Universidade Federal de Minas Gerais (UFMG)

Departamento de Ciência da Computação

Guilherme Mendes de Oliveira

guilhermemendes@ufmg.br

MINERAÇÃO DE PADRÕES FREQUENTES EM BASE DE DADOS VIAGENS

Belo Horizonte, MG – Brasil

2022

Guilherme Mendes de Oliveira

MINERAÇÃO DE PADRÕES FREQUENTES EM BASE DE DADOS VIAGENS

Trabalho prático de mineração de padrões frequentes da matéria de Mineração de Dados ministrada pelo professor Wagner Meira Jr.

Belo Horizonte, MG – Brasil

2022

1. Introdução

Este trabalho tem como objetivo a implementação de uma técnica de mineração de dados para identificação de padrões frequentes em uma base de dados. Para tal atividade foi utilizado o algoritmo ***FPGrowth*.**

Além disso a escolha para base de dados foi feita a partir de um gerador randômico[1] criado para um *Datathon* no ano de 2019, de forma aleatória cria uma base de viagens com localidades, companhias aéreas e usuários, a base foi escolhida numa tentativa de fugir dos exemplos clássico de utilização de cesta de compras de supermercado que é comum na literatura associada a esse algorítmo. A base gerada segue os "requisitos" que o conjunto deve ter, tendo itens e registros a nível de transação.

Os parâmetros da geração das viagens foram alterados em relação ao código original, foram incluídos mais destinos completando as capitais do Brasil, alguns destinos nas Américas, esse *array* vai ser comportar como os "itens do estoque" no exemplo clássico da cesta de compras do supermercado.

```
#- Places
defPlacesName = ['Aracaju','Belem','Belo Horizonte','Boa Vista','Brasília','Campo Grande','Cuiaba',
                 'Curitiba','Florianopolis','Fortaleza','Goiania','Joao Pessoa','Macapa','Maceio',
                 'Manaus','Natal','Palmas','Porto Alegre','Recife','Rio Branco','Rio de Janeiro',
                 'Salvador','São Luis','Sao Paulo','Teresina','Vitoria',
                 'Buenos Aires','Montevideu','Lima','Bogotá',
                 'Orlando','Toronto','Quebec','Cidade do Mexico','Miami','Las Vegas','Sao Franscisco']
```

Figura 1 – *Array* de destinos.

Outro parâmetro ajustado foi a quantidade de transações que são geradas, para tal instanciado uma quantidade de usuários distintos dentro da base de 25 mil e uma quantidade de viagens no intervalo de 1 a 7, gerada de forma aleatória, para cada um desses 25 mil usuários

```
defCompanies = {
    'HHD': {'usersCount': 25000}
}
```

Figura 2 – Dicionário de companhia aérea e quantidade de usuários distintos.

```
defFlightsInterval = {'min': 1, 'max': 7}
```

Figura 3 – Dicionário de intervalo de quantidade de viagens por usuário.

[1] O link para o código do gerador da base de dados estará nas referências

2. Motivação/ justificativa

A aplicação das técnicas de modelagem envolvidas no trabalho juntamente com a utilização de um algoritmo em uma base de dados que se aproxima de problemas reais encontrados no cotidiano é além de uma forma de avaliação uma oportunidade de desenvolver todas as competências técnicas necessárias para aplicar futuramente esse conhecimento no "mundo real" ou seja uma forma de ganhar experiência analítica dos resultados e competências técnicas de programação para exploração, limpeza dos dados , aplicação do algoritmo e extração de métricas para avaliação dos resultados.

3. Objetivo

A aplicação das técnicas de mineração de padrões frequentes nessa base de dados de viagens, busca simular um problema de negócio por parte de uma companhia aérea. Entender um pouco mais o fluxo migratório de clientes para auxiliar na tomada de decisões de ações de marketing como criação de pacotes, recomendações de roteiros minerados e até futura exploração de novos destinos que sejam similares.

4. Metodologia

A metodologia utilizada foi inspirada no CRISP-DM, inspirada porque uma das etapas, mais especificamente a do entendimento do negócio, é realizada de forma simulada uma vez que que proposição de um problema só é possível após uma verificação preliminar da base de dados, ou ela é proposta e partir disso que os dados são buscados. Além disso, ela está direcionada a ser um problema que seja possível explorar e propor soluções com base em uma técnica específica, nesse caso a mineração de padrões frequentes.

Além disso, a etapa de *deployment* também não é pertinente uma vez que é uma prática e nesse caso não há um problema real a ser tratado, para as demais etapas todas foram possíveis e foram aplicadas para a conclusão do trabalho prático.

O fluxo de Entendimento, preparação, modelagem e avaliação podem ser vistos aqui.

5. Desenvolvimento
5.1. Business Understanding

Essa etapa compreende o entendimento do problema, como foi propositivo a partir da base de dados e já direcionado para a aplicação de mineração de padrões frequentes esta etapa não se aplica.

5.2. Data Understanding

A base utilizada possui um total de 198.878 linhas e 12 colunas e não possui linhas com valores nulos em nenhum de seus atributos, a base possui 25.000 usuários diferentes e 99.439 viagens distintas.

Verificando a distribuição das partidas e destinos observamos que as distribuições são bem semelhantes o que indica que o algoritmo gerador da base criou uma base “uniforme”, mas que a princípio isso não vai ser um impeditivo para o prosseguimento do trabalho.

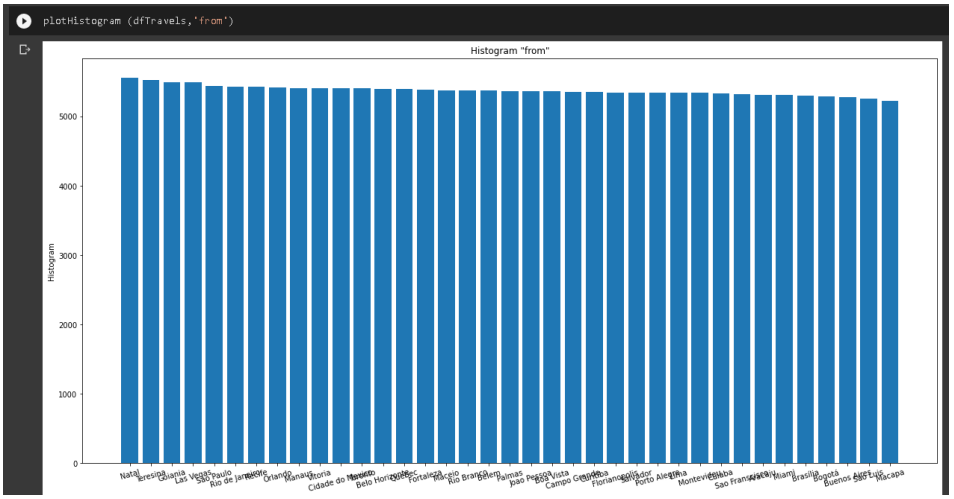


Figura 4 – Histograma das cidades de partida.

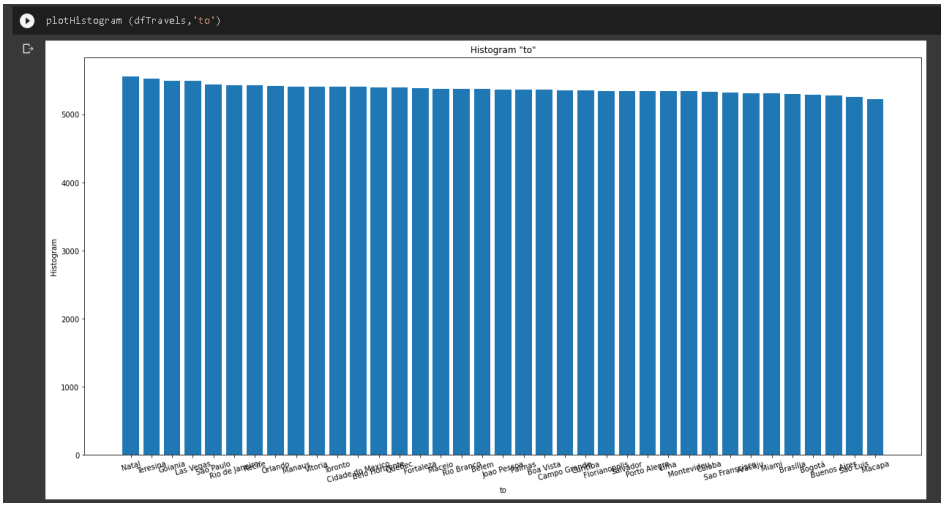


Figura 5 – Histograma das cidades de destino.

No entendimento da base de dados verificamos que a mesma está apta para a aplicação da técnica de mineração uma vez que possui uma quantidade considerável de registros, possui os atributos relacionados às rotas e está num grão transacional.

5.3. Data Preparation

Na etapa de preparação inicialmente foi feita a renomeação das colunas a partir de campos do próprio arquivo da base, posteriormente foi concatenado os campos "*from*" e *"to"* em um novo atributo denominado rota uma vez que juntos eles representam a ideia de item de uma cesta.

```
dfTravels['rota'] = dfTravels.Partida.str.cat(dfTravels.Destino, sep='-')
dfTravels.head(5)
```

	Tid	Partida	Destino	distance	agency	flightType	price	time	timeMsg	userCode	travelCode	date	rota
0	0	Porto Alegre	Macapa	271.15	CloudFy	premium	897.47	0.68	0:40h	0	0	2022-10-15 12:37:33.931741	Porto Alegre-Macapa
1	1	Macapa	Porto Alegre	271.15	CloudFy	premium	1200.83	0.68	0:40h	0	0	2022-10-16 12:37:33.931741	Macapa-Porto Alegre
2	2	Montevideu	Palmas	349.58	CloudFy	premium	1512.58	0.87	0:52h	0	1	2022-10-25 12:39:42.965289	Montevideu-Palmas
3	3	Palmas	Montevideu	349.58	CloudFy	premium	1678.33	0.87	0:52h	0	1	2022-10-28 12:39:42.965289	Palmas-Montevideu
4	4	Vitoria	Maceio	715.76	CloudFy	premium	4546.43	1.79	1:47h	0	2	2022-10-25 12:39:42.970718	Vitoria-Maceio

Figura 6 – Criação do campo rota.

Foi feito uma extração para computar apenas uma "perna" da viagem, uma vez que automaticamente quando temos uma passagem de ida é gerada uma volta, foi realizada essa extração para diminuir o tamanho da base sem que impactasse na corretude da mineração.

```
### Tratamento para obter apenas uma "perna" da viagem, quando temos uma passagem de ida automaticamente temos uma passagem de volta gerada
dfTravels['Rank'] = dfTravels.groupby(by=['travelCode'])['Partida'].transform(lambda x: x.rank())
dfTravels.head(10)
```

	Tid	Partida	Destino	distance	agency	flightType	price	time	timeMsg	userCode	travelCode	date	rota	Rank
0	0	Porto Alegre	Macapa	271.15	CloudFy	premium	897.47	0.68	0:40h	0	0	2022-10-15 12:37:33.931741	Porto Alegre-Macapa	2.0
1	1	Macapa	Porto Alegre	271.15	CloudFy	premium	1200.83	0.68	0:40h	0	0	2022-10-16 12:37:33.931741	Macapa-Porto Alegre	1.0
2	2	Montevideu	Palmas	349.58	CloudFy	premium	1512.58	0.87	0:52h	0	1	2022-10-25 12:39:42.965289	Montevideu-Palmas	1.0
3	3	Palmas	Montevideu	349.58	CloudFy	premium	1678.33	0.87	0:52h	0	1	2022-10-28 12:39:42.965289	Palmas-Montevideu	2.0
4	4	Vitoria	Maceio	715.76	CloudFy	premium	4546.43	1.79	1:47h	0	2	2022-10-25 12:39:42.970718	Vitoria-Maceio	2.0
5	5	Maceio	Vitoria	715.76	CloudFy	premium	4452.67	1.79	1:47h	0	2	2022-10-26 12:39:42.970718	Maceio-Vitoria	1.0
6	6	Las Vegas	Sao Franscisco	660.78	CloudFy	premium	3937.31	1.65	1:39h	0	3	2022-10-25 12:39:42.979634	Las Vegas-Sao Franscisco	1.0
7	7	Sao Franscisco	Las Vegas	660.78	CloudFy	premium	4143.90	1.65	1:39h	0	3	2022-10-27 12:39:42.979634	Sao Franscisco-Las Vegas	2.0
8	8	Vitoria	Rio Branco	406.15	CloudFy	premium	2047.56	1.02	1:1h	0	4	2022-10-25 12:39:42.982898	Vitoria-Rio Branco	2.0
9	9	Rio Branco	Vitoria	406.15	CloudFy	premium	3305.82	1.02	1:1h	0	4	2022-10-26 12:39:42.982898	Rio Branco-Vitoria	1.0

Figura 7 – Extração somente das rotas de ida - Rank.

```
dfRotasIda = dfTravels.loc[dfTravels['Rank'] == 1.0]
dfRotasIda.head(10)
```

	Tid	Partida	Destino	distance	agency	flightType	price	time	timeMsg	userCode	travelCode	date	rota	Rank
1	1	Macapa	Porto Alegre	271.15	CloudFy	premium	1200.83	0.68	0:40h	0	0	2022-10-16 12:37:33.931741	Macapa-Porto Alegre	1.0
2	2	Montevideu	Palmas	349.58	CloudFy	premium	1512.58	0.87	0:52h	0	1	2022-10-25 12:39:42.965289	Montevideu-Palmas	1.0
5	5	Maceio	Vitoria	715.76	CloudFy	premium	4452.67	1.79	1:47h	0	2	2022-10-26 12:39:42.970718	Maceio-Vitoria	1.0
6	6	Las Vegas	Sao Franscisco	660.78	CloudFy	premium	3937.31	1.65	1:39h	0	3	2022-10-25 12:39:42.979634	Las Vegas-Sao Franscisco	1.0
9	9	Rio Branco	Vitoria	406.15	CloudFy	premium	3305.82	1.02	1:1h	0	4	2022-10-26 12:39:42.982898	Rio Branco-Vitoria	1.0
10	10	Porto Alegre	Salvador	415.96	CloudFy	premium	2318.88	1.04	1:2h	0	5	2022-10-25 12:39:42.986193	Porto Alegre-Salvador	1.0
13	13	Belo Horizonte	Buenos Aires	257.82	CloudFy	premium	950.36	0.64	0:38h	1	6	2022-10-17 12:37:33.931741	Belo Horizonte-Buenos Aires	1.0
14	14	Cidade do Mexico	Toronto	253.55	CloudFy	premium	1180.44	0.63	0:37h	2	7	2022-10-15 12:37:33.931741	Cidade do Mexico-Toronto	1.0
16	16	Cidade do Mexico	Recife	700.86	CloudFy	premium	4717.24	1.75	1:45h	2	8	2022-10-25 12:39:42.996841	Cidade do Mexico-Recife	1.0
18	18	Porto Alegre	Recife	610.03	CloudFy	premium	3809.59	1.53	1:31h	2	9	2022-10-25 12:39:43.000471	Porto Alegre-Recife	1.0

Figura 8 – Extração somente das rotas de ida - Extração.

.

Nessa nova base gerada conseguimos encontrar 666 rotas distintas e tendo o top 10 disposto conforme imagem abaixo.

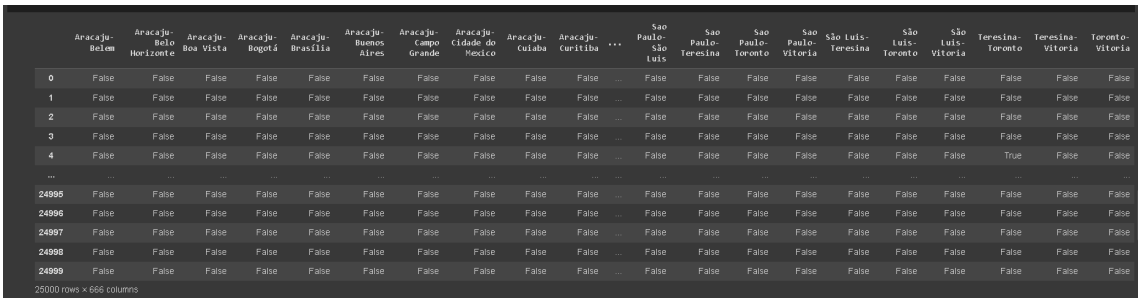
```
dfCountRotas = pd.DataFrame(dfRotasIda['rota'].value_counts().reset_index(name='QtdViagens'),index=None)
dfCountRotas.rename(columns={'index':'rota'}, inplace=True)
dfCountRotas.head(10)
```

	rota	QtdViagens
0	Buenos Aires-Sao Franscisco	189
1	Maceio-Natal	183
2	Maceio-Palmas	182
3	Bogotá-Brasília	182
4	Goiania-Rio de Janeiro	182
5	Campo Grande-Fortaleza	182
6	Curitiba-Sao Paulo	181
7	Quebec-Recife	180
8	Joao Pessoa-Rio Branco	180
9	Goiania-Rio Branco	178

Figura 9 – Top 10 de rotas mais frequentes.

5.4 Modelagem

Na etapa de modelagem da base para a aplicação do algoritmo de mineração (Fp Growth), foram selecionadas os campos dos usuários e suas respectivas rotas voadas em uma espécie de lista, para que fosse possível criar uma tabela transacional na qual cada coluna representa uma coluna e a instância de 1 ou 0 para as rotas voadas por cada usuário, dispostos em linhas.



	Aracaju-Belem	Aracaju-Belo Horizonte	Aracaju-Boa Vista	Aracaju-Bogotá	Aracaju-Brasília	Aracaju-Buenos Aires	Aracaju-Campo Grande	Aracaju-Cidade do Mexico	Aracaju-Cuiaba	Aracaju-Curitiba	...	Sao Paulo-São Luis	Sao Paulo-Teresina	Sao Paulo-Toronto	Sao Paulo-Vitoria	São Luis-Teresina	São Luis-Toronto	São Luis-Vitoria	Teresina-Toronto	Teresina-Vitoria	Toronto-Vitoria
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	True	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
24995	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
24996	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
24997	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
24998	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
24999	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False

25000 rows × 666 columns

Figura 10 – Tabela transacional de usuários e rotas.

6. Resultados experimentais e análise
6.1. Evaluation

Os parâmetros utilizados para o algoritmo foi somente o suporte mínimo considerado para considerar um item ou um conjunto de itens frequentes, o parâmetro foi determinístico utilizando uma medida “subjetiva” de suporte mínimo igual a 8, a partir da execução do algoritmo foram identificados 668 conjuntos frequentes dos quais apenas dois são expressos por mais de uma rota no mesmo conjunto e podem ser verificados ao computarmos as regras de associações dentro do conjunto de padrões frequentes.

A métrica utilizada para extrair as regras de associação foi o *lift* que traduz um grau de surpresa/decepção em relação ao valor de ocorrência esperado. Para as quatro regras identificadas tivemos um valor bem acima de 1 o que indica um grau de co-ocorrência muito valioso, considerando que as rotas são independentes elas co-ocorrem na base em mais de 7 vezes do que o esperado. No entanto, quando verificamos o *leverage* que indica o valor absoluto da diferença do suporte observado e o esperado entre as rotas do conjunto frequente esse valor é muito baixo, além disso os suportes em si também não são muito relevantes.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(Manaus-Salvador)	(Salvador-São Luis)	0.00516	0.00560	0.00032	0.062016	11.074197	0.000291	1.060145
(Salvador-São Luis)	(Manaus-Salvador)	0.00560	0.00516	0.00032	0.057143	11.074197	0.000291	1.055133
(Belo Horizonte-Las Vegas)	(Montevideu-Sao Paulo)	0.00704	0.00640	0.00032	0.045455	7.102273	0.000275	1.040914
(Montevideu-Sao Paulo)	(Belo Horizonte-Las Vegas)	0.00640	0.00704	0.00032	0.050000	7.102273	0.000275	1.045221

Figura 11 – Regras de associação.

7. Conclusões e perspectivas

Inicialmente a busca por uma base que fugisse um pouco da temática de carrinho de compras, exemplo clássico da literatura dessas técnicas de mineração de padrões frequentes, foi complexa e difícil, a motivação para fugir desse contexto "clichê" foi tentar explorar utilizações diferentes e problematizações diferentes, no entanto embora a temática e a utilização proposta para os resultados apontasse um caminho diferente ainda seguiu o padrão de recomendação o então o esforço em pensar algo fora da caixa nao ter resultado na descoberta de uma base útil foi algo bem frustrante.

Durante o desenvolvimento e o andamento do projeto os processos fluíram sem maiores dificuldades uma vez que os conceitos estavam bem fixados e a base atendia bem ao critério transacional, no entanto na etapa de entendimento dos etapas a suspeita que pela distribuição bem uniforme das rotas as regras extraídas poderiam não ser muito significativas se confirmou ao final da execução do algoritmo e a avaliação dos resultados gerados.

Embora o resultado do ponto de vista da problemática não ter sido assertivo uma vez que as regras geradas não foram capaz de apontar um caminho de recomendação de roteiros, a conclusão é satisfatória, pois possibilitou completar todas as etapas da *pipeline* de dados, desde o entendimento até a avaliação das regras de associação e a conclusão de que não foi possível indicar uma recomendação de roteiros útil.

8.Referências

Zaki, M and Meira, W. Data Mining and Machine Learning: Fundamental Concepts and Algorithms. Capítulo 8: Itemset Mining;

Zaki, M and Meira, W. Data Mining and Machine Learning: Fundamental Concepts and Algorithms. Capítulo 12: Pattern and Rule Assessment;

<http://rasbt.github.io/mlxtend/ > . Acesso em: 11 de outubro de 2022.