

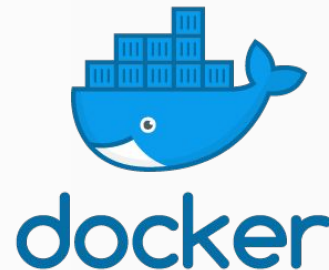


## Trabalho Prático II - GRADI

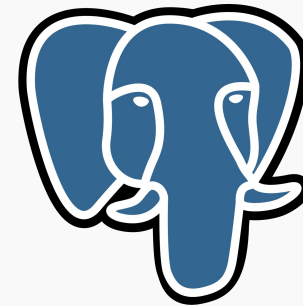
Bruno Marra (3029), Daniel Freitas (2304), Gustavo Viegas (3026), Vítor Luis (3045)

# Tecnologias Utilizadas

- Docker
- Vertabelo - Sistema web para modelagem de SGBDS
- **PostgreSQL - SGBD para os DBs**
- Luigi / Python3 - Pipelines para ETL
- Metabase - Gráficos e Relatórios
- Weka - Inferência dos dados
- Uma surpresa...

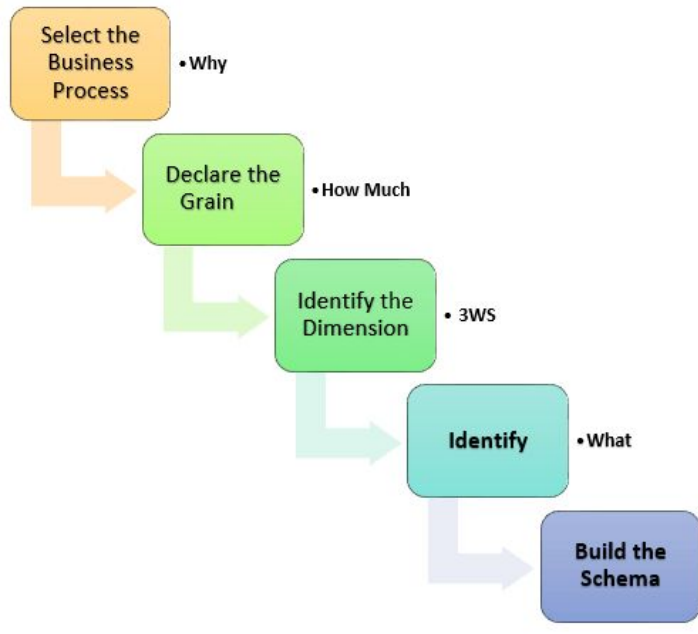


Luigi



# Modelagem

- Principal processo do negócio: **Sinistros**.
- Granularidade: **Anual**.
- Dimensões: **Veículos, Ano, Clientes, Seguros**.
- Fatos: **Sinistros**.
- Modelagem mais simples: Estrela.
  - Poderia ser modificada se houvesse necessidade durante a análise dos dados.
    - Não houve.



# Modelagem

veiculo		
veiculo_id	int	PK
cor	varchar(45)	
valor_compra	double precision	
data_compra	date	
nome_modelo	varchar(45)	
nome_montadora	varchar(45)	
pais_origem_montadora	varchar(45)	

ano		
ano_id	int	PK
ano	int	

sinistro		
data_sinistro	date	
valor	double precision	
veiculo_id	int	PK FK
ano_id	int	PK FK
cliente_id	int	PK FK
seguro_id	int	PK FK

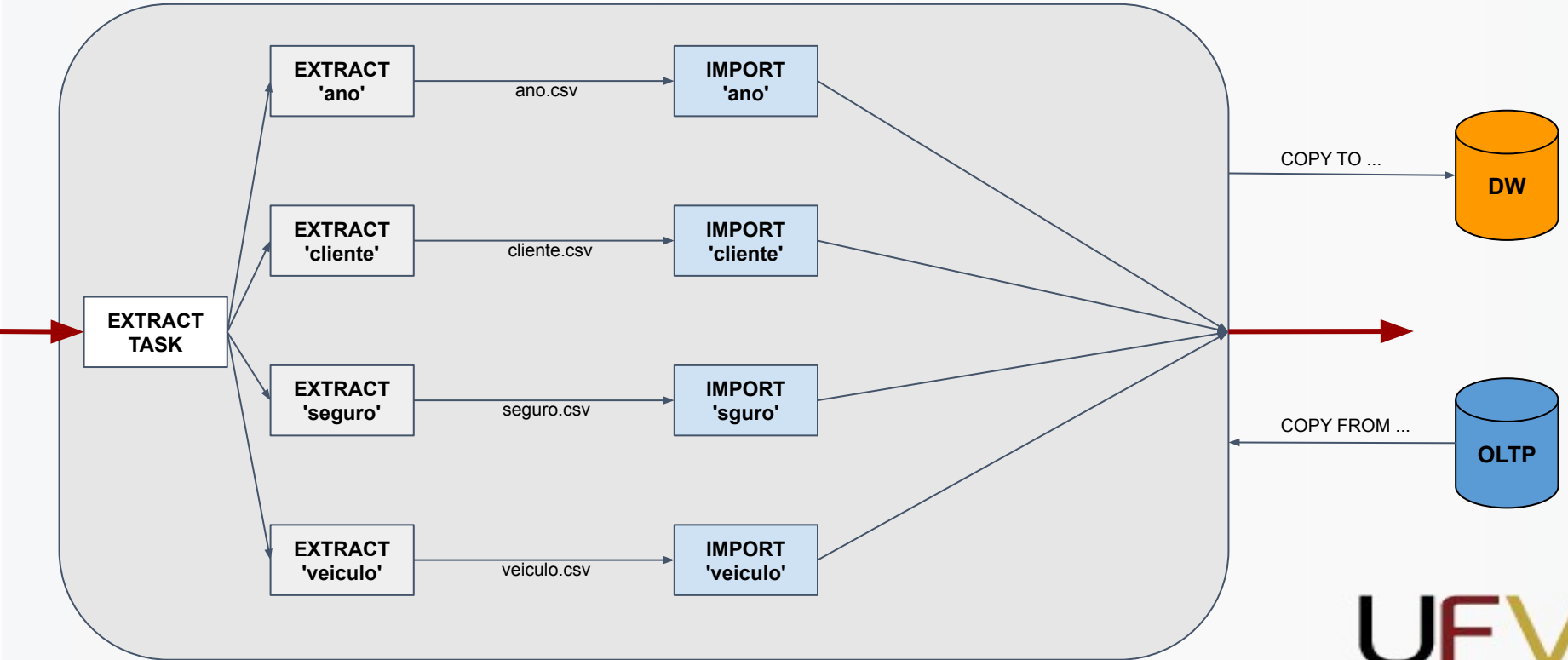
Tabelas de Fatos

Tabelas de Dimensões

cliente		
cliente_id	int	PK
regiao	varchar(45)	
estado	varchar(45)	
cidade	varchar(45)	
telefone	varchar(45)	
celular	varchar(45)	
sexo	char(1)	
cep	char(9)	
data_nasc	date	

seguro		
seguro_id	int	PK
valor_segurado	double precision	
premio	double precision	
ano_vigencia	int	

# ETL



# Cubos de Dados

- PostgreSQL permite uma facilidade quanto à criação de cubos de dados:
- Exemplo - *cubeavgveiculos*:

```
CREATE VIEW cubeavgveiculos AS
SELECT nome_montadora AS montadora, nome_modelo AS
       modelo, AVG(valor_compra) AS media
FROM veiculo
GROUP BY CUBE (nome_montadora, nome_modelo);
```

# Possíveis perguntas

- “Qual a média do valor de compra mais alto dentre os veículos da seguradora?”

```
SELECT montadora, modelo, media  
FROM public.cubeavgveiculos  
ORDER BY media DESC  
LIMIT 1;
```

# Possíveis perguntas

- “Qual a média do valor de compra mais alto dentre os veículos da seguradora?” R: **R\$315.739,49** (Camaro - Chevrolet)

Data Output Explain Messages Notifications			
	montadora character varying (45)	modelo character varying (45)	media double precision
1	chevrolet	camaro	315739.48860240757



# Possíveis perguntas

- “Quais modelos tem causado maior prejuízo para a seguradora, com relação ao custo de seus sinistros?”

```
SELECT nome_modelo, nome_montadora,  
       ROUND(CAST(custo_sinistro AS NUMERIC), 2)  
       AS custo_sinistro, qtdsinistros  
FROM public.cubesinistroveiculo  
WHERE nome_modelo IS NOT NULL  
       AND nome_montadora IS NOT NULL  
ORDER BY custo_sinistro DESC
```

# Possíveis perguntas

- “Quais modelos tem causado maior prejuízo para a seguradora, com relação ao custo de seus sinistros?”

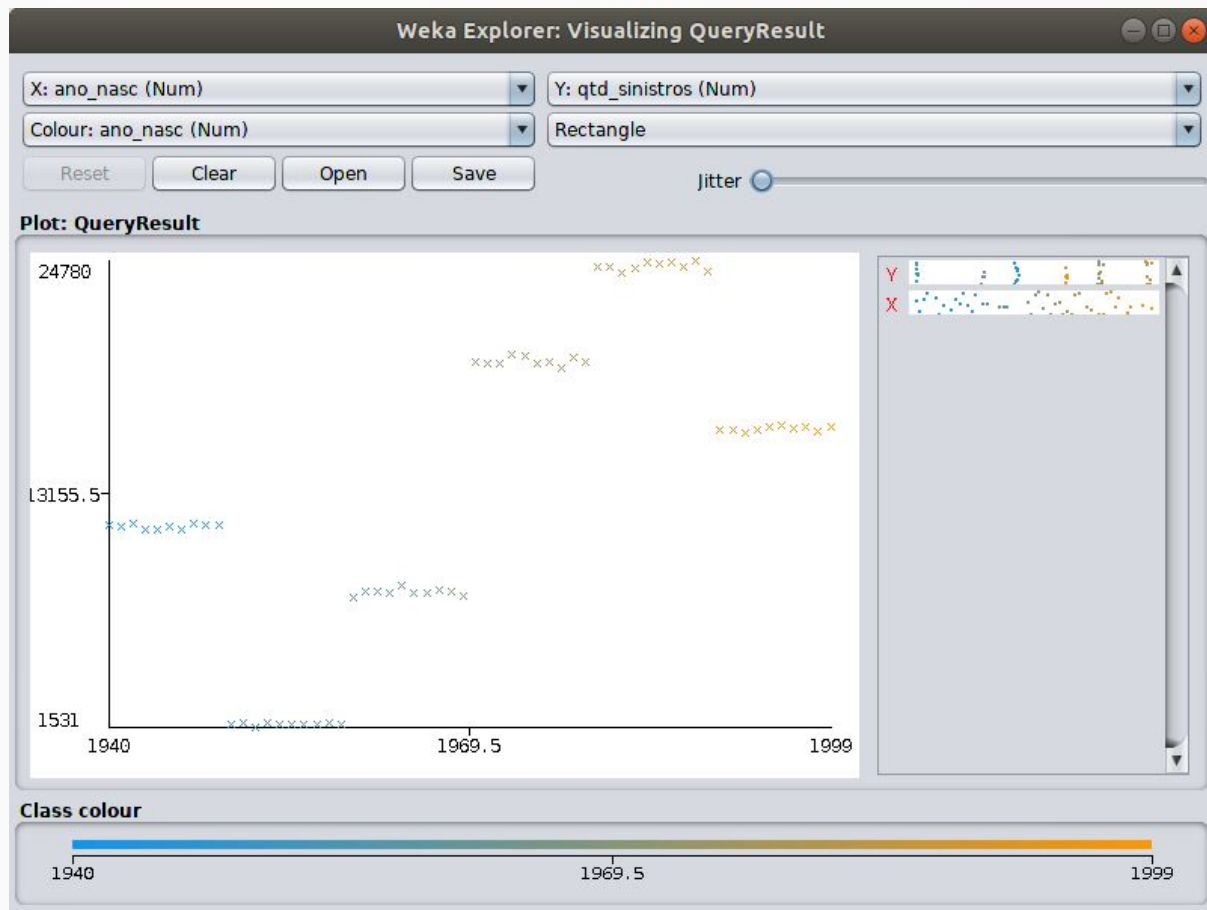
	Data Output	Explain	Messages	Notifications
	nome_modelo character varying (45)	nome_montadora character varying (45)	custo_sinistro numeric	qtdsinistros bigint
1	civic	honda	2224359651.25	53367
2	passat	volkswagen	1321770942.80	26706
3	fit	honda	895606548.22	86290
4	city	honda	756021853.51	80953
5	toro	fiat	635293601.97	25653
6	uno	fiat	506888074.33	61245
7	gol	volkswagen	501609493.42	63298
8	hilux	toyota	469187292.35	9342
9	camaro	chevrolet	451777553.75	6314
10	up	volkswagen	446942467.23	59001
11	mobi	fiat	430622682.73	56859
12	nolo	volkswagen	408575888.38	39062

# Possíveis perguntas



- “Qual o perfil das pessoas que possuem mais sinistros vinculados: os mais jovens ou os mais velhos?”

```
SELECT COUNT(*) AS qtd_sinistros,  
       EXTRACT(YEAR FROM data_nasc) AS ano_nasc  
FROM cliente  
NATURAL JOIN sinistro  
GROUP BY ano_nasc;
```



**Cluster mode**

☒ Use training set

☐ Supplied test set

☐ Percentage split %

☐ Classes to clusters evaluation

☒ Store clusters for visualization

Result list (right-click for options)

**Clusterer output**

XMeans

=====

Requested iterations : 1

Iterations performed : 1

Splits prepared : 2

Splits performed : 0

Cutoff factor : 0.5

Percentage of splits accepted by cutoff factor : 0 %

-----

Cutoff factor : 0.5

-----

Cluster centers : 2 centers

Cluster 0

20196.833333333332 1984.5

Cluster 1

7126.233333333334 1954.5

Portion: 12.73477

Value : -48.992135

Time taken to build model (full training data) : 0.01 seconds

Model and evaluation on training set ===

Clustered Instances

30 ( 50%)

30 ( 50%)

## XMeans

**Cluster 0: ano  $\geq$  1984.5 : 20197**  
sinistros aproximadamente

**Cluster 1: ano  $<$  1984.5 : 7126**  
sinistros aproximadamente

# Relatórios e Gráficos



# Divisão em três Dashboards

- Compras de veículos
- Dados de Veículos
- Ocorrência de Sinistros

Afinal...





# Ocorrência de Sinistros

**SINISTRO MESMO É DEIXAR OS  
TELETUBBIES EM PRETO E BRANCO...**



# Não, esse Dash Ocorrência de sinistros

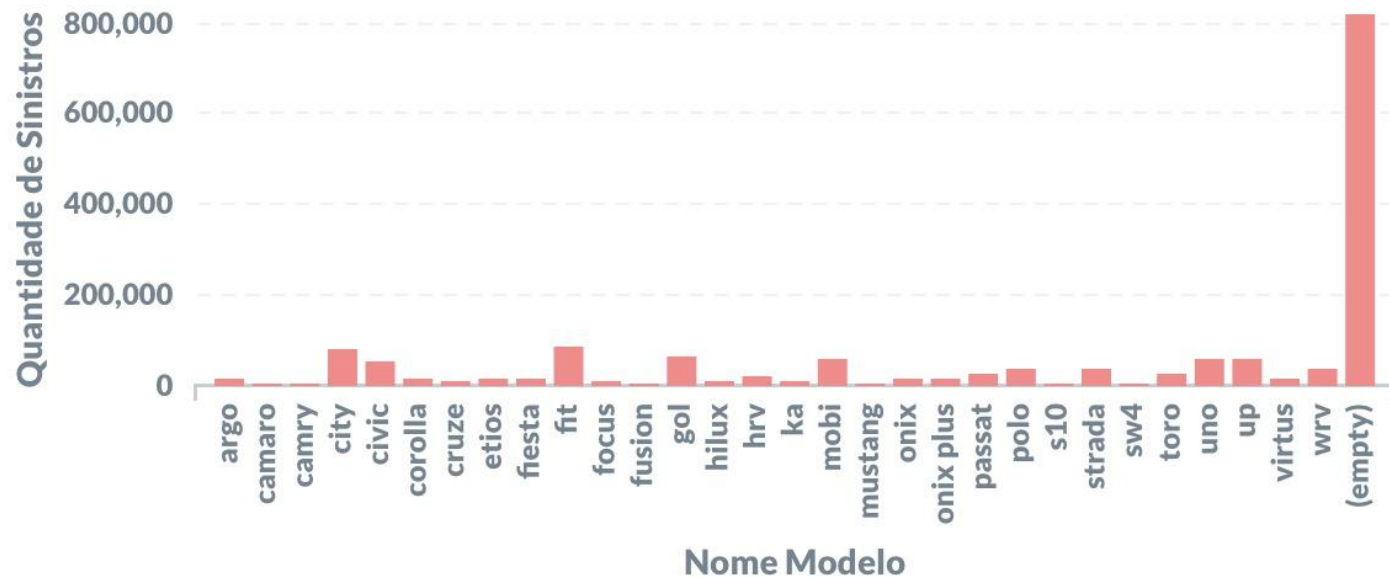
**\$12,671,295,561.94**

Total gasto em sinistros

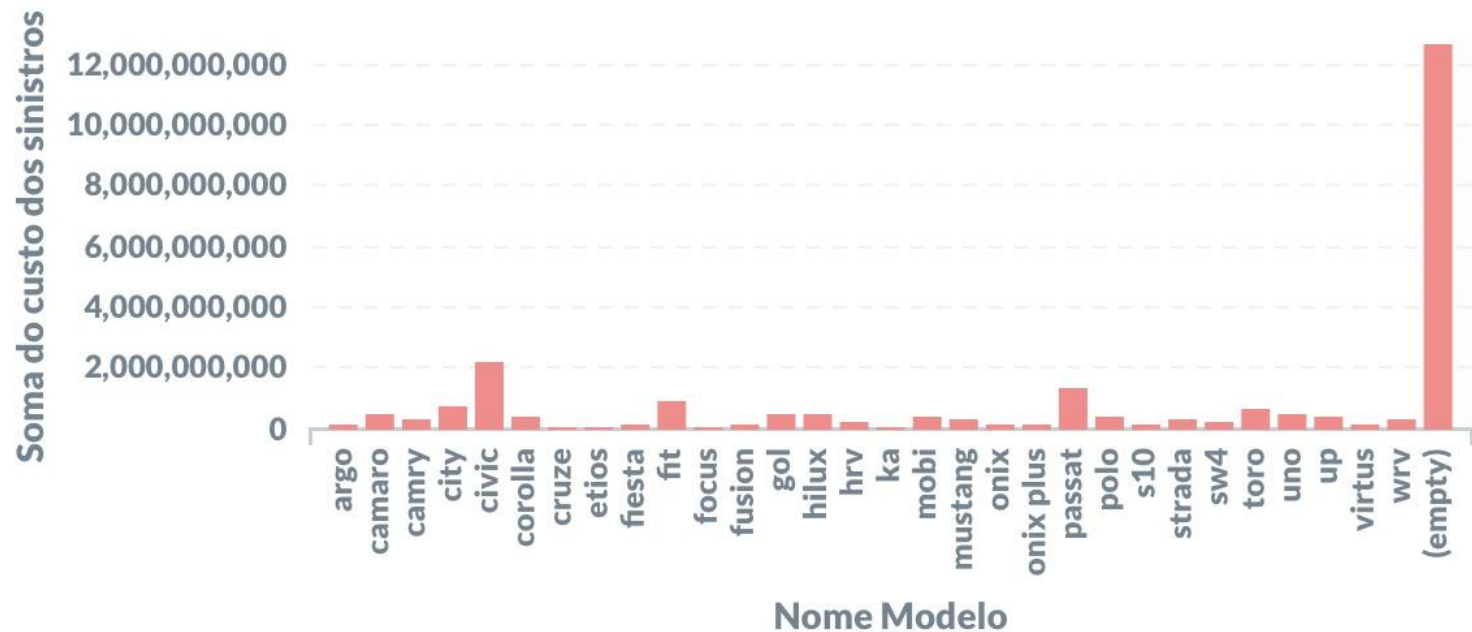
**819,692**

Quantidade de sinistros ocorridos

Quantidade de sinistros por modelo

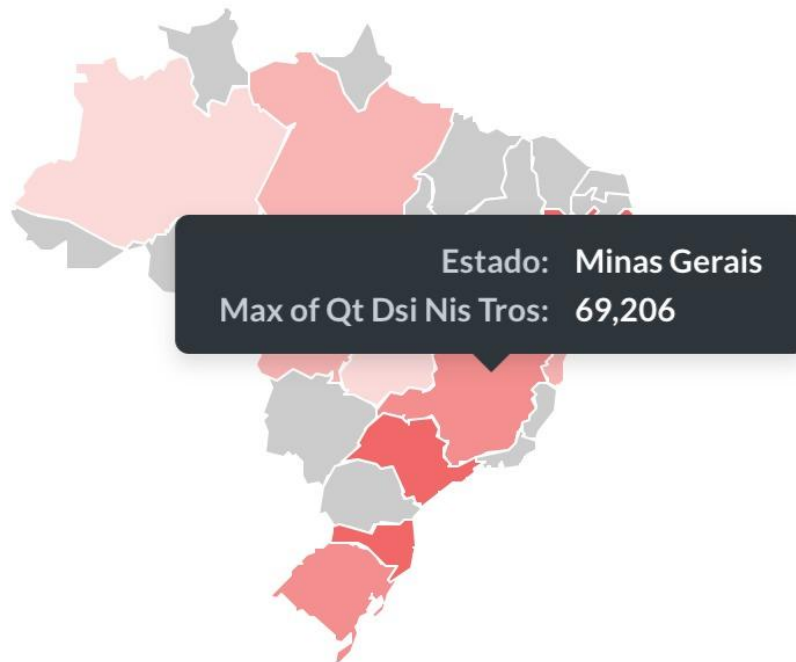


Prejuízo em \$ por modelo 



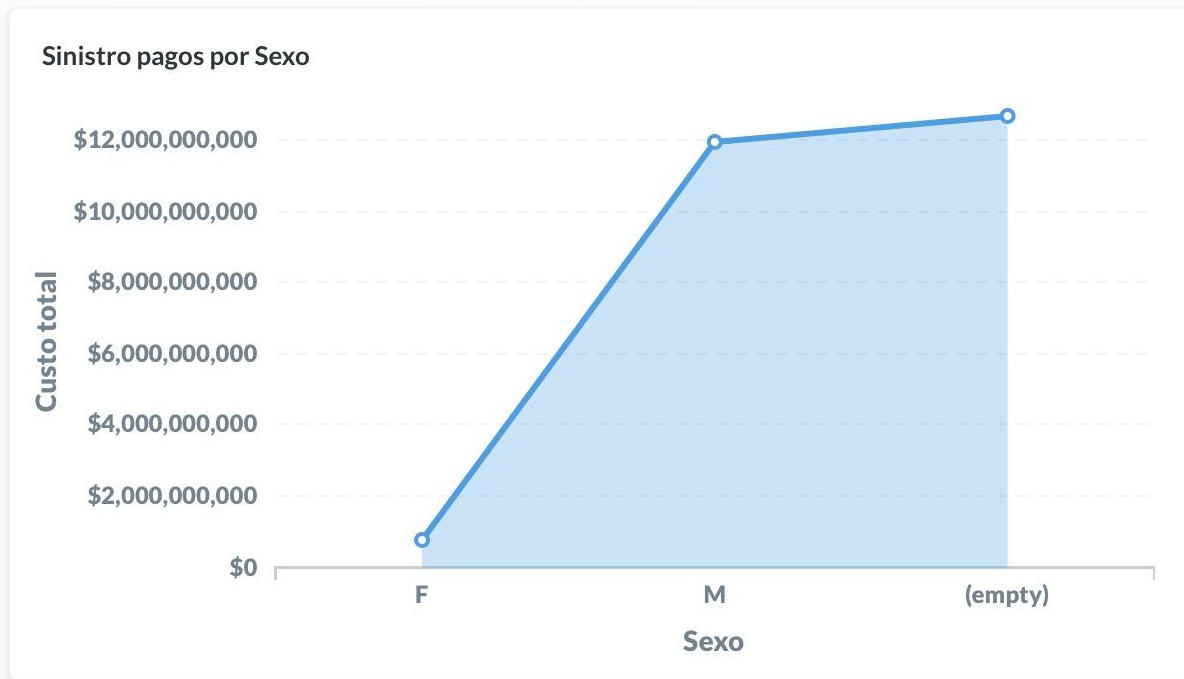
### Quantidade de sinistros ocorridos por região

- 13.1k - 16.4k
- 44.1k - 49.1k
- 66.3k - 69.2k
- 167.7k - 179...
- 819.7k +





# Mas não é bem assim...



# Inferências

- Qual a probabilidade de um cliente sofrer um sinistro?
  - Dois modelos foram criados: NaiveBayers e J48!
- NaiveBayers;
  - Fórmula matemática com base nos dados.
- J48.
  - Árvore de decisão.



# Inferências

- NaiveBayers modelo.

```
=== Summary ===
```

Correctly Classified Instances	524849	60.3339 %
Incorrectly Classified Instances	345058	39.6661 %
Kappa statistic	0.2159	
Mean absolute error	0.45	
Root mean squared error	0.479	
Relative absolute error	90.2392 %	
Root relative squared error	95.9294 %	
Total Number of Instances	869907	

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,488	0,269	0,667	0,488	0,564	0,225	0,645	0,709	true
	0,731	0,512	0,563	0,731	0,636	0,225	0,645	0,573	false
Weighted Avg.	0,603	0,384	0,618	0,603	0,598	0,225	0,645	0,644	

```
=== Confusion Matrix ===
```

a	b	<-- classified as
223237	233812	a = true
111246	301612	b = false

# Inferências

- J48 modelo.

```
=== Summary ===
```

Correctly Classified Instances	134480	61.8364 %
Incorrectly Classified Instances	82997	38.1636 %
Kappa statistic	0.2388	
Mean absolute error	0.4312	
Root mean squared error	0.4784	
Relative absolute error	86.4555 %	
Root relative squared error	95.8012 %	
Total Number of Instances	217477	

```
=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,586	0,346	0,652	0,586	0,617	0,240	0,667	0,703	true
	0,654	0,414	0,589	0,654	0,620	0,240	0,667	0,612	false
Weighted Avg.	0,618	0,378	0,622	0,618	0,618	0,240	0,667	0,659	

```
=== Confusion Matrix ===
```

a	b	<-- classified as
66839	47250	a = true
35747	67641	b = false

# Inferências V2

- Desenvolvemos nosso próprio frontend para previsões de sinistros.
- Utiliza **TODOS** os dados do warehouse disponível (e relevantes).
- Em alguns casos tem precisão de 100% em ocorrência de sinistros!
  - Em média 84% +/-
  - Pelo menos 10% melhor que do Weka
- Rede Neural, ADABOOST, Gradiente Estocástico (com vetor de suporte), Floresta Aleatória.
  - Rede Neural o mais equilibrado, Floresta Aleatória o mais enfático.

# DEMO



Obrigado!! XD



# Caso sobre tempo...

# Cubos de Dados - *cubeavgveiculos*

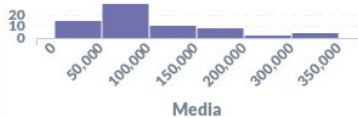
## Summary

67

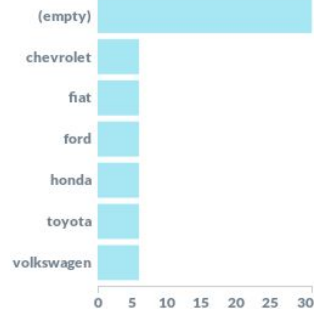
Total Cube Avg Ve Icu Los

## How these Cube Avg Ve Icu Los are distributed

Cube Avg Ve Icu Los by Media



Cube Avg Ve Icu Los per Mon Tad Ora



warehouse/admin@Datawarehouse

Query Editor

Query History

```
1 select * from cubeavgveiculos limit 100;
```

Data Output

Explain

Messages

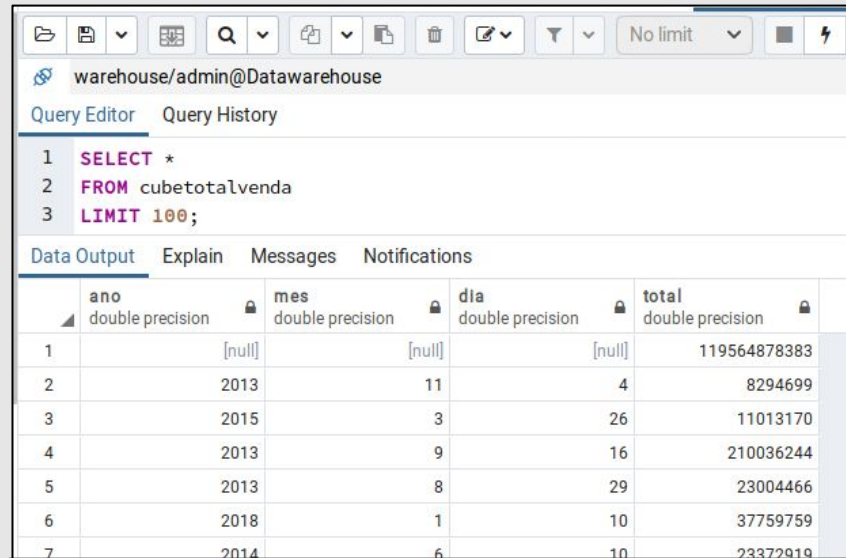
Notifications

	montadora character varying (45)	modelo character varying (45)	media double precision
1	[null]	[null]	79790.24109803869
2	toyota	corolla	103750.62808884445
3	fiat	argo	78743.57351048659
4	volkswagen	gol	48741.71763326465
5	chevrolet	s10	152776.00385038505
6	chevrolet	onix plus	45768.50712516601
7	fiat	mobi	43734.49404173188

**CUBE (nome\_montadora,  
nome\_modelo)**

# Cubos de Dados - *cubetotalvenda*

```
CREATE VIEW cubetotalvenda AS
SELECT
  EXTRACT (YEAR FROM data_compra) ano,
  EXTRACT (MONTH FROM data_compra) mes,
  EXTRACT (DAY FROM data_compra) dia,
  SUM (valor_compra) AS total
FROM veiculo
GROUP BY
  CUBE (
    EXTRACT (YEAR FROM data_compra),
    EXTRACT (MONTH FROM data_compra),
    EXTRACT (DAY FROM data_compra)
  );
```



The screenshot shows a SQL query editor interface for a warehouse. The query is: `SELECT * FROM cubetotalvenda LIMIT 100;`. Below the query, the 'Data Output' tab is active, displaying a table with 7 rows and 5 columns: `ano`, `mes`, `dia`, and `total`. The first row shows null values for the date fields and a total of 119564878383. The subsequent rows show data for the years 2013, 2015, and 2018.

	ano double precision	mes double precision	dia double precision	total double precision
1	[null]	[null]	[null]	119564878383
2	2013	11	4	8294699
3	2015	3	26	11013170
4	2013	9	16	210036244
5	2013	8	29	23004466
6	2018	1	10	37759759
7	2014	6	10	23372919

CUBE (ano\_compra,  
mes\_compra,  
dia\_compra)



# Cubos de Dados - *cubetotalvenda*

## Summary

**2,807**

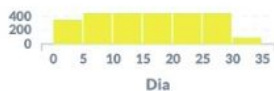
Total Cube Total Venda

## How these Cube Total Venda are distributed

Cube Total Venda by Mes



Cube Total Venda by Dia



Cube Total Venda by Total



Cube Total Venda by Ano

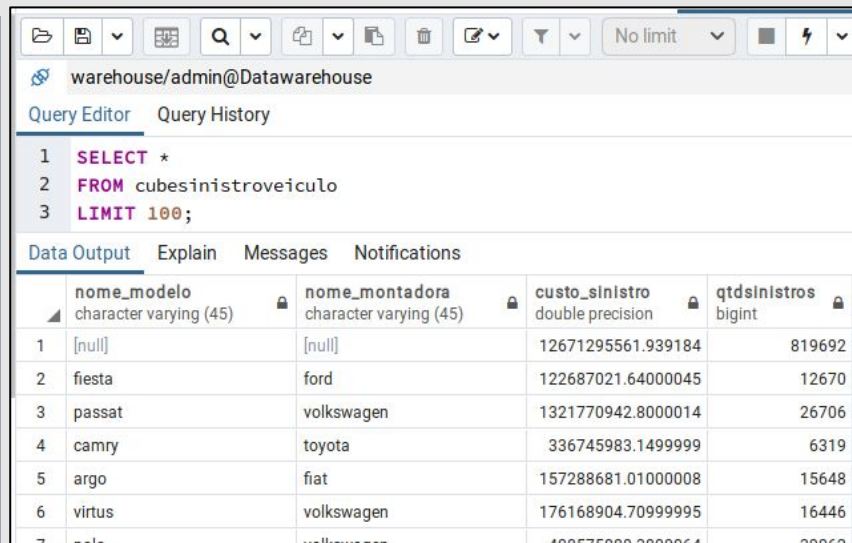


warehouse/admin@Datawarehouse				
Query Editor Query History				
<pre>1 SELECT * 2 FROM cubetotalvenda 3 LIMIT 100;</pre>				
Data Output Explain Messages Notifications				
	ano double precision	mes double precision	dia double precision	total double precision
1	[null]	[null]	[null]	119564878383
2	2013	11	4	8294699
3	2015	3	26	11013170
4	2013	9	16	210036244
5	2013	8	29	23004466
6	2018	1	10	37759759
7	2014	6	10	23372919

**CUBE (ano\_compra,  
mes\_compra,  
dia\_compra)**

# Cubos de Dados - *cubesinistroveiculo*

```
CREATE VIEW cubesinistroveiculo AS
SELECT
  nome_modelo,
  nome_montadora,
  SUM(valor) AS custo_sinistro,
  COUNT(*) AS qtdSinistros
FROM sinistro
NATURAL JOIN veiculo
GROUP BY
  CUBE(
    nome_modelo,
    nome_montadora
  );
```



The screenshot shows a SQL query editor interface. At the top, there's a toolbar with icons for file operations, search, and execution. Below the toolbar, the connection is set to 'warehouse/admin@Datawarehouse'. The 'Query Editor' tab is active, displaying a SQL query. Below the query, the 'Data Output' tab is active, showing the results of the query in a table format. The table has four columns: 'nome\_modelo', 'nome\_montadora', 'custo\_sinistro', and 'qtdsinistros'. The data is grouped by 'nome\_modelo' and 'nome\_montadora'.

	nome_modelo character varying (45)	nome_montadora character varying (45)	custo_sinistro double precision	qtdsinistros bigint
1	[null]	[null]	12671295561.939184	819692
2	fiesta	ford	122687021.64000045	12670
3	passat	volkswagen	1321770942.8000014	26706
4	camry	toyota	336745983.1499999	6319
5	argo	fiat	157288681.01000008	15648
6	virtus	volkswagen	176168904.70999995	16446
7	polo	volkswagen	108575088.28000054	20062

**CUBE (nome\_montadora,  
nome\_modelo)**

# Cubos de Dados - *cubesinistroveiculo*

## Summary

67

Total Cubes Inis Trove Icu Lo

## How these Cubes Inis Trove Icu Lo are distributed

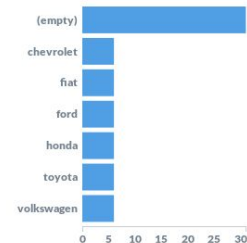
Cubes Inis Trove Icu Lo by Qt Dsi Nis Tros



Cubes Inis Trove Icu Lo by Cu Sto Si Nis Tro



Cubes Inis Trove Icu Lo per Nome Mon Tad O...



warehouse/admin@Datawarehouse

Query Editor Query History

```
1 SELECT *
2 FROM cubesinistroveiculo
3 LIMIT 100;
```

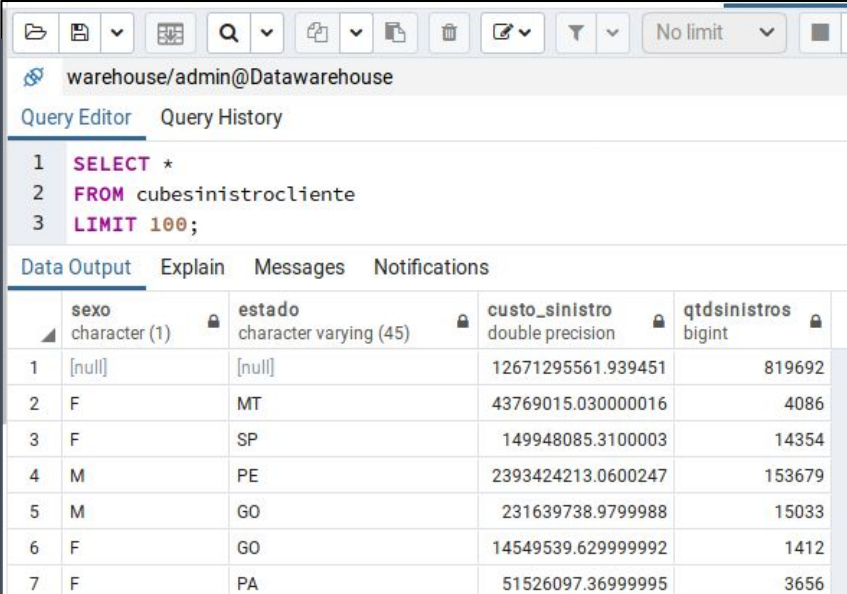
Data Output Explain Messages Notifications

	nome_modelo character varying (45)	nome_montadora character varying (45)	custo_sinistro double precision	qtddsintros bigint
1	[null]	[null]	12671295561.939184	819692
2	fiesta	ford	122687021.64000045	12670
3	passat	volkswagen	1321770942.8000014	26706
4	camry	toyota	336745983.1499999	6319
5	argo	fiat	157288681.01000008	15648
6	virtus	volkswagen	176168904.70999995	16446
7	polo	volkswagen	108575983.2000004	20062

**CUBE (nome\_montadora,  
nome\_modelo)**

# Cubos de Datos - *cubesinistrocliente*

```
CREATE VIEW cubesinistrocliente AS
SELECT
  sexo,
  estado,
  SUM(valor) AS custo_sinistro,
  COUNT(*) AS qtdSinistros
FROM sinistro
NATURAL JOIN cliente
GROUP BY
  CUBE (
    sexo,
    estado
  );
```



The screenshot shows a database query editor interface. At the top, there's a toolbar with icons for file operations, search, and execution. Below the toolbar, the user is logged in as 'warehouse/admin@Datawarehouse'. The 'Query Editor' tab is active, displaying a SQL query. The 'Data Output' tab is also visible, showing the results of the query in a table format. The table has five columns: 'sexo', 'estado', 'custo\_sinistro', and 'qtdsinistros'. The first row shows a null value for 'sexo' and 'estado', and the subsequent rows show data for different combinations of 'sexo' and 'estado'.

	sexo character (1)	estado character varying (45)	custo_sinistro double precision	qtdsinistros bigint
1	[null]	[null]	12671295561.939451	819692
2	F	MT	43769015.030000016	4086
3	F	SP	149948085.31000003	14354
4	M	PE	2393424213.0600247	153679
5	M	GO	231639738.9799988	15033
6	F	GO	14549539.629999992	1412
7	F	PA	51526097.36999995	3656

**CUBE (sexo,  
estado)**

# Cubos de Datos - *cubesinistrocliente*

33

Total Cubesinistrocliente

## How these Cubesinistrocliente are distributed

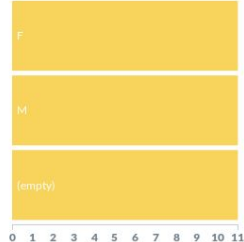
Cubesinistrocliente by Qt Dsi Nis Tros



Cubesinistrocliente by Cu Sto Si Nis Tro



Cubesinistrocliente per Sex O



warehouse/admin@Datawarehouse				
Query Editor Query History				
<pre>1 SELECT * 2 FROM cubesinistrocliente 3 LIMIT 100;</pre>				
Data Output Explain Messages Notifications				
	sexo character (1)	estado character varying (45)	custo_sinistro double precision	qtdsinistros bigint
1	[null]	[null]	12671295561.939451	819692
2	F	MT	43769015.030000016	4086
3	F	SP	149948085.31000003	14354
4	M	PE	2393424213.0600247	153679
5	M	GO	231639738.9799988	15033
6	F	GO	14549539.629999992	1412
7	F	PA	51526097.36999995	3656

**CUBE (sexo,  
estado)**

# Inferências

- Resultados
  - 1, MG, 'belo horizonte', 21, ka, 2017, ?
  - 2, MG, 'belo horizonte', 21, gol, 2014, ?
  - 3, SP ,campinas, 38, hrv, 2011, ?

=== Predictions on user test set ===

inst#	actual	predicted	error	prediction
1	1:?	1:true	0.769	
2	1:?	2:false	0.581	
3	1:?	2:false	0.534	

=== Predictions on user test set ===

inst#	actual	predicted	error	prediction
1	1:?	1:true	0.667	
2	1:?	1:true	0.694	
3	1:?	2:false	0.535	




# Possíveis perguntas

- “Como se dá a distribuição de sinistros por estado?”

```
SELECT estado, ROUND(CAST(custo_sinistro AS NUMERIC), 2)  
  AS custo_sinistro, qtdsinistros  
FROM public.cubesinistrocliente  
WHERE estado IS NOT NULL AND sexo IS NULL  
ORDER BY custo_sinistro DESC;
```

# Possíveis perguntas

- “Como se dá a distribuição de sinistros por estado?”

	Data Output	Explain	Messages	Notifications
	 estado character varying (45)		custo_sinistro numeric	 qtdsinistros bigint
1	SC		2735947152.37	179245
2	SP		2573104924.65	169563
3	PE		2543362800.71	167729
4	MG		1053061321.32	69206
5	RS		1018191534.39	66352
6	PA		829632497.54	44158
7	MT		744199036.42	49097
8	BA		680840876.63	44773
9	AM		246766139.30	13124
10	GO		246189278.61	16445