Dr Tony Diana

DATA 602 Introduction to Machine Learning

How to Use R in Regression Analysis

Load R and then RStudio to work with R

```
head(cars)  # display the first 6 observations
scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")  # scatterplot
par(mfrow=c(1, 2))  # divide graph area in 2 columns

boxplot(cars$speed, main="Speed", sub=paste("Outlier rows: ",
boxplot.stats(cars$speed)$out))  # box plot for 'speed'

boxplot(cars$dist, main="Distance", sub=paste("Outlier rows: ",
boxplot.stats(cars$dist)$out))  # box plot for 'distance'

library(e1071)  # for skewness function
par(mfrow=c(1, 2))  # divide graph area in 2 columns

plot(density(cars$speed), main="Density Plot: Speed", ylab="Frequency",
sub=paste("Skewness:", round(e1071::skewness(cars$speed), 2)))  # density plot for
'speed'

polygon(density(cars$speed), col="red")

plot(density(cars$dist), main="Density Plot: Distance", ylab="Frequency",
sub=paste("Skewness:", round(e1071::skewness(cars$dist), 2)))  # density plot for 'dist'

polygon(density(cars$dist), col="red")

cor(cars$speed, cars$dist)  # calculate correlation between speed and distance

linearMod <- lm(dist ~ speed, data=cars)  # build linear regression model on full data
print(linearMod)

summary(linearMod)  # model summary

# capture model summary as an object
modelSummary <- summary(linearMod)

# model coefficients
modelCoeffs <- modelSummary$coefficients
```

```r
# get beta estimate for speed
beta.estimate <- modelCoeffs["speed", "Estimate"]

# get std.error for speed
std.error <- modelCoeffs["speed", "Std. Error"]

# calc t statistic
t_value <- beta.estimate/std.error

# calc p Value
p_value <- 2*pt(-abs(t_value), df=nrow(cars)-ncol(cars))

# fstatistic
f_statistic <- linearMod$fstatistic[1]

# parameters for model p-value calc
f <- summary(linearMod)$fstatistic

model_p <- pf(f[1], f[2], f[3], lower=FALSE)

AIC(linearMod)

BIC(linearMod)

# Create Training and Test data -
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(cars), 0.8*nrow(cars))  # row indices for training data
trainingData <- cars[trainingRowIndex, ]  # model training data
testData  <- cars[-trainingRowIndex, ]   # test data

# Build the model on training data
lmMod <- lm(dist ~ speed, data=trainingData)  # build the model
distPred <- predict(lmMod, testData)  # predict distance

summary (lmMod)  # model summary

actuals_preds <- data.frame(cbind(actuals=testData$dist, predicteds=distPred))  # make actuals_predicteds dataframe.
correlation_accuracy <- cor(actuals_preds)  # 82.7%
head(actuals_preds)

# Min-Max Accuracy Calculation
```

```
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1,
max))

# MAPE Calculation
mape <- mean(abs((actuals_preds$predicteds -
actuals_preds$actuals))/actuals_preds$actuals)

DMwR::regr.eval(actuals_preds$actuals, actuals_preds$predicteds) # different way to
load a library and to run the regression

library(DAAG)
cvResults <- suppressWarnings(CVlm(df=cars, form.lm=dist ~ speed, m=5,
dots=FALSE, seed=29, legend.pos="topleft",  printit=FALSE, main="Small symbols are
predicted values while bigger ones are actuals."));  # performs the CV
attr(cvResults, 'ms')
```