Modelos de inferência

Previsão de vendas semanais em
lojas Walmart estadunidenses

Participantes:
Andressa Silva de Oliveira
Camila Buzin Ladeira
Luiz Ricardo Hardman Paranhos
Matheus Kwon

Orientadores:
Maria Kelly Venezuela
Barbara Agena

11/2020

Sumário

1. Introdução

1.1 Objetivo

A habilidade de prever dados de forma exata é extremamente valiosa em uma vasta gama de domínios como saúde, vendas, finanças, clima ou esportes (GIL, 2019). No contexto em questão, a previsão das vendas em lojas Walmart é uma técnica de gestão fundamental, a qual garante a qualidade da produção e evita perdas com estoque, logística e compras, além de servir como base para a tomada de decisões estratégicas. Dessa forma, é fundamental que tais previsões apresentem um alto nível de acurácia, uma vez que interferem diretamente nos resultados obtidos pela empresa, que se baseia nesses valores para estabelecer metas de vendas e definir uma logística funcional (BONOTTO, 2015).

Objetiva-se, então, desenvolver ao longo deste documento a análise exploratória de uma base de dados acerca das vendas semanais em lojas Walmart estadunidenses, a construção de modelos preditivos baseados em técnicas de regressão e a posterior validação e comparação dos resultados obtidos. Dessa forma, espera-se chegar a um modelo de previsão de vendas que se adeque melhor à base de dados utilizada, sendo capaz de prever de forma mais precisa as demandas nesses estabelecimentos e contribuindo, então, para a otimização dos processos e maximização dos lucros.

1.2 Ferramentas utilizadas

Para a previsão de dados, o Machine Learning é um vasto universo ideal para modelar previsões de dados a partir de um histórico registrado existente. Para realizar e aplicar os conceitos de ML, foi utilizado o software Jupyter Notebook, de linguagem Python, com o uso de algumas bibliotecas importadas, listadas na imagem a seguir:

```
import os
import random
import pandas as pd
import statsmodels.api as sm
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
import statsmodels.api as sm
import seaborn as sns
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, GridSearchCV
```

Bibliotecas utilizadas para o projeto

1.3 Descrição da base de dados

A base de dados (dataset) utilizada foi obtida por meio da plataforma Kaggle, a qual disponibiliza datasets variados para uso em competições e projetos científicos, e apresenta dados semanais relativos a 45 lojas Walmart estadunidenses ao longo de um período de três anos (entre 2010 e 2012) - link da base de dados. As variáveis presentes no dataset de interesse, descritas a seguir, são: *store*, *data*, *weeklysales*, *holiday*, *temperature*, *fuel*, *cpi* e *unemployment*.

Variáveis da base de dados Walmart_Store_sales:

- *store*: representa a numeração da loja representada, 1-45;
- *data*: variável que apresenta dia, mês e ano do dado, respectivamente, representando a semana em que as observações foram tomadas;
- *weeklysales*: variável que apresenta a receita de cada loja semanalmente (US\$) (variável de monitoramento);
- *holiday*: variável que representa se a data do dado é feriado ou não (yes/no);
- *temperature*: variável que apresenta a temperatura média da semana na região de cada loja, em fahrenheit (°F);
- *fuel*: preço do galão de gasolina no dia do dado (US\$);
- *cpi*: variável que representa o preço médio necessário para comprar um conjunto de bens de consumo e serviços num país no dia, comparando com períodos anteriores (Índice de preços no consumidor) (US\$);
- *unemployment*: variável que representa a taxa de desemprego durante a semana, na região de cada loja, de zero a um.

Ao longo da análise em questão, serão considerados como Target (variável resposta ou dependente), os dados correspondentes à coluna *weeklysales*, uma vez que objetiva-se prever valores para essa variável quantitativa; e como Features (variáveis explicativas ou independentes), os dados correspondentes às demais colunas da base de dados.

Na primeira célula de código, que se encontra abaixo, colocou-se, para fins de organização, os comandos utilizados para importar todas as bibliotecas que foram utilizadas ao longo do desenvolvimento do documento em questão. Cada linha do código importa uma biblioteca distinta, a qual possibilita o uso de diferentes comandos de manipulação e análise de dados.

2. Análise descritiva detalhada das variáveis

2.1 Mineração

A primeira etapa, denominada mineração, corresponde à coleta dos dados e, uma vez que todas as informações que serão utilizadas já se encontram no dataset supracitado, tendo-se já adicionado este ao repositório, garantiu-se que o dataset está sendo reconhecido.

Posteriormente, iniciou-se a etapa de limpeza e análise descritiva a fim de compreender as características e relações entre as variáveis do dataset. Para isso, separou-se a coluna única em forma de listas, as quais foram adicionadas como valores em um dicionário cujas chaves são os nomes de cada variável presente no dataset. Esse dicionário foi, então, transformado em um DataFrame, formato que possibilita o uso de comandos da biblioteca Pandas (criada e amplamente utilizada para análise e manipulação de dados diversos) e facilita a visualização dos dados. Além disso, algumas das suas variáveis foram manipuladas para adaptá-las a uma melhor categorização, em seguida, plotou-se novamente as primeiras cinco linhas, verificando-se que a manipulação para organização do dataset funcionou como esperado, uma vez que cada variável de interesse passou a se localizar em uma coluna diferente.

2.2 Análise descritiva detalhada das variáveis quantitativas

Após a criação do Data Frame e separação das variáveis em colunas distintas, iniciou-se a análise detalhada das variáveis a fim de compreender o comportamento dessas ao longo do período a que os dados referem-se. Para isso, armazenou-se na tabela seguinte as colunas *weeklysales*, *temperature*, *fuel*, *cpi* e *unemployment* do DataFrame e, utilizando-se o comando ".describe()" obteve-se as medidas resumo de tais colunas, como pode ser visto a seguir:

	weeklysales	temperature	fuel	cpi	unemployment
count	6.435000e+03	6435.000000	6435.000000	6435.000000	6435.000000
mean	1.046965e+06	60.663782	3.358607	171.578394	0.079992
std	5.643666e+05	18.444933	0.459020	39.356712	0.018759
min	2.099862e+05	-2.060000	2.472000	126.064000	0.038790
25%	5.533501e+05	47.460000	2.933000	131.735000	0.068910
50%	9.607460e+05	62.670000	3.445000	182.616521	0.078740
75%	1.420159e+06	74.940000	3.735000	212.743293	0.086220
max	3.818686e+06	100.140000	4.468000	227.232807	0.143130

Tabela de resumo sobre as variáveis selecionadas

A obtenção destes parâmetros permite a análise de características como a média (mean) de cada variável quantitativa, a qual ilustra o comportamento médio de cada variável levando em conta os dados de todas as lojas ao longo do período analisado, o desvio padrão (std), o qual indica qual o grau de variação entre os dados considerados, alguns quantis notáveis, que indicam até cada porcentagem (25%, 50% e 75%) de dados de cada coluna, em ordem crescente, o valor máximo alcançado por eles, bem como seus valores mínimos e máximos, possibilitando uma análise inicial dos valores como um todo.

Posteriormente, verificou-se se as colunas apresentavam valores indefinidos, os quais geralmente aparecem como "Nan" e podem representar algum erro, não foram encontrados nenhum.

Em seguida, a fim de verificar de maneira mais global como a variável de monitoramento (weeklysales) comporta-se ao longo de todo o período analisado, plotou-se um gráfico de dispersão das vendas semanais (eixo y) ao longo do tempo (eixo x), no qual cada dado correspondente a essa variável presente no DataFrame. A partir do gráfico, percebe-se que nas semanas próximas ao fim de cada ano ocorre, em geral, um aumento nas vendas semanais

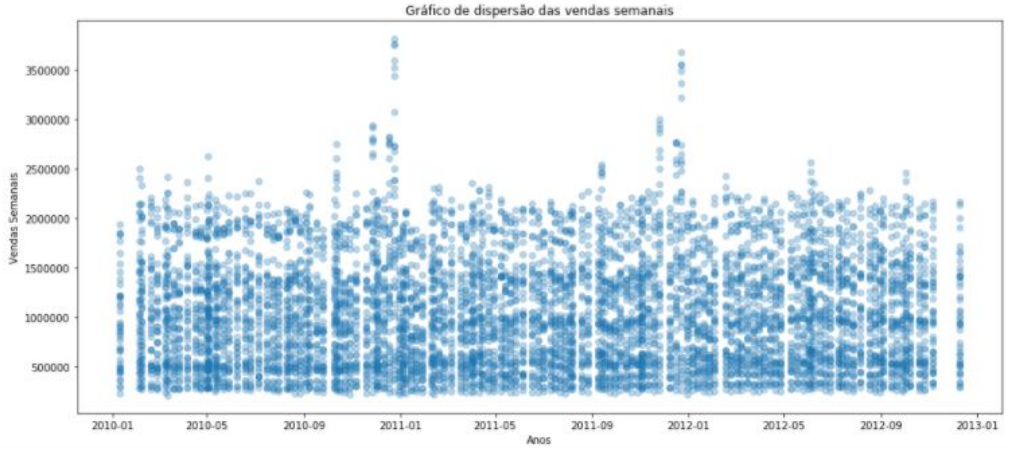

Gráfico de todos os dados

Posteriormente, visando analisar de maneira mais particular o comportamento da variável de monitoramento e de cada loja, plotou-se três gráficos de dispersão(os quais podem ser conferidos no jupyter). Para isso, utilizou-se a lógica de escolha aleatória da loja e do ano a serem plotados. Assim observou-se que em geral há aumento nas vendas semanais no período de fim de ano

2.3 Análise descritiva detalhada das variáveis weeklysales e holiday

Após a análise das variáveis anteriores, iniciou-se a de holiday, que fornece ao dia a característica de ser feriado ou não, o que é indicado pelos valores inteiros 1 ou 0, respectivamente. Para isso, utilizou-se um comando para selecionar , no DataFrame, as linhas de dias que eram feriado, nomeadas como "yes" e as de dias que não o eram, nomeadas como "no", e posteriormente selecionar apenas a coluna weeklysales para cada um desses rótulos com o uso do comando ".loc[]". Após essa seleção, plotou-se as medidas resumo para a variável weeklysales em cada situação. Analisando-se essas medidas de resumo, percebe-se que, quando há feriado, a média das vendas semanais é maior. No entanto, nessas mesmas semanas observa-se também uma variância maior, o que significa que há lojas com vendas bem maiores que outras e também que há feriados em que as vendas são maiores do que em outros.

2.4 Análise geral das relações entre weeklysales e as demais variáveis

Tendo-se realizado as análises detalhadas descritas acima, plotou-se gráficos a fim de relacionar o target às demais features, porém a A análise desses gráficos não permitiu nenhuma conclusão específica, uma vez que mesmo os dados estando agrupados não há um comportamento típico de crescimento ou decrescimento que levem à conclusão de que há uma relação positiva ou negativa, respectivamente. Dessa forma, a fim de verificar se os formatos acima permitem a percepção de alguma relação não visível apenas graficamente utilizou-se os comandos abaixo para plotar os valores de correlação, que indicam pelo seu sinal se as relações entre as variáveis é positiva ou negativa; e pelo seu valor, se são fortes ou fracas.

Procurou-se também as correlações das variáveis, as quais mostraram uma baixa relação entre as features ilustradas e o target, uma vez que os valores obtidos são pequenos. Sendo assim, faz-se também uma análise de múltiplas features de forma conjunta, onde novamente observa-se que não há nenhuma relação clara entre as variáveis

3. Modelos de predição

Após observar o comportamento da variável target em função das features selecionadas, percebeu-se que não parecia haver muita correlação entre cada feature individualmente e as vendas semanais, por isso, como primeiro modelo de inferência da variável de interesse, foi escolhida a regressão linear múltipla, na qual considera-se mais de uma feature para a previsão da target - função linear para cada feature.

Além disso, parecia razoável também monitorar as vendas semanais para cada mês do ano e para cada ano diferente. Por isso, foram adicionadas as features mês e ano de cada dado, a partir da variável de data já existente.

Finalmente, a base de dados foi dividida em um conjunto de treinamento, que serão os dados que "ensinarão" a máquina, e um conjunto de teste, a qual vai ser comparada com a previsão feita a partir do código treinado para validar o funcionamento do modelo. Para isso, embaralhou-se todos os dados e foram selecionados os primeiros 75% deles para o treinamento e o restante foi deixado para teste.

3.1. Modelo 1 - Regressão linear múltipla

Para iniciar o modelo de regressão linear múltipla, utilizou-se como features as variáveis quantitativas de temperatura (*temperature*), de preço da gasolina (*fuel*), de índice de preços no consumidor (*cpi*) e de taxa de desemprego (*unemployment*) e as variáveis qualitativas de feriado (*holiday*), de ano (*year*) e de mês (*month*), que foram quantificadas em números para o modelo. Depois de definidas as variáveis que serão trabalhadas, inicia-se o modelo.

Então, pelo método dos mínimos quadrados do modelo de regressão OLS, da biblioteca statsmodels, verificou-se a probabilidade de cada feature estar realmente associada à variação de *weeklysales*. Utilizou-se como referência um nível de significância de 0,1%, ou seja, probabilidades abaixo disso indicavam que a variável descrevia a target, visto que a hipótese inicial era de que a target fosse independente das features.

Pelos resultados obtidos por esse modelo, apenas as variáveis *cpi*, *unemployment* e *month* foram suficientes para descrever *weeklysales*. Então, restava apenas verificar se a distribuição dos resíduos - diferença entre os valores reais de vendas semanais e os previstos - seguem uma distribuição normal e se há a presença de homocedasticidade no modelo. Pelo que se observou graficamente, os resíduos não se distribuem normalmente e o modelo se mostrou altamente heterocedástico.

Como o modelo não parecia muito confiável do jeito que estava, para verificar se havia erro da distribuição de dados, decidiu-se aplicar toda a mesma metodologia de regressão linear múltipla, mas considerando os valores logarítmicos das variáveis.

A partir dos resultados obtidos pelo modelo, mas considerando a nova escala, as mesmas variáveis *cpi*, *unemployment* e *month* descreviam *weeklysales*, mas agora com a adição da variável *temperature*. Mesmo assim, os resíduos ainda não apresentavam nenhuma evidência de normalidade e o modelo ainda se mostrou muito heterocedástico.

3.2. Modelo 2 - Janelas deslizantes (médias móveis)

Tentando entender como o modelo de janelas deslizantes funciona, o grupo não teve sucesso em interpretar e implementar esse modelo de forma a considerar a influência de mais de uma feature ao mesmo tempo, e não pareceu ser um modelo muito razoável para a previsão dessa variável, ainda mais que o modelo simplesmente faz uma média dos últimos n dados para prever o próximo valor.

Mesmo assim, o grupo implementou o modelo considerando apenas o tempo como feature e usou uma janela de 2 valores, ou seja, cada valor previsto equivalia à média aritmética dos últimos 2 dados.

Independentemente do resultado, o modelo deixou de ser usado, pois a previsão de vendas que se objetiva alcançar engloba uma escala que consiga prever semanas antes de acontecer.

3.3. Modelo 3 - Random Forest Regression

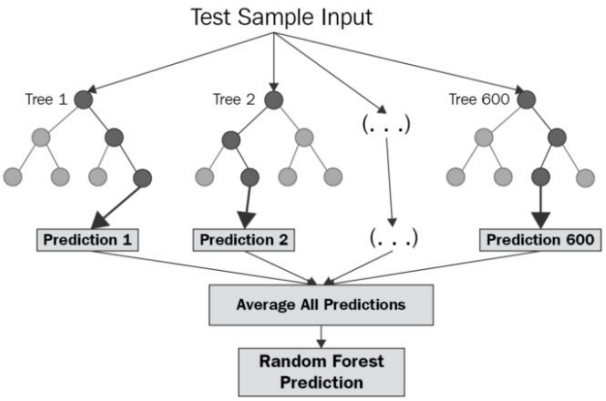

Diagrama de explicação do funcionamento do modelo

Pensando nas nossas variáveis de feature e o modo como poderíamos analisar os comportamentos de vendas pela aleatoriedade de datas e de cada loja buscamos um modelo que conseguisse processar igualmente as features. Assim encontramos o Random Forest Regression. O modelo consiste em rodar diversas árvores de decisão em um modo paralelo sem interação entre elas, fazendo então uma média das classificações ou predição (caso do projeto) dos outputs de cada árvore.

Como o modelo agrega muitas árvores de decisão, ele junta diversos resultados, trazendo a possibilidade dos nós das árvores dependerem igualmente de todas as features, tomando decisões de forma mais justa. Para isso utilizamos os chamados hiperparâmetros, que podem decidir o número que árvores, quantas features vão ser testadas na árvore, e valores mínimos de amostra para divisão dos módulos.

Para começar a implementar o modelo no projeto importamos um comando chamado GridSearchCV da biblioteca sklearn.model_selection, o qual tem a função de buscar exaustivamente por parâmetros específicos para a variável de interesse (weekly sales no caso), a partir de um modelo é um parâmetro ele prove scores e predict para o modelo. Os parâmetros que são retornados são os hiperparâmetros que podemos usar na divisão das árvores de decisão para melhores resultados.

Quando aplicado no projeto, a partir da variável grid, fizemos o teste para as nossas amostras de teste (tanto paras as features quanto target), entendendo assim qual foi a configuração de parâmetros que forneceu o melhor resultado, dando oportunidade de otimização. O resultado obtido foi o seguinte :

```
Out[52]: GridSearchCV(cv='warn', error_score='raise-deprecating',
                      estimator=RandomForestRegressor(bootstrap=True, criterion='mse',
                                                      max_depth=None,
                                                      max_features='auto',
                                                      max_leaf_nodes=None,
                                                      min_impurity_decrease=0.0,
                                                      min_impurity_split=None,
                                                      min_samples_leaf=1,
                                                      min_samples_split=2,
                                                      min_weight_fraction_leaf=0.0,
                                                      n_estimators='warn', n_jobs=None,
                                                      oob_score=False, random_state=None,
                                                      verbose=0, warm_start=False),
                      iid='warn', n_jobs=None,
                      param_grid={'min_samples_leaf': [1, 10],
                                  'min_samples_split': [2, 10],
                                  'n_estimators': [100, 250, 500, 750]},
                      pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
                      scoring=None, verbose=0)
```

Em seguida buscamos o erro quadrado, que se aproximou de 19216778902.851364, e o best score, que se aproximou de 0.9387025150496138, os valores podem variar a cada teste. Além disso também obtivemos os valores e predição no teste.

Assim concluímos a modelagem dos dados a partir do Random Forest regression, chegando a valores razoáveis de modelagem e prontos para a validação.

4. Validação

A fim de validar os modelos de predição, achou-se importante a análise quanto a qualidade de ajuste entre os dados de teste e a previsão feita pelos modelos, viabilizando a utilização da medida R^2 . Somado à tal análise, também foi realizada uma observação no tocante ao erro, utilizando a medida RMSE (Raiz do erro quadrático médio).

A medida R^2 é calculada por meio da seguinte fórmula: $R^2 = 1 - \frac{Variações\ inexplicáveis}{Variação\ Total}$. Sendo assim, tem-se que esse parâmetro tem seus valores contidos em um intervalo de 0 até 1, sendo valores maiores que 0,85, representantes de uma forte correlação entre as variáveis de teste e a prevista pelo modelo.

Já a medida RMSE, é calculada por meio da seguinte fórmula: $\sqrt{\frac{\sum_{k=0}^{n} e_k^2}{n}}$, sendo e, o erro, n a quantidade de valores contidas na base de testes.

Portanto, quanto menor é o parâmetro RMSE, melhor é para a validação do modelo.

4.1. Modelo 1 - Regressão Linear Múltipla

Para validar o modelo 1 de regressão linear múltipla, primeiro plotou-se um gráfico, o qual mostra a venda semanal das semanas e lojas contidas na base de testes e a venda semanal média prevista pelo modelo (Gráfico 1).

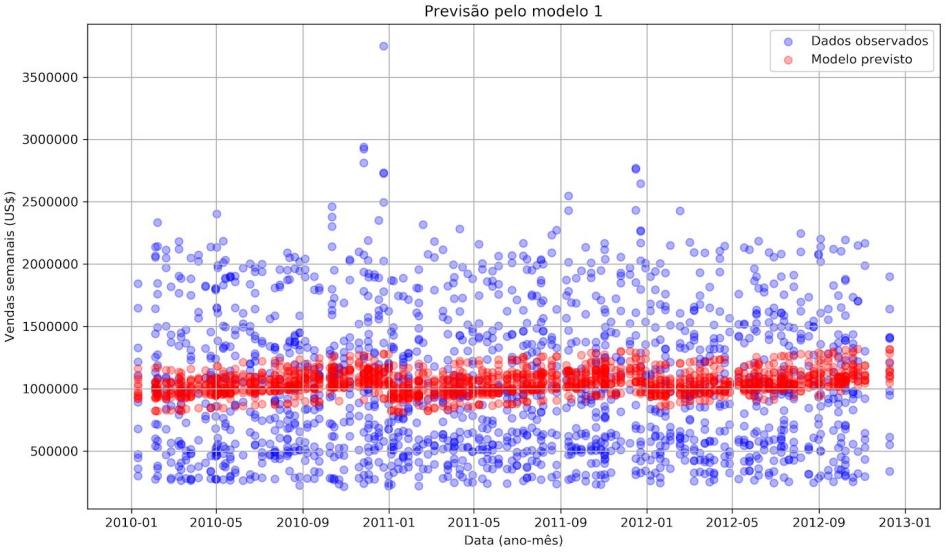

Gráfico 1

Após isso, calculou-se o RMSE, o qual deu bastante alto, comprovando o que é visto no gráfico 1. Já em relação ao parâmetro R^2, atestou-se também um valor bastante baixo de correlação.

Apesar desses resultados comprovaram que o modelo 1 é bastante falho, pode-se compreender o porquê dessa discrepância. Isso acontece, pois esse mesmo modelo não foi treinado por loja e sim, considerando as features de todas as lojas. Isso retorna então um valor target médio.

4.2. Modelo 1 - Regressão Linear Múltipla com escala logarítmica

Sendo assim, tendo em vista os resultados obtidos anteriormente, a fim de diminuir tal erro, colocou-se todas as features e o target em escala logarítmica.

Obteve-se assim, um valor de RMSE bastante inferior porém, em o valor do parâmetro R^2, o qual já era baixo, decresceu mais ainda.

Dessa forma, pode-se atestar de que o modelo 1 não obteve resultados de validação satisfatórios.

4.3. Modelo 3 - Random Forest Regression

Portanto, já que o modelo 1 não obteve resultados satisfatórios de validação, buscou-se por um modelo que satisfizesse melhor a predição da variável target. Sendo assim, os melhores resultados foram obtidos com o uso do modelo de Random Forest Regression, o qual obteve um R^2 maior que 0,85 e um RMSE menor que o obtido pelo modelo de regressão linear múltipla.

Além disso, plotou-se também um gráfico para analisar como o modelo iria prever as vendas semanais por loja (Gráfico 2).

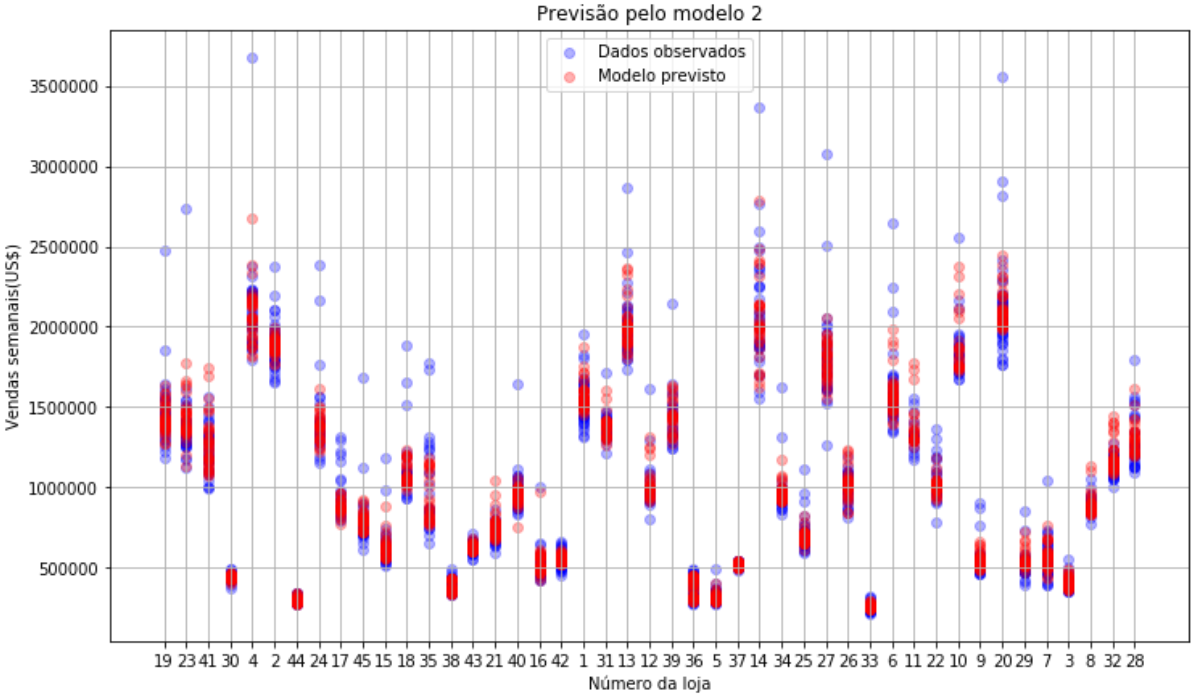

Gráfico 2

Deste modo, tendo em posse a análise dos parâmetros e do gráfico, pode-se concluir que esse modelo serve bem para prever a variável target, uma vez que tem-se um R^2 bastante alto, um RMSE razoável e analisando o gráfico de dispersão, percebe-se uma boa proximidade entre o modelo previsto e os dados observados.

5. Conclusão

Dentro de um contexto no qual a competição faz-se sempre presente, as previsões de demandas tornam-se essenciais à gestão das mais diversas organizações e, em especial no setor varejista, o qual passa, diariamente, por inúmeras transações. Tais previsões, quando adequadas, contribuem na tomada de decisões estratégicas e ajudam a reduzir desperdícios, sendo fundamentais para que a logística, estoque e compras não sejam fatores problemáticos, situação em que podem gerar prejuízo, mas sim fatores de geração de lucro (BONOTTO, 2015).

Nesse contexto, as ferramentas de Machine Learning são de grande apoio quando se tem um histórico de dados já registrado a disposição e objetiva-se prever valores futuros para uma variável de interesse buscando maior eficiência. Tendo-se isso em vista, é fundamental que o encarregado pelo manejo da base de dados use e busque conhecimentos cada vez mais refinados a fim de aperfeiçoar a análise e previsão.

Dessa forma, de acordo com as análises desenvolvidas ao longo deste relatório e com os modelos construídos, percebe-se que o modelo 1 não obteve resultados tão satisfatórios, uma vez que apresentou valores altos de erro e baixos de ajuste. Sendo assim, os melhores resultados foram obtidos com o uso do modelo de Random Forest Regression, o qual apresentou um R^2 maior que 0,85 e um RMSE menor que o obtido pelo modelo de regressão linear múltipla.

Percebe-se então, que o modelo construído é capaz de auxiliar a empresa a ter um maior entendimento acerca das vendas semanais, possibilitando uma gestão mais eficiente da logística, estoque e compras. Assim, pode-se contribuir para a redução dos desperdícios, prevenção de prejuízos e maximização de lucros.