Regression

Muhammad Zaid

2023-02-18

Linear Regression seeks to find a best fit linear line between a actual and predicted value. The value being predicted is a dependent variable, where the variables being used to make the prediction are independent variables. The predicted value is dependent on these independent variables. The goal for linear regression is the find the line that allows for the smallest distance between the predicted and actual value. Linear regression only works on quantitative data.

Data-set: https://www.kaggle.com/datasets/ahadulislam/significant-earthquake-dataset-1900-2023

Data Pre-processing

Load Data from csv file

```
dataset <-  read.csv(file = 'Significant Earthquake Dataset 1900-2023.csv')
```

Remove unneeded columns from data-set

```
dataset <- subset(dataset, select = -c(X) )
lineardf <- subset(dataset, select = -c(Time, Place, MagType, ID, Updated, Type, status, locationSource, magSource, net))
```

Determine columns to drop based on the amount of NA

```
colSums(is.na(lineardf))
```

```
##       Latitude      Longitude          Depth            Mag            nst 
##              0              0              0            134          29858 
##            gap           dmin            rms horizontalError    depthError 
##          27244          32936          17113          33361          16504 
##     magError      magNst 
##          28788          31959
```

```
lineardf <- subset(lineardf, select = -c(nst,gap,dmin,rms, horizontalError, depthError, magError, magNst))
lineardf <- na.omit(lineardf)

nrow(lineardf)
```

```
## [1] 37197
```

Divide the training and testing sets

```
train_indices <- sample(1:nrow(lineardf), 0.8*nrow(lineardf), replace=FALSE)
trainDf <- lineardf[train_indices,]
testDf <- lineardf[-train_indices,]
```

Data Exploration of the training set

```
nrow(trainDf)
```

```
## [1] 29757
```

```
summary(trainDf)
```

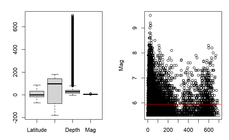
```
##     Latitude         Longitude          Depth             Mag        
##  Min.   : -69.774   Min.   :-180.00   Min.   :  -1.00   Min.   :5.500  
##  1st Qu.:-16.621   1st Qu.: -75.66   1st Qu.:  15.00   1st Qu.:5.600  
##  Median :  1.096   Median :  98.00   Median :  28.50   Median :5.800  
##  Mean   :  5.373   Mean   :  38.76   Mean   :  58.47   Mean   :5.951  
##  3rd Qu.: 33.407   3rd Qu.: 143.34   3rd Qu.:  41.00   3rd Qu.:6.160  
##  Max.   : 87.199   Max.   : 180.00   Max.   :700.00   Max.   :9.500
```

```
str(trainDf)
```

```
## 'data.frame':    29757 obs. of  4 variables:
##  $ Latitude : num  -14 -16.3 38.5 -29.6 34.4 ...
##  $ Longitude: num  -72.2 168.1 34.8 -176.7 -3.9 ...
##  $ Depth    : num  15 28.8 228.9 8.15 ...
##  $ Mag      : num  5.57 5.5 5.5 6.5 5.55 5.5 6.26 5.5 5.5 5.7 ...
##  - attr(*, "na.action")= 'omit' Named int [1:134] 25629 27700 27751 28029 28081 28445 28511 28593 28982 29025 ...
##   ..- attr(*, "names")= chr [1:134] "25629" "27700" "27751" "28029" ...
```

Useful plots

```
par(mfrow=c(1,2))
boxplot(trainDf)
plot(trainDf$Mag~trainDf$Depth, xlab="Depth", ylab = "Mag" )
abline(lm(trainDf$Mag~trainDf$Depth),col="red")
```



Model

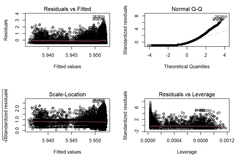
```
linear_model <- lm(Mag~Depth, data=trainDf)
summary(linear_model)
```

```
##
## Call:
## lm(formula = Mag ~ Depth, data = trainDf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.4524 -0.3515 -0.1508  0.2879  3.5482 
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)    
## (Intercept)  5.9524e+00  3.003e-03 1982.139   <2e-16 ***
## Depth       -2.3616e-03  2.417e-05  -97.707   0.329    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.457 on 29755 degrees of freedom
## Multiple R-squared:  1.286e-05,  Adjusted R-squared:  -1.523e-06 
## F-statistic: 0.9547 on 1 and 29755 DF,  p-value: 0.3285
```

The summary results seem to suggest that the prediction was not quite accurate, with the Median residuals showing how far the prediction strayed from the actual value in the model prediction. It also does not have a symmetrical shape to its output. The average depth calculated by the Estimate for the Intercept tells us how deep the average of all the magnitudes of an earthquake would be. The SlopeMag tells us how the depth decreases with a 1 magnitude increase. This varies however by about 1.3000. For our model the t-values are way closer to zero than we would have wanted to see, this means that there is a chance for the null hypothesis to be true that their is no relationship between the two variables. This is just confirmed by the slopes p value, which is far higher than what we would like to see.

Plots

```
par(mfrow=c(2,2))
plot(linear_model)
```



Even though the residuals vs Fitted graph gives us a nice linear line, the place and distribution of the graph are very abnormal. For the Normal Q-Q graph the first few values are nicely following the dashed line until about 1 on the horizontal axis where the deviate highly. The residuals seem to be spread closer to the left side of the graph for the Scale-Location, while the line also seems to neglect the higher y-value fitted values.

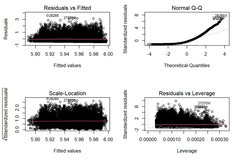
Multiple predictor linear model

```
lm2 <- lm(Mag~ Latitude + Longitude, data=trainDf)
summary(lm2)
```

```
##
## Call:
## lm(formula = Mag ~ Latitude + Longitude, data = trainDf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.4967 -0.3456 -0.1118  0.2008  3.5835 
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)    
## (Intercept) 5.944e+00  2.794e-03 2127.241  < 2e-16 ***
## Latitude    4.694e-04  8.721e-05    5.383 7.36e-08 ***
## Longitude   1.251e-04  2.138e-05    5.736 9.76e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4565 on 29754 degrees of freedom
## Multiple R-squared:  0.001481,   Adjusted R-squared:  0.002414 
## F-statistic:    37 on 2 and 29754 DF,  p-value: < 2.2e-16
```

Plots

```
par(mfrow=c(2,2))
plot(lm2)
```



This models shows promising results for rejecting the null hypothesis for the longitude and latitude variables, who have lower p values, and higher t values. Depth remains to be unrelated as was expected. Standard error this time is also exceptionally low for the model. However the plots still seem to suggest concern for the models accuracy. The plots suggest that the model still suffers from the same problems just like the model with one predictor for the Normal Q-Q graph otherwise the other graphs show a expected and wanted result.

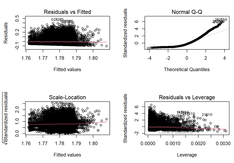
Model 3 with tuning

```
lm3 <- lm(log(Mag)~ Latitude + Longitude + Depth + I(Depth^2), data=trainDf)
summary(lm3)
```

```
##
## Call:
## lm(formula = log(Mag) ~ Latitude + Longitude + Depth + I(Depth^2), 
##     data = trainDf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.10331 -0.05660 -0.01923  0.03625  0.47557 
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)    
## (Intercept)  1.783e+00  6.365e-04 2801.138  < 2e-16 ***
## Latitude     7.365e-05  1.196e-05    5.278 1.32e-07 ***
## Longitude    2.299e-05  3.456e-06    6.575 4.93e-11 ***
## Depth       -1.894e-04  1.199e-05   -7.867 3.76e-15 ***
## I(Depth^2)   2.001e-07  2.440e-08    8.070 7.31e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0727 on 29752 degrees of freedom
## Multiple R-squared:  0.004755,   Adjusted R-squared:  0.004621 
## F-statistic: 35.53 on 4 and 29752 DF,  p-value: < 2.2e-16
```

Plots

```
par(mfrow=c(2,2))
plot(lm3)
```



Comparing the Models

After looking at all the summaries and plots for the three models built in this notebook, I find that Model three seems to perform the best. My decisions relies on many different factors in both the summary and the plots. In the summary there is an obvious relationship between the predictor and the target variable. This inference is based on the t-value, the p-value, and the F-statistic. There is also higher symmetry for the Residuals in the summary. For the plots, they all seem to be better than the ones for the other models, especially, the Normal Q-Q graph which had the residuals way close to the line than any of the other graphs.

All Coefficients and Mse

```
print("Linear Model 1\n")
```

```
## [1] "Linear Model 1\n"
```

```
pred <- predict(linear_model, newdata=testDf)
correlation <- cor(pred, testDf$Mag)
mse <- mean((pred-testDf$Mag)^2)
print(paste("MSE:", mse ) )
```

```
## [1] "MSE: 0.198278483932469"
```

```
print(paste("Correlation:", correlation ) )
```

```
## [1] "Correlation: 0.00071451541726977"
```

```
print("Linear Model 2\n")
```

```
## [1] "Linear Model 2\n"
```

```
pred <- predict(lm2, newdata=testDf)
corr2 <- cor(pred, testDf$Mag)
mse2 <- mean((pred-testDf$Mag)^2)

print(paste("MSE:", mse2 ) )
```

```
## [1] "MSE: 0.197851319292914"
```

```
print(paste("Correlation:", corr2 ) )
```

```
## [1] "Correlation: 0.0467968324152774"
```

```
print("Linear Model 3\n")
```

```
## [1] "Linear Model 3\n"
```

```
pred <- predict(lm3, newdata=testDf)
corr3 <- cor(pred, testDf$Mag)
mse3 <- mean((pred-testDf$Mag)^2)

print(paste("MSE:", mse3 ) )
```

```
## [1] "MSE: 17.4872513673825"
```

```
print(paste("Correlation:", corr3 ) )
```

```
## [1] "Correlation: 0.0179718217816121"
```