Nome: Claudio Escudero

Modelo de previsão de demanda

Desafio da B2W

Table of Contents

Introdução

A B2W enviou 2 arquivos, um contendo o histórico de 9 produtos de 6 concorrentes de e-commerce e o outro arquivo um histórico de vendas desses dos 9 produtos contendo o preço e quantidade do produto vendido em uma loja própria.

Detalhes dos arquivos abaixo:

- **comp_prices.csv** (Histórico do concorrente)
  - **PROD_ID:** Código do produto. P1 ... P9 (9 produtos)
  - **DATE_EXTRACTION:** Data e hora da extração
  - **COMPETITOR:** Código do concorrente: C1 ... C6 (6 concorrentes)
  - **COMPETITOR_PRICE:** Preço da extração do produto no concorrente
  - **PAY_TYPE:** Tipo de pagamento. 1 ou 2
- **sales.csv (**Histórico das vendas da loja própria)
  - **PROD_ID:** Código do produto. P1 ... P9 (9 produtos)
  - **DATE_ORDER:** Data da venda (sem a hora)
  - **QTY_ORDER:** Quantidade de produtos vendido no pedido
  - **REVENUE:** Valor total do pedido.

O desafio foi desenvolvido em 3 etapas:

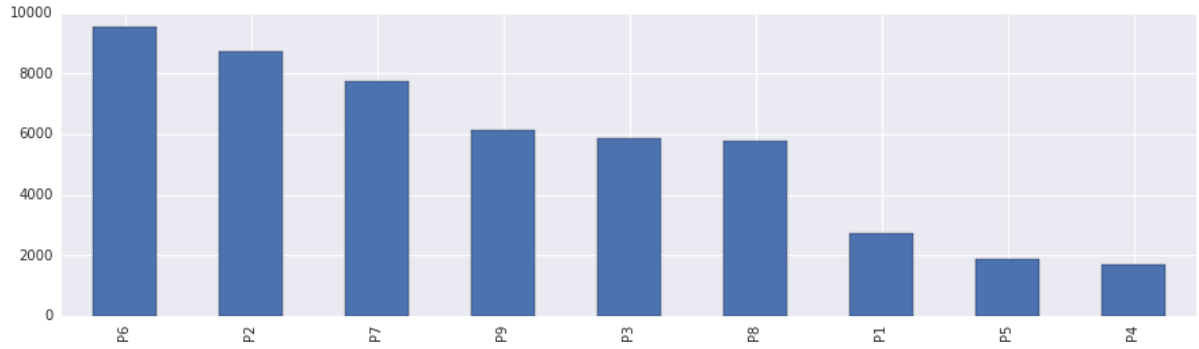
- Análise de exploratória e outliers
- Modelagem / Cross-Validation
- Backtesting

O desafio foi desenvolvido usando a estratégia de Backtesting, com isso as duas bases foram divididas na data 14/SET/2016 (30 dias antes do último registro), assim podemos desenvolver a modelagem com a primeira parte da base com o esperado do resultado da segunda parte em diante os resultados continuem a ser semelhantes aos que foram obtidos na primeira parte.

Exploração dos dados

Base dos concorrentes

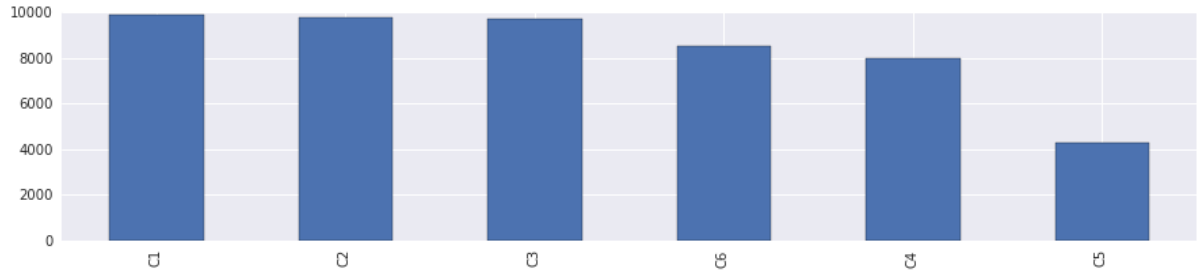
Quantidade de extrações por produto em todos os concorrentes



	P6	P2	P7	P9	P3	P8	P1	P5	P4
Qtd.	9542	8755	7748	6123	5853	5795	2713	1896	1689

OBS: Essa contagem não está considerando quantas vezes os valores foram alterados, sim somente a quantidade de extrações.

Quantidade de extrações dos concorrentes

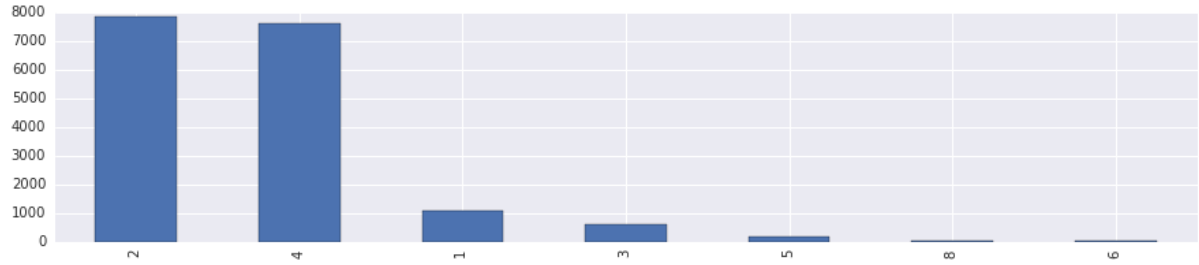


	C1	C2	C3	C6	C4	C5
Qtd.	9868	9770	9695	8505	7989	4287

OBS: Essa contagem não está considerando quantas vezes os valores foram alterados, sim somente a quantidade de extrações.

Quantidade de registros por dia

Para calcular a quantidade de registros por dia, foi agrupado e somado a quantidade de registros com a seguinte chave: **['PROD_ID', 'COMPETITOR', 'PAY_TYPE', 'DATE_ONLY']**



	2	4	1	3	5	8	6
Qtd.	**7845**	**7617**	**1070**	**579**	**173**	**34**	**2**

No documento diz que as extrações foram feitas apenas 2x ao dia, mas na tabela acima podemos verificar que existem casos com mais vezes.

Removendo os registros com valores repetidos dentro do mesmo dia. Assim deixaremos uma base menor e limpa.

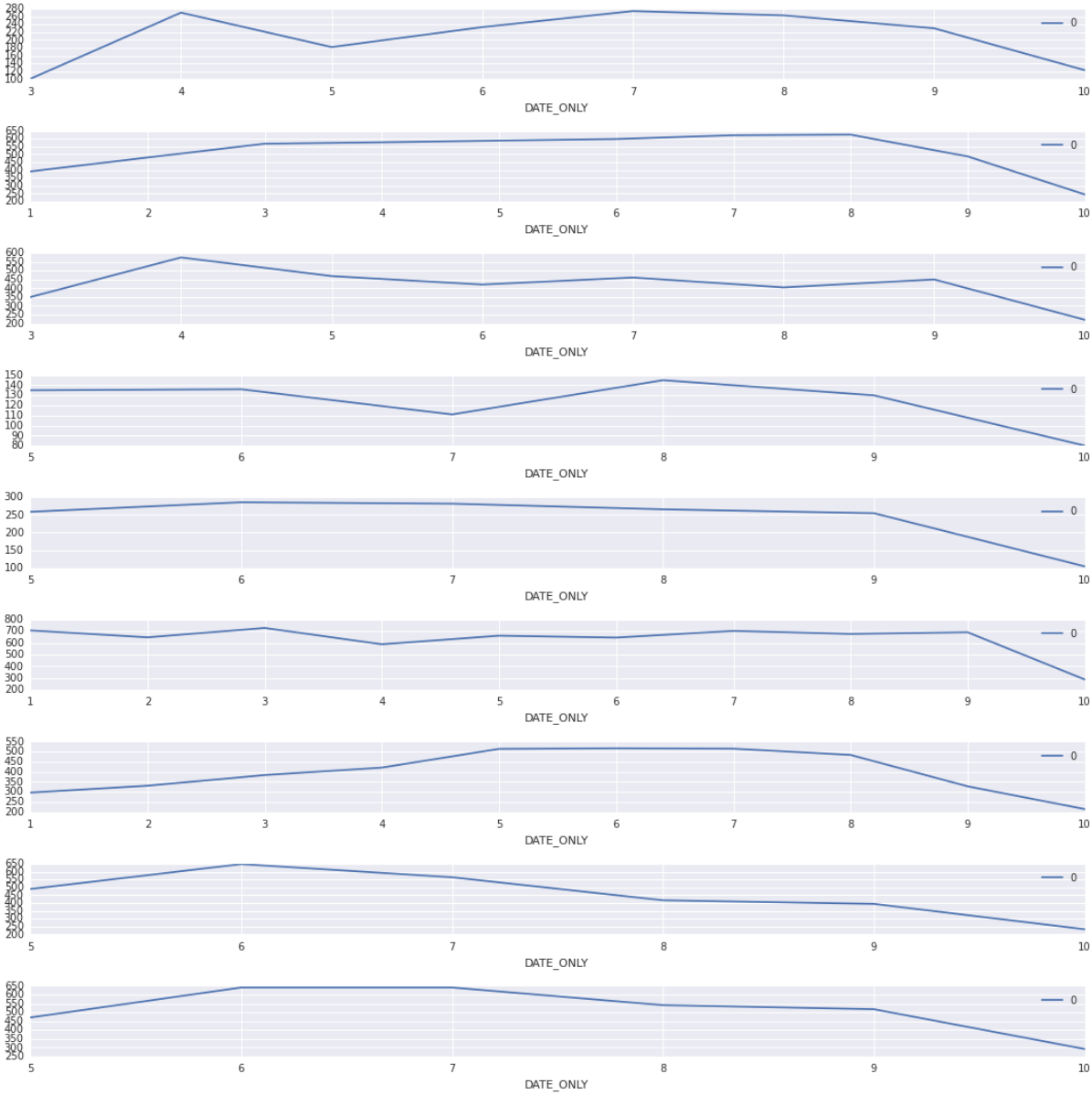
	1	2	3	4
Qtd.	**9084**	**5863**	**1710**	**663**

Como podemos perceber, existem extrações que os preços foram alterados (até 4x) dentro do mesmo dia.

OBS: Foram desconsiderados casos que um produto pode ter iniciado com o preço **X**, depois foi alterado para **Y** e logo em seguida voltou para **X**.

Quantidade de extrações por produto durante os meses

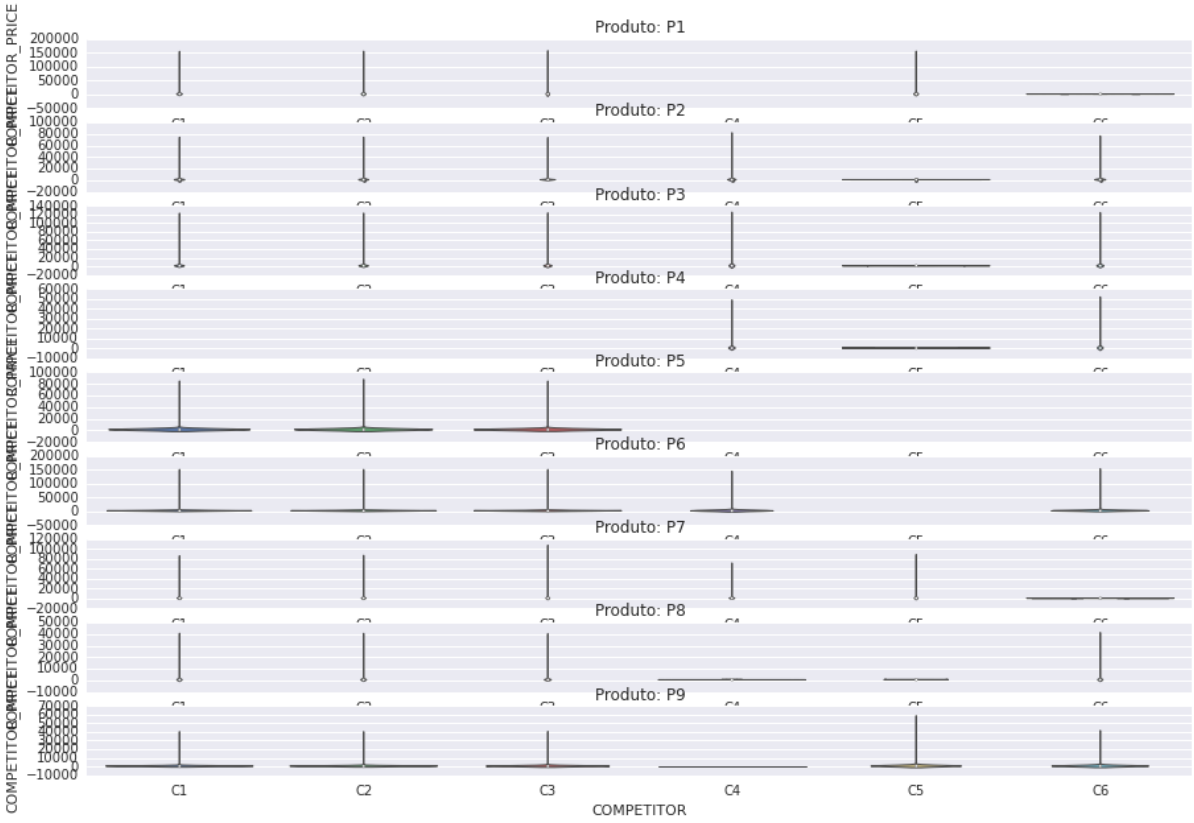
A tabela abaixo informa a quantidade de extrações por produto para cada linha.



Podemos verificar quem nem todos os produtos foram extraídos a partir de janeiro, mas todos foram coletados até outubro.

Valor médio dos produtos para cada concorrente e removendo outliers

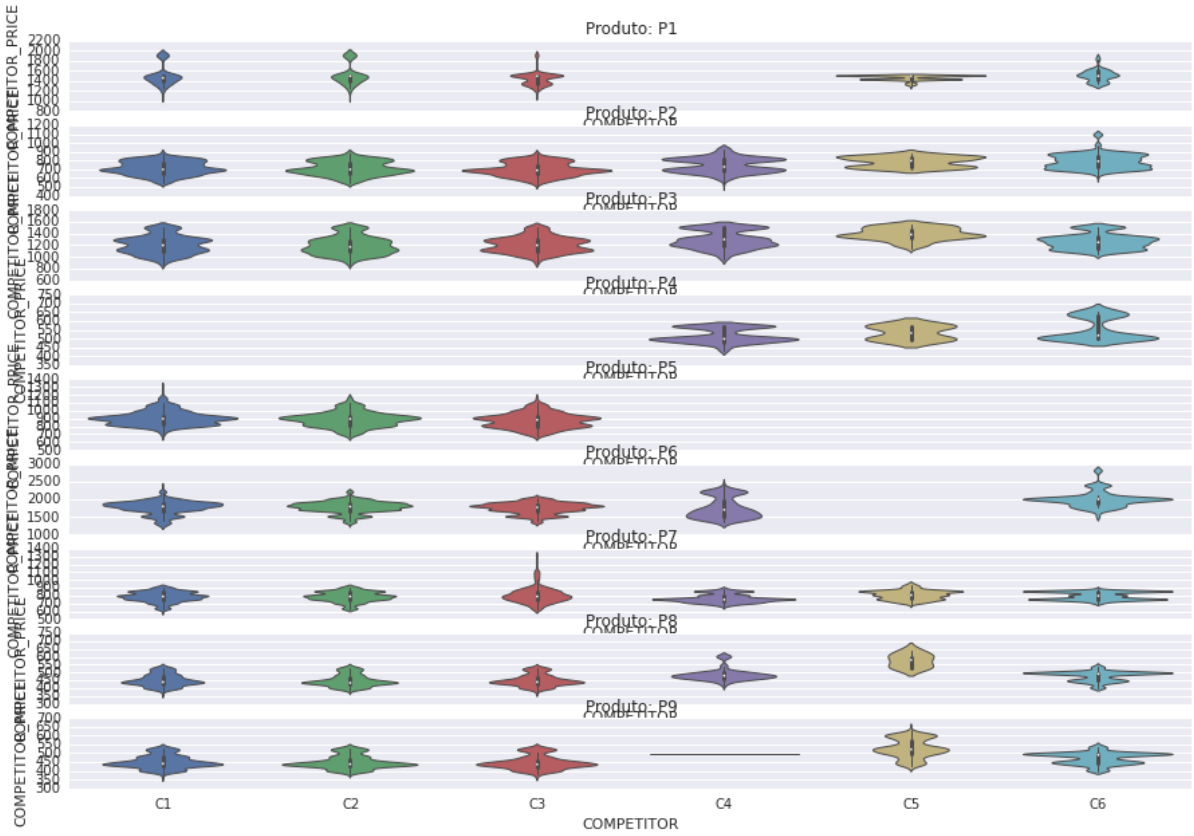
A grid de gráficos (violin) abaixo possui o preço dos produtos por concorrente.



Podemos perceber existem alguns outliers com os valores extremamente grande.

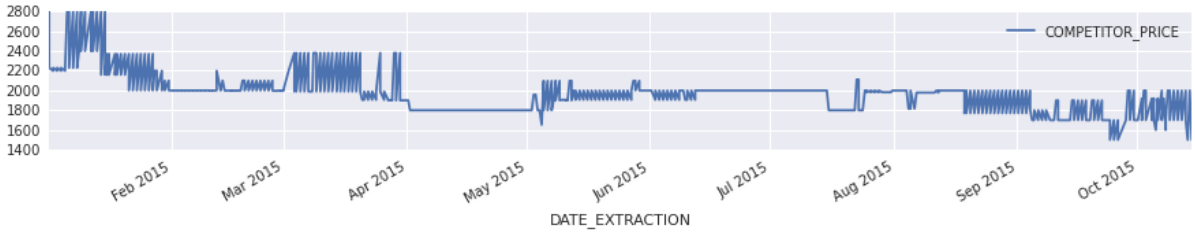
Esses erros podem ter ocorrido com algum problema de extração ou até mesmo erro de digitação.

Abaixo possui o mesmo gráfico sem os outliers.



Verificando histórico de preço de 1 produto de 1 concorrente

Selecionei o produto P6 e concorrente C6 para explorar o histórico.



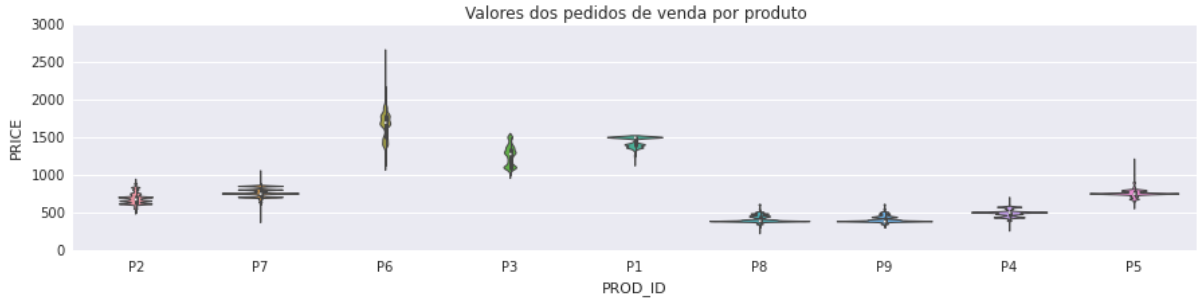
Podemos perceber que o histórico teve diversas alterações bem próximos, provavelmente são alterações dentro do mesmo dia.

Vamos normalizar os valores com valor médio do dia e gerar o gráfico novamente.

Gráfico abaixo são com todos os produtos e concorrentes.

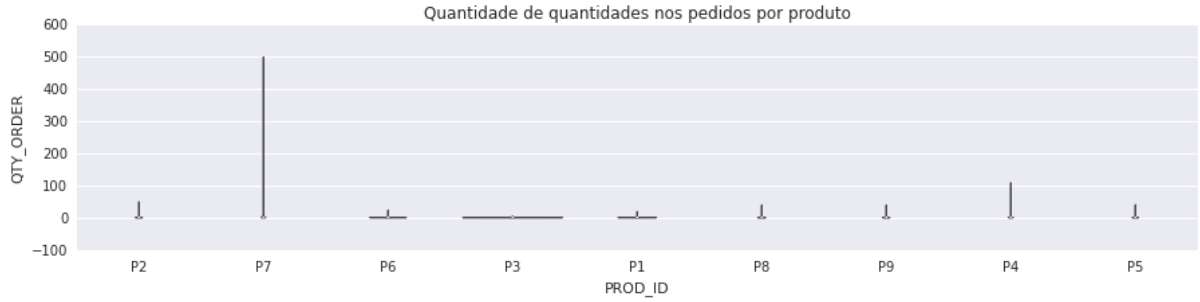
Base das vendas

Valores dos pedidos de venda por produto



Os preços acima aparentemente possuem um comportamento normal.

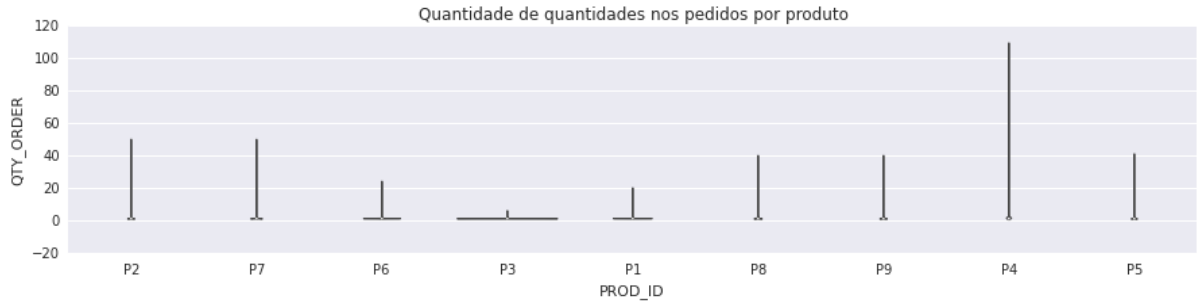
Quantidade de quantidades nos pedidos por produto



Visualmente existe um pedido com um grande número de quantidade de itens para o produto P7, aproximadamente 500 itens.

Será que vendeu mesmo com essa quantidade? Ou é um erro de extração? Como é um caso raro, resolvi remover das análises e entendi que se encaixa como um valor outlier.

Exibindo novamente o gráfico sem o outlier.



OBS: Podemos estudar mais esses valores, podem existir mais casos raros e entender que são outliers.

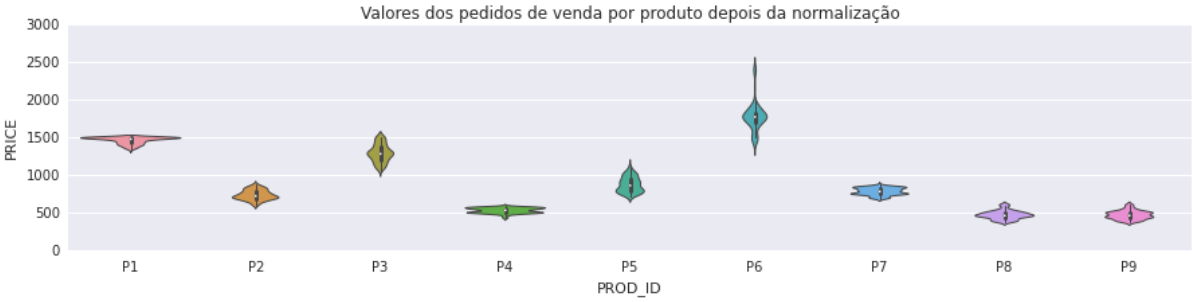
Normalizando os preços e quantidades de venda por dia

Os pedidos foram agrupados diariamente somando as colunas de quantidades e tirando uma média ponderada dos valores do pedido.

Assim para cada produto podemos ter 1 registro por dia, com o valor médio do produto com a quantidade de vendas durante o dia.

Existiam **351090** registros antes da normalização, depois da normalização existem apenas **2162** registros.

Valores dos pedidos de venda por produto depois da normalização



Treinamento e modelando

Como foi feito?

O desafio foi desenvolvido com a estratégia de backtesting, a base foi dividida na data 14/SET/2016 (30 dias antes do último registro), assim podemos desenvolver a modelagem com a primeira parte da base com o esperado da segunda parte em diante os resultados continuem a ser semelhantes aos que foram obtidos na primeira parte.

Foi utilizado o método XGBoost com a métrica Root Mean Squared Error.

A modelagem foi criada e testada em 5 etapas:

1) Criação das features (Comum para todos os produtos)
2) Seleção das features mais relevantes
3) Hyperparameter no método XGBoost
4) Treinamento com dados de treino do backtesting
5) Backtesting

Criação de features

As features criadas foram:

- Preço da venda
- Preços dos concorrentes
- Preços dos concorrentes do dia anterior
- Diferença do preço da venda para cada concorrente
- Quantidade vendida dos últimos 3 dias
- Dia
- Semana
- Valores mínimos dos concorrentes
- Valores máximos dos concorrentes
- Meses com mais relevância
- Início de mês
- Final de mês

OBS: Para os valores em branco, que provavelmente são produtos ausentes nas lojas concorrências, colocarei o valor de 99999.

Seleção das features mais relevantes

A seleção de feature foi utilizado o método GradientBoostingRegressor do sklearn para rankear as mais relevantes para cada produto.

A seleção das features foram com o critério de estar acima de 0.005, que é um grau de relevância.

Hyperparameter no método XGBoost

O método utilizado foi o XGBoost, é um método simples que facilmente pode ser utilizado para preditivos que normalmente trazem bons resultados.

Foi executado para cada produto a busca de um melhor parâmetro foi utilizado o GridSearchCV do sklearn com os intervalores de:

- **learning_rate:** [0.1, 0.2, 0.3]
- **max_depth:** [2, 3, 4, 5, 6, 7, 8, 9]
- **min_child_weight:** [1, 2, 3, 4, 5, 6, 7, 8, 9]
- **gamma:** [0.0, 0.1, 0.2]

A métrica utilizada foi Root Mean Squared Error, que é bem popular para modelos de regressão.

OBS: O ideal é criar uma análise mais profunda para verificação de overfiting.

Treinamento com dados

Depois de executar o cross-validation e hyperparameter, foi utilizado a configuração encontrada no XGBoost.

Essa configuração pode ser diferente para cada produto, pois eles podem ter comportamentos diferenciados nas séries temporais.

Backtesting

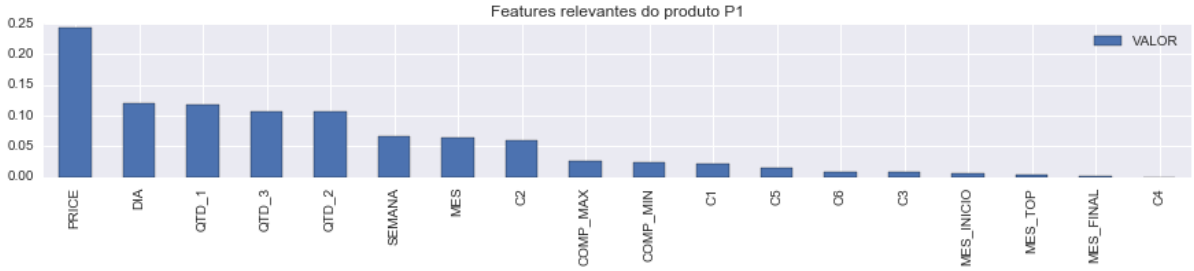
Com os treinamentos finalizados para cada produto, foram testados com os dados do backtesting e foi gerado um gráfico comparativo com o valor realizado e previsto.

O resultado será mostrado na sessão de “Resultados”.

Resultados

Produto P1

Ranking de features mais relevantes



Features selecionadas

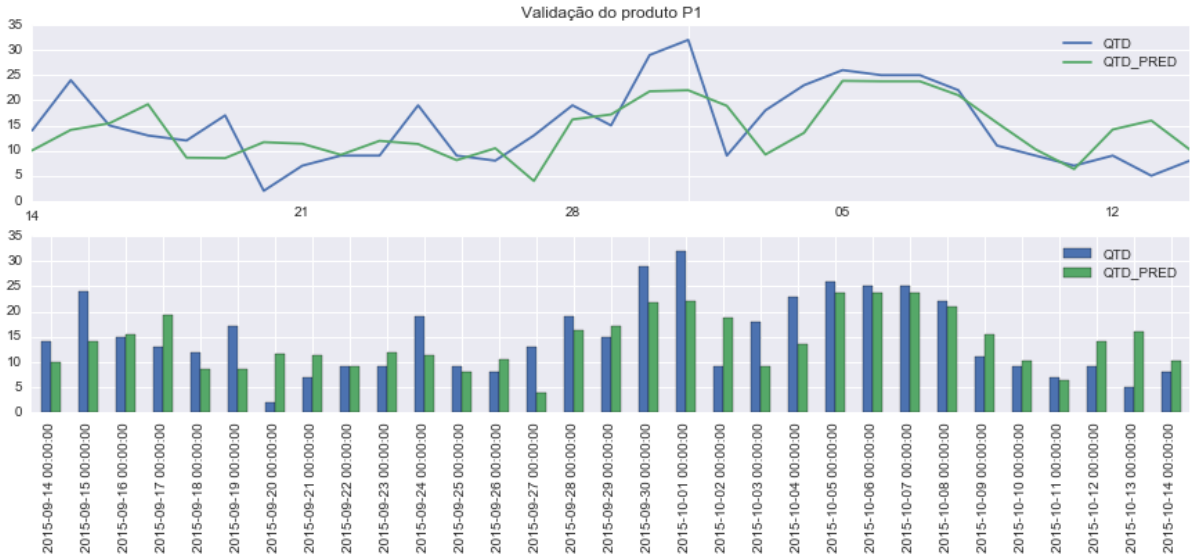
PRICE, DIA, QTD_1, QTD_3, QTD_2, SEMANA, MÊS, C2, COMP_MAX, COMP_MIN, C1, C5, C6, C3, MES_INICIO

Ranking da relevância dos concorrentes: C2, C1, C5, C6, C3, C4

Parâmetros selecionados para o método XGBoost

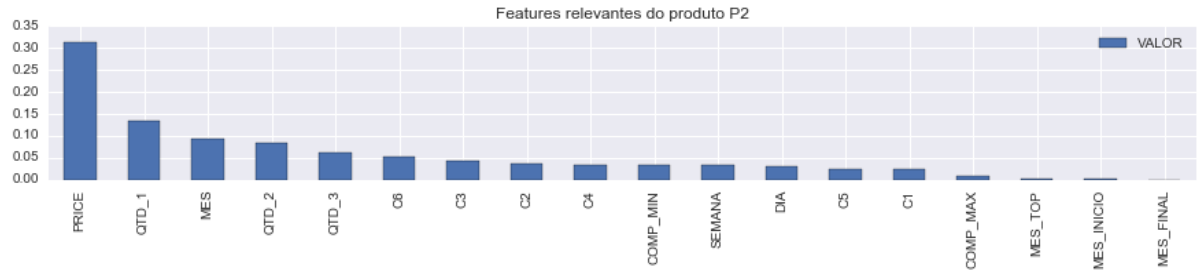
learning_rate = 0.1, max_depth = 4, gamma = 0.0, min_child_weight = 5

Gráfico comparativo no backtesting



Produto P2

Ranking de features mais relevantes



Features selecionadas

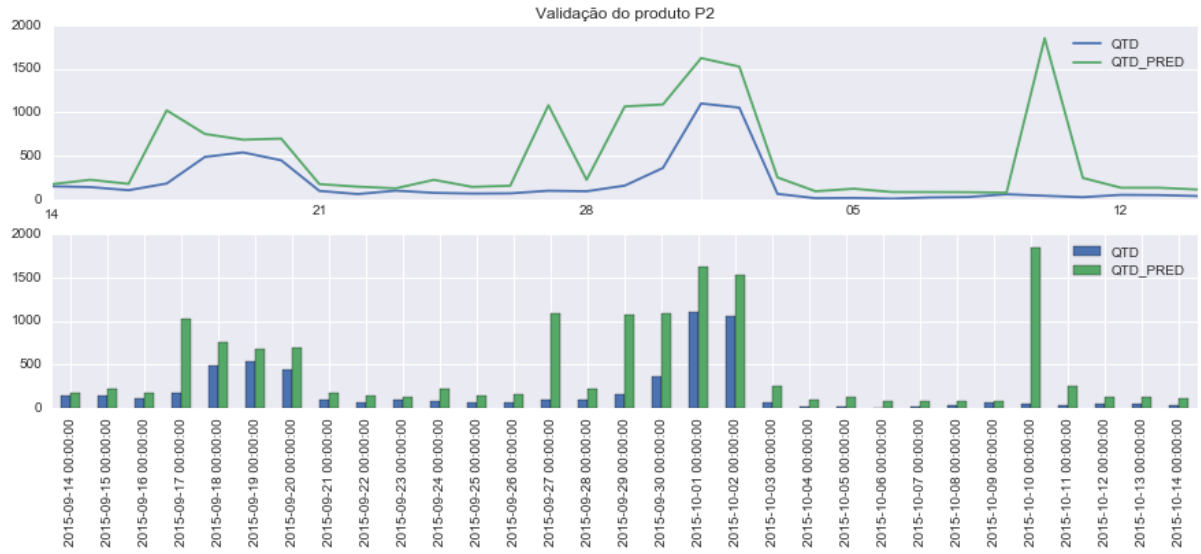
PRICE, QTD_1, MES, QTD_2, QTD_3, C6, C3, C2, C4, COMP_MIN, SEMANA, DIA, C5, C1, COMP_MAX

Ranking da relevância dos concorrentes: C6, C3, C2, C4, C5, C1

Parâmetros selecionados para o método XGBoost

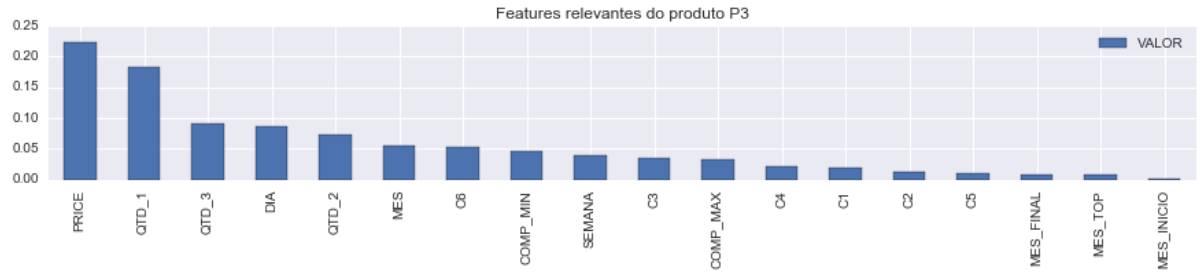
learning_rate = 0.1, max_depth = 2, gamma = 0.0, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P3

Ranking de features mais relevantes



Features selecionadas

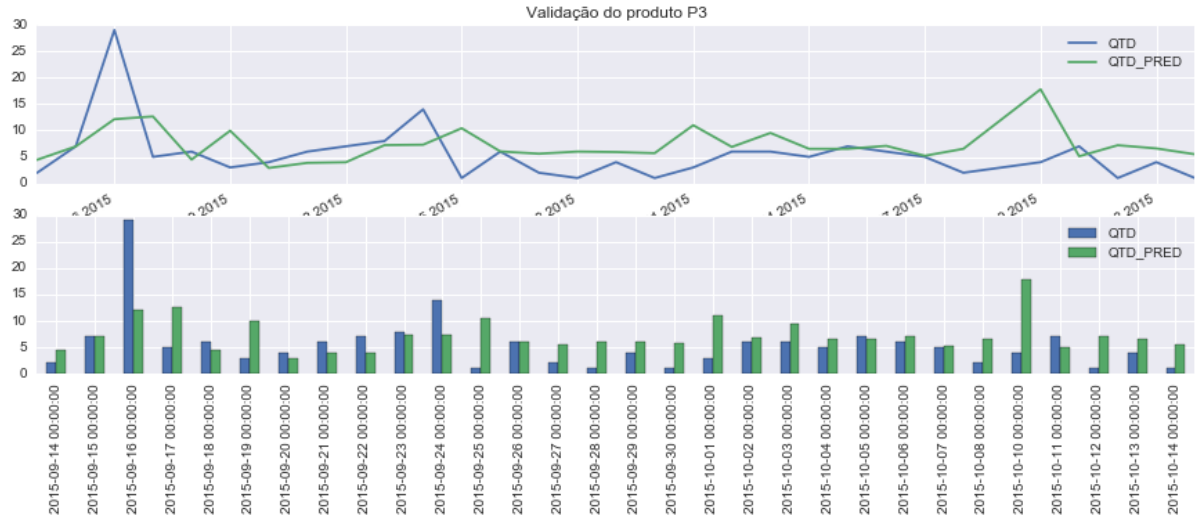
PRICE, QTD_1, QTD_3, DIA, QTD_2, MES, C6

Ranking da relevância dos concorrentes: C6, C3, C4, C1, C2, C5

Parâmetros selecionados para o método XGBoost

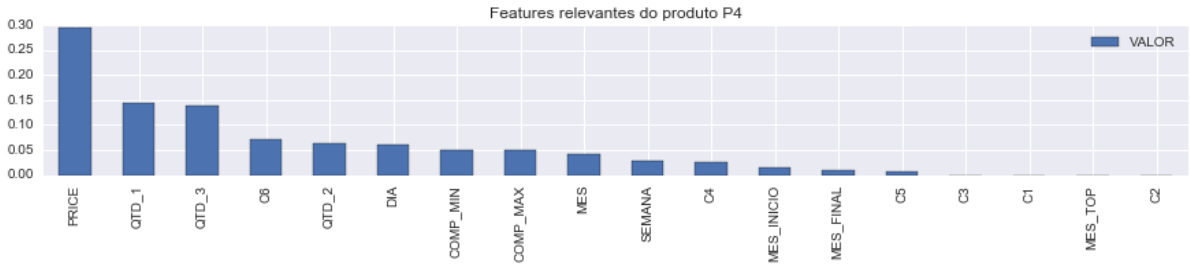
learning_rate = 0.1, max_depth = 2, gamma = 0.0, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P4

Ranking de features mais relevantes



Features selecionadas

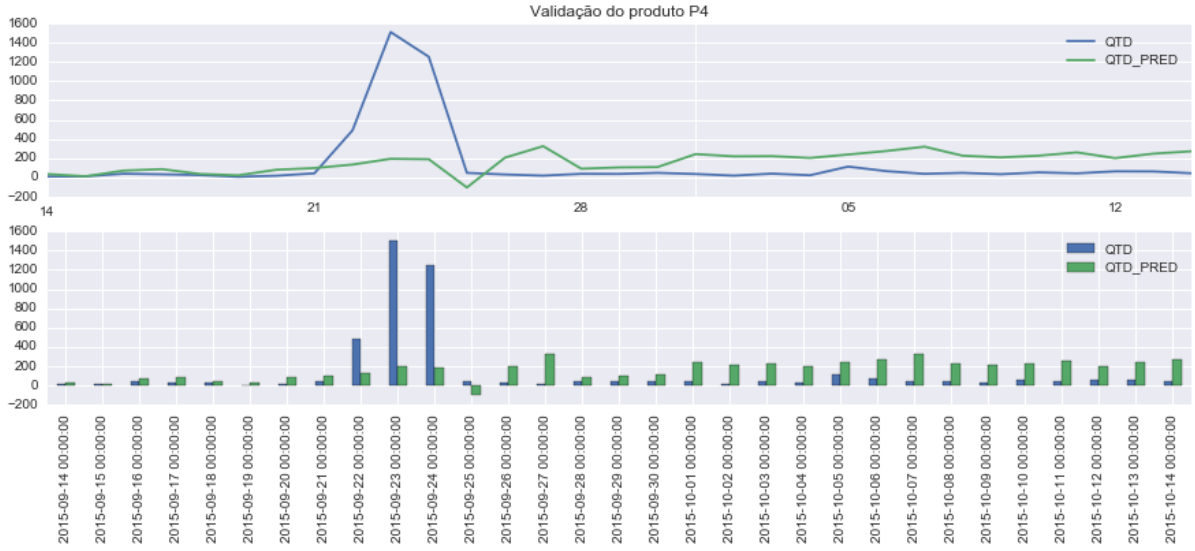
PRICE, QTD_1, QTD_3, C6, QTD_2, DIA, COMP_MIN, COMP_MAX, MES, SEMANA, C4, MES_INICIO, MES_FINAL, C5

Ranking da relevância dos concorrentes: C6, C4, C5, C3, C1, C2

Parâmetros selecionados para o método XGBoost

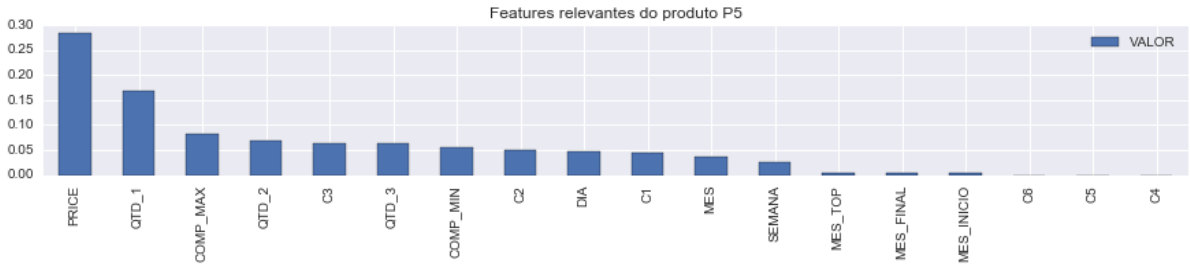
learning_rate = 0.1, max_depth = 2, gamma = 0.0, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P5

Ranking de features mais relevantes



Features selecionadas

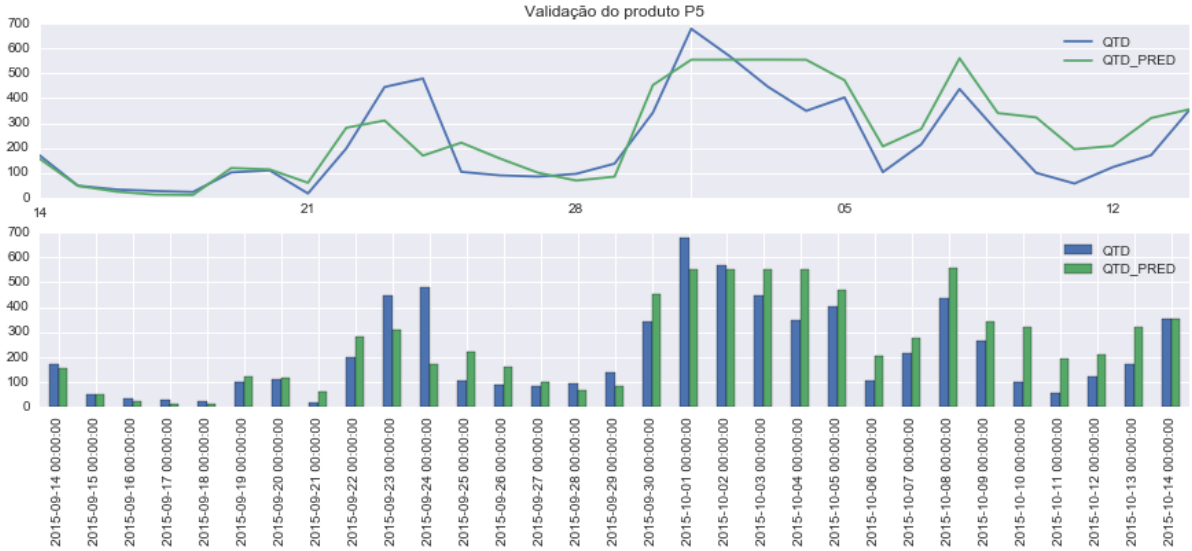
PRICE, QTD_1, COMP_MAX, QTD_2, C3, QTD_3, COMP_MIN, C2, DIA, C1, MES, SEMANA, MES_TOP

Ranking da relevância dos concorrentes: C3, C2, C1, C6, C5, C4

Parâmetros selecionados para o método XGBoost

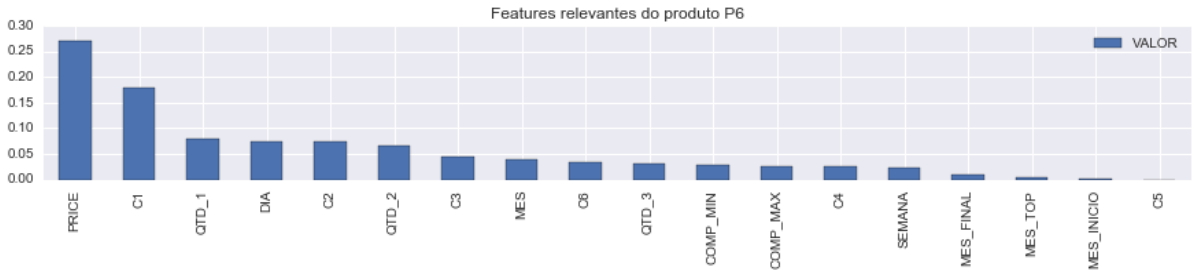
learning_rate = 0.2, max_depth = 6, gamma = 0.1, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P6

Ranking de features mais relevantes



Features selecionadas

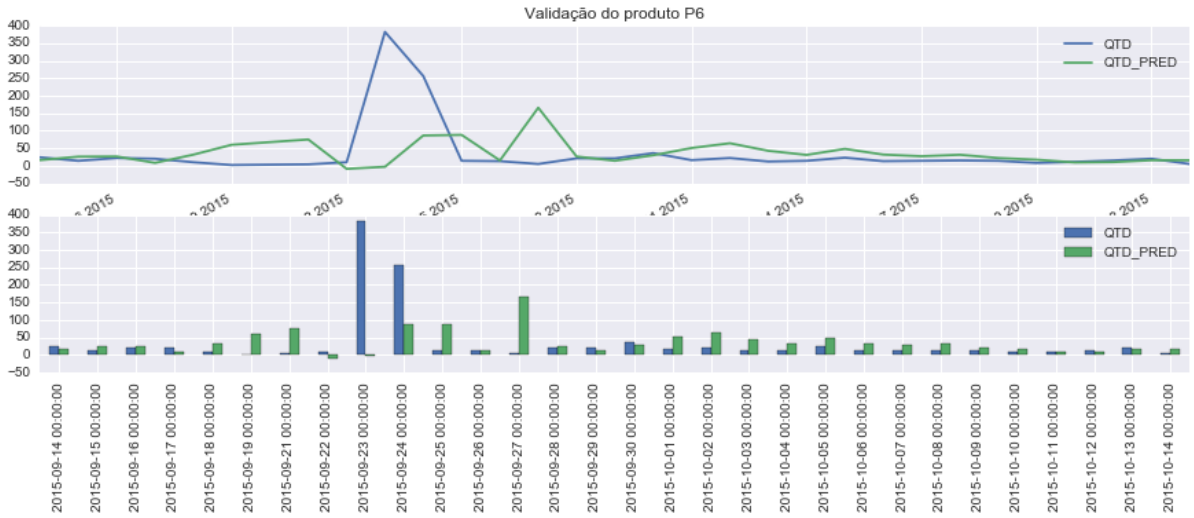
PRICE, C1, QTD_1, DIA, C2, QTD_2, C3, MES, C6, QTD_3, COMP_MIN, COMP_MAX, C4, SEMANA, MES_FINAL

Ranking da relevância dos concorrentes: C1, C2, C3, C6, C4, C5

Parâmetros selecionados para o método XGBoost

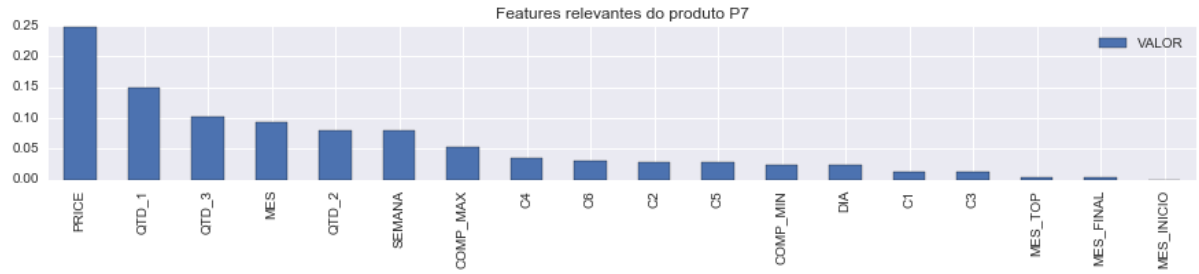
learning_rate = 0.1, max_depth = 6, gamma = 0.0, min_child_weight = 6

Gráfico comparativo no backtesting



Produto P7

Ranking de features mais relevantes



Features selecionadas

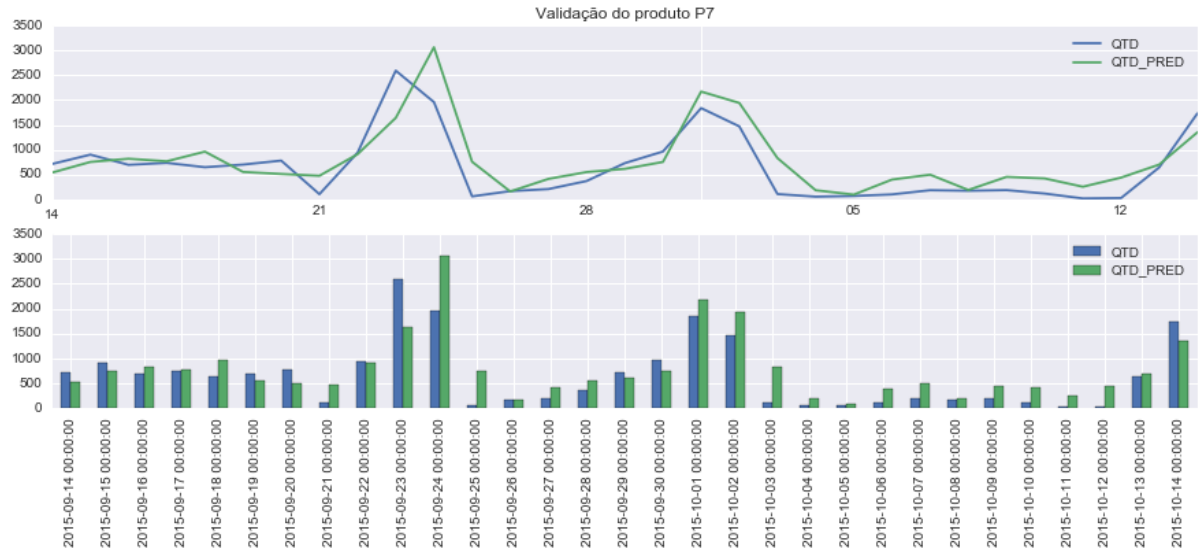
PRICE, QTD_1, QTD_3, MES, QTD_2, SEMANA, COMP_MAX, C4, C6, C2, C5, COMP_MIN, DIA, C1, C3

Ranking da relevância dos concorrentes: C4, C6, C2, C5, C1, C3

Parâmetros selecionados para o método XGBoost

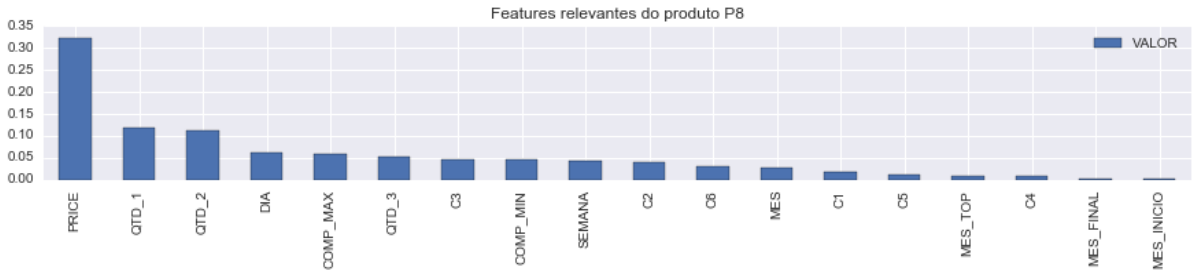
learning_rate = 0.1, max_depth = 2, gamma = 0.0, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P8

Ranking de features mais relevantes



Features selecionadas

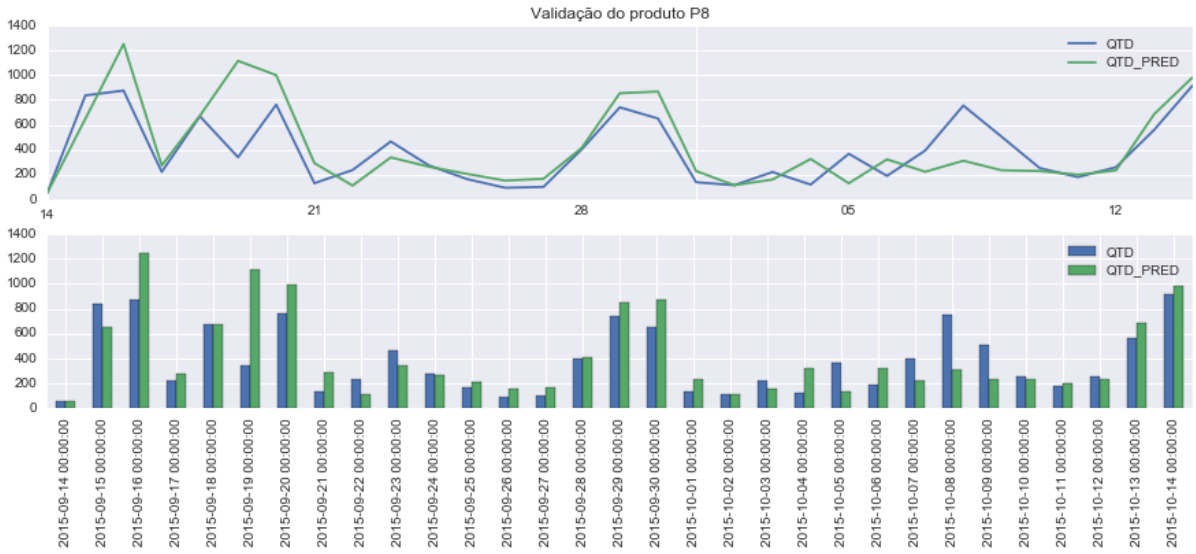
PRICE, QTD_1, QTD_2, DIA, COMP_MAX, QTD_3, C3, COMP_MIN, SEMANA, C2, C6, MES, C1, C5, MES_TOP, C4

Ranking da relevância dos concorrentes: C3, C2, C6, C1, C5, C4

Parâmetros selecionados para o método XGBoost

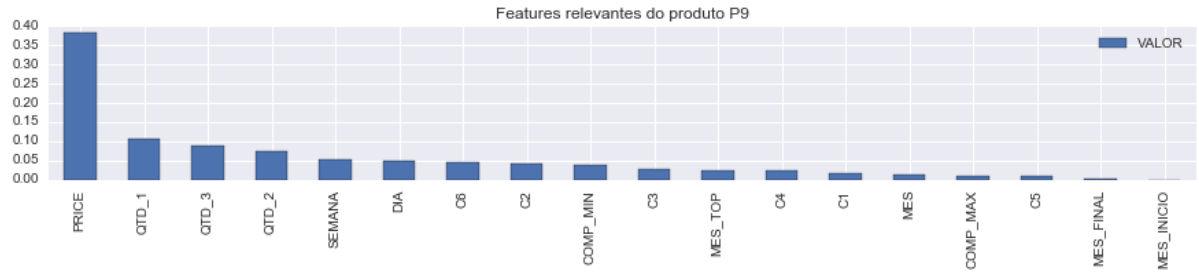
learning_rate = 0.1, max_depth = 2, gamma = 0.0, min_child_weight = 1

Gráfico comparativo no backtesting



Produto P9

Ranking de features mais relevantes



Features selecionadas

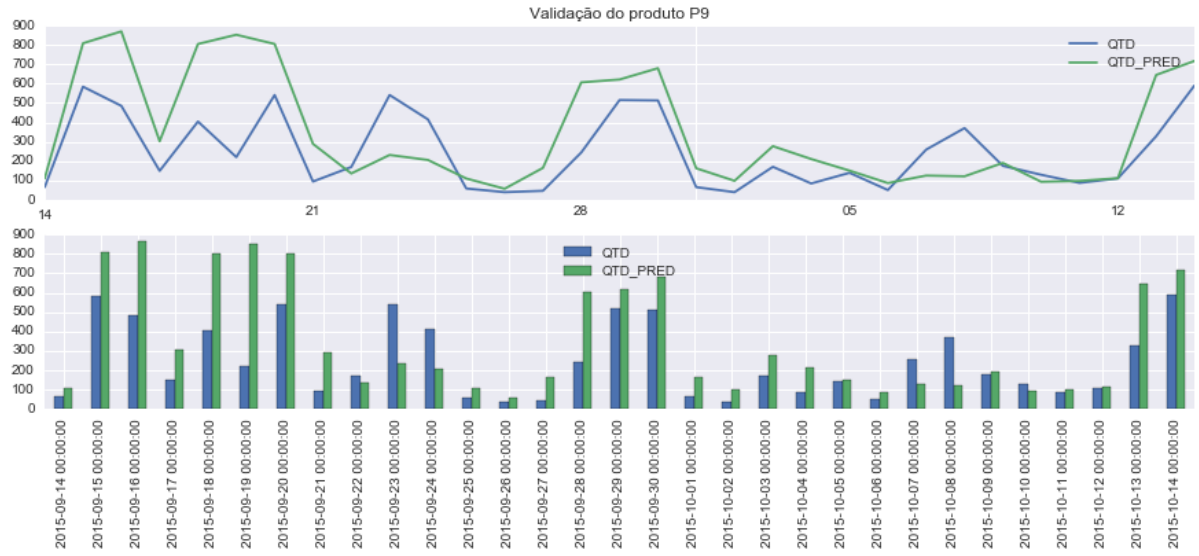
PRICE, QTD_1, QTD_3, QTD_2, SEMANA, DIA, C6, C2, COMP_MIN, C3, MES_TOP, C4, C1, MES, COMP_MAX, C5

Ranking da relevância dos concorrentes: C6, C2, C3, C4, C1, C5

Parâmetros selecionados para o método XGBoost

learning_rate = 0.1, max_depth = 6, gamma = 0.0, min_child_weight = 5

Gráfico comparativo no backtesting



Considerações finais e melhorias

Visualizando os resultados, podemos perceber que a curva de previsão tende a andar próximo com a curva de realizado, com exceção de P3, P6 e P4. Nestes casos com bastante diferença podemos usar técnicas que melhor se encaixam nas séries.

É possível melhorar a modelagem? Sim e bastante. Por questão de tempo e esforço foi desenvolvido superficialmente em todas as etapas e processos. Citarei abaixo alguns pontos para explorações e melhorias.

Diferentes modelos

Existem modelos mais específicos para séries temporais, por exemplo: ARIMA, Croston, Auto-ARIMA (R), etc. Além de fornecerem intervalos de confiança, eles são padrões de mercado para séries temporais.

Seleção de outliers

Uso de pacotes do sklearn para detecção de outliers.

Criação de features

Poderia ter criado features com uso de auto regressivo, eles são bons para series sazonais. Mas se for usar o ARIMA por exemplo, são serão necessários criar essas features.

Seleção de features

O critério de seleção de feature foi utilizado de forma mais simples. É importante selecionar quais features são melhoras se trabalharem juntas, se existem features que prejudicam as outras, etc.

Adição de feriados

Será que feriados fazem as vendas aumentarem? É bom ter uma exploração de dados neste assunto.

Cross-validations

O cross-validation escolheu registros de uma forma aleatórios, como se trata de séries temporais, o ideal é treinar os dados de acordo com a linha do tempo e testando o resultado com as métricas.

Diferentes métricas

Poderia ser usado as métricas WMAPE/MAPE, pois são padrões de mercado em séries temporais.

Overfiting

Criar estratégias de evitar overfiting.