30/09/2022

SPRINT 2

Bootcamp - Precificação dinâmica

Ana Chaves / Paulo Angellotti/ Victor Mitsuo

SPRINT 2

BOOTCAMP - PRECIFICAÇÃO DINÂMICA

MERCARI PRICE SUGGESTION CHALLENGE - KAGGLE

Com base no dataset da Mercari, será criado um produto de sugestão de preços aos vendedores, que será oferecido a sites de e-commerce.

“O preço do produto fica ainda mais difícil em escala, considerando quantos produtos são vendidos online. As roupas têm fortes tendências de preços sazonais e são fortemente influenciadas por marcas, enquanto os eletrônicos têm preços flutuantes com base nas especificações do produto.”

ENTENDENDO O NEGÓCIO: MERCARI

Entre as muitas plataformas de comércio eletrônico online, Mercari é um famoso site de compras e venda online, uma escolha popular para encontrar itens a preços mais baratos do que em outras plataformas. Tudo começou em 2013 e, devido aos seus métodos de compra e venda muito fáceis, rapidamente ganhou fama e agora tem cerca de 16 milhões de usuários ativos por mês. Mercari expandiu seus serviços e agora possui Mercari USA e Mercari UK.

A Mercari tem várias lojas de conveniência parceiras onde o produto pode ser enviado. Neste caso, a taxa de envio está incluída na venda do item e o vendedor não paga nenhuma taxa para mandar o mesmo.

Os vendedores da Mercari Japão estão todos localizados no Japão. Uma pequena parte deles já enviou encomendas para o estrangeiro. De fato, a maioria dos vendedores na Mercari são indivíduos. Além disso, uma envio internacional não tem os mesmos custos que um envio doméstico. Portanto, mais difícil para o vendedor calcular o seu custos. É por isso que a maioria deles não se dá ao trabalho de negociar com compradores estrangeiros. Podemos observar que é um aplicativo mais restrito a venda local.

Os produtos ofertados podem ser novos ou usando, respeitando um sistema de classificação, no dataset chamado de , cujo vai de 1 a 5, sendo 1 ruim e 5 ótimo.

BRAINSTORM

Durante a análise exploratória, foram observados alguns pontos e levantado algumas dúvidas de como funciona o negócio e como pode ser feito um modelo para identificar o preço de uma maneira mais precisa e rápida.

- Onde estão o grande volume de vendas?
- Como o mercari ganha dinheiro?
    - *O valor pelo qual você vendeu o item será creditado na sua conta Mercari após dedução de 10% da taxa Mercari.*

- Comissão? Porcentagem? Anúncio?
    - *10% da taxa*

- Mercado restrito? Ou generalista?
- Quão importante é a métrica?
- Quão importante a margem de erro?
- O frete interfere no preço?
    - A maioria dos itens do Mercari é vendida com custos de envio incluídos no preço do item. Você pode optar por cobrar o frete separadamente, mas suas chances de venda são menores. O preço mais baixo que você pode cobrar por um item é 300 Yen

- As marcas influenciam no preço?
- Rede neural poderia ser utilizada, entretanto um cliente não iria esperar 90 minutos pelo resultado.
- Verificar para o vendedor não perder dinheiro e deixar de utilizar o aplicativo.
- Como será normalizado os dados?
- Dataset pesado, qual melhor método para limpar e processar?

DICIONÁRIO DE DADOS

Name	Texto	O título da listagem. Obs: Observe que limpamos os dados para remover textos que parecem preços (por exemplo, US$ 20) para evitar vazamentos. Esses preços removidos são representados como [rm]
Item_condition_id	1 - 2 – 3 – 4 – 5	A condição dos itens fornecidos pelo vendedor. Categoria da condição do item, entre 1 que significa ruim até 5 ótimo.
Category_name	Texto	Categoria da listagem
Brand_name	Texto	Marca
Price	Números	O preço pelo qual o item foi vendido. Esta é a variável de destino que você irá prever.
Shipping	0 ou 1	Frete: 1 se a taxa de envio for paga pelo vendedor e 0 pelo comprador.
item_description	Texto	A descrição completa do item.
Date	Date	Criado para este desafio.
Stock	Números	Criado para este desafio, é o que tem em estoque disponível.
gen_cat		Categoria principal do item.
sub1_cat		Subcategoria nível 1
sub2_cat		Subcategoria nível 2

ANÁLISE EXPLORATÓRIA

Ao abrir o dataset, podemos observar que os produtos contém nome, condição do item, a qual categoria pertence, sendo a primeira categoria principal e as demais subcategorias. Alguns produtos possuem marca, mas em sua grande maioria não possui. O atributo de envio, possui duas categorias 1 e 0, cujo 0 é o envio por conta do comprador, e 1 envio pago pelo vendedor, e a descrição do item. Foi solicitado para colocarmos data e número de estoque aleatório.

É um dataset com mais de um milhão e meio de linhas e sete atributos, sendo solicitados a inclusão de datas aleatórias e estoque.

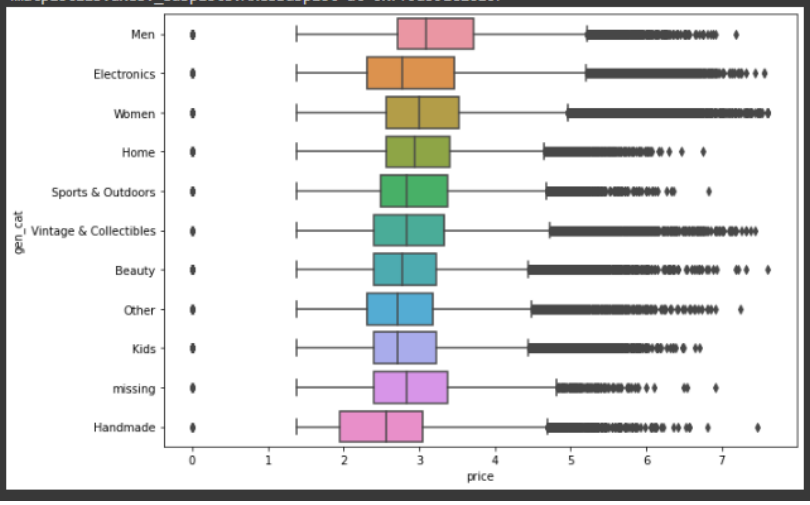
	name	item_condition_id	category_name	brand_name	price	shipping	item_description	date	stock	gen_cat	sub1_cat	sub2_cat
0	MLB Cincinnati Reds T Shirt Size XL	3	Men/Tops/T-shirts	NaN	10.0	1	No description yet	18-6-2018	27	Men	Tops	T-shirts
1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.0	0	This keyboard is in great condition and works ...	18-3-2018	15	Electronics	Computers & Tablets	Components & Parts
2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.0	1	Adorable top with a hint of lace and a key hol...	25-10-2018	14	Women	Tops & Blouses	Blouse
3	Leather Horse Statues	1	Home/Home Décor/Home Décor Accents	NaN	35.0	1	New with tags. Leather horses. Retail for [rm]...	20-3-2018	1	Home	Home Décor	Home Décor Accents
4	24K GOLD plated rose	1	Women/Jewelry/Necklaces	NaN	44.0	0	Complete with certificate of authenticity	16-5-2018	13	Women	Jewelry	Necklaces
...	...	...	...	...	...	...	...	...	...	...	...	...
1482530	Free People Inspired Dress	2	Women/Dresses/Mid-Calf	Free People	20.0	1	Lace, says size small but fits medium perfectl...	13-10-2018	2	Women	Dresses	Mid-Calf
1482531	Little mermaid handmade dress	2	Kids/Girls 2T-5T/Dresses	Disney	14.0	0	Little mermaid handmade dress never worn size 2t	6-10-2018	10	Kids	Girls 2T-5T	Dresses
1482532	21 day fix containers and eating plan	2	Sports & Outdoors/Exercise/Fitness accessories	NaN	12.0	0	Used once or twice, still in great shape.	6-8-2018	15	Sports & Outdoors	Exercise	Fitness accessories
1482533	World markets lanterns	3	Home/Home Décor/Home Décor Accents	NaN	45.0	1	There is 2 of each one that you see! So 2 red ...	13-2-2018	20	Home	Home Décor	Home Décor Accents
1482534	Brand new lux de ville wallet	1	Women/Women's Accessories/Wallets	NaN	22.0	0	New with tag, red with sparkle. Firm price, no...	28-10-2018	9	Women	Women's Accessories	Wallets

1482535 rows × 12 columns

Aqui verificamos os preços em relação com a categoria principal. Preço normalizado com log para melhorar a distribuição.

```
plt.figure(figsize = (10,7))
sns.boxplot(data = pd.concat([train['price'].apply(lambda x: math.log(x+1)), train['gen_cat']], axis = 1), x = 'price', y = 'gen_cat')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9a391c2610>
```

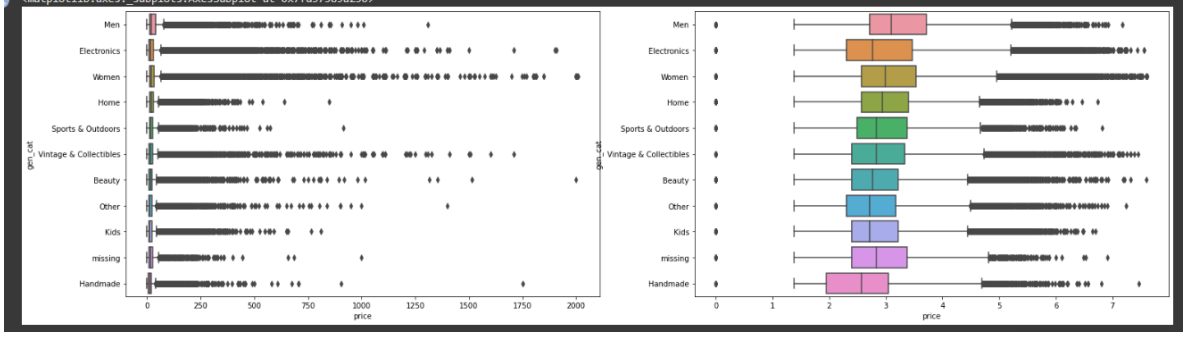


Verificação de outliers nos preços.

Os preços possuem uma grande amplitude e dificultam a leitura em uma escala simples, a aplicação do log permite a leitura e interpretação dos valores em uma escala logarítmica.

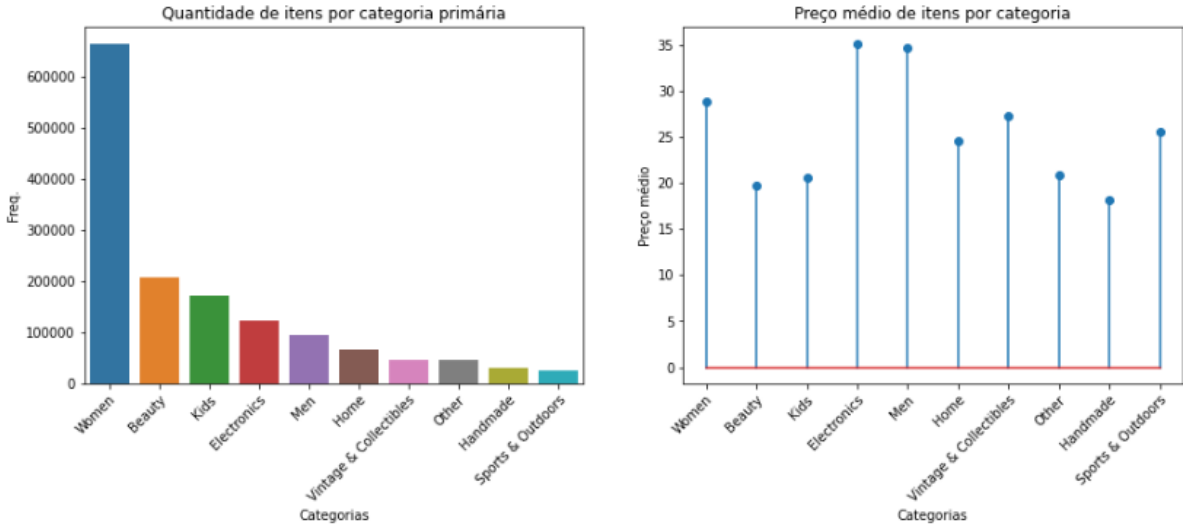
```
# Verificação de Outliers nos preços

plt.subplots(1,2, figsize = (25,7))

plt.subplot(1,2,1)
sns.boxplot(data = train[['price', 'gen_cat']], x = 'price', y = 'gen_cat')

# Os preços possuem uma grande amplitude e dificultam a leitura em uma escala simples, a aplicação do log permite a leitura e interpretação dos valores em uma escala logarítmica.

plt.subplot(1,2,2)
sns.boxplot(data = pd.concat([train['price'].apply(lambda x: math.log(x+1)), train['gen_cat']], axis = 1), x = 'price', y = 'gen_cat')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd37589a250>
```



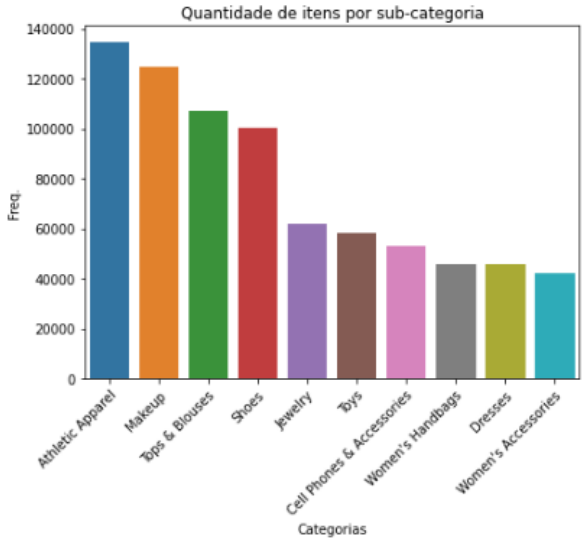
Em ambas as visualizações, nota-se uma grande presença de outliers. Porém, por tratar-se de um serviço de vendas de diversos tipos de produto, espera-se que tais discrepâncias sejam naturais e por hora serão mantidas na baseline inicial.

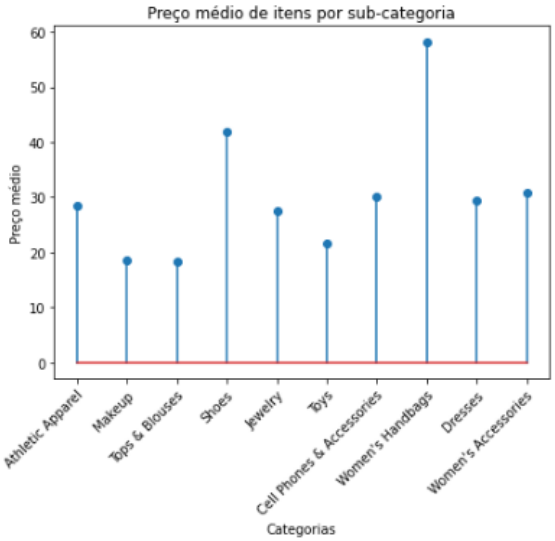
Aqui apresentou a categoria que mais tem volume no dataset, que são os produtos femininos, e o outro o preço médio dos produtos dessas mesmas categorias.



A frequência permite a retirada de alguns insights sobre o comportamento dos dados. Podemos ver que a variável ‘Women’ aparece numa frequência muito acima das demais e pode vir a gerar ruídos nos resultados do modelo de predição a ser desenvolvido.

Distribuição por subcategorias.



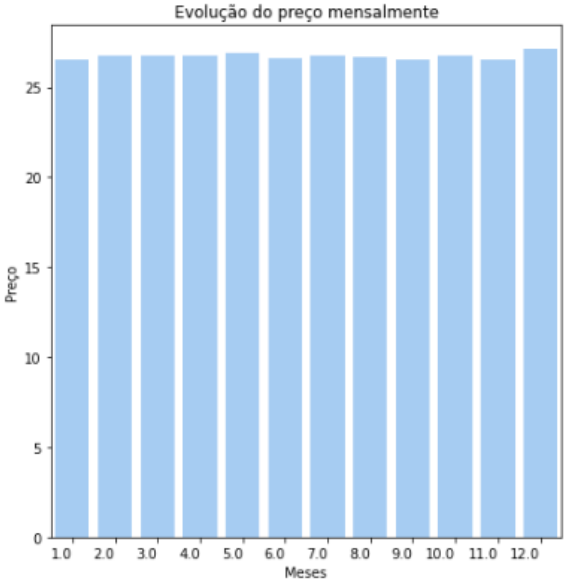


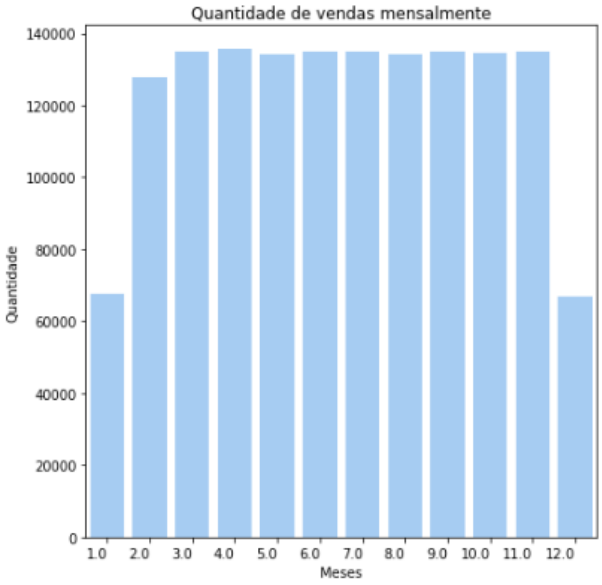
A análise geral das sub-categorias mostra um equilíbrio muito mais saudável em relação ao equilíbrio das categorias principais, porém, seus preços mostram variações mais extremas.

Verificando o volume de vendas por mês.

	datetime_month	price	
		count	mean
11	12.0	67055	27.106420
10	11.0	134979	26.569626
9	10.0	134584	26.795039
8	9.0	134935	26.531641
7	8.0	134265	26.675165
6	7.0	134846	26.787973
5	6.0	134857	26.648776
4	5.0	134173	26.907828
3	4.0	135564	26.784390
2	3.0	134889	26.765437
1	2.0	127916	26.794291
0	1.0	67575	26.557558

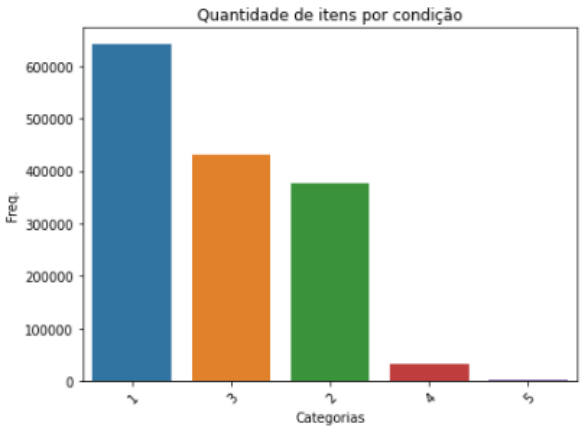
E a evolução dos preços por mês, o que não houve diferença, ou algo que se pode apurar.

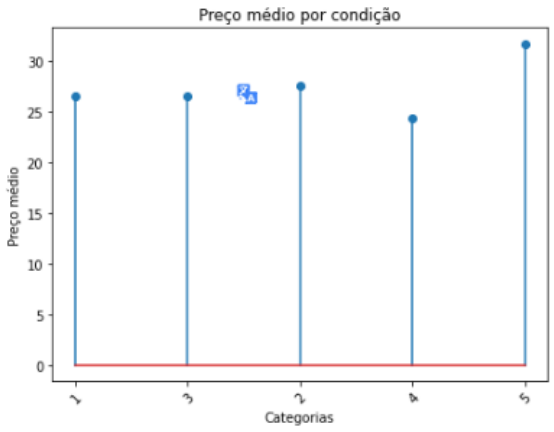




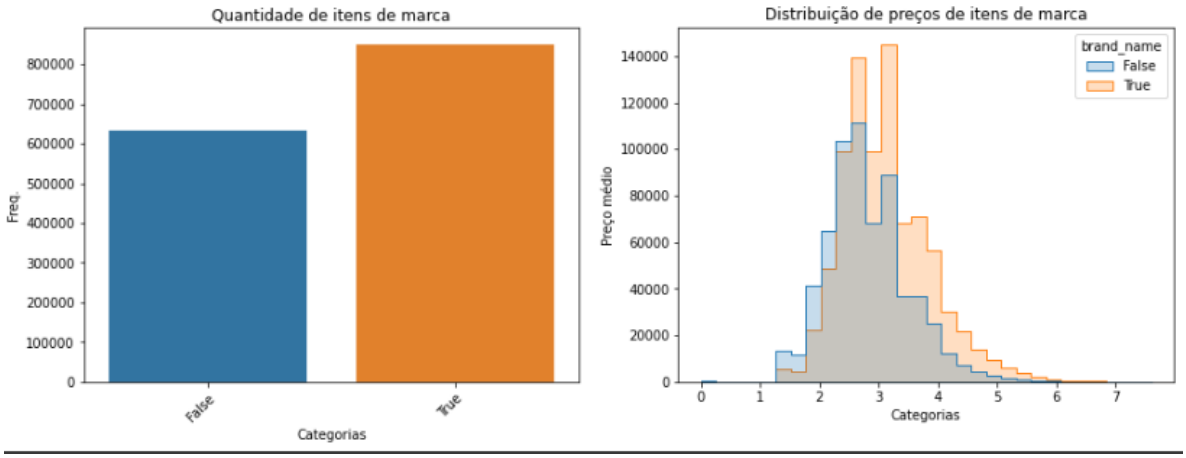
A análise da variação do preço no tempo nos mostra um comportamento atípico dos dados. Não há grande variação na média geral dos preços no decorrer do ano e além disso há grandes quedas no primeiro mês e no último mês.

A distribuição de itens por condição nos mostra uma grande valorização de itens com qualidade 5 apesar da grande maioria pertencer a qualidade 1.





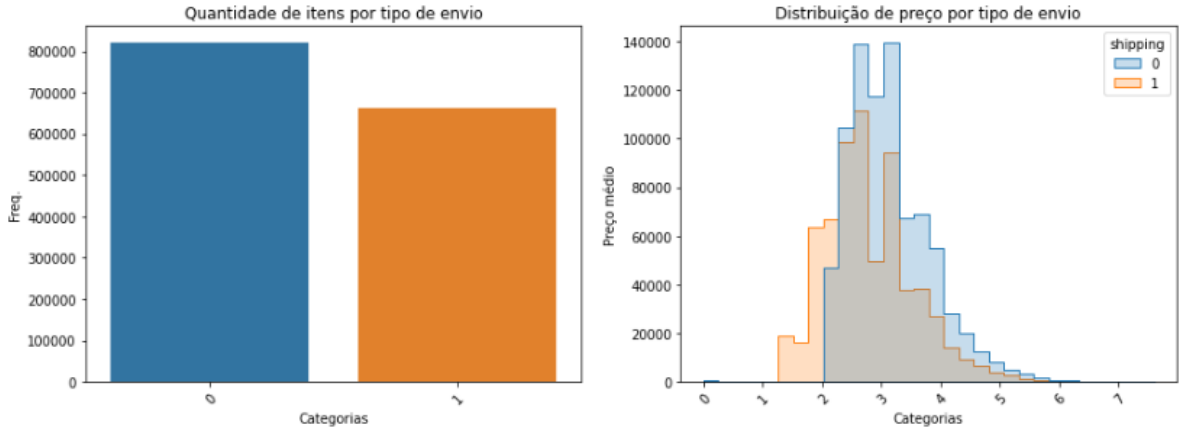
Aqui foram avaliadas as marcas, se itens que possuem marca tem valores maiores.



Apesar de haver marcas diferentes, a análise acima mostra que há uma variação considerável entre itens que possuem marcas. Isso nos abre uma possibilidade de no pré-processamento substituir as marcas por valores que indiquem ausência ou presença.

HANDMADE

Quantidade por tipo de envio:



Além disso, o modo de cobrança do frete também influencia no preço final do produto. E, ao contrário do que se espera, o preço tende a subir quando o frete é pago pelo vendedor. Isso provavelmente acontece porque geralmente o vendedor paga o frete para vendas mais caras como forma de impulsionamento.

Avaliando o impacto da hierarquia na disposição das categorias e como isso pode influenciar o modelo.

```
[ ] train[train['gen_cat'] == 'Women']['sub1_cat'].unique()

    array(['Tops & Blouses', 'Jewelry', 'Other', 'Swimwear', 'Dresses',
           'Shoes', 'Athletic Apparel', 'Jeans', 'Underwear',
           "Women's Handbags", 'Coats & Jackets', 'Pants', 'Sweaters',
           'Maternity', "Women's Accessories", 'Skirts', 'Suits & Blazers'],
          dtype=object)

[ ] train[train['sub1_cat'] == 'Tops & Blouses']['sub2_cat'].unique()

    array(['Blouse', 'Tank, Cami', 'T-Shirts', 'Halter', 'Other', 'Tunic',
           'Button Down Shirt', 'Polo Shirt', 'Wrap', 'Knit Top',
           'Turtleneck'], dtype=object)
```

Quantas categorias e subcategorias diferentes no dataset. Isso faz questionar se usar dummies será uma boa ideia devido a hierarquia das subcategorias.

```
len(train['category_name'].unique())

1288
```

Estas estatísticas indicam a dimensão do problema que será enfrentado durante a modelagem.

A Correlação dos demais atributos com o preço.

A partir das correlações acima, observa-se que o frete é inversamente proporcionais ao preço.

Estoque:

O estoque no dataset apresenta somente os valores que o vendedor possui em inventário, não há informação sobre a quantidade de produto vendido, sendo difícil mensurar seu valor ou conseguir fazer uma gestão de estoque, sazonalidade.

Iremos averiguar após a limpeza do das categorias e nomes, se o estoque que o dataset possui, será capaz de ter interferência no preço, caso o nome ou categoria seja a mesma, como se ele fosse um fator de oferta e demanda.

Juntamente com as datas de venda, será possível verificar alguma sazonalidade nos produtos ofertados.

Filtrando uma marca ‘Apple’ e avaliando os produtos apresentados, entre aparelhos celulares, capinhas, tablets.

	name	item_condition_id	category_name	brand_name	price	shipping	item_description	stock	gen_cat	sub1_cat	sub2_cat	datetime_date	datetime_month
26	Otterbox Defender iPhone 6 Plus/6s Plus	1	Electronics/Cell Phones & Accessories/Cases, C...	Apple	13.0	1	Brand new Otterbox Defender iPhone 6 Plus/6s Plus	2	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-02-18	2.0
149	LIKE NEW IPHONE 5C	2	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	104.0	0	Just Upgraded So Now Finally Getting Rid Of My...	23	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	NaT	NaN
300	Jordan iPhone case only for plus	1	Electronics/Cell Phones & Accessories/Cases, C...	Apple	5.0	1	JORDAN IPHONE CASE ONLY FOR IPHONE 6 plus iPho...	16	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-12-15	12.0
582	iPhone 6/6s plus cases	3	Electronics/Cell Phones & Accessories/Cases, C...	Apple	20.0	1	4 iPhone 6/6s plus cases.	7	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-06-26	6.0
757	iPod nano 7th generation	3	Electronics/TV, Audio & Surveillance/Portable ...	Apple	62.0	1	In perfect working condition. One light scratc...	3	Electronics	TV, Audio & Surveillance	Portable Audio & Accessories	2018-04-15	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1482223	LuMee Duo iPhone 6 Plus 6s Plus 7 Plus	1	Electronics/Cell Phones & Accessories/Cases, C...	Apple	24.0	1	Black Marble Free Shipping	18	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-02-09	2.0
1482230	iPhone 7 case	1	Electronics/Cell Phones & Accessories/Cases, C...	Apple	10.0	1	Brand new iPhone 7 glitter case. Protects phon...	5	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-09-14	9.0
1482291	iPhone 7 Plus Case	1	Electronics/Cell Phones & Accessories/Cases, C...	Apple	7.0	1	Flexible slim glitter bling case new	25	Electronics	Cell Phones & Accessories	Cases, Covers & Skins	2018-07-13	7.0
1482387	IPAD PRO 8.7 INCH 32GB	1	Electronics/Computers & Tablets/iPad/Tablet/eB...	Apple	509.0	0	BRAND NEW IN BOX NEVER OPEN .IPAD PRO 32GB,8.9...	1	Electronics	Computers & Tablets	iPad	2018-11-15	11.0

Analisando sobre a sub-categoria2 ‘Cell Phones & Smartphones’, verificamos vários aparelhos, de modelos distintos, preços variados, pois alguns modelos são obsoletos.

	name	item_condition_id	category_name	brand_name	price	shipping	item_description	stock	gen_cat	sub1_cat	sub2_cat	datetime_date	datetime_month
149	LIKE NEW IPHONE 5C	2	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	104.0	0	Just Upgraded So Now Finally Getting Rid Of My...	23	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	NaT	NaN
796	iPhone 4	3	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	30.0	1	Black iPhone 4 for Verizon comes with charger	1	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-06-12	6.0
1330	iPhone 6 64gb Gold (Sprint)	3	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	305.0	1	Fully functional iPhone 6 64gb Rose Gold. Only...	20	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-07-02	7.0
1536	iPhone 6 Plus	2	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	310.0	1	iPhone 6 Plus US Cellular. Clean ESN. Excellen...	44	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-02-06	2.0
1593	iPhone 5c	3	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	34.0	0	Blue iPhone 5c Screen Was Recently Replaced An...	9	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-04-15	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1481376	iPod 5th generation	2	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	15.0	1	This iPod 5th generation is iCloud locked and ...	34	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-12-08	12.0
1481408	IPhone 5s 32gb unlocked	3	Electronics/Cell Phones & Accessories/Cell Pho...	Apple	167.0	1	For sale is an Apple iphone 5S 32GB unlocked. ...	11	Electronics	Cell Phones & Accessories	Cell Phones & Smartphones	2018-04-20	4.0

Valores nulos:

```
[21] # Faça uma cópia de trabalho dos dados
     df = train.copy()
     df.isna().sum().loc[df.isna().sum()>0].sort_values()
```

```
item_description         4
category_name         6327
datetime_date         6897
datetime_month        6897
brand_name          632682
dtype: int64
```

* 43% dos registros não tem nomes de marca

* 0,4% dos registros não tem categorias

* 0,0003% não possuí descrição do item

Iremos excluir os valores nulos, exceto a marca iremos avaliar o impacto dos valores nulo no modelo.

ANÁLISE ESTATÍSTICA

As informações padrões do dataset.

```
[ ] train.info()
```

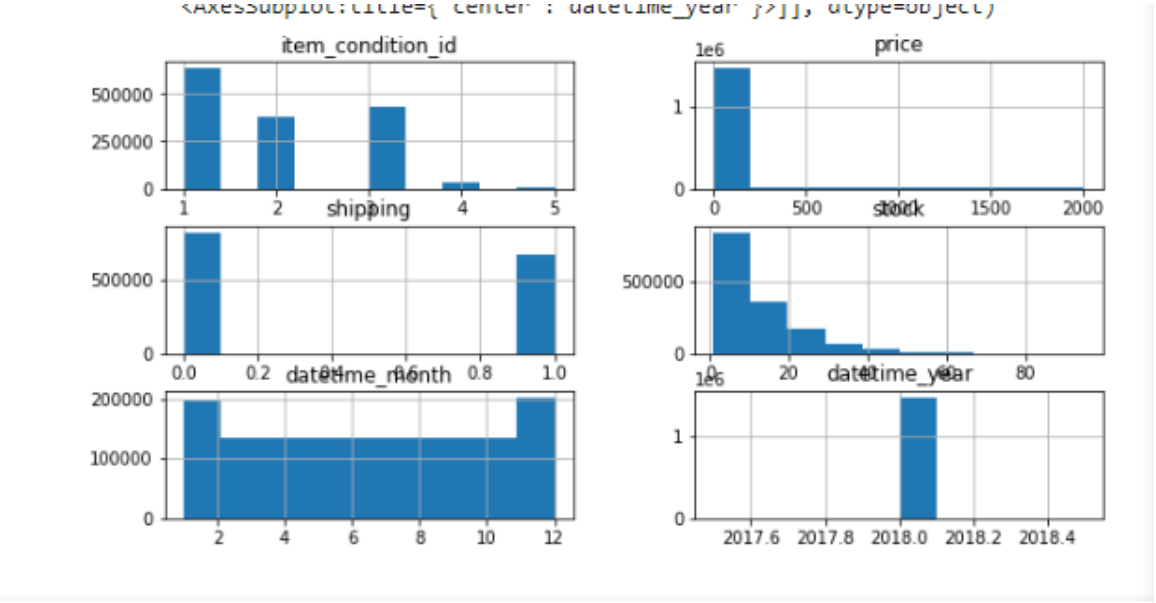
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1482535 entries, 0 to 1482534
Data columns (total 12 columns):
 #   Column             Non-Null Count    Dtype  
---  ------             --------------    -----  
 0   name               1482535 non-null  object 
 1   item_condition_id  1482535 non-null  int64  
 2   brand_name         849853 non-null   object 
 3   price              1482535 non-null  float64
 4   shipping           1482535 non-null  int64  
 5   item_description   1482531 non-null  object 
 6   stock              1482535 non-null  int32  
 7   gen_cat            1482535 non-null  object 
 8   sub1_cat           1482535 non-null  object 
 9   sub2_cat           1482535 non-null  object 
 10  datetime_month     1475638 non-null  float64
 11  datetime_year      1475638 non-null  float64
dtypes: float64(3), int32(1), int64(2), object(6)
memory usage: 130.1+ MB
```

Alguns atributos têm valores nulos.

```
[ ] train.isnull().sum()  #necessário valair forma de tratar os valores nulos.
```

```
name                     0
item_condition_id        0
brand_name           632682
price                    0
shipping                 0
item_description         4
stock                    0
gen_cat                  0
sub1_cat                 0
sub2_cat                 0
datetime_month        6897
datetime_year         6897
dtype: int64
```

Não há valores duplicados.



Utilizando histograma para entender as distribuições dos dados em cada feature.

```
train.describe()
```

	item_condition_id	price	shipping	stock	datetime_month	datetime_year
count	1.482535e+06	1.482535e+06	1.482535e+06	1.482535e+06	1.475638e+06	1475638.0
mean	1.907380e+00	2.673752e+01	4.472744e-01	1.178544e+01	6.517901e+00	2018.0
std	9.031586e-01	3.858607e+01	4.972124e-01	1.056072e+01	3.194517e+00	0.0
min	1.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	2018.0
25%	1.000000e+00	1.000000e+01	0.000000e+00	4.000000e+00	4.000000e+00	2018.0
50%	2.000000e+00	1.700000e+01	0.000000e+00	9.000000e+00	7.000000e+00	2018.0
75%	3.000000e+00	2.900000e+01	1.000000e+00	1.700000e+01	9.000000e+00	2018.0
max	5.000000e+00	2.009000e+03	1.000000e+00	9.500000e+01	1.200000e+01	2018.0

em desenvolvimento

HIPÓTESES

- Qual o melhor modelo para o negócio?
- O Estoque, juntamente com o nome do produto, pode ser um fator de alteração do preço de venda, devido a oferta e demanda?
- O modelo poderá identificar que o produto é obsoleto/encalhado e irá sugerir um valor mais baixo para ter maior atratividade? (Exemplo Iphone 5 para iphone 10)
- Etapas para comprovação da hipótese 1:
    - Qual é a relevância da marca para o modelo?

➢ Usar PCA, irá melhorar o processamento?
- da pra aplicar PCA em matriz esparsa

Em desenvolvimento.

HIPOTESE 1

Vamos avaliar o impacto da "brand" para o nosso modelo.

A razão de tal análise serve para entendermos melhor qual tratamento deveremos dar para nossos dados 632.682 nulos, um número significativo dentro de nosso dataset. Para isso vamos separa-los em dois datasets, um contendo apenas os valores nulos/missing e outro dataset contendo todas as colunas preenchidas. Após essa etapa vamos gerar modelos probatórios para um teste a/b, onde poderemos analisar o melhor caminho para o tratamento desses valores nulos.

Resultados

In [430…

```
print("---------------Modelo com brand----------------- ")


print('MAE: %2f' % mean_absolute_error(y1predict,y1val))
print('RMSE: %2f' % (mean_squared_error(y1predict,y1val)))
```

```
---------------Modelo com brand-----------------
MAE: 13.470141
RMSE: 1266.732524
```

In [431…

```
print("---------------Modelo sem brand----------------- ")


print('MAE: %2f' % mean_absolute_error(y2predict,y2val))
print('RMSE: %2f' % (mean_squared_error(y2predict,y2val)))
```

```
---------------Modelo sem brand-----------------
MAE: 10.244517
RMSE: 624.229604
```

Conforme levantado na hipótese, verificaríamos o quão explicativo a brand é no atual projeto, para saber como tratar tantos valores nulos, aproximadamente 42%. O objetivo não foi alcançar a melhor métrica, apenas comparar em condições similares.

Na preparação do treino/teste foi utilizado o mesmo sample de dados, pré-processamento, tratamento e parâmetros de modelo, no caso o lightgbm.

obtendo os resultados :

---------------Modelo com brand-----------------

MAE: 13.470141

RMSE: 1266.732524

----------Modelo sem brand----------

MAE: 10.244517

RMSE: 624.229604

Podemos observar que com as brands o modelo perde em ambas as métricas, isso pode ser devido ao grande desbalanceamento entre as marcas.

Contudo, essas informações de brand não serão descartadas.

HIPOTESE 2

- Usar PCA, irá melhorar o processamento?

  da pra aplicar PCA em matriz esparsa

Após os tratamentos iniciais no *dataset*, aplicamos um modelo simples de regressão para obter as primeiras predições e um esboço inicial de um modelo preditivo.

Escolhemos uma regressão linear por sua simplicidade e facilidade na aplicação. Também utilizamos uma amostra menor do *dataset* para verificar o funcionamento do modelo. As métricas deste modelo inicial são:

```
import math
from sklearn.metrics import mean_squared_error, mean_squared_log_error, mean_absolute_error

# x_val_pca = svd.transform(x_val)

y_true = [math.exp(i)-1 for i in model.predict(x_val)]
print(mean_absolute_error(yval,y_true))
print(mean_squared_error(yval,y_true))
```

```
5.8680294155017
70.62437242284109
```

Para esta primeira analise as colunas categóricas foram tratadas de duas formas. A ideia da extração de características em textos é a de estruturar dados não estruturados para que possam ser utilizados em algum modelo de *machine learning*. A técnica TF-IDF (*Term Frequency - Inverse Document Frequency*) mostra o quão importante é uma palavra é em um texto seguindo a seguinte fórmula:

$TFIDF = (a/b) * \log(\alpha/\beta)$

onde:
a = Nº de vezes que uma palavra aparece no texto
b = Nº de palavras no documento
α = Total de documentos
β = Nº de documentos com o respectivo termo

```
[ ]  vec_name = TfidfVectorizer(stop_words='english', ngram_range=(1,2)) #vetorização, com stop word.
     vec_desc = TfidfVectorizer(stop_words='english', ngram_range=(1,2)) #vetorização, com stop word.

     vce_xtrain_name = vec_name.fit_transform(Xtrain["name"])
     vce_xtrain_descrip = vec_desc.fit_transform(Xtrain["item_description"])

[ ]  vce_xtrain_descrip

     <30000x275746 sparse matrix of type '<class 'numpy.float64'>'
            with 869323 stored elements in Compressed Sparse Row format>
```

Uma das características deste tipo de tratamento é que cada termo se torna uma coluna e, de forma geral, sua saída é uma matriz esparsa. Uma matriz esparsa caracteriza-se pela presença de zeros na maioria dos elementos da matriz, sendo assim para a maioria das observações há muitas colunas com valores iguais a zero.

O tratamento das colunas categóricas foi feito utilizando o *One-Hot Encoding* também presente no *Scitkit-Learn*, neste tipo de tratamento tornamos dados categóricos em dados numéricos transformando cada uma das colunas são transformadas em diversas colunas binárias que representam a presença ou ausência das características presentes nesta coluna. Em colunas com diversas características possíveis, sua saída também pode ser uma matriz esparsa.

```
ohe = OneHotEncoder(handle_unknown="ignore")

#aplicação dummy nas colunas categóricas.

ohe_condition = ohe.fit_transform(Xtrain[["item_condition_id",
                                          "shipping",
                                          "brand_name",
                                          "gen_cat",
                                          "sub1_cat",
                                          "sub2_cat",
                                          "datetime_month"]])
```

```
ohe_condition

<30000x1967 sparse matrix of type '<class 'numpy.float64'>'
        with 210000 stored elements in Compressed Sparse Row format>
```

Após isso, ajustamos o modelo de regressão linear e então aplicamos a mesma transformação no conjunto de validação separado anteriormente.

```
[ ]  lr = LinearRegression()

     model = lr.fit(x_train, ytrain)
```

```
Xval["name"] = Xval["name"].apply(lambda x: text_preprocess(x))
Xval["item_description"] = Xval["item_description"].astype(str)
Xval["item_description"] = Xval["item_description"].apply(lambda x: text_preprocess(x))

vce_xval_name = vec_name.transform(Xval["name"])
vce_xval_descrip = vec_desc.transform(Xval["item_description"])

ohe_val_condition = ohe.transform(Xval[["item_condition_id",
                                        "shipping",
                                        "brand_name",
                                        "gen_cat",
                                        "sub1_cat",
                                        "sub2_cat",
                                        "datetime_month"]])

x_val = hstack([(vce_xval_name), (vce_xval_descrip), (ohe_val_condition)])
```

```
[ ]  import math
     from sklearn.metrics import mean_squared_error, mean_squared_log_error, mean_absolute_error

     # x_val_pca = svd.transform(x_val)

     y_true = [math.exp(i)-1 for i in model.predict(x_val)]
     print(mean_absolute_error(yval,y_true))
     print(mean_squared_error(yval,y_true))
```

```
5.8680294155017
70.62437242284109
```

As métricas mostram-se aceitáveis e provam que este esqueleto inicial do modelo é funcional, podendo ser escalado para as mais de 1 milhão de linhas presentes no *dataset* de treino completo.

Com este grande número de colunas, técnicas de redução de dimensionalidade começam a ser viáveis, como temos diversas matrizes esparsas, a única técnica de redução de dimensionalidade fornecida pelo *Scikit-Learn* é a *Truncated SVD.* Assim como em técnicas como **o PCA**, as novas componentes são escolhidas baseadas na variação que melhor explica os dados. Por recomendação da própria documentação do *Scikit-Learn,* será utilizado o valor de 100 componentes na redução de dimensionalidade.

```
#juntando cada matrix gerada por cada pré-processamento
from sklearn.decomposition import TruncatedSVD

x_train = hstack([(ohe_condition), (vce_xtrain_name), (vce_xtrain_descrip)])

svd = TruncatedSVD(n_components=100, algorithm = 'arpack', tol=0.1)

x_train_pca = svd.fit_transform(x_train)
x_train_pca
```

```
array([[ 1.1647684 , -1.07449353,  0.25624242, ..., -0.0116467 ,
         0.02083135,  0.03555446],
       [ 1.22429451, -0.57418048, -0.42709261, ...,  0.02759847,
        -0.03082447, -0.0522711 ],
       [ 1.00354222, -0.16984752, -0.94103193, ...,  0.00255599,
         0.02989365, -0.00409764],
       ...,
       [ 1.30721958,  0.07598229, -1.03066188, ...,  0.02154496,
        -0.05384449, -0.08933553],
       [ 0.88368252,  0.96874785,  0.04352169, ...,  0.00392473,
        -0.00629709, -0.01578896],
       [ 1.18568331, -1.07432416,  0.27923496, ..., -0.00852345,
         0.00948863,  0.05140185]])
```

Aplicando a redução para 100 colunas, temos os seguintes resultados:

```
import math
from sklearn.metrics import mean_squared_error, mean_squared_log_error, mean_absolute_error

x_val_pca = svd.transform(x_val)

y_true = [math.exp(i)-1 for i in model.predict(x_val_pca)]
print(mean_absolute_error(yval,y_true))
print(mean_squared_error(yval,y_true))
print(mean_squared_log_error(yval,y_true))
```

```
6.454353383112329
42.22088552310839
0.9952568253085067
```

Com a confirmação do funcionamento, tentamos utilizar todo o *dataset* para podermos comparar os resultados com modelos futuros, porém, devido ao grande volume de dados, a máquina virtual oferecida na versão gratuita do *google colab* não possui capacidade o suficiente de memória RAM para o processamento.

Ao analisar nosso problema, notamos que as funções de vetorização de texto são as que mais geram colunas na matriz esparsa. Assim, decidimos por adotar dois argumentos para limitar a entrada de um novo termo na matriz do *Tfidf*.

```
[ ] vec_name = TfidfVectorizer(stop_words='english', ngram_range=(1,2), min_df = 50, max_df = 0.5) #vetorização, com stop word.
    vec_desc = TfidfVectorizer(stop_words='english', ngram_range=(1,2), min_df = 50, max_df = 0.5) #vetorização, com stop word.

    vce_xtrain_name = vec_name.fit_transform(Xtrain["name"])
    vce_xtrain_descrip = vec_desc.fit_transform(Xtrain["item_description"])
```

```
[ ] vce_xtrain_descrip

    <888996x39930 sparse matrix of type '<class 'numpy.float64'>'
            with 17587568 stored elements in Compressed Sparse Row format>
```

O argumento *min_df* ignora os termos que possuem frequência menor que um certo limite, neste modelo inicial um termo deve aparecer pelo menos 50 vezes para aparecer na matriz esparsa de vetorização.

O argumento *max_df* ignora os termos que são muito frequentes e que não adicionariam nenhuma informação útil ao modelo. O valor 0.5 representa uma proporção entre documentos e o total da contagem de um termo. Com este tratamento, os vetores da coluna de descrição passam a ter 39930 linhas.

```
[ ] import math
    from sklearn.metrics import mean_squared_error, mean_squared_log_error, mean_absolute_error

    x_val_pca = svd.transform(x_val)

    y_true = [math.exp(i)-1 for i in yval.values]
    y_pred = [math.exp(i)-1 for i in model.predict(x_val_pca)]

    print(mean_absolute_error(y_true,y_pred))
    print(mean_squared_error(y_true,y_pred))
    print(mean_squared_log_error(y_true,y_pred))

    14.315523059997263
    1327.9410643163303
    0.4139522366756458
```

Nesta configuração o modelo é capaz de convergir, tendo métricas aceitáveis considerando a distribuição dos valores que estamos tentando prever. Isso significa que o PCA funciona e é escalável para todo o dataset, e pode trazer mais capacidade de generalização para os modelos a serem estudados.

RELATÓRIO MODELAGEM EM DESENVOLVIMENTO

Para baseline estamos aplicando uma random forest regression e o linear regression.

Primeira tentativa de baseline teve dificuldade de memória, e serão realizados pré-processamentos para tentar contornar esse problema.

Separamos o dataset em treino e teste para começarmos as transformações, protegendo nossos dados de validação de possíveis vazamentos.

Normalizado, com log, o target (price), foi realizado de forma separada para nao terem dados vazados.

Em desenvolvimento*

PRÉ PROCESSAMENTO

Criado um arquivo separado, para compor a inclusão das datas, e a divisão das categorias e subcategorias.

Será realizada a limpeza nas colunas de nome e descrição do item. Foi utilizado REGEX.

Foi realizada vetorização com stop words.

Aplicado dummie nas colunas categóricas, com OneHotEnconder.

Unimos todas as matrizes geradas em cada pré-processamento.

Em desenvolvimento*

LINKS

- Kaban: https://trello.com/b/iUi8g2cW/grupo-3-nan
- GitHub: https://github.com/anachavesv8/Bootcamp_Blue
-

REFERÊNCIAS

Como comprar e vender suas coisas usando o app Mercari Japão. <https://jn8.jp/pb/life_list/3073/>. Acessado 14/09/2022.

フリマアプリはメルカリ(メルペイ)-フリマアプリ&スマホ決済. IMAGEM<https://play.google.com/store/apps/details?id=com.kouzoh.mercari&hl=pt&gl=US>. Acessado 15/09/2022.

Como comprar da Mercari Japan. <https://www.whiterabbitexpress.com/pt/lojas/mercari-japan/>. Acessado 15/09/2022.