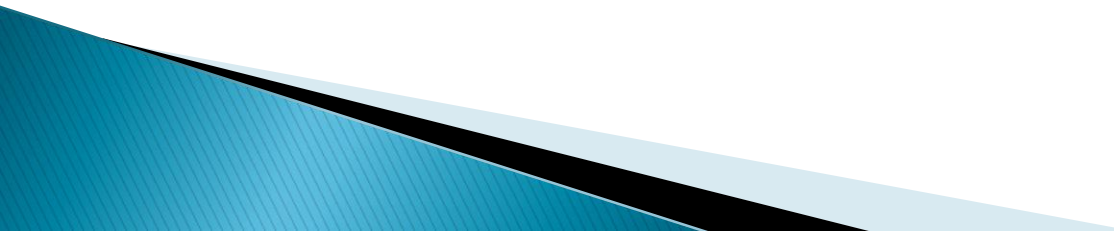


# Tragédia de Brumadinho

Análise dos dados e previsões de categorias de risco, danos associados e classe

Daniel Rodrigues de Rezende

# Contexto

- ▶ A cidade de Brumadinho foi vítima da tragédia com o rompimento da barragem de minério recentemente.
  - ▶ A ideia é avaliar os riscos baseado em alguns dados disponíveis em sites públicos, como IBGE e ANM.
  - ▶ Nos slides a seguir há informações de como foram coletado os dados, como estes foram tratados antes de serem treinados pelo algoritmo, o treinamento e as conclusões.
- 

# Storytelling – Análise dos dados – Target (y)

- ▶ Como a ideia avaliar os riscos, o ponto de observação (TARGET) são as colunas 'CATEGORIA\_DE\_RISCO', 'DANO\_POTENCIAL\_ASSOCIADO' e 'CLASSE', mas as mesmas estão incompletas, com apenas 390 linhas preenchidas, sendo que o total de linhas com as informações são 714.
- ▶ O objetivo aqui neste caso é fazer a predição do restante dos dados destas colunas, que serão o nosso TARGET (y). As 390 linhas já preenchidas de todas as colunas serão usadas para treinar o algoritmo para que este faça a predição do restante das 324 linhas faltantes.
- ▶ Target: No caso, iremos fazer as predições dos 3 itens abaixo (avaliar)
  - ▶ – CATEGORIA\_DE\_RISCO
  - ▶ – 'DANO\_POTENCIAL\_ASSOCIADO':
  - ▶ – 'CLASSE':

Dados com NaN:

CATEGORIA_DE_RISCO	324
DANO_POTENCIAL_ASSOCIADO	324
CLASSE	324
dtype:	int64

# Storytelling – Análise dos dados – Features (X)

- ▶ A seguir foi escolhido como FEATURES as colunas 'MINERIO\_PRINCIPAL', 'ALTURA\_ATUAL\_metros', 'VOLUME\_ATUAL\_m3', pelo fato de conter dados que provoquem algum efeito sensato da predição dos dados restantes.
- ▶ Feature:
  - ▶ – MINERIO\_PRINCIPAL: o tipo de minério pode afetar o peso da barragem, pra um mesmo volume ocupado, o peso varia de acordo com o tipo de minério sedentado.
  - ▶ – 'ALTURA\_ATUAL\_metros': a altura da barragem pode influenciar nos dados, visto que quanto maior estiver, maior o risco de rompimento,
  - ▶ – 'VOLUME\_ATUAL\_m3': volume da barragem pode influenciar nos dados, visto que quanto maior estiver, maior o risco de rompimento,

# Storytelling – Tratamento dos dados

- ▶ Para tal, deve-se primeiro analisar os dados contidos nestas colunas em busca de inconsistências e dados faltantes.
- ▶ A ordem da análise dos dados foi a seguinte:
- ▶ 1) Criar 2 bancos de dados, um com dados de treino (train) e outro com os dados a serem dados a serem previstos (test).

```
train = data[0:390]
```

```
test = data[390:714].reset_index()
```

# Storytelling – Tratamento dos dados

- ▶ 2) Checar o tipo de dados das features e dos targets:

## Features (train)

MINERIO_PRINCIPAL	390 non-null object
ALTURA_ATUAL_metros	390 non-null object
VOLUME_ATUAL_m3	390 non-null object

## Target (train)

CATEGORIA_DE_RISCO	390 non-null object
DANO_POTENCIAL_ASSOCIADO	390 non-null object
CLASSE	390 non-null object

## Features (test)

MINERIO_PRINCIPAL	324 non-null object
ALTURA_ATUAL_metros	324 non-null object
VOLUME_ATUAL_m3	324 non-null object

## Target (test)

CATEGORIA_DE_RISCO	0 non-null object
DANO_POTENCIAL_ASSOCIADO	0 non-null object
CLASSE	0 non-null object

Neste caso, faz-se necessário convertê-los de string para float64 para as predições. Nos próximos slides será mostrado que foi feita a conversão

# Storytelling – Tratamento dos dados

- ▶ 2) Se existem dados inconsistentes ou faltantes nas seguintes colunas:

```
A feature MINERIO_PRINCIPAL no df train possui " 0 " dados inválidos
A feature MINERIO_PRINCIPAL no df test possui " 0 " dados inválidos

A feature ALTURA_ATUAL_metros no df train possui " 0 " dados inválidos
A feature ALTURA_ATUAL_metros no df test possui " 17 " dados inválidos

A feature VOLUME_ATUAL_m3 no df train possui " 0 " dados inválidos
A feature VOLUME_ATUAL_m3 no df test possui " 7 " dados inválidos

0 target CATEGORIA_DE_RISCO no df train possui " 0 " dados inválidos

0 target DANO_POTENCIAL_ASSOCIADO no df train possui " 0 " dados inválidos

0 target CLASSE no df train possui " 0 " dados inválidos
```

# Storytelling – Tratamento dos dados

- ▶ 3) Transformar os dados object 'ALTURA\_ATUAL\_metros' e 'VOLUME\_ATUAL\_m3' em float64, pois estão como string e devem trabalhar como float64;
- ▶ – Os dados de treino (**train**), conforme visto no slide anterior, não possui dados inválidos, portanto já está apto a ser convertido para float64.
- ▶ – Os dados de **test** estão com dados inválidos, precisam ser tratados para depois serem convertidos para float64, sendo assim foi feito.

Os dados inválidos citados eram dados com o string '–' no lugar de um número, sendo assim foi retirado esta string que no lugar ficou preenchido com 'Nan', sendo assim no lugar deste, foi feito o uso da função '**Imputation**' que preenche o dado com um valor médio dos valores da coluna. Próximo slide tem-se uma nova verificação dos dados



# Storytelling – Tratamento dos dados

- ▶ 2) Se existem dados inconsistentes ou faltantes nas seguintes colunas:

```
A feature MINERIO_PRINCIPAL no df train possui " 0 " dados inválidos
A feature MINERIO_PRINCIPAL no df test possui " 0 " dados inválidos

A feature ALTURA_ATUAL_metros no df train possui " 0 " dados inválidos
A feature ALTURA_ATUAL_metros no df test possui " 0 " dados inválidos

A feature VOLUME_ATUAL_m3 no df train possui " 0 " dados inválidos
A feature VOLUME_ATUAL_m3 no df test possui " 0 " dados inválidos

0 target CATEGORIA_DE_RISCO no df train possui " 0 " dados inválidos

0 target DANO_POTENCIAL_ASSOCIADO no df train possui " 0 " dados inválidos

0 target CLASSE no df train possui " 0 " dados inválidos
```

# Storytelling – Tratamento dos dados – One hot encoding

- ▶ Fazer one hot encoding nos dados 'DANO\_POTENCIAL\_ASSOCIADO' , 'CLASSE ' e 'MINERIO\_PRINCIPAL' para que o algoritmo possa ser treinado. Neste caso transforma valores (e.g. Sim, não em 1, 0).
- ▶ Exemplo prático. A coluna CATEGORIA\_DE\_RISCO de ambos dados train e test, que antes era apenas uma coluna com dados Alta, Média e Baixa, foi criado um novo DF com 3 colunas com os mesmos Alta, Média e Baixa

0	Baixa
1	Baixa
2	Baixa
3	Baixa
4	Baixa
5	Baixa
6	Baixa
7	Baixa
8	Baixa
9	Média
10	Baixa
11	Baixa
12	Baixa
13	Média
14	Média
15	Baixa
...	...



	Alta	Baixa	Média
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0
6	0	1	0
7	0	1	0
8	0	1	0
9	0	0	1
10	0	1	0
11	0	1	0
12	0	1	0
..	...	...	...

# Storytelling – Target and Features creation – Permutation Importance

- ▶ A coluna 'MINERIO\_PRINCIPAL' possui muitas colunas, sendo que muitas delas não afetam em nada o modelo de treinamento, sendo assim eu usei a função 'PermutationImportance()' para classificar as features mais importantes de acordo com a função para melhorar a eficiência das features.

Weight	Feature
0.3495 ± 0.1688	ALTURA_ATUAL_metros
0.2655 ± 0.0397	VOLUME_ATUAL_m3
0.0508 ± 0.0386	Argila
0.0220 ± 0.0051	Carvão Mineral Camada Bonito
0.0182 ± 0.0399	Minério de Ouro Primário
0.0175 ± 0.0232	Aluvião Estanífero
0.0158 ± 0.0105	Rocha Aurífera
0.0096 ± 0.0093	Bauxita Grau Não Metalúrgico
0.0070 ± 0.0039	Carvão Mineral
0.0038 ± 0.0036	Cascalho
0.0011 ± 0.0025	Minério de Vanádio
0.0006 ± 0.0000	Minério de Nióbio
0 ± 0.0000	Sedimentos
-0.0053 ± 0.0199	Areia

- ▶ Também foram retirados minérios não presentes nos dados a serem previstos.

# Storytelling – Target and Features creation

- ▶ O motivo das escolhas foram citadas no começo desta. Usou-se 25% para a criação dos dados de teste ( $X_{test}$ ,  $y_{test}$ ) e 75% para os de treino ( $X_{train}$ ,  $y_{train}$ )
- ▶ Features (X):
  - ▶ 'ALTURA\_ATUAL\_metros'
  - ▶ 'VOLUME\_ATUAL\_m3'
  - ▶ train\_MINERIO\_PRINCIPAL(['Aluvião Estanífero', 'Argila', 'Minério de Ouro Primário', 'Rocha Aurífera', 'Bauxita Grau Não Metalúrgico', 'Cascalho'])
- ▶ Targets (y):
  - ▶ train\_CATEGORIA\_DE\_RISCO,
  - ▶ train\_DANO\_POTENCIAL\_ASSOCIADO,
  - ▶ train\_CLASSE

# Storytelling – Target and Features creation

- ▶ O modelo de treinamento foi escolhido o random forest, que treinará em todos os dados de treinamento. Poderia ter escolhido o XGBoost, mas o mesmo não trabalha com mais de um target. No mesmo modelo foi usado pipeline de Imputation com Random Forest com objetivo de simplificar a construção do mesmo, a validação de modelos e a implantação de modelos. Em seguida foi feito o **fit**.

```
train_model = make_pipeline(Imputer(), RandomForestRegressor(random_state=1))  
train_model.fit(X_train, y_train)
```

# Storytelling – Target and Features creation – Validação do train data

- ▶ Predição.

```
# predict  
train_predictions = train_model.predict(X_test)
```

- ▶ Validação com cross-validation: executa processo de modelagem em diferentes subconjuntos dos dados para obter várias medidas de qualidade do modelo.

```
#mae using cross validation  
scores = cross_val_score(train_model, X, y, scoring='neg_mean_absolute_error')
```

```
[-0.20923077 -0.2172028 -0.2641958 ]  
Mean Absolute Error 0.230210
```

# Storytelling – Target and Features creation – Previsão dos dados

- ▶ Agora com o algoritmo treinado, será feita a geração dos novos dados faltantes.

```
X_test = pd.concat([test_MINERIO_PRINCIPAL[features_minerio], test[features]], axis=1, join_axes=[test_MINERIO_PRINCIPAL.index])
X_test = my_imputer.fit_transform(X_test)

# make predictions which we will submit.
test_preds = train_model.predict(X_test)
```

- ▶ O novos dados faltantes do target y para a base de dados **test** foram gerados com sucesso na variável **test\_preds**, porém estes estão em dados binários, será feita a conversão para melhor entendimento.

# Storytelling – Invertendo One hot encoding

- ▶ Agora com os dados já previstos, é preciso pegar os dados previstos, que estão em modelo **one hot encoding**, ou seja, em dados binários, e transforma-los de volta da maneira inicial, para melhor entendimento.

	Alta	Baixa	Média
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
5	0	1	0
6	0	1	0
7	0	1	0
8	0	1	0
9	0	0	1
10	0	1	0
11	0	1	0
12	0	1	0
..	...	...	...



0	Baixa
1	Baixa
2	Baixa
3	Baixa
4	Baixa
5	Baixa
6	Baixa
7	Baixa
8	Baixa
9	Média
10	Baixa
11	Baixa
12	Baixa
13	Média
14	Média
15	Baixa
...	...



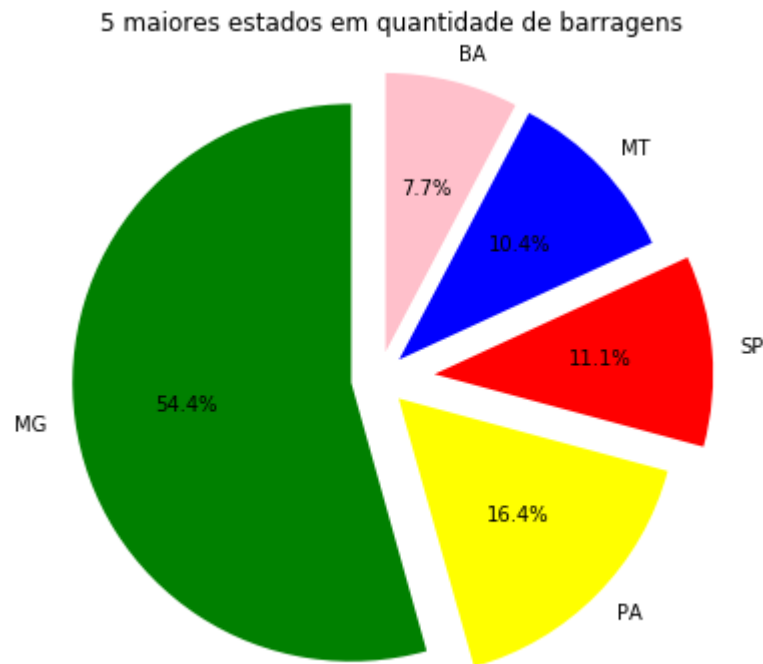
# Storytelling – Output file

- ▶ Finalmente, a fim de se gerar um output completo, com todos os dados, tanto os que já estavam preenchidos quanto com os novos dados previstos, foi feito um merge entre os dados treino (train) e teste (test), separados inicialmente.
- ▶ Foi gerado um **new\_data** e o mesmo gerado um novo arquivo CSV.

```
output = pd.DataFrame({'NOME_BARRAGEM_MINERACAO' : new_data.NOME_BARRAGEM_MINERACAO,  
                        'NOME_DO_EMPREENDEDOR' : new_data.NOME_DO_EMPREENDEDOR,  
                        'CPF_CNPJ' : new_data.CPF_CNPJ,  
                        'POSICIONAMENTO' : new_data.POSICIONAMENTO,  
                        'UF' : new_data.UF,  
                        'MUNICIPIO' : new_data.MUNICIPIO,  
                        'MINERIO_PRINCIPAL' : new_data.MINERIO_PRINCIPAL,  
                        'ALTURA_ATUAL_metros' : new_data.ALTURA_ATUAL_metros,  
                        'VOLUME_ATUAL_m3' : new_data.VOLUME_ATUAL_m3,  
                        'CATEGORIA_DE_RISCO' : new_data.CATEGORIA_DE_RISCO,  
                        'DANO_POTENCIAL_ASSOCIADO' : new_data.DANO_POTENCIAL_ASSOCIADO,  
                        'CLASSE' : new_data.CLASSE,  
                        'INSERIDA_NA_PNSB' : new_data.INSERIDA_NA_PNSB,  
                        'LATITUDE' : new_data.LATITUDE,  
                        'LONGITUDE' : new_data.LONGITUDE  
                        })  
output.to_csv('submission.csv', index=False)
```

# Storytelling – Conclusões

- ▶ Minas Gerais é disparado o estado com mais barragens construídas no Brasil. Sendo assim, somado a tragédia de Brumadinho, as análises foram feitas em cima no geral do país e concentradas no estado de MG.
- ▶ Listo abaixo os estados que concentram a maior quantidade de barragens no país.

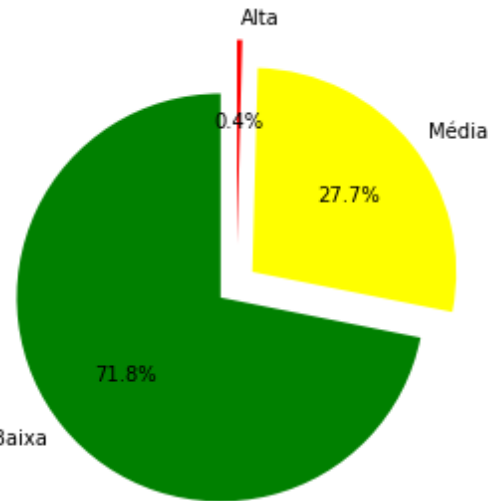


MG	324
PA	98
SP	66
MT	62
BA	46

# Storytelling – Conclusões Nível Brasil

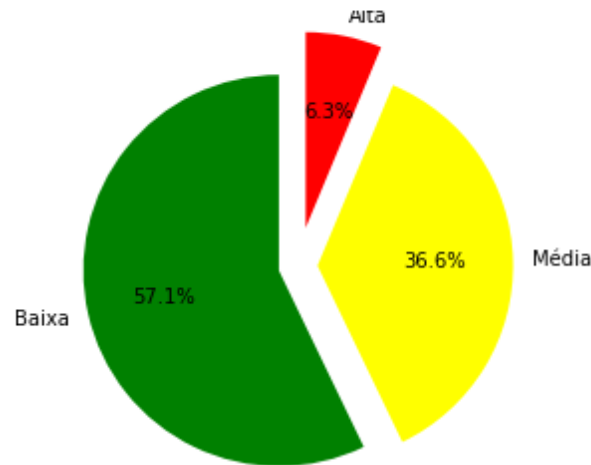
- ▶ A seguir um panorama sobre as barragens nas categorias de RISCO, DANO ASSOCIADO e CLASSE:

CATEGORIA DE RISCO



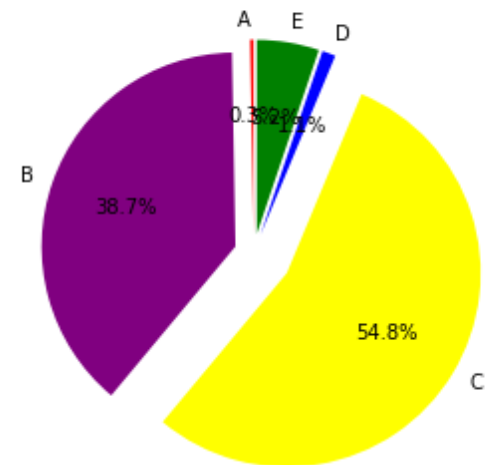
Baixa	513
Média	198
Alta	3

DANO POTENCIAL ASSOCIADO



Média	408
Alta	261
Baixa	45

CLASSE DA BARRAGEM



C	391
B	276
E	37
D	8
A	2

# Storytelling – Conclusões Nível Minas Gerais

- ▶ A seguir um panorama sobre os 5 municípios com maior quantidade de barragens nas categorias de RISCO:

## Alto Risco

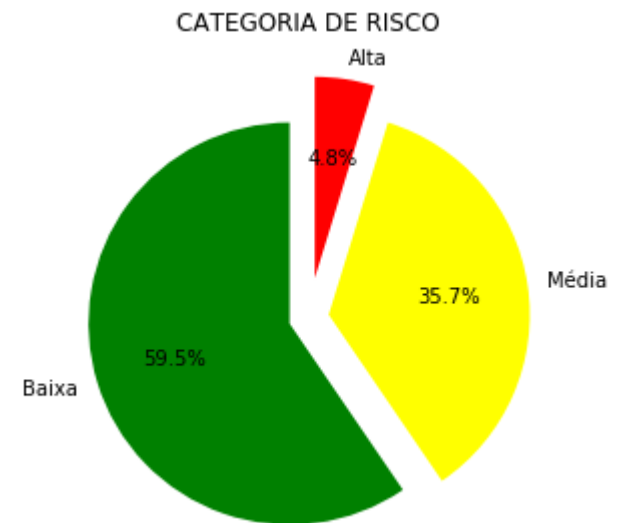
RIO ACIMA 2

## Médio Risco

ITABIRITO	15
BRUMADINHO	8
UBERABA	8
ITATIAIUÇU	5
OURO PRETO	4

## Baixo Risco

NOVA LIMA	25
OURO PRETO	20
BRUMADINHO	18
ITATIAIUÇU	17
ITABIRA	17



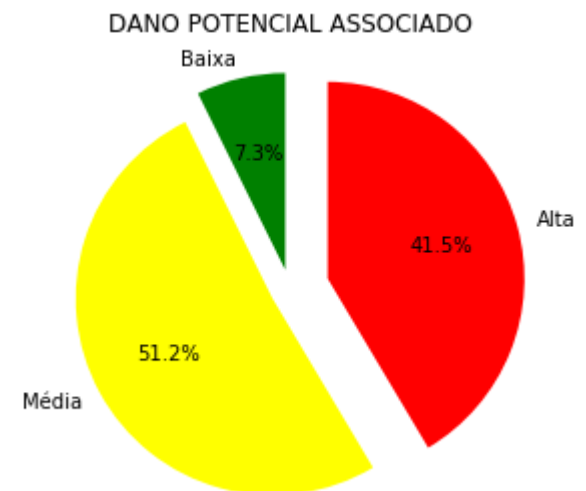
# Storytelling – Conclusões Nível Minas Gerais

- ▶ A seguir um panorama sobre os 5 municípios com maior quantidade de barragens nas categorias de DANO ASSOCIADO:

Alto Dano	
NOVA LIMA	17
OURO PRETO	15
ITABIRA	13
BRUMADINHO	8
ITATIAIUÇU	7

Médio Dano	
ITABIRITO	21
BRUMADINHO	16
ITATIAIUÇU	14
MATEUS LEME	11
OURO PRETO	8

Baixo Dano	
NOVA LIMA	3
CONGONHAS	3
BRUMADINHO	2
LAGAMAR	2
ITABIRA	2



# Storytelling – Conclusões Nível Minas Gerais

- ▶ A seguir um panorama sobre os 5 municípios com maior quantidade de barragens nas categorias de CLASSE:

Classe A		
RIO ACIMA	2	

Classe B		
NOVA LIMA	17	
OURO PRETO	15	
ITABIRA	13	
ITATIAIUÇU	9	
CONGONHAS	8	

Classe C		
ITABIRITO	21	
BRUMADINHO	16	
ITATIAIUÇU	12	
MATEUS LEME	10	
OURO PRETO	8	

Classe D		
0		

Classe E		
NOVA LIMA	3	
CONGONHAS	3	
BRUMADINHO	2	
LAGAMAR	2	
ITABIRA	2	

