DataMining_Process

```
               STEPS FOR DATA MINING PROCESS..........

1) load the dataset
2) clean the data or pre-process the data
3) understand data use visual effact that is- (ggplot2::)
4) split your data into test and train
5) create model on train & predict on test.
6) Than see the model accuracy ........
```

```
                   AT THE END OF STEP CHECK YOUR MODEL ACCURECY.......
```

```
               IF YOU SEE THIS SIGN(#) IT MEANS IT'S OPTIONAL PART.....
```

STEP-1 : Load the dataset

```
library(psych)#(Optional)
Data = read.csv("C:/Users/Dell/Desktop/data.csv")
describe(Data)#(Optional)
```

```
##          vars   n   mean     sd median trimmed    mad   min    max  range skew
## Duration    1 169  63.85  42.30   60.0   55.66  22.24  15.0  300.0  285.0 2.81
## Pulse       2 169 107.46  14.51  105.0  105.61   8.90  80.0  159.0   79.0 1.39
## Maxpulse    3 169 134.05  16.45  131.0  132.88  13.34 100.0  184.0   84.0 0.69
## Calories    4 164 375.79 266.38  318.6  326.90 100.89  50.3 1860.4 1810.1 3.05
##          kurtosis    se
## Duration     9.70  3.25
## Pulse        2.40  1.12
## Maxpulse     0.59  1.27
## Calories    11.41 20.80
```

```
#You can also use summary function for Summarization purpose.........
```

STEP-2 : Clean the data or pre-process the data

```
cal = mean(Data$Calories,na.rm = T)
Data$Calories = ifelse(is.na(Data$Calories),cal,Data$Calories)
```

STEP-3 : To understand data we use graph

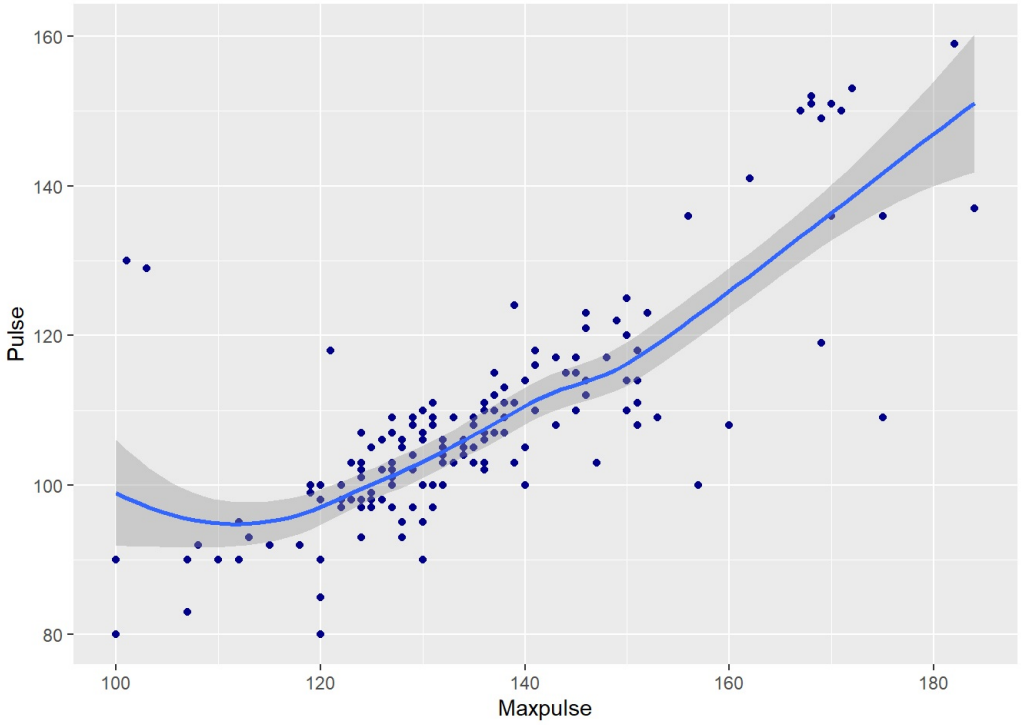
```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
ggplot(Data) + geom_point(aes(x = Maxpulse,y = Pulse),color = "darkblue") + geom_smooth(aes(x = Maxpulse ,y = Pul
se))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



STEP-4 : Split your data into test and train

```
Data$no = c(1:dim(Data)[1])#(We have to add new column for split data into test & train)

#(As below we split data into test & train)
Train = subset(Data,Data$no <= 135)
Test = subset(Data,Data$no > 135)

#(After split data we have to delete column which we added)
Data = Data[-5]
Train = Train[-5]
Test = Test[-5]
```

STEP-5 : Create model on train & predict on test

```
lm_model = lm(Duration ~.,data = Train)
predect = predict(lm_model ,newdata = Test)
```

```
compare_result = cbind(actual = Test$Duration , predect = predect)#(we created another dataframe and we added actual test column and predicted test column for check the error..or model)
compare_result = as.data.frame(compare_result)
compare_result$error = compare_result$actual - compare_result$predec #(Now in same dataframe we added new column which is actual test data -(minus) predicted test data)
rmse = sqrt(mean(compare_result$error^2)) #(Now we do Square Root mean on error column.)
```

```
summary(lm_model)#(Now.. AS BELOW YOU CAN SEE OUR MODEL IS 0.8911% CORRECT....)
```

```
##
## Call:
## lm(formula = Duration ~ ., data = Train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.571  -6.236  -1.493   4.383  77.691
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 81.412326  10.817716   7.526 7.48e-12 ***
## Pulse       -0.367320   0.148372  -2.476   0.0146 *
## Maxpulse    -0.260752   0.129215  -2.018   0.0456 *
## Calories     0.151828   0.004812  31.552  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.27 on 131 degrees of freedom
## Multiple R-squared:  0.8936, Adjusted R-squared:  0.8911
## F-statistic: 366.7 on 3 and 131 DF,  p-value: < 2.2e-16
```