resumotrabalho.R

ander

2020-02-29

setwd("c:/FCD/Hamburgueria209/")  
getwd()

## [1] "c:/FCD/Hamburgueria209"

#Pacote para ler graficos e corplot  
library(psych)  
library(corrplot)

## corrplot 0.84 loaded

library(gmodels)  
library(class)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages -------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v stringr 1.4.0  
## v tidyr 1.0.0 v forcats 0.4.0  
## v readr 1.3.1

## -- Conflicts ----------------------------------- tidyverse\_conflicts() --  
## x ggplot2::%+%() masks psych::%+%()  
## x ggplot2::alpha() masks psych::alpha()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(xlsx)  
library(pdftables)  
#write.xlsx(vend1, file = "c:/fcd/Hamburgueria209/predict.xlsx")  
  
vend1 <- read.xlsx('predict.xlsx', sheetIndex = T)  
  
# Analisando o dataset  
#View(vend1)  
dim(vend1)

## [1] 10532 19

str(vend1)

## 'data.frame': 10532 obs. of 19 variables:  
## $ NA. : Factor w/ 10532 levels "1","10","100",..: 1 1645 2756 3867 4978 6089 7200 8311 9422 2 ...  
## $ DataPedido : POSIXct, format: "2018-03-05 19:03:29" "2018-03-05 19:06:18" ...  
## $ Quantidade : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 2 1 1 1 ...  
## $ PrecoUnidade : num 17 22 3 15 1.5 3 22 17 1.5 15 ...  
## $ TotalProduto : num 12 16 22 9 3 22 30 12 3 9 ...  
## $ NomeProduto : Factor w/ 55 levels "\* ExcluÃƒÂ­do \* ÃƒÂ\201GUA MINERAL 350ML",..: 12 49 26 22 50 26 49 12 19 21 ...  
## $ TipoProduto : Factor w/ 2 levels "Complemento",..: 2 2 2 2 1 2 2 2 1 2 ...  
## $ Categoria : Factor w/ 5 levels "Cervejas","Complemento",..: 3 3 4 3 2 4 3 3 2 3 ...  
## $ ValorProduto : num 17 22 4 15 1.5 4 22 17 1.5 16 ...  
## $ DataAberturaPedido : POSIXct, format: "2018-03-05 19:02:22" "2018-03-05 19:06:00" ...  
## $ DataFechamentoPedido: POSIXct, format: "2018-03-05 19:58:54" "2018-03-05 19:59:16" ...  
## $ TipoVenda : Factor w/ 4 levels "BalcÃƒÂ£o ",..: 3 3 3 3 3 3 3 3 3 3 ...  
## $ StatusPedido : Factor w/ 2 levels "Finalizado Fiado",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ DiaSemana : Factor w/ 7 levels "domingo","quarta-feira",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ Hora : Factor w/ 6 levels "18","19","20",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Mes : Factor w/ 10 levels "abril","agosto",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ TempoAbert : num 2 6 6 9 9 9 31 42 42 42 ...  
## $ TempoFech : num 58 59 59 58 58 58 32 37 37 37 ...  
## $ TempoGasto : num -56 -53 -53 -49 -49 -49 -1 5 5 5 ...

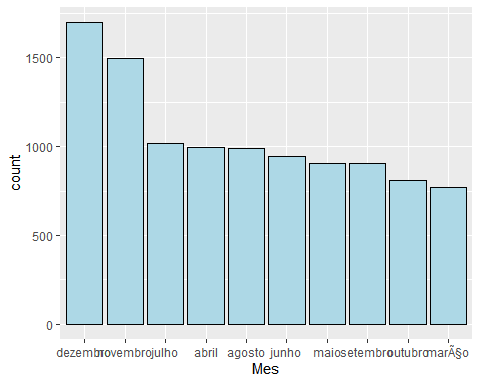
glimpse(vend1)

## Observations: 10,532  
## Variables: 19  
## $ NA. <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13...  
## $ DataPedido <dttm> 2018-03-05 19:03:29, 2018-03-05 19:06:18...  
## $ Quantidade <fct> 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1,...  
## $ PrecoUnidade <dbl> 17.0, 22.0, 3.0, 15.0, 1.5, 3.0, 22.0, 17...  
## $ TotalProduto <dbl> 12, 16, 22, 9, 3, 22, 30, 12, 3, 9, 7, 22...  
## $ NomeProduto <fct> BACON, PRIME, COCA COLA LATA, CLASSICO, Q...  
## $ TipoProduto <fct> Produto, Produto, Produto, Produto, Compl...  
## $ Categoria <fct> Hamburgueres, Hamburgueres, Refrigerantes...  
## $ ValorProduto <dbl> 17.0, 22.0, 4.0, 15.0, 1.5, 4.0, 22.0, 17...  
## $ DataAberturaPedido <dttm> 2018-03-05 19:02:22, 2018-03-05 19:06:00...  
## $ DataFechamentoPedido <dttm> 2018-03-05 19:58:54, 2018-03-05 19:59:16...  
## $ TipoVenda <fct> Delivery , Delivery , Delivery ,...  
## $ StatusPedido <fct> Finalizado Pago , Finalizado Pago , Final...  
## $ DiaSemana <fct> segunda-feira, segunda-feira, segunda-fei...  
## $ Hora <fct> 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 1...  
## $ Mes <fct> marÃ§o, marÃ§o, marÃ§o, marÃ§o, marÃ§o, m...  
## $ TempoAbert <dbl> 2, 6, 6, 9, 9, 9, 31, 42, 42, 42, 45, 45,...  
## $ TempoFech <dbl> 58, 59, 59, 58, 58, 58, 32, 37, 37, 37, 3...  
## $ TempoGasto <dbl> -56, -53, -53, -49, -49, -49, -1, 5, 5, 5...

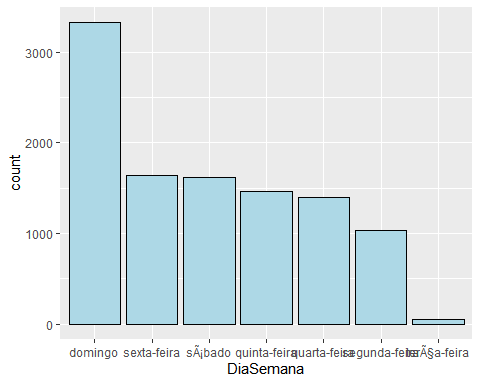
# Medias de tendencia  
summary(vend1)

## NA. DataPedido Quantidade PrecoUnidade   
## 1 : 1 Min. :2018-03-05 19:03:29 1:9782 Min. : 1.00   
## 10 : 1 1st Qu.:2018-05-29 20:13:53 2: 588 1st Qu.: 5.00   
## 100 : 1 Median :2018-08-19 19:35:08 3: 136 Median :15.00   
## 1000 : 1 Mean :2018-08-17 18:11:42 4: 20 Mean :13.24   
## 10000 : 1 3rd Qu.:2018-11-15 20:18:10 5: 5 3rd Qu.:20.00   
## 10001 : 1 Max. :2018-12-30 22:47:23 9: 1 Max. :30.00   
## (Other):10526   
## TotalProduto NomeProduto TipoProduto Categoria   
## Min. : 2.00 BACON :1662 Complemento: 838 Cervejas : 182   
## 1st Qu.:12.00 MAXX :1619 Produto :9694 Complemento : 838   
## Median :15.00 PRIME :1100 Hamburgueres :7191   
## Mean :17.88 GAROTO : 708 Refrigerantes:2179   
## 3rd Qu.:22.00 CLASSICO: 671 Sucos : 142   
## Max. :47.00 CHICKEN : 523   
## (Other) :4249   
## ValorProduto DataAberturaPedido   
## Min. : 1.00 Min. :2018-03-05 19:02:22   
## 1st Qu.: 6.00 1st Qu.:2018-05-29 20:13:27   
## Median :16.00 Median :2018-08-19 19:30:40   
## Mean :13.48 Mean :2018-08-17 18:08:22   
## 3rd Qu.:20.00 3rd Qu.:2018-11-15 20:14:53   
## Max. :22.00 Max. :2018-12-30 22:34:25   
##   
## DataFechamentoPedido TipoVenda   
## Min. :2018-03-05 19:58:35 BalcÃƒÂ£o : 80   
## 1st Qu.:2018-05-29 20:43:48 Caixa : 8   
## Median :2018-08-19 20:11:50 Delivery :8177   
## Mean :2018-08-17 18:57:35 Mesa/Comanda :2267   
## 3rd Qu.:2018-11-15 20:54:38   
## Max. :2018-12-30 23:12:11   
##   
## StatusPedido DiaSemana Hora Mes   
## Finalizado Fiado: 32 domingo :3326 18: 186 dezembro:1699   
## Finalizado Pago :10500 quarta-feira :1401 19:2996 novembro:1498   
## quinta-feira :1464 20:3492 julho :1017   
## sÃ¡bado :1619 21:2786 abril : 994   
## segunda-feira:1031 22:1046 agosto : 990   
## sexta-feira :1642 23: 26 junho : 943   
## terÃ§a-feira : 49 (Other) :3391   
## TempoAbert TempoFech TempoGasto   
## Min. : 0.00 Min. : 0.00 Min. :-59.00   
## 1st Qu.:14.00 1st Qu.:15.00 1st Qu.:-24.00   
## Median :28.00 Median :30.00 Median : 1.50   
## Mean :28.48 Mean :30.13 Mean : -1.65   
## 3rd Qu.:43.00 3rd Qu.:45.00 3rd Qu.: 19.00   
## Max. :59.00 Max. :59.00 Max. : 59.00   
##

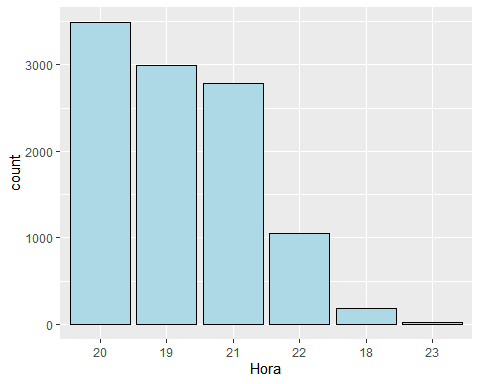
# Transformancao dos dados  
#str(vend1)  
#str(vend1$Qtd.)  
#vend1$Qtd. <- as.numeric(vend1$Qtd.)  
#class(vend1$Qtd.)  
  
# Ajustando os tipos de variaveis  
#vend1$Valor..Tot..Item <- as.numeric(vend1$Valor..Tot..Item)  
#vend1$Tipo.de.Item <- as.numeric(vend1$Tipo.de.Item)  
#vend1$Cod..Ped. <- as.character(vend1$Cod..Ped.)  
#vend1$NÃÂºm..Mesa.Com. = NULL  
#vend1$TipoItem = NULL  
#vend1$CodigoMesa = NULL  
  
#table(vend1$Stat..Ped.)  
  
#Verificando os valores "NA" e nulos  
#any(is.na(vend1))  
#sapply(vend1, function(x) sum(is.na(x)))  
#sapply(vend1, function(x) sum(anyNA(x)))  
  
#removendo valores "NA"(Nulos) do dataset inteiro  
#vend1 <- na.omit(vend1)  
  
# Extraindo o meses, dias da semana para insigths  
  
#vend1$Mes <- months.Date(vend1$Data.Hora.Item)  
#vend1$DiaSemana <- weekdays.Date(vend1$Data.Hora.Item)  
#vend1$Hora <- hour(vend1$Data.Hora.Item)  
  
#table(vend1$Mes)  
#table(vend1$Hora)  
#table(vend1$DiaSemana)  
  
# Copia do conjunto de dados  
#vend2 <- vend1  
  
# Alterando o nome das colunas do nosso dataset  
#str(hamb)  
#names(vend1) <- c("DataPedido", "Quantidade", "PrecoUnidade", "TotalProduto","TipoItem",  
# "NomeProduto", "TipoProduto", "Categoria", "ValorProduto",  
# "CodigoMesa", "DataAberturaPedido", "DataFechamentoPedido",  
# "TipoVenda", "StatusPedido", "DiaSemana", "Hora", "Mes")  
  
# Extraindo os minutos para saber quanto tempo um pedido Ã© feito  
  
#vend1$TempoAbert <- minute(vend1$DataAberturaPedido)  
#vend1$TempoFech <- minute(vend1$DataFechamentoPedido)  
#hamb$TempoGasto = (hamb$TempoAbert - (hamb$TempoFech))  
  
  
##############################################################################  
# Graficos de AnÃ¡lises  
  
#Naturalmente a ordem das barras Ã© controlada atravÃ©s de fatores,   
#que sÃ£o uma estrutura de dados utilizada para representar categorias -   
#que Ã© diferente do caractere  
  
  
# Grafico de vendas por mÃªs  
vend1$Mes = factor(vend1$Mes, levels = names(sort(table(vend1$Mes), decreasing = TRUE)))  
ggplot(vend1, aes(x=Mes)) +  
 geom\_bar(color = "black", fill = "light blue")



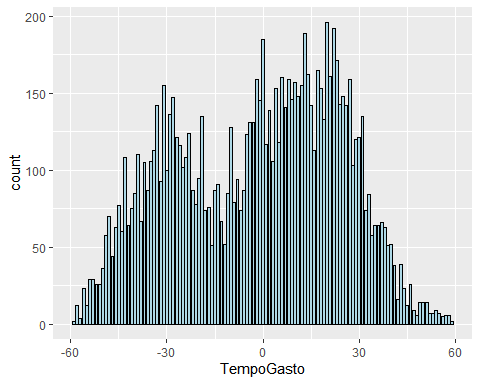
# grafico de vendas por dia da semana  
vend1$DiaSemana = factor(vend1$DiaSemana, levels = names(sort(table(vend1$DiaSemana), decreasing = TRUE)))  
ggplot(vend1, aes(x=DiaSemana))+  
 geom\_bar(color = "black", fill = "light blue")



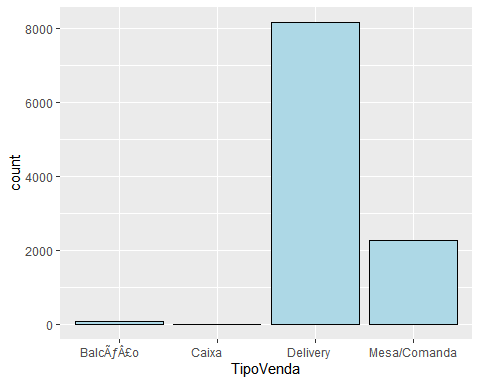
# Grafico de vendas por horario  
vend1$Hora = factor(vend1$Hora, levels = names(sort(table(vend1$Hora), decreasing = TRUE)))  
ggplot(vend1, aes(x=Hora)) +  
 geom\_bar(color = "black", fill = "light blue")



# Grafico de tempo gasto por Hamburgueres apenas  
  
ggplot(vend1, aes(x=TempoGasto)) +  
 geom\_bar(color = "black", fill = "light blue", na.rm = FALSE)



# Grafico por tipo de Venda  
ggplot(vend1, aes(x=TipoVenda)) +  
 geom\_bar(color = "black", fill = "light blue", na.rm = FALSE)



###############################################################################  
  
  
# Treinando o modelo  
library(caTools)  
  
# Criando dados de treino e dados de teste  
amostra <- sample.split(vend1, SplitRatio = 0.70)  
  
dados\_treino <- subset(vend1, amostra == TRUE)  
dados\_teste <- subset(vend1, amostra == FALSE)  
  
# Criando modelo de treino  
modelov1 <- lm(PrecoUnidade ~ Categoria, dados\_treino)  
  
# Sumario do modelo  
summary(modelov1)

##   
## Call:  
## lm(formula = PrecoUnidade ~ Categoria, data = dados\_treino)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12.4726 -1.7403 -0.1308 2.5274 12.5274   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.0000 0.2973 20.179 < 2e-16 \*\*\*  
## CategoriaComplemento -4.3692 0.3278 -13.329 < 2e-16 \*\*\*  
## CategoriaHamburgueres 11.4726 0.3010 38.119 < 2e-16 \*\*\*  
## CategoriaRefrigerantes -1.2597 0.3092 -4.074 4.66e-05 \*\*\*  
## CategoriaSucos -0.8333 0.4470 -1.864 0.0624 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.271 on 7201 degrees of freedom  
## Multiple R-squared: 0.7869, Adjusted R-squared: 0.7868   
## F-statistic: 6647 on 4 and 7201 DF, p-value: < 2.2e-16

# Prevendo com dados de testes  
previsao <- predict(modelov1, dados\_teste)  
View(previsao)  
  
  
  
resultado <- cbind(dados\_teste$PrecoUnidade, previsao)  
colnames(resultado) <- c("PrecoAtual", "PrecoPrevisto")  
resultado <- as.data.frame(resultado)  
View(resultado)  
  
  
# RandomForest  
  
require(randomForest)

## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

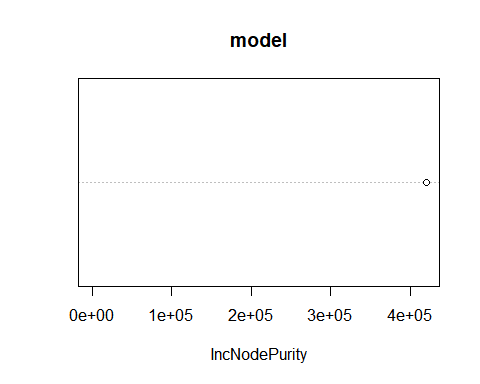
## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:psych':  
##   
## outlier

model <- randomForest(PrecoUnidade ~ Categoria,   
 data = vend1,   
 ntree = 40,   
 nodesize = 5)  
print(model)

##   
## Call:  
## randomForest(formula = PrecoUnidade ~ Categoria, data = vend1, ntree = 40, nodesize = 5)   
## Type of random forest: regression  
## Number of trees: 40  
## No. of variables tried at each split: 1  
##   
## Mean of squared residuals: 10.61682  
## % Var explained: 78.97

scores <- data.frame(actual = vend1$NomeProduto,vend1$PrecoUnidade,  
 prediction = predict(model, newdata = vend1))  
  
View(scores)  
varImpPlot(model)



# Gravando o resultado  
df\_saida <- vend1[, c("PrecoUnidade", rownames(model$importance))]  
  
View(df\_saida)  
  
  
  
################################################################################