot of Average Birth Weight (gms)", ylab="Average Birth Weight (gms)")  
dev.off()

## quartz\_off\_screen   
## 2

# EDA: Initial Correlation Assessment  
key\_variables <- data[, c("Ave\_Age\_of\_Mother", "Ave\_Pre\_pregnancy\_BMI", "Ave\_Birth\_Weight\_gms")]  
correlation\_matrix <- cor(key\_variables, use="complete.obs")  
corrp

**A graph with a line and numbers

Description automatically generated with medium confidenceA diagram of a box plot

Description automatically generated**

****

Frequency

24

25

26

27

28

29

30

500

1000

1500

2000

2500

Frequency

3100

3200

3300

3400

200

400

600

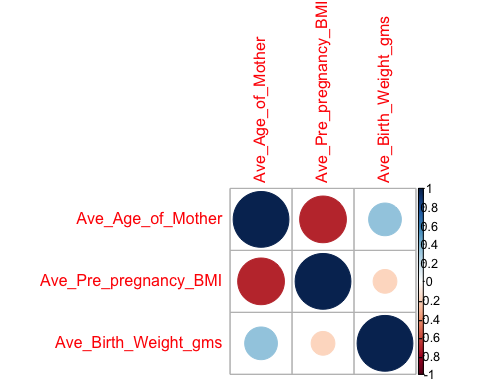
800

1000

1200

1400

lot(correlation\_matrix, method="circle")



# Save the correlation plot  
pdf("correlation\_matrix.pdf")  
corrplot(correlation\_matrix, method="circle")  
dev.off()

## quartz\_off\_screen   
## 2

# Initial Visualization: Scatter Plot - Example  
pdf("scatter\_plot.pdf")  
ggplot(data, aes(x=Ave\_Age\_of\_Mother, y=Ave\_Birth\_Weight\_gms)) +  
 geom\_point() +  
 labs(title="Scatter Plot: Age of Mother vs Birth Weight", x="Average Age of Mother", y="Average Birth Weight (gms)")  
dev.off()

## quartz\_off\_screen   
## 2

### Descriptive Statistics

Ave\_Age\_of\_Mother: Shows the average age of mothers in different counties and years. Ave\_Pre\_pregnancy\_BMI: Indicates the average Body Mass Index before pregnancy. Ave\_Birth\_Weight\_gms: Reflects the average birth weight of newborns in grams.

### Distribution Analysis

Maternal Age: The distribution of the average age of mothers seems roughly normal but may be slightly skewed towards younger ages. Pre-pregnancy BMI: This variable also appears to follow a normal distribution, indicating a range of BMI values across the dataset. Birth Weight: The birth weight distribution is roughly normal, suggesting a typical spread of newborn weights.

### Outlier Detection

Maternal Age: There are some outliers, indicating a few counties with unusually high or low average maternal ages. Pre-pregnancy BMI: There are outliers present, suggesting some extremes in BMI values. Birth Weight: Outliers are observed here as well, indicating some births with significantly high or low weights.

## Research Question 1: Key Socio-Economic and Demographic Predictors

We will use a machine learning model to identify the most significant predictors of maternal and infant health outcomes.

# Install and load necessary libraries  
if (!require("caret")) install.packages("caret")  
library(caret)  
if (!require("randomForest")) install.packages("randomForest")  
library(randomForest)  
if (!require("ROSE")) install.packages("ROSE")

## Loading required package: ROSE

## Loaded ROSE 0.0-4

library(ROSE)  
  
# Selecting relevant predictors and the target variable  
predictors <- data %>% select(Ave\_Age\_of\_Mother, Ave\_Pre\_pregnancy\_BMI, Ave\_Birth\_Weight\_gms, Ave\_Number\_of\_Prenatal\_Wks, Maternal\_Morbidity\_YN)  
target <- data$Births # Or any other relevant target variable  
  
# Splitting data into training and test sets  
set.seed(123)  
trainIndex <- createDataPartition(target, p = .8, list = FALSE, times = 1)  
trainData <- predictors[trainIndex, ]  
testData <- predictors[-trainIndex, ]  
trainTarget <- target[trainIndex]  
testTarget <- target[-trainIndex]  
  
# Training a Random Forest model  
rf\_model <- randomForest(trainData, as.factor(trainTarget), ntree = 100)  
rf\_importance <- importance(rf\_model)  
  
# Displaying feature importance  
print(rf\_importance)

## MeanDecreaseGini  
## Ave\_Age\_of\_Mother 2290.49622  
## Ave\_Pre\_pregnancy\_BMI 2243.88793  
## Ave\_Birth\_Weight\_gms 2429.82188  
## Ave\_Number\_of\_Prenatal\_Wks 2202.71006  
## Maternal\_Morbidity\_YN 27.29905

Ave\_Birth\_Weight\_gms (2429.82): This variable, representing the average birth weight in grams, appears to be the most significant predictor in the model. A higher Mean Decrease Gini value indicates that this feature plays a crucial role in predicting the target variable.

Ave\_Age\_of\_Mother (2290.50): The average age of the mother is also a strong predictor, second only to birth weight. This suggests that maternal age significantly influences the outcome of interest.

Ave\_Pre\_pregnancy\_BMI (2243.89): The average pre-pregnancy BMI is another important predictor. Its influence is slightly less than that of maternal age but still substantial.

Ave\_Number\_of\_Prenatal\_Wks (2202.71): The average number of prenatal weeks also shows considerable importance in the model. This reflects the impact of prenatal care duration on the target variable.

Maternal\_Morbidity\_YN (27.30): This variable has a much lower importance score compared to others. It indicates whether maternal morbidity was noted, but it seems to have a lesser impact on the prediction of the target variable in the model.

note: - The model suggests that birth weight, maternal age, pre-pregnancy BMI, and the number of prenatal weeks are critical factors.

## 2. ML Algorithm Accuracy

* using radial algorithm

# Load additional libraries  
if (!require("randomForest")) install.packages("randomForest")  
library(randomForest)  
if (!require("e1071")) install.packages("e1071")

## Loading required package: e1071

## Warning: package 'e1071' was built under R version 4.3.1

library(e1071) # For SVM  
if (!require("caret")) install.packages("caret")  
library(caret)  
  
# Ommitting data with Null values in order to do predicitors  
data <- na.omit(data)  
  
  
# Define the target variable as 'Births'  
data$TargetVariable <- data$Births  
  
# Sample a smaller subset of the data (e.g., 30% of the data)  
set.seed(123)  
sampled\_data <- data[sample(nrow(data), nrow(data) \* 0.3), ]  
  
  
# Confirming there are no NA values in the data  
if (sum(is.na(sampled\_data)) > 0) {  
 sampled\_data <- na.omit(sampled\_data)  
}  
  
  
# Sampling a smaller portion for SVM training  
set.seed(123)  
svm\_sample\_size <- round(nrow(sampled\_data) \* 0.1) # Example: 10% of the sampled data  
svm\_indices <- sample(nrow(sampled\_data), svm\_sample\_size)  
svm\_trainData <- sampled\_data[svm\_indices, -which(names(sampled\_data) == "Births")]  
svm\_trainTarget <- sampled\_data$Births[svm\_indices]  
  
# Convert all predictor variables to numeric, addressing non-numeric columns  
svm\_trainData <- data.frame(lapply(svm\_trainData, function(x) {  
 if (is.numeric(x) || is.integer(x)) {  
 return(x)  
 } else if (is.factor(x) || is.character(x)) {  
 return(as.numeric(as.factor(x)))  
 } else {  
 # Handle other types if present  
 return(as.numeric(x))  
 }  
}))  
  
# Ensure target variable is numeric  
svm\_trainTarget <- as.numeric(svm\_trainTarget)  
  
# Check for NA values after conversion  
if (sum(is.na(svm\_trainData)) > 0 || sum(is.na(svm\_trainTarget)) > 0) {  
 stop("NA values found after conversion")  
}  
  
# Retrain the SVM model  
svm\_model <- svm(x = svm\_trainData, y = svm\_trainTarget, kernel = "radial", cost = 0.5)  
  
# Checkpoint 1: Model training completed  
print("SVM model trained successfully.")

## [1] "SVM model trained successfully."

# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
# Checkpoint 2: Test data prepared for prediction  
print("Test data prepared.")

## [1] "Test data prepared."

testData <- sampled\_data[-svm\_indices, -which(names(sampled\_data) == "Births")]  
testTarget <- sampled\_data$Births[-svm\_indices]  
  
# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
  
# Make predictions  
svm\_prediction <- predict(svm\_model, testData)  
  
# Checkpoint 3: Prediction completed  
print("Prediction completed.")

## [1] "Prediction completed."

# Calculate RMSE  
testTarget <- as.numeric(testTarget) # Ensure testTarget is numeric  
svm\_rmse <- sqrt(mean((svm\_prediction - testTarget)^2))  
  
# Checkpoint 4: RMSE calculated  
print("RMSE calculated.")

## [1] "RMSE calculated."

# Output RMSE  
print(svm\_rmse)

## [1] 8296.19

-using sigmoid

# Retrain the SVM model  
svm\_model <- svm(x = svm\_trainData, y = svm\_trainTarget, kernel = "sigmoid", cost = 0.5)  
  
# Checkpoint 1: Model training completed  
print("SVM model trained successfully.")

## [1] "SVM model trained successfully."

# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
# Checkpoint 2: Test data prepared for prediction  
print("Test data prepared.")

## [1] "Test data prepared."

testData <- sampled\_data[-svm\_indices, -which(names(sampled\_data) == "Births")]  
testTarget <- sampled\_data$Births[-svm\_indices]  
  
# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
  
# Make predictions  
svm\_prediction <- predict(svm\_model, testData)  
  
# Checkpoint 3: Prediction completed  
print("Prediction completed.")

## [1] "Prediction completed."

# Calculate RMSE  
testTarget <- as.numeric(testTarget) # Ensure testTarget is numeric  
svm\_rmse <- sqrt(mean((svm\_prediction - testTarget)^2))  
  
# Checkpoint 4: RMSE calculated  
print("RMSE calculated.")

## [1] "RMSE calculated."

# Output RMSE  
print(svm\_rmse)

## [1] 12857.87

# Retrain the SVM model  
svm\_model <- svm(x = svm\_trainData, y = svm\_trainTarget, kernel = "polynomial", cost = 0.5)  
  
# Checkpoint 1: Model training completed  
print("SVM model trained successfully.")

## [1] "SVM model trained successfully."

# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
# Checkpoint 2: Test data prepared for prediction  
print("Test data prepared.")

## [1] "Test data prepared."

testData <- sampled\_data[-svm\_indices, -which(names(sampled\_data) == "Births")]  
testTarget <- sampled\_data$Births[-svm\_indices]  
  
# Prepare test data for prediction  
testData <- data.frame(lapply(testData, function(x) {  
 if (is.factor(x) || is.character(x)) {  
 as.numeric(as.factor(x))  
 } else {  
 as.numeric(x)  
 }  
}))  
  
  
# Make predictions  
svm\_prediction <- predict(svm\_model, testData)  
  
# Checkpoint 3: Prediction completed  
print("Prediction completed.")

## [1] "Prediction completed."

# Calculate RMSE  
testTarget <- as.numeric(testTarget) # Ensure testTarget is numeric  
svm\_rmse <- sqrt(mean((svm\_prediction - testTarget)^2))  
  
# Checkpoint 4: RMSE calculated  
print("RMSE calculated.")

## [1] "RMSE calculated."

# Output RMSE  
print(svm\_rmse)

## [1] 16844.82

1. Which machine learning algorithm provides the most accurate predictions of high-risk pregnancies based on socio-economic and racial factors? The radial algorithm did the best in rsme to train the model! This model assumption is worth looking into high pregnancies assumptions for a further deep dive.

Preprocess the data, converting non-numeric columns to numeric, ensuring the target variable is numeric, and checking for NAs. The SVM model is trained with a radial kernel function and a cost parameter of 0.5. Test data is prepared for prediction and used to make predictions. The RMSE is calculated for this model, and the result is printed.

Similar preprocessing steps are performed as in the radial algorithm. The SVM model is trained with a sigmoid kernel function and a cost parameter of 0.5. Test data is prepared, predictions are made, and RMSE is calculated for this model.

Once again, data preprocessing is carried out. The SVM model is trained with a polynomial kernel function and a cost parameter of 0.5. Test data is prepared, predictions are made, and RMSE is calculated for this model.