ZYFRA - MINERAÇÃO DE OURO

A EMPRESA ZYFRA DESENVOLVE SOLUÇÕES DE EFICIÊNCIA PARA A INDÚSTRIA PESADA.

O OBJETIVO É CRIAR UM MODELO CAPAZ DE PREVER A QUANTIDADE DE OURO PURO EXTRAÍDO DO MINÉRIO DE OURO. ELE AJUDARÁ A OTIMIZAR A PRODUÇÃO E ELIMINAR PARÂMETROS NÃO RENTÁVEIS.

DETRALHE DO PROCESSO

1. Flotação A mistura de minério de ouro é alimentada nos bancos de flotação para obter concentrado de Au bruto e outros restos de minérios brutos (resíduos de produtos com baixa concentração de metais valiosos). A estabilidade deste processo é afetada pelo estado físico-químico volátil e não ótimo da polpa de flotação (uma mistura de partículas sólidas e líquidas).

2. Purificação O concentrado bruto passa por duas etapas de purificação. Após a purificação, temos o concentrado final e novos restos de minério.

DESCRIÇÃO DOS DADOS

Processo tecnológico

Rougher feed — matéria-prima
Rougher additions (ou aditivos reagentes) — reagentes de flotação: Xantato, Sulfeto, Depressor
Xanthate — promotor ou ativador da flotação
Sulphate — sulfeto de sódio para este processo específico
Depressant — silicato de sódio
Rougher process — flotação
Rougher tails — resíduos do produto
Float banks — unidade de flotação
Cleaner process — purificação
Rougher Au — concentrado de ouro bruto
Final Au — concentrado de ouro final

Processo tecnológico

Rougher feed — matéria-prima
Rougher additions (ou aditivos reagentes) — reagentes de flotação: Xantato, Sulfeto, Depressor
Xanthate — promotor ou ativador da flotação
Sulphate — sulfeto de sódio para este processo específico
Depressant — silicato de sódio
Rougher process — flotação
Rougher tails — resíduos do produto
Float banks — unidade de flotação
Cleaner process — purificação
Rougher Au — concentrado de ouro bruto
Final Au — concentrado de ouro final

IMPORTAÇÃO DE BIBLIOTECAS

In [1]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

import warnings

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold
```

PREPARAÇÃO DOS DADOS

carregamento

In [2]:

```
df_train= pd.read_csv('/datasets/gold_recovery_train.csv')
df_test= pd.read_csv('/datasets/gold_recovery_test.csv')
df_full= pd.read_csv('/datasets/gold_recovery_full.csv')
```

DF_TRAIN

In [17]:

```
df_train.shape
```

Out[17]:

```
(16860, 87)
```

In [18]:

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16860 entries, 0 to 16859
Data columns (total 87 columns):
 #   Column                                              Non-Null Count  D
type
---  ------                                              --------------  -
----
 0   date                                                16860 non-null  o
bject
 1   final.output.concentrate_ag                        16788 non-null  f
loat64
 2   final.output.concentrate_pb                        16788 non-null  f
loat64
 3   final.output.concentrate_sol                       16490 non-null  f
loat64
 4   final.output.concentrate_au                        16789 non-null  f
loat64
 5   final.output.recovery                              15339 non-null  f
loat64
 6   final.output.tail_ag                               16794 non-null  f
loat64
 7   final.output.tail_pb                               16677 non-null  f
loat64
 8   final.output.tail_sol                              16715 non-null  f
loat64
 9   final.output.tail_au                               16794 non-null  f
loat64
 10  primary_cleaner.input.sulfate                      15553 non-null  f
loat64
 11  primary_cleaner.input.depressant                   15598 non-null  f
loat64
 12  primary_cleaner.input.feed_size                    16860 non-null  f
loat64
 13  primary_cleaner.input.xanthate                     15875 non-null  f
loat64
 14  primary_cleaner.output.concentrate_ag              16778 non-null  f
loat64
 15  primary_cleaner.output.concentrate_pb              16502 non-null  f
loat64
 16  primary_cleaner.output.concentrate_sol             16224 non-null  f
loat64
 17  primary_cleaner.output.concentrate_au              16778 non-null  f
loat64
 18  primary_cleaner.output.tail_ag                     16777 non-null  f
loat64
 19  primary_cleaner.output.tail_pb                     16761 non-null  f
loat64
 20  primary_cleaner.output.tail_sol                    16579 non-null  f
loat64
 21  primary_cleaner.output.tail_au                     16777 non-null  f
loat64
 22  primary_cleaner.state.floatbank8_a_air             16820 non-null  f
loat64
 23  primary_cleaner.state.floatbank8_a_level           16827 non-null  f
loat64
 24  primary_cleaner.state.floatbank8_b_air             16820 non-null  f
loat64
 25  primary_cleaner.state.floatbank8_b_level           16833 non-null  f
loat64
 26  primary_cleaner.state.floatbank8_c_air             16822 non-null  f
loat64
```

```
 27  primary_cleaner.state.floatbank8_c_level            16833 non-null  f
loat64
 28  primary_cleaner.state.floatbank8_d_air              16821 non-null  f
loat64
 29  primary_cleaner.state.floatbank8_d_level            16833 non-null  f
loat64
 30  rougher.calculation.sulfate_to_au_concentrate       16833 non-null  f
loat64
 31  rougher.calculation.floatbank10_sulfate_to_au_feed  16833 non-null  f
loat64
 32  rougher.calculation.floatbank11_sulfate_to_au_feed  16833 non-null  f
loat64
 33  rougher.calculation.au_pb_ratio                     15618 non-null  f
loat64
 34  rougher.input.feed_ag                               16778 non-null  f
loat64
 35  rougher.input.feed_pb                               16632 non-null  f
loat64
 36  rougher.input.feed_rate                             16347 non-null  f
loat64
 37  rougher.input.feed_size                             16443 non-null  f
loat64
 38  rougher.input.feed_sol                              16568 non-null  f
loat64
 39  rougher.input.feed_au                               16777 non-null  f
loat64
 40  rougher.input.floatbank10_sulfate                   15816 non-null  f
loat64
 41  rougher.input.floatbank10_xanthate                  16514 non-null  f
loat64
 42  rougher.input.floatbank11_sulfate                   16237 non-null  f
loat64
 43  rougher.input.floatbank11_xanthate                  14956 non-null  f
loat64
 44  rougher.output.concentrate_ag                       16778 non-null  f
loat64
 45  rougher.output.concentrate_pb                       16778 non-null  f
loat64
 46  rougher.output.concentrate_sol                      16698 non-null  f
loat64
 47  rougher.output.concentrate_au                       16778 non-null  f
loat64
 48  rougher.output.recovery                             14287 non-null  f
loat64
 49  rougher.output.tail_ag                              14610 non-null  f
loat64
 50  rougher.output.tail_pb                              16778 non-null  f
loat64
 51  rougher.output.tail_sol                             14611 non-null  f
loat64
 52  rougher.output.tail_au                              14611 non-null  f
loat64
 53  rougher.state.floatbank10_a_air                     16807 non-null  f
loat64
 54  rougher.state.floatbank10_a_level                   16807 non-null  f
loat64
 55  rougher.state.floatbank10_b_air                     16807 non-null  f
loat64
 56  rougher.state.floatbank10_b_level                   16807 non-null  f
loat64
 57  rougher.state.floatbank10_c_air                     16807 non-null  f
```

```
loat64
 58  rougher.state.floatbank10_c_level                  16814 non-null  f
loat64
 59  rougher.state.floatbank10_d_air                    16802 non-null  f
loat64
 60  rougher.state.floatbank10_d_level                  16809 non-null  f
loat64
 61  rougher.state.floatbank10_e_air                    16257 non-null  f
loat64
 62  rougher.state.floatbank10_e_level                  16809 non-null  f
loat64
 63  rougher.state.floatbank10_f_air                    16802 non-null  f
loat64
 64  rougher.state.floatbank10_f_level                  16802 non-null  f
loat64
 65  secondary_cleaner.output.tail_ag                   16776 non-null  f
loat64
 66  secondary_cleaner.output.tail_pb                   16764 non-null  f
loat64
 67  secondary_cleaner.output.tail_sol                  14874 non-null  f
loat64
 68  secondary_cleaner.output.tail_au                   16778 non-null  f
loat64
 69  secondary_cleaner.state.floatbank2_a_air           16497 non-null  f
loat64
 70  secondary_cleaner.state.floatbank2_a_level         16751 non-null  f
loat64
 71  secondary_cleaner.state.floatbank2_b_air           16705 non-null  f
loat64
 72  secondary_cleaner.state.floatbank2_b_level         16748 non-null  f
loat64
 73  secondary_cleaner.state.floatbank3_a_air           16763 non-null  f
loat64
 74  secondary_cleaner.state.floatbank3_a_level         16747 non-null  f
loat64
 75  secondary_cleaner.state.floatbank3_b_air           16752 non-null  f
loat64
 76  secondary_cleaner.state.floatbank3_b_level         16750 non-null  f
loat64
 77  secondary_cleaner.state.floatbank4_a_air           16731 non-null  f
loat64
 78  secondary_cleaner.state.floatbank4_a_level         16747 non-null  f
loat64
 79  secondary_cleaner.state.floatbank4_b_air           16768 non-null  f
loat64
 80  secondary_cleaner.state.floatbank4_b_level         16767 non-null  f
loat64
 81  secondary_cleaner.state.floatbank5_a_air           16775 non-null  f
loat64
 82  secondary_cleaner.state.floatbank5_a_level         16775 non-null  f
loat64
 83  secondary_cleaner.state.floatbank5_b_air           16775 non-null  f
loat64
 84  secondary_cleaner.state.floatbank5_b_level         16776 non-null  f
loat64
 85  secondary_cleaner.state.floatbank6_a_air           16757 non-null  f
loat64
 86  secondary_cleaner.state.floatbank6_a_level         16775 non-null  f
loat64
dtypes: float64(86), object(1)
memory usage: 11.2+ MB
```

In [19]:

```
df_train.describe()
```

Out[19]:

	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_sol	fina
count	16788.000000	16788.000000	16490.000000	
mean	4.716907	9.113559	8.301123	
std	2.096718	3.389495	3.825760	
min	0.000000	0.000000	0.000000	
25%	3.971262	8.825748	6.939185	
50%	4.869346	10.065316	8.557228	
75%	5.821176	11.054809	10.289741	
max	16.001945	17.031899	18.124851	

8 rows × 86 columns

In [68]:

```
#corrigir a tipagem da coluna date:

df_train['date']= pd.to_datetime(df_train['date'], format= '%Y-%m-%d %H:%M:%S')
```

In [69]:

```
df_train.head()
```

Out[69]:

	date	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_so
0	2016-01-15 00:00:00	6.055403	9.889648	5.507324
1	2016-01-15 01:00:00	6.029369	9.968944	5.257781
2	2016-01-15 02:00:00	6.055926	10.213995	5.383759
3	2016-01-15 03:00:00	6.047977	9.977019	4.858634
4	2016-01-15 04:00:00	6.148599	10.142511	4.939416

5 rows × 87 columns

DF_TEST

In [70]:

```
df_test.shape
```

Out[70]:

```
(5278, 53)
```

In [71]:

```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5278 entries, 0 to 5855
Data columns (total 53 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   date                                        5278 non-null   object 
 1   primary_cleaner.input.sulfate               5276 non-null   float64
 2   primary_cleaner.input.depressant            5274 non-null   float64
 3   primary_cleaner.input.feed_size             5278 non-null   float64
 4   primary_cleaner.input.xanthate              5276 non-null   float64
 5   primary_cleaner.state.floatbank8_a_air      5278 non-null   float64
 6   primary_cleaner.state.floatbank8_a_level    5278 non-null   float64
 7   primary_cleaner.state.floatbank8_b_air      5278 non-null   float64
 8   primary_cleaner.state.floatbank8_b_level    5278 non-null   float64
 9   primary_cleaner.state.floatbank8_c_air      5278 non-null   float64
 10  primary_cleaner.state.floatbank8_c_level    5278 non-null   float64
 11  primary_cleaner.state.floatbank8_d_air      5278 non-null   float64
 12  primary_cleaner.state.floatbank8_d_level    5278 non-null   float64
 13  rougher.input.feed_ag                       5278 non-null   float64
 14  rougher.input.feed_pb                       5278 non-null   float64
 15  rougher.input.feed_rate                     5271 non-null   float64
 16  rougher.input.feed_size                     5277 non-null   float64
 17  rougher.input.feed_sol                      5254 non-null   float64
 18  rougher.input.feed_au                       5278 non-null   float64
 19  rougher.input.floatbank10_sulfate           5271 non-null   float64
 20  rougher.input.floatbank10_xanthate          5277 non-null   float64
 21  rougher.input.floatbank11_sulfate           5274 non-null   float64
 22  rougher.input.floatbank11_xanthate          5249 non-null   float64
 23  rougher.state.floatbank10_a_air             5278 non-null   float64
 24  rougher.state.floatbank10_a_level           5278 non-null   float64
 25  rougher.state.floatbank10_b_air             5278 non-null   float64
 26  rougher.state.floatbank10_b_level           5278 non-null   float64
 27  rougher.state.floatbank10_c_air             5278 non-null   float64
 28  rougher.state.floatbank10_c_level           5278 non-null   float64
 29  rougher.state.floatbank10_d_air             5278 non-null   float64
 30  rougher.state.floatbank10_d_level           5278 non-null   float64
 31  rougher.state.floatbank10_e_air             5278 non-null   float64
 32  rougher.state.floatbank10_e_level           5278 non-null   float64
 33  rougher.state.floatbank10_f_air             5278 non-null   float64
 34  rougher.state.floatbank10_f_level           5278 non-null   float64
 35  secondary_cleaner.state.floatbank2_a_air    5275 non-null   float64
 36  secondary_cleaner.state.floatbank2_a_level  5278 non-null   float64
 37  secondary_cleaner.state.floatbank2_b_air    5277 non-null   float64
 38  secondary_cleaner.state.floatbank2_b_level  5278 non-null   float64
 39  secondary_cleaner.state.floatbank3_a_air    5270 non-null   float64
 40  secondary_cleaner.state.floatbank3_a_level  5278 non-null   float64
 41  secondary_cleaner.state.floatbank3_b_air    5278 non-null   float64
 42  secondary_cleaner.state.floatbank3_b_level  5278 non-null   float64
 43  secondary_cleaner.state.floatbank4_a_air    5278 non-null   float64
 44  secondary_cleaner.state.floatbank4_a_level  5278 non-null   float64
 45  secondary_cleaner.state.floatbank4_b_air    5278 non-null   float64
 46  secondary_cleaner.state.floatbank4_b_level  5278 non-null   float64
 47  secondary_cleaner.state.floatbank5_a_air    5278 non-null   float64
 48  secondary_cleaner.state.floatbank5_a_level  5278 non-null   float64
 49  secondary_cleaner.state.floatbank5_b_air    5278 non-null   float64
 50  secondary_cleaner.state.floatbank5_b_level  5278 non-null   float64
 51  secondary_cleaner.state.floatbank6_a_air    5278 non-null   float64
 52  secondary_cleaner.state.floatbank6_a_level  5278 non-null   float64
dtypes: float64(52), object(1)
memory usage: 2.2+ MB
```

In [23]:

```
df_test.describe()
```

Out[23]:

	primary_cleaner.input.sulfate	primary_cleaner.input.depressant	primary_cleaner.input.fee
count	5554.000000	5572.000000	5856.0
mean	170.515243	8.482873	7.2
std	49.608602	3.353105	0.6
min	0.000103	0.000031	5.6
25%	143.340022	6.411500	6.8
50%	176.103893	8.023252	7.2
75%	207.240761	10.017725	7.6
max	274.409626	40.024582	15.5

8 rows × 52 columns

In [72]:

```
# corrigindo a tipagem da coluna date:

df_test['date']= pd.to_datetime(df_train['date'], format= '%Y-%m-%d %H:%M:%S')
```

In [73]:

```
df_test.head()
```

Out[73]:

	date	primary_cleaner.input.sulfate	primary_cleaner.input.depressant	primary_cleaner.inp
0	2016-01-15 00:00:00	210.800909	14.993118	
1	2016-01-15 01:00:00	215.392455	14.987471	
2	2016-01-15 02:00:00	215.259946	12.884934	
3	2016-01-15 03:00:00	215.336236	12.006805	
4	2016-01-15 04:00:00	199.099327	10.682530	

5 rows × 53 columns

DF_FULL

In [25]:

```
df_full.shape
```

Out[25]:

```
(22716, 87)
```

In [26]:

```
df_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22716 entries, 0 to 22715
Data columns (total 87 columns):
 #   Column                                              Non-Null Count  D
type
---  ------                                              --------------  -
----
 0   date                                                22716 non-null  o
bject
 1   final.output.concentrate_ag                        22627 non-null  f
loat64
 2   final.output.concentrate_pb                        22629 non-null  f
loat64
 3   final.output.concentrate_sol                       22331 non-null  f
loat64
 4   final.output.concentrate_au                        22630 non-null  f
loat64
 5   final.output.recovery                              20753 non-null  f
loat64
 6   final.output.tail_ag                               22633 non-null  f
loat64
 7   final.output.tail_pb                               22516 non-null  f
loat64
 8   final.output.tail_sol                              22445 non-null  f
loat64
 9   final.output.tail_au                               22635 non-null  f
loat64
 10  primary_cleaner.input.sulfate                      21107 non-null  f
loat64
 11  primary_cleaner.input.depressant                   21170 non-null  f
loat64
 12  primary_cleaner.input.feed_size                    22716 non-null  f
loat64
 13  primary_cleaner.input.xanthate                     21565 non-null  f
loat64
 14  primary_cleaner.output.concentrate_ag              22618 non-null  f
loat64
 15  primary_cleaner.output.concentrate_pb              22268 non-null  f
loat64
 16  primary_cleaner.output.concentrate_sol             21918 non-null  f
loat64
 17  primary_cleaner.output.concentrate_au              22618 non-null  f
loat64
 18  primary_cleaner.output.tail_ag                     22614 non-null  f
loat64
 19  primary_cleaner.output.tail_pb                     22594 non-null  f
loat64
 20  primary_cleaner.output.tail_sol                    22365 non-null  f
loat64
 21  primary_cleaner.output.tail_au                     22617 non-null  f
loat64
 22  primary_cleaner.state.floatbank8_a_air             22660 non-null  f
loat64
 23  primary_cleaner.state.floatbank8_a_level           22667 non-null  f
loat64
 24  primary_cleaner.state.floatbank8_b_air             22660 non-null  f
loat64
 25  primary_cleaner.state.floatbank8_b_level           22673 non-null  f
loat64
 26  primary_cleaner.state.floatbank8_c_air             22662 non-null  f
loat64
```

```
 27  primary_cleaner.state.floatbank8_c_level            22673 non-null  f
loat64
 28  primary_cleaner.state.floatbank8_d_air              22661 non-null  f
loat64
 29  primary_cleaner.state.floatbank8_d_level            22673 non-null  f
loat64
 30  rougher.calculation.sulfate_to_au_concentrate       22672 non-null  f
loat64
 31  rougher.calculation.floatbank10_sulfate_to_au_feed  22672 non-null  f
loat64
 32  rougher.calculation.floatbank11_sulfate_to_au_feed  22672 non-null  f
loat64
 33  rougher.calculation.au_pb_ratio                     21089 non-null  f
loat64
 34  rougher.input.feed_ag                               22618 non-null  f
loat64
 35  rougher.input.feed_pb                               22472 non-null  f
loat64
 36  rougher.input.feed_rate                             22163 non-null  f
loat64
 37  rougher.input.feed_size                             22277 non-null  f
loat64
 38  rougher.input.feed_sol                              22357 non-null  f
loat64
 39  rougher.input.feed_au                               22617 non-null  f
loat64
 40  rougher.input.floatbank10_sulfate                   21415 non-null  f
loat64
 41  rougher.input.floatbank10_xanthate                  22247 non-null  f
loat64
 42  rougher.input.floatbank11_sulfate                   22038 non-null  f
loat64
 43  rougher.input.floatbank11_xanthate                  20459 non-null  f
loat64
 44  rougher.output.concentrate_ag                       22618 non-null  f
loat64
 45  rougher.output.concentrate_pb                       22618 non-null  f
loat64
 46  rougher.output.concentrate_sol                      22526 non-null  f
loat64
 47  rougher.output.concentrate_au                       22618 non-null  f
loat64
 48  rougher.output.recovery                             19597 non-null  f
loat64
 49  rougher.output.tail_ag                              19979 non-null  f
loat64
 50  rougher.output.tail_pb                              22618 non-null  f
loat64
 51  rougher.output.tail_sol                             19980 non-null  f
loat64
 52  rougher.output.tail_au                              19980 non-null  f
loat64
 53  rougher.state.floatbank10_a_air                     22646 non-null  f
loat64
 54  rougher.state.floatbank10_a_level                   22647 non-null  f
loat64
 55  rougher.state.floatbank10_b_air                     22646 non-null  f
loat64
 56  rougher.state.floatbank10_b_level                   22647 non-null  f
loat64
 57  rougher.state.floatbank10_c_air                     22646 non-null  f
```

```
loat64
 58  rougher.state.floatbank10_c_level                  22654 non-null  f
loat64
 59  rougher.state.floatbank10_d_air                    22641 non-null  f
loat64
 60  rougher.state.floatbank10_d_level                  22649 non-null  f
loat64
 61  rougher.state.floatbank10_e_air                    22096 non-null  f
loat64
 62  rougher.state.floatbank10_e_level                  22649 non-null  f
loat64
 63  rougher.state.floatbank10_f_air                    22641 non-null  f
loat64
 64  rougher.state.floatbank10_f_level                  22642 non-null  f
loat64
 65  secondary_cleaner.output.tail_ag                   22616 non-null  f
loat64
 66  secondary_cleaner.output.tail_pb                   22600 non-null  f
loat64
 67  secondary_cleaner.output.tail_sol                  20501 non-null  f
loat64
 68  secondary_cleaner.output.tail_au                   22618 non-null  f
loat64
 69  secondary_cleaner.state.floatbank2_a_air           22333 non-null  f
loat64
 70  secondary_cleaner.state.floatbank2_a_level         22591 non-null  f
loat64
 71  secondary_cleaner.state.floatbank2_b_air           22538 non-null  f
loat64
 72  secondary_cleaner.state.floatbank2_b_level         22588 non-null  f
loat64
 73  secondary_cleaner.state.floatbank3_a_air           22585 non-null  f
loat64
 74  secondary_cleaner.state.floatbank3_a_level         22587 non-null  f
loat64
 75  secondary_cleaner.state.floatbank3_b_air           22592 non-null  f
loat64
 76  secondary_cleaner.state.floatbank3_b_level         22590 non-null  f
loat64
 77  secondary_cleaner.state.floatbank4_a_air           22571 non-null  f
loat64
 78  secondary_cleaner.state.floatbank4_a_level         22587 non-null  f
loat64
 79  secondary_cleaner.state.floatbank4_b_air           22608 non-null  f
loat64
 80  secondary_cleaner.state.floatbank4_b_level         22607 non-null  f
loat64
 81  secondary_cleaner.state.floatbank5_a_air           22615 non-null  f
loat64
 82  secondary_cleaner.state.floatbank5_a_level         22615 non-null  f
loat64
 83  secondary_cleaner.state.floatbank5_b_air           22615 non-null  f
loat64
 84  secondary_cleaner.state.floatbank5_b_level         22616 non-null  f
loat64
 85  secondary_cleaner.state.floatbank6_a_air           22597 non-null  f
loat64
 86  secondary_cleaner.state.floatbank6_a_level         22615 non-null  f
loat64
dtypes: float64(86), object(1)
memory usage: 15.1+ MB
```

In [27]:

```
df_full.describe()
```

Out[27]:

	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_sol	fina
count	22627.000000	22629.000000	22331.000000	
mean	4.781559	9.095308	8.640317	
std	2.030128	3.230797	3.785035	
min	0.000000	0.000000	0.000000	
25%	4.018525	8.750171	7.116799	
50%	4.953729	9.914519	8.908792	
75%	5.862593	10.929839	10.705824	
max	16.001945	17.031899	19.615720	

8 rows × 86 columns

In [74]:

```
#corrigindo a tipagem da coluna date:
df_full['date']= pd.to_datetime(df_train['date'], format= '%Y-%m-%d %H:%M:%S')
```

In [77]:

```
df_full.head()
```

Out[77]:

	date	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_so
0	2016-01-15 00:00:00	6.055403	9.889648	5.507324
1	2016-01-15 01:00:00	6.029369	9.968944	5.257781
2	2016-01-15 02:00:00	6.055926	10.213995	5.383759
3	2016-01-15 03:00:00	6.047977	9.977019	4.858634
4	2016-01-15 04:00:00	6.148599	10.142511	4.939416

5 rows × 87 columns

VERIFICAÇÃO DO CÁLCULO DE RETIRADA DO OURO

In [78]:

```
df_copy = df_train.copy()
```

In [79]:

```
#verificando valores ausentes

df_copy[['rougher.output.concentrate_au', 'rougher.input.feed_au', 'rougher.output.tail_a
```

Out[79]:

```
rougher.output.concentrate_au      0
rougher.input.feed_au              0
rougher.output.tail_au           959
dtype: int64
```

In [80]:

```
df_copy['numerator'] = df_copy['rougher.output.concentrate_au'] * (df_copy['rougher.input
                                                                   - df_copy['rougher.out

df_copy['denominator'] = df_copy['rougher.input.feed_au'] * (df_copy['rougher.output.conc
                                                             - df_copy['rougher.output.ta
```

In [81]:

```
# valores ausentes do resultado

df_copy[['numerator', 'denominator']].isnull().sum()
```

Out[81]:

```
numerator      959
denominator    959
dtype: int64
```

In [82]:

```
#substituição dos valores ausentes

df_copy['numerator'] = df_copy['numerator'].fillna(0)
df_copy['denominator'] = df_copy['denominator'].fillna(1)
```

In [83]:

```
df_copy['formula'] = df_copy['numerator'] / df_copy['denominator'] * 100
df_copy.head(2)
```

Out[83]:

	date	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_so
0	2016-01-15 00:00:00	6.055403	9.889648	5.507324
1	2016-01-15 01:00:00	6.029369	9.968944	5.257781

2 rows × 90 columns

In [84]:

```
x = (df_copy['formula'] - df_copy['rougher.output.recovery']).abs().mean()

print('A diferença entre o resultado do cálculo e o resultado original é {} '.format(x))
```

```
A diferença entre o resultado do cálculo e o resultado original é 9.507869
053285221e-15
```

OBSERVAÇÃO

A DIFERENÇA É MÍNIMA. PORTANTO, CONSIDERA-SE QUE NÃO HÁ DIFERENÇA.

ANÁLISE DAS CARACTERÍSTICAS NÃO DISPONÍVEIS NO CONJUNTO DE TESTES

In [85]:

```
#coluna não disponível no conjunto de teste

colunas_ausentes_teste = set(list(df_train.columns.values))-set(list(df_test.columns.valu
colunas_ausentes_teste
```

Out[85]:

```
{'final.output.concentrate_ag',
 'final.output.concentrate_au',
 'final.output.concentrate_pb',
 'final.output.concentrate_sol',
 'final.output.recovery',
 'final.output.tail_ag',
 'final.output.tail_au',
 'final.output.tail_pb',
 'final.output.tail_sol',
 'primary_cleaner.output.concentrate_ag',
 'primary_cleaner.output.concentrate_au',
 'primary_cleaner.output.concentrate_pb',
 'primary_cleaner.output.concentrate_sol',
 'primary_cleaner.output.tail_ag',
 'primary_cleaner.output.tail_au',
 'primary_cleaner.output.tail_pb',
 'primary_cleaner.output.tail_sol',
 'rougher.calculation.au_pb_ratio',
 'rougher.calculation.floatbank10_sulfate_to_au_feed',
 'rougher.calculation.floatbank11_sulfate_to_au_feed',
 'rougher.calculation.sulfate_to_au_concentrate',
 'rougher.output.concentrate_ag',
 'rougher.output.concentrate_au',
 'rougher.output.concentrate_pb',
 'rougher.output.concentrate_sol',
 'rougher.output.recovery',
 'rougher.output.tail_ag',
 'rougher.output.tail_au',
 'rougher.output.tail_pb',
 'rougher.output.tail_sol',
 'secondary_cleaner.output.tail_ag',
 'secondary_cleaner.output.tail_au',
 'secondary_cleaner.output.tail_pb',
 'secondary_cleaner.output.tail_sol'}
```

In [86]:

```
df_full[colunas_ausentes_teste].dtypes
```

Out[86]:

```
secondary_cleaner.output.tail_au                    float64
final.output.concentrate_au                         float64
secondary_cleaner.output.tail_sol                   float64
secondary_cleaner.output.tail_pb                    float64
rougher.output.tail_sol                             float64
final.output.tail_pb                                float64
rougher.output.tail_pb                              float64
final.output.recovery                               float64
primary_cleaner.output.tail_au                      float64
rougher.output.tail_au                              float64
rougher.output.concentrate_au                       float64
rougher.output.tail_ag                              float64
final.output.concentrate_sol                        float64
final.output.concentrate_pb                         float64
rougher.calculation.au_pb_ratio                     float64
rougher.calculation.floatbank10_sulfate_to_au_feed  float64
final.output.tail_ag                                float64
primary_cleaner.output.concentrate_sol              float64
rougher.output.concentrate_sol                      float64
rougher.calculation.floatbank11_sulfate_to_au_feed  float64
final.output.tail_sol                               float64
primary_cleaner.output.concentrate_pb               float64
final.output.tail_au                                float64
rougher.output.concentrate_ag                       float64
rougher.output.concentrate_pb                       float64
primary_cleaner.output.tail_ag                      float64
final.output.concentrate_ag                         float64
primary_cleaner.output.tail_sol                     float64
rougher.calculation.sulfate_to_au_concentrate       float64
primary_cleaner.output.concentrate_au               float64
primary_cleaner.output.concentrate_ag               float64
primary_cleaner.output.tail_pb                      float64
rougher.output.recovery                             float64
secondary_cleaner.output.tail_ag                    float64
dtype: object
```

OBSERVAÇÃO

AS COLUNAS SÃO STRINGS E O OBJETIVO ESTÁ AUSENTE.

ANÁLISE DOS DADOS

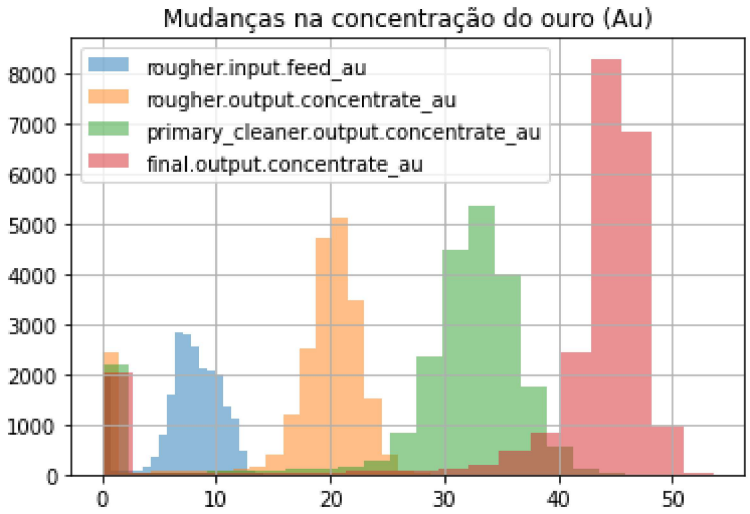
In [87]:

```
#Observe como a concentração de metais (Au, Ag, Pb) muda dependendo do estágio de purific

fase = [
    'rougher.input.feed_{}',
    'rougher.output.concentrate_{}',
    'primary_cleaner.output.concentrate_{}',
    'final.output.concentrate_{}',
]

def progresso(logam):
    step = [s.format(logam) for s in fase]
    for i in step:
        df_full[i].hist(alpha=0.5, bins=20)
    plt.legend(step)

progresso('au')
plt.title('Mudanças na concentração do ouro (Au)');
```

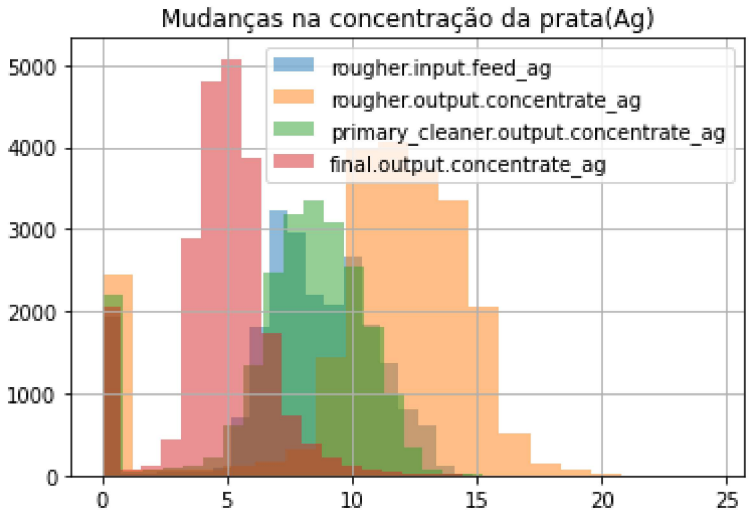


OBSERVAÇÃO

A CADA PROCESSO OBSERVA-SE A ALTERAÇÃO NA CONCENTRAÇÃO DO OURO

In [88]:

```
progresso('ag')
plt.title('Mudanças na concentração da prata(Ag)');
```

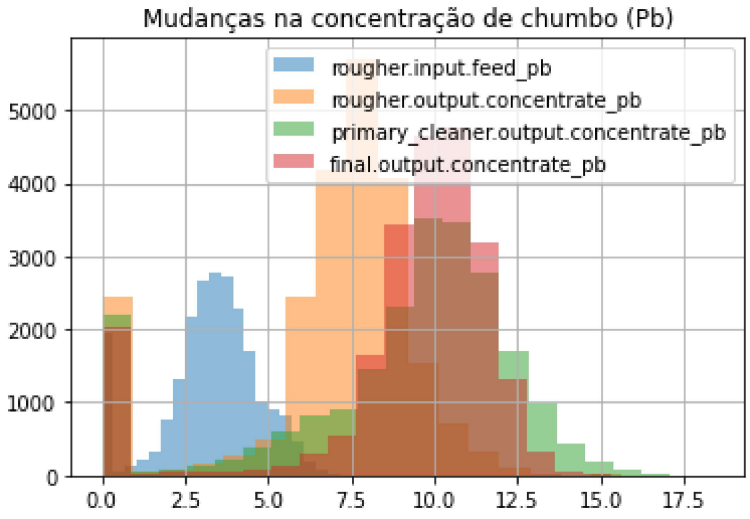


OBSERVAÇÃO

HÁ UMA ALTERAÇÃO NA CONCENTRAÇÃO DA PRATA, A CONCENTRAÇÃO FINAL ESTÁ ABAIXO DA CONCENTRAÇÃO INICIAL

In [89]:

```
progresso('pb')
plt.title('Mudanças na concentração de chumbo (Pb)');
```



OBSERVAÇÃO

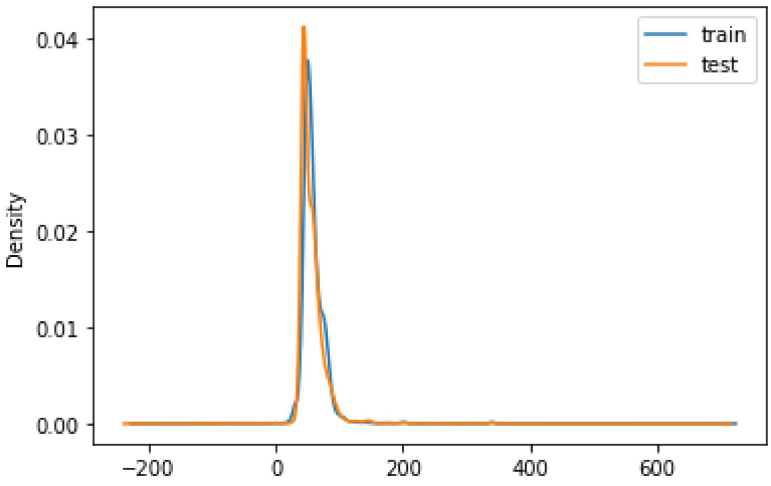
A CONCENTRAÇÃO DE CHUMBO SOFRE ALTERAÇÃO.

In [90]:

```
# Compare as distribuições de tamanho de partícula de minério no conjunto de treinamento
df_train['rougher.input.feed_size'].plot.kde();
df_test['rougher.input.feed_size'].plot.kde();
plt.legend(['train', 'test'])
```

Out[90]:

```
<matplotlib.legend.Legend at 0x7f3ddfc59c70>
```



OBSERVAÇÃO

TRATA-SE DE UMA DISTRIBUIÇÃO EQUILIBRADA ENTRE OS CONJUNTOS DE TREINAMENTO E TESTE

In [91]:

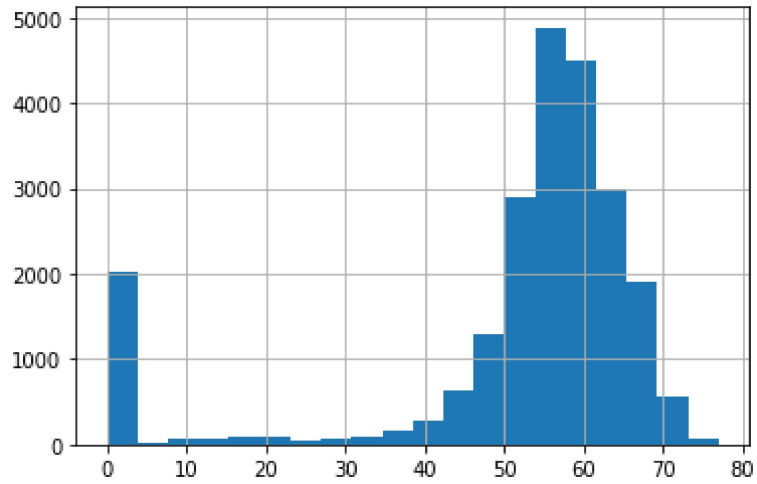
```
#Considere as concentrações totais de todas as substâncias em diferentes estágios:
#minério bruto, concentrado bruto e concentrado final.

feature_input = [
    'rougher.input.feed_au',
    'rougher.input.feed_ag',
    'rougher.input.feed_pb',
    'rougher.input.feed_sol',
]

df_full[feature_input].sum(1).hist(bins=20)
```

Out[91]:

<AxesSubplot:>



OBSERVAÇÃO

HÁ UMA IRREGULARIDADE NO PONTO ZERO, SERÁ REMOVIDO

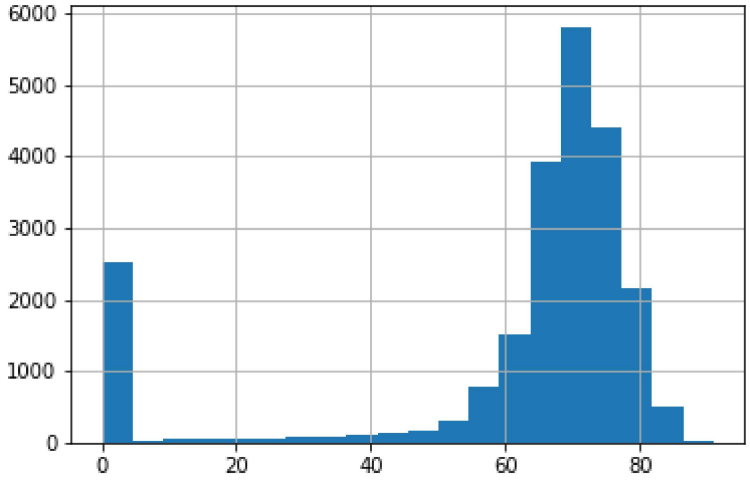
In [92]:

```
feature_output = [
    'rougher.output.concentrate_au',
    'rougher.output.concentrate_ag',
    'rougher.output.concentrate_pb',
    'rougher.output.concentrate_sol',
]

df_full[feature_output].sum(1).hist(bins=20)
```

Out[92]:

<AxesSubplot:>



OBSERVAÇÃO

HÁ UMA IRREGULARIDADE NO PONTO ZERO, SERÁ REMOVIDO

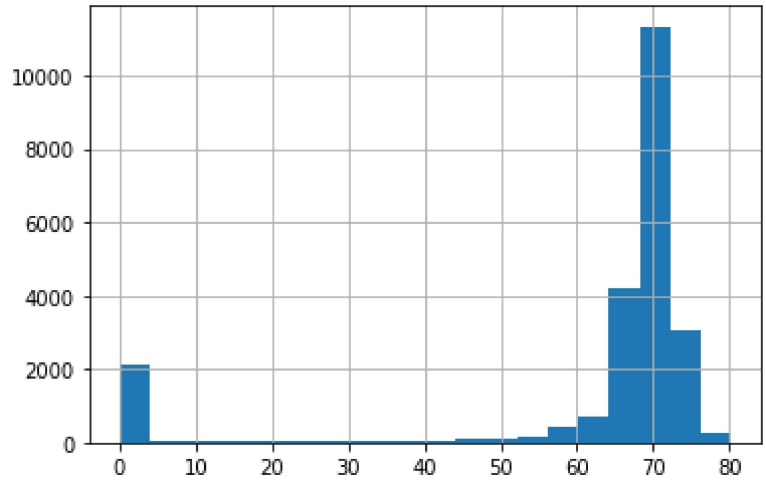
In [93]:

```
feature_final = [
    'final.output.concentrate_au',
    'final.output.concentrate_ag',
    'final.output.concentrate_pb',
    'final.output.concentrate_sol',
]

df_full[feature_final].sum(1).hist(bins=20)
```

Out[93]:

<AxesSubplot:>



OBSERVAÇÃO

HÁ UMA IRREGULARIDADE NO PONTO ZERO, SERÁ REMOVIDO

REMOÇÃO DAS IRREGULARIDADES

In [94]:

```
x = 0

df_train = df_train[df_train[feature_input].sum(1) > x]
df_train = df_train[df_train[feature_output].sum(1) > x]
df_train = df_train[df_train[feature_final].sum(1) > x]

df_full_test = df_full[df_full['date'].isin(df_test['date'])].reset_index(drop = True)
df_test = df_test[df_full_test[feature_input].sum(1) > x]
df_test = df_test[df_full_test[feature_output].sum(1) > x]
df_test = df_test[df_full_test[feature_final].sum(1) > x]

print(df_train.shape)
print(df_test.shape)
```

```
(15096, 87)
(4550, 53)

/tmp/ipykernel_27/403481252.py:8: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df_test = df_test[df_full_test[feature_input].sum(1) > x]
/tmp/ipykernel_27/403481252.py:9: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df_test = df_test[df_full_test[feature_output].sum(1) > x]
/tmp/ipykernel_27/403481252.py:10: UserWarning: Boolean Series key will be
reindexed to match DataFrame index.
  df_test = df_test[df_full_test[feature_final].sum(1) > x]
```

TRATANDO VALORES AUSENTES

In [95]:

```
col_x = df_test.columns.values.tolist()
col_x.append('rougher.output.recovery')
col_x.append('final.output.recovery')

df_train_y = df_train[col_x]

df_train_y = df_train_y[df_train_y['rougher.output.recovery'].notnull()].reset_index(drop
df_train_y = df_train_y[df_train_y['final.output.recovery'].notnull()].reset_index(drop =
```

In [96]:

```
print(df_train_y.shape)
print(df_test.shape)
```

```
(13795, 55)
(4550, 53)
```

In [97]:

```
df_full_y = pd.concat([df_train_y, df_test])
df_full_y.isnull().mean().sort_values(ascending = False)
```

Out[97]:

```
final.output.recovery                         0.248024
rougher.output.recovery                       0.248024
date                                          0.028128
rougher.input.floatbank11_xanthate            0.022949
rougher.state.floatbank10_e_air               0.020333
secondary_cleaner.state.floatbank2_a_air      0.011774
rougher.input.feed_size                       0.007577
rougher.input.feed_pb                         0.005451
rougher.input.feed_sol                        0.005288
primary_cleaner.input.xanthate                0.005288
rougher.input.floatbank11_sulfate             0.001744
rougher.input.floatbank10_sulfate             0.001744
primary_cleaner.input.depressant              0.001472
primary_cleaner.input.sulfate                 0.001090
secondary_cleaner.state.floatbank2_b_air      0.001090
secondary_cleaner.state.floatbank3_a_air      0.000654
rougher.input.feed_rate                       0.000600
rougher.input.floatbank10_xanthate            0.000491
secondary_cleaner.state.floatbank4_a_air      0.000218
primary_cleaner.state.floatbank8_b_air        0.000218
primary_cleaner.state.floatbank8_a_air        0.000218
primary_cleaner.state.floatbank8_d_air        0.000164
primary_cleaner.state.floatbank8_c_air        0.000109
secondary_cleaner.state.floatbank2_a_level    0.000055
rougher.state.floatbank10_c_air               0.000055
secondary_cleaner.state.floatbank3_b_air      0.000055
secondary_cleaner.state.floatbank2_b_level    0.000055
secondary_cleaner.state.floatbank3_a_level    0.000055
secondary_cleaner.state.floatbank5_b_air      0.000055
secondary_cleaner.state.floatbank3_b_level    0.000055
secondary_cleaner.state.floatbank4_a_level    0.000055
secondary_cleaner.state.floatbank4_b_air      0.000055
secondary_cleaner.state.floatbank5_a_level    0.000055
rougher.state.floatbank10_b_air               0.000055
secondary_cleaner.state.floatbank5_b_level    0.000055
secondary_cleaner.state.floatbank6_a_air      0.000055
rougher.state.floatbank10_c_level             0.000055
rougher.state.floatbank10_b_level             0.000055
secondary_cleaner.state.floatbank5_a_air      0.000055
rougher.state.floatbank10_a_level             0.000055
rougher.state.floatbank10_a_air               0.000055
secondary_cleaner.state.floatbank6_a_level    0.000055
primary_cleaner.state.floatbank8_d_level      0.000055
primary_cleaner.state.floatbank8_c_level      0.000055
primary_cleaner.state.floatbank8_b_level      0.000055
primary_cleaner.state.floatbank8_a_level      0.000055
secondary_cleaner.state.floatbank4_b_level    0.000055
primary_cleaner.input.feed_size               0.000000
rougher.state.floatbank10_f_level             0.000000
rougher.state.floatbank10_f_air               0.000000
rougher.state.floatbank10_e_level             0.000000
rougher.state.floatbank10_d_level             0.000000
rougher.input.feed_au                         0.000000
rougher.input.feed_ag                         0.000000
rougher.state.floatbank10_d_air               0.000000
dtype: float64
```

In [98]:

```
for i in df_full_y.isnull().mean().sort_values(ascending = False)[2:-2].index:
    median_cols = df_full_y[i].median()
    df_full_y[i] = df_full_y[i].fillna(median_cols)
```

In [99]:

```
df_test_final = df_full_y[df_full_y['rougher.output.recovery'].isnull()]
df_test_final = df_test_final.drop(columns = ['rougher.output.recovery', 'final.output.re

df_train_final = df_full_y[df_full_y['rougher.output.recovery'].notnull()].reset_index(dr
df_train_final.head(2)
```

Out[99]:

	date	primary_cleaner.input.sulfate	primary_cleaner.input.depressant	primary_cleaner.inp
0	2016-01-15 00:00:00	127.092003	10.128295	
1	2016-01-15 01:00:00	125.629232	10.296251	

2 rows × 55 columns

CONSTRUÇÃO DO MODELO

In [100]:

```
# separação dos dados - características e objetivos
feature = df_test_final.drop(columns = ['date']).columns.values
target = ['rougher.output.recovery', 'final.output.recovery']
```

In [101]:

```
#conjunto de treino
feature_train = df_train_final[feature].reset_index(drop = True)
target_train = df_train_final[target].reset_index(drop = True)
target_train.columns = [0,1]
```

In [102]:

```
#conjunto de teste
feature_test = df_test_final[feature].reset_index(drop = True)
target_test = df_full[target].loc[df_test_final.index].reset_index(drop = True)
target_test.columns = [0,1]
```

FUNÇÃO PARA O CÁLCULO sMAPE

In [103]:

```
def smape(i_true, i_pred):
    error = (i_true - i_pred).abs()
    scale = (i_true.abs() + i_pred.abs()) / 2

    return (error / scale).mean() * 100

def smape_weighted(j_true, j_pred):
    rougher = smape(j_true[0], j_pred[0])
    final = smape(j_true[1], j_pred[1])

    return 0.25 * rougher + 0.75 * final
```

MODELO DE BASE

In [104]:

```
pred_median = target_train.copy()
pred_median[0] = target_train[0].median()
pred_median[1] = target_train[1].median()
print(smape_weighted(target_train, pred_median))

pred_median = target_test.copy()
pred_median[0] = target_train[0].median()
pred_median[1] = target_train[1].median()
print(smape_weighted(target_test, pred_median))
```

```
9.468182190998984
12.983663690775849
```

MODELO REAL

In [105]:

```
# função para treinar o modelo com validação cruzada para obter o melhor hiperparâmetro p

def score_model(model, cv = 4):
    score = []

    for subtrain_index, valid_index in KFold(n_splits=cv).split(feature_train):

        feature_subtrain = feature_train.loc[subtrain_index].reset_index(drop = True)
        target_subtrain = target_train.loc[subtrain_index].reset_index(drop = True)

        feature_valid = feature_train.loc[valid_index].reset_index(drop = True)
        target_valid = target_train.loc[valid_index].reset_index(drop = True)

        model.fit(feature_subtrain, target_subtrain)
        pred_valid = pd.DataFrame(model.predict(feature_valid))

        score.append(smape_weighted(target_valid, pred_valid))

    return pd.Series(score).mean()
```

In [106]:

```
model = LinearRegression()
linear_score = score_model(model)
linear_score
```

Out[106]:

```
10.30152978934856
```

In [ ]:

```
for depth in range(1,5):
    model = RandomForestRegressor(max_depth= depth, n_estimators = 50, random_state=12345
    score = score_model(model)
    print('RF, depth = ', depth, 'score = ', score)
```

```
RF, depth =  1 score =  10.234362942796965
RF, depth =  2 score =  9.803892473144353
RF, depth =  3 score =  9.363486569667147
```

TREINANDO O MODELO

In [ ]:

```
#APLICANDO FLORESTA ALEATÓRIA COM DEPTH= 3
model = RandomForestRegressor(max_depth= 3, n_estimators = 50, random_state=12345)
model.fit(feature_train, target_train)

predict = pd.DataFrame(model.predict(feature_train))
print('train : ', smape_weighted(target_train, predict))

predict = pd.DataFrame(model.predict(feature_test))
print('test : ', smape_weighted(target_test, predict))
```

OBSERVAÇÃO

O SMAPE PARA OS DADOS DE TREINAMENTO É 7,48%, E PARA OS DADOS DE TESTE É 13.98% ESTE RESULTADO SIGNIFICA QUE, O MODELO DE TREINAMENTO POSSUI UM MELHOR DESEMPENHO

CONCLUSÃO

IMPORTAÇÃO DAS BIBLIOTECAS

AS BIBLIOTECAS NECESSÁRIAS FORAM IMPORTADAS CONFORME A DEMANDA DAS ATIVIDADES APLICADAS NESTE PROJETO.

PREPARAÇÃO DOS DADOS

OS DADOS FORAM CARREGADOS E FORAM DETECTADOS VALORES AUSENTES. CONSTATA-SE QUE NÃO HÁ DIFERENÇA ENTRE O RESULTADO DO CÁLCULO E O RESULTADO ORIGINAL.

ANÁLISE DOS DADOS

- OURO; A concentração de ouro muda consideravelmente a cada processo de refino\
- PRATA; A concentração de prata muda, mas não na extensão do processo de refino de ouro\
- CHUMBO; HÁ ALTERAÇÃO NA CONCENTRAÇÃO DE CHUMBO.

OS DADOS NO CONJUNTO DE TESTES E TREINAMENTO SÃO DISTRIBUÍDOS UNIFORMEMENTE. EM TODAS AS FASES DE PURIFICAÇÃO, EXISTE UMA IRREGULARIDADE QUE FOI REMOVIDA, E OS VALORES AUSENTES SENDO SUBSTITUÍDOS PELA MEDIANA.

CONSTRUÇÃO DO MODELO

FOI CRIADA UMA FUNÇÃO PARA CALCULAR O VALOR SMAP. E FOI APLICADO A FLORESTA ALEATÓRIA COM DEPTH =3.

MODELO TREINADO

O SMAPE PARA OS DADOS DE TREINAMENTO É 7,48%, E PARA OS DADOS DE TESTE É 13.98% ESTE RESULTADO SIGNIFICA QUE, O MODELO DE TREINAMENTO POSSUI UM MELHOR DESEMPENHO