



Prevendo a venda de carros no Brasil

A verdade sobre um projeto de
Machine Learning

Jessica Cabral Carvalho
Novembro, 2019

QUEM SOU EU

- Cientista de Dados na OMOTOR
- Técnica de Informática pela ETEC de Ferraz de Vasconcelos
- Graduação pela Fatec Mogi das Cruzes – Julho/16
- Aluna Especial no Mestrado na EACH-USP
Área de Pesquisa: Inteligência Artificial
- Especializações
 - Formação Cientista de Dados
 - Formação Inteligência Artificial
 - Formação Análise Estatística para Cientista de Dados



/jessica-cabral-carvalho/



/jcabralc



<https://jcabralc.wordpress.com>





OMOTOR

O **OMOTOR** é uma startup focada em inovações baseadas em Inteligência Artificial.

- ChatBots (Cognição via Whatsapp, Telegram, Skype, voz...)
- Processamento de Linguagem Natural
- Visão Computacional
- Modelos Preditivos (Machine Learning)



OPTUM® ABSOLUT. quod

MULTILASER

Pernod Ricard

DHL

OMOTOR

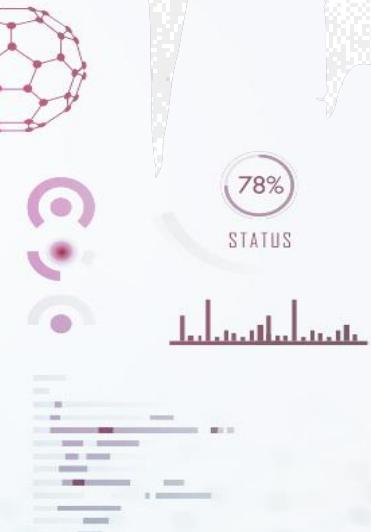
www.omotor.com.br

Agenda

- Quem sou eu
- OMotor
- Prevendo a venda de carros no Brasil
 - O problema de Negócio
 - Desafios
 - Timeline da Solução
 - Ferramentas de Modelagem
 - Amazon Forecast
- Problemas
 - Os malditos dados
 - Modelagem
- Resultados

Prevendo a venda de carros no Brasil

EVOLUTION



O Problema de Negócio

- IHS Markit - Provedora global de informações.
- Imprecisão e falta de informações a nível Brasil
- Descentralização das informações

Solução Proposta:

- Criação de um modelo de previsão das vendas de carros no Brasil utilizando Inteligência Artificial.

Desafios

Qual pergunta queremos responder?

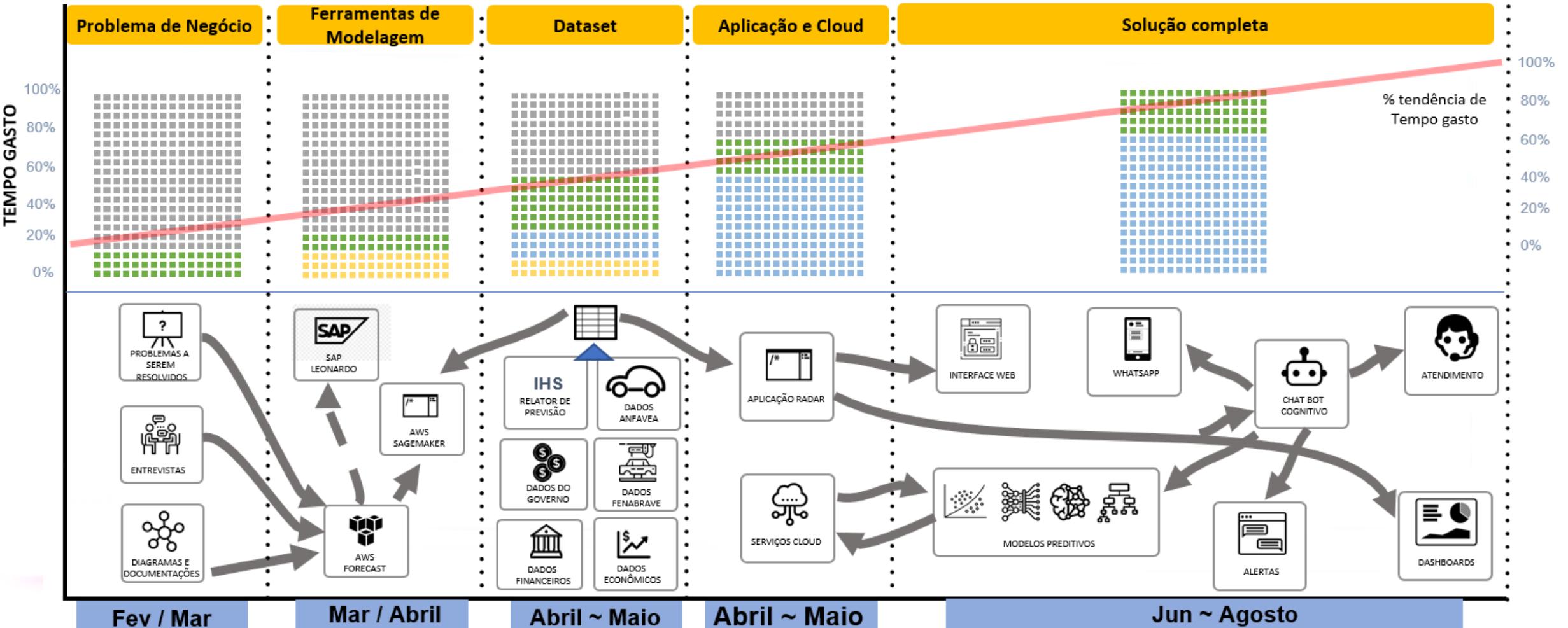


Desafios

Fonte de Dados



Timeline da solução



FERRAMENTAS DE MODELAGEM

- Chegamos a testar o **SAP Leonardo** mas a solução não atendia os requisitos do problema
- H2O.ai não possuía módulos para treinamento de series temporais (na versão que testamos). Utilizamos para testar a hipótese de resolver o problema com algoritmos mais simples (sem uso de Deep Learning).
- Fomos a primeira startup do Brasil a usar e testar o **Amazon Forecast**!
- Decidimos utilizar o Algoritmo próprio da AWS **DeepAR**
- A solução final ficou foi produzida utilizando o **Amazon Sagemaker**



Amazon Forecast



Amazon SageMaker



Amazon Forecast

The screenshot shows the AWS Amazon Forecast landing page. The URL in the browser is aws.amazon.com/pt/forecast/. The page features the AWS logo and navigation links for Produtos, Soluções, Definição de preço, Documentação, Aprenda, Rede de parceiros, AWS Marketplace, Capacitação de clientes, Explore mais, and a search bar. The main content area is titled "Amazon Forecast" and includes tabs for Visão geral, Recursos, Definição de preço, Perguntas frequentes, Recursos, and Clientes. The "Visão geral" tab is currently selected.

Machine Learning



A OMOTOR ajuda no desenvolvimento de empresas por meio da IA ao disponibilizar a elas o melhor dos algoritmos de machine learning, técnicas de visão computacional e bots cognitivos que podem se comunicar usando o WhatsApp e outras plataformas.

"Na OMOTOR, usamos a IA para inovar em nome dos nossos clientes. Portanto, o acesso às tecnologias mais avançadas de aprendizado profundo da AWS é essencial para o sucesso deles. O uso do Amazon Forecast nos proporciona a capacidade de criar e refinar várias previsões de dados de séries temporais, sem necessidade de criar e treinar manualmente um modelo todas as vezes. Prevemos vendas reais para os próximos 12 meses para que possamos planejar adequadamente o inventário, estimar a lucratividade futura, monitorar os ganhos e as perdas da participação de mercado, entre outras informações. Isso significa que podemos usar dados mais contextuais, otimizar com maior frequência, gerar previsões com 50% a mais de melhorias, além de operar com grande velocidade. Por exemplo, ajudamos clientes na indústria automotiva a prever as vendas de 185 veículos no Brasil."

Marcio Rodrigues, CEO — OMOTOR

<https://aws.amazon.com/pt/forecast/>

mento	C	D	E	F	G	
	N_NomeSubSegmento	Montadora	N_ModeloParaModelagem	MarktShare	N_MesAtualForecast	N_M
S	SEDANS GRANDES	BMW	320	0,0468	46	SEDAN
S	SEDANS GRANDES	HYUNDAI	AZERA	0,3432	495	SEDAN
S	SEDANS GRANDES	MBENZ	CLASSE C	0,0108	196	SEDAN
S	SEDANS GRANDES	FORD	FUSION	0,048	830	SEDAN
S	SEDANS GRANDES	KIA	MAGENTIS	0,0902	122	SEDAN
S	SEDANS GRANDES	VW	PASSAT	0,0878	101	SEDAN
S	SEDANS MEDIOS	PEUGEOT	307 SEDAN	0,0206	347	SEDAN
S	SEDANS MEDIOS	VW	BORA	0,0121	339	SEDAN
S	SEDANS MEDIOS	CITROEN	C4 SEDAN	0,1017	1909	SEDAN
S	SEDANS MEDIOS	KIA	CERATO	0,0072	106	SEDAN
S	SEDANS MEDIOS	HONDA	CIVIC	0,3237	5408	SEDA
S	SEDANS MEDIOS	TOYOTA	COROLLA	0,1767	3937	SEDA
S	SEDANS MEDIOS	FORD	FOCUS SEDAN	0,0328	524	SEDA
S	SEDANS MEDIOS	VW	JETTA	0,0211	420	SEDA
S	SEDANS MEDIOS	RENAULT	MEGANE	0,0478	682	SEDA
S	SEDANS MEDIOS	NISSAN	SENTRA	0,0424	552	SEDA
S	SEDANS MEDIOS	GM	VECTRA	0,1544	2740	SEDA
S	HATCH MEDIOS	GM	ASTRA	0,1933	2485	HAT
S	HATCH MEDIOS	FORD	FOCUS	0,0945	1073	HAT
S	HATCH MEDIOS	VW	GOLF	0,1279	1615	HAT
S	HATCH MEDIOS	FIAT	PUNTO	0,3427	3610	HAT
S	SEDANS PEQUENOS	GM	ASTRA SEDAN	0,2304	687	SE
S	SEDANS PEQUENOS	GM	CLASSIC	0,2942	10204	VE
S	SEDANS PEQUENOS	FORD	FIESTA SEDAN	0,0988	3448	SE
S	SEDANS PEQUENOS	RENAULT	LOGAN	0,1035	3498	SE
S	SEDANS PEQUENOS	VW	POLO SEDAN	0,7684	2584	SI
S	SEDANS PEQUENOS	GM	PRISMA	0,1441	4573	SI
S	SEDANS PEQUENOS	FIA	SIENA	0,2709	8632	S
SUV MED'Ô	GM		BLAZER	0,0198	153	S
SUV MED'Ô	LAND ROVER		FREELANDER	0,0144	147	S
SUV MÉDIO	NISSAN	IBISHI	OUTLANDER	0,0108		
			SPORTAGE	0,0425		
		DAIHATSU	TUCSON	0,1468		
		FIAT	C3	0,1099		
		LTI	CLIO	0,0135		
		CO		0,1461		
				0,1789		

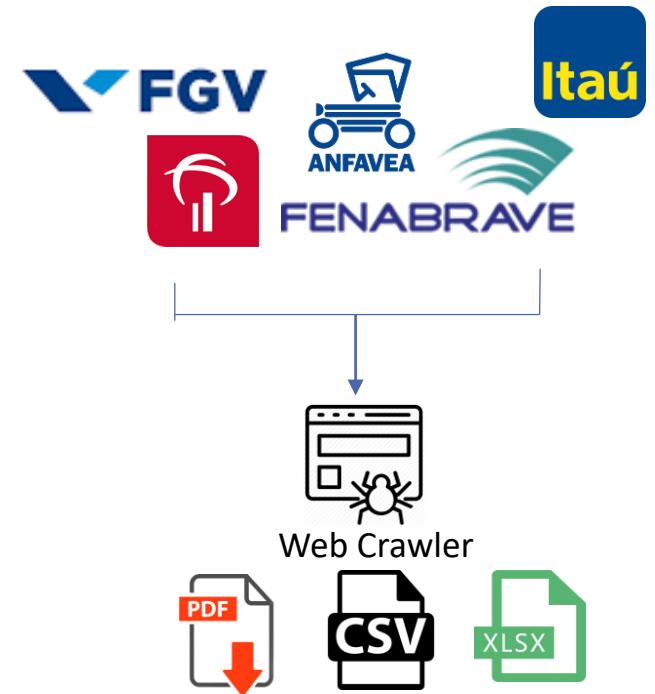
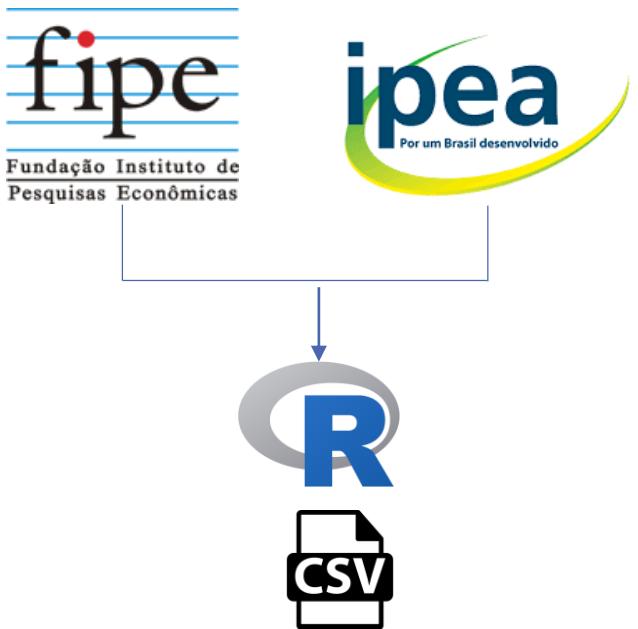
Problemas!

Os malditos Dados

planilhas, PDFs, webscrapping, APIs...



Formato dos dados



Extração dos dados - Fenabrade

DADOS DE
MERCADO
FENABRAVE

Ed. 202
Informativo - Emplacamentos
São Paulo, Novembro de 2019

Modelos mais emplacados acumulado até Out/2019

AUTOMÓVEIS

Veículos de Entrada

	Modelo	2019 Set	2019 Out	2019 Acumulado	Part.
1º	RENAULT/KWID	8.826	6.066	70.935	31,11%
2º	VW/COL	6.850	6.570	66.845	29,22%
3º	FIAT/MOBI	4.377	4.569	45.441	19,93%
4º	FIAT/UNO	1.493	1.527	17.093	7,50%
5º	TOYOTA/ETIOS HB	1.345	1.686	15.215	6,67%
6º	VW/UP	1.161	1.229	11.009	4,83%
7º	CHERY/QQ	22	21	1.360	0,60%
8º	VW/FUSCA	6	7	34	0,01%
9º	FIAT/PALIO	0	1	33	0,01%
Total		24.095	21.688	227.987	100%

Hatch Pequenos

	Modelo	2019 Set	2019 Out	2019 Acumulado	Part.
1º	GM/ONIX	21.043	21.198	200.588	32,61%
2º	FORD/KA	7.890	9.691	86.442	14,05%
3º	HYUNDAI/HB20	7.145	8.332	85.061	13,96%
4º	FIAT/ARGO	5.730	7.585	62.696	10,19%
5º	VW/POLO	6.282	7.245	58.560	9,52%
6º	RENAULT/SANDERO	5.855	4.288	40.307	6,55%
7º	VW/FOX/CROSS FOX	2.552	2.762	32.409	5,27%
8º	TOYOTA/YARIS HB	2.711	3.192	30.445	4,95%
9º	NISSAN/MARCH	918	685	5.738	0,93%
10º	PEUGEOT/208	240	290	4.795	0,78%
11º	FORD/FIESTA	185	150	3.233	0,53%
12º	CITROËN/C3	153	142	2.641	0,43%
Total		60.336	65.819	615.066	100%

Hatch Médios

	Modelo	2019 Set	2019 Out	2019 Acumulado	Part.
1º	GM/CRUZE HB	241	302	3.957	55,43%
2º	VW/COLF	20	19	1.114	15,60%
3º	FORD/FOCUS	8	7	519	7,27%
4º	M.BENZ/A250	54	8	405	5,67%
5º	VOLVO/V40	15	32	363	5,08%
6º	AUDI/A3	37	32	282	3,95%
Total		453	478	7.139	100%

www.fenabrade.org.br

11

```
1 # -*- coding: utf-8 -*-
2 #"""
3 #Created on Mon Mar 11 16:49:31 2019
4 #
5 #@author: Jéssica
6 #
7 #Relatorio - Fenabrade
8 #"""
9 #
10 #the table will be returned in a list of dataframe,for working with dataframe you need pandas
11 #####Import Labaries GLOBAIS #####
12 import pandas as pd
13 import numpy as np
14 import tabula
15 import datetime
16 import sys
17 import ipdb
18 from unidecode import unidecode
19 #
20 #
21 ##### Seta Diretorios#####
22 #Diretorio geral do projeto
23 DIR_PROJECT_PATH='C:/Users/Omotor2/Desktop/Export-PDF-Fenabrade/'
24 #
25 #Diretorio dos dados
26 DIR_DATA_PATH=DIR_PROJECT_PATH+'data/'
27 #
28 #Diretorio dos logs
29 DIR_LOG_PATH=DIR_PROJECT_PATH+'log/'
30 #
31 #Diretorio dos Sida
32 DIR_SAIDA_PATH=DIR_PROJECT_PATH+'Resultado/'
33 #
34 #### FUNCAO: Lista os arquivos contidos em um diretorio recursivamente
35 ## PARAMETROS ENTRADA:
36 ## RETORNO:
37 #####
38 #####
39 def f_ListafilesFromFolder(Path='.',Extension='.*', Reverse = False):
40     ## Importa library necessarias para a funcao
41     import os
42     from glob import glob
43 #
44 #
45     ## processo de listagem dos arquivos
46     result = [y for x in os.walk(Path) for y in glob(os.path.join(x[0],Extension))]
47     result.sort(reverse=Reverse)
48     return result
49 #
50 #####
51 #### FUNCAO: Função que log e imprime o acompanhamento da execução
52 ## PARAMETROS ENTRADA:
53 ## RETORNO:
54 #####
55 #####
```

 python™

1

11 Anos de dados
131 PDFs
17 tabelas em arquivos
2.227 tabelas para extração
PDF com formatos diferentes



Extração dos dados - FIPE

```

Extrato_preco_carro_Fipe.R
459   'GM - Chevrolet',
460   'VW - Volkswagen'
461
462 # ATENÇÃO
463
464 # DUCATO está vindo no ARGO
465
466
467 carro = "UNO"
468 montadora = "FIAT"
469 ano_corrente = as.integer(format(sys.date(), "%Y")) #2019
470
471 anos <- c("0", toString(ano_corrente), toString(ano_corrente-1), toString(ano_corrente-2))
472
473 # Extração carro individual
474 df2 = fipe_carro(carro, montadora, anos, data2019)
475
476
477 carros_montadoras = Map(c, carros, montadoras)
478
479 # Extração mes a mes
480 #anos <- c("0", "2019", "2018")
481
482 #df_2019e <- data.frame()
483
484 # for (mes in data2008){
485 #   for (car_montadora in carros_montadoras){
486 #     print(paste('Extraindo do modelo e montadora:', car_montadora[1], car_montadora[2], car_montadora[3]))
487 #     df_2008e <- rbind(df_2008e, fipe_carro(car_montadora[1], car_montadora[2], car_montadora[3], mes))
488 #   }
489 # }
490
491 # extração anual
492 df_2019 <- data.frame()
493
494 for (car_montadora in carros_montadoras){
495   print(paste('Extraindo do modelo e montadora:', car_montadora[1], car_montadora[2], car_montadora[3]))
496   df_2019 <- rbind(df_2019, fipe_carro(car_montadora[1], car_montadora[2], car_montadora[3], ano))
497 }
498
499
500 path = sprintf("c:/users/omotor2/desktop/coleta_dados_FIPE/por_ano/%s-all_cars.xlsx", anos)
501 write.xlsx(df_2019, path)
502

```



2015-all_cars.xlsx
2016-all_cars.xlsx
2017-all_cars.xlsx
2018-all_cars.xlsx
2019-all_cars.xlsx

2019-all_cars_all_versions.xlsx

2018-all_cars_filtered.xlsx

2019-all_cars_filtered.xlsx

Script corrige e filtra versões

python™

```

# -*- coding: utf-8 -*-
Created on Thu Aug 15 14:21:16 2019
@author: Jessica

#####
### Junta e Corriga nomenclatura dos dados coletados da FIPE #####
#####

import pandas as pd
import glob
from unidecode import unidecode
import re
import math
from datetime import datetime
import logging

logging.basicConfig(filename='log_corrige_e_filtrar_versoes_fipe.log', filemode='w',
                    format='%(asctime)s - %(levelname)s - %(message)s')

# TODO
# ADD log em tudo

#####
### ATENÇÃO COM
## DUSTER
## FOX/CROSSFOX/SPACEFOX
## RENEGADE SPORT
## ETIOS
## PALIO e WEEKEND
#####

planilha = '{}-all_cars.xlsx'.format(datetime.now().year)

path="c:/users/omotor2/desktop/coleta_dados_FIPE/por_ano/"
path_to_save_versoes = "c:/users/omotor2/desktop/coleta_dados_FIPE/por_carro_Versoes"
path_to_save_todas_versoes = "c:/users/omotor2/desktop/coleta_dados_FIPE/por_carro_Todas_Versoes"

logging.warning('Planilha a ser corrigida: {}'.format(planilha))
#####
## Leitura dos Arquivos
#####

fipe_full = pd.read_excel(path=planilha)

fipe_full['carro'] = [str(carro).upper().strip() for carro in fipe_full['carro']]
fipe_full['data_referencia'] = pd.to_datetime(fipe_full['data_referencia'])
fipe_full['ano'] = fipe_full['ano'].astype(str)

print('Tamanho final do Dataframe: {}'.format(fipe_full.shape))
logging.warning('Tamanho final do Dataframe: {}'.format(fipe_full.shape))

```

Data Preparation dos Dados Coletados via API FIPE

Gera Estatísticas dos Preços

```

[1]: import pandas as pd
import numpy as np
from datetime import datetime

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', -1)

[2]: versao_planilha_todas_versoes = "({}-all_cars_all_versions)".format(datetime.now().year)
versao_planilha_versoes_filtradas = "({}-all_cars_filtered)".format(datetime.now().year)

path = "C:/Users/Omotor2/Desktop/Coleta_dados_FIPE/"
path_all_versions = path+'por_carro_Todas_Versoes/'
path_filtered_versions = path+'por_carro_Versoes_filtradas/'

[3]: # Categorias dos Preços - Baseado no Guia Quatro Rodas
# preco_nivel_1 > 0 - 40.000
# preco_nivel_2 > 40.000 - 56.000
# preco_nivel_3 > 56.000 - 75.000
# preco_nivel_4 > 75.000 - 180.000
# preco_nivel_5 > 180.000 - 250.000
# preco_nivel_6 > 250.000

categorias_preco = [0, 40000, 56000, 75000, 180000, 250000, 320000, 750000]
labels_categorias_preco = ['preco_nivel_1', 'preco_nivel_2', 'preco_nivel_3',
                           'preco_nivel_4', 'preco_nivel_5', 'preco_nivel_6', 'preco_nivel_7']

[4]: lista_cols_default = ['modelo', 'marca', 'ano', 'data_referencia']

```

	A	B	C	D	E	F	G	H
1	data_referencia	marca	modelo	_anterior_ano	anterior_ano	anterior_ano	anterior_ano	anterior_ano
2	2019-01-01 0:00:00	RENAULT	DUSTER	6314,5	76957	67506,44	66694	5579
3	2019-02-01 0:00:00	RENAULT	DUSTER	6661,21	76803	66927,89	66187	5464
4	2019-03-01 0:00:00	RENAULT	DUSTER	7011,18	76649	66494,11	65690	5367
5	2019-04-01 0:00:00	RENAULT	DUSTER	7251,16	76495	66153	65197	5303
6	2019-05-01 0:00:00	RENAULT	DUSTER	7377,13	76342	66785,9	66980,5	5277
7	2019-06-01 0:00:00	RENAULT	DUSTER	7199,58	76196	66302,2	66583,5	5343
8	2019-07-01 0:00:00	RENAULT	DUSTER	7191,23	76051	65634,8	66268	5264
9	2019-08-01 0:00:00	RENAULT	DUSTER	7071,7	75906	64634,4	64782	5205
10	2019-09-01 0:00:00	RENAULT	DUSTER	6950,58	75761	63750,3	63548	5182

950 registros
37 colunas

Dados até
Out/19

Tabelão

```
display(df.head())
display(df.tail())
print(df.shape)
```

	Data	NomeSegmento	N_NomeSubSegmento	Montadora	N_ModeloParaModelagem	MarktShare	N_MesAtualForecast	N_NomeSubSegmento_N
14	2008-06-01	AUTOMOVEIS	SEDANS MEDIOS	RENAULT	MEGANE	0.0478	682.0	SEDANS M
38	2008-06-01	AUTOMOVEIS	HATCH PEQUENOS	VW	GOL	0.3092	24470.0	HATCH PEQU
56	2008-06-01	AUTOMOVEIS	SW MEDIOS	VW	PARATI	0.2556	1489.0	SW M
55	2008-06-01	AUTOMOVEIS	SUV PEQUENO	FORD	ECOSPORT	0.3382	3772.0	SUV PEC
54	2008-06-01	AUTOMOVEIS	MONOCAB	GM	MERIVA	0.2247	2173.0	MON

< >

	Data	NomeSegmento	N_NomeSubSegmento	Montadora	N_ModeloParaModelagem	MarktShare	N_MesAtualForecast	N_NomeSubSegmento_N
13753	2019-09-01	AUTOMOVEIS	SEDANS PEQUENOS	TOYOTA	ETIOS SEDAN	0.0411	883.0	
13751	2019-09-01	AUTOMOVEIS	SEDANS PEQUENOS	NISSAN	VERSA	0.0709	1890.0	
13781	2019-09-01	AUTOMOVEIS	SEDANS GRANDES	TOYOTA	CAMRY	0.0231	1.0	
13782	2019-09-01	AUTOMOVEIS	SEDANS GRANDES	MBENZ	CLASSE E	0.0209	16.0	
13765	2019-09-01	AUTOMOVEIS	SEDANS MEDIOS	KIA	CERATO	0.0157	99.0	

< >

(13843, 161)

Resumão da Análise dos dados

Total de Carros no Dataset

```
print(len(set(df.N_ModeloParaModelagem)))  
  
list(set(df.N_ModeloParaModelagem))  
  
170  
  
['C4 CACTUS',  
 'UP',  
 'FREELANDER',  
 '208',  
 'WR-V',  
 'FIESTA SEDAN',  
 'A5',  
 '207 SEDAN',  
 'C4 PICASSO',  
 'T-CROSS',  
 'C4',  
 'CORSA',  
 'SAVEIRO',  
 'MERIVA',  
 'VOYAGE',  
 'C3',  
 'DUCATO',  
 'TRACKER']
```

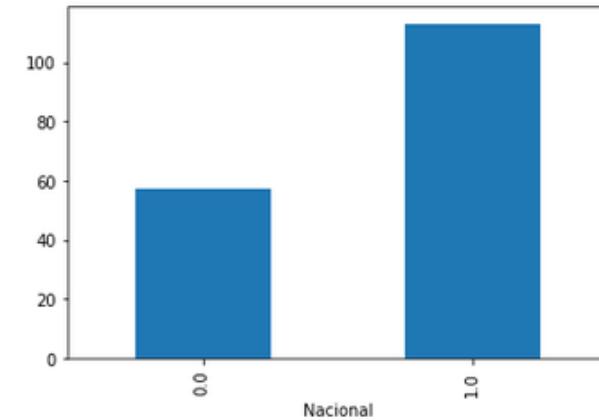
SubSegmentos no Dataset

```
print(len(set(df.N_NomeSubSegmento)))  
  
list(set(df.N_NomeSubSegmento))  
  
19  
  
['MONOCAB',  
 'OFFROAD',  
 'HATCH MEDIOS',  
 "PICK-UP'S MEDIAS",  
 'HATCH PEQUENOS',  
 'SEDANS GRANDES',  
 "PICK-UP'S PEQUENAS",  
 'SEDANS MEDIOS',  
 'FURGOES PEQUENOS',  
 'SW GRANDES',  
 'SUV MEDIO',  
 'SEDANS PEQUENOS',  
 'SPORTS',  
 'SUV PEQUENO',  
 'SUV GRANDE',  
 'GRANDCAB',  
 "PICK-UP'S GRANDES",  
 'SW MEDIOS',  
 'FURGOES']
```

Nacionais vs Importados

```
: print(df.groupby('Nacional')['N_ModeloParaModelagem'].nunique())  
df.groupby('Nacional')['N_ModeloParaModelagem'].nunique().plot(kind='bar')  
plt.show()
```

Nacional
0.0 57
1.0 113
Name: N_ModeloParaModelagem, dtype: int64

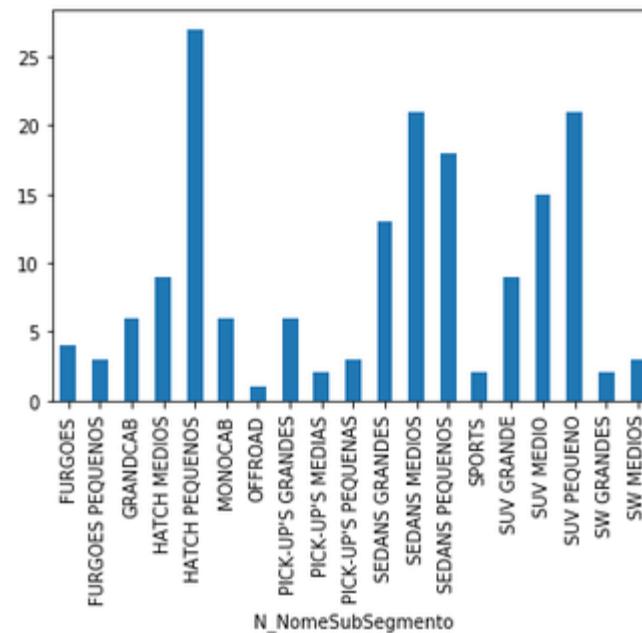


Resumão da Análise dos dados

Total de Carros por SubSegmentos

```
: print(df.groupby('N_NomeSubSegmento')['N_ModeloParaModelagem'].nunique())
df.groupby('N_NomeSubSegmento')['N_ModeloParaModelagem'].nunique().plot(kind='bar')
plt.show()
```

N_NomeSubSegmento	
FURGOES	4
FURGOES PEQUENOS	3
GRANDCAB	6
HATCH MEDIOS	9
HATCH PEQUENOS	27
MONOCAB	6
OFFROAD	1
PICK-UP'S GRANDES	6
PICK-UP'S MEDIAS	2
PICK-UP'S PEQUENAS	3
SEDANS GRANDES	13
SEDANS MEDIOS	21
SEDANS PEQUENOS	18
SPORTS	2
SUV GRANDE	9
SUV MEDIO	15
SUV PEQUENO	21
SW GRANDES	2
SW MEDIOS	3
Name: N_ModeloParaModelagem, dtype: int64	



Resumão da Análise dos dados

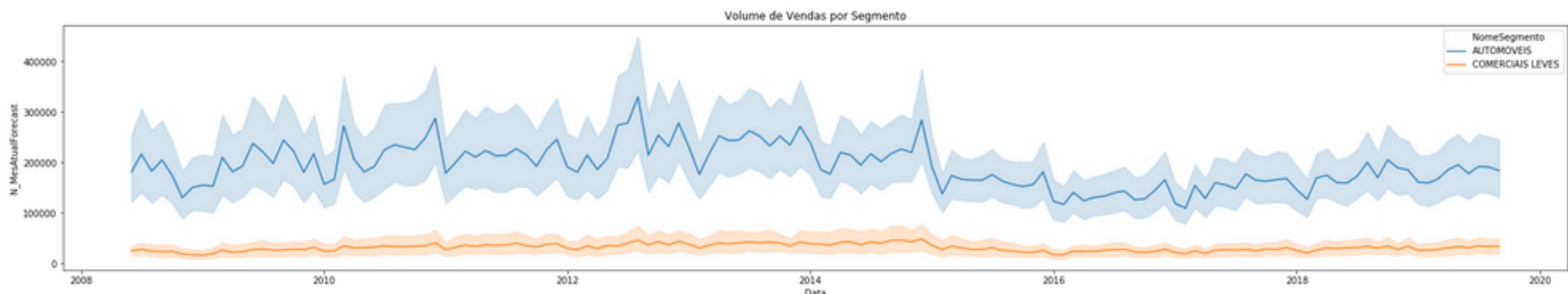
Volume de vendas por segmento

```
fig, ax1 = plt.subplots()
fig.set_size_inches(30, 5)

plt.title('Volume de Vendas por Segmento')
sns.lineplot(x="Data", y="N_MesAtualForecast", data=df, ax=ax1, color='green', hue='NomeSegmento', estimator='sum')

#plt.savefig(path_save_plots+os.sep+'Volume_de_Vendas_por_Segmento.png', dpi=200)
plt.savefig('Volume_de_Vendas_por_Segmento.png', dpi=200)

plt.show()
```



Problemas nos dados

- Muitas fontes e formatos diferentes, e pior...todas abertas
 - Difícil encontrar a informação
 - Sites do governo são complicados de entender e interagir
 - Incrível quantidade de inconsistência nos dados públicos
 - Dependência na liberação da informação
- Alta demanda de criação de novas features (Feature Engineering)
 - Facelift (100% manual, não existe fonte dessa informação em nenhum lugar)
 - Preço
 - Markshare
- Inconsistência dos dados oficiais de vendas

Segmento	C	D	E	F	G
	N_NomeSubSegmento	Montadora	N_ModeloParaModelagem	MarktShare	N_MesAtualForecast
S	SEDANS GRANDES	BMW	320	0,0468	46 SEDAN
S	SEDANS GRANDES	HYUNDAI	AZERA	0,3432	495 SEDAN
S	SEDANS GRANDES	MBENZ	CLASSE C	0,0108	196 SEDAN
S	SEDANS GRANDES	FORD	FUSION	0,048	830 SEDAN
S	SEDANS GRANDES	KIA	MAGENTIS	0,0902	122 SEDAN
S	SEDANS GRANDES	VW	PASSAT	0,0878	101 SEDAN
S	SEDANS MEDIOS	PEUGEOT	307 SEDAN	0,0206	347 SEDAN
S	SEDANS MEDIOS	VW	BORA	0,0121	339 SEDAN
S	SEDANS MEDIOS	CITROEN	C4 SEDAN	0,1017	1909 SEDAN
S	SEDANS MEDIOS	KIA	CERATO	0,0072	106 SEDAN
S	SEDANS MEDIOS	HONDA	CIVIC	0,3237	5408 SEDA
S	SEDANS MEDIOS	TOYOTA	COROLLA	0,1767	3937 SEDA
S	SEDANS MEDIOS	FORD	FOCUS SEDAN	0,0328	524 SEDA
S	SEDANS MEDIOS	VW	JETTA	0,0211	420 SED/
S	SEDANS MEDIOS	RENAULT	MEGANE	0,0478	682 SED/
S	SEDANS MEDIOS	NISSAN	SENTRA	0,0424	552 SED
S	SEDANS MEDIOS	GM	VECTRA	0,1544	2740 SED
S	HATCH MEDIOS	GM	ASTRA	0,1933	2485 HAT
S	HATCH MEDIOS	FORD	FOCUS	0,0945	1073 HAT
S	HATCH MEDIOS	VW	GOLF	0,1279	1615 HAT
S	HATCH MEDIOS	FIAT	PUNTO	0,3427	3610 HA
S	SEDANS PEQUENOS	GM	ASTRA SEDAN	0,2304	687 SE
S	SEDANS PEQUENOS	GM	CLASSIC	0,2942	10204 VE
S	SEDANS PEQUENOS	FORD	FIESTA SEDAN	0,0988	3448 SE
S	SEDANS PEQUENOS	RENAULT	LOGAN	0,1035	3498 SE
S	SEDANS PEQUENOS	VW	POLO SEDAN	0,7684	2584 SI
S	SEDANS PEQUENOS	GM	PRISMA	0,1441	4573 SI
S	SEDANS PEQUENOS	FIA	SIENA	0,2709	8632 S
SUV MED'Ô	GM		BLAZER	0,0198	153 S
SUV MED'Ô	LAND ROVER		FREELANDER	0,0144	147
SUV MÉDIO	NISSAN	IBISHI	OUTLANDER	0,0108	
			SPORTAGE	0,0425	
		DAI	TUCSON	0,1468	
		EN	C3	0,1099	
		LT	CLIO	0,0135	
			CO	0,1461	
				0,1789	

Problemas! Modelagem

Aiai AWS....

Voltando ao Problema de Negócio....

- Temos uma quantidade limitada de produção, consequentemente uma limitação na quantidade de compras
- Um carro influencia no outro
 - Se o preço de um Hatch top de linha está próximo de um SUV básico, as pessoas preferem pegar o SUV.
- Mudança de gostos...
 - Crescimento dos SUVs 
 - Queda dos Hatchs e Sedans 
- Mudança de comportamento
 - Gerações atuais não querem mais comprar carros
 - Uber/99

Arquiteturas do modelo

- Abordagens de arquitetura do modelo
 1. Um único modelo para todos os carros
 2. Um modelo para cada carro
 3. Um modelo para cada sub-segmento
 4. Um modelo para carros similares
 - 1. Por quantidade de vendas por exemplo
 5. [....]

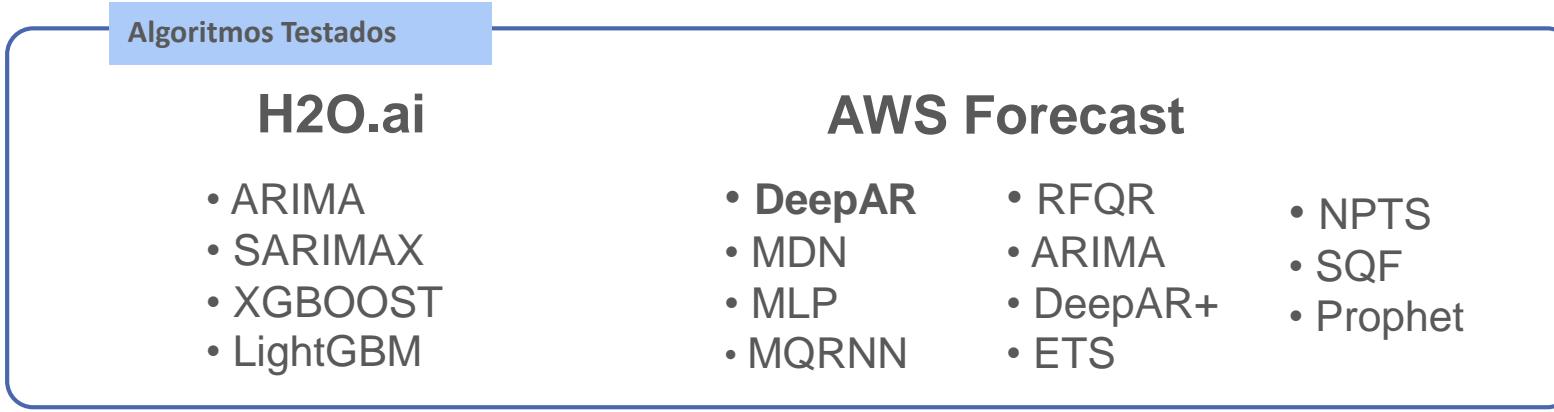
Qual melhor abordagem? Alguém já fez algo parecido? Qual melhor algoritmo para o nosso problema?

O nosso problema é similar ao o que? Vendas em Lojas.

Uma loja (nossa cliente) possui diversos produtos (veículos), quanto de cada produto ela ira vender nos próximos meses?

Amazon.

Algoritmos



Vencedor



Algoritmo que obteve o melhor
resultado

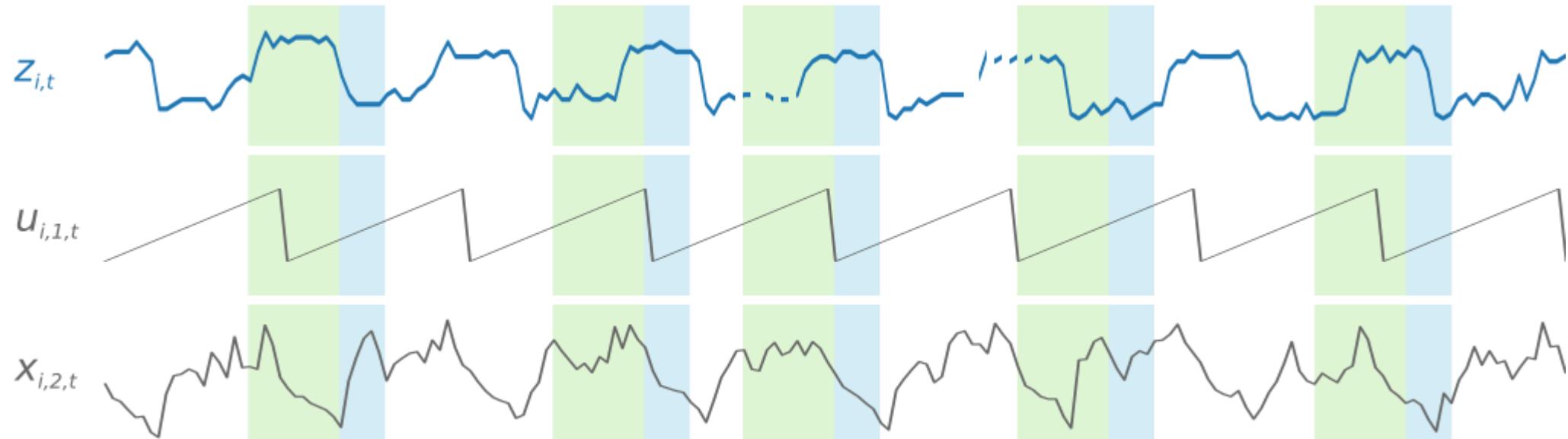
* Exclusivo AWS

DeepAR

Algoritmo de aprendizado supervisionado para previsão de séries temporais escalares (unidimensionais) usando redes neurais recorrentes (RNN).

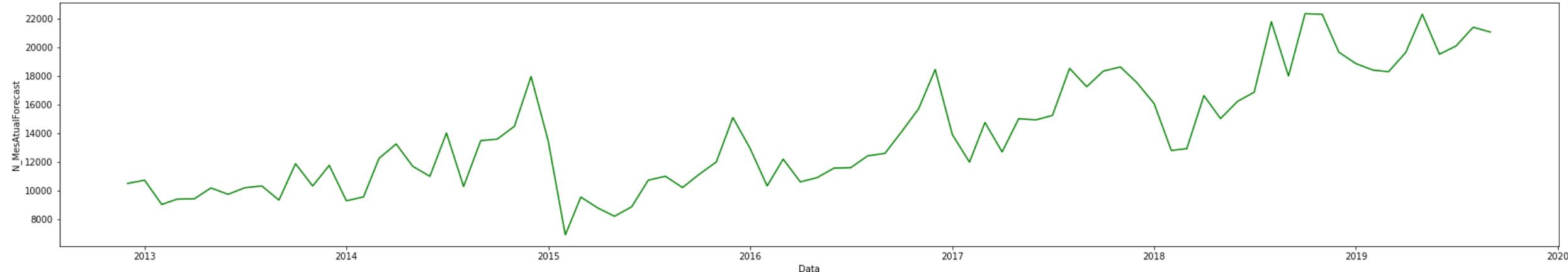
Métodos clássicos de previsão, como média móvel integrada autoregressiva (ARIMA) ou suavização exponencial (ETS), ajustam um modelo único a cada série temporal individual

Em diversos casos (como o nosso) nos temos diversas séries temporais semelhantes em um conjunto de dados. Nesse caso, a criação de um único modelo em conjunto em todas as séries temporais relacionadas.

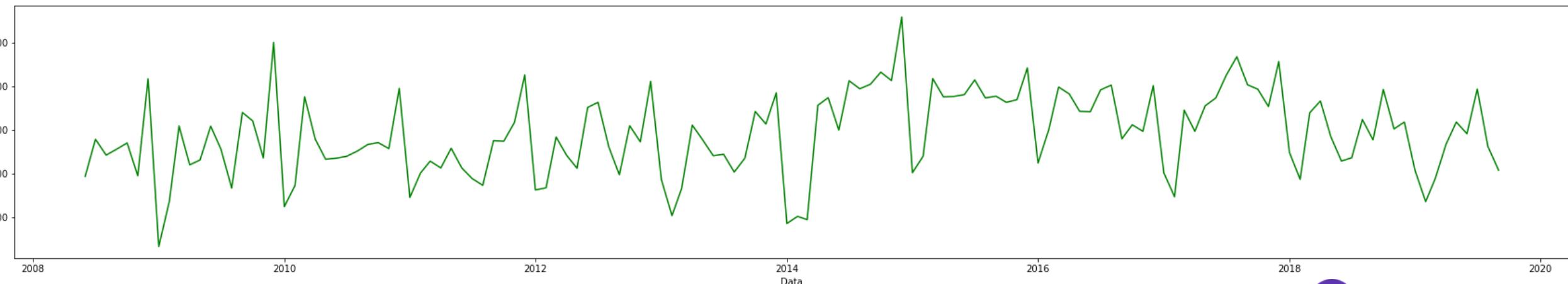


Cada carro uma serie temporal diferente....

Serie Temporal - Onix

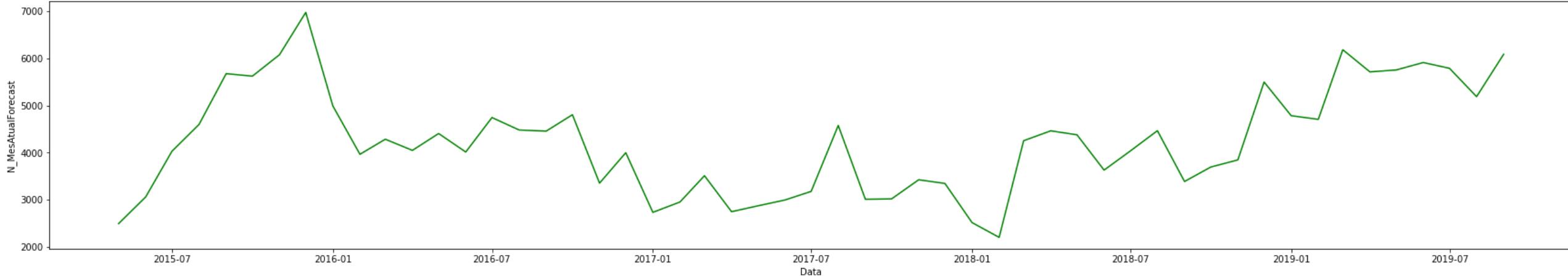


Serie Temporal - COROLLA

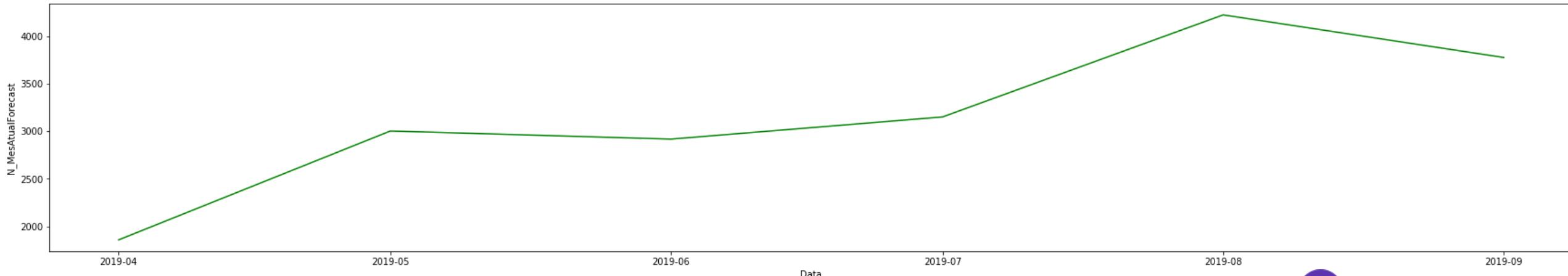


Cada carro uma serie temporal diferente....

Serie Temporal - RENEGADE



Serie Temporal - T-CROSS



Mas nem tudo parece flores....

- Parecia que tínhamos encontrado o Algoritmo e serviço perfeito para resolver nosso problema de negócio...até que...

Amazon Forecast Service Announcement ➔ Caixa de entrada x Serviços/AWS/AWS Forecast x

 Amazon Web Services <aws-marketing-email-replies@amazon.com>
para eu ▾

⇄ inglês ▾ > português ▾ Traduzir mensagem



Dear Amazon Forecast preview customer,

As of 6/6, the Amazon Forecast service will no longer support the following recipes for new predictors: DeepAR, MDN, MQRNN, MLP, RFQR and SQF. Predictors previously created using these recipes can still be used to generate forecasts, but they cannot be updated with new data. These changes have been made to ensure Amazon Forecast offers the highest level of service accuracy and robustness.

Thank you for using the Amazon Forecast preview, and please let us know if you have any questions.

Sincerely,
The Amazon Forecast team

Muita atenção em usar serviço beta!!!

Amazon Sagemaker

- Felizmente, o Amazon Sagemaker possui de forma nativa a implementação do DeepAR

DeepAR Forecasting Algorithm

```
{"start": "2009-11-01 00:00:00", "target": [4.3, "NaN", 5.1, ...], "cat": [0, 1], "dynamic_feat": [[1.1, 1.2, 0.5, ...]]}  
{"start": "2012-01-30 00:00:00", "target": [1.0, -5.0, ...], "cat": [2, 3], "dynamic_feat": [[1.1, 2.05, ...]]}  
{"start": "1999-01-30 00:00:00", "target": [2.0, 1.0], "cat": [1, 4], "dynamic_feat": [[1.3, 0.4]]}
```

- Tivemos 4 dias (quinta, sexta, sábado e domingo) para:
 1. Aprender a usar o Sagemaker;
 2. Aprender como utilizar o DeepAR do Sagemaker
 3. Entender como ele espera receber os dados (JSON Lines – Imagem acima)
 4. Montar todo o script para leitura e preparação dos dados, montagem dos Jsons de treino e teste e gerar os resultados
 5. Gerar as versões do modelo e fazer Tuning dos Hyperparametros
 6. Gerar o modelo final
 7. Gerar os gráficos
 8. Os gráficos fazem sentido? (conhecimento de negócio...)



Felizmente conseguimos!

```
In [1]: %matplotlib inline
import sys
from urllib.request import urlretrieve
import zipfile
from dateutil.parser import parse
import json
from random import shuffle
import random
import datetime
from datetime import datetime
import os
from datetime import timedelta, date
from dateutil.relativedelta import relativedelta

import uuid
import boto3
import s3fs
import sagemaker
import numpy as np
from numpy import exp, log
import pandas as pd
import matplotlib.pyplot as plt

from sklearn import preprocessing
from numpy import log
from sklearn.preprocessing import MinMaxScaler

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', -1)

In [2]: # set random seeds for reproducibility
np.random.seed(42)
```

Macro Economical Scenario

Automotive Industry Performance

Confidence Indexes

Last updated on 2019-10-14



Filter

Undo Redo

GDP - Source: Itaú e Bradesco

4%

2%

1%

1.1%

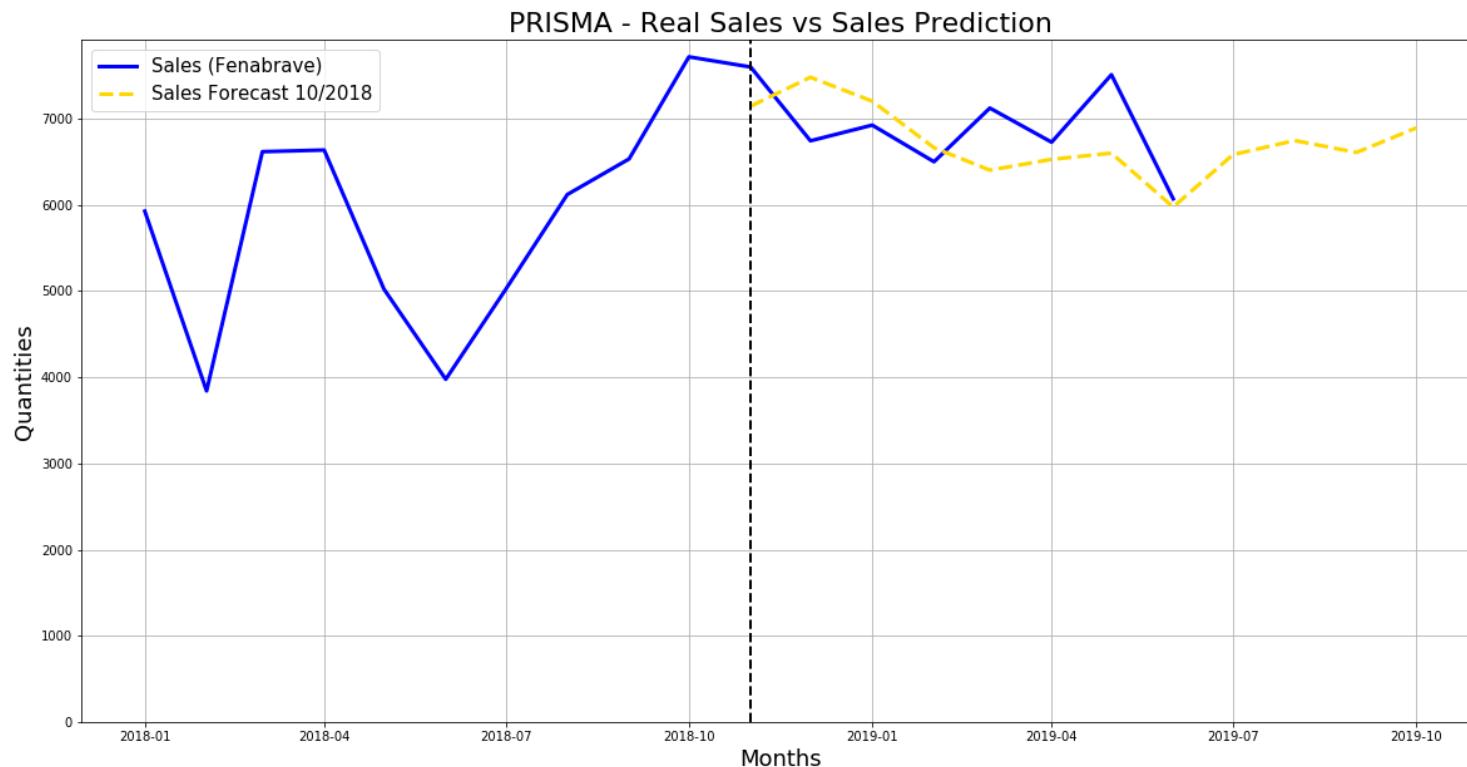
0.8%

Resultados

Radar

Resultados do Modelo

Modelo de validação (real x previsto)



Resultados do Modelo - Produção

Project	Real Sales	Month Forecast	Average Forecast	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
PRISMA *	7251	9177	10175	1926	73,44	59,67	86,53	* Pre facelift (10/2019)
Radar - Performance Analysis, August 2019								
					Forecast generated at 08/27/2019	Results analysis at 09/05/2019		
Radar - Performance Analysis, September 2019								
					Forecast generated at 09/24/2019	Results analysis at 10/08/2019		
Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
PRISMA	8946	8927	9831	19	99,79	90,11	99,89	
Radar - Performance Analysis, October 2019								
					Forecast generated at 10/16/2019	Results analysis at 11/07/2019		
Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
PRISMA	11399	9493	8384	1906	83,28	73,55	93,14	

Onix Plus
Chevrolet Onix Plus tem entregas suspensas por risco de incêndio

Modelo se tornou o sedã mais vendido do país em outubro, logo após o lançamento. Suspensão consta de comunicado interno da GM obtido pelo G1. Fabricante diz que fará recall e oferecerá veículos alugados aos clientes.

Por Guilherme Fontana, G1
06/11/2019 10h46 · Atualizado há 5 dias



Vai afetar previsões dos próximos meses

Onix Plus

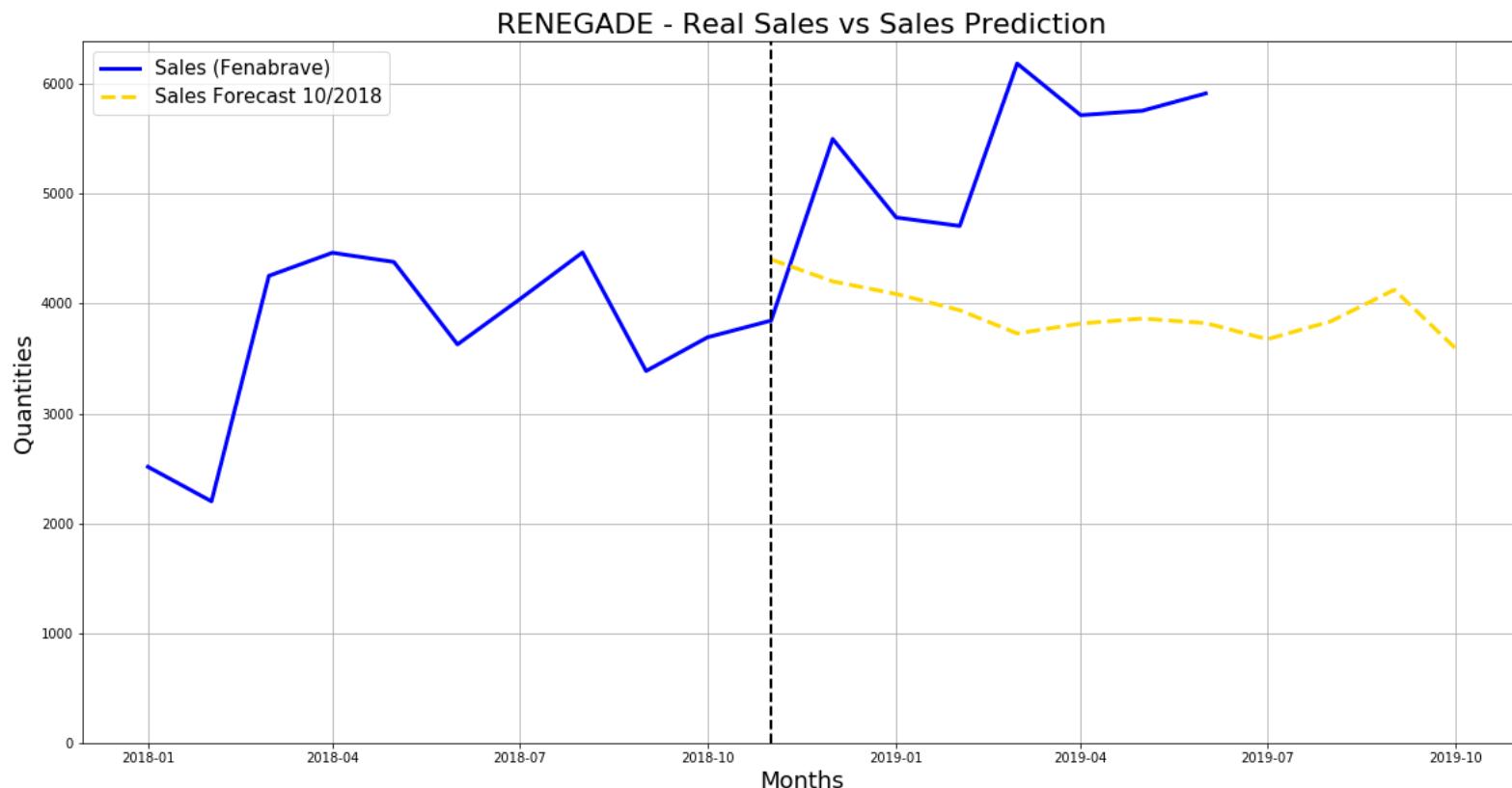
De todos os carros, o quanto esse representa em relação a faturamento?

Nível de importância



Resultados do Modelo

Modelo de validação (real x previsto)



Resultados do Modelo - Produção

Radar - Performance Analysis, August 2019

Forecast generated at 08/27/2019
Results analysis at 09/05/2019

Project	Real Sales	Month Forecast	Average Forecast	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
RENEGADE	5188	6472	5579	1284	75,25	92,45	91,15	

Radar - Performance Analysis, September 2019

Forecast generated at 09/24/2019
Results analysis at 10/08/2019

Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
RENEGADE	6089	5183	5399	906	85,12	88,66	95,41	

Radar - Performance Analysis, October 2019

Forecast generated at 10/16/2019
Results analysis at 11/07/2019

Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
RENEGADE	6680	6391	6068	289	95,67	90,84	98,80	

Resultados do Modelo – Produção – T-CROSS (Carro Novo)

Radar - Performance Analysis, August 2019

Forecast generated at 08/27/2019
Results analysis at 09/05/2019

Project	Real Sales	Month Forecast	Average Forecast	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
T-CROSS	4224	3253	3497	971	77,01	82,78	95,87	

Radar - Performance Analysis, September 2019

Forecast generated at 09/24/2019
Results analysis at 10/08/2019

Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
T-CROSS	3776	4056	4549	280	92,58	79,54	98,21	

Radar - Performance Analysis, October 2019

Forecast generated at 10/16/2019
Results analysis at 11/07/2019

Project	Real Sales	Month Forecast	Forecast Average	Difference Real vs Month Forecast	Month Forecast Performance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)	Comments
T-CROSS	5084	3999	4155	1085	78,66	81,74	96,31	

Resumão dos resultados até agora....

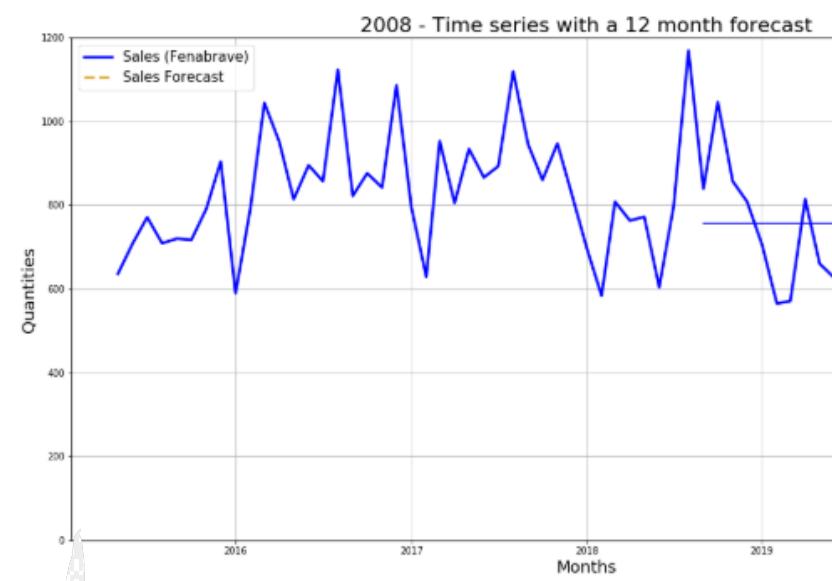
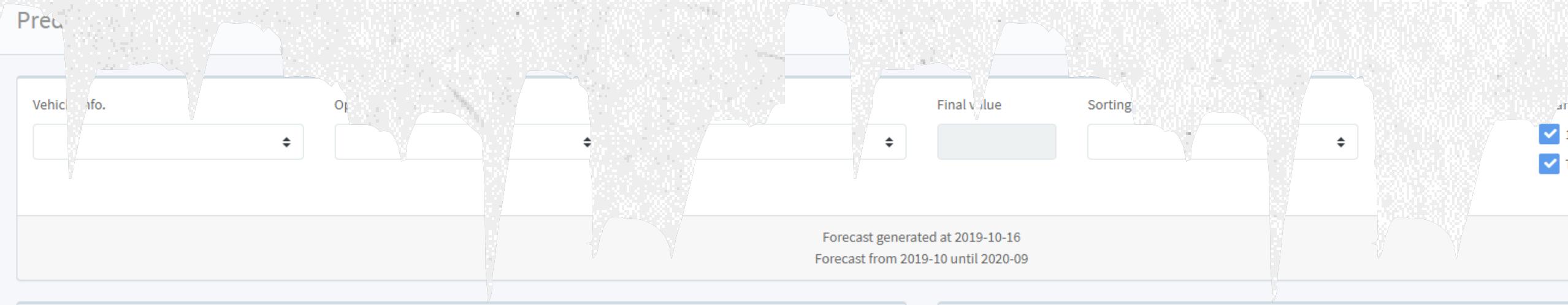
TOTAL AVERAGE	Month Forecast Perfomance (%)	Average Forecast Performance (%)	Weight Forecast Performance (%)
August 2019	78,64	81,74	95,84
September 2019	81,82 	73,02 	97,01 
October 2019	84,18 	80,12 	97,27 

O projeto em números

Dado coletados	Dados em Meses	Veículos Treinados	Veículos Previstos	Features coletadas/criadas
13.843	131 months	185	131	161
Quantidade de modelos treinados	Organização, Analise e limpeza	Horas de Treinamento	Time	Serviços Usados/ Testados
146+	790 hours+	240 hours+	<ul style="list-style-type: none">• 01 PM• 03 Data Scientists• 02 Software Analyst	03

Tudo isso será feito de forma mensal...infinitamente....até que o cliente pare de pagar







MUITO OBRIGADO

WWW.OMOTOR.COM.BR

-  Jessica Cabral
-  /jessica-cabral-carvalho/
-  @jcabralc
-  jessica@omotor.com.br