R Notebook

Code

Hide

```
#Loading the data file
attach(parkinsons_updrs)
```

Hide

```
#1-Divide the data into training(80%) and testing(20%)

#Divide the data into training and testing sets
library(caret)
library(ggplot2)
library(lattice)
set.seed(42) # for reproducibility

# Create a vector of row indices
rows <- 1:nrow(parkinsons_updrs)


# Randomly sample 80% of the row indices for the training set
training_rows <- sample(rows, floor(0.5 * length(rows)))

# The remaining rows are for the testing set
testing_rows <- setdiff(rows, training_rows)

# Write the training and testing sets to separate files
write.table(parkinsons_updrs[training_rows, ], file = "Park_training_data3.txt", row.names =
FALSE, col.names = FALSE)
write.table(parkinsons_updrs[testing_rows, ], file = "Park_testing_data3.txt", row.names = FA
LSE, col.names = FALSE)

training_data <- parkinsons_updrs[training_rows, ]
testing_data <- parkinsons_updrs[-training_rows, ]

# Remove the variable 'motor_UPDRS' (Training and Testing)
training_data_new <- subset(training_data, select = -motor_UPDRS)
testing_data_new <- subset(testing_data, select = -motor_UPDRS)

#a) Division Verification in number of Examples
cat("Number of examples in training data:", nrow(training_data_new), "\n")
```

```
Number of examples in training data: 2937
```

Hide

```
cat("Number of examples in testing data:", nrow(testing_data_new), "\n")
```

```
Number of examples in testing data: 2938
```

Hide

```
# Multiple Regression Model
parkinsons_updrs_model=lm(total_UPDRS~., data=training_data_new)

# Use the model to make predictions on the testing data
predictions <- predict(parkinsons_updrs_model, newdata = testing_data_new)

# Calculate the residuals
residuals <- predictions - testing_data_new$total_UPDRS

# Calculate the RMSE
rmse <- sqrt(mean(residuals^2))
rmse
```

```
[1] 9.173239
```

Hide

```
# Calculate the RSE
rse <- rmse / sqrt(nrow(testing_data_new))
rse
```

```
[1] 0.1692376
```

Hide

```
#R-Squared
R2 <- summary(parkinsons_updrs_model)$r.squared
R2
```

```
[1] 0.245939
```

Hide

```
# Multiple Regression Model with Interaction

parkinsons_updrs_model_with_interaction=lm(total_UPDRS~.+(Shimmer.dB.*Jitter.Abs.), data=trai
ning_data_new)

# Use the model to make predictions on the testing data
predictions1 <- predict(parkinsons_updrs_model_with_interaction, newdata = testing_data_new)

# Significant predictors
summary(parkinsons_updrs_model_with_interaction)
```

```
Call:
lm(formula = total_UPDRS ~ . + (Shimmer.dB. * Jitter.Abs.), data = training_data_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.152  -6.900  -1.022   6.908  23.303 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              3.378e+01  4.404e+00   7.669 2.35e-14 ***
subject.                 2.623e-01  1.548e-02  16.946  < 2e-16 ***
age                      3.090e-01  2.051e-02  15.066  < 2e-16 ***
sex                     -4.874e+00  4.478e-01 -10.885  < 2e-16 ***
test_time                1.905e-02  3.285e-03   5.797 7.46e-09 ***
Jitter...               -4.450e+02  2.956e+02  -1.506 0.132265    
Jitter.Abs.             -9.022e+04  1.682e+04  -5.365 8.75e-08 ***
Jitter.RAP              -8.123e+03  6.333e+04  -0.128 0.897958    
Jitter.PPQ5             -6.191e+02  2.915e+02  -2.124 0.033744 *  
Jitter.DDP               3.217e+03  2.111e+04   0.152 0.878888    
Shimmer                 -7.920e+01  8.750e+01  -0.905 0.365498    
Shimmer.dB.              5.192e+00  7.078e+00   0.734 0.463299    
Shimmer.APQ3            -7.492e+04  6.421e+04  -1.167 0.243427    
Shimmer.APQ5             1.110e+02  7.616e+01   1.457 0.145246    
Shimmer.APQ11            3.860e+00  3.482e+01   0.111 0.911737    
Shimmer.DDA              2.490e+04  2.140e+04   1.163 0.244758    
NHR                     -3.552e+01  9.341e+00  -3.802 0.000146 ***
HNR                     -5.230e-01  9.727e-02  -5.377 8.18e-08 ***
RPDE                     4.857e+00  2.534e+00   1.917 0.055333 .  
DFA                     -3.336e+01  3.211e+00 -10.391  < 2e-16 ***
PPE                      2.709e+01  4.244e+00   6.384 2.00e-10 ***
Jitter.Abs.:Shimmer.dB.  8.425e+04  1.950e+04   4.321 1.61e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.35 on 2915 degrees of freedom
Multiple R-squared:  0.2507,	Adjusted R-squared:  0.2453 
F-statistic: 46.45 on 21 and 2915 DF,  p-value: < 2.2e-16
```

Hide

```
summary(predictions1)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  15.42   25.08   28.69   29.04   33.37   44.19 
```

Hide

```
# Extraction of the RSE and R-squared values
# Calculate the residuals
residuals1 <- predictions1 - testing_data_new$total_UPDRS

# Calculate the RMSE
rmse1 <- sqrt(mean(residuals1^2))
rmse1
```

```
[1] 9.179639
```

Hide

```
# Calculate the RSE
rse1 <- rmse1 / sqrt(nrow(testing_data_new))
rse1
```

```
[1] 0.1693557
```

Hide

```
#R-Squared
R2_1 <- summary(parkinsons_updrs_model_with_interaction)$r.squared
R2_1
```

```
[1] 0.2507376
```

Hide

```
# Multiple Regression Model with non-linear transformation

parkinsons_updrs_model_with_non_linear_transformation=lm(total_UPDRS~.+I(Shimmer.dB.^2), data
=training_data_new)

# Use the model to make predictions on the testing data
predictions2 <- predict(parkinsons_updrs_model_with_non_linear_transformation, newdata = test
ing_data_new)

# Significant predictors
summary(parkinsons_updrs_model_with_non_linear_transformation)
```

```
Call:
lm(formula = total_UPDRS ~ . + I(Shimmer.dB.^2), data = training_data_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-27.438  -6.800  -1.246   7.079  23.817 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       3.280e+01  4.457e+00   7.358 2.42e-13 ***
subject.          2.658e-01  1.548e-02  17.169  < 2e-16 ***
age               3.103e-01  2.080e-02  14.922  < 2e-16 ***
sex              -4.808e+00  4.486e-01 -10.717  < 2e-16 ***
test_time         1.930e-02  3.293e-03   5.860 5.14e-09 ***
Jitter...        -5.401e+02  2.979e+02  -1.813 0.069991 .  
Jitter.Abs.      -4.341e+04  1.296e+04  -3.349 0.000821 ***
Jitter.RAP       -1.681e+04  6.345e+04  -0.265 0.791072    
Jitter.PPQ5      -2.367e+02  2.733e+02  -0.866 0.386449    
Jitter.DDP        6.118e+03  2.115e+04   0.289 0.772426    
Shimmer          -8.497e+01  8.931e+01  -0.951 0.341506    
Shimmer.dB.       9.623e-01  7.054e+00   0.136 0.891490    
Shimmer.APQ3     -6.625e+04  6.435e+04  -1.029 0.303346    
Shimmer.APQ5      6.662e+01  7.600e+01   0.877 0.380780    
Shimmer.APQ11     2.803e+01  3.451e+01   0.812 0.416680    
Shimmer.DDA       2.204e+04  2.145e+04   1.027 0.304313    
NHR              -2.578e+01  9.092e+00  -2.836 0.004604 ** 
HNR              -4.597e-01  9.776e-02  -4.703 2.69e-06 ***
RPDE              3.787e+00  2.524e+00   1.500 0.133667    
DFA              -3.450e+01  3.205e+00 -10.764  < 2e-16 ***
PPE               2.200e+01  4.029e+00   5.461 5.12e-08 ***
I(Shimmer.dB.^2)  5.046e+00  2.282e+00   2.211 0.027116 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.372 on 2915 degrees of freedom
Multiple R-squared:  0.2472,	Adjusted R-squared:  0.2418 
F-statistic: 45.58 on 21 and 2915 DF,  p-value: < 2.2e-16
```

Hide

```
summary(predictions2)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  15.47   25.15   28.70   29.04   33.27   56.18 
```

Hide

```
# Extraction of the RSE and R-squared values
# Calculate the residuals
residuals2 <- predictions2 - testing_data_new$total_UPDRS

# Calculate the RMSE
rmse2 <- sqrt(mean(residuals2^2))
rmse2
```

```
[1] 9.174831
```

Hide

```
# Calculate the RSE
rse2 <- rmse2 / sqrt(nrow(testing_data_new))
rse2
```

```
[1] 0.169267
```

Hide

```
#R-Squared
R2_2 <- summary(parkinsons_updrs_model_with_non_linear_transformation)$r.squared
R2_2
```

```
[1] 0.2472014
```

Hide

```
#b) Analyzing the performance of the model we can conclude that overall, looking at the Perfo
rmance coefficient the model performs poorly, but other techniques may be applied for improve
ment.
```

Hide

```
#2- LOOCV
# Multiple Linear Regression
library(boot)
library(Metrics)

glm.fit=glm(total_UPDRS~.,data=training_data_new)

# cv.glm(): produces a list with several components
cv.err=cv.glm(training_data_new,glm.fit)

# The two numbers in the delta vector contain the cross-validation results
# Standard estimate & bias-corrected
cv.err$delta
```

```
[1] 88.47039 88.47018
```

Hide

```
cv.error=rep(0,5)
for (i in 1:5){
  glm.fit=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RAP
+ Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.APQ
11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE, family = gaussian, data = training_data_new)
  cv.error[i]=cv.glm(training_data_new, glm.fit)$delta[1]
}
cv.error
```

```
[1] 88.47039 88.47039 88.47039 88.47039 88.47039
```

Hide

```
# Predict on testing data using the fitted model
test_predictions = predict(glm.fit, newdata = testing_data_new)

#LOOCV - Multiple Regression
postResample(test_predictions, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE
9.1732386 0.2563938 7.5253642
```

Hide

```
glm.fit2=glm(total_UPDRS~.+(Shimmer.dB.*Jitter.Abs.),data=training_data_new)

# cv.glm(): produces a list with several components
cv.err=cv.glm(training_data_new,glm.fit2)

# The two numbers in the delta vector contain the cross-validation results
# Standard estimate & bias-corrected
cv.err$delta
```

```
[1] 87.89746 87.89726
```

Hide

```
cv.error=rep(0,5)
for (i in 1:5){
  glm.fit2=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RA
P + Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.A
PQ11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE+(Shimmer.dB.*Jitter.Abs.), family = gaussian, da
ta = training_data_new)
  cv.error[i]=cv.glm(training_data_new, glm.fit2)$delta[1]
}
cv.error
```

```
[1] 87.89746 87.89746 87.89746 87.89746 87.89746
```

Hide

```
# Predict on testing data using the fitted model
test_predictions_2 = predict(glm.fit2, newdata = testing_data_new)

#LOOCV - Multiple Regression with Interaction term
postResample(test_predictions_2, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE 
9.1796393 0.2554357 7.5331618 
```

Hide

```
glm.fit3=glm(total_UPDRS~.+I(Shimmer.dB.^2),data=training_data_new)

# cv.glm(): produces a list with several components
cv.err=cv.glm(training_data_new,glm.fit3)

# The two numbers in the delta vector contain the cross-validation results
# Standard estimate & bias-corrected
cv.err$delta
```

```
[1] 88.43614 88.43592
```

Hide

```
cv.error=rep(0,5)
for (i in 1:5){
  glm.fit3=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RA
P + Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.A
PQ11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE+I(Shimmer.dB.^2), family = gaussian, data = trai
ning_data_new)
  cv.error[i]=cv.glm(training_data_new, glm.fit3)$delta[1]
}
cv.error
```

```
[1] 88.43614 88.43614 88.43614 88.43614 88.43614
```

Hide

```
# Predict on testing data using the fitted model
test_predictions_3 = predict(glm.fit3, newdata = testing_data_new)

#LOOCV - Multiple Regression with non-linear transformation
postResample(test_predictions_3, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE 
9.1748309 0.2561518 7.5135149 
```

Hide

```
set.seed(17)
cv.error.10=rep(0,10)
for (i in 1:10){
  glm.fit4=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RA
P + Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.A
PQ11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE, family = gaussian, data = training_data_new)
  cv.error.10[i]=cv.glm(training_data_new, glm.fit4, K=10)$delta[1]
}
cv.error.10
```

```
 [1] 88.52718 88.21618 88.37092 88.63245 88.54856 88.49954 88.41154
 [8] 88.50637 88.66553 88.50193
```

Hide

```
#Make predictions on the testing data using the trained model
predictions_4 <- predict(glm.fit4, newdata = testing_data_new)

#K=10 Multiple Regression
postResample(predictions_4, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE
9.1732386 0.2563938 7.5253642
```

Hide

```
set.seed(17)
cv.error.10=rep(0,10)
for (i in 1:10){
  glm.fit5=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RA
P + Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.A
PQ11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE+(Shimmer.dB.*Jitter.Abs.), family = gaussian, da
ta = training_data_new)
  cv.error.10[i]=cv.glm(training_data_new, glm.fit5, K=10)$delta[1]
}
cv.error.10
```

```
 [1] 88.01265 87.66204 87.81379 87.96497 87.90431 87.91479 87.83533
 [8] 87.92205 88.15034 87.90933
```

Hide

```
#Make predictions on the testing data using the trained model
predictions_5 <- predict(glm.fit5, newdata = testing_data_new)

#K=10 Multiple Regression with Interaction term
postResample(predictions_5, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE
9.1796393 0.2554357 7.5331618
```

Hide

```
set.seed(17)
cv.error.10=rep(0,10)
for (i in 1:10){
  glm.fit6=glm(total_UPDRS ~ subject. +age+sex+ test_time +Jitter...+ Jitter.Abs. + Jitter.RAP + Jitter.PPQ5 + Jitter.DDP + Shimmer +Shimmer.dB. + Shimmer.APQ3 + Shimmer.APQ5 + Shimmer.APQ11 + Shimmer.DDA+ NHR + HNR +RPDE +DFA+PPE+I(Shimmer.dB.^2), family = gaussian, data = training_data_new)
  cv.error.10[i]=cv.glm(training_data_new, glm.fit6, K=10)$delta[1]
}
cv.error.10
```

```
 [1] 88.46466 88.18130 88.35050 88.61538 88.45633 88.42367 88.35083
 [8] 88.44937 88.63237 88.51967
```

Hide

```
#Make predictions on the testing data using the trained model
predictions_6 <- predict(glm.fit6, newdata = testing_data_new)

#K=10 Multiple Regression with non-linear transformation
postResample(predictions_6, testing_data_new$total_UPDRS)
```

```
     RMSE  Rsquared       MAE 
9.1748309 0.2561518 7.5135149 
```

Hide

```
#2/3-a) After performing resampling and cross validation (LOOCV and 10-fold), from the R-Squared generated we can say that for each model slight increase on all of them but not enough to turn it good but it can be seen as positive technique for improvement.
```

Hide

```
#4- Comments

# Looking that the RMSE which focuses more on large error we can conclude both methods showed to have similar errors RMSE, with LOOCV overall having a couple of figures lower but not that significant. In the end, not looking at comparisons both techniques can have their advantages and disadvantages when looking at different applications
```