

Smart BackOrders

**Análise e Predição de Pedidos Pendentes
(BackOrders) utilizando Modelos de
Inteligência Artificial**

**João Luiz de Souza Torres
Novembro de 2018**

I. Definição

Visão geral do projeto

Pedidos Pendentes (BackOrders)

BackOrders de produtos é um problema comum da cadeia de suprimentos, sendo uma preocupação importante na Gestão de Inventário de Produtos.

Definição de BackOrder: Pedido ou compromisso não efetivado ou cumprido. Um pedido pendente é uma demanda imediata de determinado item cujo estoque é insuficiente para satisfazer tal demanda.

Existem diversos fatores que contribuem para a ocorrência de Backorders. Eis alguns:

- Falha de gestão
- Falta de comunicação
- Incapacidade de fabricar itens suficientes para atender a demanda
- Falta de dados para prever com precisão a demanda
- Ausência de sistema informatizado que auxilie
- Estratégia da empresa.

O problema é amplamente estudado e debatido, dentro e fora do Brasil, visando a criação de soluções que possam minimizá-lo. [1] [2] [3] [6]

Outro fato “curioso”, é que a importância do problema pode também ser constatada pela quantidade de vezes em que seu entendimento é requisitado em anúncios de vagas de trabalho relacionados à área de logística. [4] [5]

Contextualização:

Conceito de BackOrders	Produtos que estão sem estoque temporariamente (*), mas é permitido ao cliente gerar um pedido a ser atendido com estoque futuro, ou ainda, venda com estoque atual negativo.
(*) % BackOrders	Importante KPI (<i>Key Performance Indicator</i>) na Logística.
Dilema	Forte demanda <i>versus</i> planejamento de estoque subotimizado.
Algumas Questões	O cliente pode esperar? (pedidos em atraso, pedidos pendentes no fornecedor, perda de vendas, baixa no índice de satisfação do cliente (NPS), dentre outros fatores).

Áreas de Negócio onde é relevante:

Produção	Programação, controle de estoque, planejamento agregado.
Comercial e Marketing	Alocação da força de vendas, promoções, lançamento de novos produtos.
Finanças	Investimento na fábrica, em equipamentos, planejamento orçamentário.
Pessoal	Planejamento de mão de obra, contratações, demissões.

Alguns números sobre Backorders no Brasil:



"Análise de rupturas na cadeia farmacêutica"

"A análise dos mais de 3.000.000 de pedidos das farmácias para os distribuidores mostrou que 14% dos pedidos apresentavam algum tipo de ruptura, fazendo com que 8,2% do valor do pedido não fosse faturado, gerando com isso uma perda anual no faturamento na ordem de R\$ 2 bilhões." - Pesquisa ILOS - Aché, ABRAFARMA, BD, Fidelize e Plannera (2015)).

Percepção do risco da ocorrência de Backorders nas operações:

Tabela 1 - Risco total da probabilidade – Gerentes versus funcionários

Fatores de risco	Probabilidade de provocar o desabastecimento (%)						Risco total
	Gerentes			Funcionários			
	Baixa	Média	Alta	Baixa	Média	Alta	
Back order	0,0	14,3	85,7	0,0	11,1	88,9	38,6
Transporte	28,6	14,3	57,1	5,6	19,4	75,0	22,6
Logística	0,0	14,3	85,7	0,0	22,2	77,8	34,4
Tempo de entrega	14,3	57,1	28,6	33,3	38,9	27,8	12,2
Sistema fora do ar	71,4	28,6	0,0	88,9	11,1	0,0	11,6
Fatores totais percebidos	22,9	25,7	51,4	25,6	20,6	53,9	16,6

Tabela 2 - Risco total do impacto – Gerentes versus funcionários

Fatores de risco	Impacto ou dano quando ocorre o desabastecimento (%)						Risco total
	Gerentes			Funcionários			
	Baixo	Médio	Alto	Baixo	Médio	Alto	
Back order	0,0	14,3	85,7	0,0	11,1	88,9	38,6
Transporte	42,9	14,3	42,9	5,6	25,0	69,4	16,5
Logística	0,0	28,6	71,4	2,8	11,1	86,1	31,8
Tempo de entrega	57,1	14,3	28,6	44,4	25,0	30,6	9,8
Sistema fora do ar	57,1	42,9	0,0	91,7	8,3	0,0	9,9
Fatores totais percebidos	31,4	22,9	45,7	28,9	16,1	55,0	15,3

"Inovando a gestão do risco de ruptura na cadeia logística automotiva"

Anais do V SINGEP - São Paulo - SP - Brasil - 20, 21 e 22/11/2016

VI Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade (VI SINGEP)

Sugestão de Compras Backorder

1. A funcionalidade backorder tem como objetivo fazer uma sugestão de compras para peças que estão em pedido e não possuem estoque. O diferencial desta sugestão é que as peças serão reservadas automaticamente no momento em que forem lançadas no estoque, para garantir que sejam utilizadas, exclusivamente, no pedido que deu origem à sugestão.
2. Para utilizar esta funcionalidade insira um novo pedido com uma peça sem estoque.
3. Clique no botão Confirmar para finalizar a inclusão do pedido.
4. Em seguida, altere o mesmo pedido, pois as opções de Backorder só estão disponíveis na alteração.
5. Acesse a opção de **sugestão de compras backorder** em Ações Relacionadas\Sug. Compras BO.
6. Selecione a forma de pedido referente ao backorder para a sugestão e confirme.

Como a Inteligência Artificial pode auxiliar na resolução do problema?

A partir dos fatos já ocorridos, utilizando os dados disponíveis, aplicar modelos e algoritmos que possam estabelecer uma “previsão” sobre a probabilidade de nova ocorrência.

A proposta deste trabalho é contribuir com a análise preditiva de Backorders, apresentando-a como uma ferramenta auxiliar ao processo de tomada de decisão nas empresas.

Visa especialmente, uma mudança de postura: REATIVA x PRÓ-ATIVA, atuando diretamente nos índices de satisfação de clientes, com a consequente melhoria nas vendas.

Trabalhando para identificar peças em risco de atraso antes do evento, então o negócio tem tempo para reagir.

Este é o objetivo do projeto “Smart BackOrders: Análise e Predição de Pedidos Pendentes (BackOrders) utilizando Modelos de Inteligência Artificial”.

Para a realização da tarefa utilizamos um conjunto de dados disponibilizado na plataforma Kaggle [7], utilizada para a prática de ciência de dados e machine learning.

O arquivo de dados (Dataset) que será utilizado para análise e predição contém os dados históricos para as 8 semanas anteriores à semana que estamos tentando prever.

Descrição do problema

Para que possamos chegar ao objetivo desejado, ou seja, a partir da análise de um conjunto de dados (fatos já ocorridos), conseguirmos realizar a predição de novos fatos (ocorrência de Backorders ou pedidos pendentes), faz-se necessário que analisemos o conjunto de dados existente.

Essa análise é dividida em etapas:

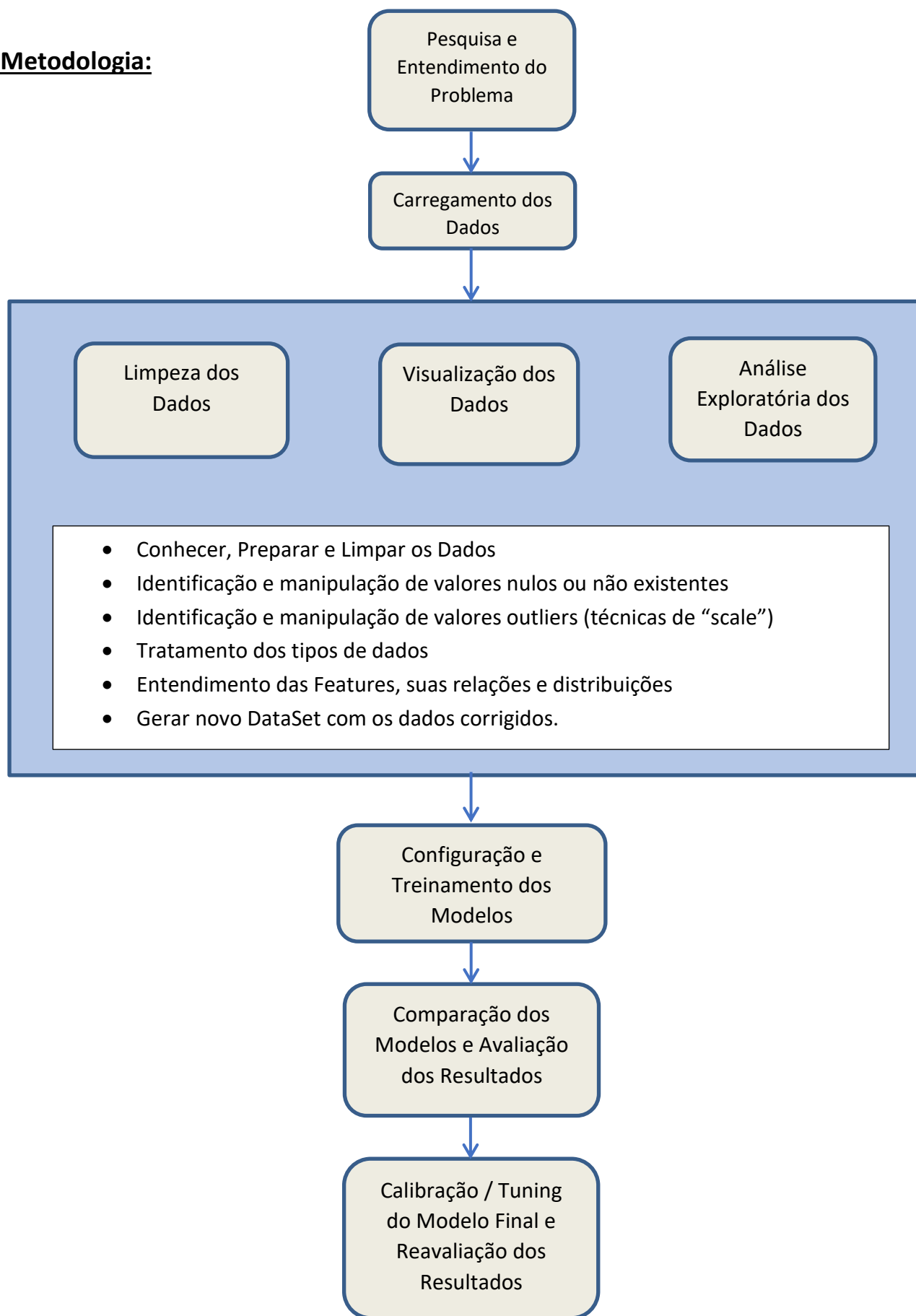
- Conhecer, Preparar e Limpar os Dados
- Identificação e manipulação de valores nulos ou não existentes
- Identificação e manipulação de valores outliers (análise de valores discrepantes através de técnicas de “scale”)
- Tratamento dos tipos de dados
- Entendimento das Features, suas relações e distribuições
- Gerar novo DataSet (conjunto de dados) com os dados corrigidos.

Com base nas características ou informações disponibilizadas no conjunto de dados, poderemos fazer nossas análises e predições:



Visando facilitar a visualização das etapas e tarefas, apresentamos abaixo um esquema:

Metodologia:



Métricas

Durante o processo de criação de um modelo de machine learning é necessário medir a qualidade do modelo de acordo com o objetivo da tarefa. [8]

Existem funções matemáticas que nos ajudam a avaliar a capacidade de erro e acerto dos nossos modelos.

Esse conjunto de funções quando aplicadas, nos fornecem as métricas de avaliação dos modelos.

Tão importante quanto saber escolher um bom modelo, é saber escolher a métrica correta para decidir qual é o melhor entre eles.

Existem métricas mais simples, outras mais complexas, algumas que funcionam melhor para datasets com determinadas características, ou outras personalizadas de acordo com o objetivo final do modelo.

Nosso objetivo pode ser descrito como uma "Classificação" binária (ou seja, dois valores: Sim/Não). Nas tarefas de "Classificação" buscamos prever qual é a categoria a que uma amostra de dados pertence como, por exemplo, determinar "se um pedido vai para BackOrder ou Não".

No nosso projeto, utilizaremos algumas métricas:

Precisão Geral (Accuracy)

$$Precisão\ Geral = \frac{P}{P + N}$$

Esta é a métrica mais simples. É basicamente o número de acertos (positivos) dividido pelo número total de exemplos. Ela deve ser usada em datasets com a mesma proporção de exemplos para cada classe. Veremos mais adiante que este não será o nosso caso.

Em problemas com classes desproporcionais, ela causa uma falsa impressão de bom desempenho. Por exemplo, num dataset em que 80% dos exemplos pertençam a uma classe, só de classificar todos os exemplos naquela classe já se atinge uma precisão de 80%, mesmo que todos os exemplos da outra classe estejam classificados incorretamente. [8]

F1 Score

$$F1 = \frac{2 * precisão * recall}{precisão + recall}$$

O F1 Score é uma média harmônica entre precisão (que, apesar de ter o mesmo nome, não é a mesma citada acima) e recall. Temos a seguir as definições destes dois termos.

Ela é muito boa quando você possui um dataset com classes desproporcionais, e o seu modelo não emite probabilidades. Isso não significa que não possa ser usada com

modelos que emitem probabilidades, tudo depende do objetivo de sua tarefa de machine learning.

Em geral, quanto maior o F1 score, melhor. [8]

Precisão (Precision)

$$Precisão = \frac{PV}{PV + FP}$$

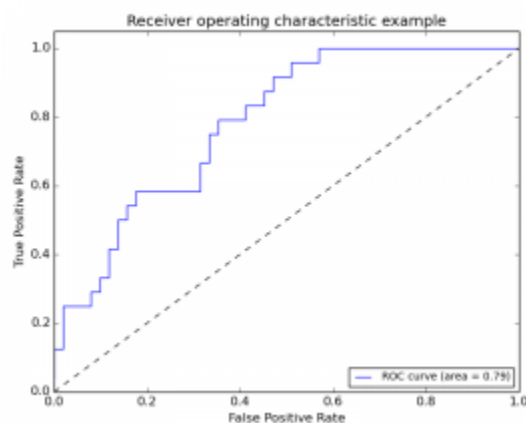
Número de exemplos classificados como pertencentes a uma classe, que realmente são daquela classe (positivos verdadeiros), dividido pela soma entre este número, e o número de exemplos classificados nesta classe, mas que pertencem a outras (falsos positivos). [8]

Recall

$$Recall = \frac{PV}{P}$$

Número de exemplos classificados como pertencentes a uma classe, que realmente são daquela classe, dividido pela quantidade total de exemplos que pertencem a esta classe, mesmo que sejam classificados em outra. No caso binário, positivos verdadeiros divididos por total de positivos. [8]

AUC - Area Under the ROC Curve



Esta é uma métrica interessante para tarefas com classes desproporcionais. Nela, mede-se a área sob uma curva formada pelo gráfico entre a taxa de exemplos positivos, que realmente são positivos, e a taxa de falsos positivos.

Uma das vantagens em relação ao F1 Score, é que ela mede o desempenho do modelo em vários pontos de corte, não necessariamente atribuindo exemplos com probabilidade maior que 50% para a classe positiva, e menor, para a classe negativa. [8]

II. Análise

Exploração dos dados

O conjunto de dados utilizado contém dois arquivos:

Dataset de Treinamento: “Kaggle_Training_Dataset.csv” (1.6 milhões de registros).

Dataset de Testes/Validação: “Kaggle_Test_Dataset.csv” (241.8 mil registros).

Temos abaixo o significado dos campos do conjunto de dados disponibilizado.

sku	chave de identificação de produto (Random ID for the product)
national_inv	estoque atual (Current inventory level for the part)
lead_time	tempo de transporte do fornecedor (Transit time for product - if available)
in_transit_qty	quantidade em transporte no momento (Amount of product in transit from source)
forecast_3_month	previsão de vendas próximos 3 meses (Forecast sales for the next 3 months)
forecast_6_month	previsão de vendas próximos 6 meses (Forecast sales for the next 6 months)
forecast_9_month	previsão de vendas próximos 9 meses (Forecast sales for the next 9 months)
sales_1_month	vendas do último mês (Sales quantity for the prior 1 month time period)
sales_3_month	vendas dos últimos 3 meses (Sales quantity for the prior 3 months time period)
sales_6_month	vendas dos últimos 6 meses (Sales quantity for the prior 6 months time period)
sales_9_month	vendas dos últimos 6 meses (Sales quantity for the prior 9 months time period)

min_bank	quantidade de estoque mínima recomendada (Minimum recommend amount to stock)
potential_issue	problema de origem para parte identificada (Source issue for part identified)
pieces_past_due	partes atrasadas da fonte (Parts overdue from source)
perf_6_month_avg	desempenho da fonte para o período anterior de 6 meses (Source performance for prior 6 month period)
perf_12_month_avg	desempenho da fonte para o período anterior de 12 meses (Source performance for prior 12 month period)
local_bo_qty	quantidade de pedidos atrasados (Amount of stock orders overdue)
deck_risk	indicativo de risco (Part risk flag)
oe_constraint	indicativo de risco (Part risk flag)
ppap_risk	indicativo de risco (Part risk flag)
stop_auto_buy	indicativo de risco (paralisar compra automática) (Part risk flag)
rev_stop	indicativo de risco (Part risk flag)
went_on_backorder	o produto entrou em “backorder”. Este é o valor alvo. (Product actually went on backorder. This is the target value.)

Abaixo temos a estrutura dos arquivos:

```
# informações gerais sobre o dataset
data_frame.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1693050 entries, 0 to 1693049
Data columns (total 23 columns):
sku                int64
national_inv       int64
lead_time          float64
in_transit_qty     int64
forecast_3_month   int64
forecast_6_month   int64
forecast_9_month   int64
sales_1_month      int64
sales_3_month      int64
sales_6_month      int64
sales_9_month      int64
min_bank           int64
potential_issue     object
pieces_past_due     int64
perf_6_month_avg    float64
perf_12_month_avg  float64
local_bo_qty       int64
deck_risk           object
oe_constraint       object
ppap_risk           object
stop_auto_buy       object
rev_stop           object
went_on_backorder  object
dtypes: float64(3), int64(13), object(7)
memory usage: 297.1+ MB
```

➔

```
# informações gerais sobre o dataset
data_frame.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241817 entries, 0 to 241816
Data columns (total 23 columns):
sku                241817 non-null int64
national_inv       241817 non-null int64
lead_time          227490 non-null float64
in_transit_qty     241817 non-null int64
forecast_3_month   241817 non-null int64
forecast_6_month   241817 non-null int64
forecast_9_month   241817 non-null int64
sales_1_month      241817 non-null int64
sales_3_month      241817 non-null int64
sales_6_month      241817 non-null int64
sales_9_month      241817 non-null int64
min_bank           241817 non-null int64
potential_issue     241817 non-null object
pieces_past_due     241817 non-null int64
perf_6_month_avg    241817 non-null float64
perf_12_month_avg  241817 non-null float64
local_bo_qty       241817 non-null int64
deck_risk           241817 non-null object
oe_constraint       241817 non-null object
ppap_risk           241817 non-null object
stop_auto_buy       241817 non-null object
rev_stop           241817 non-null object
went_on_backorder  241817 non-null object
dtypes: float64(3), int64(13), object(7)
memory usage: 42.4+ MB
```

Treino

Teste

A seguir apresentamos uma amostra dos dados. A amostra apresenta valores que deverão ser tratados (ou corrigidos) para que os mesmos não interfiram negativamente na análise desejada.

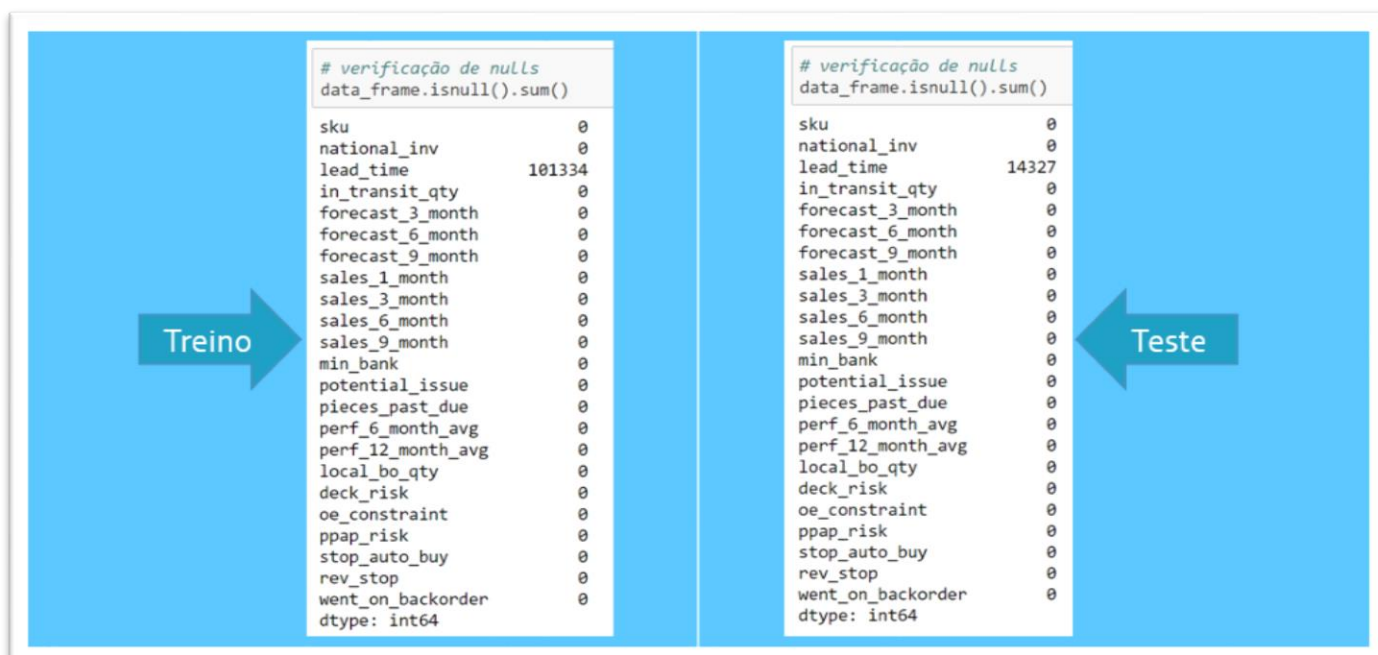
```
# Listando os dados do início do dataset
data_frame.head()
```

	sku	national_inv	lead_time	in_transit_qty	forecast_3_month	forecast_6_month	forecast_9_month	sales_1_r
0	2079834	23	NaN	0	0	0	0	0
1	2079835	2	NaN	0	0	0	0	0
2	2079836	6	NaN	0	0	0	0	0
3	2079837	87	NaN	0	0	0	0	0
4	2080337	315	NaN	0	0	0	0	0

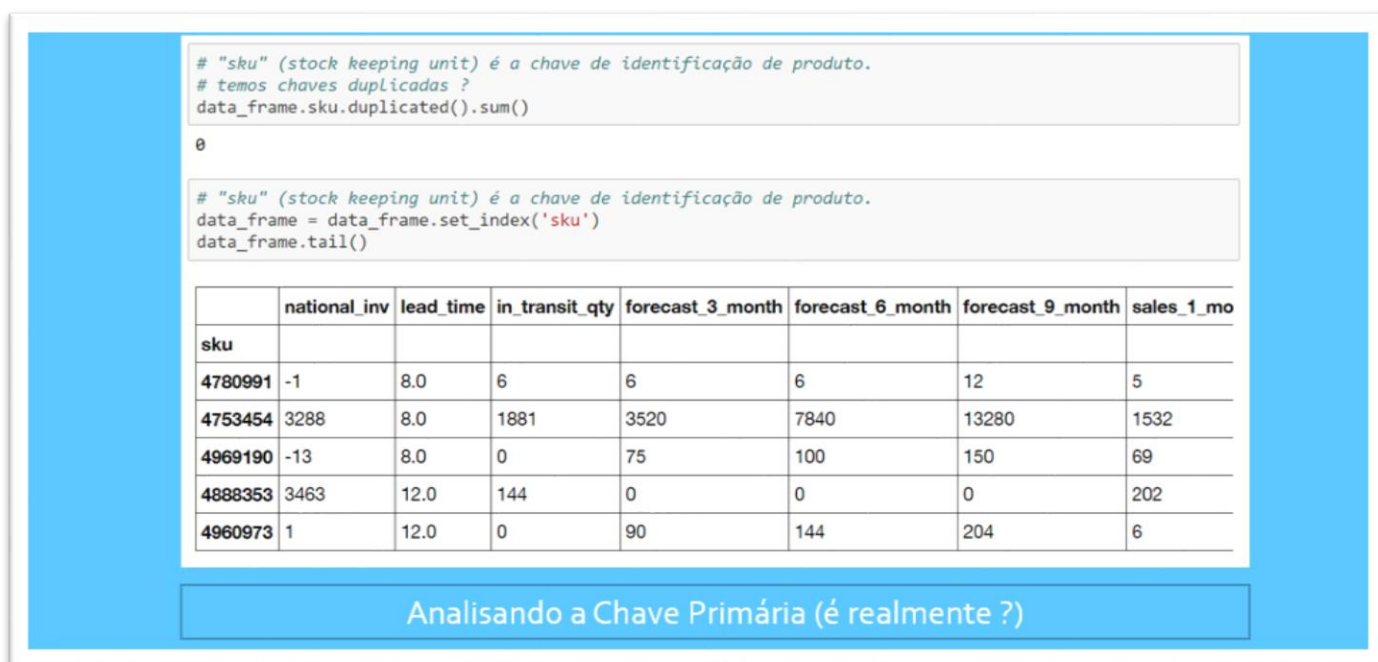
	perf_6_month_avg	perf_12_month_avg	local_bo_qty	deck_risk	oe_constraint	ppap_risk	stop_auto_buy	rev_stop
	-99.0	-99.0	0	Yes	No	No	Yes	No
	-99.0	-99.0	0	Yes	No	No	Yes	No
	-99.0	-99.0	0	Yes	No	No	Yes	No
	-99.0	-99.0	0	No	No	No	Yes	No
	-99.0	-99.0	0	Yes	No	No	No	No

Temos dados nulos "NaN", valores "negativos" e campos com "Yes/No"

Ilustrando ainda valores inválidos, podemos verificar abaixo que o campo “lead_time” (tempo de transporte do fornecedor) apresenta muitos valores nulos, o que como já dissemos é prejudicial às nossas análises.



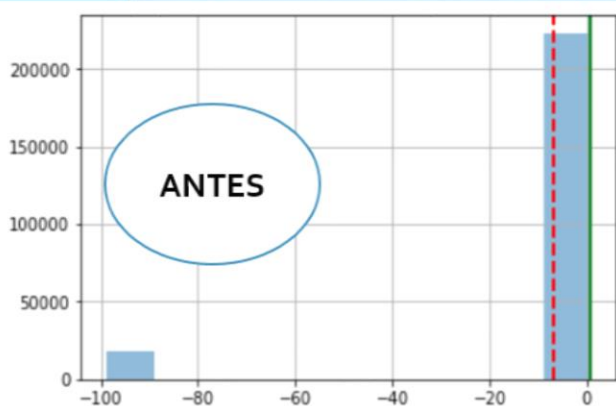
Outra análise relevante é a verificação se há algum campo que possa ser utilizado com identificador do registro (linha de dados). O campo identificador é conhecido como chave primária. No nosso caso, verificamos que o campo “sku” poderá ser nosso identificador.



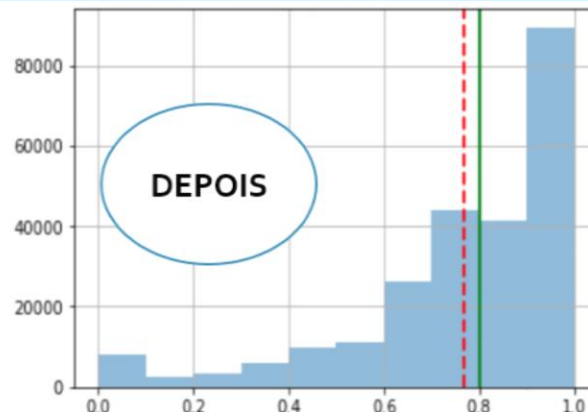
A verificação e o tratamento adequado para dados faltantes (ou nulos) deverá também ser realizado, visando também que os mesmos não atrapalhem as análises.

Tratamento para dados faltantes, inválidos ou nulos

- "lead_time", "perf_6_month_avg", "perf_12_month_avg" contêm dados faltantes, inválidos ou nulos. (NaN or -99.00)
- usar "sklearn.preprocessing.Imputer" para preencher com novos valores
 - <http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html>
- algumas possíveis estratégias: deleção, imputação (média, mediana, regressão linear), under/over sampling.



Mediana para perf_6_month_avg: 0.8
Média para perf_6_month_avg : -6.893688078180515



Mediana para perf_6_month_avg : 0.8
Média para perf 6 month avg : 0.7691854204369831

Os valores do tipo Yes/No deverão ser convertidos em valores numéricos: 1/0. Veja abaixo nova amostra de dados:

```
# Valores "Yes" / "No" -> transformar em "1" / "0"
# colunas com tipo de dados "object", Lembrando que "sku"(também "object") é a chave de identificação
# de produto, assim sendo,
# não será modificado

#visualização: "Yes" / "No"
data_frame[object_cols].tail()
```

	potential_issue	deck_risk	oe_constraint	ppap_risk	stop_auto_buy	rev_stop	went_on_backorder
sku							
5228278	No	Yes	No	No	Yes	No	No
5228279	No	Yes	No	No	Yes	No	No
5228280	No	Yes	No	Yes	Yes	No	No
5228281	No	Yes	No	No	Yes	No	No
5228282	No	Yes	No	No	Yes	No	No

	potential_issue	deck_risk	oe_constraint	ppap_risk	stop_auto_buy	rev_stop	went_on_backorder
sku							
5228278	0	1	0	0	1	0	0
5228279	0	1	0	0	1	0	0
5228280	0	1	0	1	1	0	0
5228281	0	1	0	0	1	0	0
5228282	0	1	0	0	1	0	0

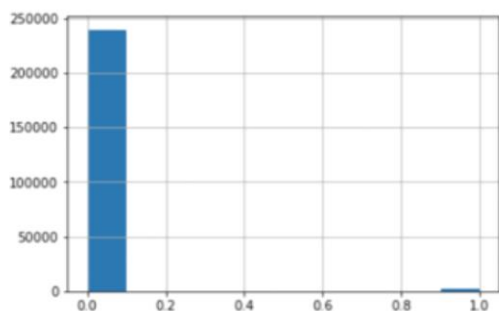
Valores Yes / No => Valores: 1 / 0

Abaixo temos a demonstração do desbalanceamento da nossa classe target (objetivo). Neste caso, como dissemos na seção Métricas, a métrica *Precisão Geral (Accuracy)* não deverá ser utilizada, por causar uma falsa impressão de bom desempenho. No nosso caso, já apresentaria uma “falsa” taxa de acerto de pelo menos 99%.

```
# analisando a proporção de backorder (SIM/NÃO)
```

```
print("Proporção backorder = NÃO, Classe 0: {}".format(100*data_frame.went_on_backorder.value_counts()[0]/len(data_frame.went_on_backorder)))  
print("Proporção backorder = SIM, Classe 1: {}".format(100*data_frame.went_on_backorder.value_counts()[1]/len(data_frame.went_on_backorder)))  
data_frame.went_on_backorder.hist()  
plt.show()
```

```
Proporção backorder = NÃO, Classe 0: 99.02860427513367%  
Proporção backorder = SIM, Classe 1: 0.9713957248663245%
```



Classes Target
Desbalanceadas

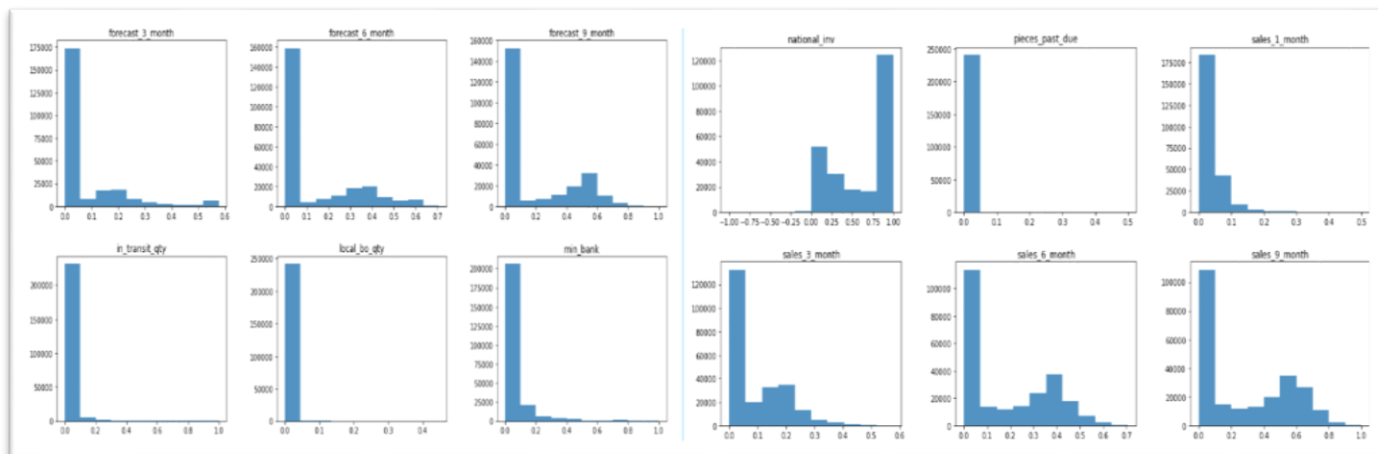
Nota: Classes (target) muito desbalanceadas. Acurária não será uma boa métrica de validação.

Análise do Balanceamento das Classes Target (went_on_backorder)

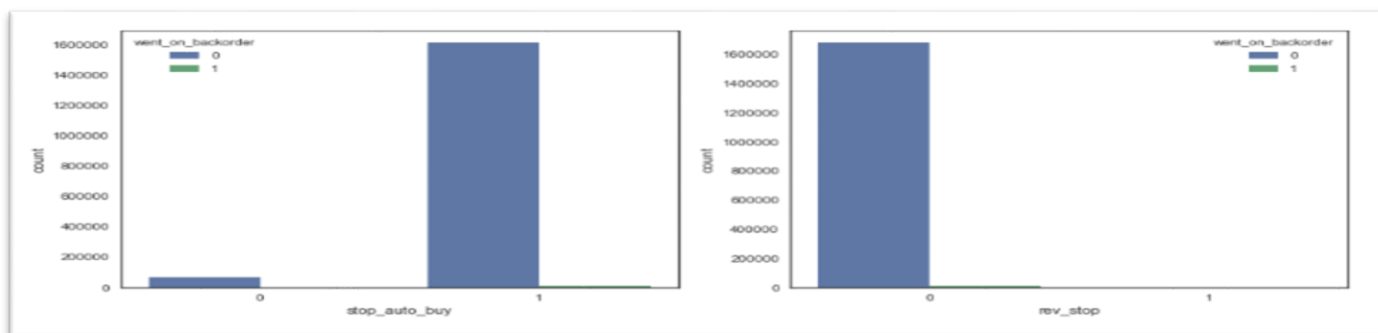
Visualização exploratória

Ao realizar análises exploratórias, algumas características importantes e/ou curiosas se revelaram.

- Analisando as *Features* de Vendas (passadas e previsões) e Estoque Mínimo Recomendado versus Estoque Atual, verificamos um possível problema, pois temos um bom nível de vendas (passadas e previstas), porém o estoque mínimo recomendado está com valores baixos (muitos com “0”). Tal característica é fator relevante para a ocorrência de Backorder.



- Outro fator observado é que a configuração de “Ligar” ou “Desligar” a *Feature* “Compra Automática”, interfere diretamente na geração ou não de Backorder.



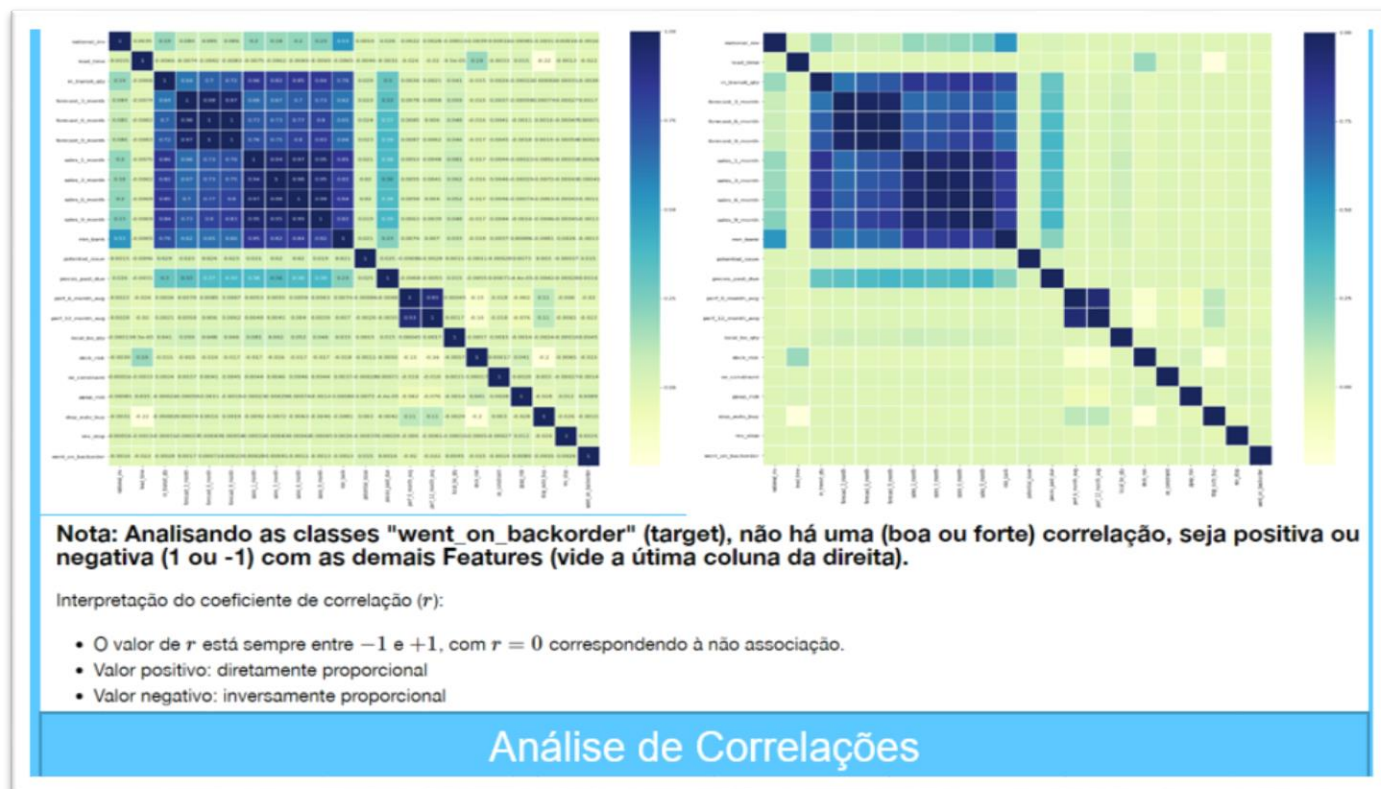
Durante as análises exploratórias, além de dados inválidos, procuramos também compreender como as informações se relacionam, caso as relações existam.

Foram utilizados mapas de calor para analisar correlações.

Interpretação do coeficiente de correlação (r): [9]

- O valor de r está sempre entre -1 e $+1$, com $r=0$ correspondendo à não associação.
 - Valor positivo: diretamente proporcional
 - Valor negativo: inversamente proporcional
- 0.9 para mais ou para menos indica uma correlação muito forte.
- 0.7 a 0.9 positivo ou negativo indica uma correlação forte.
- 0.5 a 0.7 positivo ou negativo indica uma correlação moderada.
- 0.3 a 0.5 positivo ou negativo indica uma correlação fraca.
- 0 a 0.3 positivo ou negativo indica uma correlação desprezível.

Vide abaixo os mapas de calor gerados visando análise das correlações.



Alguns detalhes:

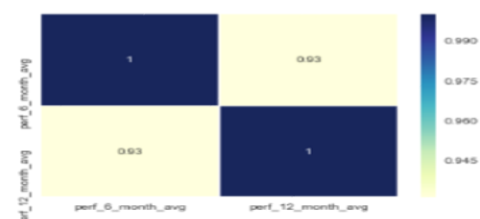
Nota: Sobre o HeatMap acima, as Features muito correlacionadas terão notas altas em Features_Importance ao final ?

```
# Analisando: "perf_6_month_avg" e "perf_12_month_avg"
perf_correlation_dataframe = data_frame[['perf_6_month_avg', 'perf_12_month_avg']]
perf_correlation = perf_correlation_dataframe.corr()
perf_correlation
```

	perf_6_month_avg	perf_12_month_avg
perf_6_month_avg	1.000000	0.931038
perf_12_month_avg	0.931038	1.000000

```
# Analisando: "perf_6_month_avg" e "perf_12_month_avg"
sns.heatmap(perf_correlation, annot=True, linewidths=.5, cmap="YlGnBu")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22077asf0e>
```



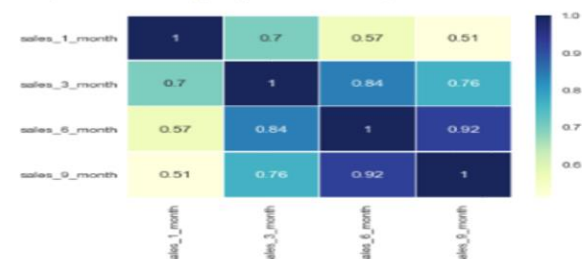
Detalhando algumas *Features* do HeatMap

	sales_1_month	sales_3_month	sales_6_month	sales_9_month
sales_1_month	1.000000	0.698478	0.567297	0.510173
sales_3_month	0.698478	1.000000	0.838094	0.763764
sales_6_month	0.567297	0.838094	1.000000	0.923772
sales_9_month	0.510173	0.763764	0.923772	1.000000

```
# Analisando: "sales_1_month", "sales_3_month", "sales_6_month" e "sales_9_month"
```

```
sns.heatmap(sales_correlation, annot=True, linewidths=.5, cmap="YlGnBu")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22076774208>
```



Nota: "sales_3_month", "sales_6_month" e "sales_9_month" apresentam forte correlação, diferentemente de "sales_1_month".

Detalhando algumas *Features* do HeatMap

Algoritmos e técnicas

Nesta seção, você deverá discutir os algoritmos e técnicas que você pretende utilizar para solucionar o problema. Você deverá justificar o uso de cada algoritmo ou técnica baseado nas características do problema e domínio do problema. Questões para se perguntar ao escrever esta seção:

- *Os algoritmos que serão utilizados, incluindo quaisquer variáveis/parâmetros padrão do projeto, foram claramente definidos?*
- *As técnicas a serem usadas foram adequadamente discutidas e justificadas?*
- *Ficou claro como os dados de entrada ou conjuntos de dados serão controlados pelos algoritmos e técnicas escolhidas?*

Benchmark

Nesta seção, você deverá definir claramente um resultado de referência (benchmark) ou limiar para comparar entre desempenhos obtidos pela sua solução. O raciocínio por trás da referência (no caso onde não é estabelecido um resultado) deve ser discutido. Questões para se perguntar ao escrever esta seção:

- *Algum resultado ou valor que funcione como referência para a medida de desempenho foi fornecido?*
- *Ficou claro como esse resultado ou valor foi obtido (seja por dados ou por hipóteses)?*

III. Metodologia

(aprox. 3-5 páginas)

Pré-processamento de dados

Nesta seção, você deve documentar claramente todos os passos de pré-processamento que você pretende fazer, caso algum seja necessário. A partir da seção anterior, quaisquer anormalidades ou características que você identificou no conjunto de dados deverão ser adequadamente direcionadas e tratadas aqui. Questões para se perguntar ao escrever esta seção:

- *Se os algoritmos escolhidos requerem passos de pré-processamento, como seleção ou transformações de atributos, tais passos foram adequadamente documentados?*
- *Baseado na seção de **Exploração de dados**, se existiram anormalidade ou características que precisem ser tratadas, elas foram adequadamente corrigidas?*
- *Se não é necessário um pré-processamento, foi bem definido o porquê?*

Implementação

Nesta seção, o processo de escolha de quais métricas, algoritmos e técnicas deveriam ser implementados para os dados apresentados deve estar claramente documentado. Deve estar bastante claro como a implementação foi feita, e uma discussão deve ser elaborada a respeito de quaisquer complicações ocorridas durante o processo. Questões para se perguntar ao escrever esta seção:

- *Ficou claro como os algoritmos e técnicas foram implementados com os conjuntos de dados e os dados de entrada apresentados?*
- *Houve complicações com as métricas ou técnicas originais que acabaram exigindo mudanças antes de chegar à solução?*
- *Houve qualquer parte do processo de codificação (escrita de funções complicadas, por exemplo) que deveriam ser documentadas?*

Refinamento

Nesta seção, você deverá discutir o processo de aperfeiçoamento dos algoritmos e técnicas usados em sua implementação. Por exemplo, ajuste de parâmetros para que certos modelos obtenham melhores soluções está dentro da categoria de refinamento. Suas soluções inicial e final devem ser registradas, bem como quaisquer outros resultados intermediários significativos, conforme o necessário. Questões para se perguntar ao escrever esta seção:

- *Uma solução inicial foi encontrada e claramente reportada?*
- *O processo de melhoria foi documentado de forma clara, bem como as técnicas utilizadas?*
- *As soluções intermediárias e finais foram reportadas claramente, conforme o processo foi sendo melhorado?*

IV. Resultados

(aprox. 2-3 páginas)

Modelo de avaliação e validação

Nesta seção, o modelo final e quaisquer qualidades que o sustentem devem ser avaliadas em detalhe. Deve ficar claro como o modelo final foi obtido e por que tal modelo foi escolhido. Além disso, algum tipo de análise deve ser realizada para validar a robustez do modelo e sua solução, como, por exemplo, manipular os dados de entrada ou o ambiente para ver como a solução do modelo é afetada (técnica chamada de análise sensível). Questões para se perguntar ao escrever esta seção:

- *O modelo final é razoável e alinhado com as expectativas de solução? Os parâmetros finais do modelo são apropriados?*
- *O modelo final foi testado com várias entradas para avaliar se o modelo generaliza bem com dados não vistos? -O modelo é robusto o suficiente para o problema? Pequenas perturbações (mudanças) nos dados de treinamento ou no espaço de entrada afetam os resultados de forma considerável?*
- *Os resultados obtidos do modelo são confiáveis?*

Justificativa

Nesta seção, a solução final do seu modelo e os resultados dela obtidos devem ser comparados aos valores de referência (benchmark) que você estabeleceu anteriormente no projeto, usando algum tipo de análise estatística. Você deverá também justificar se esses resultados e a solução são significativas o suficiente para ter resolvido o problema apresentado no projeto. Questões para se perguntar ao escrever esta seção:

- *Os resultados finais encontrados são mais fortes do que a referência reportada anteriormente?*
- *Você analisou e discutiu totalmente a solução final?*
- *A solução final é significativa o suficiente para ter resolvido o problema?*

V. Conclusão

(aprox. 1-2 páginas)

Foma livre de visualização

Nesta seção, você deverá fornecer alguma forma de visualização que enfatize uma qualidade importante do projeto. A visualização é de forma livre, mas deve sustentar de forma razoável um resultado ou característica relevante sobre o problema que você quer discutir. Questões para se perguntar ao escrever esta seção:

- *Você visualizou uma qualidade importante ou relevante acerca do problema, conjunto de dados, dados de entrada, ou resultados?*
- *A visualização foi completamente analisada e discutida?*
- *Se um gráfico foi fornecido, os eixos, títulos e dados foram claramente definidos?*

Reflexão

Nesta seção, você deverá resumir os procedimentos desde o problema até a solução e discutir um ou dois aspectos do projeto que você achou particularmente interessante ou difícil. É esperado que você reflita sobre o projeto como um todo de forma a mostrar que você possui um entendimento sólido de todo o processo empregado em seu trabalho. Questões para se perguntar ao escrever esta seção:

- *Você resumiu inteiramente o processo que você utilizou neste projeto?*
- *Houve algum aspecto interessante do projeto?*
- *Houve algum aspecto difícil do projeto?*
- *O modelo e solução final alinham-se com suas expectativas para o problema, e devem ser usadas de forma geral para resolver esses tipos de problemas?*

Melhorias

Nesta seção, você deverá discutir como um aspecto da sua implementação poderia ser melhorado. Por exemplo, considere maneiras de tornar a sua implementação mais geral e o que precisaria ser modificado. Você não precisa fazer a melhoria, mas as possíveis soluções que resultariam de tais mudanças devem ser consideradas e comparadas/contrastadas com a sua solução atual. Questões para se perguntar ao escrever esta seção:

- *Existem melhorias futuras que podem ser feitas nos algoritmos ou técnicas que você usou neste projeto?*
- *Existem algoritmos ou técnicas que você pesquisou, porém não soube como implementá-las, mas consideraria usar se você soubesse como?*
- *Se você usou sua solução final como nova referência, você acredita existir uma solução ainda melhor?*

VI. Referências

Referências utilizadas

[1] BackOrder - Dicionário de Logística On-line

Revista LOGÍSTICA & SUPPLY CHAIN

<https://www.imam.com.br/logistica/dicionario-da-logistica/?pag=13&a=P>

[2] LogWeb - Indicadores de atendimento

<http://www.logweb.com.br/colunas/indicadores-de-atendimento/>

[3] Impact of Backorder on Supply Chain Performance - an Experimental Study

<https://www.sciencedirect.com/science/article/pii/S1474667016345785>

<https://doi.org/10.3182/20130619-3-RU-3018.00477>

[4] BNE - Banco Nacional de Empregos

<https://www.bne.com.br/curriculo-de-assistente-de-pcp-em-fortaleza-ce/10684485>

[5] Catho Empresas

<https://www.catho.com.br/buscar/curriculos/curriculo/18295544/>

[6] KPI - % of backorders: Percentage of unfulfilled orders.

<http://kpilibrary.com/kpis/of-backorders>

[7] Desafio Kaggle : “Can You Predict Product Backorders?” (disponibilizado em Abril / 2017)

www.kaggle.com/tiredgeek/predict-bo-trial (*)

(*) Obs: visando preservar os dados em caso de mudanças na estrutura do desafio, armazenado na plataforma Kaggle, os dados foram movidos e estão disponíveis no repositório Github do projeto, em formato ZIP devido à limitação imposta pelo repositório:

<https://github.com/TorresJLST/br-machine-learning/tree/master/projects/capstone/SmartBackorders/data>

ou

<https://bit.ly/2DQIE53>

[8] As Métricas Mais Populares para Avaliar Modelos de Machine Learning

<http://mariofilho.com/as-metricas-mais-populares-para-avaliar-modelos-de-machine-learning/>

[9] Coeficiente de correlação de Pearson

https://pt.wikipedia.org/wiki/Coeficiente_de_correlação_de_Pearson