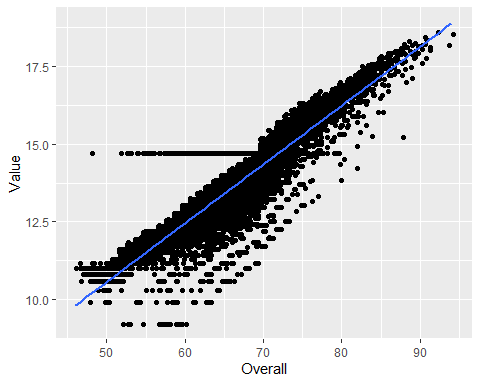
6\_regression.r

abhia

2019-11-19

fifa = read.csv("1\_cleaned\_data.csv")  
options(scipen = 999)  
library("ggplot2")  
  
##CORRELATION  
#SCATTER PLOT BETWEEN LOG(VALUE) VS OVERALL  
print(ggplot(fifa, aes(x=Overall, y =unlist(lapply(lapply(fifa$Value, log), as.numeric))))+ylab("Value") + geom\_jitter()+geom\_smooth(method = lm))



#PEARSON CORRELATION TEST  
print(cor.test(unlist(lapply(lapply(fifa$Value, log), as.numeric)), fifa$Overall, method="pearson"))

##   
## Pearson's product-moment correlation  
##   
## data: unlist(lapply(lapply(fifa$Value, log), as.numeric)) and fifa$Overall  
## t = 343.89, df = 18204, p-value < 0.00000000000000022  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.9289499 0.9328261  
## sample estimates:  
## cor   
## 0.9309142

##REGRESSION  
  
#CREATING THE TRAINING AND TEST DATA  
set.seed(100)  
trainingRows<-sample(1:nrow(fifa), 0.8\*nrow(fifa))  
trainingData<-fifa[trainingRows, ]  
testData<-fifa[-trainingRows, ]  
  
#FIT THE MODEL ON TRAINING DATA AND PREDICT ON TEST DATA  
model = lm(log(Value)~Overall, data = trainingData)  
predictedData = predict(model,testData)  
predictedData = exp(predictedData)  
  
  
#REVIEW DIAGNOSTIC MEASURES  
summary(model)

##   
## Call:  
## lm(formula = log(Value) ~ Overall, data = trainingData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.2437 -0.2204 0.0497 0.3128 4.5086   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.117921 0.040926 27.32 <0.0000000000000002 \*\*\*  
## Overall 0.188936 0.000614 307.69 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5104 on 14562 degrees of freedom  
## Multiple R-squared: 0.8667, Adjusted R-squared: 0.8667   
## F-statistic: 9.467e+04 on 1 and 14562 DF, p-value: < 0.00000000000000022

##CALCULATE PREDICTION ACCURACY AND ERROR RATES  
diff = data.frame(cbind(actuals=testData$Value, predicteds=predictedData))  
  
#CORERLATION ACCURACY  
correlation\_accuracy<-cor(diff)  
print(correlation\_accuracy)

## actuals predicteds  
## actuals 1.0000000 0.9395014  
## predicteds 0.9395014 1.0000000

#MIN-MAX ACCURACY CALCULATION  
min\_max\_accuracy <-mean(apply(diff, 1, min)/apply(diff,1,max))  
print(min\_max\_accuracy)

## [1] 0.7325135

#MEAN-ABSOLUTE-PERCENTAGE-ERROR (MAPE) CALCULATION  
mape <- mean(abs((diff$predicteds - diff$actuals))/diff$actuals)