```
install.packages("readxl")
library(readxl)
install.packages("dplyr")
library(dplyr)
install.packages("ggplot2")
library(ggplot2)
install.packages("lattice")
library(lattice)
install.packages("caTools")
library(caTools)
install.packages("e1071")
library(e1071)
install.packages("caret")
library(caret)
install.packages("randomForest")
library(randomForest)


 #Analysis of Sales Report of a Clothes Manufacturing Outlet#


read_excel("C:\\Users\\Public\\Skill\\Simplilearn\\Data Analyst\\Data Science
with R Programming\\Project\\Attribute DataSet.xlsx")

# load data
attribset = read_excel("C:\\Users\\Public\\Skill\\Simplilearn\\Data Analyst\
\Data Science with R Programming\\Project\\Attribute DataSet.xlsx")

dresssale = read_excel("C:\\Users\\Public\\Skill\\Simplilearn\\Data Analyst\
\Data Science with R Programming\\Project\\Dress Sales.xlsx")

#remove Dress_ID column
attribset_ = attribset[2:14]
dresssale_ = dresssale[2:24]

# check the unique values for each columns
lapply(attribset[2:14], unique)

# values checking
# style
attribset_$Style[attribset_$Style == 'sexy'] = 'Sexy'

# Price
attribset_$Price[attribset_$Price == 'low'] = 'Low'
attribset_$Price[attribset_$Price == 'high'] = 'High'

# Size
attribset_$Size[attribset_$Size == 's'] = 'S'
attribset_$Size[attribset_$Size == 'small'] = 'S'

# Season
attribset_$Season[attribset_$Season == 'Spring'] = 'Spring'
attribset_$Season[attribset_$Season == 'Summer'] = 'Summer'
attribset_$Season[attribset_$Season == 'Automn'] = 'Autumn'
```

```
attribset_$Season[attribset_$Season == 'winter'] = 'Winter'

# NeckLine
attribset_$NeckLine[attribset_$NeckLine == 'sweetheart'] = 'Sweetheart'


# SleeveLength
attribset_$SleeveLength[attribset_$SleeveLength == 'sleevless'] = 'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'sleeevless'] =
'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'sleveless'] = 'sleeveless'
attribset_$SleeveLength[attribset_$SleeveLength == 'threequater'] =
'threequarter'
attribset_$SleeveLength[attribset_$SleeveLength == 'thressqatar'] =
'threequarter'
attribset_$SleeveLength[attribset_$SleeveLength == 'urndowncollor'] =
'turndowncollar'

# FabricType
attribset_$FabricType[attribset_$FabricType == 'shiffon'] = 'chiffon'
attribset_$FabricType[attribset_$FabricType == 'sattin'] = 'satin'
attribset_$FabricType[attribset_$FabricType == 'wollen'] = 'woolen'
attribset_$FabricType[attribset_$FabricType == 'flannael'] = 'flannel'
attribset_$FabricType[attribset_$FabricType == 'knitting'] = 'knitted'

# Decoration
attribset_$Decoration[attribset_$Decoration == 'embroidary'] = 'embroidery'
attribset_$Decoration[attribset_$Decoration == 'sequined'] = 'sequins'
attribset_$Decoration[attribset_$Decoration == 'ruched'] = 'ruche'
attribset_$Decoration[attribset_$Decoration == 'none'] = 'null'

# Pattern Type
attribset_$'Pattern Type'[attribset_$'Pattern Type' == 'none'] = 'null'
attribset_$'Pattern Type'[attribset_$'Pattern Type' == 'leapord'] = 'leopard'

# factoring

attribset_$Style = factor(attribset_$Style,
                          levels = c('Sexy', 'Casual', 'vintage', 'Brief',
'cute', 'bohemian', 'Novelty', 'Flare', 'party', 'work', 'OL', 'fashion'),
                          labels = c(0,1,2,3,4,5,6,7,8,9,10,11))

attribset_$Price = factor(attribset_$Price,
                          levels = c('Low', 'High', 'Average', 'Medium',
'very-high'),
                          labels = c(0,1,2,3,4))

attribset_$Size = factor(attribset_$Size,
                         levels = c('M', 'L', 'XL', 'free', 'S'),
                         labels = c(0,1,2,3,4))

attribset_$Season = factor(attribset_$Season,
                           levels = c('Summer', 'Autumn', 'Spring', 'Winter'),
                           labels = c(0,1,2,3))
```

```
attribset_$NeckLine = factor(attribset_$NeckLine,
                             levels = c('o-neck', 'v-neck', 'boat-neck',
'peterpan-collor', 'ruffled', 'turndowncollor', 'slash-neck', 'mandarin-
collor', 'open', 'sqare-collor', 'Sweetheart', 'Scoop', 'halter', 'backless',
'bowneck', 'NULL'),
                             labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15))

attribset_$SleeveLength = factor(attribset_$SleeveLength,
                                 levels = c('sleeveless', 'Petal', 'full',
'butterfly', 'short', 'threequarter', 'halfsleeve', 'cap-sleeves',
'turndowncollor', 'capsleeves', 'half', 'turndowncollar', 'NULL'),
                                 labels = c(0,1,2,3,4,5,6,7,8,9,10,11,12))

attribset_$waiseline = factor(attribset_$waiseline,
                              levels = c('empire', 'natural', 'null',
'princess', 'dropped'),
                              labels = c(0,1,2,3,4))

attribset_$Material = factor(attribset_$Material,
                             levels = c('null', 'microfiber', 'polyster',
'silk', 'chiffonfabric', 'cotton', 'nylon', 'other', 'milksilk', 'linen',
'rayon', 'lycra', 'mix', 'acrylic', 'spandex', 'lace', 'modal', 'cashmere',
'viscos', 'knitting', 'sill', 'wool', 'model', 'shiffon'),
                             labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23))

attribset_$FabricType = factor(attribset_$FabricType,
                               levels = c('chiffon', 'null', 'broadcloth',
'jersey', 'other', 'batik', 'satin', 'flannel', 'worsted', 'woolen', 'poplin',
'dobby', 'knitted', 'tulle', 'organza', 'lace', 'Corduroy', 'terry'),
                               labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17))

attribset_$Decoration = factor(attribset_$Decoration,
                               levels = c('ruffles', 'null', 'embroidery',
'bow', 'lace', 'beading', 'sashes', 'hollowout', 'pockets', 'sequins',
'applique', 'button', 'Tiered', 'rivet', 'feathers', 'flowers', 'pearls',
'pleat', 'crystal', 'ruche', 'draped', 'tassel', 'plain', 'cascading'),
                               labels =
c(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23))

attribset_$`Pattern Type` = factor(attribset_$`Pattern Type`,
                                   levels = c('animal', 'print', 'dot',
'solid', 'null', 'patchwork', 'striped', 'geometric', 'plaid', 'leopard',
'floral', 'character', 'splice'),
                                   labels = c(0,1,2,3,4,5,6,7,8,9,10,11,12))

attribset_$Recommendation = sapply(attribset_$Recommendation, factor)


# count of missing values in attribset_ dataset
colSums(is.na(attribset_))
```

```
# Create the function.
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}


# fill missing Value with mode
attribset_$Price[is.na(attribset_$Price) ==TRUE] <- getmode(attribset_$Price)
attribset_$Season[is.na(attribset_$Season) ==TRUE] <-
getmode(attribset_$Season)
attribset_$NeckLine[is.na(attribset_$NeckLine) ==TRUE] <-
getmode(attribset_$NeckLine)
attribset_$waiseline[is.na(attribset_$waiseline) ==TRUE] <-
getmode(attribset_$waiseline)
attribset_$Material[is.na(attribset_$Material) ==TRUE] <-
getmode(attribset_$Material)
attribset_$FabricType[is.na(attribset_$FabricType) ==TRUE] <-
getmode(attribset_$FabricType)
attribset_$Decoration[is.na(attribset_$Decoration) ==TRUE] <-
getmode(attribset_$Decoration)
attribset_$`Pattern Type`[is.na(attribset_$`Pattern Type`) ==TRUE] <-
getmode(attribset_$`Pattern Type`)


attribset_data <- data.frame(attribset_)
str(attribset_data)

# Dresses dataset

# Update columns name in dresssale_ dataset

dresssale_ = rename(dresssale_,c('41314'='2/9/2013'))
dresssale_ = rename(dresssale_,c('41373'='4/9/2013'))
dresssale_ = rename(dresssale_,c('41434'='6/9/2013'))
dresssale_ = rename(dresssale_,c('41495'='8/9/2013'))
dresssale_ = rename(dresssale_,c('41556'='10/9/2013'))
dresssale_ = rename(dresssale_,c('41617'='12/9/2013'))
dresssale_ = rename(dresssale_,c('41315'='2/10/2013'))
dresssale_ = rename(dresssale_,c('41374'='4/10/2013'))
dresssale_ = rename(dresssale_,c('41435'='6/10/2013'))
dresssale_ = rename(dresssale_,c('40400'='8/10/2013'))
dresssale_ = rename(dresssale_,c('41557'='10/10/2013'))
dresssale_ = rename(dresssale_,c('41618'='12/10/2013'))


# Convert all variable types to numeric
dresssale_ <- as.data.frame(apply(dresssale_, 2, as.numeric))

# mean row
dresssale_ = as.matrix(dresssale_)
k <- which(is.na(dresssale_), arr.ind=TRUE)
```

```
dresssale_[k] <- rowMeans(dresssale_, na.rm=TRUE)[k[,1]]
dresssale_ = as.data.frame(dresssale_)

# sum all values on row on (total sales)
dresssale_$total_sales = rowSums(dresssale_)
head(dresssale_)

# Merged Data
merged_data <- data.frame(attribset_ ,dresssale_)
head(merged_data)

str(merged_data)

# spliting dataset
set.seed(100)

spl = sample.split(merged_data$Recommendation, SplitRatio = 0.7)
train = subset(merged_data, spl==TRUE)
test = subset(merged_data, spl==FALSE)

print(dim(train)); print(dim(test))


#Classification - Predict recommendation

#First model (Naive Bayes)

# naive bayes model
naive_model = naiveBayes(Recommendation ~.,data = train) # build model
confusionMatrix(train$Recommendation,predict(naive_model,train),positive =
'1') # create confusion Matrix
print('---------------')
naive_predict = predict(naive_model,test) # predict test set
table(naive_predict,test$Recommendation) # create table

# Second model (Support Vector Machine)


# Support vector machine
svm_model = svm(Recommendation ~.,train) # build model
confusionMatrix(train$Recommendation,predict(svm_model),positive = '1')#
create confusion Matrix
print('---------------')
svm_predict = predict(svm_model,test) # predict test set
table(svm_predict,test$Recommendation) # create table

#Third model (Random Forest)
# Random Forest
randomForest_model = randomForest(x = train, y = train$Recommendation,ntree
=800)# build model
confusionMatrix(train$Recommendation,predict(randomForest_model),positive =
'1') # create confusion Matrix
print('---------------')
randomForest_predict = predict(randomForest_model,test) # predict test set
```

```
table(randomForest_predict,test$Recommendation )# create table


#Regresstion model

# regression (total sales and (Style+Season+Material+Price))
regressor_Sales = lm(formula = total_sales ~ Style+Season+Material+Price, data
= train) # build model
summary(regressor_Sales) # print model summary
plot(regressor_Sales, pch = 16, col = "green") # Plot the results
abline(regressor_Sales) # Add regression line

# regression (total sales and Rating)
regressor_Rating = lm(formula = total_sales ~ Rating, data = train) # build
model
summary(regressor_Rating) # print model summary
plot(regressor_Rating, pch = 16, col = "red") # Plot the results
abline(regressor_Rating) # Add regression line

# evaluation
original = test$total_sales
pred = predict(regressor_Rating,test)
predicted = pred
d = original-predicted

mse = mean((d)^2) # MSE
mae = mean(abs(d)) # MAE
rmse = sqrt(mse) # RMSE
R2 = 1-(sum((d)^2)/sum((original-mean(original))^2)) # R^2

cat(" MAE:", mae, "\n", "MSE:", mse, "\n", "RMSE:", rmse, "\n", "R-squared:",
R2)
```