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

getwd()

vend1 <- read.xlsx('Historico\_Itens\_Vendidos de 01-01-18 à 31-12-18.xls', sheetIndex = T)

# Analisando o dataset

View(vend1)

dim(vend1)

> dim(vend1)

[1] 10532 18

str(vend1)

> str(vend1)

'data.frame': 10532 obs. of 18 variables:

$ DataPedido : POSIXct, format: "2018-03-05 19:03:29" ...

$ Quantidade : Factor w/ 7 levels "","1","2","3",..: 2 2 2 2 2 2 3 2 2 2 ...

$ 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 "\* ExcluÃ­do \* Ã\201GUA MINERAL 350ML",..: 10 16 32 12 52 32 16 10 27 11 ...

$ 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" ...

$ DataFechamentoPedido: POSIXct, format: "2018-03-05 19:58:54" ...

$ TipoVenda : Factor w/ 4 levels "BalcÃ£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","sexta-feira",..: 6 6 6 6 6 6 6 6 6 6 ...

$ Hora : Factor w/ 6 levels "20","19","21",..: 2 2 2 2 2 2 2 2 2 2 ...

$ Mes : Factor w/ 10 levels "dezembro","novembro",..: 10 10 10 10 10 10 10 10 10 10 ...

$ TempoAbert : int 2 6 6 9 9 9 31 42 42 42 ...

$ TempoFech : int 58 59 59 58 58 58 32 37 37 37 ...

$ TempoGasto : int -56 -53 -53 -49 -49 -49 -1 5 5 5 ...

- attr(\*, "na.action")= 'omit' Named int 10533

..- attr(\*, "names")= chr "10533"

glimpse(vend1)

|  |
| --- |
| > glimpse(vend1)  Observations: 10,532  Variables: 18  $ DataPedido *<dttm>* 2018-03-05 19:03:29, 2018-03-05 19:...  $ Quantidade *<fct>* 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, ...  $ PrecoUnidade *<dbl>* 17.0, 22.0, 3.0, 15.0, 1.5, 3.0, 22....  $ TotalProduto *<dbl>* 12, 16, 22, 9, 3, 22, 30, 12, 3, 9, ...  $ NomeProduto *<fct>* BACON, PRIME, COCA COLA LATA, CLASSI...  $ TipoProduto *<fct>* Produto, Produto, Produto, Produto, ...  $ Categoria *<fct>* Hamburgueres, Hamburgueres, Refriger...  $ ValorProduto *<dbl>* 17.0, 22.0, 4.0, 15.0, 1.5, 4.0, 22....  $ DataAberturaPedido *<dttm>* 2018-03-05 19:02:22, 2018-03-05 19:...  $ DataFechamentoPedido *<dttm>* 2018-03-05 19:58:54, 2018-03-05 19:...  $ TipoVenda *<fct>* Delivery , Delivery , Delivery...  $ StatusPedido *<fct>* Finalizado Pago , Finalizado Pago , ...  $ DiaSemana *<fct>* segunda-feira, segunda-feira, segund...  $ Hora *<fct>* 19, 19, 19, 19, 19, 19, 19, 19, 19, ...  $ Mes *<fct>* março, março, março, março, março, m...  $ TempoAbert *<int>* 2, 6, 6, 9, 9, 9, 31, 42, 42, 42, 45...  $ TempoFech *<int>* 58, 59, 59, 58, 58, 58, 32, 37, 37, ...  $ TempoGasto *<int>* -56, -53, -53, -49, -49, -49, -1, 5,... |
|  |
| |  | | --- | | > | |

# Medias de tendencia

summary(vend1)

> summary(vend1)

DataPedido Quantidade PrecoUnidade TotalProduto NomeProduto TipoProduto Categoria

Min. :2018-03-05 19:03:29 : 0 Min. : 1.00 Min. : 2.00 BACON :1662 Complemento: 838 Cervejas : 182

1st Qu.:2018-05-29 20:13:53 1:9782 1st Qu.: 5.00 1st Qu.:12.00 MAXX :1619 Produto :9694 Complemento : 838

Median :2018-08-19 19:35:08 2: 588 Median :15.00 Median :15.00 PRIME :1100 Hamburgueres :7191

Mean :2018-08-17 18:11:42 3: 136 Mean :13.24 Mean :17.88 GAROTO : 708 Refrigerantes:2179

3rd Qu.:2018-11-15 20:18:10 4: 20 3rd Qu.:20.00 3rd Qu.:22.00 CLASSICO: 671 Sucos : 142

Max. :2018-12-30 22:47:23 5: 5 Max. :30.00 Max. :47.00 CHICKEN : 523

9: 1 (Other) :4249

ValorProduto DataAberturaPedido DataFechamentoPedido TipoVenda StatusPedido

Min. : 1.00 Min. :2018-03-05 19:02:22 Min. :2018-03-05 19:58:35 BalcÃ£o : 80 Finalizado Fiado: 32

1st Qu.: 6.00 1st Qu.:2018-05-29 20:13:27 1st Qu.:2018-05-29 20:43:48 Caixa : 8 Finalizado Pago :10500

Median :16.00 Median :2018-08-19 19:30:40 Median :2018-08-19 20:11:50 Delivery :8177

Mean :13.48 Mean :2018-08-17 18:08:22 Mean :2018-08-17 18:57:35 Mesa/Comanda :2267

3rd Qu.:20.00 3rd Qu.:2018-11-15 20:14:53 3rd Qu.:2018-11-15 20:54:38

Max. :22.00 Max. :2018-12-30 22:34:25 Max. :2018-12-30 23:12:11

DiaSemana Hora Mes TempoAbert TempoFech TempoGasto

domingo :3326 20:3492 dezembro:1699 Min. : 0.00 Min. : 0.00 Min. :-59.00

sexta-feira :1642 19:2996 novembro:1498 1st Qu.:14.00 1st Qu.:15.00 1st Qu.:-24.00

sábado :1619 21:2786 julho :1017 Median :28.00 Median :30.00 Median : 1.50

quinta-feira :1464 22:1046 abril : 994 Mean :28.48 Mean :30.13 Mean : -1.65

quarta-feira :1401 18: 186 agosto : 990 3rd Qu.:43.00 3rd Qu.:45.00 3rd Qu.: 19.00

segunda-feira:1031 23: 26 junho : 943 Max. :59.00 Max. :59.00 Max. : 59.00

terça-feira : 49 (Other) :3391

#Pacotes necessários

library(psych)

library(corrplot)

library(gmodels)

library(class)

library(dplyr)

library(tidyverse)

# 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$NÃºm..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)

# 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 é feito

vend1$TempoAbert <- minute(vend1$DataAberturaPedido)

vend1$TempoFech <- minute(vend1$DataFechamentoPedido)

hamb$TempoGasto = (hamb$TempoAbert - (hamb$TempoFech))

##############################################################################

# Graficos de Análises

#Naturalmente a ordem das barras é controlada através de fatores,

#que são uma estrutura de dados utilizada para representar categorias -

#que é diferente do caractere

# Grafico de vendas por mês

vend1$Mes = factor(vend1$Mes, levels = names(sort(table(vend1$Mes), decreasing = TRUE)))

ggplot(hamb, aes(x=Mes)) +

geom\_bar(color = "black", fill = "light blue")

Uma imagem contendo captura de tela

Descrição gerada automaticamente

# grafico de vendas por dia da semana

vend1$DiaSemana = factor(vend1$DiaSemana, levels = names(sort(table(vend1$DiaSemana), decreasing = TRUE)))

ggplot(hamb, aes(x=DiaSemana))+

geom\_bar(color = "black", fill = "light blue")

Uma imagem contendo captura de tela

Descrição gerada automaticamente

# Grafico de vendas por horario

vend1$Hora = factor(vend1$Hora, levels = names(sort(table(vend1$Hora), decreasing = TRUE)))

ggplot(hamb, aes(x=Hora)) +

geom\_bar(color = "black", fill = "light blue")

Uma imagem contendo captura de tela

Descrição gerada automaticamente

# Grafico por tipo de Venda

ggplot(vend1, aes(x=TipoVenda)) +

geom\_bar(color = "black", fill = "light blue", na.rm = FALSE)

Uma imagem contendo captura de tela

Descrição gerada automaticamente

# Grafico de tempo gasto por Hamburgueres apenas

ggplot(hamb, aes(x=TempoGasto)) +

geom\_bar(color = "black", fill = "light blue", na.rm = FALSE)

Uma imagem contendo música

Descrição gerada automaticamente

######################################################################

# 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)

> summary(modelov1)

Call:

lm(formula = PrecoUnidade ~ Categoria, data = dados\_treino)

Residuals:

Min 1Q Median 3Q Max

-12.4560 -1.7267 -0.1158 2.5440 12.5440

Coefficients:

Estimate Std. Error t value

(Intercept) 6.0360 0.2940 20.531

CategoriaComplemento -4.4202 0.3260 -13.558

CategoriaHamburgueres 11.4200 0.2978 38.347

CategoriaRefrigerantes -1.3093 0.3062 -4.276

CategoriaSucos -0.9249 0.4544 -2.035

Pr(>|t|)

(Intercept) < 2e-16 \*\*\*

CategoriaComplemento < 2e-16 \*\*\*

CategoriaHamburgueres < 2e-16 \*\*\*

CategoriaRefrigerantes 1.92e-05 \*\*\*

CategoriaSucos 0.0418 \*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.287 on 7017 degrees of freedom

Multiple R-squared: 0.7858, Adjusted R-squared: 0.7857

F-statistic: 6435 on 4 and 7017 DF, p-value: < 2.2e-16

# RandomForest

require(randomForest)

odel <- randomForest(PrecoUnidade ~ Categoria,

data = vend1,

ntree = 40,

nodesize = 5)

print(model)

> summary(modelov1)

Call:

lm(formula = PrecoUnidade ~ Categoria, data = dados\_treino)

Residuals:

Min 1Q Median 3Q Max

-12.4560 -1.7267 -0.1158 2.5440 12.5440

Coefficients:

Estimate Std. Error t value

(Intercept) 6.0360 0.2940 20.531

CategoriaComplemento -4.4202 0.3260 -13.558

CategoriaHamburgueres 11.4200 0.2978 38.347

CategoriaRefrigerantes -1.3093 0.3062 -4.276

CategoriaSucos -0.9249 0.4544 -2.035

Pr(>|t|)

(Intercept) < 2e-16 \*\*\*

CategoriaComplemento < 2e-16 \*\*\*

CategoriaHamburgueres < 2e-16 \*\*\*

CategoriaRefrigerantes 1.92e-05 \*\*\*

CategoriaSucos 0.0418 \*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.287 on 7017 degrees of freedom

Multiple R-squared: 0.7858, Adjusted R-squared: 0.7857

F-statistic: 6435 on 4 and 7017 DF, p-value: < 2.2e-16

> View(previsao)

> 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: 10.61758

% Var explained: 78.97

scores <- data.frame(actual = vend1$NomeProduto,vend1$PrecoUnidade,

prediction = predict(model, newdata = vend1))

View(scores)

Produto Preço Atual Preço Previsto

|  |  |  |  |
| --- | --- | --- | --- |
| **189** | BACON | 17.0 | 17.508705 |
| **190** | GAROTO | 12.0 | 17.508705 |
| **191** | CHICKEN | 15.0 | 17.508705 |
| **192** | MAXX | 20.0 | 17.508705 |
| **193** | COCA COLA 2 L | 8.0 | 4.744215 |
| **194** | BACON | 17.0 | 17.508705 |
| **195** | MAXX | 20.0 | 17.508705 |
| **196** | BACON | 17.0 | 17.508705 |
| **197** | GUARANÃ ANTÃRTICA LATA | 3.0 | 4.744215 |