Project Name : Analyze the Sales Report of a Clothes Manufacturing Outlet

Library

library(readxl) library(plyr) library(e1071) library(rpart) library(randomForest) library(dplyr) library(tseries) library(ggplot2) library(imputeTS) library(forecast) library(taRifx)

Task 1:To automate the process of recommendations, the store needs to analyze the given

attributes of the product, like style, season, etc., and come up with a model to predict

the recommendation of products (in binary output- 0 or 1) accordingly.

Solution: We Can use Naive Bayes,SVM, Decision Tree and Random Forest

Path of dataset

setwd("C:/Users/User/Downloads/sda") getwd()

Loading Dataset

data<-read_excel("Attribute DataSet.xlsx")

Descriptive Analysis

View(data) str(data)

boxplot(data$Rating) data <- data[data$Rating>=4.4,]

data Cleaning

Handling Missing Values

Dataa<-na.omit(data) str(Dataa)

Data preparation

values checking

style

Dataa$Style[Dataa$Style=='sexy']<-'Sexy'

Price

Dataa$Price[Dataa$Price == 'low'] = 'Low' Dataa$Price[Dataa$Price == 'high'] = 'High'

Size

Dataa$Size[Dataa$Size == 's'] = 'S' Dataa$Size[Dataa$Size == 'small'] = 'S'

Season

Dataa$Season[Dataa$Season == 'spring'] = 'Spring' Dataa$Season[Dataa$Season == 'summer'] = 'Summer' Dataa$Season[Dataa$Season == 'Automn'] = 'Autumn' Dataa$Season[Dataa$Season == 'winter'] = 'Winter'

NeckLine

Dataa$NeckLine[Dataa$NeckLine == 'sweetheart'] = 'Sweetheart'

SleeveLength

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

FabricType

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

Decoration

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

Pattern Type

Dataa$'Pattern Type'[Dataa$'Pattern Type' == 'none'] = 'null' Dataa$'Pattern Type'[Dataa$'Pattern Type' == 'leapord'] = 'leopard'

Dataa$Recommendation<-sapply(Dataa$Recommendation,factor)

To decide the which model is suitable for Dataset I compare Naive bays, SVM, Decision Tree and Random Forest

1. Naive Bayes———71.49% Accuracy

naive_bayes<-naiveBayes(Recommendation~.,data = Dataa) summary(naive_bayes)

Predictions<-predict(naive_bayes,Dataa) Predictions table(Predictions,Dataa$Recommendation)

Accuracy of Model

table_mat<-table(Predictions,Dataa$Recommendation) accuracy_Test <- sum(diag(table_mat)) / sum(table_mat) accuracy_Test

2. Support Vector Machine——63.51% Accuracy

svms<-svm(Recommendation~.,data = Dataa,method = 'class') summary(svms)

Predictions<-predict(svms,Dataa,type='class') Predictions table(Predictions,Dataa$Recommendation)

Accuracy of Model

table_mat<-table(Predictions,Dataa$Recommendation) accuracy_Test <- sum(diag(table_mat)) / sum(table_mat) accuracy_Test

3. Decision Tree———-79.44% Accuracy

tree<-rpart(Recommendation~.,data = Dataa,method = 'class') summary(tree)

Predictions<-predict(tree,Dataa,type = 'class') Predictions table(Predictions,Dataa$Recommendation)

Accuracy of Model

table_mat<-table(Predictions,Dataa$Recommendation) accuracy_Test <- sum(diag(table_mat)) / sum(table_mat) accuracy_Test

4. Random Forest---------100% Accuracy

forest<-randomForest(x = Dataa, y = Dataa$Recommendation,ntree =800) # build model summary(forest)

Predictions<-predict(forest,Dataa) Predictions table(Predictions,Dataa$Recommendation)

Accuracy of Model

table_mat<-table(Predictions,Dataa$Recommendation) accuracy_Test <- sum(diag(table_mat)) / sum(table_mat) accuracy_Test

Task2: In order to stock the inventory, the store wants to analyze the sales data

and predict the trend of total sales for each dress for an extended period of

three more alternative days.

Solution: Time Series Analysis using ARIMA

Loading Dataset

sales <- read_excel("Dress Sales.xlsx")

str(sales)

Data Cleaning

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

Data Transformation

final_df <- as.data.frame(t(sales)) colnames(final_df)<-NULL suppressWarnings(final_df<-final_df[-1,])

final_df<-final_df[-1,]

suppressWarnings(final_df[is.na(final_df)] <- 0)

final_df[is.na(final_df)] <- 0

final_df

Basic line plot

plot(final_df[,1],col="red",pch =19)

Time series Analysis using ARIMA

vect<-c() j=0 for(i in c(1:500)){ suppressWarnings(d<-as.numeric(final_df[,i]))
na_mean(d) timeseries<-ts(d) fit <- auto.arima(timeseries) e<-forecast(fit, 3) vect=append(vect,sum(e$mean)) }

Add Predicted Result in Sales Data

result<-data.frame(vect) result sales$Predicted_Sum_Three_Day<-result View(sales)

Task3: To decide the pricing for various upcoming clothes, they wish to find how

the style, season, and material affect the sales of a dress and if the style of

the dress is more influential than its price.

Solution: Anova Analysis

Loading Dataset

Dataa<-read_excel("Attribute DataSet.xlsx") data<-read_excel("Dress Sales.xlsx")

Data Preparation

style

Dataa$Style[Dataa$Style=='sexy']<-'Sexy'

Price

Dataa$Price[Dataa$Price == 'low'] = 'Low' Dataa$Price[Dataa$Price == 'high'] = 'High'

Size

Dataa$Size[Dataa$Size == 's'] = 'S' Dataa$Size[Dataa$Size == 'small'] = 'S'

Season

Dataa$Season[Dataa$Season == 'spring'] = 'Spring' Dataa$Season[Dataa$Season == 'summer'] = 'Summer' Dataa$Season[Dataa$Season == 'Automn'] = 'Autumn' Dataa$Season[Dataa$Season == 'winter'] = 'Winter'

NeckLine

Dataa$NeckLine[Dataa$NeckLine == 'sweetheart'] = 'Sweetheart'

SleeveLength

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

FabricType

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

Decoration

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

Pattern Type

Dataa$'Pattern Type'[Dataa$'Pattern Type' == 'none'] = 'null' Dataa$'Pattern Type'[Dataa$'Pattern Type' == 'leapord'] = 'leopard'

Date Formating

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

data[is.na(data)]<-0 df<-data colnames(df)<-NULL df<-df[,-1]

df <- apply(df , 2 ,as.numeric)

suppressWarnings(df <- apply(df , 2 ,as.numeric)) data$Total_Sale <- rowSums(df) View(data)

Merge Both Dataset Data1 and Data

Sales<-merge(Dataa,data) Sales str(Sales)

Transform Data

Sales$Style<-recode(Sales$Style,'bohemian'=1,'Brief'=2,'Casual'=3,'cute'=4,'fashion'=5, 'Flare'=6,'Novelty'=7,'OL'=8,'party'=9,'Sexy'=10,'vintage'=11,'work'=12) Sales$Price<-recode(Sales$Price,'Average'=1,'High'=2,'Low'=3,'Medium'=4,'very-high'=5) Sales$Season<-recode(Sales$Season,'Autumn'=1,'Spring'=2,'Summer'=3,'Winter'=4) Sales$Material<-recode(Sales$Material,"null"=1,
"chiffonfabric"=2, "cotton" =3,"rayon" =4,
"silk"=5, "lace"=6, "viscos"=7, "polyster"=8,
"shiffon"=9, "milksilk"=10, "cashmere"=11, "mix"=12,
"microfiber"=13, "nylon"=14, "knitting"=15,"spandex" =16,
"acrylic"=17,"lycra"=18,"wool"=19,"modal"=20,
"model"=22,"other"=23,"linen"=24,
"sill"=25)

Anova Analysis of Sales to Style,Price,Season,Material

spsm<-aov(Total_Sale~Style+Price+Season+Material,data = Sales) spsm summary(spsm)

Result: The attribute Season is affect sales mostly. The Style and Material not much

significant to Sales. In the Style and Price, Style has significant value that the price.

Task4: Also, to increase the sales, the management wants to analyze the attributes

of dresses and find which are the leading factors affecting the sale of a

dress.

Solution: Anova Analysis

result<-
aov(Sales$Total_Sale~Sales$Style+Sales$Price+Sales$Rating+Sales$Size+Sales$Season+Sales$NeckLine+Sales$SleeveLength+Sales$waiseline+Sales$Material+Sales$FabricType+Sales$Decoration+Sales$ `Pattern Type` +Sales$Recommendation) result summary(result)

Result: The attributes Rating, Size, NeckLine,SleeveLength are most significant.

Attribute Season and Pattern are significant but not that much. Other Attributes

Style,Price, waistLine, Material, FabricType,Decoration and Recommendation is not significant.

Task5: To regularize the rating procedure and find its efficiency, the store wants to

find if the rating of the dress affects the total sales.

Solution: Linear regression

Build Regression Model

regression<-lm(Sales$Total_Sale~Sales$Rating) regression summary(regression) Predictions<-predict(regression,Sales) Predictions