



Trabalho Final – Turma 12

Caso de Uso: Olist

05/Jun/2020

Coordenadores:

Profª Drª Alessandra de Álvila Montini

Profª Dr. Adolpho Walter Pimazoni Canton

GRUPO 1:

- Alan Ferreirós



Agenda

- 1. Objetivo do Trabalho
- 2. Contextualização do Problema
- 3. Base de Dados
 - i. Bases originais
 - ii. Filtros
 - iii. Principais variáveis
- 4. Análise Exploratória de Dados
- 5. Modelagem com Estatística Tradicional
- 6. Modelagem com Inteligência Artificial
- 7. Conclusões

1. Objetivo do Trabalho

Trabalho Final | Grupo 1 - T12

O objetivo do trabalho é predizer o **prazo de entrega** das mercadorias compradas a partir da plataforma Olist.

Desta forma, a plataforma poderá apresentar **prazos de entrega mais competitivos**



Dados:

- A predição utilizará dados históricos de 2 anos (Out/2016 a Ago/2018)
- Uso de dados geográficos de clientes e fornecedores
- *Feature Engineering* a partir de comentários de usuários
 - Análise de sentimento
- Datas de envio fornecidas pelos vendedores





Vaso Min-Gui - Colecionador

(Cód.1234567890) ★★★★★

R\$ 1.234,56

12x de R\$ 102,88 s/ juros

Corra! Temos apenas 2 no estoque

R\$ 1.234,56 em até 12x de R\$ 102,88 s/ juros

R\$ 1.234,56 em até 24x de R\$ 51,44 s/ juros

Formas de parcelamento

Este produto é vendido por parceira.
O Acaso garante sua comodidade à entrega.



Calcular frete e prazo
69990-000

Prazo

29 dias úteis

Entrega

Frete

Convencional

R\$ 123,45

Prazo

29 dias úteis

2. Contextualização do Problema

Trabalho Final | Grupo 1 - T12

4

A plataforma tem apresentado **prazos de entrega muito mais longos** do que os reais. Quanto maior o prazo, maior o erro.

Prazo estimado (media)	Entrega (média)	Antecedência (prazo – entrega)
3	3	0
5	4	1
10	5	5
20	11	9
40	18	22

Como consequência, a plataforma pode estar em **desvantagem em relação a concorrentes** que prometam prazos menores, além de potencial **desistência de compra** pelos clientes.

Por outro lado, o algoritmo atual tem conseguido **evitar atrasos em 93%** dos casos.

Este trabalho irá propor modelos que **reduzam o prazo de entrega** exibido aos clientes, mas **mantendo o nível de atraso existente**.

3. Bases de Dados

Trabalho Final | Grupo 1 - T12

Bases fornecidas

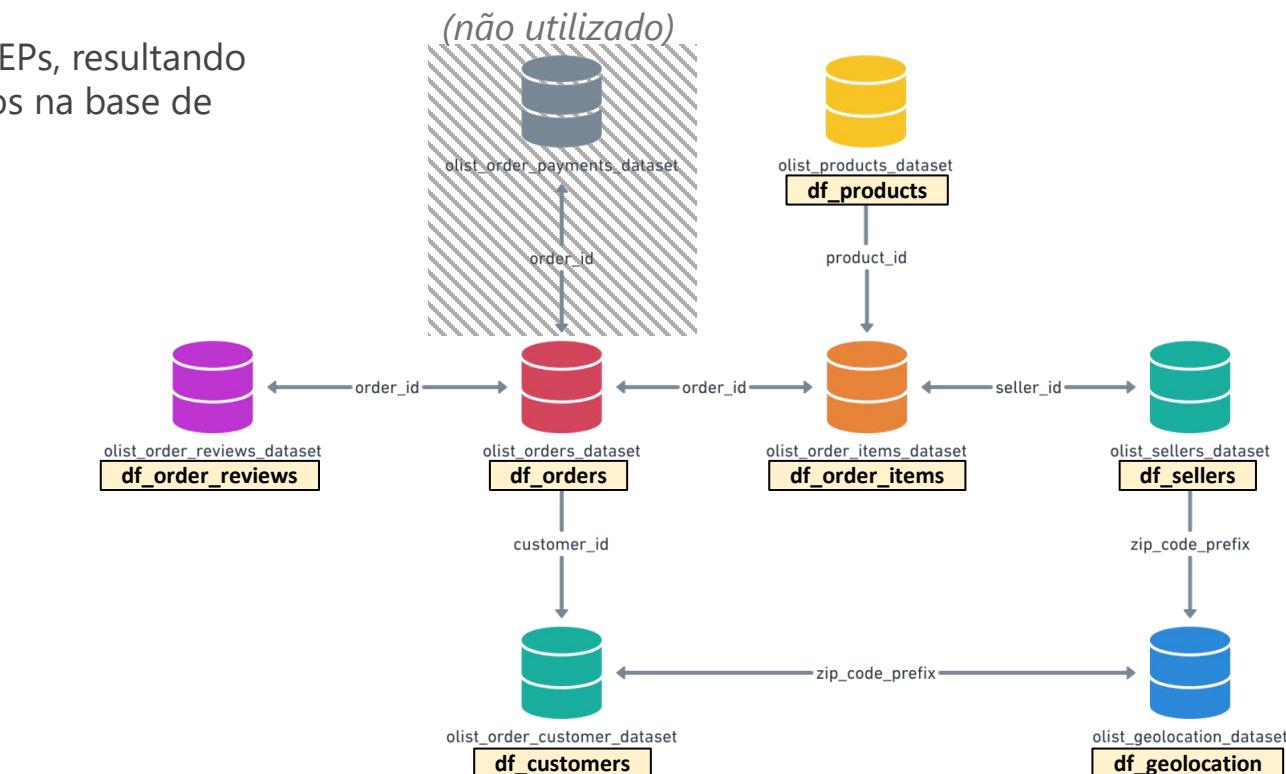
Bases fornecidas foram populadas a partir da base de Reviews. Isto naturalmente introduz uma limitação e um bias nos resultados, já que não temos dados de pedidos sem comentários de clientes.

Além disso, não temos os finais dos CEPs, resultando em muitos registros com CEP repetidos na base de geolocalização.

Período analisado:

- Out/2016 a Ago/2018 (2 anos)

Base	Registros
Order Reviews	100.000
Orders	99.441
Order Items	112.650
Products	32.951
Customers	99.441
Sellers	3.095
Geolocation	1.000.163 19.015 (CEP únicos)

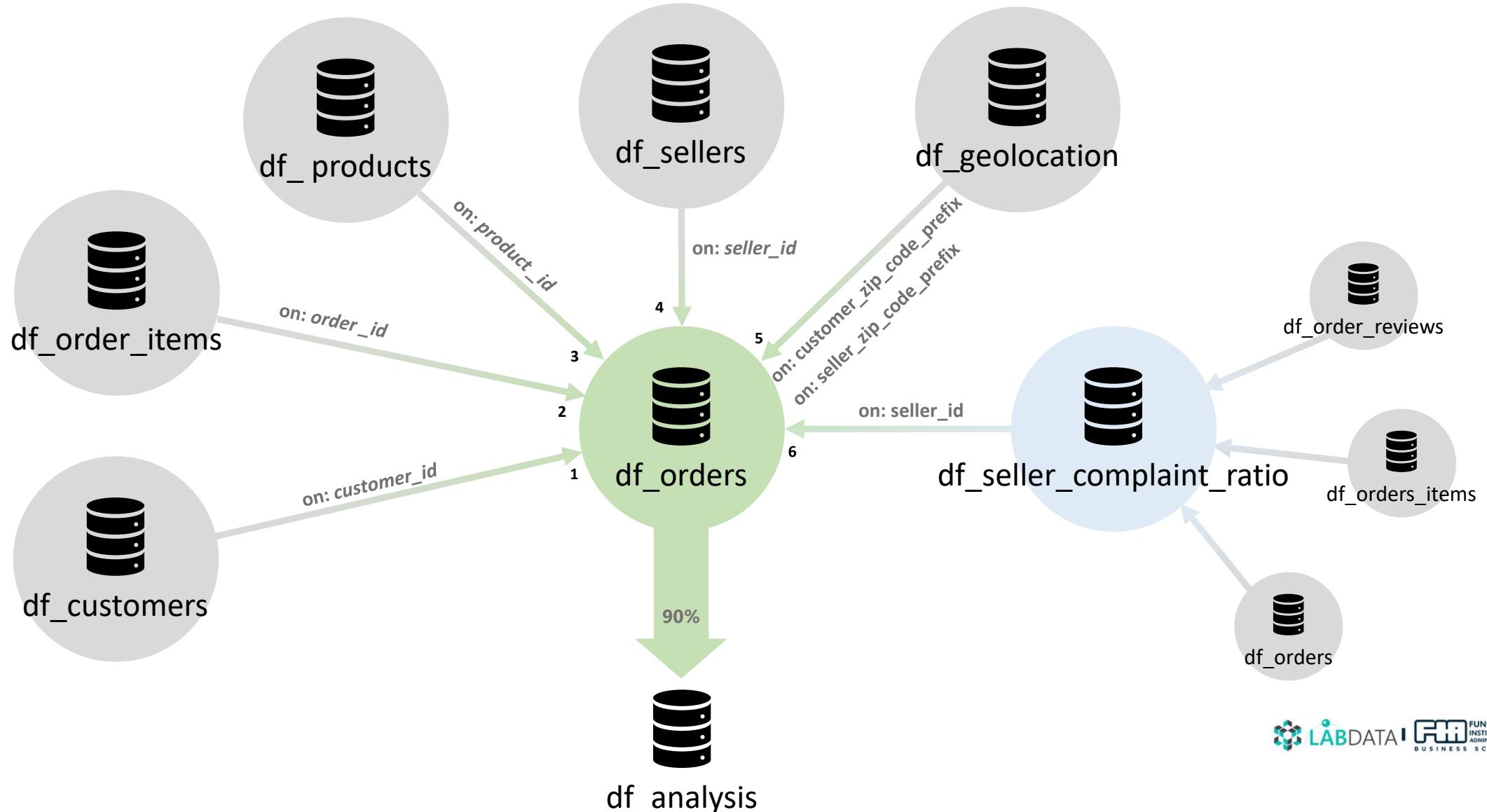


3. Bases de Dados – Construção do DataFrame de estudo

Trabalho Final | Grupo 1 - T12

6

O dataframe utilizado para o estudo foi criado a partir de operações de *inner join* a partir do dataframe df_orders



3.i. Base original

7



Visão da base

- Base de pedidos através da plataforma Olist

Filtros de inclusão

- Pedidos com somente 1 item

Quando há mais de 1 item, não é possível saber qual dos itens justifica a data de entrega do pedido

- Somente pedidos entregues

Filtros de exclusão

- Eliminação de CEPs repetidos
- Eliminação de pedidos sem data de entrega (target)

Período de Análise

- Out/2016 a Ago/2018



3.ii. Filtros

8



Base Original

99.441 pedidos, 1.000.163 CEPs

Base original

Base original, onde a maioria dos pedidos contêm apenas 1 item.

Quanto à base de geolocalização, ela possui 1 milhão de CEPs, muitos deles repetidos.



Tratamento da base de geolocalização

99.441 pedidos, 19.015 CEPs

Amostragem de CEPs

Foram escolhidos apenas 1 representante de cada CEP, de forma aleatória. A existência de CEPs repetidos dificultaria o processo de *inner join* das tabelas, replicando artificialmente o número de registros.



Pedidos com 1 item e com data de entrega

86.593 pedidos, 19.015 CEPs

Somente 1 item

Selecionou-se somente pedidos com um único item. Pedidos com mais de 1 item não permitem descobrir qual deles foi o gargalo para a entrega.

Também foram eliminados os pedidos sem data de entrega



Ajustes

86.593 pedidos, 19.015 CEPs

Ajustes na base

- Tratamento de valores faltantes.
 - CEPs inexistentes
 - Produtos sem características
- Tratamento de valores inválidos
 - CEPs inválidos
 - latitude/longitude inválidos
- Ajuste nos formatos das datas



Base Final – sem balanceamento

86.593 pedidos, 19.015 CEPs

Base final

Por se tratar de um problema de regressão, a base não precisou passar por processo de balanceamento.

Nota: Apesar disso, ensaiou-se balancear a base replicando os registros menos frequentes. O resultado piorou, devido à maior ponderação aos *outliers*.



3.iii Bases de Dados – Principais Variáveis

Trabalho Final | Grupo 1 - T12

9

Dados	Propósito
 Ids de Clientes	Identificar frequência de compra de produtos
 Geolocalização - Clientes <i>Cidade, Estado, Latitude, Longitude</i>	Calcular distância do vendedor e considerar dificuldades regionais de entrega.
 Ids de Vendedores	Identificar perfil de vendedores <ul style="list-style-type: none">- Número de categorias atendidas- Número de pedidos já enviados
 Geolocalização – Vendedores <i>Cidade, Estado, Latitude, Longitude</i>	Calcular distância do cliente e considerar dificuldades regionais de envio
 Data de aprovação do pedido	Levantamento de questões sazonais e tendências temporais
 Data limite para envio (<i>shipping limit</i>) (assumi que a informação estaria disponível no momento da compra)	Considerar os prazos já conhecidos pela logística do vendedor
 Dados do produto <i>Dimensões, peso, categoria, nome*, descrição*, fotos*</i>	Mapear se produtos com diferentes características afetam o prazo de entrega
 Comentários (<i>reviews</i>)	Criar indicador de taxa de atraso de um vendedor de acordo com a frequência de comentários negativos.
 Preço	Mapear se o valor do produto interfere no prazo de entrega.
 Valor do Frete	O valor do frete pode auxiliar a estimativa de dificuldade de entrega.

* apenas o tamanho e quantidade estavam disponíveis na base



Variável resposta:
Data de Entrega

Formato: Data
Tipo: Quantitativa

Por conveniência, a variável resposta foi substituída pelo número de dias para entregar:

days_to_deliver
Formato: inteiro ≥ 0
Tipo: Quantitativa

3.iii Bases de Dados – Principais Variáveis

Trabalho Final | Grupo 1 - T12



22 variáveis explicativas

1 variável resposta

- days_to_deliver
- Origem: df_order

order_delivered_customer_date



order_approved_at



=
days_to_deliver

Dados	Tipo	Origem
Ids de Clientes	customer_id: Categórica	df_customers
Geolocalização - Clientes <i>Cidade, Estado, Latitude, Longitude</i>	geolocation_city: Categórica geolocation_state: Categórica geolocation_lat: Quantitativa geolocation_lng: Quantitativa	df_geolocation (inner join com df_customers)
Ids de Vendedores	seller_id: Categórica	df_sellers
Geolocalização – Vendedores <i>Cidade, Estado, Latitude, Longitude</i>	geolocation_city: Categórica geolocation_state: Categórica geolocation_lat: Quantitativa geolocation_lng: Quantitativa	df_geolocation (inner join com df_sellers)
Data de aprovação do pedido	order_approved_at: Quantitativa	df_order
Data limite para envio (<i>shipping limit</i>)	shipping_limit_date: Quantitativa	df_order_items
Dados do produto <i>Dimensões, peso, categoria, nome</i>	product_category_name: Categórica product_name_length (sic): Quantitativa product_description_length (sic): Quantitativa product_photos_qty: Quantitativa product_weight_cm: Quantitativa product_height_cm: Quantitativa product_width_cm: Quantitativa	df_products
Comentários (reviews)	review_comment_message: Texto	df_order_reviews
Preço	price: Quantitativa	df_order_items
Valor do Frete	freight_value: Quantitativa	df_order_items

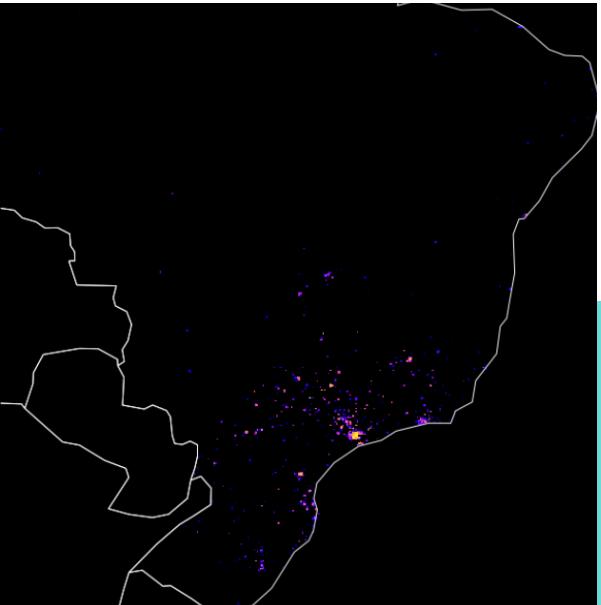
4. Análise exploratória

ANÁLISE EXPLORATÓRIA | UNIVARIADA

11

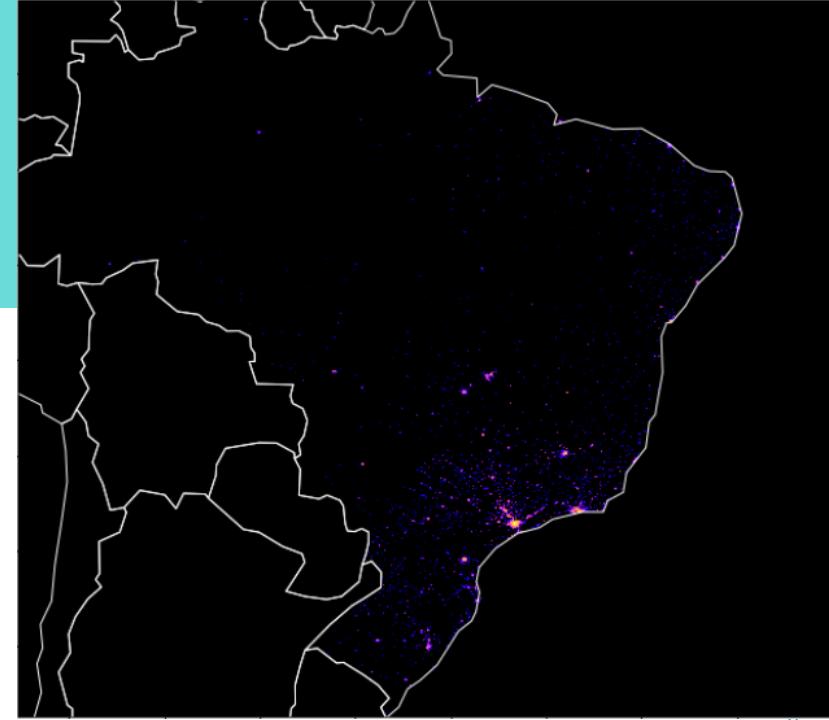
Cientes

- A maior parte dos clientes **realizaram uma única compra (97%)**
- Clientes estão distribuídos pelo Brasil mas se concentram na região **Sudeste (60%)**
- Em particular, os clientes se concentram no **estado de SP (36%)**, principalmente na cidade de **São Paulo (13%)**



Vendedores

- A maior parte dos vendedores **realizaram até 100 vendas (93%)**
- Vendedores se concentram na região **Sudeste (73%)**
- Em particular, os vendedores se concentram no **estado de SP (60%)**, principalmente na cidade de **São Paulo (22%)**
- A maioria **não tem reclamações de atraso** na entrega (96%)



4. Análise exploratória

ANÁLISE EXPLORATÓRIA | UNIVARIADA

12



Produtos

- Maioria dos produtos **só foram vendidos uma única vez (60%)**
- **Categorias são variadas.** Principais categorias de produtos:
 - BELEZA_SAÚDE (8%)
 - CAMA_MESA_BANHO (8%)
 - ESPORTE_LAZER (7%)
- Mais que 90% dos produtos **custam menos de R\$ 1.000,00**
- Mais que 90% dos produtos têm **frete menor que R\$ 50,00**
- Mais que 70% dos produtos pesam **menos que 5 kg**
- Aumento gradual do número de pedidos por mês:
 - Out-2016: **200** pedidos
 - Ago-2017: **3.500** pedidos
 - Ago-2018: **6.000** pedidos

Prazos

- Em geral, produtos são **entregues ao cliente em até 30 dias** após a compra
- Em geral, produtos chegam ao cliente em até **10 dias após entrega à transportadora**
- Em geral, a **estimativa de entrega é de até 40 dias**.
- **7% dos pedidos atrasam**
- **30% dos pedidos chegam pelo menos 2 semanas antes do prazo**



4. Análise exploratória

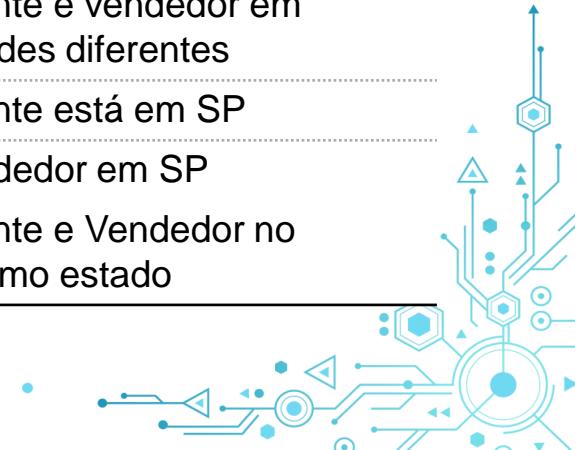
ANÁLISE EXPLORATÓRIA | UNIVARIADA - Ajustes

13

Categóricas – Grande número de categorias dificultaria o ajuste dos modelos

Solução: Estas variáveis foram substituídas por novas com apenas 2 categorias

Variáveis Categóricas	Número de categorias	Novas variáveis	Categoria 0	Categoria 1
seller_id	3095	seller_class	Vendeu no máximo 100 itens	Já vendeu mais que 100 itens
customer_id	99441	customer_class	Fez apenas um pedido	Já fez mais de um pedido
product_category_name	73	category_is_exclusive	Categoria é atendida por 50 ou mais vendedores	Categoria é vendida por poucos vendedores
		category_is_popular	Existem poucos pedidos desta categoria	Existem mais de 2000 pedidos nesta categoria
customer_geolocation_city	4119	customer_is_sao_paulo	Cliente fora de São Paulo	Cliente está em São Paulo
seller_geolocation_city	611	seller_is_sao_paulo	Vendedor fora de São Paulo	Vendedor em São Paulo
		is_same_city	Cliente e vendedor na mesma cidade	Cliente e vendedor em cidades diferentes
customer_geolocation_state	27	customer_is_sp	Cliente fora de SP	Cliente está em SP
seller_geolocation_state	23	seller_is_sp	Vendedor fora de SP	Vendedor em SP
		is_same_state	Cliente e Vendedor em estados diferentes	Cliente e Vendedor no mesmo estado



4. Análise exploratória

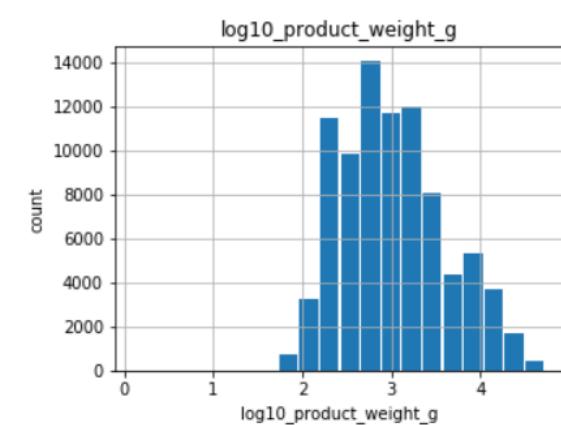
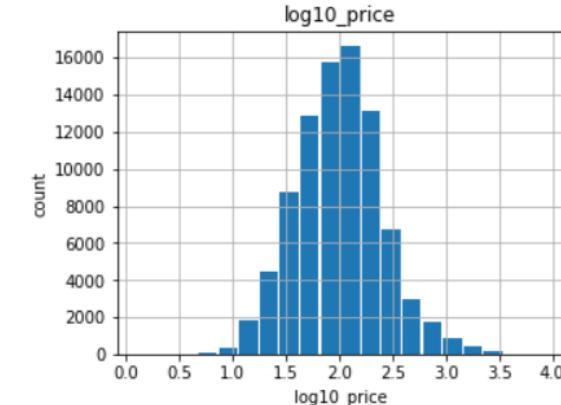
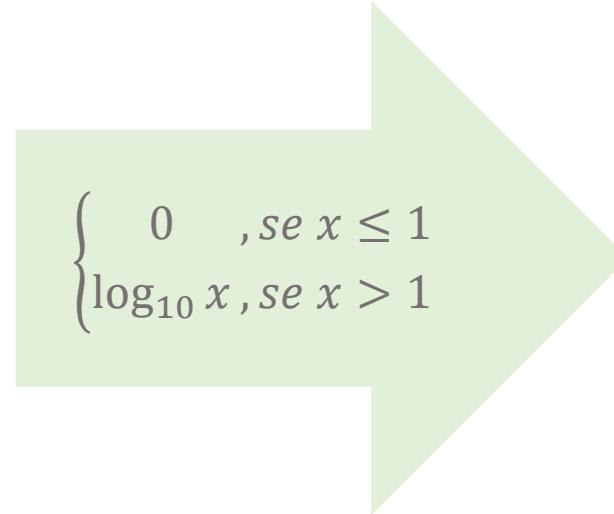
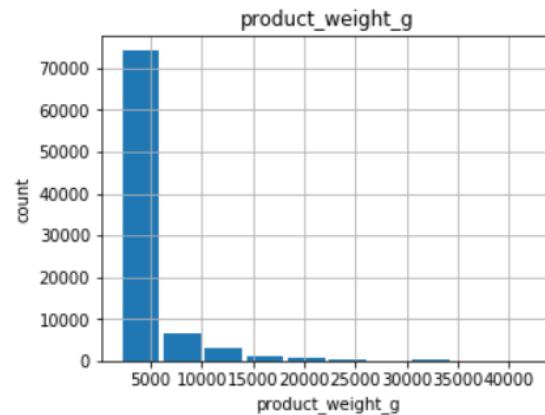
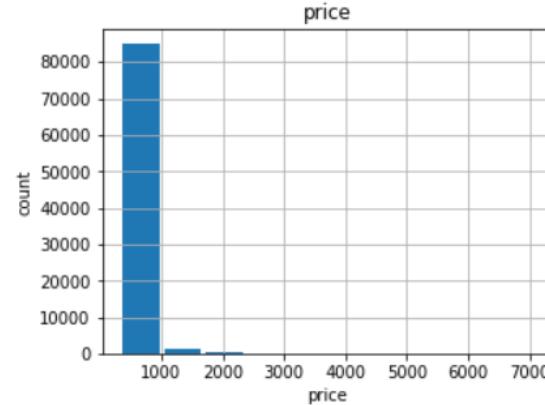
ANÁLISE EXPLORATÓRIA | UNIVARIADA - Ajustes

Quantitativas (Exceto datas, coordenadas e índice de reclamações) – **Valores se concentram em valores baixos.**

Solução: Estas variáveis foram transformadas para a **escala logarítmica**. Além de **harmonizar a distribuição**, tratar estas variáveis em relação à **ordem de grandeza** faz sentido para este estudo.

Os **outliers** também perdem a importância com esta transformação.

Exemplos:

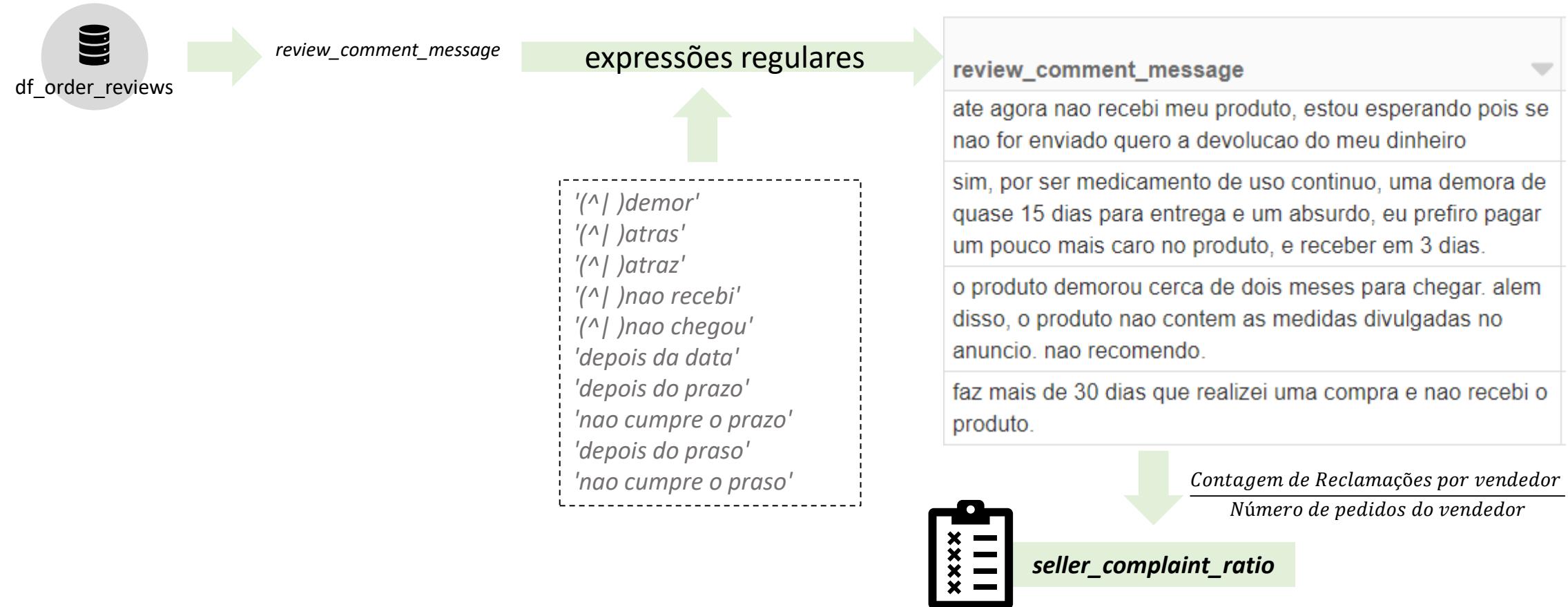


4. Análise exploratória

ANÁLISE EXPLORATÓRIA | UNIVARIADA - Ajustes

15

Índice de reclamações - Índice criado a partir dos comentários dos clientes para cada vendedor, através de **agrupamentos e expressões regulares** (considerando potenciais erros de ortografia, acentos, maiúsculos e minúsculos)

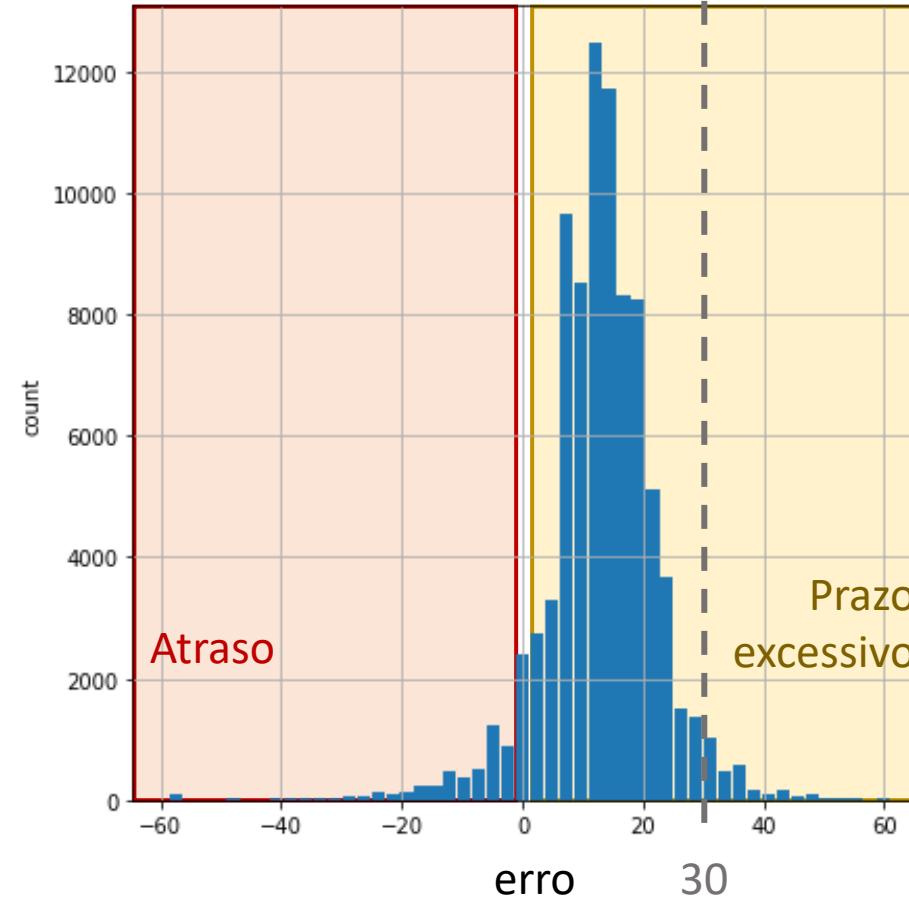
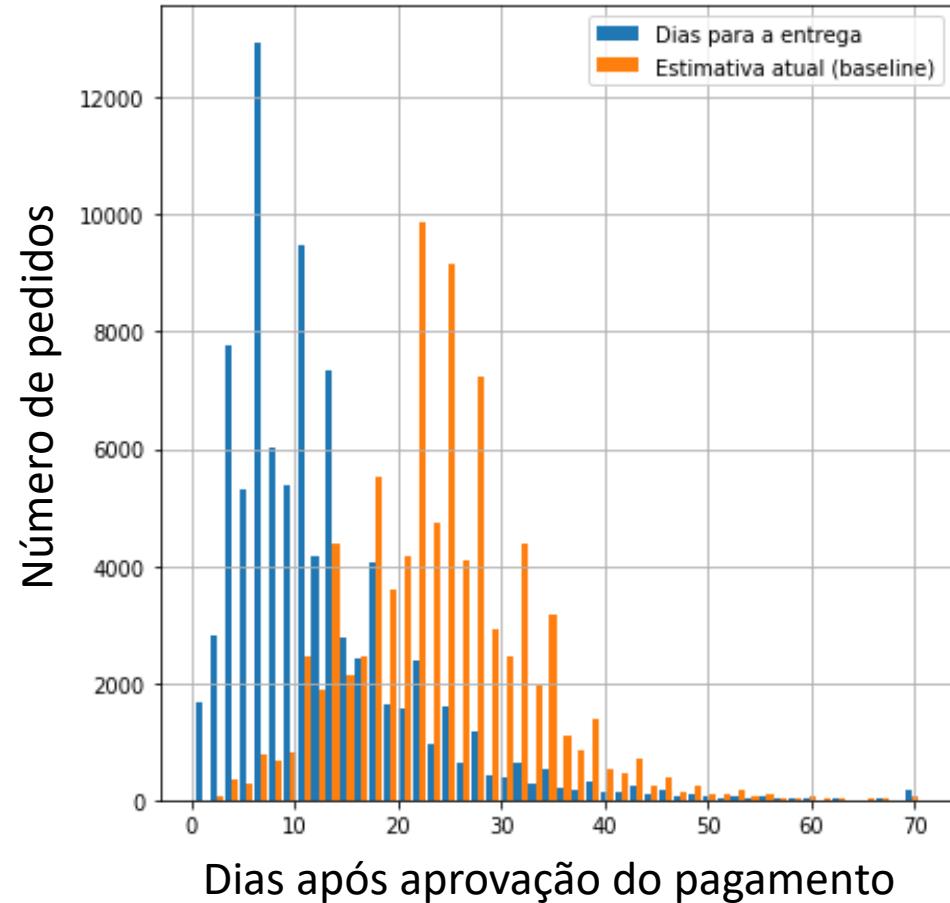


4. Análise exploratória

ANÁLISE EXPLORATÓRIA | UNIVARIADA

16

TARGET - Diferença relevante entre a **previsão existente** e a **entrega real**. A diferença pode chegar a mais de 30 dias.

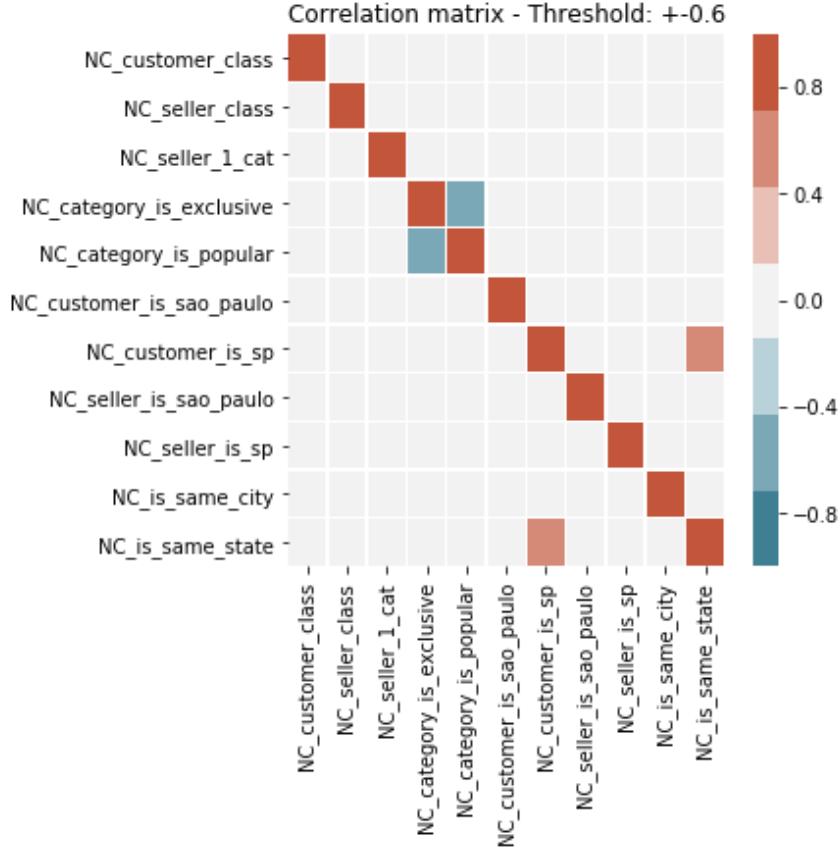


4. Análise exploratória

ANÁLISE EXPLORATÓRIA | BIVARIADA

17

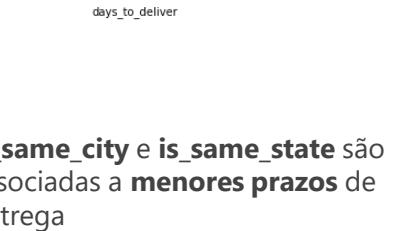
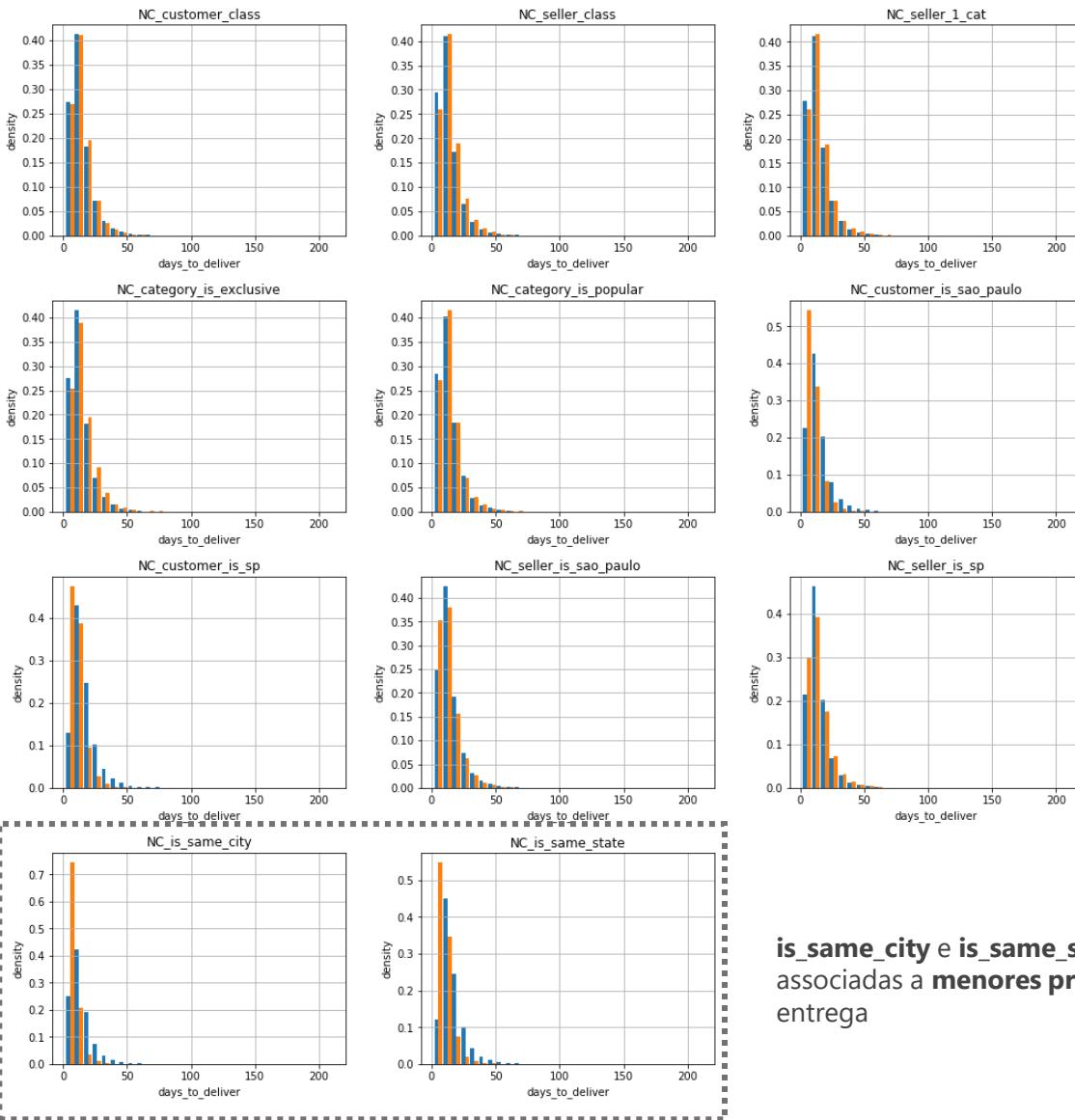
Categóricas



Correlações foram próximas ao *threshold*
→ Todas as variáveis foram mantidas
(ponto de atenção durante a modelagem)

@2020 LABDATA FIA. Copyright all rights reserved.

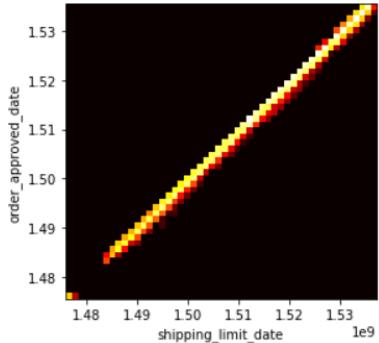
As variáveis categóricas têm pouca correlação com o target



4. Análise exploratória

ANÁLISE EXPLORATÓRIA | BIVARIADA

18



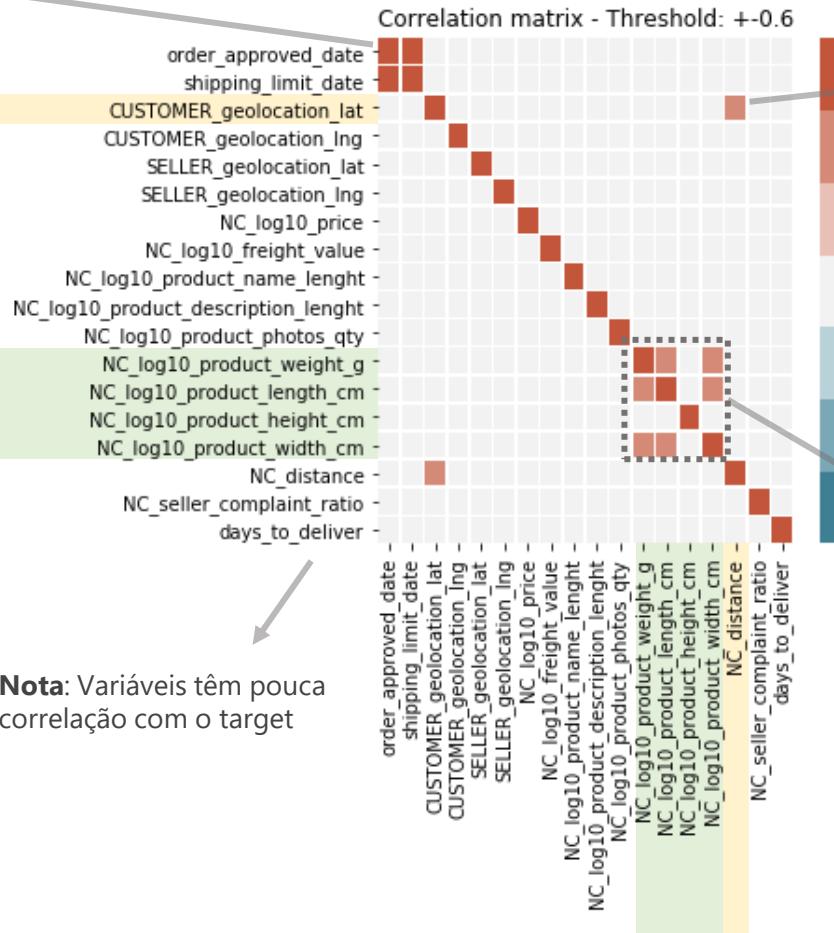
Data de **envio** tende a ser em uma data próxima à data de **compra**

Solução

Substituir por **days_for_shipping**

Número de dias até o envio tende a ser independente da data da compra

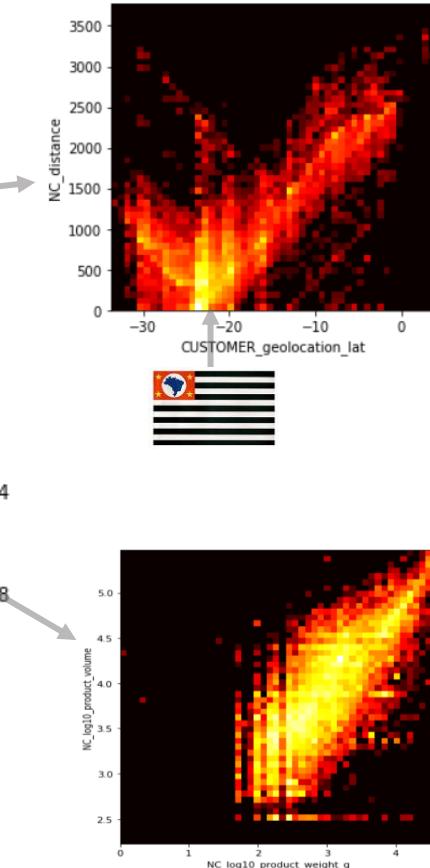
Quantitativas



Nota: Variáveis têm pouca correlação com o target

Correlações foram próximas ao *threshold*.

Todas as variáveis foram mantidas



Latitude reflete **distância** até São Paulo, onde está a maioria dos vendedores

Solução

Eliminar *customer latitude*

(redundante com variável de distância)

Volume tem 0.8 de correlação com o **peso**

Solução

Eliminar dimensões do produto

(redundante com o peso)



5. Modelagem Estatística Tradicional

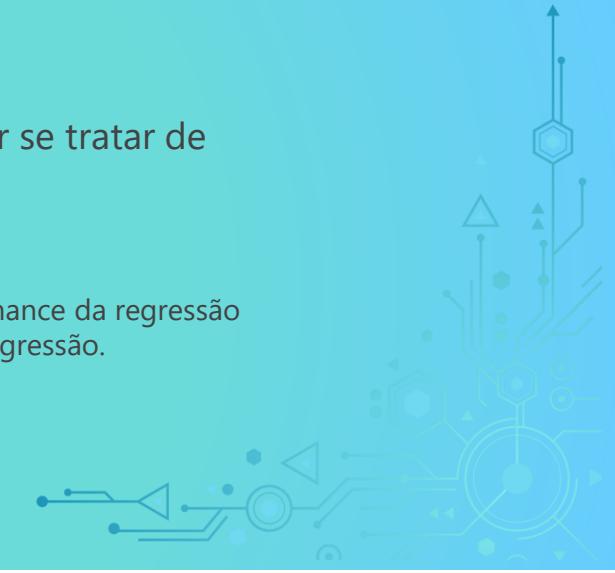
BALANÇAMENTO | BASES DE TREINO E TESTE

19



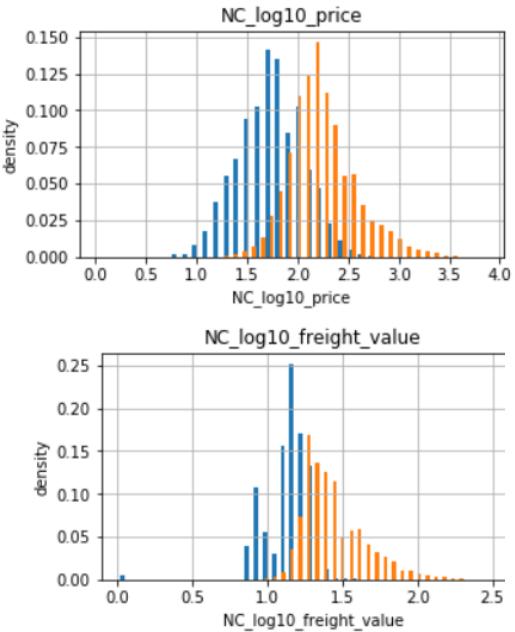
Tratamento das base de dados para modelagem

1. Balanceamento da resposta: Balanceamento não foi realizado, por se tratar de problema de regressão
2. Normalização: *StandardScaling* (Média=0, Desvio Padrão=1)
Nota: A normalização é mandatória para o agrupamento, mas não interfere na performance da regressão linear e da árvore de decisão, apesar de ajudar a entender melhor os coeficientes da regressão.
1. 70% aleatório para treino e 30% para teste
 - Treino: 60.798 pedidos
 - Teste: 25.795 pedidos



5. Agrupamento

MODELAGEM COM ESTATÍSTICA TRADICIONAL | KMEANS e PCA



Cluster 0: Produtos mais baratos, leves para clientes mais próximos



Ponto de atenção

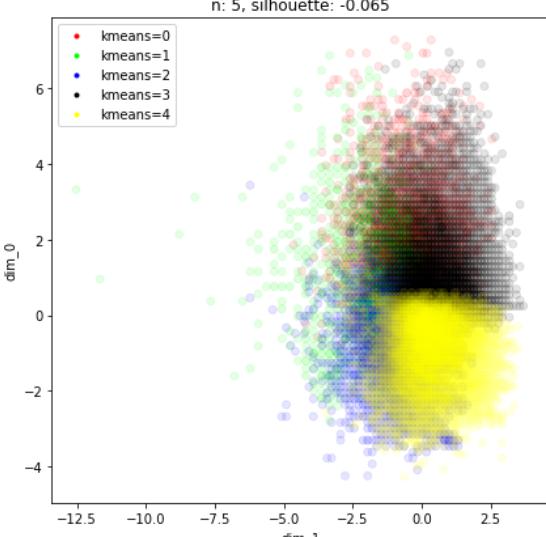
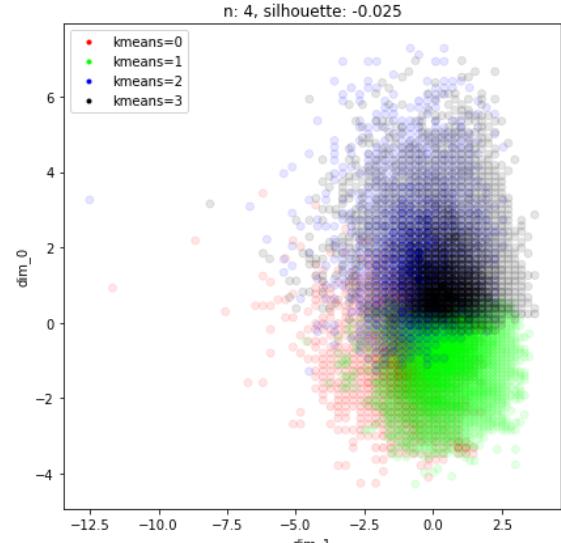
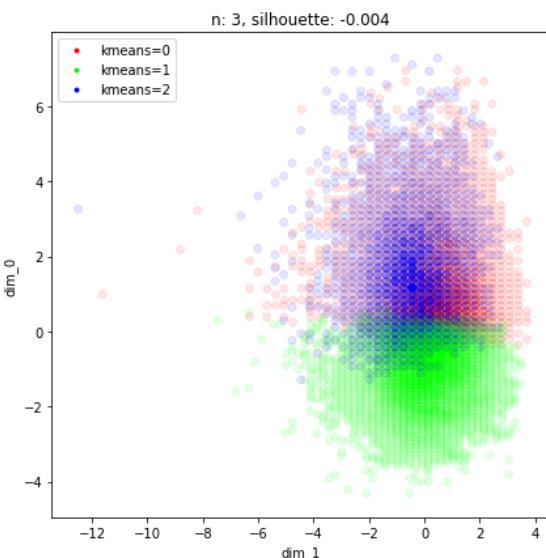
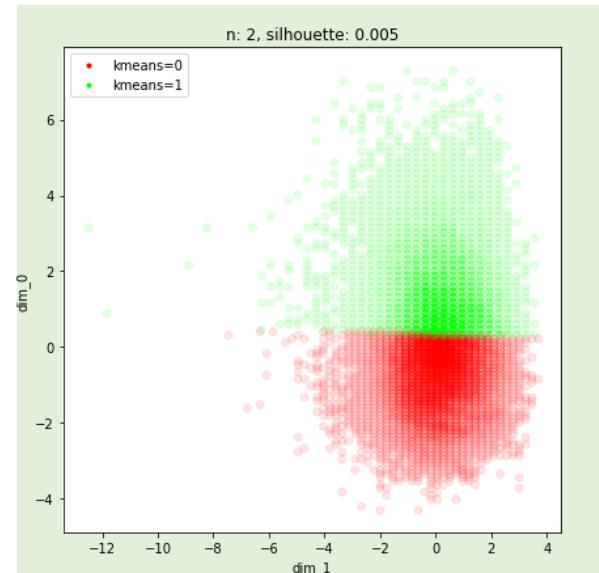
O valor da silhueta é baixo, o que reflete uma separação não muito clara entre os dois grupos

Cluster 1: Produtos mais caros, pesados para clientes mais distantes



Observação

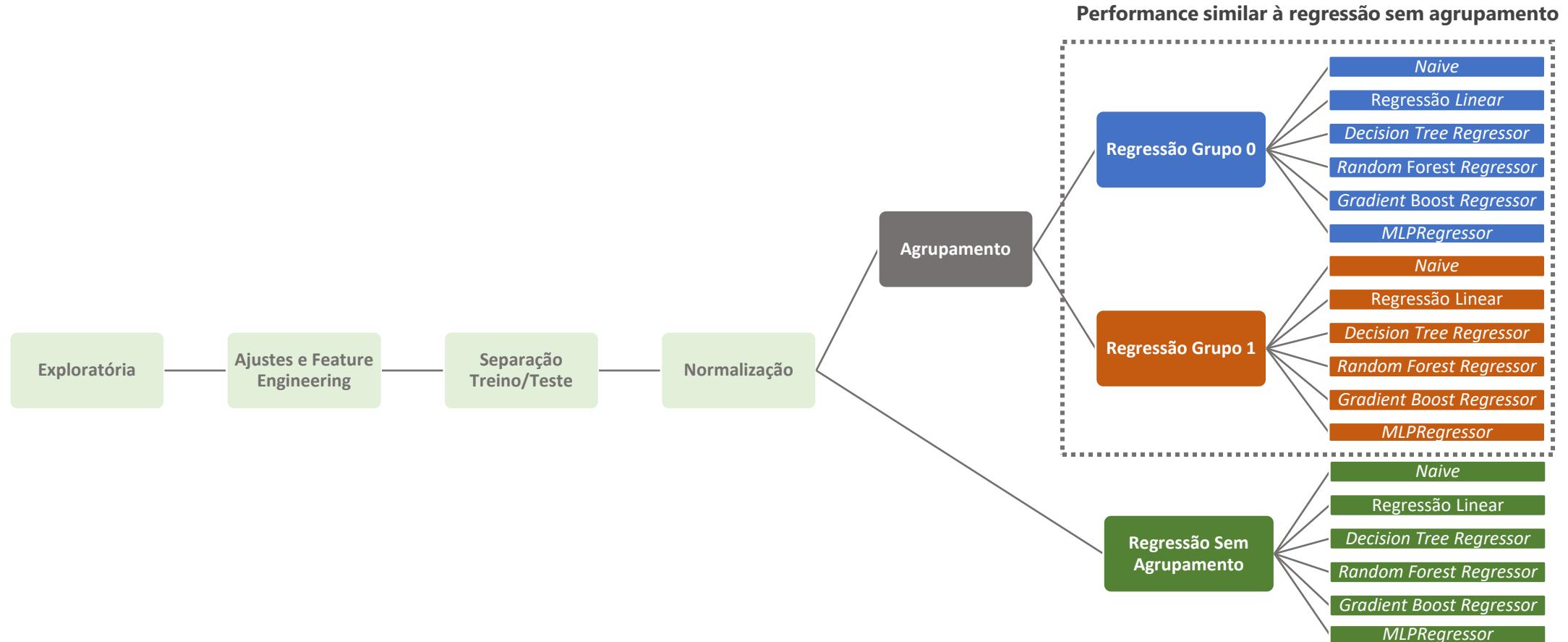
O KMEANS foi executado sem as variáveis de latitude/longitude, caso contrário ele simplesmente filtraria o estado de São Paulo



5. Regressão

MODELAGEM COM ESTATÍSTICA TRADICIONAL | Resumo

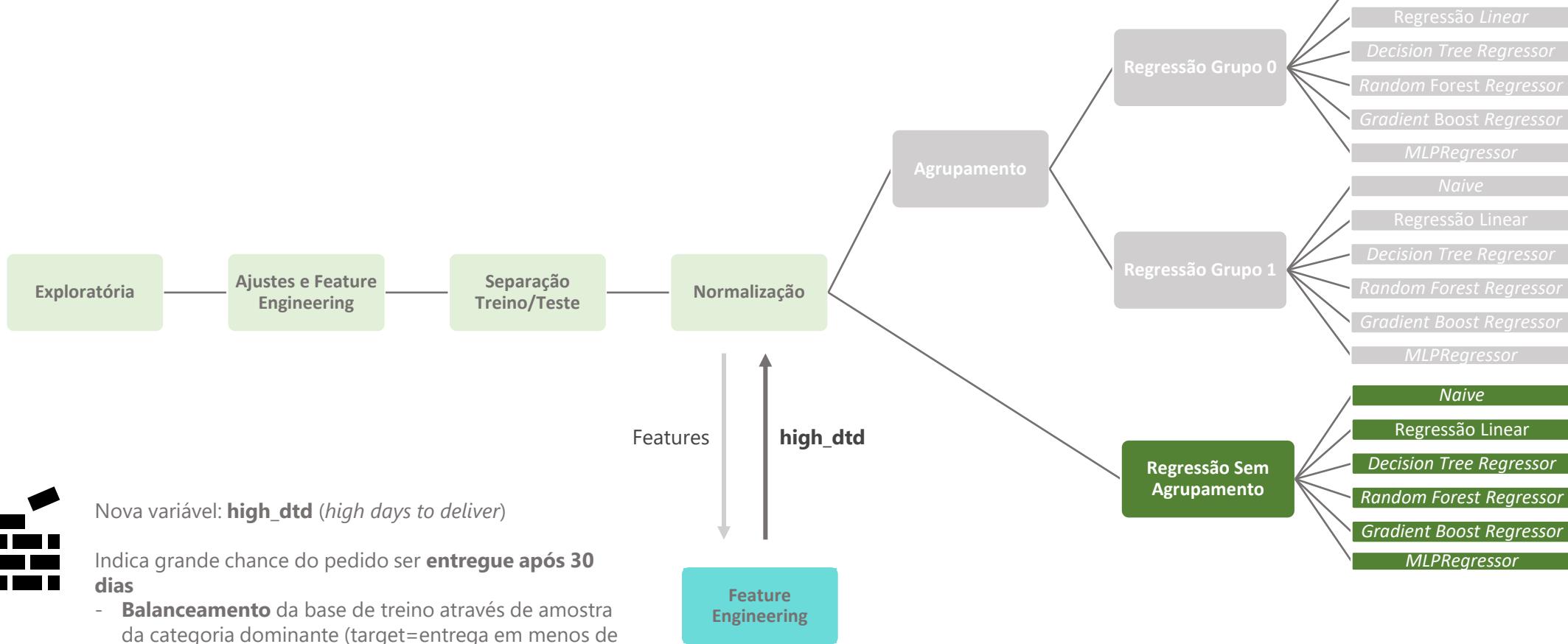
21



5. Regressão

MODELAGEM COM ESTATÍSTICA TRADICIONAL | Resumo

22



5. Predição “Naïve”

MODELAGEM COM ESTATÍSTICA TRADICIONAL | Avaliação dos modelos (sem agrupamento)

23

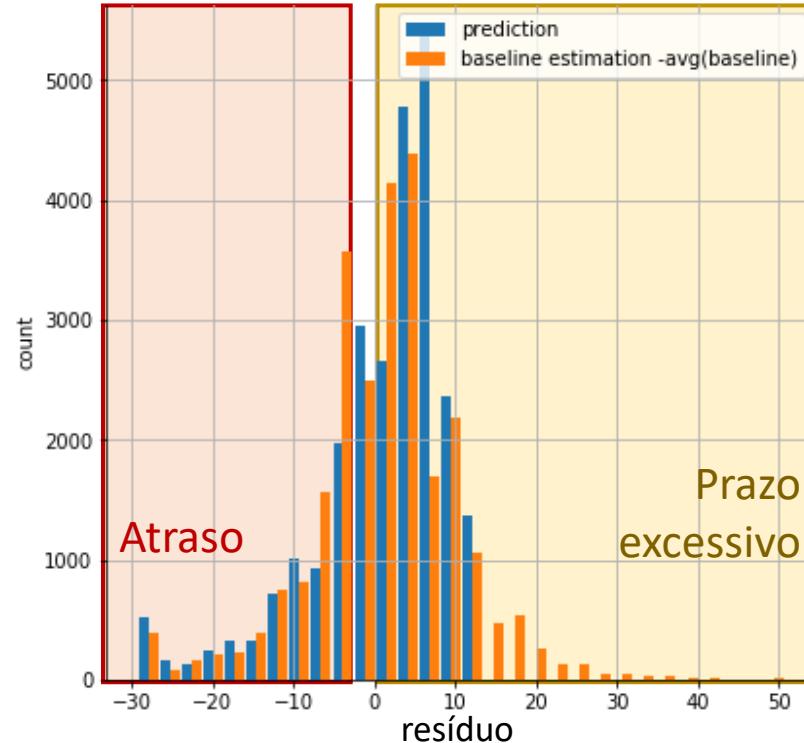
A predição inocente (*naïve*) é simplesmente a média do target na base de treino.
É equivalente ao intercepto de uma regressão linear sem variáveis explicativas

$$\hat{y} = \frac{\sum_{i=1}^N y_i}{N}$$

Predição: 12 dias

Performance

- Modelo apresentou chance de **atraso similar** ao baseline
- Modelo **reduziu o problema de prazos excessivos**



5. Regressão Linear

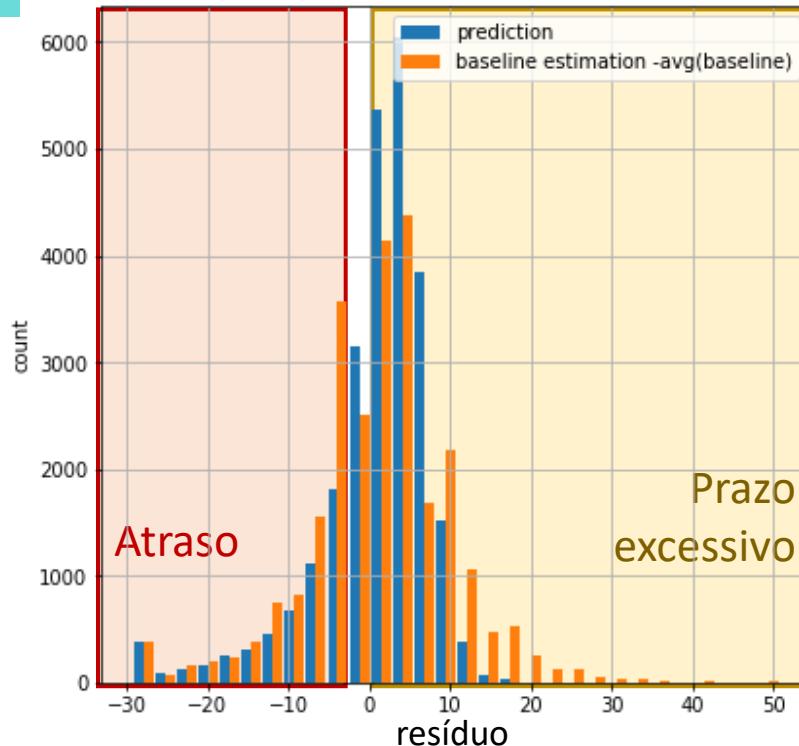
MODELAGEM COM ESTATÍSTICA TRADICIONAL | Avaliação dos modelos (sem agrupamento)

24

Estimativa do target através de uma soma ponderada dos atributos de entrada

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots$$

Variável	Descrição	Coeficiente (β)
Intercepto	-	12,06
is_same_state	Cliente e Vendedor no mesmo estado Clientes e vendedores no mesmo estado reduzem o tempo de entrega	-1,64
order_approved_date	Data de aprovação do pedido Tendência de redução do prazo em novos pedidos	-0,42
distance	Distância entre vendedor e cliente Distância aumenta o prazo de entrega	2,20
days_for_shipping_limit	Prazo para o vendedor entregar mercadoria à logística Tempo para enviar o produto aumenta o tempo de entrega	1,40
seller_complaint_ratio	Taxa de reclamações de atraso associada ao vendedor Vendedores com reclamações tendem a entregar mais tarde	0,84
high_dtd	Alta chance de entregar após 30 dias Maior chance de entregar após 30 dias	0,90



Nota: Também foi utilizado o algoritmo **Generalized Linear Regression**. Após tentar várias combinações de hiperparâmetros, a que forma que se mostrou melhor foi a regressão linear simples.

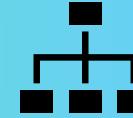


5. Decision Tree Regressor

MODELAGEM COM ESTATÍSTICA TRADICIONAL | Avaliação dos modelos (sem agrupamento)

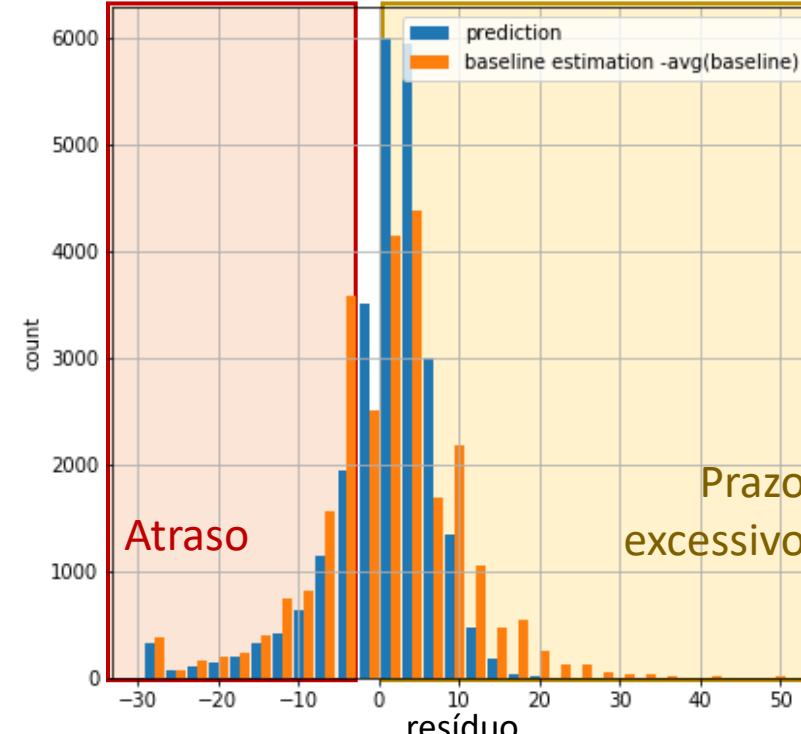
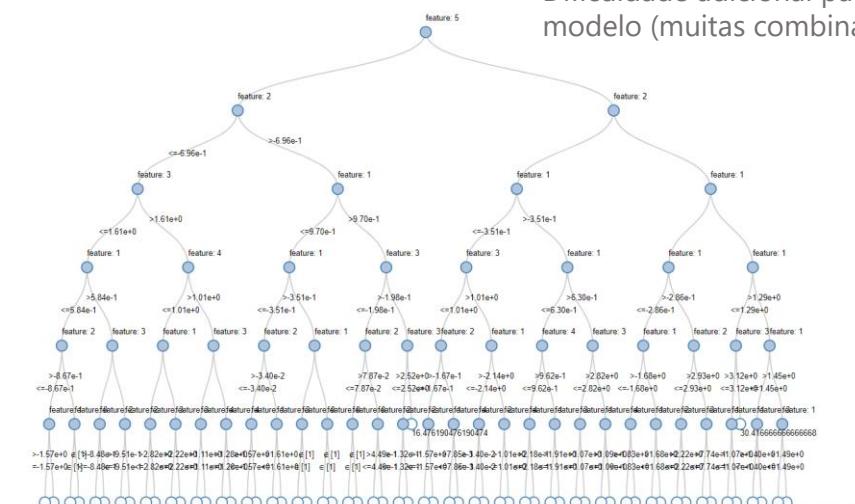
25

Estimativa do target através de uma árvore de decisão



Parâmetro	Valor
Profundidade	6 níveis
Número de nós	123 nós

Por se tratar de regressão, foi necessário aumentar o número de nós para obter resolução.
→ Cuidado adicional para evitar *overfit*
→ Dificuldade adicional para entender o resultado do modelo (muitas combinações)



Nota: Hiperparâmetros foram otimizados através de GridSearch e CrossValidator

Melhor configuração:
MaxDepth: 6
maxBins: 60
MinInstancesPerNode: 12



6. Modelagem Inteligência Artificial

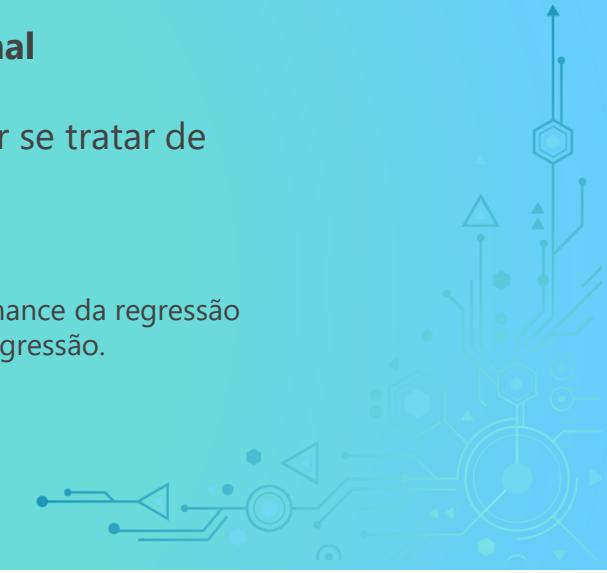
BALANÇAMENTO | BASES DE TREINO E TESTE

26



Mesma base utilizada para modelagem com Estatística Tradicional

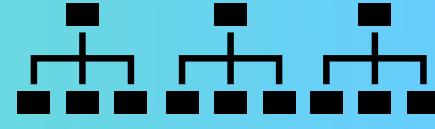
1. Balanceamento da resposta: Balanceamento não foi realizado, por se tratar de problema de regressão
2. Normalização: *StandardScaling* (Média=0, Desvio Padrão=1)
Nota: A normalização é mandatória para o agrupamento, mas não interfere na performance da regressão linear e da árvore de decisão, apesar de ajudar a entender melhor os coeficientes da regressão.
1. 70% aleatório para treino e 30% para teste
 - Treino: 60.798 pedidos
 - Teste: 25.795 pedidos



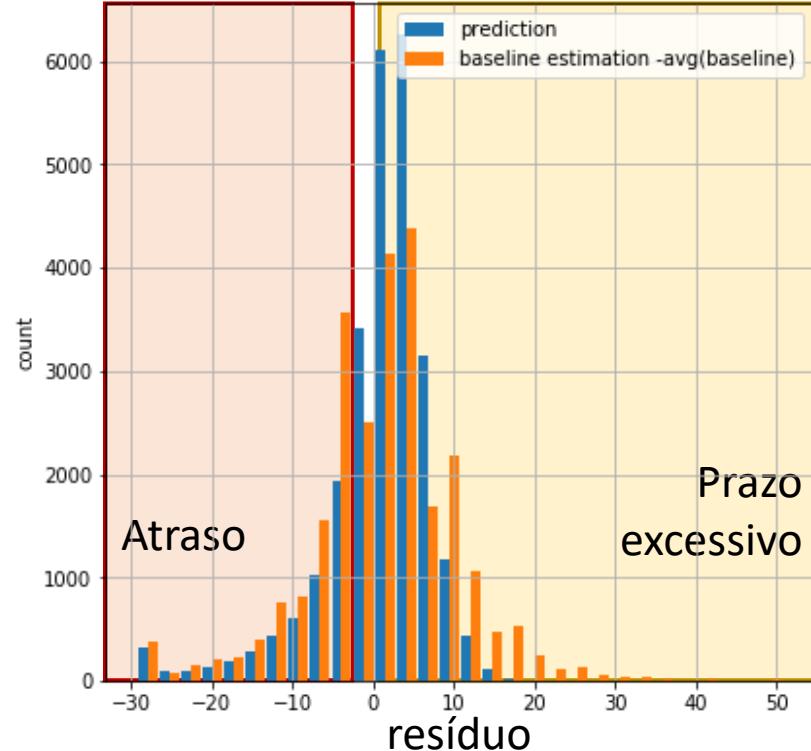
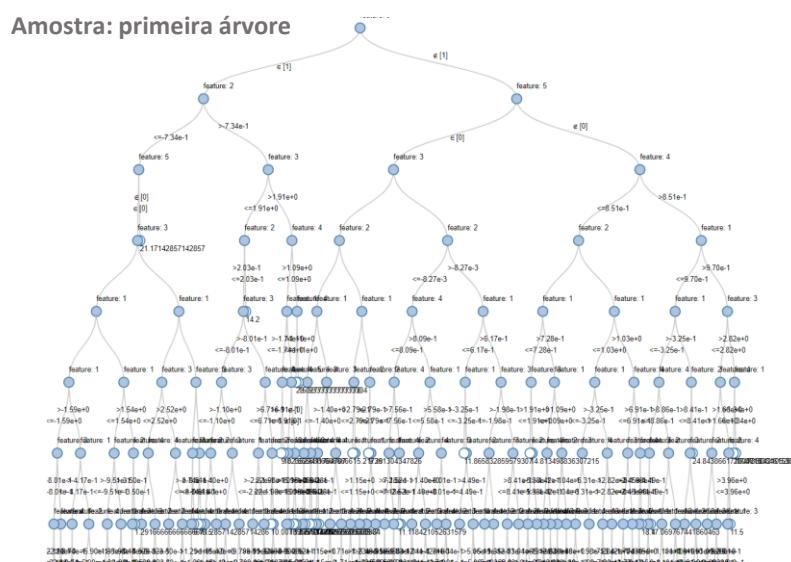
6. Random Forest Regressor

MODELAGEM COM INTELIGÊNCIA ARTIFICIAL | Avaliação dos modelos (sem agrupamento)

Estimativa do target através de um conjunto de árvores de decisão independentes



Parâmetro	Valor
Número de árvores	20 árvores, 27336 nós (média de 1367 nós por árvore)
Número de nós	Por se tratar de regressão, foi necessário aumentar o número de nós para obter resolução. → Cuidado adicional para evitar <i>overfit</i>
Amostra: primeira árvore	



Nota: Hiperparâmetros foram otimizados através de GridSearch e CrossValidator

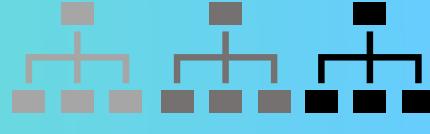
Melhor configuração:
MaxDepth: 12
maxBins: 50
MinInstancesPerNode: 12

6. Gradient Boost Regressor

MODELAGEM COM INTELIGÊNCIA ARTIFICIAL | Avaliação dos modelos (sem agrupamento)

28

Estimativa do target através de um conjunto de árvores de decisão, onde uma árvore é uma otimização da anterior



Parâmetro	Valor
-----------	-------

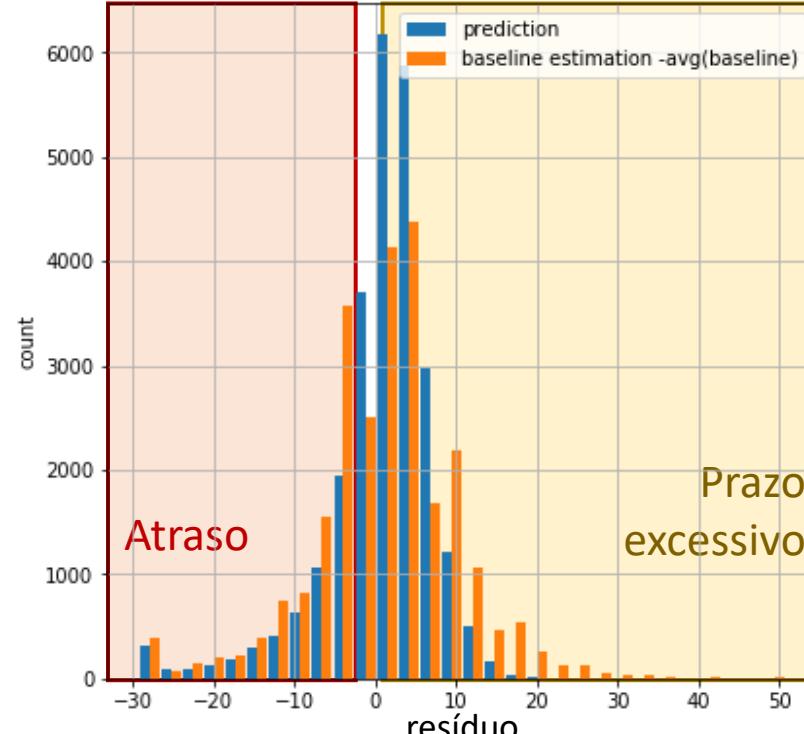
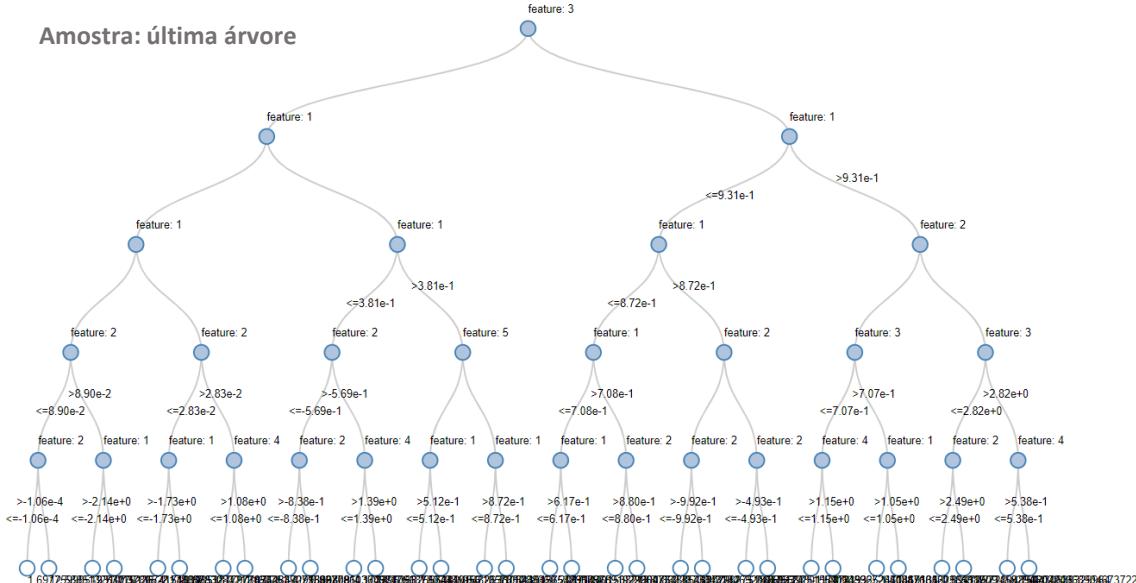
Número de árvores

20 árvores,
1174 nós (média de 58.7 nós por árvore)

Número de nós

Por se tratar de regressão, foi necessário aumentar o número de nós para obter resolução.
→ Cuidado adicional para evitar overfit

Amostra: última árvore



Nota: Hiperparâmetros foram otimizados através de GridSearch e CrossValidator

Melhor configuração:

MaxDepth: 5
maxBins: 60
MinInstancesPerNode: 12



6. Multi Layer Perceptron Regressor

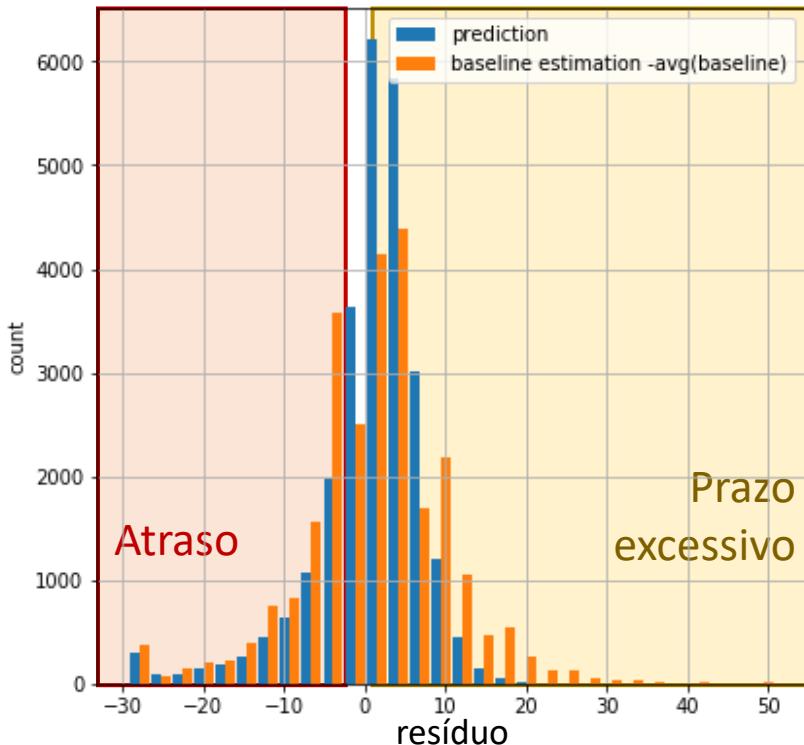
MODELAGEM COM INTELIGÊNCIA ARTIFICIAL | Avaliação dos modelos (sem agrupamento)

29

Estimativa do target através de um conjunto de árvores de decisão, onde uma árvore é uma otimização da anterior

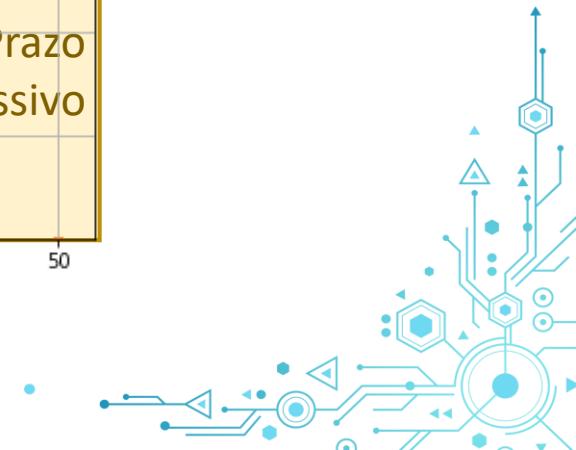


Parâmetro	Valor
Camadas	8 neurônios → 8 neurônios → 9 neurônios → 11 neurônios
Ativação	relu
RMSE	Treino: 7.787 Teste: 8.163



Nota: O Spark não possui biblioteca nativa para redes neurais regressoras, então este algoritmo foi executado pelo Pandas (*sklearn*)

Nota: Hiperparâmetros foram otimizados através da comparação de 92 combinações diferentes



6. Seleção de variáveis

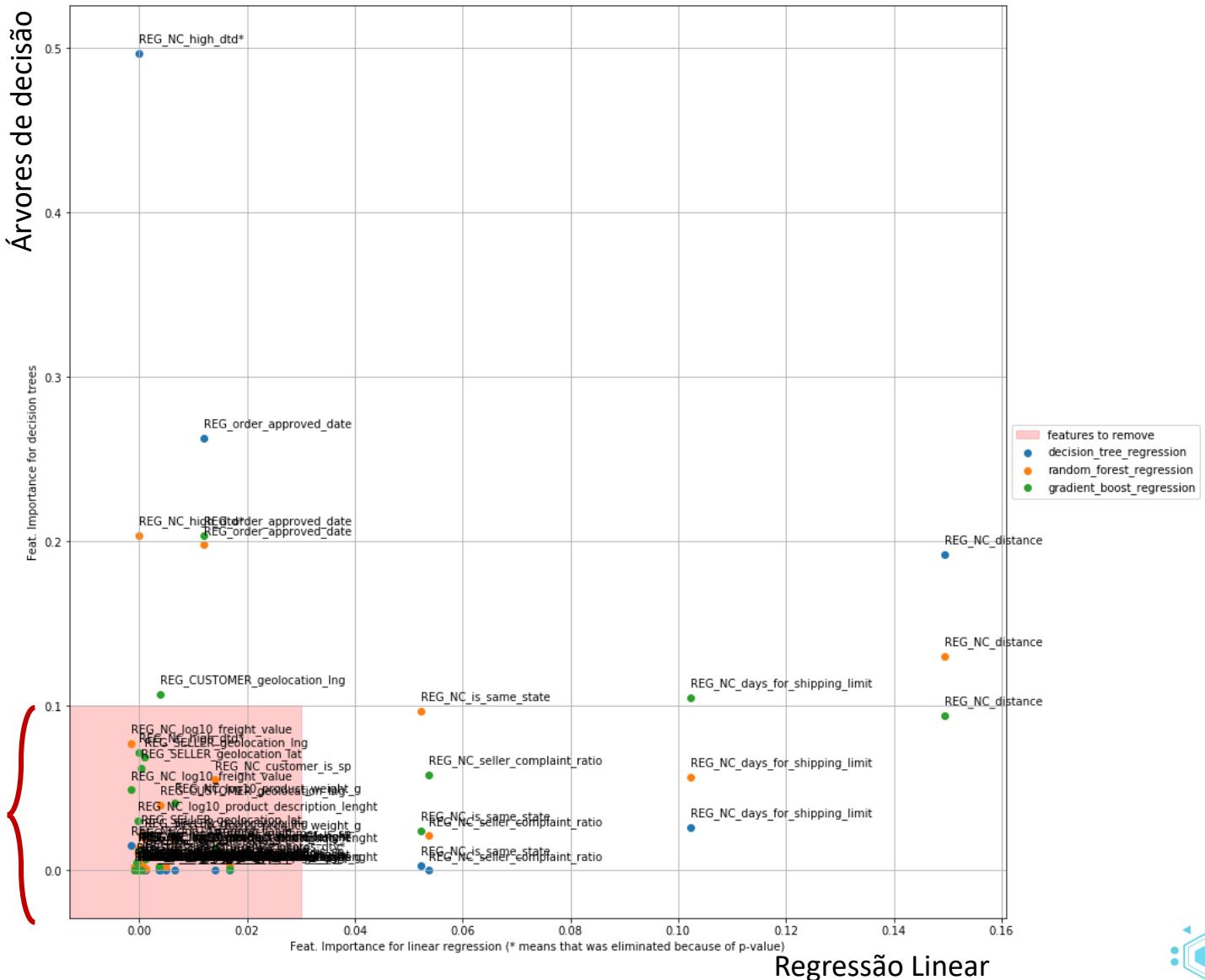
Variáveis selecionadas:

- is_same_state
 - order_approved_date
 - distance
 - days_for_shipping_limit
 - seller_complaint_ratio
 - high_dtd

Pouca redução nos erros quadráticos

Todas as variáveis selecionadas têm baixo p-valor

Variáveis eliminadas



6. Seleção do modelo

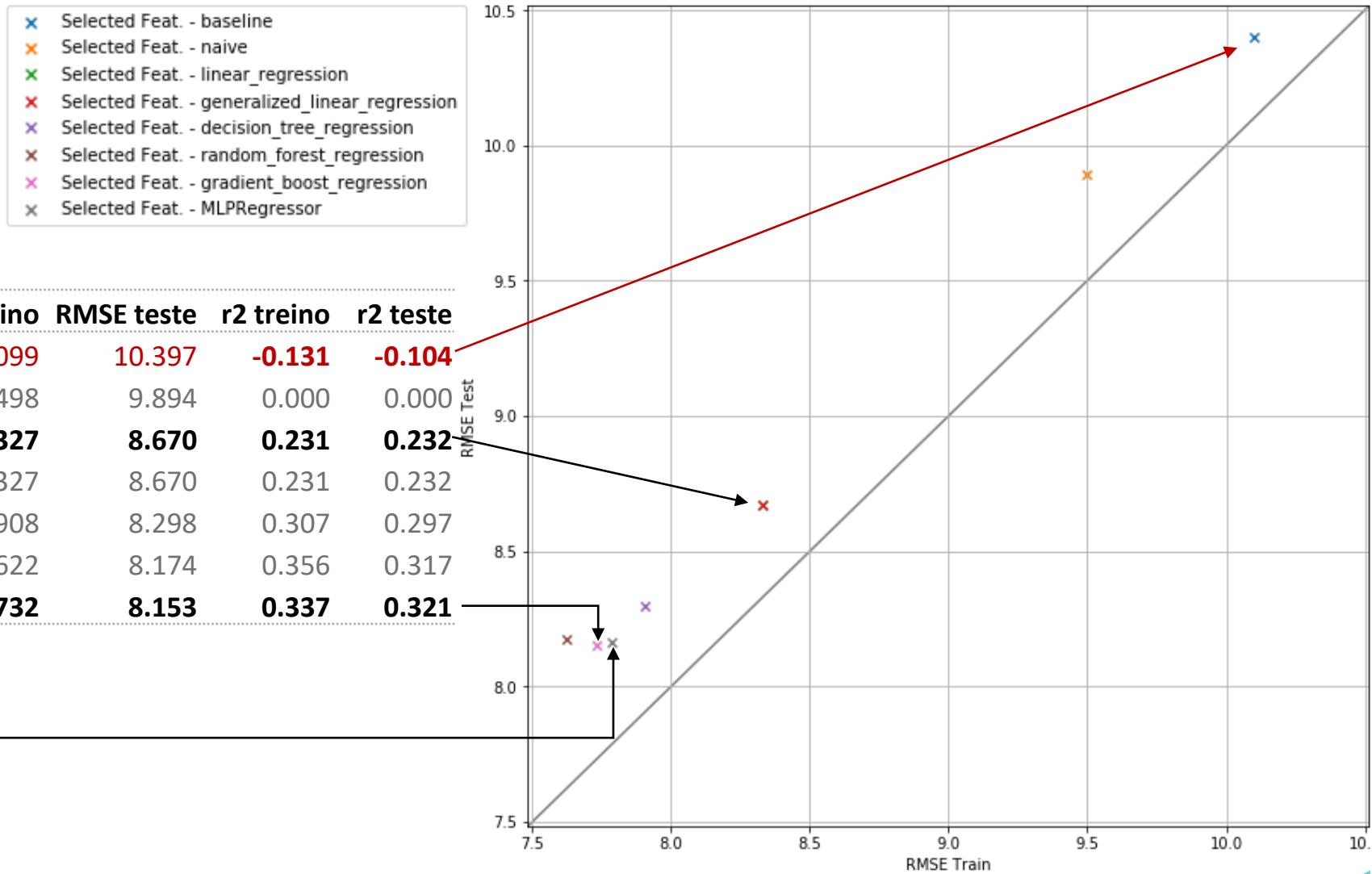
31

Método	RMSE treino	RMSE teste	r2 treino	r2 teste
baseline	10.099	10.397	-0.131	-0.104
naive	9.498	9.894	0.000	0.000
linear_regression	8.327	8.670	0.231	0.232
generalized_linear_regression	8.327	8.670	0.231	0.232
decision_tree_regression	7.908	8.298	0.307	0.297
random_forest_regression	7.622	8.174	0.356	0.317
gradient_boost_regression	7.732	8.153	0.337	0.321

Rede Neural Regressora:

RMSE Treino: 7.787

RMSE Teste: 8.163





7. Conclusões

- A Performance do estimador ***naïve* (média)** já foi **melhor que o *baseline***
- A **Régressão Linear trouxe mais refinamento** em relação ao *naïve*
 - *Mesmo Intercepto*
- O **Random Forest** precisou de muito **mais nós** e teve **resultado inferior** ao *Gradient Boost*
- O **Gradient Boost** e a **Rede Neural** apresentaram os **melhores resultados**
- Apesar do resultado ser levemente inferior, a **Régressão Linear** ainda é uma solução interessante porque é **mais previsível e simples de implementar**
- Os indicadores criados (***high_dtd*** e ***complaint_ratio***) **contribuíram** para o refinamento do modelo
- O **agrupamento não auxiliou** o refinamento do modelo
- Variáveis que melhor explicam o prazo de entrega:
is_same_state, order_approved_date, distance, days_for_shipping_limit, seller_complaint_ratio, high_dtd

Desafios encontrados

Manter a “governança” da análise

Garantir a integridade dos dados desde a aquisição até o uso pelos modelos.

Garantir controle dos arquivos parquet

Gerenciar os notebooks (9 notebooks no total)

Garantir a integridade ao adicionar novas variáveis no modelo

Coerência entre as análises de agrupamento (*cluster*)

Gestão de execução ao rodar um notebook anterior usando `%run`

Gestão de nomes de colunas

Criação de gráficos escaláveis

Criação de funções em PySpark para gerar os dados por trás dos gráficos.

- Histogramas
- Heatmaps
- Amplo uso de *Window()* no Spark

Manter performance do código

Uso de *repartition* e *cache*

Otimização do número de partições

Uso de *parquet* para armazenar principais resultados intermediários

- Grande melhoria no tempo de carregamento dos dados e cache

Exploratória

Escolha de estratégias para lidar com dados faltantes e inválidos

Escolha de estratégias para lidar com variáveis com muitas categorias

Escolha de variáveis relevantes para o problema

- Impactos só serão sentidos muitas etapas a frente



Backup



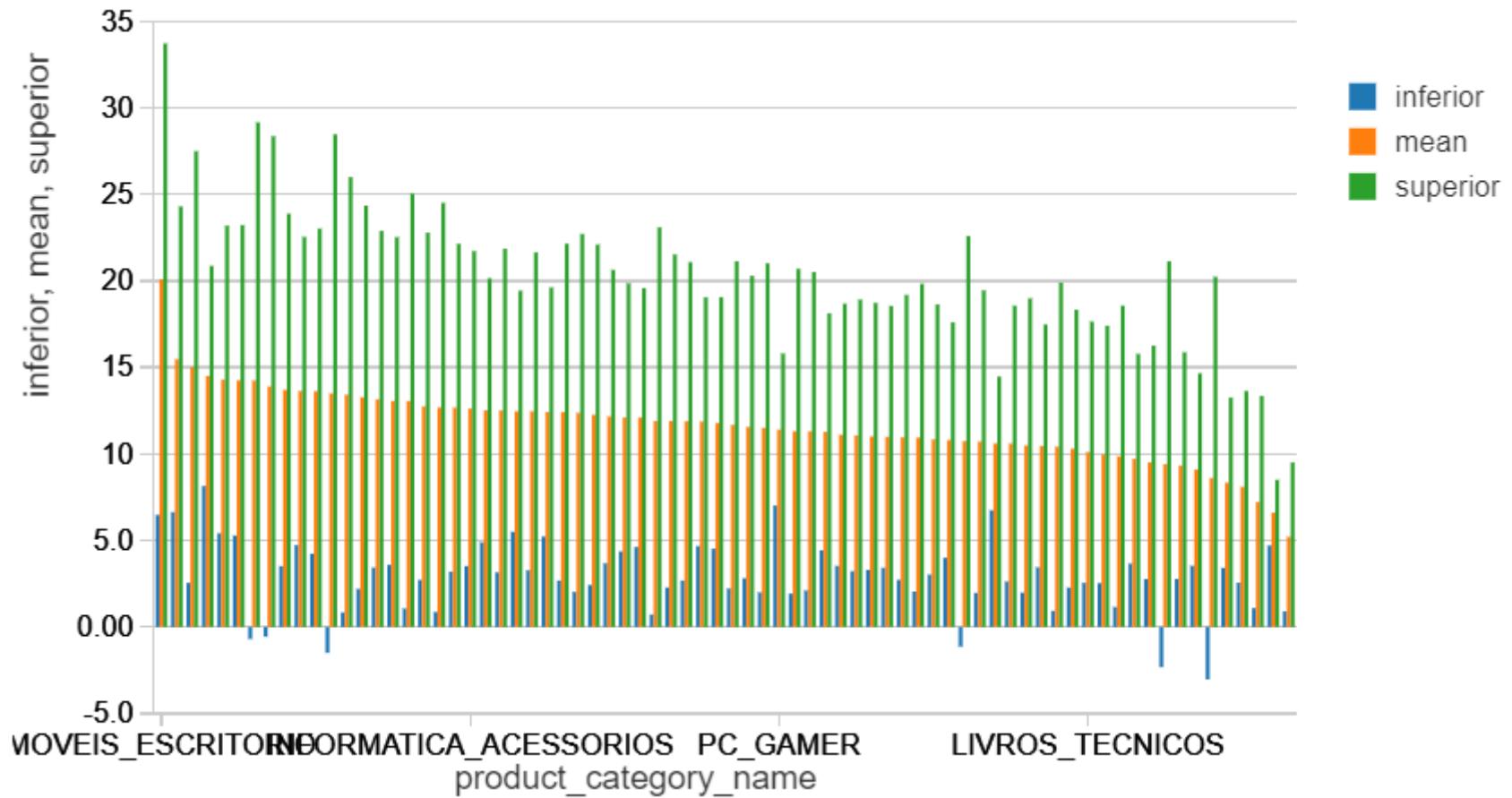
Casos de entrega com 20 dias de antecedência

baseline	8.345	de 8.345 - redução de	0,00%
naive	5.550	de 8.345 - redução de	33,49%
linear_regression	3.028	de 8.345 - redução de	63,71%
generalized_linear_regression	3.026	de 8.345 - redução de	63,74%
decision_tree_regression	3.445	de 8.345 - redução de	58,72%
random_forest_regression	2.589	de 8.345 - redução de	68,98%
gradient_boost_regression	3.179	de 8.345 - redução de	61,91%



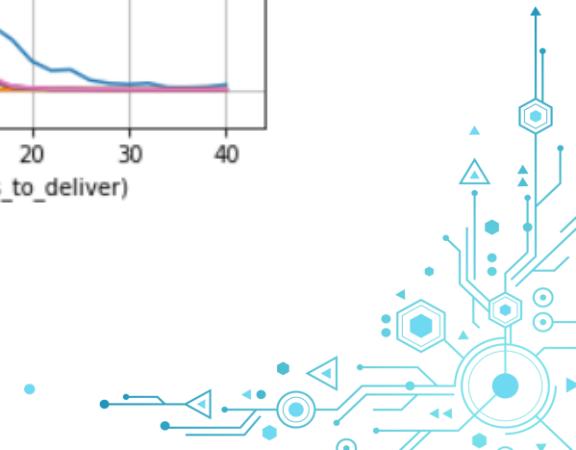
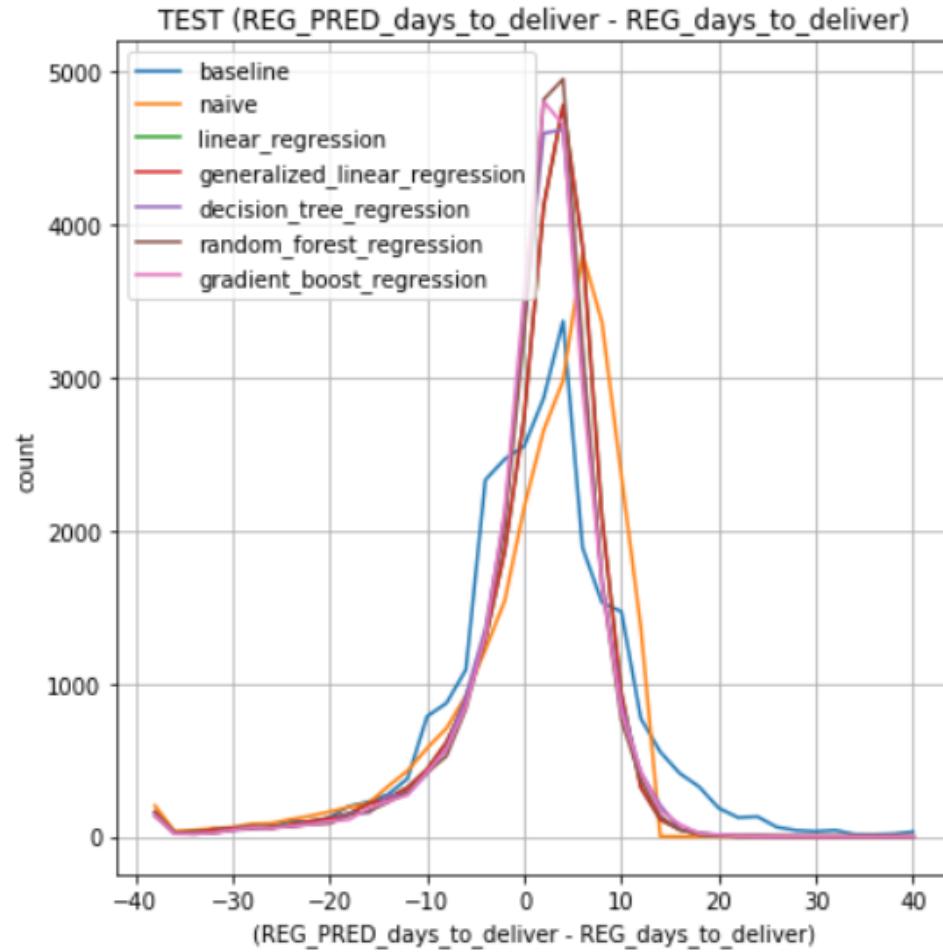
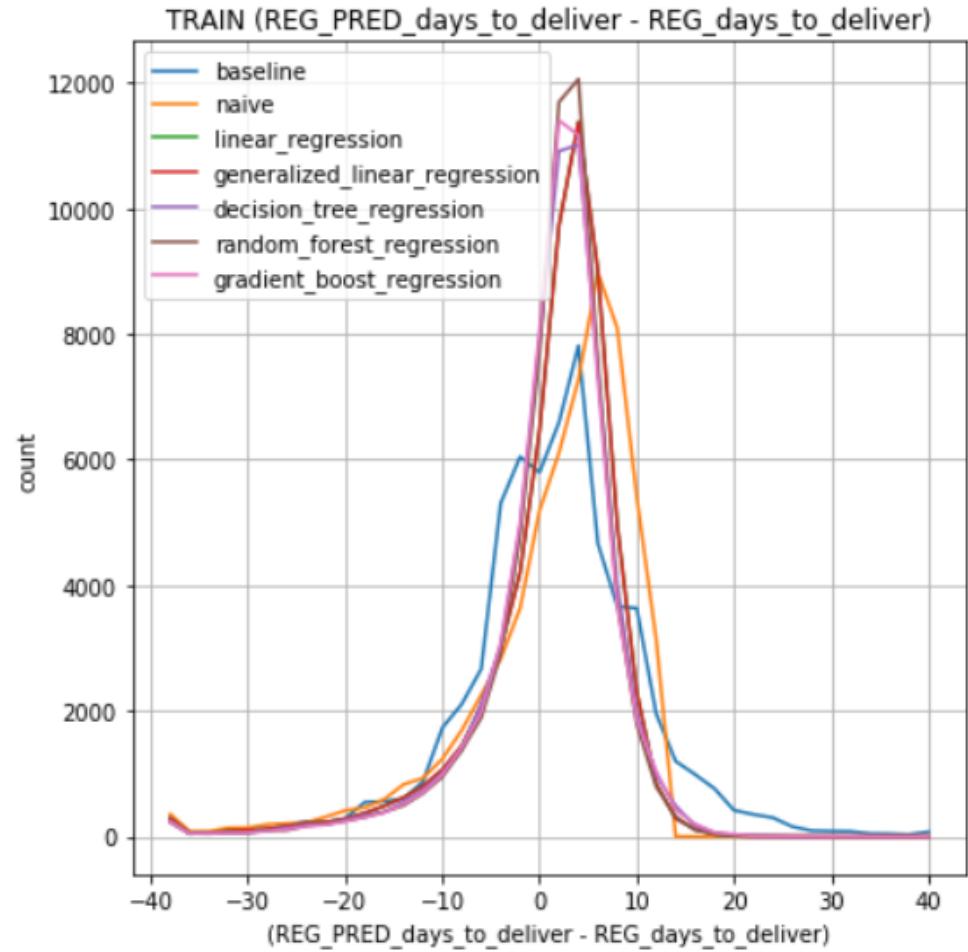
Backup

36



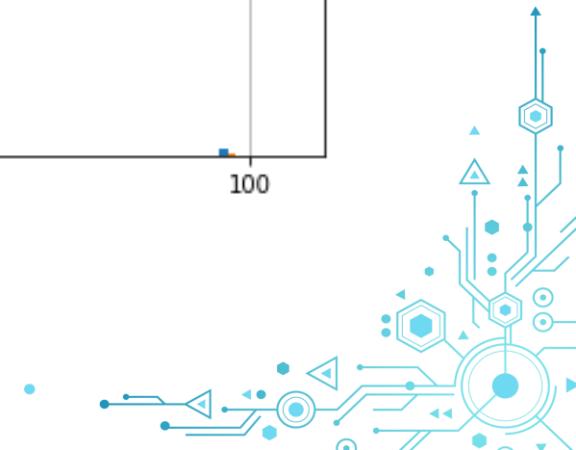
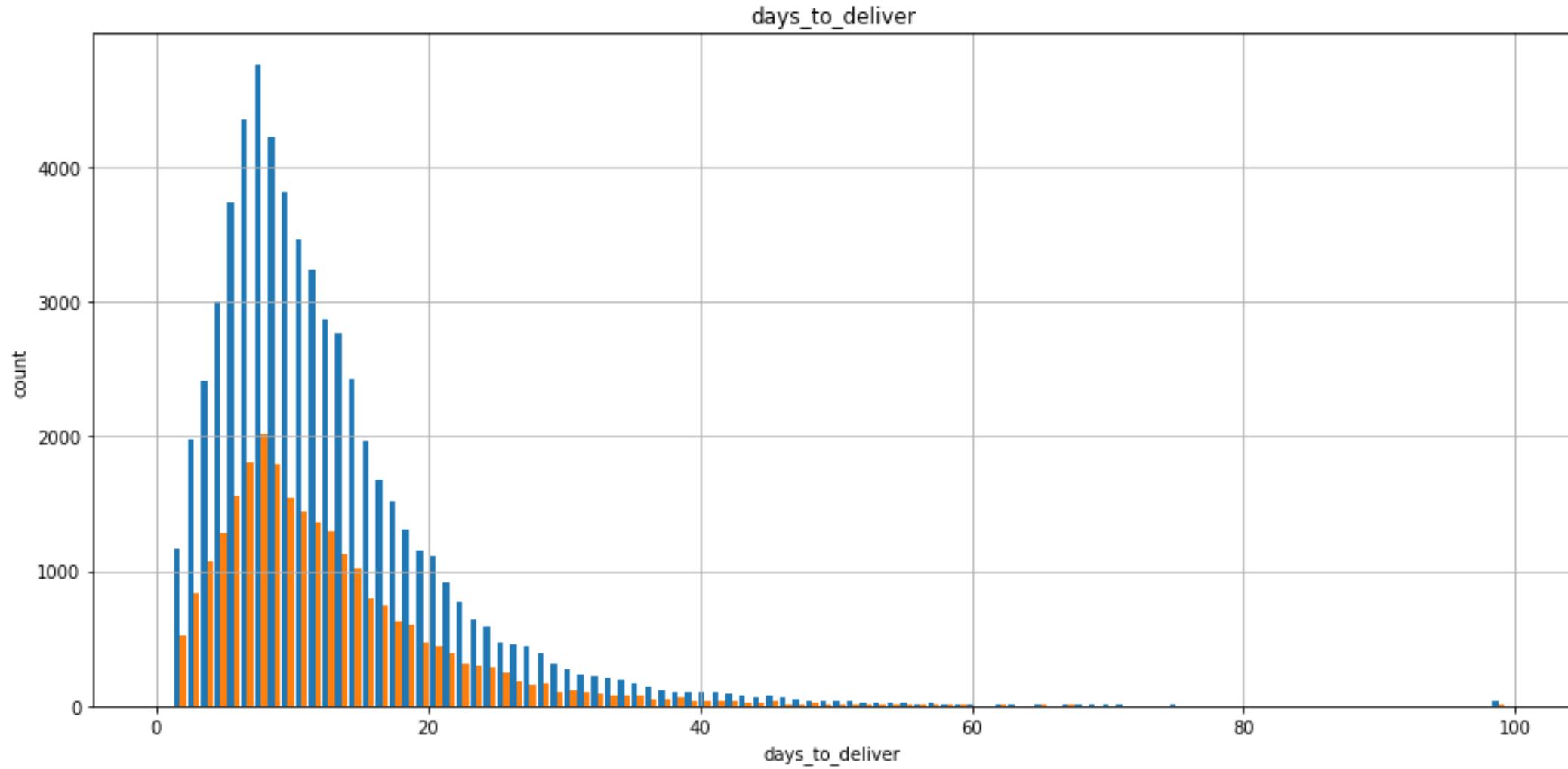
Backup

37



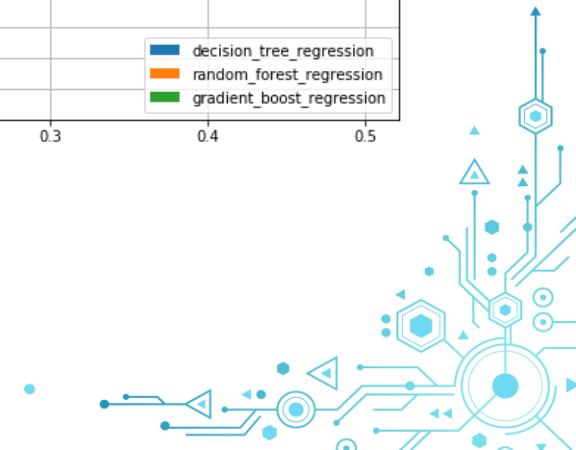
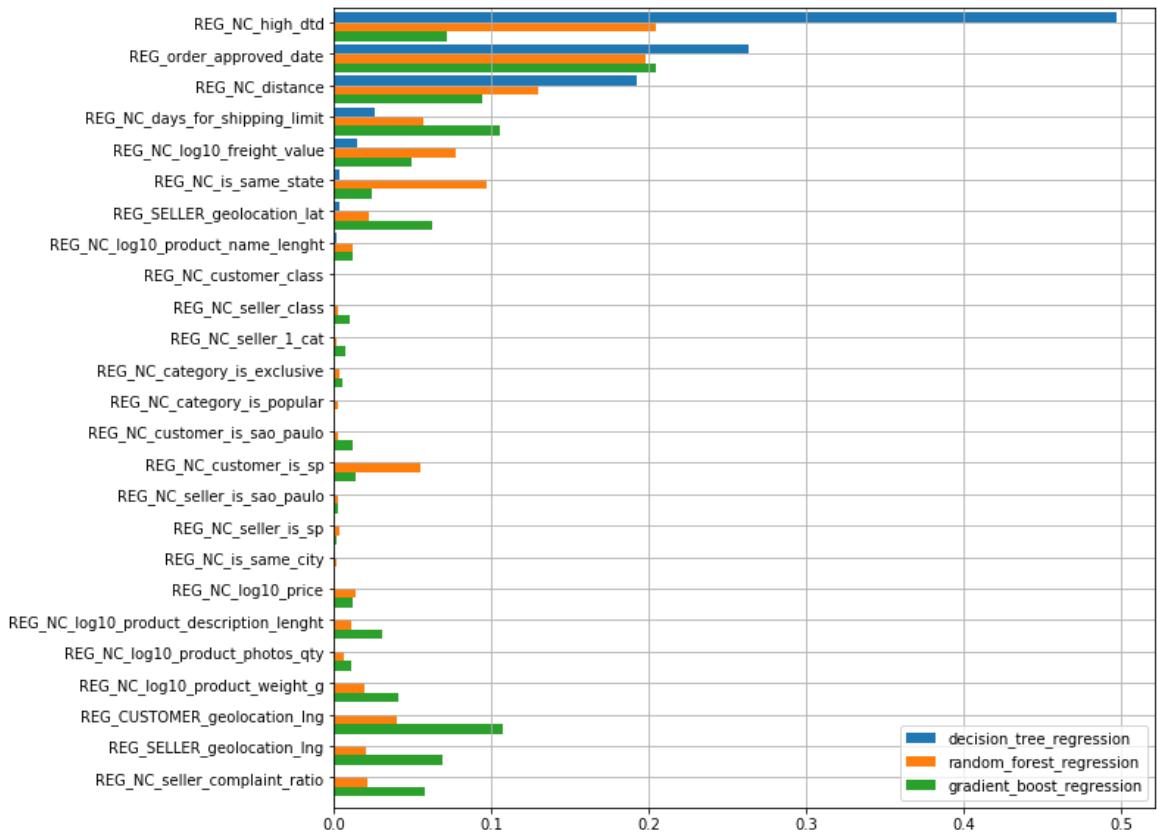
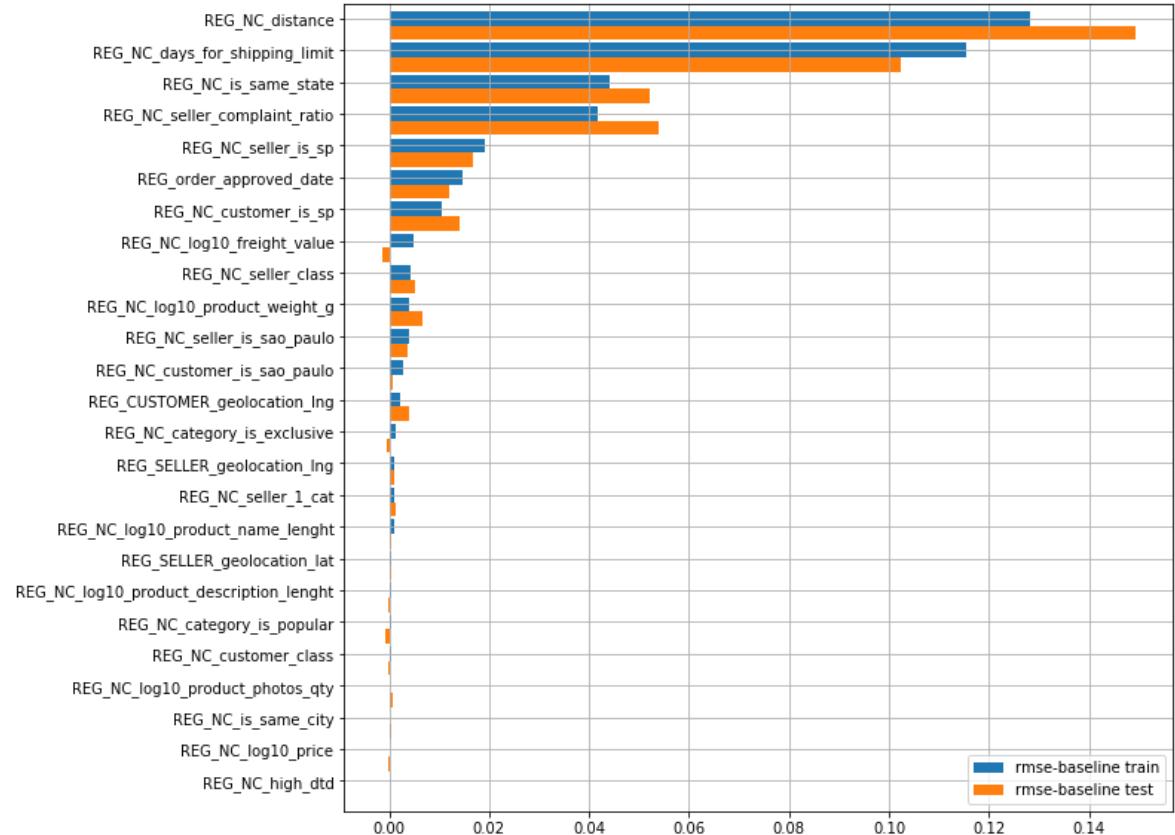
Backup – Target Treino/Teste

38



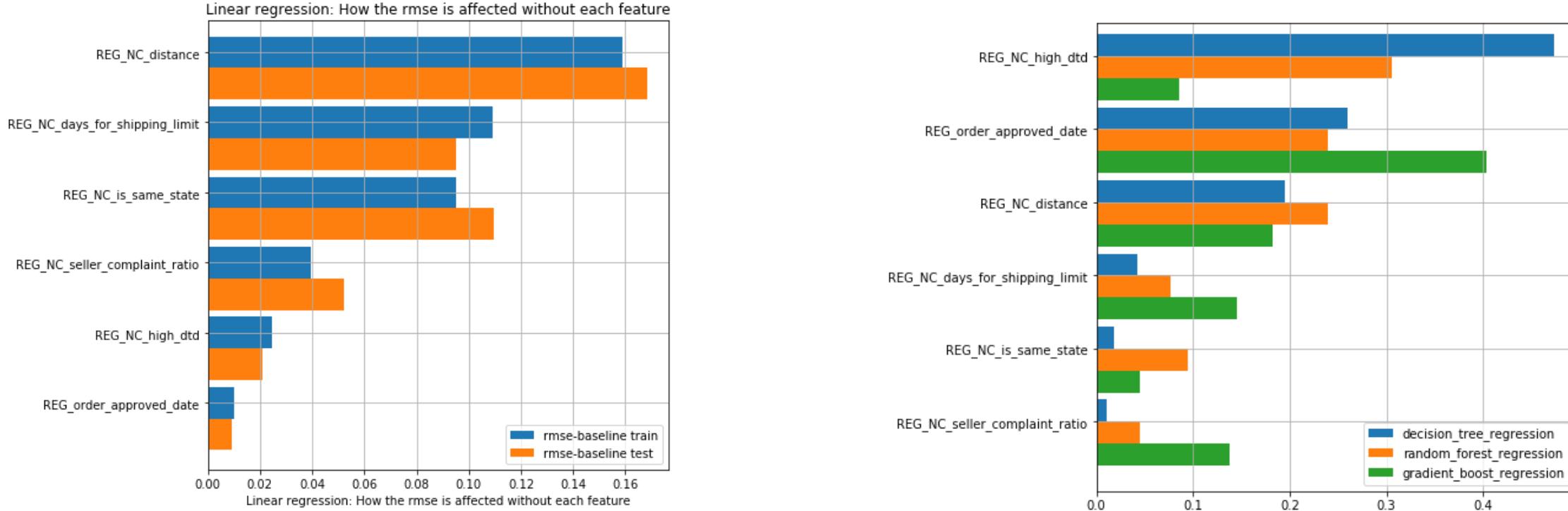
Backup

39



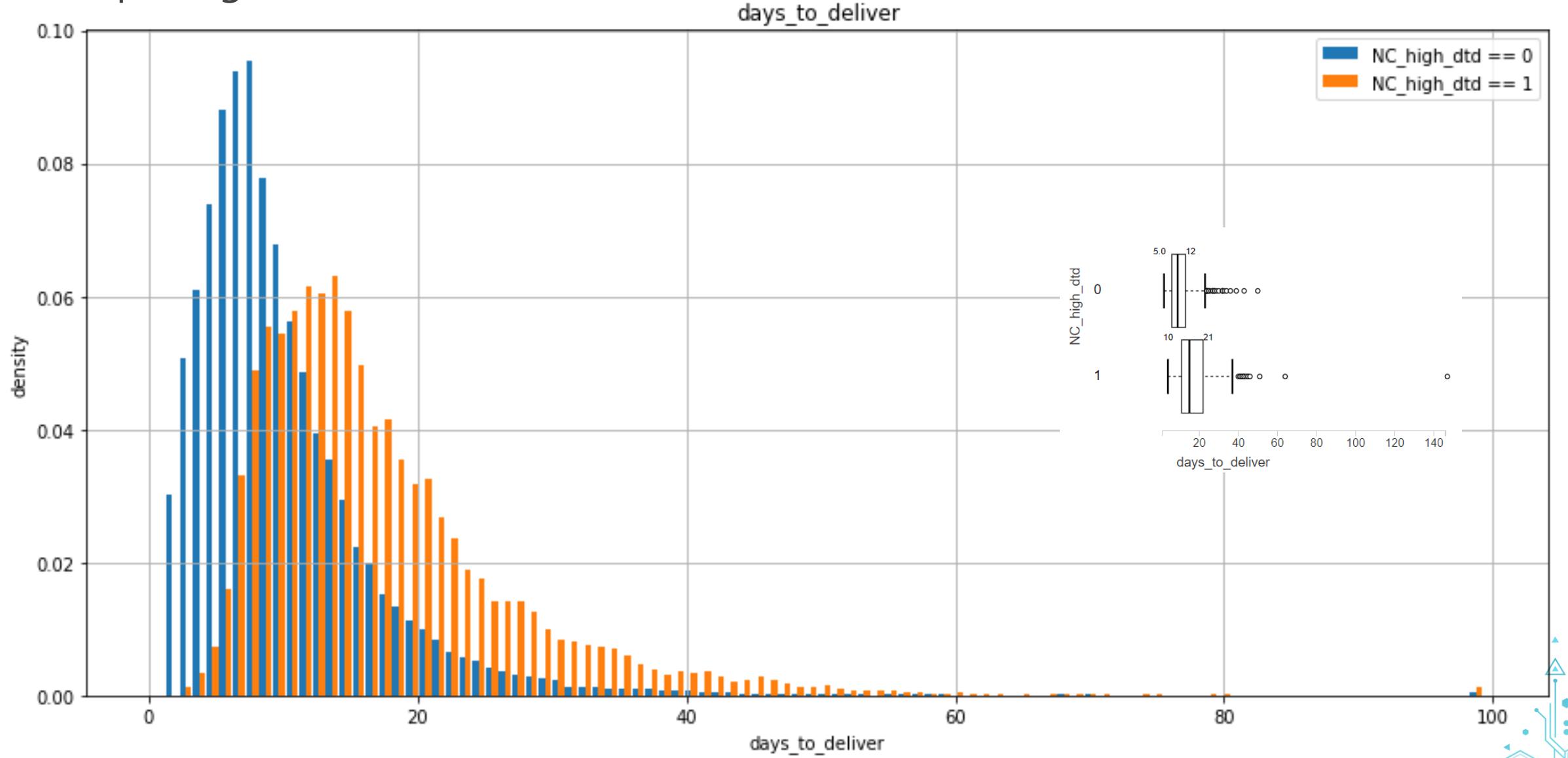
Backup

40



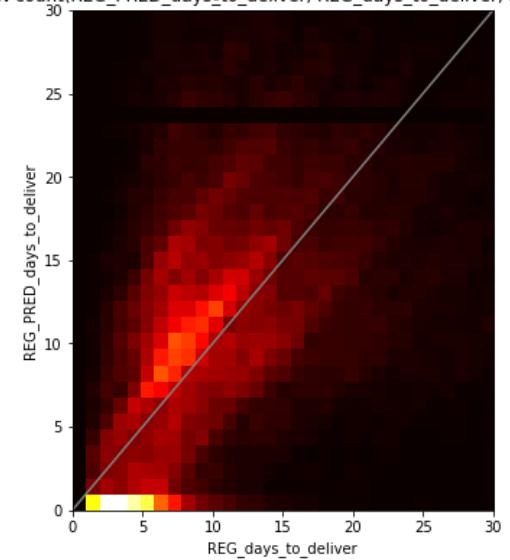
Backup – high_dtd

41

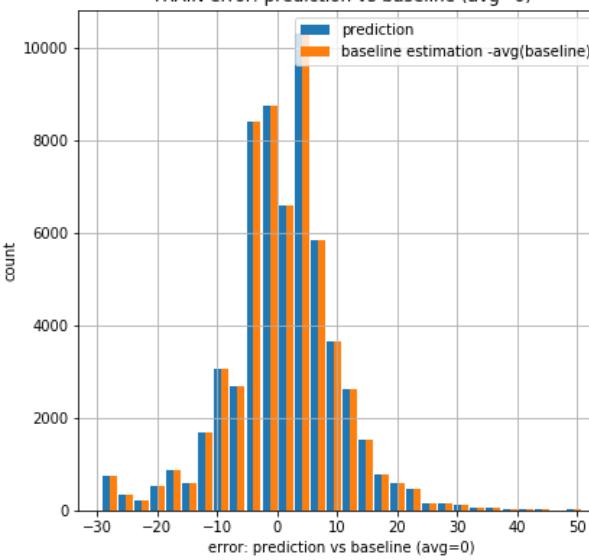


Backup - Baseline

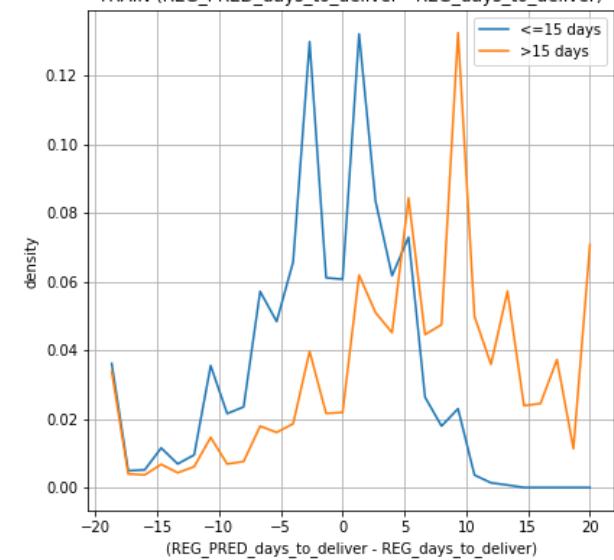
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



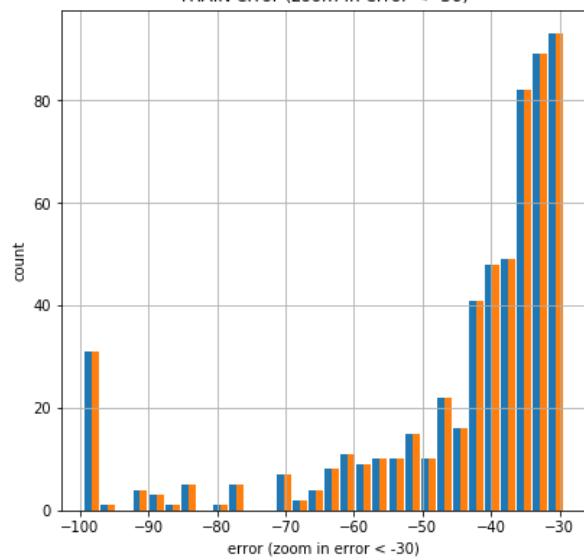
TRAIN error: prediction vs baseline (avg=0)



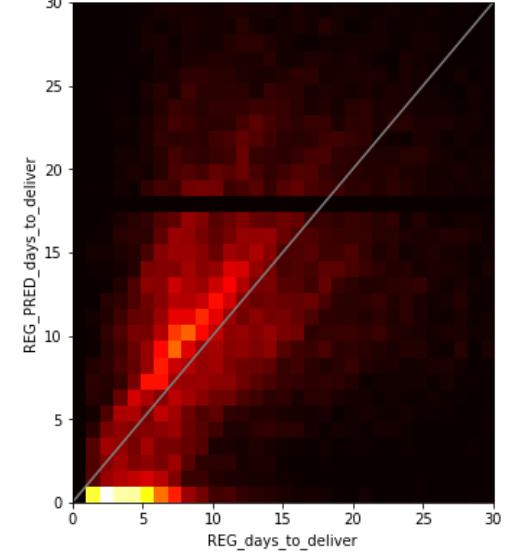
TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)



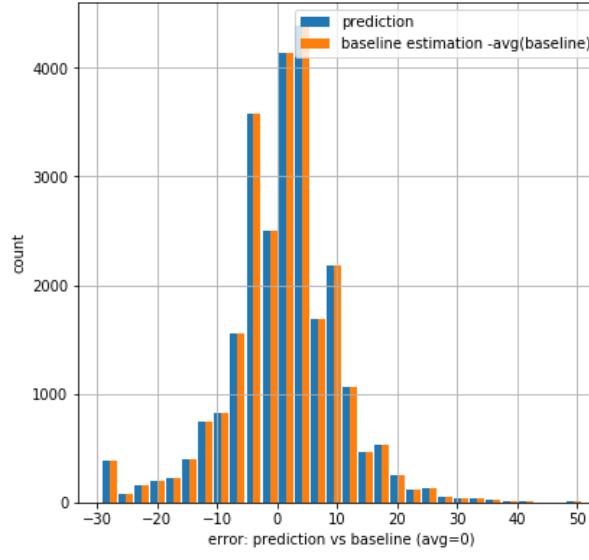
TRAIN error (zoom in error < -30)



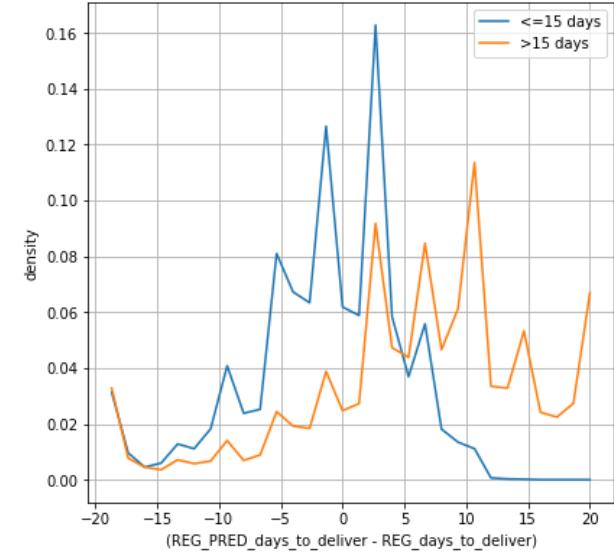
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



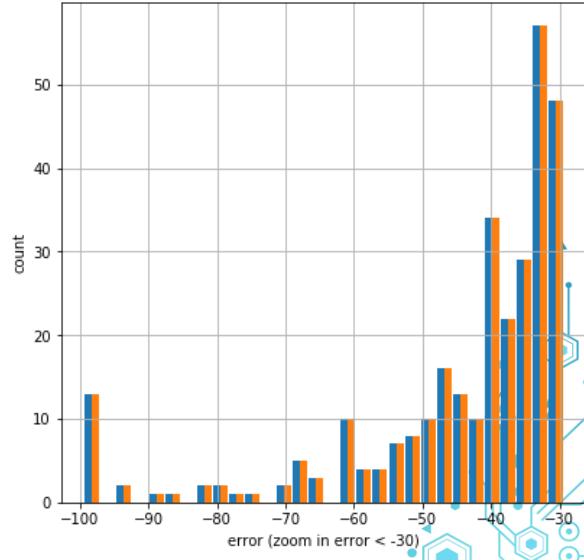
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)

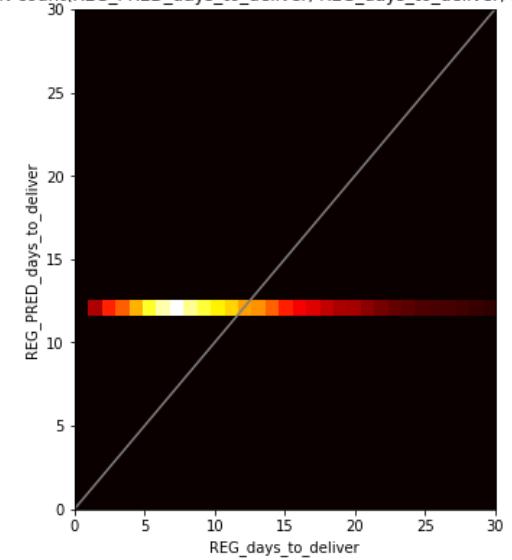


TEST error (zoom in error < -30)

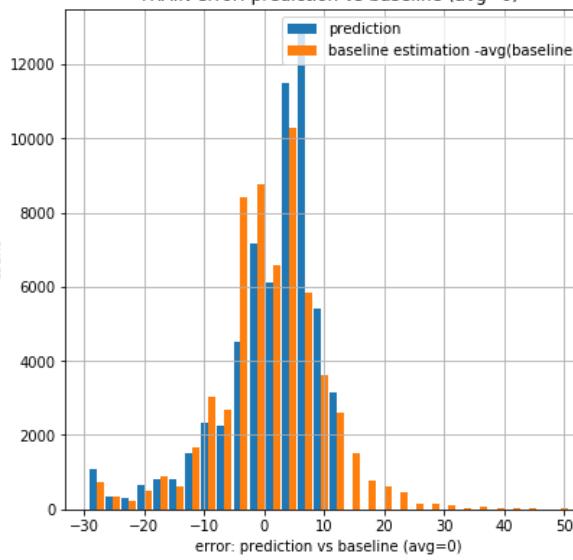


Backup – Naïve

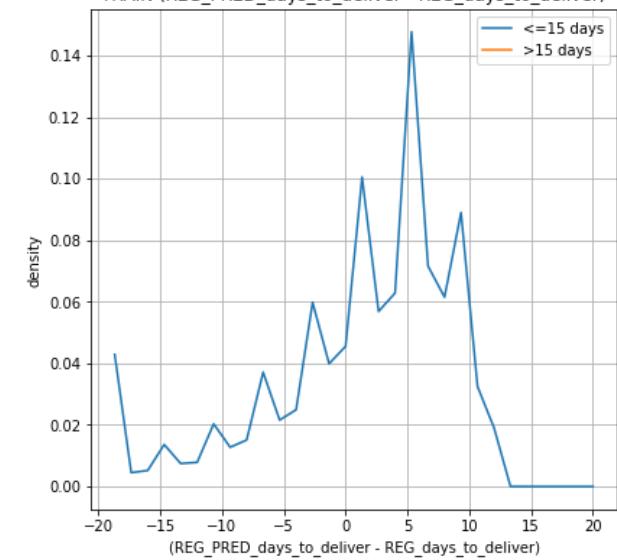
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



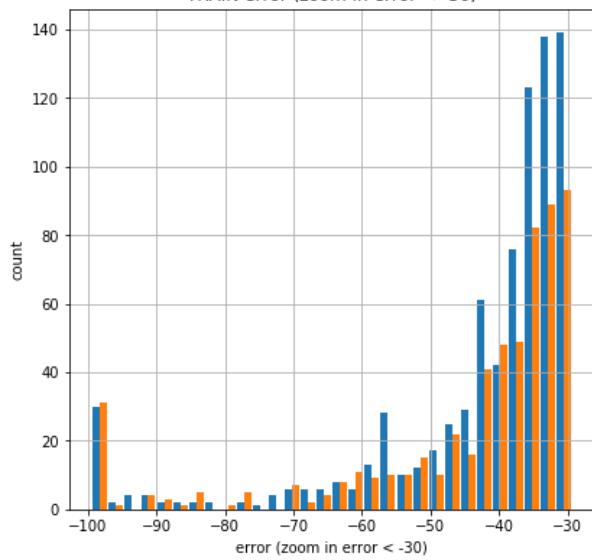
TRAIN error: prediction vs baseline (avg=0)



TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)

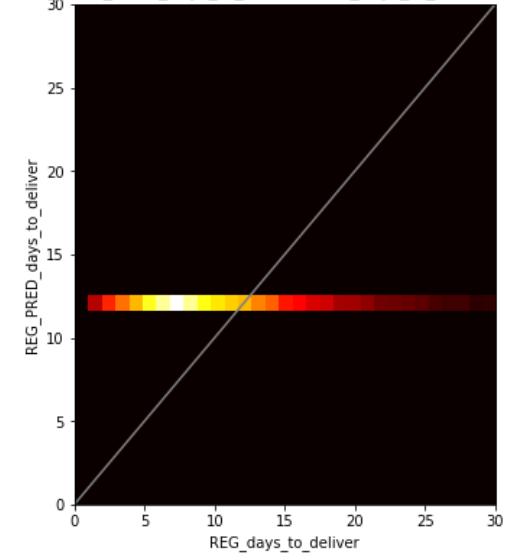


TRAIN error (zoom in error < -30)

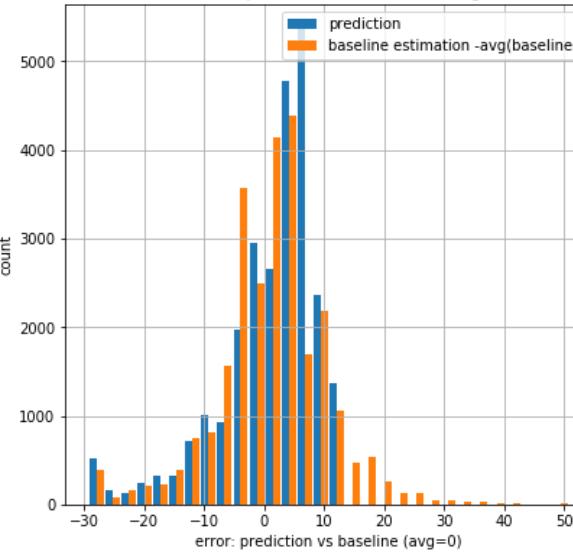


43

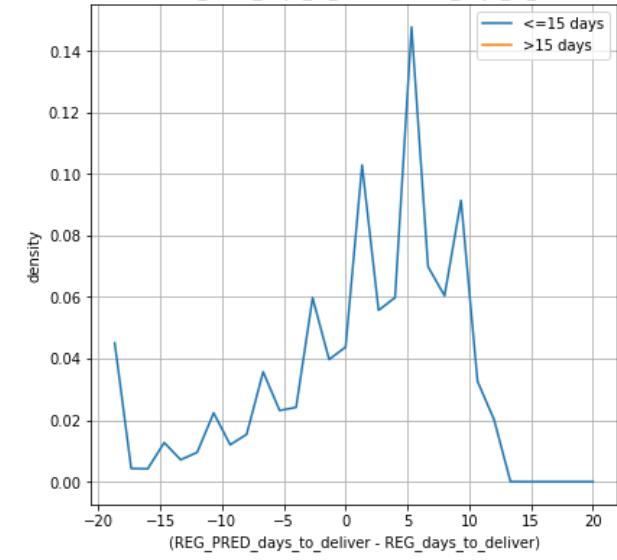
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



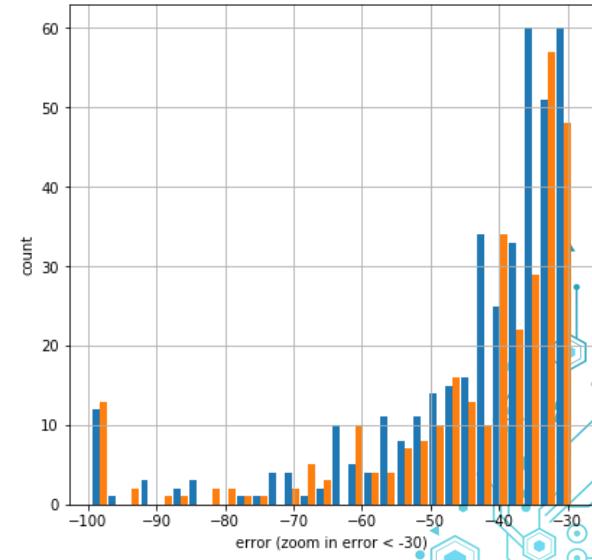
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)

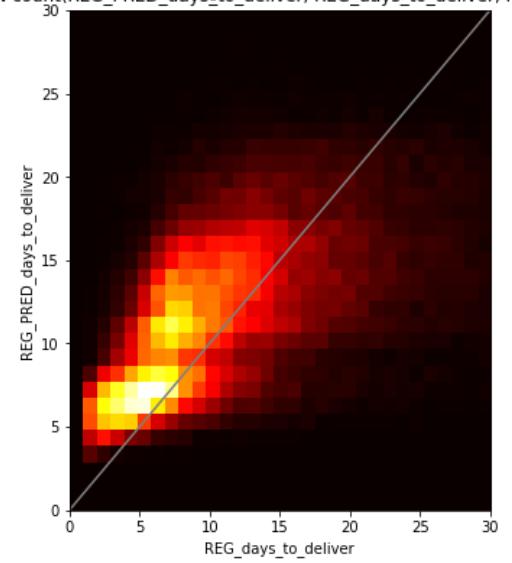


TEST error (zoom in error < -30)

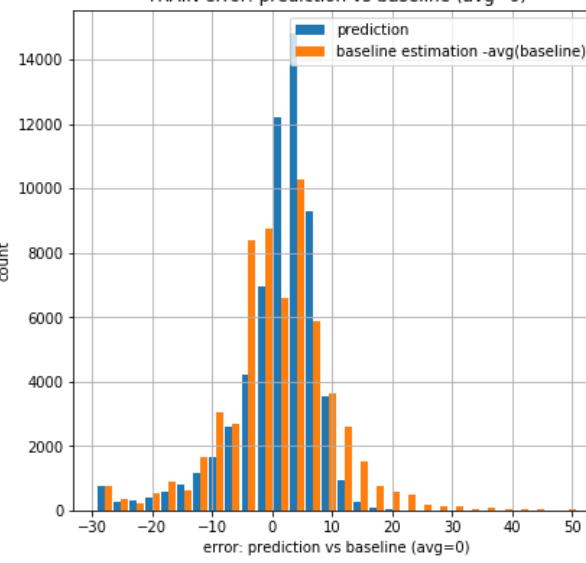


Backup – Linear Regression

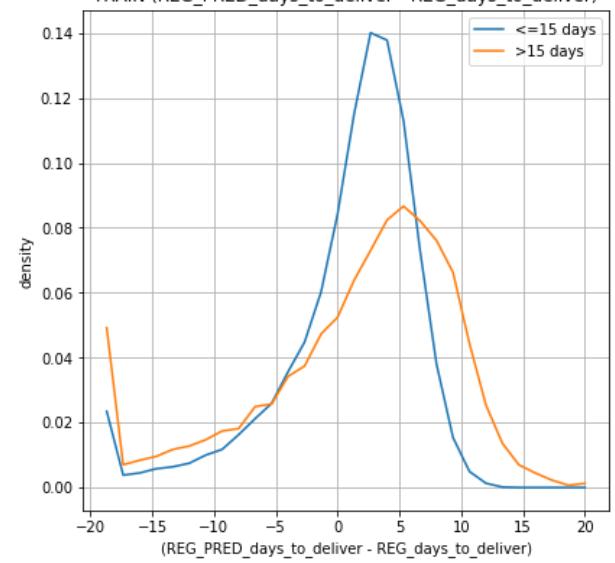
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



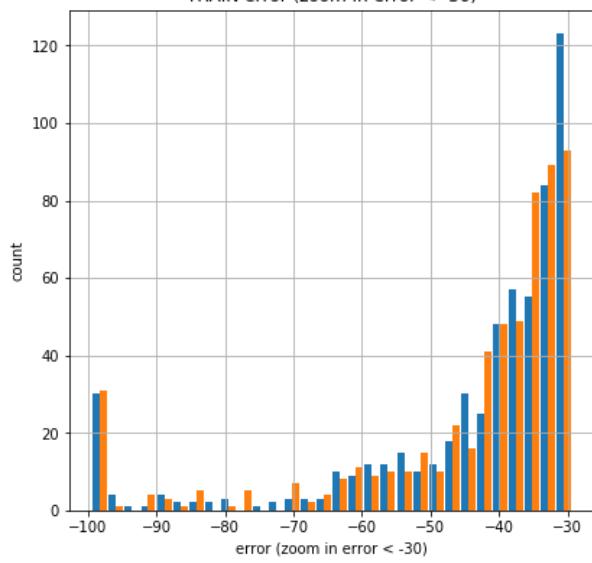
TRAIN error: prediction vs baseline (avg=0)



TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)

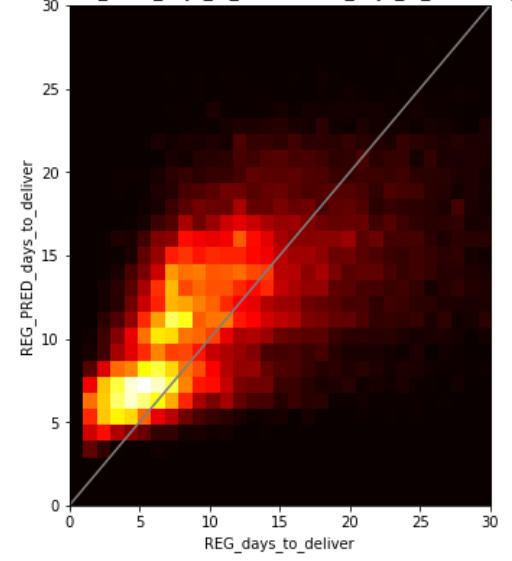


TRAIN error (zoom in error < -30)

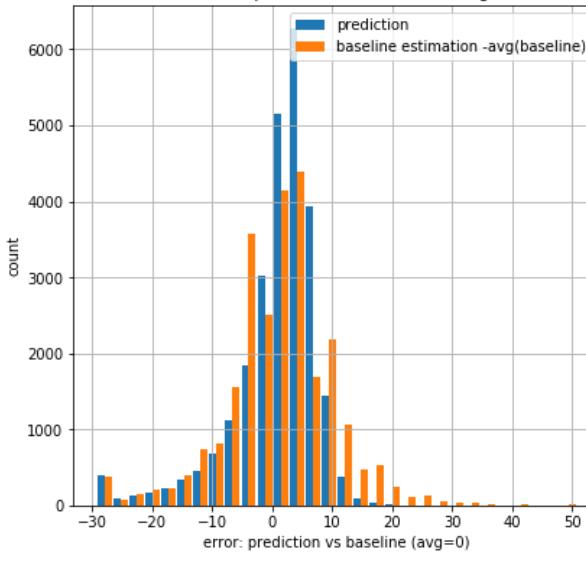


44

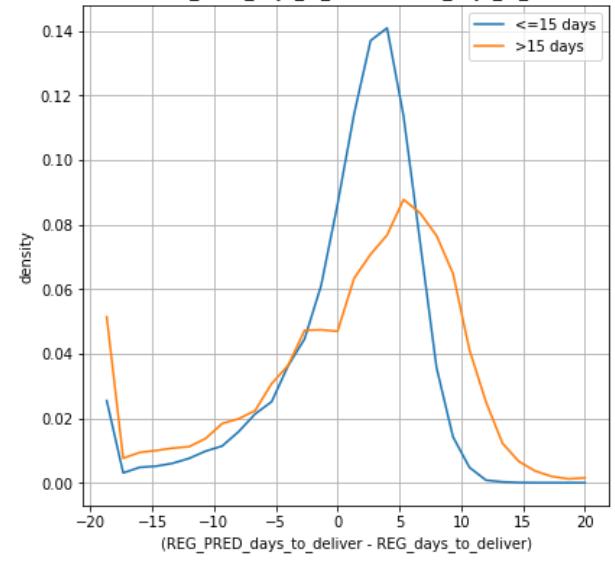
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



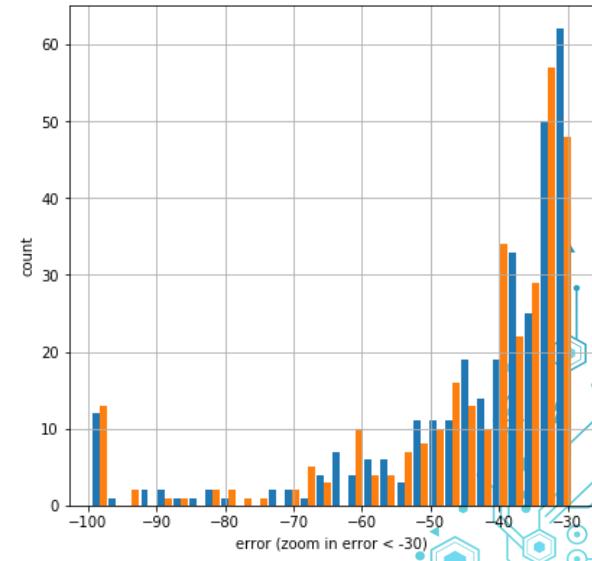
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)



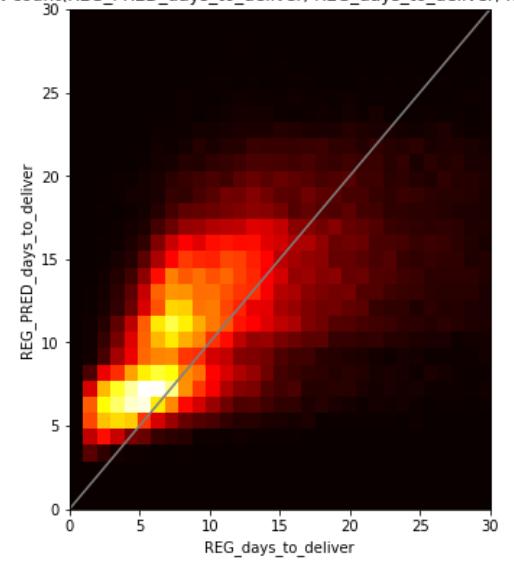
TEST error (zoom in error < -30)



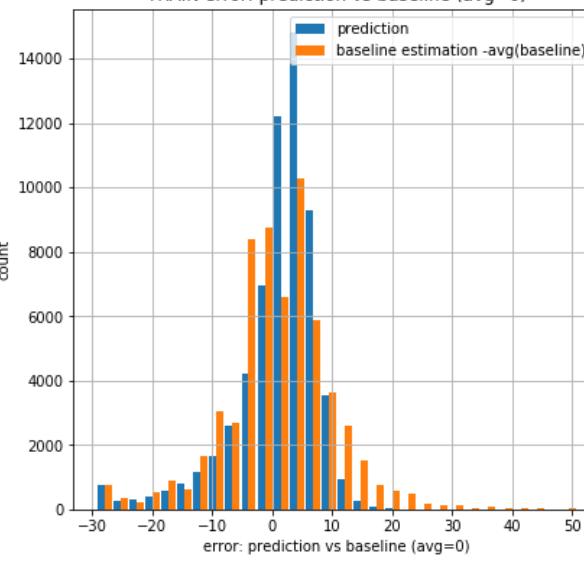
Backup – Generalized Linear Regression

45

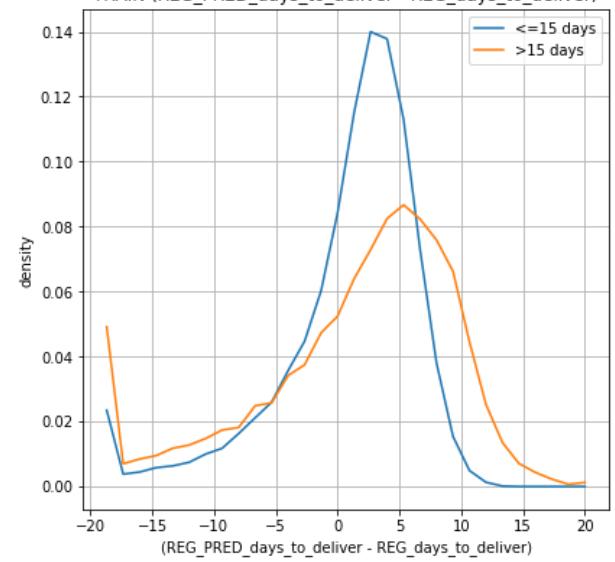
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



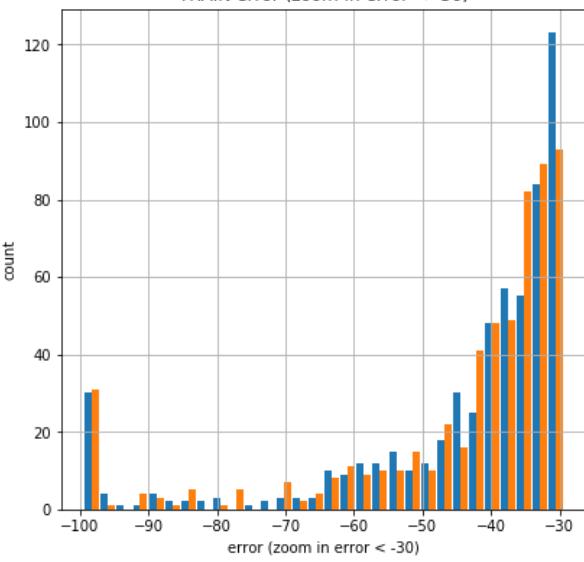
TRAIN error: prediction vs baseline (avg=0)



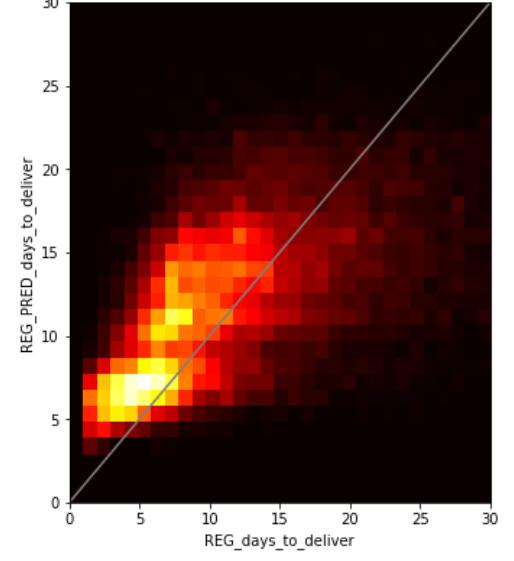
TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)



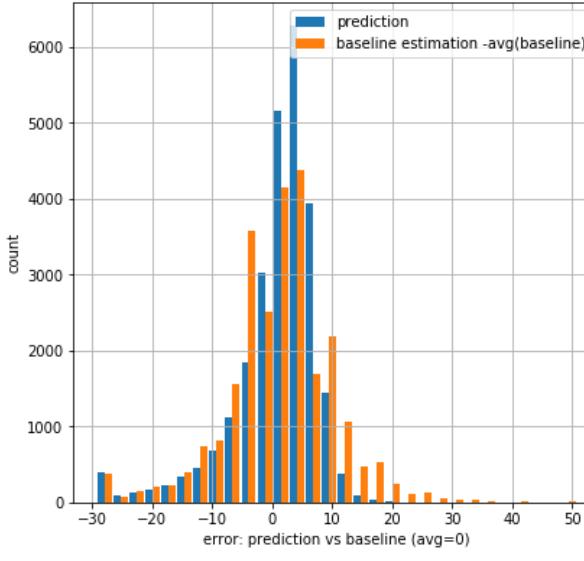
TRAIN error (zoom in error < -30)



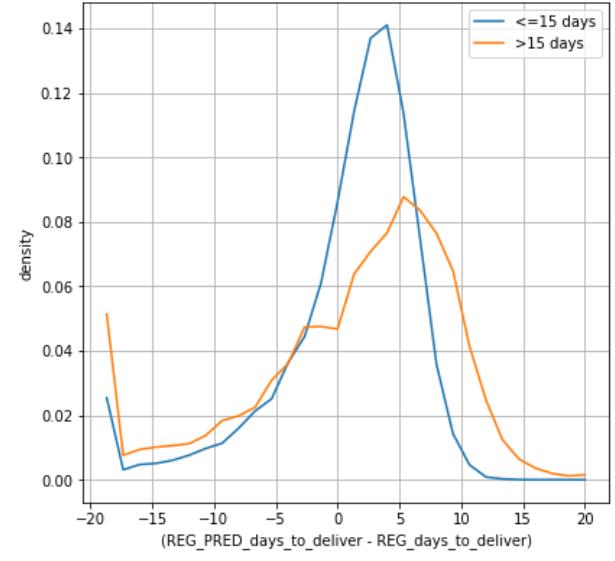
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



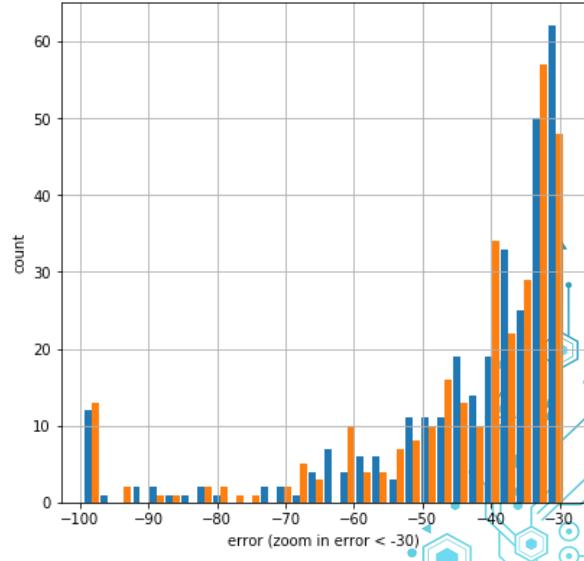
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)

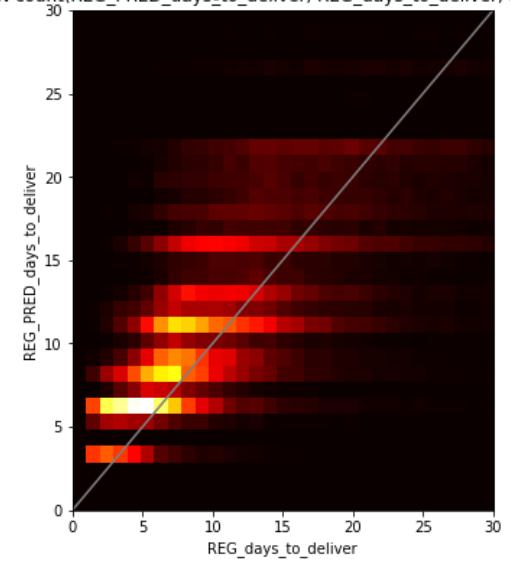


TEST error (zoom in error < -30)

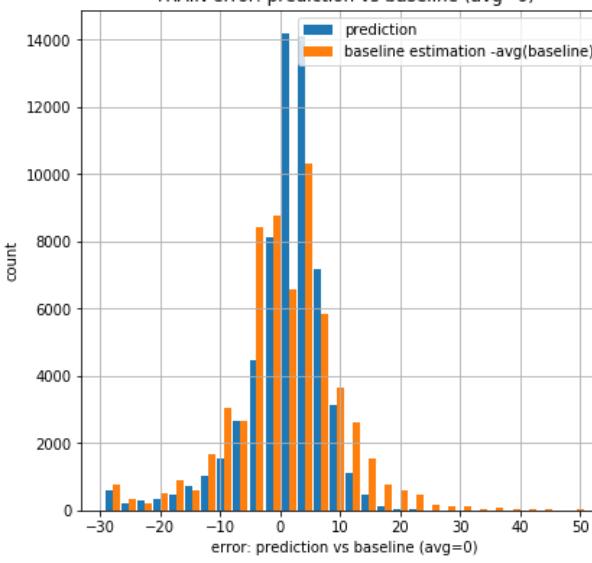


Backup – Decision Tree

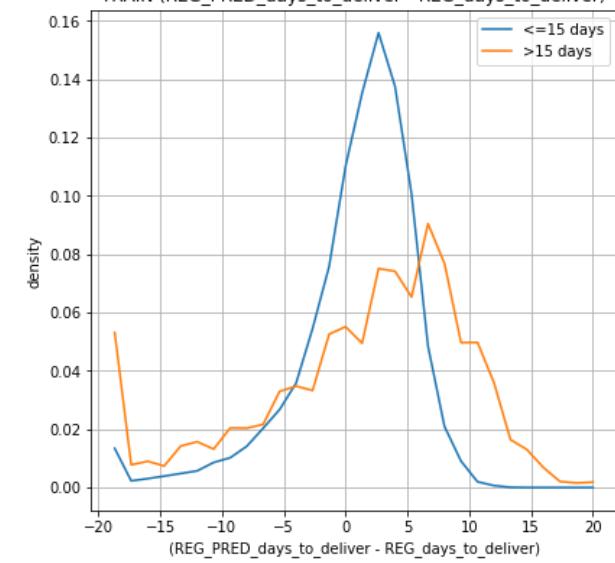
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



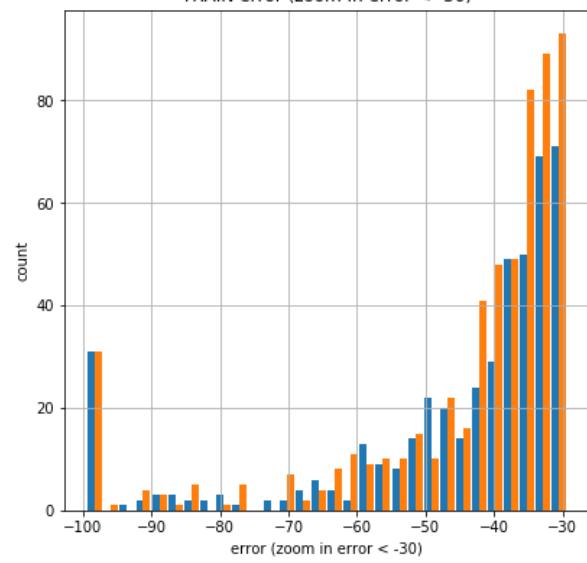
TRAIN error: prediction vs baseline (avg=0)



TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)

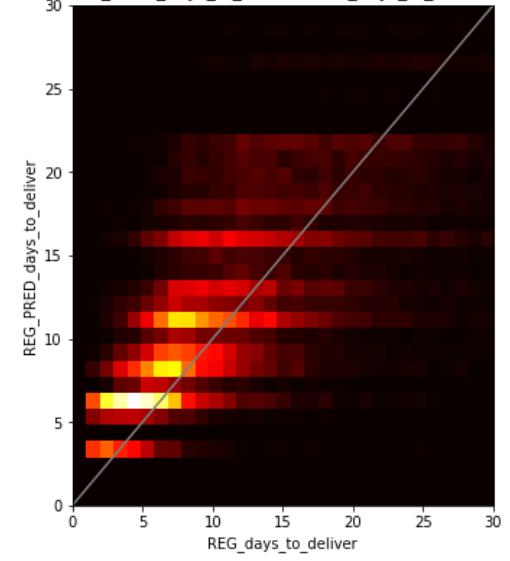


TRAIN error (zoom in error < -30)

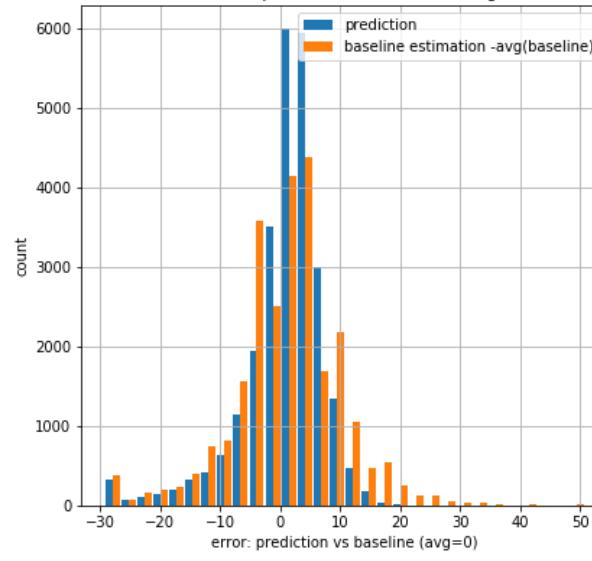


46

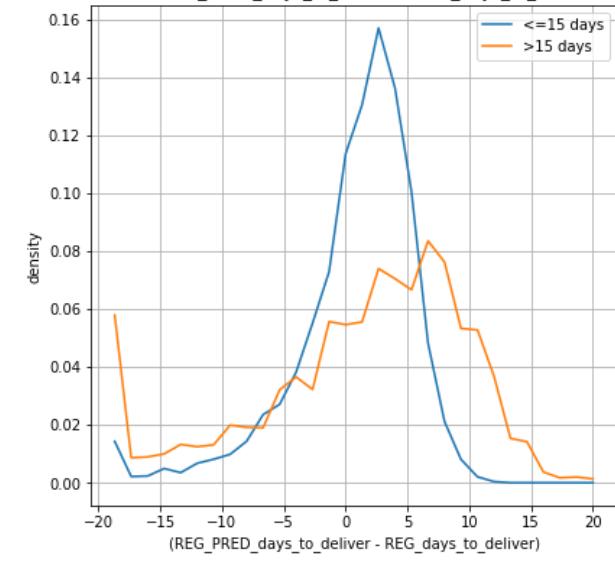
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



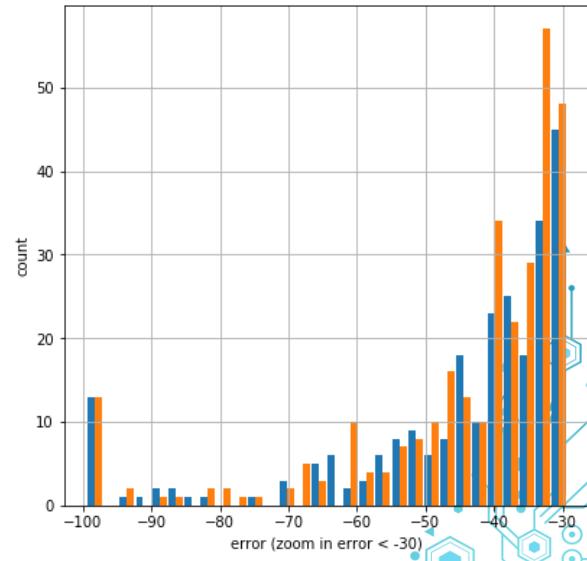
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)

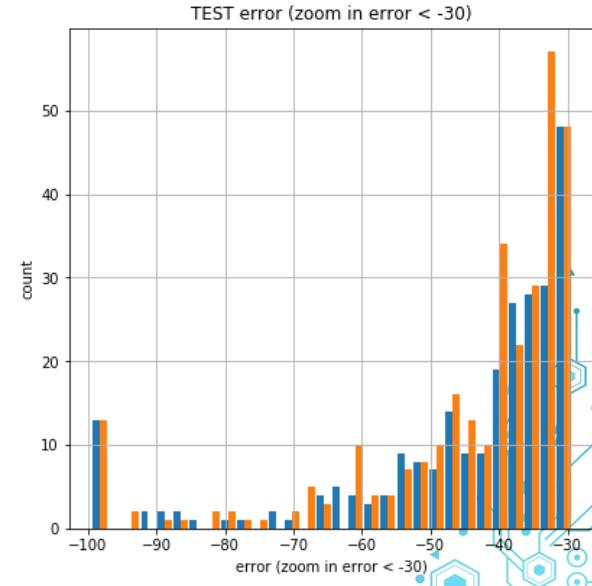
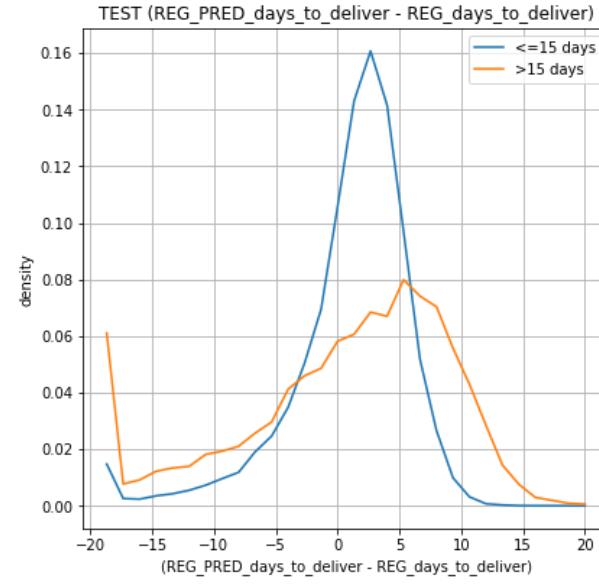
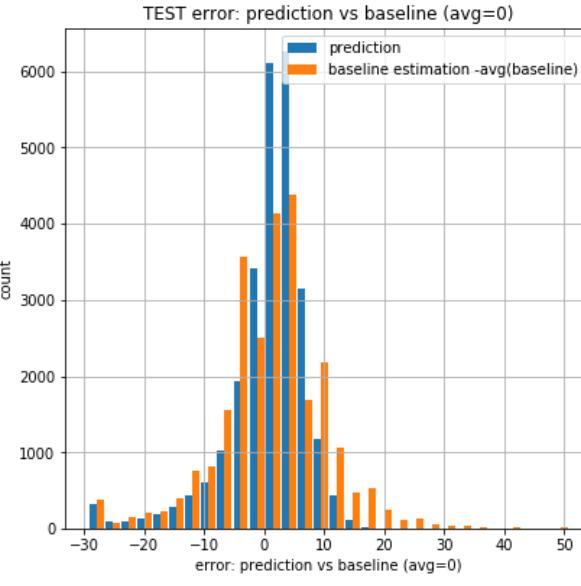
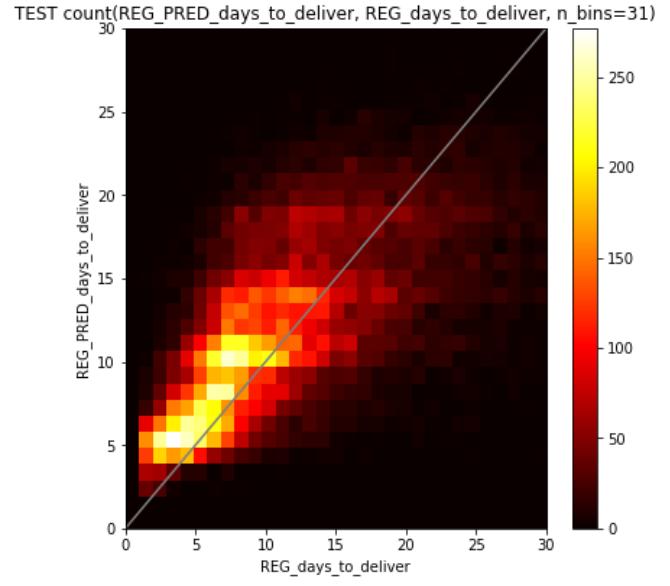
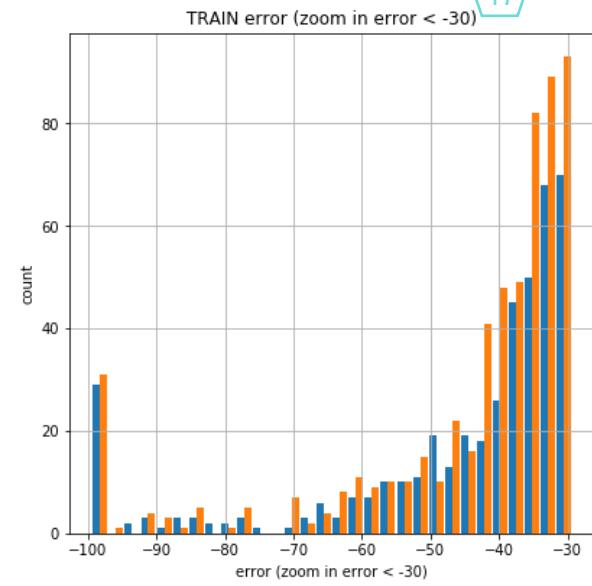
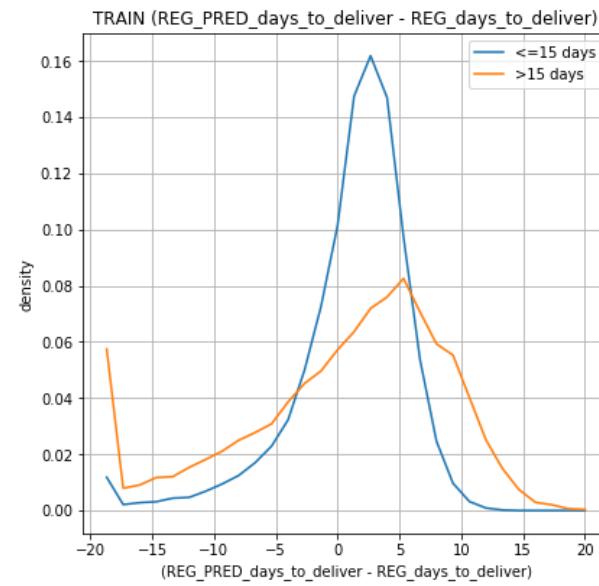
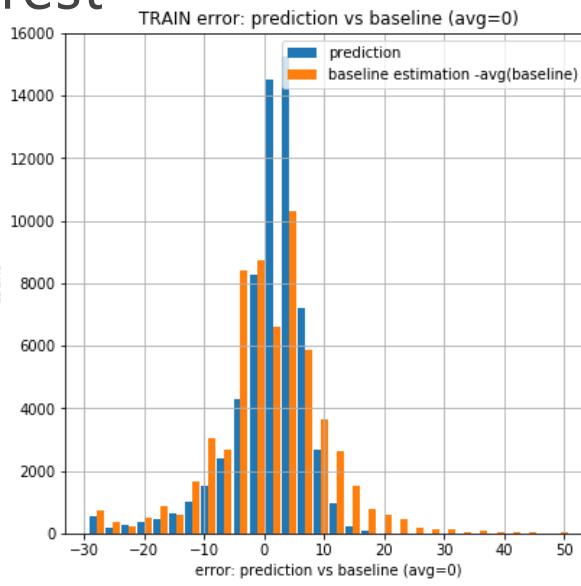
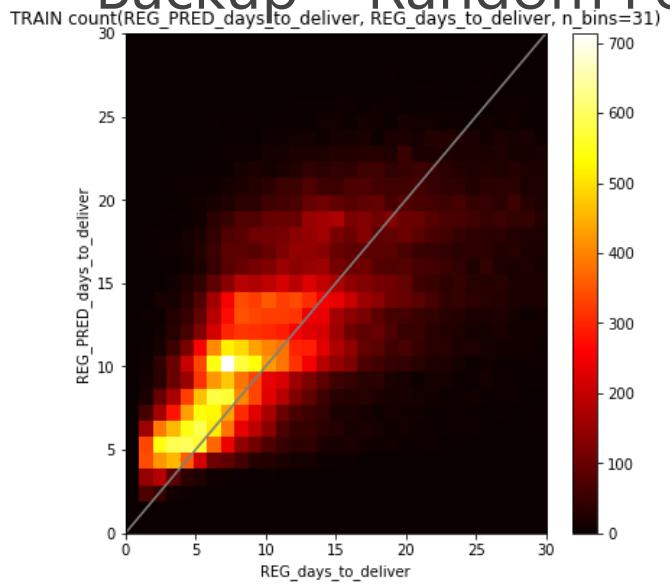


TEST error (zoom in error < -30)



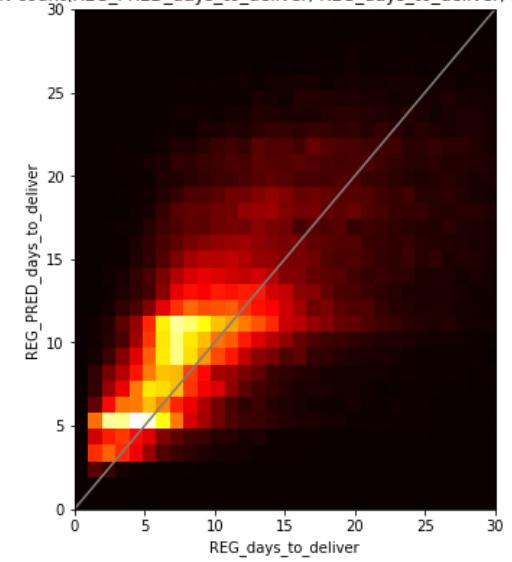
Backup – Random Forest

47

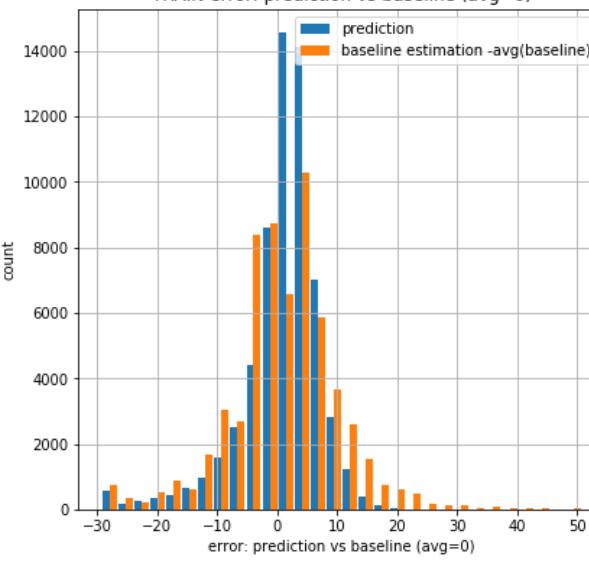


Backup – Gradient Boost

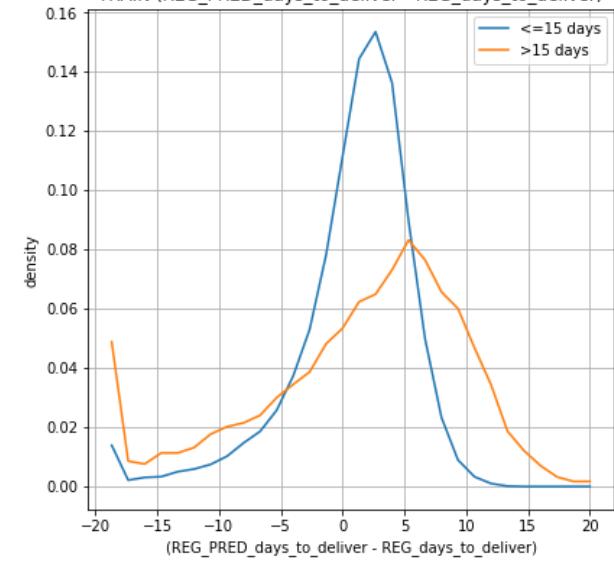
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



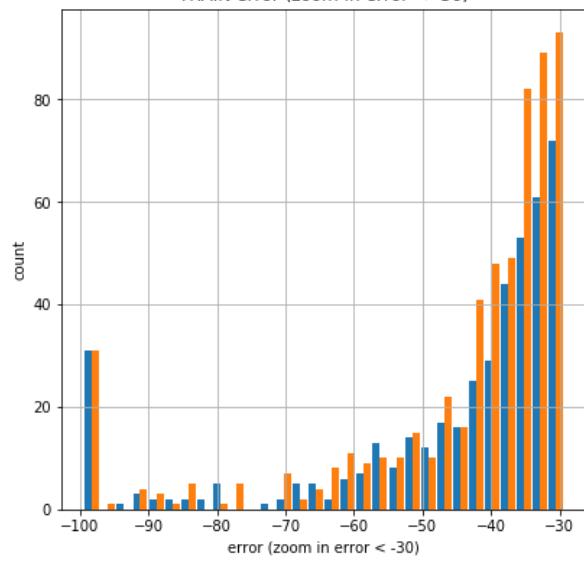
TRAIN error: prediction vs baseline (avg=0)



TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)

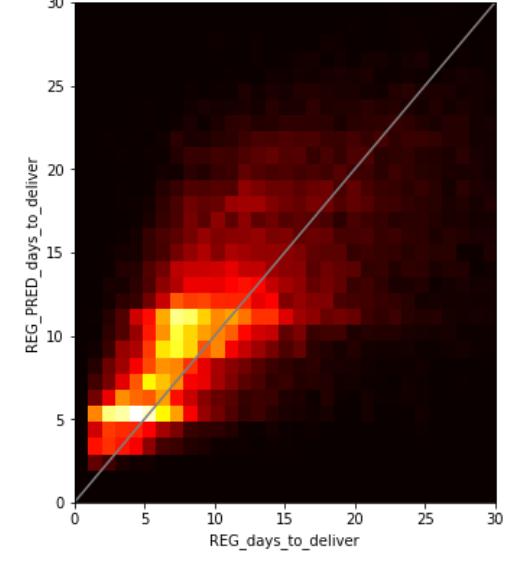


TRAIN error (zoom in error < -30)

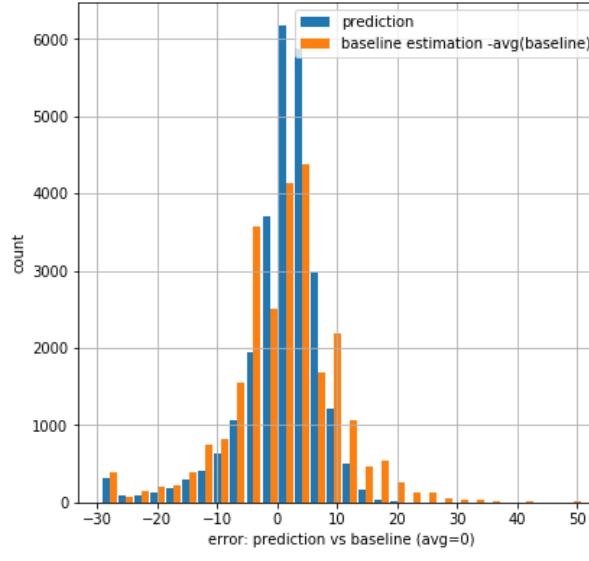


48

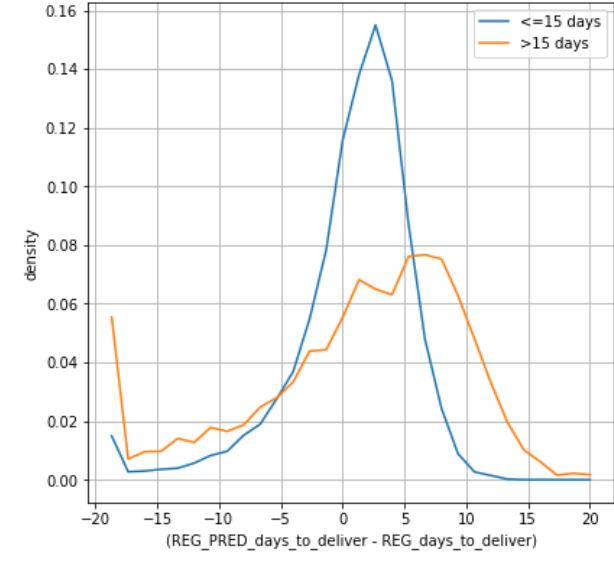
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



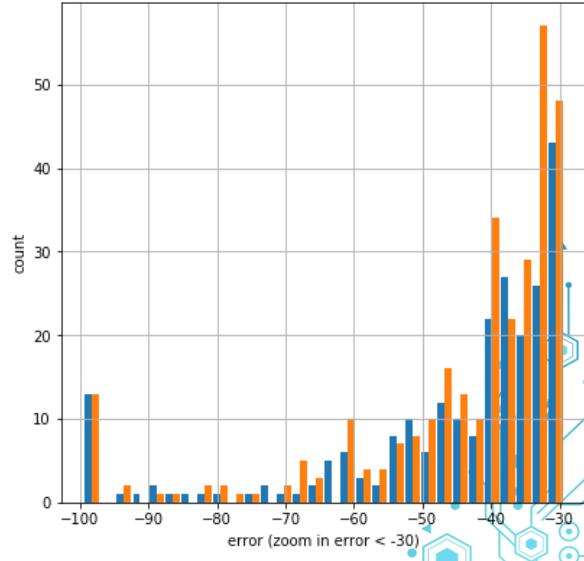
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)

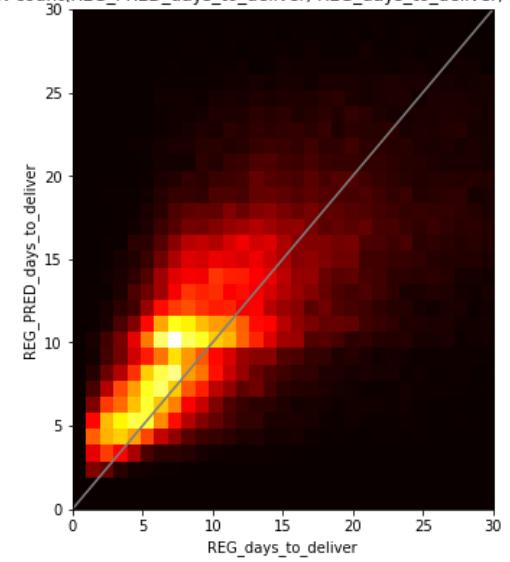


TEST error (zoom in error < -30)

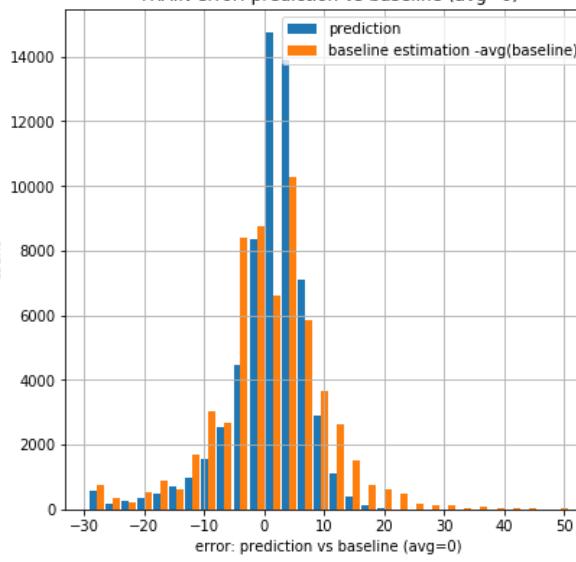


Backup – Deep Learning (MLP Regressor)

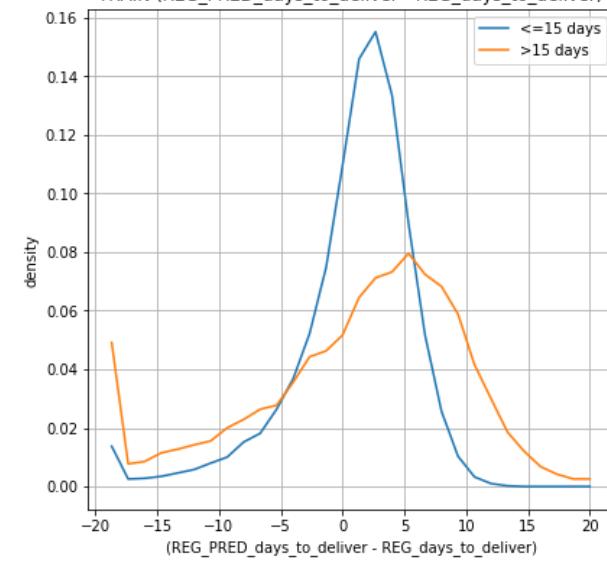
TRAIN count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



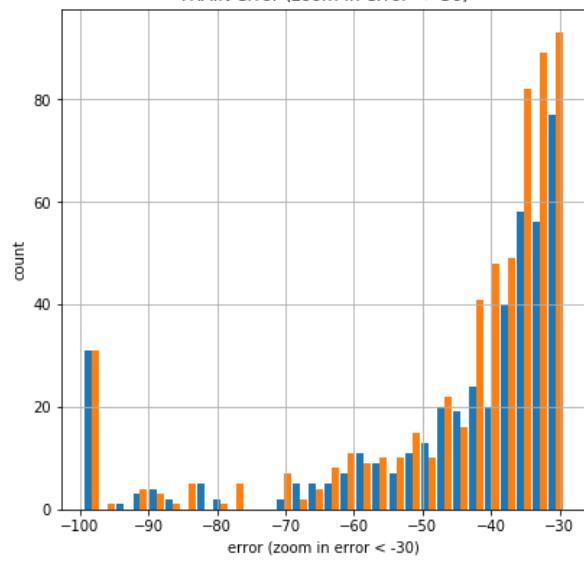
TRAIN error: prediction vs baseline (avg=0)



TRAIN (REG_PRED_days_to_deliver - REG_days_to_deliver)

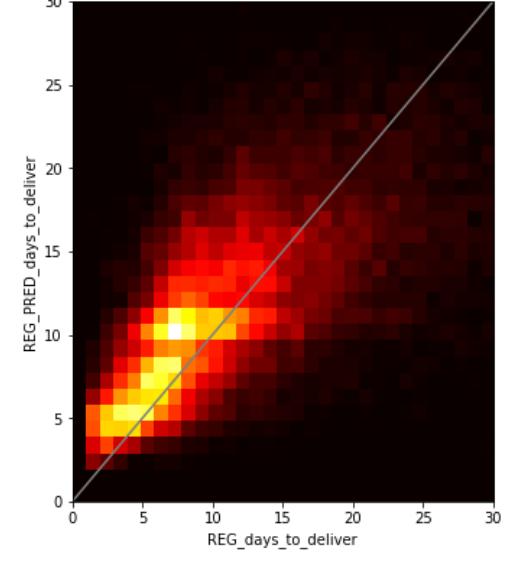


TRAIN error (zoom in error < -30)

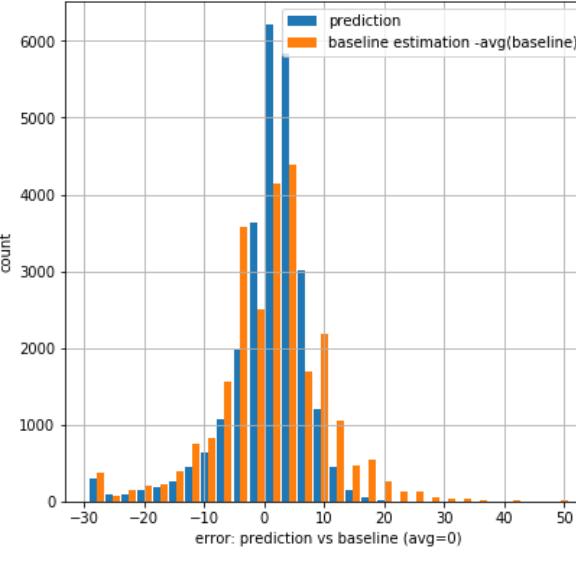


49

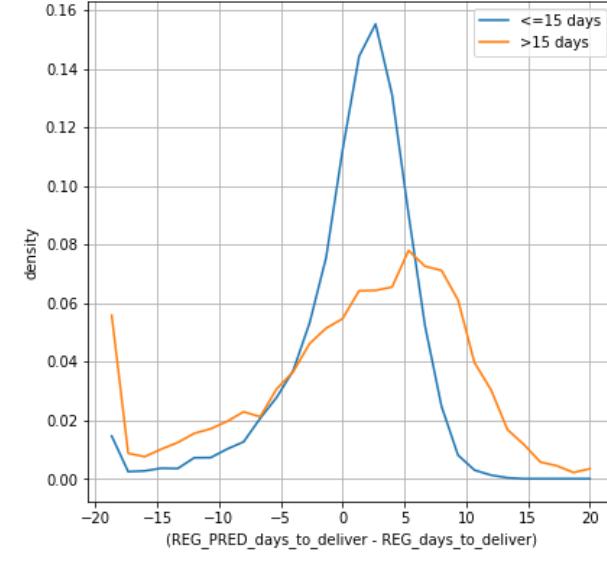
TEST count(REG_PRED_days_to_deliver, REG_days_to_deliver, n_bins=31)



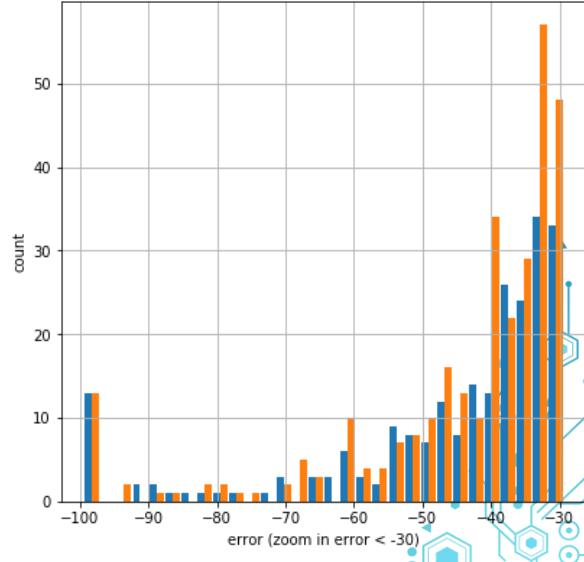
TEST error: prediction vs baseline (avg=0)



TEST (REG_PRED_days_to_deliver - REG_days_to_deliver)



TEST error (zoom in error < -30)



Backup

1. Method	2. rmse train	3. rmse test	4. mse train	5. mse test	6. r2 train	7. r2 test	8. mae train	9. mae test
baseline	10.099	10.397	101.987	108.106	-0.131	-0.104	6.807	6.863
naive	9.498	9.894	90.203	97.895	0	0	6.422	6.542
linear_regression	8.327	8.67	69.335	75.168	0.231	0.232	5.226	5.281
generalized_linear_regression	8.327	8.67	69.335	75.169	0.231	0.232	5.226	5.281
decision_tree_regression	7.908	8.298	62.54	68.852	0.307	0.297	4.871	4.952
random_forest_regression	7.622	8.174	58.087	66.819	0.356	0.317	4.639	4.813
gradient_boost_regression	7.732	8.153	59.78	66.477	0.337	0.321	4.727	4.818

1. Method	2. rmse train (delta)	3. rmse test (delta)	4. mse train (delta)	5. mse test (delta)	6. r2 train (delta)	7. r2 test (delta)	8. mae train (delta)	9. mae test (delta)
baseline	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (-0.00%)	0.000000 (-0.00%)	0.000000 (0.00%)	0.000000 (0.00%)
naive	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)	0.000000 (0.00%)
linear_regression	0.080000 (0.92%)	0.070000 (0.78%)	1.260000 (1.85%)	1.160000 (1.57%)	-0.010000 (-5.71%)	-0.010000 (-4.92%)	0.060000 (1.24%)	0.060000 (1.23%)
generalized_linear_regression	0.080000 (0.92%)	0.070000 (0.78%)	1.260000 (1.85%)	1.160000 (1.57%)	-0.010000 (-5.71%)	-0.010000 (-4.92%)	0.060000 (1.24%)	0.060000 (1.23%)
decision_tree_regression	-0.080000 (-1.05%)	-0.070000 (-0.80%)	-1.330000 (-2.08%)	-1.120000 (-1.60%)	0.020000 (5.14%)	0.010000 (4.21%)	-0.080000 (-1.56%)	-0.070000 (-1.32%)
random_forest_regression	0.240000 (3.22%)	0.110000 (1.34%)	3.560000 (6.53%)	1.760000 (2.70%)	-0.040000 (-10.10%)	-0.020000 (-5.37%)	0.140000 (3.16%)	0.080000 (1.65%)
gradient_boost_regression	0.140000 (1.86%)	0.080000 (1.04%)	2.160000 (3.76%)	1.370000 (2.11%)	-0.020000 (-6.65%)	-0.010000 (-4.18%)	0.070000 (1.59%)	0.060000 (1.26%)



Backup

51

Cluster 0

1. Method	2. rmse train	3. rmse test	4. mse train	5. mse test	6. r2 train	7. r2 test	8. mae train	9. mae test
baseline	9.102	9.297	82.842	86.433	-0.187	-0.176	6.287	6.267
naive	8.354	8.572	69.795	73.473	0	0	5.694	5.735
linear_regression	7.327	7.548	53.685	56.968	0.231	0.225	4.621	4.615
generalized_linear_regression	7.327	7.548	53.685	56.968	0.231	0.225	4.621	4.615
decision_tree_regression	6.914	7.118	47.798	50.673	0.315	0.31	4.29	4.272
random_forest_regression	6.63	6.937	43.954	48.123	0.37	0.345	4.062	4.097
gradient_boost_regression	6.754	7.012	45.619	49.162	0.346	0.331	4.148	4.161

Cluster 1

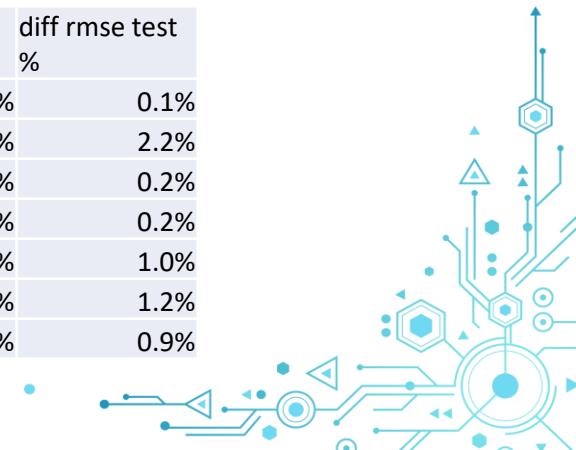
baseline	11.42	11.908	130.409	141.8	-0.16	-0.129	7.527	7.745
naive	10.604	11.209	112.449	125.639	0	0	7.121	7.4
linear_regression	9.615	10.142	92.449	102.859	0.178	0.181	6.1	6.283
generalized_linear_regression	9.615	10.142	92.449	102.861	0.178	0.181	6.1	6.283
decision_tree_regression	9.092	9.69	82.667	93.901	0.265	0.252	5.605	5.859
random_forest_regression	8.763	9.599	76.784	92.137	0.317	0.266	5.382	5.756
gradient_boost_regression	8.848	9.523	78.278	90.695	0.304	0.278	5.427	5.725

Sem agrupamento

baseline	10.099	10.397	101.987	108.106	-0.131	-0.104	6.807	6.863
naive	9.498	9.894	90.203	97.895	0	0	6.422	6.542
linear_regression	8.327	8.67	69.335	75.168	0.231	0.232	5.226	5.281
generalized_linear_regression	8.327	8.67	69.335	75.169	0.231	0.232	5.226	5.281
decision_tree_regression	7.908	8.298	62.54	68.852	0.307	0.297	4.871	4.952
random_forest_regression	7.622	8.174	58.087	66.819	0.356	0.317	4.639	4.813
gradient_boost_regression	7.732	8.153	59.78	66.477	0.337	0.321	4.727	4.818

Diferença

	diff mse train	diff mse test		diff rmse train	diff rmse test	diff rmse train %	diff rmse test %
100.6 108.0	1.38	0.14	10.0 10.4	0.07	0.01	0.7%	0.1%
85.8 93.8	4.44	4.14	9.3 9.7	0.24	0.21	2.6%	2.2%
68.3 74.8	1.07	0.36	8.3 8.6	0.06	0.02	0.8%	0.2%
68.3 74.8	1.07	0.36	8.3 8.6	0.06	0.02	0.8%	0.2%
60.9 67.5	1.63	1.37	7.8 8.2	0.10	0.08	1.3%	1.0%
56.3 65.2	1.78	1.58	7.5 8.1	0.12	0.10	1.6%	1.2%
57.9 65.3	1.88	1.17	7.6 8.1	0.12	0.07	1.6%	0.9%



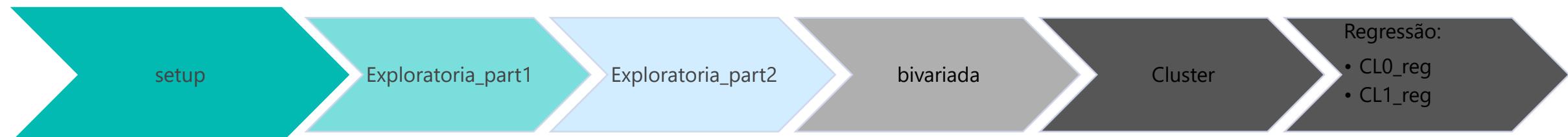
Backup

52



Instruções

- Minhas análises estão distribuídas em vários notebooks, sendo que cada notebook chama o notebook anterior para herdar os resultados dos passos anteriores.
- Para executar qualquer um destes notebook, basta utilizar o "run all" no Databricks.
- Todos os notebooks precisam estar na mesma pasta.



setup

Resumo: Carrega os arquivos CSV em dataframes e verifica a criação dos dataframes

Exploratória – Parte 1

- Resumo:
- Análise de tipos de dados das colunas
 - Verifica quais variáveis são qualitativas ou quantitativas
 - Análise de valores faltantes
 - Verificação de quantidade de valores distintos em cada coluna
 - Proposta de Business Case
 - Preparação dos dados (redução de categorias, tratamento de valores faltantes)

Exploratória – Parte 1

- Resumo:
- Métricas das variáveis

Análise Bivariada

Resumo: Análise bivariada das variáveis
Nota: Neste notebook são instaladas as bibliotecas geopandas e descartes, nas últimas etapas. O objetivo é apenas melhorar as visualizações

Análise de cluster

Resumo: Execução de K-Means para vários valores e seleção do melhor agrupamento.

Regressão

A regressão foi feita em 2 notebooks: CL0_reg e CL1_reg, referentes a cada um dos grupos identificados no notebook "Cluster"

Resumo: Predições e análises comparando com a base de teste e com o baseline.

Técnicas adotadas:
Regressão Linear,
Regressão Linear Generalizada, Decision Tree, Random Forest e
• Gradient Boost



Nota:

Além dos notebooks listados no slide anterior, também foi criado um notebook "library", que contém códigos de uso geral dos demais notebooks. Dentre várias funções de uso geral, este notebook também possui os códigos que mostram como os gráficos são implementados de forma distribuída usando Spark, afim de não sobrecarregar o ambiente do Pandas

Nota 1: As constantes globais estão no início deste notebook. Para utilizar os resultados armazenados em Parquet (caso existam), alterar o valor da constante GLOBAL_GET_FROM_PARQUET para "True"

Nota 2: As configurações dos algoritmos de regressão estão implementadas dentro deste notebook.

