Projeto02

Fábio Teixeira Trindade

15 de dezembro de 2018

Descrição do Projeto

O Grupo Bimbo, se esforça para atender a demanda diária dos consumidores por produtos frescos de panificação nas prateleiras de mais de 1 milhão de lojas ao longo das suas 45.000 lojas em todo o México.

Atualmente, os cálculos diários de estoque são realizados por funcionários de vendas de entregas diretas, que devem, sozinhos, prever a necessidade de estoque dos produtos e demanda com base em suas experiências pessoais em cada loja. Como alguns pães têm uma vida útil de uma semana, a margem aceitável para o erro é pequena.

Neste projeto de aprendizado de máquina, você deve desenvolver um modelo para prever com precisão a demanda de estoque com base nos dados históricos de vendas. Isso fará com que os consumidores dos mais de 100 produtos de panificação não fiquem olhando para as prateleiras vazias, além de reduzir o valor gasto com reembolsos para os proprietários de lojas com produtos excedentes impróprios para venda.

Deve-se, aqui, prever a demanda de um produto para uma determinada semana, em uma determinada loja. O conjunto de dados consiste em 9 semanas de transações de vendas no México. Toda semana, há caminhões de entrega que entregam produtos para os fornecedores. Cada transação consiste em vendas e devoluções. Devoluções são os produtos não vendidos e expirados. A demanda por um produto em uma determinada semana é definida como as vendas desta semana subtraídas pelo retorno na próxima semana.

Os arquivos de dados de treino e de teste contêm os seguintes campos:

1. **Semana**: Número da samana
2. **Agencia_ID**: ID Agência
3. **Canal_ID**: ID Canal de Vendas
4. **Ruta_SAK**: ID Rota (Várias Rotas = Agência de Vendas)
5. **Cliente_ID**: ID Cliente
6. **NombreCliente**: Nome do Cliente
7. **Producto_ID**: ID Produto
8. **NombreProducto**: Nome do Produto
9. **Venta_uni_hoy**: Unidade de vendas (inteiro)
10. **Venta_hoy**: Vendas (unidade: pesos)

11. **Dev_uni_proxima**: Retorno até a semana seguinte (inteiro)
12. **Dev_proxima**: Retorno na próxima semana (unidade: pesos)
13. **Demanda_uni_equil**: Demanda Ajustada (inteiro) (Essa é a variável target para previsão)

No final, deve ser gerado um arquivo de resultado da previsão contendo, em cada linha, duas colunas: **id** e **Demanda_uni_equi** (variável target a ser prevista). O **id** corresponde à coluna **id** no **test.csv**. O arquivo deve conter também um cabeçalho e ter o seguinte formato:

id,Demanda_uni_equil

0,1

1,0

2.500

3.100

etc.

Carregamento e Preparação dos Dados

Durante a carga e preparação dos dados vamos eliminar as linhas contendo 'NA' dos dados de treino e calcular a quantidade de linhas de treino para ser usada nas camadas dos gráficos.

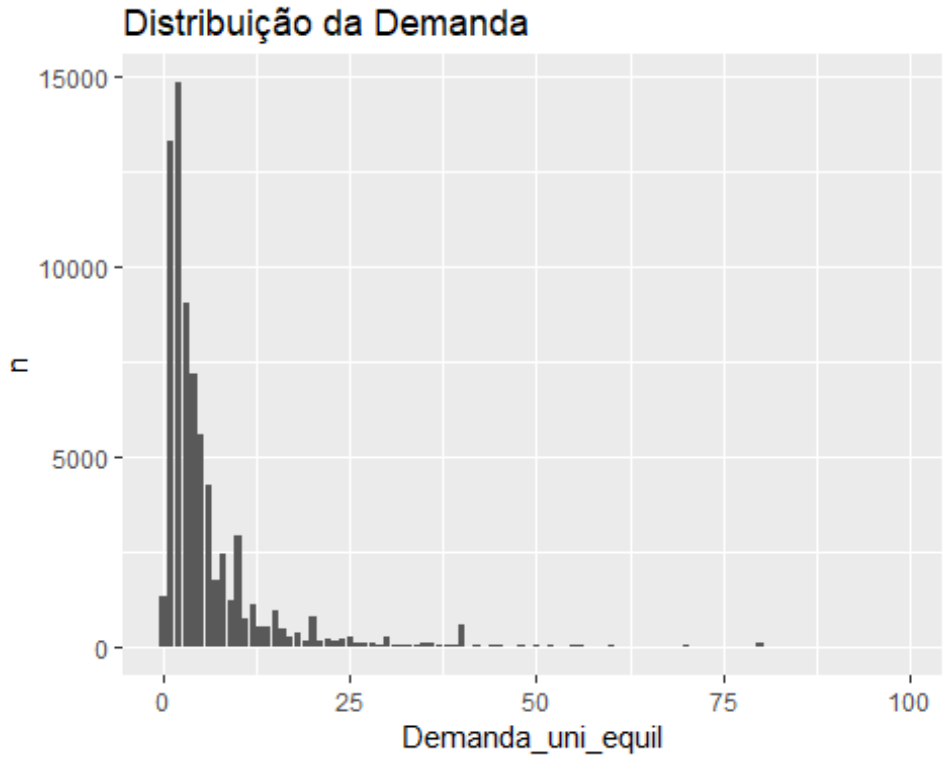
```
library(readr)
library(dplyr)
library(ggplot2)
library(dtplyr)
library(xgboost)

train = read_csv("train.csv",col_names=TRUE)
train <- na.omit(train)
n = nrow(train)
head(train)
```

```
## # A tibble: 6 x 11
##   Semana Agencia_ID Canal_ID Ruta_SAK Cliente_ID Producto_ID
Venta_uni_hoy
##    <int>      <int>    <int>    <int>      <int>       <int>
<int>
## 1      4      22090        1     1203    2151334        1212
1
## 2      9       2014        1     1229    2098006        1232
1
## 3      6       1215        1     1020    2174651        2233
2
## 4      3       1423        1     1230    2479203       43206
```

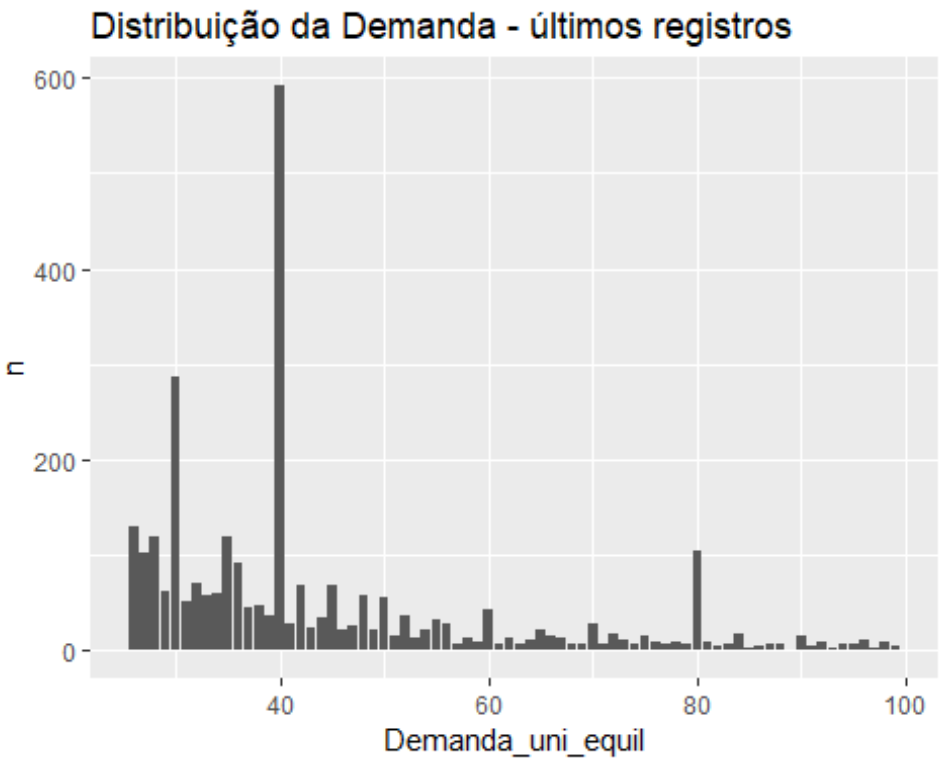
```
6
## 5      8       1335        4     6607    2312960        1230
1
## 6      9       1420        1     5505    1200523       33736
0
## # ... with 4 more variables: Venta_hoy <dbl>, Dev_uni_proxima <int>,
## #   Dev_proxima <dbl>, Demanda_uni_equil <int>
```

Vamos explorar um pouco a variável Demand, já que existem alguns outliers, vamos apenas olhar para a Demand menos de 100.

```
train %>%
    count(Demanda_uni_equil) %>%
    filter(Demanda_uni_equil < 100) %>%
    ggplot(aes(x = Demanda_uni_equil, y = n)) +
        geom_bar(stat = "identity") +
        ggtitle("Distribuição da Demanda")
```




```
train %>%
    count(Demanda_uni_equil) %>%
    filter(Demanda_uni_equil > 25, Demanda_uni_equil < 100) %>%
    ggplot(aes(x = Demanda_uni_equil, y = n)) +
        geom_bar(stat = "identity") +
        ggtitle("Distribuição da Demanda - últimos registros")
```



Observa-se que há certos valores “redondos” - 40, 60, 80. São muito mais comuns como demandas.

Um exame nos atributos(features):

Vamos dar uma olhada mais sobre a relação entre atributos e demanda. Todos os atributos são categóricos e muitos deles tem milhares de níveis assim representando alguns desafios.

Producto_ID:

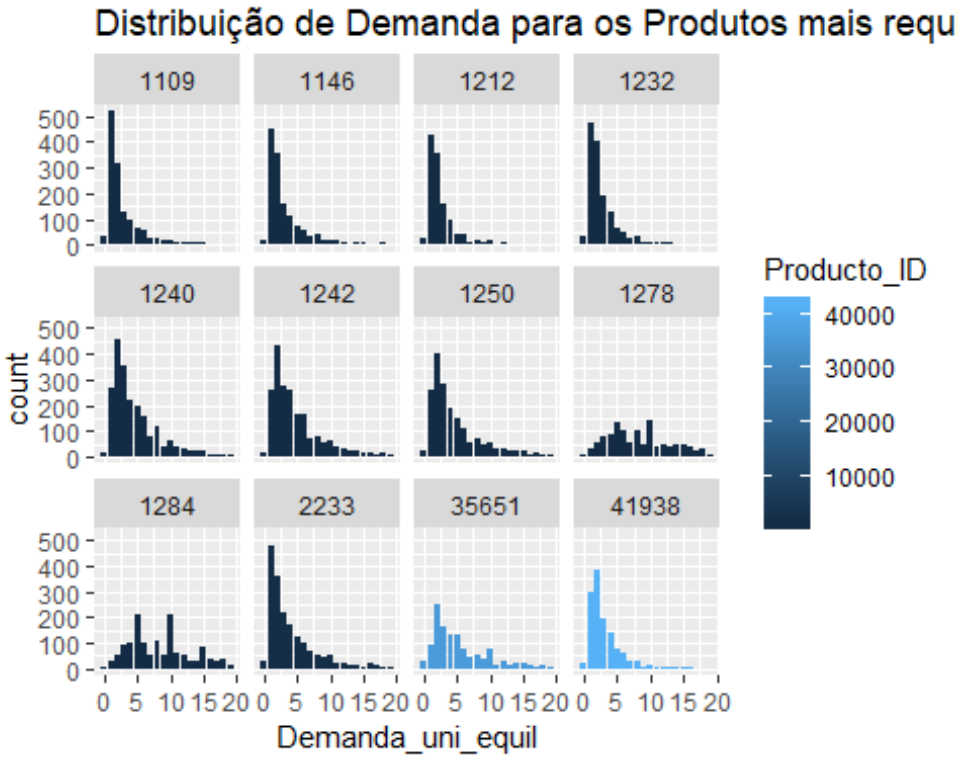
Temos um total de 891 Product ID’s: muita coisa para se ver de uma vez só.

```
train %>% count(Producto_ID, sort = TRUE) -> product_count
top_products = product_count$Producto_ID[1:12]
```

Isso parece legal, mas não é muito útil, pois as estimativas ficam confusas, já que a demanda é dada em números inteiros. Vamos, em vez disso, examinar os histogramas de demanda por produto:

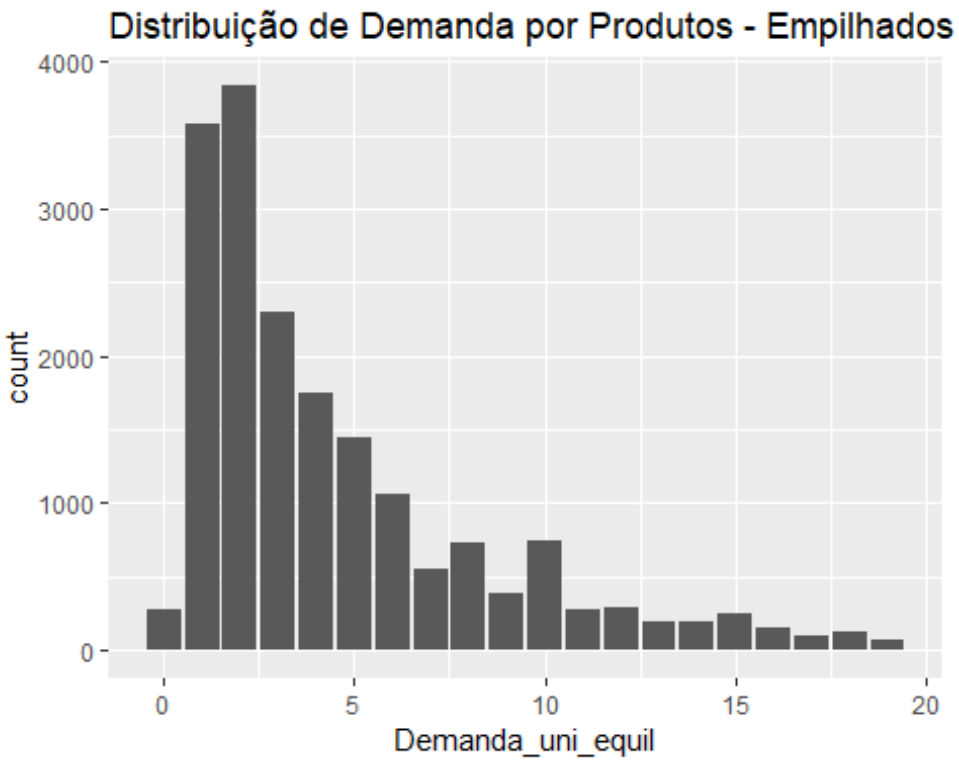
```
train %>%
    filter(Producto_ID %in% top_products) %>%
    filter(Demanda_uni_equil < 20) %>%
    ggplot(aes(x = Demanda_uni_equil, fill = Producto_ID)) +
    geom_bar() +
```

```
    facet_wrap( ~ Producto_ID) +
    ggtitle("Distribuição de Demanda para os Produtos mais requisitados")
```



Examinando o gráfico parece que existem aproximadamente dois tipos diferentes de distribuições. Também podemos empilhar essas distribuições umas em cima das outras para uma perspectiva diferente:

```
train %>%
    filter(Producto_ID %in% top_products) %>%
    filter(Demanda_uni_equil < 20) %>%
    ggplot(aes(x = Demanda_uni_equil, fill = Producto_ID)) +
    geom_bar(position = "dodge") +
    ggtitle("Distribuição de Demanda por Produtos - Empilhados e
Normalizados")
```



Previsão da Damanda_uni_equil

Carregamento dos dados de treino e teste. Carregou-se os dados novamente com 'fread' para selecionar apenas alguns campos necessários à previsão:

```
library(data.table)
train = fread('train.csv',
              select = c('Semana','Producto_ID', 'Cliente_ID',
'Demanda_uni_equil'))
test = fread('test.csv', select = c("Semana",'id','Cliente_ID',
'Producto_ID'))
train = train[Semana > 3]
train$Demanda_uni_equil = log1p(train$Demanda_uni_equil)
```

Neste ponto faz-se uma mescla dos dados de teste + treino:

```
train$id = 0; test$Demanda_uni_equil = 0; train$tst = 0; test$tst = 1
rec_train = rbind(train[Semana == 9], test)
```

Agora processa a média (demanda + contagem) nos dados de treino nas semanas de 3 - 8, depois junta com as semanas 9, 10 ,11:

```
train[Semana <= 8][, .(mean_client_prod = mean(Demanda_uni_equil),
                       count_client_prod = .N),
                   by = .(Producto_ID, Cliente_ID)] %>%
  merge(rec_train, all.y = TRUE, by = c("Producto_ID", "Cliente_ID")) ->
```

```
rec_train


train[Semana <= 8][, .(mean_prod = mean(Demanda_uni_equil),
                       count_prod = .N),
                   by = .(Producto_ID)] %>%
  merge(rec_train, all.y = TRUE, by = c("Producto_ID")) -> rec_train


train[Semana <= 8][, .(mean_cliente = mean(Demanda_uni_equil),
                       count_cliente = .N),
                   by = .(Cliente_ID)] %>%
  merge(rec_train, all.y = TRUE, by = c("Cliente_ID")) -> rec_train
```

Aplica-se o algoritmo XGBoosting nos dados de treino na semana 9, fazendo as previsões para as semanas 10 e 11 e gravando-as no arquivo de resultado:

```
y_train = rec_train$Demanda_uni_equil[rec_train$tst == 0]

dtrain = xgb.DMatrix(as.matrix(rec_train[tst == 0] %>% select(-
Demanda_uni_equil, -id, - tst)),
                     label = y_train)
model = xgb.train(data = dtrain, nrounds = 25, max_depth = 8, eta = 0.5)

preds = predict(model, as.matrix(rec_train[tst == 1] %>% select(-
Demanda_uni_equil, -id, - tst)))
preds = expm1(preds)
preds[preds < 0] = 0
id = rec_train[tst == 1]$id

solution = data.frame(id = as.integer(id), Demanda_uni_equil = preds)
write.csv(solution, "xgboost_mean_resultado_previsao.csv", row.names =
FALSE)
```

Este trabalho foi um esforço de revisão dos capítulos do curso de R, principalmente o capítulo 8, e a composição de alguns trabalhos disponíveis no site do kaggle. Percebe-se agora a dimensão da aplicabilidade dos conceitos vistos no curso com os trabalhos do mundo profissional.