

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Deive de Freitas Flores

**MACHINE LEARNING NA EXPLORAÇÃO DE DADOS DE CONSUMO DE
COMPRAS**

Belo Horizonte
ano

Deive de Freitas Flores

**MACHINE LEARNING NA EXPLORAÇÃO DE DADOS DE CONSUMO DE
COMPRAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2023

SUMÁRIO

1. Introdução	4
1.1. Contextualização	4
1.2. O problema proposto	4
1.3. Objetivos	4
3. Processamento/Tratamento de Dados	6
4. Análise e Exploração dos Dados	7
5. Criação de Modelos de Machine Learning	8
6. Interpretação dos Resultados	9
7. Apresentação dos Resultados	10
8. Links	11
REFERÊNCIAS	12
APÊNDICE	29

1. Introdução

1.1. Contextualização

A tecnologia está presente em todos os nossos principais momentos, influenciando quase todos os aspectos da rotina diária, principalmente quando decidimos comprar. Sendo on-line ou off-line haverá uma escolha de produtos que farão parte de uma preferência de compra. A análise da cesta de compras, por exemplo, é uma técnica de mineração de dados que analisa padrões e determina o relacionamento entre os produtos adquiridos em conjunto. Também nos referimos a isso como mineração frequente de conjuntos de itens ou análise de associação. Ele aproveita esses padrões reconhecidos em qualquer ambiente de varejo para entender o comportamento do cliente, identificando as relações entre os itens comprados por eles. Simplificando, a análise da cesta de compras ajuda varejistas a conhecer os produtos que costumam ser comprados juntos, de modo a manter esses itens sempre disponíveis em seu estoque.

Alguns sites de comércio eletrônico são exemplos que aproveitam essa análise de dados para recomendar de forma inteligente outros produtos similares ou correlacionados à compra, com o título: “Quem comprou este produto, comprou também... Quem viu este produto, viu também...”. Por exemplo, ao adquirir um notebook, é comum que o cliente compre também mouse, impressora ou outros produtos relacionados.

Na compra em lojas físicas não é diferente, num arranjo de prateleira o uso alternativo na localização de produtos de uma loja e separar os itens que costumam ser comprados ao mesmo tempo. Isso serve para encorajar os clientes a passearem pela loja para encontrar o que estão procurando, aumentando potencialmente a probabilidade de compras adicionais por impulso.

1.2. O problema proposto

Este trabalho busca explorar os dados de consumo na expectativa de encontrarmos alguns padrões nos dados através do aprendizado de máquina e exploração de dados.

(Why?) Esse estudo analisa a experiência de compra do cliente e pode ajudar os gestores a melhorar o layout da loja facilitando a localização dos produtos e com isso aumento das vendas, além de identificar quais produtos possuem melhor venda.

(Who?) Os dados são de um atacado, mas foram extraídas quaisquer informações que pudessem infringir a LGPD ou identificar clientes.

(What?): Explorar os dados com técnicas de Machine Learning objetivando encontrar informações ocultas e que sejam de valia para gestão.

(Where?): Os dados foram extraídos de uma base de dados de um ERP onde contempla a movimentação de compra dos clientes, uma segunda fonte de dados foi utilizada para contemplar a tabela de NCM (Nomenclatura Comum do Mercosul).

(When?): Em função da quantidade de dados, será explorado um mês de movimentação, compreendendo de 1º a 31 de março de 2022, foi selecionado o março por ser um mês menos atípico, onde as pessoas já voltaram de suas férias de verão e praias.

Neste trabalho vamos fazer uso da linguagem de programação Python que é uma das mais utilizada para ciência de dados e machine learning(ML) e no ambiente Jupyter notebook que atende bem às nossas necessidades.

1.3. Objetivos

O objetivo deste trabalho é explorar os dados de consumo na expectativa de encontrarmos alguns padrões, através do aprendizado de máquina e exploração de dados. Além disso, entregar um conjunto de informações identificadas a respeito dos dados analisados, de forma que o gestor possa considerar algum desses dados na decisão. Essas informações, que às vezes estão implícitas, e quando melhores trabalhadas começam a ter mais valor seja para o marketing ou até mesmo para o cliente, que de alguma forma pode encontrar suas mercadorias de forma mais acessível.

2. Coleta de Dados

Recebi um arquivo csv com os dados de compra oriundos de ERP da organização onde poderemos trabalhar, são dados de movimentação de compra de um atacado, esses dados representam apenas os registros do mês de março de 2022 constituindo 575.219 registros de compra. O mês de março foi o escolhido porque a partir do início do ano é o mês menos atípico, em que a maioria das pessoas já voltaram de suas férias de verão.

Os registros são inseridos no banco de dados do sistema ERP considerando a data da compra, a unicidade da compra e itens pertencentes a essa compra, incluindo, é claro, vários outros campos que nesse contexto do trabalho não se fazem necessários.

Estrutura de registros de dados recebidos e que requer um refinamento quanto a dados e colunas:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 575219 entries, 0 to 575218
Data columns (total 46 columns):
#   Column                Non-Null Count  Dtype
---  -
0   chave_nfe_sai          575219 non-null object
1   codi_emp               575219 non-null int64
2   razao_emp              575219 non-null object
3   cgce_emp               575219 non-null int64
4   sigl_est               575219 non-null object
5   codi_pdi               575219 non-null object
6   desc_pdi               575219 non-null object
7   cncm_pdi               575219 non-null int64
8   nume_sai               575219 non-null int64
9   ddoc_sai               575219 non-null object
10  dsai_sai               575219 non-null object
11  codi_nat               575219 non-null int64
12  modelo_especie         575219 non-null object
13  vpro_msp               575219 non-null object
14  cgce_cli               575219 non-null int64
15  nome_cli               575219 non-null object
16  cst_msp                575219 non-null int64
17  bicms_msp              575219 non-null object
18  aliicms_msp            575219 non-null int64
19  valor_icms_msp         575219 non-null object
20  vseg_msp               575219 non-null int64
21  vsai_msp               0 non-null      float64
22  vipi_msp               575219 non-null int64
23  isentas                575219 non-null object
24  n_incid                575219 non-null object
25  cst_pis_msp            575219 non-null int64
26  bc_pis_msp             575219 non-null object
27  aliq_pis_msp           575219 non-null object
28  valor_pis_msp          575219 non-null object
29  cst_cofins_msp         575219 non-null int64
30  bc_cofins_msp          575219 non-null object
31  aliq_cofins_msp        575219 non-null object
32  valor_cofins_msp       575219 non-null object
33  UNIDADE                575219 non-null object
34  CODIGO_UNIDADE         575219 non-null int64
35  qtde_msp               575219 non-null object
36  valor_unit_msp         575219 non-null object
37  vdesace_msp            575219 non-null int64
38  vfre_msp               575219 non-null int64
39  bicmsst_msp            575219 non-null int64
40  aliq_st_msp            575219 non-null int64
41  valor_subtri_msp       575219 non-null int64
42  codi_acu               575219 non-null int64
43  nome_acumulador        575219 non-null object
44  vdes_msp               575219 non-null object
45  cat_dpi                192486 non-null object
dtypes: float64(1), int64(19), object(26)
memory usage: 201.9+ MB
```

A partir do arquivo recebido criou um novo dataframe de movimentação com a estrutura de tabela abaixo:

Tabela principal (movimento)

Nome da coluna	Descrição	Tipo
data_compra	Identifica em que data foi realizada a compra	data
Id_compra	Identifica uma compra realizada por um cliente	numérico
ncm_produto	Identifica em que classe esse produto se classifica	numérico
cod_produto	Código do produto	numérico
desc_produto	Descrição do produto	texto
categoria	Identifica o nome da categoria desse produto vinculada a sua NCM	texto
valor	Valor de cada item da compra	numérico

Foi necessário também buscar junto a receita federal a tabela de NCM para podermos categorizar melhor categorizar os dados de produtos, pois a tabela como pode ser vista possui uma descrição para formar categorias muito extensa. (<https://portalunico.siscomex.gov.br/classif/#/sumario?perfil=publico>). Na data de 02/04/2023.

O que é a NCM?

A Nomenclatura é um sistema ordenado que permite, pela aplicação de regras e procedimentos próprios, determinar um único código numérico para uma dada mercadoria. Esse código, uma vez conhecido, passa a representar a própria mercadoria. A Nomenclatura Comum do Mercosul (NCM) é uma Nomenclatura regional para categorização de mercadorias adotada pelo Brasil, Argentina, Paraguai e Uruguai desde 1995, sendo utilizada em todas as operações de comércio exterior dos países do Mercosul.

Qual é a utilidade da NCM?

A Nomenclatura Comum do Mercosul (NCM) é fundamental para determinar os tributos envolvidos nas operações de comércio exterior e de saída de produtos industrializados.

Sumário

* Selecione a Data:

06/04/2023



* Pesquisa:

Digite o código NCM ou descrição

☐ Expressão Exata



Pesquisar

Seção I - ANIMAIS VIVOS E PRODUTOS DO REINO ANIMAL



Capítulo 01 Animais vivos.

Capítulo 02 Carnes e miudezas, comestíveis.

Capítulo 03 Peixes e crustáceos, moluscos e outros invertebrados aquáticos.

Capítulo 04 Leite e laticínios; ovos de aves; mel natural; produtos comestíveis de origem animal, não especificados nem compreendidos noutros Capítulos.

Capítulo 05 Outros produtos de origem animal, não especificados nem compreendidos noutros Capítulos.

Seção II - PRODUTOS DO REINO VEGETAL



Capítulo 06 Plantas vivas e produtos de floricultura.

Capítulo 07 Produtos hortícolas, plantas, raízes e tubérculos, comestíveis.

Capítulo 08 Fruta; cascas de citros (cítricos) e de melões.

Capítulo 09 Café, chá, mate e especiarias.

Capítulo 10 Cereais.

Capítulo 11 Produtos da indústria de moagem; malte; amidos e féculas; inulina; glúten de trigo.

Capítulo 12 Sementes e frutos oleaginosos; grãos, sementes e frutos diversos; plantas industriais ou medicinais; palhas e forragens.

Capítulo 13 Gomas, resinas e outros sucos e extratos vegetais.

Capítulo 14 Matérias para entrançar e outros produtos de origem vegetal, não especificados nem compreendidos noutros Capítulos.

O código NCM é composto por 8 dígitos, os 6 primeiros representam a classificação SH da mercadoria e os 2 últimos dígitos representam classificações específicas do Mercosul, seguindo a estrutura apresentada a seguir:

- 2 primeiros dígitos do SH – Capítulo: características de cada produto.
- 4 primeiros dígitos do SH – Posição: desdobramento da característica de uma mercadoria identificada no Capítulo.
- 6 primeiros dígitos do SH – Subposição: desdobramento da característica de uma mercadoria identificada na Posição.
- 7º dígito da NCM – Item: classificação do produto.
- 8º dígito da NCM – Subitem: classificação e descrição mais completa de uma mercadoria.

Agora vamos entender a estrutura do código NCM analisando o caso real do NCM 3102.50.11.

- 31: Capítulo do SH – Adubos ou fertilizantes;
- 3102: Posição do SH – Adubos ou fertilizantes minerais ou químicos nitrogenados.
- 3102.50: Subposição do SH – Nitrato de sódio
- 3102.50.1: Item – Natural
- 3102.50.11: Subitem: Com teor de nitrogênio não superior a 16,3%, em peso

Então criada uma tabela reduzida de CNM para fins de esclarecimento de como os produtos foram categorizados.

Nome da coluna	Descrição	Tipo
Id	Id de chave primária	numérico
categoria	Nome da categoria	texto
lista_ncm	Lista reduzida da tabela de ncm	texto

```
1 SELECT * FROM tabela_ncm_reduzida
2
```

tabela_ncm_reduzida (7r x 3c)

id	categoria	lista_ncm
1	CARNES	20120,20130,20230,20312,20319,20329,20629,20712,20311,20322,20442,20610,20711,30319,30474,30617,50400
2	BEBIDAS	22011,22021,22030,22041,22042,22086,22060
3	LATICINIOS	40110,40120,40150,40610,19053
4	LIMPEZA	28289,68051,68053,39232,33029,96039,29039,38089,34039,34029
5	CEREAIS	10063,90300,71333,71340,11010,17011,11062,11041,21013,17001,19054,10059,19019
6	HIGIENE_PESSOAL	33043,96019,96032,96033,34011,33051,33061,33071
7	PET	23099,42010

3. Processamento/Tratamento de Dados

O arquivo de consumo de compras do atacado a ser trabalhado foi extraído de um segundo arquivo com dados em csv que foi extraído de uma base de dados do EPR organizacional. Foi necessário uma análise e limpeza, principalmente nas colunas que continham informações que poderiam infringir a LGPD. Como o movimento de compras realizado no atacado é de grande volume de dados nos foi liberado um mês do ano de 2022, optei por trabalhar com o mês de março.

```
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
```

```
df = pd.read_csv("d:/dados/movimento_de_compra.csv", sep=";", low_memory=False)
```

Descrevendo suas colunas originais e tamanho, como se observa a linha acima no processo de importação o parâmetro “low_memory=False” foi necessário em função do tamanho do arquivo.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 575219 entries, 0 to 575218
Data columns (total 46 columns):
#   Column              Non-Null Count  Dtype
---  -
0   chave_nfe_sai        575219 non-null object
1   codi_emp             575219 non-null int64
2   razao_emp            575219 non-null object
3   cgce_emp             575219 non-null int64
4   sigl_est             575219 non-null object
5   codi_pdi             575219 non-null object
6   desc_pdi            575219 non-null object
7   cncm_pdi            575219 non-null int64
8   nume_sai            575219 non-null int64
9   ddoc_sai            575219 non-null object
10  dsai_sai            575219 non-null object
11  codi_nat            575219 non-null int64
12  modelo_especie      575219 non-null object
13  vpro_msp            575219 non-null object
14  cgce_cli            575219 non-null int64
15  nome_cli            575219 non-null object
16  cst_msp             575219 non-null int64
17  bicms_msp           575219 non-null object
18  aliicms_msp         575219 non-null int64
19  valor_icms_msp      575219 non-null object
20  vseg_msp            575219 non-null int64
21  vsai_msp            0 non-null      float64
22  vipi_msp            575219 non-null int64
23  isentas             575219 non-null object
24  n_incid             575219 non-null object
25  cst_pis_msp         575219 non-null int64
26  bc_pis_msp          575219 non-null object
27  aliq_pis_msp        575219 non-null object
28  valor_pis_msp       575219 non-null object
29  cst_cofins_msp      575219 non-null int64
30  bc_cofins_msp       575219 non-null object
31  aliq_cofins_msp     575219 non-null object
32  valor_cofins_msp    575219 non-null object
33  UNIDADE             575219 non-null object
34  CODIGO_UNIDADE      575219 non-null int64
35  qtde_msp            575219 non-null object
36  valor_unit_msp      575219 non-null object
37  vdesace_msp         575219 non-null int64
38  vfre_msp            575219 non-null int64
39  bicmsst_msp         575219 non-null int64
40  aliq_st_msp         575219 non-null int64
41  valor_subtri_msp    575219 non-null int64
42  codi_acu            575219 non-null int64
43  nome_acumulador     575219 non-null object
44  vdes_msp            575219 non-null object
45  cat_dpi             192486 non-null object
dtypes: float64(1), int64(19), object(26)
memory usage: 201.9+ MB
```

Foi então reduzido o número de colunas que necessitamos para o nosso dataframe, como informado anteriormente esse arquivo continha dados de clientes que infringiam a LGPD, feito então uma redução de colunas.

```
selecao = df[{"codi_pdi", "desc_pdi", "cncm_pdi", "nume_sai", "ddoc_sai", "vpro_msp", "cat_dpi"}]
```

Foi então exportado para um outro arquivo chamado de movimento.csv

```
selecao.to_csv('movimento.csv', index=False)
```

```
df = pd.read_csv("movimento.csv", low_memory=False)
```

```
df.info()
```

```
df
```

	cod_produto	data_compra	categoria	desc_produto	id_compra	valor_produto	ncm_produto
0	78989901985472	2022-03-01	NaN	CLDO MAGGI 57g GALIN	19927	1,99	21041011
1	305	2022-03-01	NaN	BANANA CATURRA kg	19928	3,74	8039000
2	86151	2022-03-01	NaN	ALFACE LISA SENTIER	19928	2,99	7069000
3	78989901988855	2022-03-01	NaN	ACTIVIA AVEIA 850g	19928	10,99	4039000
4	78989901990257	2022-03-01	NaN	MORT. PERD. CT 400g	19928	3,49	16010000
...
575214	78989902002464	2022-03-31	NaN	BEB LAC BIO LAT 900g	436874	3,69	22029900
575215	78989902002464	2022-03-31	NaN	BEB LAC BIO LAT 900g	436874	3,69	22029900
575216	78989902002817	2022-03-31	NaN	GRAN. TROP 200g FITO	436874	6,99	19041000
575217	78989902004571	2022-03-31	NaN	GRAN. SAC 200g FITO	436874	6,25	19041000
575218	78989902006022	2022-03-31	NaN	GRAN. LIGHT 200g FIT	436874	5,99	19041000

575219 rows × 7 columns

Observa-se na imagem acima que nosso dataframe ainda possui valores nulos, vamos retirá-los para melhor expor os dados sem que os mesmos sejam enviesados por conter valores missing.

```
df = df.dropna()
```

```
df
```

	cod_produto	data_compra	categoria	desc_produto	id_compra	valor_produto	ncm_produto
5	78989901987026	2022-03-01	LATICINIOS	BISC RECH 115GR GLUB	19929	1,19	19053100
7	78989902002358	2022-03-01	LATICINIOS	LEITE DALIA INT 1LT	19930	3,29	4012010
25	78989902006136	2022-03-01	LIMPEZA	L.R UZZILIM 1kg LAVA	19933	4,99	34029090
27	6081	2022-03-01	CARNES	AGULHA BOV SO kg	19933	33,43	2012090
28	71511	2022-03-01	BEBIDAS	REF 3LT FANTA GUARAN	19933	6,99	22021000
...
575208	69176	2022-03-31	LATICINIOS	CISC TRAK CHOC BR 12	436874	2,39	19053100
575209	69176	2022-03-31	LATICINIOS	CISC TRAK CHOC BR 12	436874	2,39	19053100
575210	74208	2022-03-31	LATICINIOS	WAF ISABELA 100g CHO	436874	1,99	19053200
575211	74208	2022-03-31	LATICINIOS	WAF ISABELA 100g CHO	436874	1,99	19053200
575212	74210	2022-03-31	LATICINIOS	WAF ISABELA 100g MG	436874	1,99	19053200

```
192486 rows x 7 columns
```

```
df.isnull().sum()
```

```
cod_produto      0
data_compra      0
categoria         0
desc_produto     0
id_compra        0
valor_produto    0
ncm_produto      0
dtype: int64
```

Observa-se agora que os dados são exibidos sem os valores de categoria com valores null, porém ainda tenho mais uma questão para resolver que é a coluna do valor do produto quem além de ter vindo com vírgula preciso que esteja em float numérico, para, assim ter certeza de quem os cálculos envolvendo essa coluna estarão retornando valores calculados a partir das funções.

Alterado o tipo de coluna (valor_produto) para float:

```
df['valor_produto'] = df['valor_produto'].str.replace(',', '.')
```

```
df["valor_produto"] = df["valor_produto"].astype(float)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 192486 entries, 5 to 575212
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   cod_produto      192486 non-null object
1   data_compra      192486 non-null object
2   categoria        192486 non-null object
3   desc_produto     192486 non-null object
4   id_compra        192486 non-null int64
5   valor_produto    192486 non-null float64
6   ncm_produto      192486 non-null int64
dtypes: float64(1), int64(2), object(4)
memory usage: 11.7+ MB
```

```
df.head()
```

	cod_produto	data_compra	categoria	desc_produto	id_compra	valor_produto	ncm_produto
5	78989901987026	2022-03-01	LATICINIOS	BISC RECH 115GR GLUB	19929	1.19	19053100
7	78989902002358	2022-03-01	LATICINIOS	LEITE DALIA INT 1LT	19930	3.29	4012010
25	78989902006136	2022-03-01	LIMPEZA	L.R UZZILIM 1kg LAVA	19933	4.99	34029090
27	6081	2022-03-01	CARNES	AGULHA BOV SO kg	19933	33.43	2012090
28	71511	2022-03-01	BEBIDAS	REF 3LT FANTA GUARAN	19933	6.99	22021000

A categorização dos registros de produtos está relacionada a tabela de NCM (reduzida), podemos observar que os NCM dessa tabela são de apenas 5 dígitos, foi assim entregue, em função de termos uma categoria mais fácil de trabalhar no projeto. Um exemplo onde temos CARNES, nessa tabela temos animais vivos e uma divisão pela espécie de animal (bovino, suíno, etc...).

Para facilitar nossa análise e exploração o dataframe foi exportado para um segundo arquivo csv com o

```
df_ncm = pd.read_csv("d:/dados/tabela_reduzida_ncm.csv", sep=";")
```

Estrutura da tabela de NCM reduzida:

```
df_ncm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           7 non-null      int64
1   categoria    7 non-null      object
2   lista_cnm    7 non-null      object
dtypes: int64(1), object(2)
memory usage: 296.0+ bytes
```

```
df_ncm
```

	id	categoria	lista_cnm
0	1	CARNES	20120,20130,20230,20312,20319,20329,20629,2071...
1	2	BEBIDAS	22011,22021,22030,22041,22042,22086,22060
2	3	LATICINIOS	40110,40120,40150,40610,19053
3	4	LIMPEZA	28289,68051,68053,39232,33029,96039,29039,3808...
4	5	CEREAIS	10063,90300,71333,71340,11010,17011,11062,1104...
5	6	HIGIENE_PESSOAL	33043,96019,96032,96033,34011,33051,33061,33071
6	7	PET	23099,42010

4. Análise e Exploração dos Dados

A análise exploratória dos dados, se refere ao conjunto de técnicas e práticas iniciais referentes às investigações dos dados, a fim de, descobrir padrões, identificar possíveis anomalias, testar hipóteses e checar suposições. Através de técnicas estatísticas, os dados que inicialmente parecem confusos e desorganizados, são sumarizados, resumidos e por fim representados em forma de tabelas e gráficos. Essa análise busca construir uma narrativa a partir das informações obtidas e o cientista de dados se utiliza da parte visual como meio facilitador na compreensão da história contada (VANDERPLAS, 2016).

No dataframe, são identificados a **Compra** (o carrinho), os **Produtos**, as **Categorias** e **Preços** pagos por cada produto, dessa forma faremos nossa análise particionando um a um, avaliando variação de preço entre produto e categoria e posteriormente a análise de cesta de compras (market basket analysis) processo que examina padrões de compras de consumidores para determinar produtos que costumam ser adquiridos em conjunto. Um exemplo clássico do WalMart, que detectou um comportamento comum entre pais que compravam fraldas: grande parte deles levava também cervejas. A partir daí, a rede de supermercados teria alterado o layout das gôndolas de cervejas próximas às gôndolas de fraldas, o que resultou em um crescimento de 30% nas vendas de ambos os produtos.

Vamos verificar se existem duplicados, para garantir que estamos lidando com dados únicos nas análises.

```
df = df.drop_duplicates()
```

```
df
```

	Unnamed: 0	cod_produto	categoria	desc_produto	id_compra	valor_produto	ncm_produto
data_compra							
2022-03-01	5	78989901987026	LATICINIOS	BISC RECH 115GR GLUB	19929	1.19	19053100
2022-03-01	7	78989902002358	LATICINIOS	LEITE DALIA INT 1LT	19930	3.29	4012010
2022-03-01	25	78989902006136	LIMPEZA	L.R UZZILIM 1kg LAVA	19933	4.99	34029090
2022-03-01	27	6081	CARNES	AGULHA BOV SO kg	19933	33.43	2012090
2022-03-01	28	71511	BEBIDAS	REF 3LT FANTA GUARAN	19933	6.99	22021000
...
2022-03-31	575208	69176	LATICINIOS	CISC TRAK CHOC BR 12	436874	2.39	19053100
2022-03-31	575209	69176	LATICINIOS	CISC TRAK CHOC BR 12	436874	2.39	19053100
2022-03-31	575210	74208	LATICINIOS	WAF ISABELA 100g CHO	436874	1.99	19053200
2022-03-31	575211	74208	LATICINIOS	WAF ISABELA 100g CHO	436874	1.99	19053200
2022-03-31	575212	74210	LATICINIOS	WAF ISABELA 100g MG	436874	1.99	19053200

192486 rows x 7 columns

Em seguida vamos nos certificar da não existência de valores nulos e se for o caso trata-los

```
df.isnull().sum()
```

```
Unnamed: 0      0
cod_produto     0
categoria       0
desc_produto    0
id_compra       0
valor_produto   0
ncm_produto     0
dtype: int64
```

```
df.describe()
```

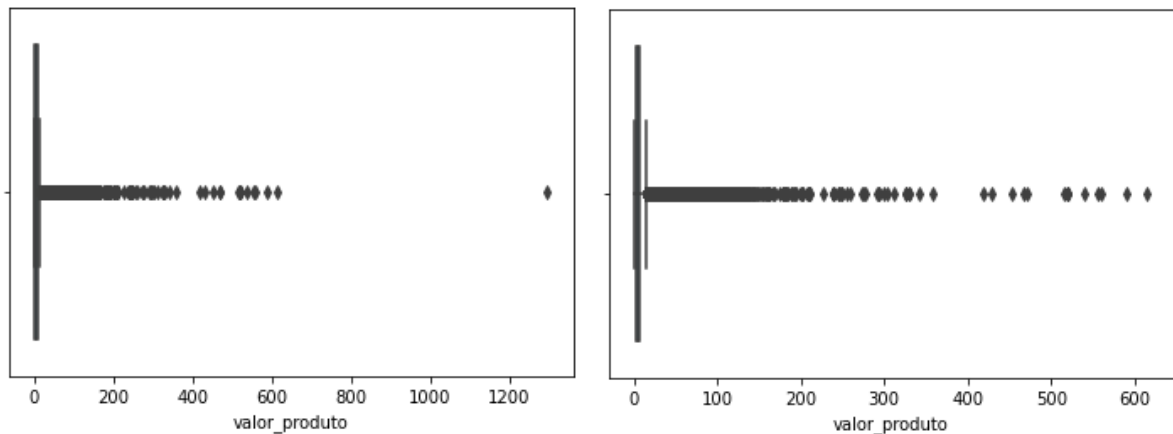
	Unnamed: 0	cod_produto	id_compra	valor_produto	ncm_produto
count	192486.000000	1.924860e+05	192486.000000	192486.000000	1.924860e+05
mean	283320.564914	4.664099e+13	319261.537156	7.131576	1.866813e+07
std	165262.159694	3.884295e+13	83668.246397	11.746499	1.395073e+07
min	5.000000	3.800000e+01	7732.000000	0.570000	2.012020e+06
25%	140779.250000	7.420800e+04	267653.000000	2.590000	4.061010e+06
50%	278380.500000	7.898990e+13	338235.000000	3.990000	1.905320e+07
75%	425539.750000	7.898990e+13	377924.000000	7.730000	2.203000e+07
max	575212.000000	7.898990e+13	436874.000000	1296.840000	9.603900e+07

A partir do retorno acima com base em nosso dataframe, podemos ver que, a média dos valores dos produtos é de R\$ 7,13 mas não é o que chama muito atenção sabendo que estamos analisando um atacado popular e há uma variação grande de preços nas mercadorias já o de maior valor R\$ 1.296,84 é interessante pelo mesmo motivo e investigando melhor, percebe-se através da imagem abaixo se trata de um erro, pois, o produto é um Biscoito.

```
1 SELECT m.data_compra,m.id_compra,m.categoria,desc_produto,max(valor) AS valor_maximo
2 FROM consumo_compra m
3
4
5
```

data_compra	id_compra	categoria	desc_produto	valor_maximo
2022-03-01	19.929	LATICINIOS	BISC RECH 115GR GLUB	1.296,84

Outliers podem ser detectados usando visualização, implementação de fórmulas matemáticas no conjunto de dados ou usando a abordagem estatística. Como a ilustrada abaixo a partir da geração de gráficos tipo boxplot do valor dos itens e pode-se observar o erro já identificado acima. Na mesma imagem ilustra-se os 2 gráficos com e sem o Outlier.

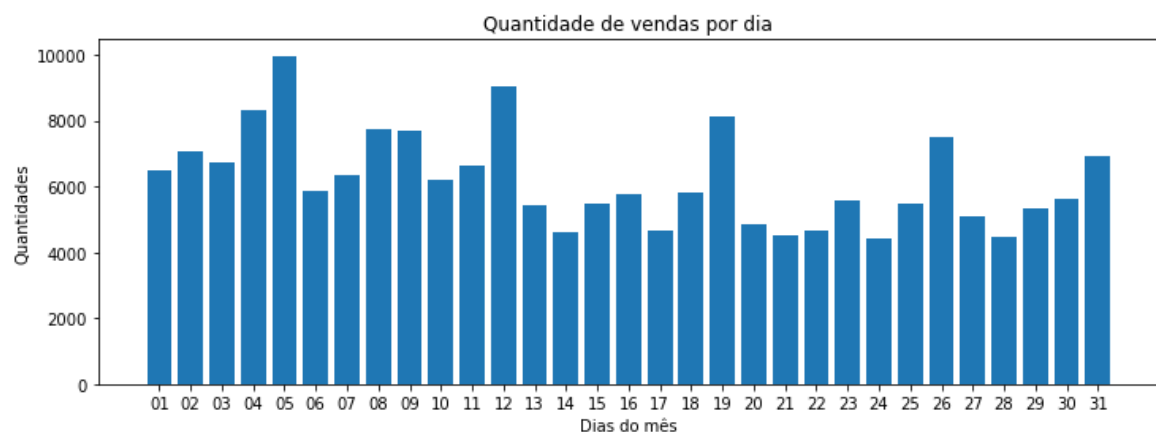


Também poderíamos o Z-Score e entender a que distância o ponto de dados está da média. E depois de definir um valor limite, e utilizar os valores de pontuação z dos pontos de dados para definir os valores discrepantes, mas como os produtos de maior valor são carnes e eles podem sim se distanciar da média, preferi não usar o score-z para remoção do outlier para não perder dados que podem ser importantes nas análises.

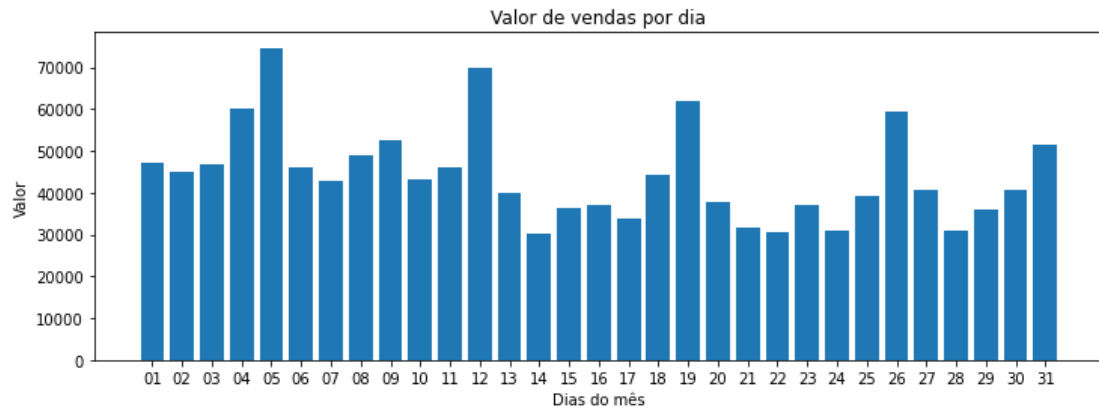
```
z = np.abs(stats.zscore(df['valor_produto']))
print(z.head(10))
```

```
0    0.505818
1    0.327041
2    0.182317
3    2.238837
4    0.012053
5    0.327041
6    0.134643
7    0.256381
8    0.267449
9    0.728595
Name: valor_produto, dtype: float64
```

Análise da Vendas no período contido no dataframe (03/2022)

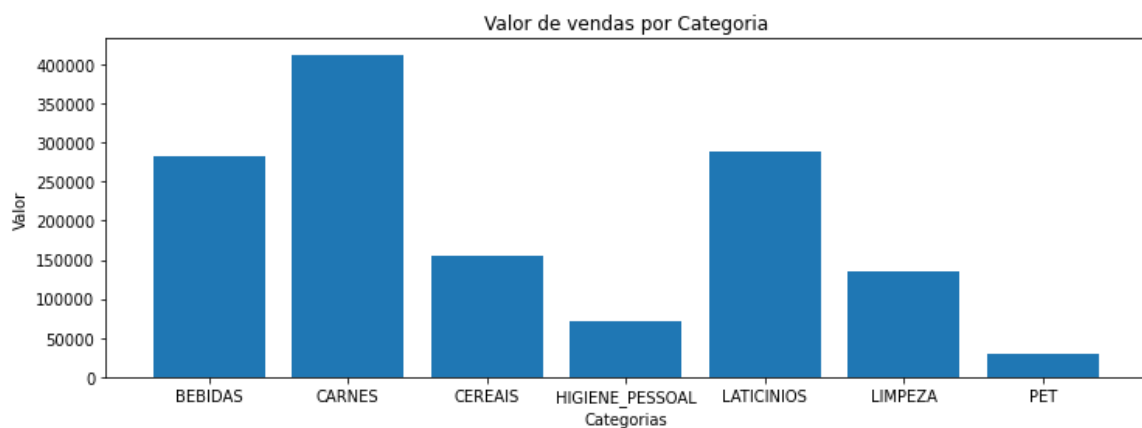


Observa-se que a maior quantidade de vendas mesmo é no início do mês, geralmente quando a população recebe seus salários.



Já no gráfico acima a quantidade de vendas em produtos parece acompanhar os valores, ou seja, duas variáveis diretamente proporcionais.

Análise de Vendas x Categoria



E entre as categorias analisadas a de mais valor em venda é carnes, o que não deixa de estar coerente as pessoas consomem uma grande variedade de carnes.

5. Criação de Modelos de Machine Learning

Nos dias de hoje as organizações de varejo produzem diariamente um enorme volume de dados transacionais sobre suas vendas. A análise da cesta de compras é um dos métodos mais populares para se extrair informações úteis de um banco de dados grande (Hahsler, Chelluboina, 2011).

Regressão Linear

Foi selecionado da nossa amostra os dias do mês e o valor gasto em cada dia e realizado uma análise para procurar ali uma relação entre as variáveis x (dias do mês) e (valores gasto), sendo assim, colocamos a prova de uma equação linear em que nos ilustra através de gráfico o quão estão relacionadas essas variáveis através de um coeficiente de relação R ;

Com a análise de regressão, se assume que uma variável dependente (y) é influenciada por uma variável independente (x).

A regressão linear foi uma maneira de mostrar de forma gráfica a relação de variáveis como dia de compra e valor gasto, para tentar identificar se havia alguma relação entre elas.

Onde:

$R = 1$: Significa uma forte relação entre as variáveis e positiva;

$R = 0$: Significa inexistência de relação entre as variáveis;

$R = -1$: Significa uma forte relação entre as variáveis, porém, negativa;

$R > 0$: Significa uma relação positiva entre x e y ;

$R < 0$: Significa uma relação negativa entre x e y ;

Submetendo nossa amostra a esse modelo tivemos os seguintes resultados:

Para gerar o gráfico acima foi necessário converter a data em dias (numéricos), pois esses dados só suportam variáveis contínuas numéricas. Dessa forma:

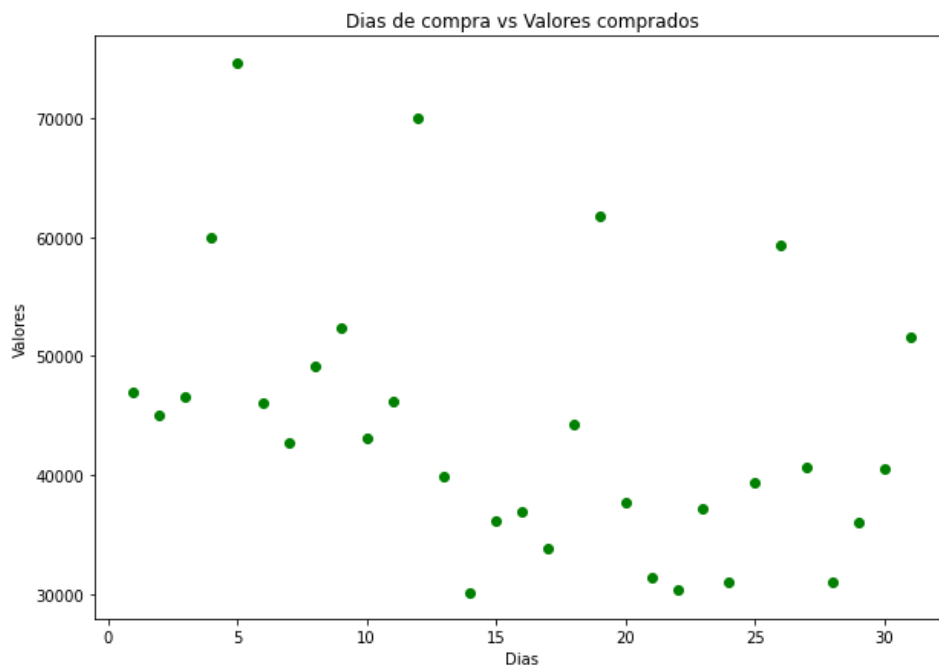
```
dfd['data_compra'] = dfd['data_compra'].dt.day
```

```
dfd.rename(columns={'data_compra': 'dia_compra'}, inplace = True)
```

```
dfd
```

	dia_compra	valor_produto
0	1	47020.98
1	2	45015.73
2	3	46633.62
3	4	59985.23
4	5	74546.46
5	6	46041.47

```
plt.figure(figsize=(10, 7))
plt.scatter(dfd.dia_compra, dfd.valor_produto, color='g')
plt.title('Dias de compra vs Valores comprados')
plt.xlabel('Dias')
plt.ylabel('Valores')
plt.show()
```



Dias do mês: Variáveis independentes, entradas;

Valores gastos : Variáveis dependentes, saídasostas.

Em uma Regressão Linear é comum denotar as saídas com y e as entradas com x . Se houver duas ou mais Variáveis Independentes, elas podem ser representadas como um vetor $\mathbf{x} = (x_1, \dots, x_r)$, onde r (ou n) é o número de entradas.

Vamos utilizar uma reta de melhor ajuste para ver se conseguimos uma reta que fique o mais próximo possível de todos os pontos, tanto os pontos acima da linha quanto os abaixo.

Para criar essa reta de melhor ajuste nós precisamos dos melhores valores possíveis para os termos **m** e **b** para a amostra que estamos trabalhando mas para isso precisamos calcular m e b vamos usar a forma abaixo (Método dos Mínimos Quadrados Ordinários)

$$m = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$b = \bar{y} - m * \bar{x}$$

x = independent variables
 \bar{x} = average of independent variables
 y = dependent variables
 \bar{y} = average of dependent variables

Executando um algoritmo em python obtemos o seguinte resultado:

```
dfd['(x_i - x_mean)'] = dfd['dia_compra'] - dfd['dia_compra'].mean()
dfd['(y_i - y_mean)'] = dfd['valor_produto'] - dfd['valor_produto'].mean()
dfd['(x_i - x_mean)(y_i - y_mean)'] = dfd['(x_i - x_mean)'] * dfd['(y_i - y_mean)']
dfd['(x_i - x_mean)^2'] = (dfd['dia_compra'] - dfd['dia_compra'].mean())**2

m = (sum(dfd['(x_i - x_mean)'] * dfd['(y_i - y_mean)'])) / sum(dfd['(x_i - x_mean)^2'])
b = dfd['valor_produto'].mean() - (m * dfd['dia_compra'].mean())

print("Angular Coefficient (m): {0}\nLinear Coefficient (b): {1}".format(round(m), round(b)))

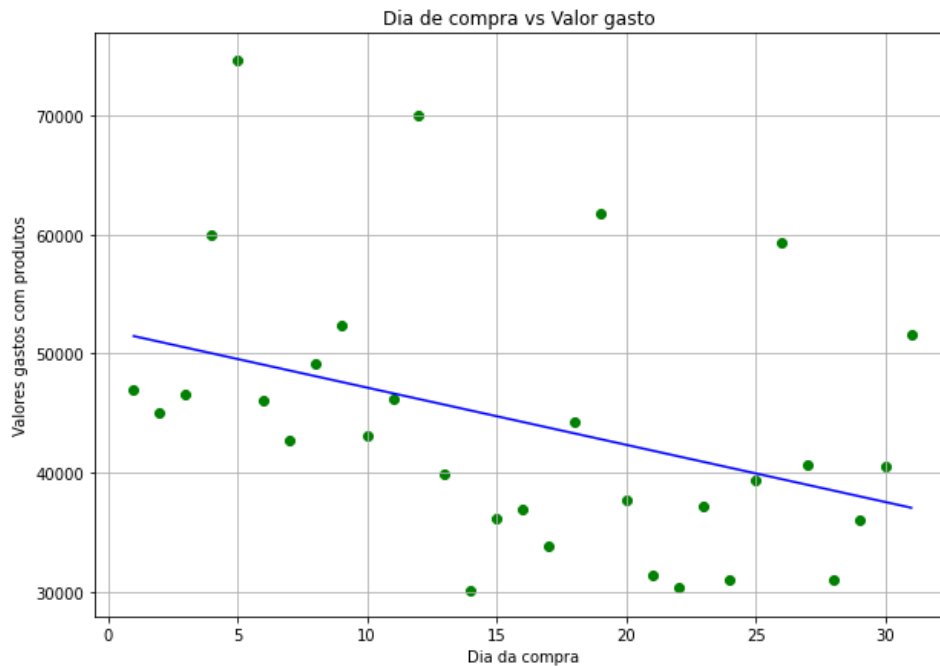
Angular Coefficient (m): -479
Linear Coefficient (b): 51953
```

```
dfd['(x_i - x_mean)'] = dfd['dia_compra'] - dfd['dia_compra'].mean()
dfd['(y_i - y_mean)'] = dfd['valor_produto'] - dfd['valor_produto'].mean()
dfd['(x_i - x_mean)(y_i - y_mean)'] = dfd['(x_i - x_mean)'] * dfd['(y_i - y_mean)']
dfd['(x_i - x_mean)^2'] = (dfd['dia_compra'] - dfd['dia_compra'].mean())**2

m = (sum(dfd['(x_i - x_mean)'] * dfd['(y_i - y_mean)'])) / sum(dfd['(x_i - x_mean)^2'])
b = dfd['valor_produto'].mean() - (m * dfd['dia_compra'].mean())

regression_line = [(m*x) + b for x in dfd['dia_compra']]

plt.figure(figsize=(10, 7))
plt.scatter(dfd.dia_compra, dfd.valor_produto, color='g')
plt.plot(dfd.dia_compra, regression_line, color='b')
plt.title('Dia de compra vs Valor gasto ')
plt.xlabel('Dia da compra')
plt.ylabel('Valores gastos com produtos')
plt.grid()
plt.show()
```



Observa-se assim que há uma relação negativa entre as variáveis x e y e que quanto mais próximo do fim do mês há um menor valor em de compras, ou seja aumenta x e diminui y .

Regras de Associação

Análise de cestas de compras que faz uso de regras de associação para identificar padrões nos hábitos de compra dos clientes e com isso, fornecendo informações do perfil de consumo, conhecer o melhor público e se estender a estratégias de marketing com campanhas direcionadas e obter um melhor aproveitamento.

A regra de associação pode ser feita através de um processo exaustivo computacionalmente, e que resulta em um conjunto de regras bastante expressivo mesmo com um conjunto de dados pequeno. Uma alternativa elegante para esse cálculo é já definir o suporte e confiança na parametrização do algoritmo para que haja a poda de regras que não atendam esse valor mínimo aceitável. As regras de associação descritiva é uma tarefa descritiva de aprendizado não supervisionado, não temos dados rotulados.

Em uma Regressão Linear é comum identificar a interdependências entre as entradas x e saídas y . Se houver duas ou mais Variáveis Independentes, elas podem ser representadas como um vetor $\mathbf{x} = (x_1, \dots, x_r)$, onde r (ou n) é o número de entradas.

Analisando o certo de compras através de regras de associação, lembrando que quanto maior a amostra em estivermos trabalhando maior será nossa combinação de regras, vai além de um fatorial em função de que são mais de uma variável para uma variável.

Vamos ilustrar um modelo hipotético, ou melhor foi retirado da nossa amostra, mas somente algumas compras para conseguirmos mostra na prática o como é realizado a busca de padrões em um carrinho de compras. Foi selecionado alguns dos itens de compra da nossa amostra e os IDs únicos de cada compra;

ID	AÇUCAR	AGUA	ARROZ	BISCOITO	BOMBOM	LEITE
19938	1	0	1	1	1	1
111898	1	1	1	0	0	0
111914	0	0	0	1	0	1
111940	0	1	1	0	0	0
205409	0	1	1	1	0	1
205413	0	0	0	0	1	1
205416	0	1	0	1	0	1
205439	0	0	1	0	0	0
205475	0	0	0	0	1	1
243380	1	1	1	1	1	1

Ao se observar a nossa tabela acima de 10 compras ou 10 cestas, e selecionado aleatoriamente os itens de produtos que também foram comprados nas amostras trabalhadas até então.

Aplicando as regras de associação podemos fazer algumas suposições.

Suporte de 50% para os itens (leite e biscoito) isso por que aparecem juntos 5 vezes em 10 carrinhos. $(5/10)=0,5$

Confiança é de 71% por que das 5 vezes em que aparece (leite e biscoito) o leite aparece 7 vezes $(5/7)=0,71$ dos clientes que compraram leite também compraram biscoito.

Lift de 1,42, indica que os clientes que compram leite têm uma chance 1,42 vezes maior de comprar biscoito.

A imagem abaixo tem o objetivo de ilustrar as formulas de Suporte, Confiança e Lift.

Rule: $X \Rightarrow Y$

$$\text{Support} = \frac{\text{freq}(X, Y)}{N}$$

$$\text{Confidence} = \frac{\text{freq}(X, Y)}{\text{freq}(X)}$$

$$\text{Lift} = \frac{\text{Support}}{\text{Supp}(X) \times \text{Supp}(Y)}$$



Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B \& C \Rightarrow D$	1/5	1/3	5/9











6. Interpretação dos Resultados

Esse trabalho foi realizado, buscando se uma forma diferente de ver os dados e que pudessem ser interpretados e sugeridos de interpretação com gráficos e suas explicações para que o leitor possa entender melhor do que se trata.

Posso dizer, por mim, que foi uma trabalho de muita busca de conhecimento e aplicação sempre buscando a melhor forma de encontrar uma informação escondida e que possa ser de grande valia para quem necessita.

Uma rápida interpretação seria que os dados recebidos foram estressados ao ponto de revelarem algo escondido e tenho certeza de quem tem muito mais.

7. Apresentação dos Resultados

<p>PREDICTION TASK </p> <p>Type of task? Entity on which predictions are made? Possible outcomes? Wait time before observation?</p> <p>O objetivo desse trabalho foi realizar uma exploração na base de dados e verificar se havia padrão de consumo, em que a organização não tem conhecimento por não se aprofundar nos dados.</p>	<p>DECISIONS </p> <p>How are predictions turned into proposed value for the end-user? Mention parameters of the process / application that does that.</p> <p>Decisões são tomadas a partir de alguma experiência anterior ou estudo de um caso específico, assim como o ML através de uma exploração de dados procura experiências ocultas, essa base de dados nos mostrou muito isso.</p>	<p>VALUE PROPOSITION </p> <p>Who is the end-user? What are their objectives? How will they benefit from the ML system? Mention workflow/interfaces.</p> <p>O beneficiário é sempre o usuário do ML proposto, nesse caso foram identificados outliers que possivelmente o cliente não tenha percebido.</p> <p>Como um exemplo um erro grande nos valores de um produto quem nem mesmo a contabilidade talvez o perceba, a não ser que se faça uma exploração da forma que foi realizada. Mas o que empolga num trabalho desses é ver que tem espaço seja em dados como em ferramentas para se explorar os dados quase que de forma infinita.</p>	<p>DATA COLLECTION </p> <p>Strategy for initial train set & continuous update. Mention collection rate, holdout on production entities, cost/constraints to observe outcomes.</p> <p>Os dados foram recebidos de um banco de dados, um ERP organizacional em cliente para quem pudéssemos fazer essa análise e dispor de dados ainda não detectados, mas recebemos um arquivo csv para a coleta e transformação.</p>	<p>DATA SOURCES </p> <p>Where can we get (raw) information on entities and observed outcomes? Mention database tables, API methods, websites to scrape, etc.</p> <p>Essa amostra trabalhada foi entregue pela organização, ou seja, exportada de um banco de dados, com os dados de uma cliente, tanto que foram tratados de forma a não infringir a LGPD.</p>
<p>IMPACT SIMULATION </p> <p>Can models be deployed? Which test data to assess performance? Cost/gain values for (in)correct decisions? <u>Fairness constraint?</u></p> <p>Sim, os modelos podem ser implantados e há mundo muito maior ainda a ser explorado com outras ferramentas quem podem nos trazer mais informações sobre os dados, espera-se com isso que as organizações trabalhem mais os dados, pois, há muito dado não estruturado dentro das organizações precisando de tratamento e organização para futuros processos de ML e ou um DW.</p>	<p>MAKING PREDICTIONS </p> <p>When do we make real-time / batch pred? Time available for this + featurization + post-processing? Compute target?</p> <p>Há hoje, uma manancial de ferramentas muito boas e disponíveis no mercado para se fazer qualquer coisa nesse sentido de dados, ainda faltam pessoas querem serem capacitadas para esse fim.</p>	<p>BUILDING MODELS </p> <p>How many prod models are needed? When would we update? Time available for this (including featurization and analysis)?</p> <p>A necessidade de modelos de ML depende muito da organização, há aqueles que nem se quer sabe de suas existências já outras mais voltadas a dados (data driven) essas já inclusive se preparam e planejam sem novos modelos.</p>	<p>FEATURES </p> <p>Input representations: available at prediction time, extracted from raw data sources.</p> <p>Nesse trabalho foram gerados alguns gráficos interessantes e que podem sim futuramente serem utilizados como modelo na exploração de dados.</p>	
	<p>MONITORING</p> <p>Metrics to quantify value creation and measure the ML system's impact in production (on end-users and business)?</p>	<p>Todo o processo de inteligência artificial tem seu impacto sobre onde irá se instalar, desde o aprimoramento de muitas tarefas manuais como a previsão de variáveis que são úteis e muito suscetíveis a falha humana, porém, há caso em que a tecnologia, e não é de hoje, quem vem retirando postos de trabalho e por isso precisa ser bem medido para não termos aí um problema social, ou por falta de trabalho ou por que as pessoas não se atualizam profissionalmente buscando treinamentos.</p>		

8. Links

Aqui você deve disponibilizar os links para o vídeo com sua apresentação de 5 minutos e para o repositório contendo os dados utilizados no projeto, scripts criados, etc.

Link para o vídeo: <youtube.com/...>

Link para o repositório: <github.com/...>

REFERÊNCIAS

Escovedo, Tatiana (2020-02-27T22:58:59). **Introdução a Data Science**. Casa do Código. Edição do Kindle.

<https://www.gov.br/receitafederal/pt-br/assuntos/aduana-e-comercio-exterior/classificacao-fiscal-de-mercadorias/ncm>

ASSAF NETO, Alexandre. Mercado Financeiro. 4. ed. - São Paulo : Atlas, 2001

<https://pt.stackoverflow.com/questions/485440/customizar-personalizar-legenda-do-gr%C3%A1fico-em-python> (acessado em 06/04/2023)

<https://acervolima.com/lidando-com-linhas-e-colunas-no-pandas-dataframe/> (acessado em 06/04/2022)

<https://drigols.medium.com/regress%C3%A3o-linear-e-gradiente-descendente-do-zero-a-bruxaria-c4d1484357e0> (acessado em 08/04/2022)

Hahsler M, Chelluboina S (2011). "Visualizing Association Rules in Hierarchical Groups." In 42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization
VANDERPLAS, J. Python Data Science Handbook. 1. ed. United States of America: O'Reilly Media, 2016.

APÊNDICE

Programação/Scripts

Gráficos

