County_Poisson_regression.R

rocka

2023-12-04

```r
df <- read.csv("C:/Users/rocka/OneDrive/Documents/county_surgery_data.csv", header = TRUE)

# Filtering records for the year 2016 onwards
df_filtered <- subset(df, Year >= 2016)

# Ensure 'County' and 'Surgery' are factors
df_filtered$County <- as.factor(df_filtered$County)
df_filtered$Surgery <- as.factor(df_filtered$Surgery)


# Building the initial model on the filtered data
model_initial <- glm(TotalCases ~ Year + County + Surgery, family = 'poisson', data = df_filtered)

# Displaying summary of the initial model
summary(model_initial)
```

```
##
## Call:
## glm(formula = TotalCases ~ Year + County + Surgery, family = "poisson",
##     data = df_filtered)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           21.189675   2.036584  10.405  < 2e-16 ***
## Year                  -0.008703   0.001009  -8.626  < 2e-16 ***
## CountyAmador          -5.128262   0.115034 -44.581  < 2e-16 ***
## CountyButte           -1.562443   0.021213 -73.654  < 2e-16 ***
## CountyCalaveras       -5.546973   0.141693 -39.148  < 2e-16 ***
## CountyContra Costa    -0.453713   0.014168 -32.023  < 2e-16 ***
## CountyDel Norte       -5.196316   0.119006 -43.664  < 2e-16 ***
## CountyEl Dorado       -3.016456   0.040870 -73.806  < 2e-16 ***
## CountyFresno          -0.468803   0.014234 -32.935  < 2e-16 ***
## CountyHumboldt        -2.562301   0.033002 -77.642  < 2e-16 ***
## CountyImperial        -3.285210   0.046490 -70.665  < 2e-16 ***
## CountyInyo            -4.926396   0.104070 -47.337  < 2e-16 ***
## CountyKern            -1.164946   0.018110 -64.325  < 2e-16 ***
## CountyKings           -3.492849   0.051401 -67.953  < 2e-16 ***
## CountyLake            -5.064546   0.111424 -45.453  < 2e-16 ***
## CountyLassen          -6.759624   0.447289 -15.112  < 2e-16 ***
## CountyLos Angeles      2.035287   0.009390 216.749  < 2e-16 ***
## CountyMadera          -4.038461   0.067103 -60.183  < 2e-16 ***
```

```
## CountyMarin           -1.556878   0.021165 -73.560  < 2e-16 ***
## CountyMendocino       -2.909345   0.038841 -74.905  < 2e-16 ***
## CountyMerced          -3.487734   0.051274 -68.022  < 2e-16 ***
## CountyMonterey        -1.910967   0.024599 -77.683  < 2e-16 ***
## CountyNapa            -2.673408   0.034755 -76.921  < 2e-16 ***
## CountyNevada          -2.912210   0.038893 -74.877  < 2e-16 ***
## CountyOrange           0.869073   0.010521  82.605  < 2e-16 ***
## CountyPlacer          -0.797010   0.015844 -50.304  < 2e-16 ***
## CountyPlumas          -5.697796   0.152754 -37.300  < 2e-16 ***
## CountyRiverside        0.089244   0.012219   7.304 2.80e-13 ***
## CountySacramento       0.302640   0.011645  25.989  < 2e-16 ***
## CountySan Benito      -5.040155   0.110119 -45.770  < 2e-16 ***
## CountySan Bernardino   0.177592   0.011970  14.836  < 2e-16 ***
## CountySan Diego        0.883166   0.010499  84.119  < 2e-16 ***
## CountySan Francisco    0.287429   0.011683  24.603  < 2e-16 ***
## CountySan Joaquin     -0.989104   0.016961 -58.317  < 2e-16 ***
## CountySan Luis Obispo -2.243021   0.028509 -78.679  < 2e-16 ***
## CountySan Mateo       -0.587911   0.014778 -39.784  < 2e-16 ***
## CountySanta Barbara   -1.342280   0.019403 -69.178  < 2e-16 ***
## CountySanta Clara      0.528281   0.011134  47.448  < 2e-16 ***
## CountySanta Cruz      -2.612053   0.033774 -77.339  < 2e-16 ***
## CountyShasta          -2.073145   0.026419 -78.473  < 2e-16 ***
## CountySiskiyou        -4.377591   0.079304 -55.200  < 2e-16 ***
## CountySolano          -1.284575   0.018967 -67.726  < 2e-16 ***
## CountySonoma          -1.295055   0.019045 -67.999  < 2e-16 ***
## CountyStanislaus      -1.228685   0.018559 -66.204  < 2e-16 ***
## CountySutter          -3.077180   0.042072 -73.141  < 2e-16 ***
## CountyTehama          -4.805035   0.097982 -49.040  < 2e-16 ***
## CountyTrinity         -7.688884   1.000041  -7.689 1.49e-14 ***
## CountyTulare          -1.936595   0.024876 -77.850  < 2e-16 ***
## CountyTuolumne        -3.312666   0.047110 -70.318  < 2e-16 ***
## CountyVentura         -0.840872   0.016087 -52.271  < 2e-16 ***
## CountyYolo            -3.077180   0.042072 -73.141  < 2e-16 ***
## CountyYuba            -3.559098   0.053082 -67.049  < 2e-16 ***
## SurgeryBrain           1.217011   0.014824  82.097  < 2e-16 ***
## SurgeryBreast          3.377834   0.013241 255.097  < 2e-16 ***
## SurgeryColon           2.018018   0.013859 145.606  < 2e-16 ***
## SurgeryEsophagus      -0.763012   0.023091 -33.044  < 2e-16 ***
## SurgeryLiver           0.452278   0.016656  27.155  < 2e-16 ***
## SurgeryLung            1.163119   0.014918  77.970  < 2e-16 ***
## SurgeryPancreas        0.103782   0.017955   5.780 7.46e-09 ***
## SurgeryProstate        1.649258   0.014217 116.003  < 2e-16 ***
## SurgeryRectum          1.209755   0.014836  81.539  < 2e-16 ***
## SurgeryStomach         0.002709   0.018402   0.147    0.883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1302358  on 3266  degrees of freedom
## Residual deviance:   31246  on 3205  degrees of freedom
## AIC: 42991
##
## Number of Fisher Scoring iterations: 5
```

```r
#Splitting the data into testing and training sets
set.seed(1234)
nobs <- nrow(df_filtered)
train <- sample(1:nobs, 2600)
test <- setdiff(1:nobs, train)

train_data <- df_filtered[train, ]
test_data <- df_filtered[test, ]

# Building the model on the training set
model_train <- glm(TotalCases ~ Year + County + Surgery, family = 'poisson', data = train_data)

# Predictions on the training set
predictions_train <- predict(model_train, newdata = train_data, type = "response")

# Calculate Mean Squared Error (MSE) for the training set
mse_train <- mean((train_data$TotalCases - predictions_train)^2)
rmse_train <- sqrt(mse_train)

print(paste("Mean Squared Error (MSE) on the training set:", mse_train))
```

```
## [1] "Mean Squared Error (MSE) on the training set: 3320.24020401863"
```

```r
print(paste("Root Mean Squared Error (RMSE) on the training set:", rmse_train))
```

```
## [1] "Root Mean Squared Error (RMSE) on the training set: 57.6215255266522"
```

```r
# Predictions on the test set
predictions_test <- predict(model_train, newdata = test_data, type = "response")

# Calculate Mean Squared Error (MSE) for the test set
mse_test <- mean((test_data$TotalCases - predictions_test)^2)
rmse_test <- sqrt(mse_test)

print(paste("Mean Squared Error (MSE) on the test set:", mse_test))
```

```
## [1] "Mean Squared Error (MSE) on the test set: 2180.34950883501"
```

```r
print(paste("Root Mean Squared Error (RMSE) on the test set:", rmse_test))
```

```
## [1] "Root Mean Squared Error (RMSE) on the test set: 46.6942127981082"
```

```r
# Testing new data for prediction by using an unknown year 2022
new_data <- data.frame(Year = 2022, County = "Los Angeles", Surgery = "Breast")

# Predict using the new data
predicted_cases <- predict(model_train, newdata = new_data, type = "response")

# Display the predicted cases
print(round(predicted_cases, 0))
```

```
##    1
## 8128
```