Содержание

Восстановление золота из руды

Подготовьте прототип модели машинного обучения для «Цифры». Компания разрабатывает решения для эффективной работы промышленных предприятий.

Модель должна предсказать коэффициент восстановления золота из золотосодержащей руды. Используйте данные с параметрами добычи и очистки.

Модель поможет оптимизировать производство, чтобы не запускать предприятие с убыточными характеристиками.

Вам нужно:

1. Подготовить данные;
2. Провести исследовательский анализ данных;
3. Построить и обучить модель.

Чтобы выполнить проект, обращайтесь к библиотекам *pandas*, *matplotlib* и *sklearn.* Вам поможет их документация.

Подготовка данных

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import mean_absolute_error as mae
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer
from sklearn.model_selection import GridSearchCV
```

```
import warnings
warnings.filterwarnings("ignore")
```

In [2]:

```
pd.set_option('display.max_columns', None)
df_train = pd.read_csv('/home/cookie/projects/gold_recovery_train_new.csv')
df_test = pd.read_csv('/home/cookie/projects/gold_recovery_test_new.csv')
df_full = pd.read_csv('/home/cookie/projects/gold_recovery_full_new.csv')
```

In [3]:

```
df_train.head(10)
```

Out[3]:

	date	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_sol	final.output.concen
0	2016-01-15 00:00:00	6.055403	9.889648	5.507324	42
1	2016-01-15 01:00:00	6.029369	9.968944	5.257781	42
2	2016-01-15 02:00:00	6.055926	10.213995	5.383759	42
3	2016-01-15 03:00:00	6.047977	9.977019	4.858634	42
4	2016-01-15 04:00:00	6.148599	10.142511	4.939416	42
5	2016-01-15 05:00:00	6.482968	10.049416	5.480257	41
6	2016-01-15 06:00:00	6.533849	10.058141	4.569100	41
7	2016-01-15 07:00:00	6.130823	9.935481	4.389813	42
8	2016-01-15 08:00:00	5.834140	10.071156	4.876389	43
9	2016-01-15 09:00:00	5.687063	9.980404	5.282514	43

In [4]:

```
df_test.head(10)
```

Out[4]:

	date	primary_cleaner.input.sulfate	primary_cleaner.input.depressant	primary_cleaner.input.feed_size	primary_
0	2016-09-01 00:59:59	210.800909	14.993118	8.080000	
1	2016-09-01 01:59:59	215.392455	14.987471	8.080000	
2	2016-09-01 02:59:59	215.259946	12.884934	7.786667	

	date	primary_cleaner.input.sulfate	primary_cleaner.input.depressant	primary_cleaner.input.feed_size	primary_
3	2016-09-01 03:59:59	215.336236	12.006805	7.640000	
4	2016-09-01 04:59:59	199.099327	10.682530	7.530000	
5	2016-09-01 05:59:59	168.485085	8.817007	7.420000	
6	2016-09-01 06:59:59	144.133440	7.924610	7.420000	
7	2016-09-01 07:59:59	133.513396	8.055252	6.988000	
8	2016-09-01 08:59:59	133.735356	7.999618	6.935000	
9	2016-09-01 09:59:59	126.961069	8.017856	7.030000	

In [5]:
```
df_full.head(10)
```

Out[5]:

	date	final.output.concentrate_ag	final.output.concentrate_pb	final.output.concentrate_sol	final.output.concen
0	2016-01-15 00:00:00	6.055403	9.889648	5.507324	42
1	2016-01-15 01:00:00	6.029369	9.968944	5.257781	42
2	2016-01-15 02:00:00	6.055926	10.213995	5.383759	42
3	2016-01-15 03:00:00	6.047977	9.977019	4.858634	42
4	2016-01-15 04:00:00	6.148599	10.142511	4.939416	42
5	2016-01-15 05:00:00	6.482968	10.049416	5.480257	41
6	2016-01-15 06:00:00	6.533849	10.058141	4.569100	41
7	2016-01-15 07:00:00	6.130823	9.935481	4.389813	42
8	2016-01-15 08:00:00	5.834140	10.071156	4.876389	43
9	2016-01-15 09:00:00	5.687063	9.980404	5.282514	43

In [6]:

```
display(df_train.shape)
display(df_test.shape)
display(df_full.shape)
```

```
(14149, 87)
(5290, 53)
(19439, 87)
```

In [7]:

```
print('df_train')
display(df_train.info())
print('df_test')
display(df_test.info())
print('df_full')
display(df_full.info())
```

```
df_train
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14149 entries, 0 to 14148
Data columns (total 87 columns):
 #   Column                                              Non-Null Count  Dtype  
---  ------                                              --------------  -----  
 0   date                                                14149 non-null  object 
 1   final.output.concentrate_ag                         14148 non-null  float64
 2   final.output.concentrate_pb                         14148 non-null  float64
 3   final.output.concentrate_sol                        13938 non-null  float64
 4   final.output.concentrate_au                         14149 non-null  float64
 5   final.output.recovery                               14149 non-null  float64
 6   final.output.tail_ag                                14149 non-null  float64
 7   final.output.tail_pb                                14049 non-null  float64
 8   final.output.tail_sol                               14144 non-null  float64
 9   final.output.tail_au                                14149 non-null  float64
 10  primary_cleaner.input.sulfate                       14129 non-null  float64
 11  primary_cleaner.input.depressant                    14117 non-null  float64
 12  primary_cleaner.input.feed_size                     14149 non-null  float64
 13  primary_cleaner.input.xanthate                      14049 non-null  float64
 14  primary_cleaner.output.concentrate_ag               14149 non-null  float64
 15  primary_cleaner.output.concentrate_pb               14063 non-null  float64
 16  primary_cleaner.output.concentrate_sol              13863 non-null  float64
 17  primary_cleaner.output.concentrate_au               14149 non-null  float64
 18  primary_cleaner.output.tail_ag                      14148 non-null  float64
 19  primary_cleaner.output.tail_pb                      14134 non-null  float64
 20  primary_cleaner.output.tail_sol                     14103 non-null  float64
 21  primary_cleaner.output.tail_au                      14149 non-null  float64
 22  primary_cleaner.state.floatbank8_a_air              14145 non-null  float64
 23  primary_cleaner.state.floatbank8_a_level            14148 non-null  float64
 24  primary_cleaner.state.floatbank8_b_air              14145 non-null  float64
 25  primary_cleaner.state.floatbank8_b_level            14148 non-null  float64
 26  primary_cleaner.state.floatbank8_c_air              14147 non-null  float64
 27  primary_cleaner.state.floatbank8_c_level            14148 non-null  float64
 28  primary_cleaner.state.floatbank8_d_air              14146 non-null  float64
 29  primary_cleaner.state.floatbank8_d_level            14148 non-null  float64
 30  rougher.calculation.sulfate_to_au_concentrate       14148 non-null  float64
 31  rougher.calculation.floatbank10_sulfate_to_au_feed  14148 non-null  float64
 32  rougher.calculation.floatbank11_sulfate_to_au_feed  14148 non-null  float64
 33  rougher.calculation.au_pb_ratio                     14149 non-null  float64
 34  rougher.input.feed_ag                               14149 non-null  float64
 35  rougher.input.feed_pb                               14049 non-null  float64
 36  rougher.input.feed_rate                             14141 non-null  float64
 37  rougher.input.feed_size                             14005 non-null  float64
 38  rougher.input.feed_sol                              14071 non-null  float64
 39  rougher.input.feed_au                               14149 non-null  float64
 40  rougher.input.floatbank10_sulfate                   14120 non-null  float64
 41  rougher.input.floatbank10_xanthate                  14141 non-null  float64
```

```
 42  rougher.input.floatbank11_sulfate                   14113 non-null  float64
 43  rougher.input.floatbank11_xanthate                  13721 non-null  float64
 44  rougher.output.concentrate_ag                       14149 non-null  float64
 45  rougher.output.concentrate_pb                       14149 non-null  float64
 46  rougher.output.concentrate_sol                      14127 non-null  float64
 47  rougher.output.concentrate_au                       14149 non-null  float64
 48  rougher.output.recovery                             14149 non-null  float64
 49  rougher.output.tail_ag                              14148 non-null  float64
 50  rougher.output.tail_pb                              14149 non-null  float64
 51  rougher.output.tail_sol                             14149 non-null  float64
 52  rougher.output.tail_au                              14149 non-null  float64
 53  rougher.state.floatbank10_a_air                     14148 non-null  float64
 54  rougher.state.floatbank10_a_level                   14148 non-null  float64
 55  rougher.state.floatbank10_b_air                     14148 non-null  float64
 56  rougher.state.floatbank10_b_level                   14148 non-null  float64
 57  rougher.state.floatbank10_c_air                     14148 non-null  float64
 58  rougher.state.floatbank10_c_level                   14148 non-null  float64
 59  rougher.state.floatbank10_d_air                     14149 non-null  float64
 60  rougher.state.floatbank10_d_level                   14149 non-null  float64
 61  rougher.state.floatbank10_e_air                     13713 non-null  float64
 62  rougher.state.floatbank10_e_level                   14149 non-null  float64
 63  rougher.state.floatbank10_f_air                     14149 non-null  float64
 64  rougher.state.floatbank10_f_level                   14149 non-null  float64
 65  secondary_cleaner.output.tail_ag                    14147 non-null  float64
 66  secondary_cleaner.output.tail_pb                    14139 non-null  float64
 67  secondary_cleaner.output.tail_sol                   12544 non-null  float64
 68  secondary_cleaner.output.tail_au                    14149 non-null  float64
 69  secondary_cleaner.state.floatbank2_a_air            13932 non-null  float64
 70  secondary_cleaner.state.floatbank2_a_level          14148 non-null  float64
 71  secondary_cleaner.state.floatbank2_b_air            14128 non-null  float64
 72  secondary_cleaner.state.floatbank2_b_level          14148 non-null  float64
 73  secondary_cleaner.state.floatbank3_a_air            14145 non-null  float64
 74  secondary_cleaner.state.floatbank3_a_level          14148 non-null  float64
 75  secondary_cleaner.state.floatbank3_b_air            14148 non-null  float64
 76  secondary_cleaner.state.floatbank3_b_level          14148 non-null  float64
 77  secondary_cleaner.state.floatbank4_a_air            14143 non-null  float64
 78  secondary_cleaner.state.floatbank4_a_level          14148 non-null  float64
 79  secondary_cleaner.state.floatbank4_b_air            14148 non-null  float64
 80  secondary_cleaner.state.floatbank4_b_level          14148 non-null  float64
 81  secondary_cleaner.state.floatbank5_a_air            14148 non-null  float64
 82  secondary_cleaner.state.floatbank5_a_level          14148 non-null  float64
 83  secondary_cleaner.state.floatbank5_b_air            14148 non-null  float64
 84  secondary_cleaner.state.floatbank5_b_level          14148 non-null  float64
 85  secondary_cleaner.state.floatbank6_a_air            14147 non-null  float64
 86  secondary_cleaner.state.floatbank6_a_level          14148 non-null  float64
dtypes: float64(86), object(1)
memory usage: 9.4+ MB
None
df_test
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5290 entries, 0 to 5289
Data columns (total 53 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   date                                        5290 non-null   object
 1   primary_cleaner.input.sulfate               5286 non-null   float64
 2   primary_cleaner.input.depressant            5285 non-null   float64
 3   primary_cleaner.input.feed_size             5290 non-null   float64
 4   primary_cleaner.input.xanthate              5286 non-null   float64
 5   primary_cleaner.state.floatbank8_a_air      5290 non-null   float64
 6   primary_cleaner.state.floatbank8_a_level    5290 non-null   float64
 7   primary_cleaner.state.floatbank8_b_air      5290 non-null   float64
 8   primary_cleaner.state.floatbank8_b_level    5290 non-null   float64
 9   primary_cleaner.state.floatbank8_c_air      5290 non-null   float64
 10  primary_cleaner.state.floatbank8_c_level    5290 non-null   float64
 11  primary_cleaner.state.floatbank8_d_air      5290 non-null   float64
```

```
 12  primary_cleaner.state.floatbank8_d_level    5290 non-null   float64
 13  rougher.input.feed_ag                       5290 non-null   float64
 14  rougher.input.feed_pb                       5290 non-null   float64
 15  rougher.input.feed_rate                     5287 non-null   float64
 16  rougher.input.feed_size                     5289 non-null   float64
 17  rougher.input.feed_sol                      5269 non-null   float64
 18  rougher.input.feed_au                       5290 non-null   float64
 19  rougher.input.floatbank10_sulfate           5285 non-null   float64
 20  rougher.input.floatbank10_xanthate          5290 non-null   float64
 21  rougher.input.floatbank11_sulfate           5282 non-null   float64
 22  rougher.input.floatbank11_xanthate          5265 non-null   float64
 23  rougher.state.floatbank10_a_air             5290 non-null   float64
 24  rougher.state.floatbank10_a_level           5290 non-null   float64
 25  rougher.state.floatbank10_b_air             5290 non-null   float64
 26  rougher.state.floatbank10_b_level           5290 non-null   float64
 27  rougher.state.floatbank10_c_air             5290 non-null   float64
 28  rougher.state.floatbank10_c_level           5290 non-null   float64
 29  rougher.state.floatbank10_d_air             5290 non-null   float64
 30  rougher.state.floatbank10_d_level           5290 non-null   float64
 31  rougher.state.floatbank10_e_air             5290 non-null   float64
 32  rougher.state.floatbank10_e_level           5290 non-null   float64
 33  rougher.state.floatbank10_f_air             5290 non-null   float64
 34  rougher.state.floatbank10_f_level           5290 non-null   float64
 35  secondary_cleaner.state.floatbank2_a_air    5287 non-null   float64
 36  secondary_cleaner.state.floatbank2_a_level  5290 non-null   float64
 37  secondary_cleaner.state.floatbank2_b_air    5288 non-null   float64
 38  secondary_cleaner.state.floatbank2_b_level  5290 non-null   float64
 39  secondary_cleaner.state.floatbank3_a_air    5281 non-null   float64
 40  secondary_cleaner.state.floatbank3_a_level  5290 non-null   float64
 41  secondary_cleaner.state.floatbank3_b_air    5290 non-null   float64
 42  secondary_cleaner.state.floatbank3_b_level  5290 non-null   float64
 43  secondary_cleaner.state.floatbank4_a_air    5290 non-null   float64
 44  secondary_cleaner.state.floatbank4_a_level  5290 non-null   float64
 45  secondary_cleaner.state.floatbank4_b_air    5290 non-null   float64
 46  secondary_cleaner.state.floatbank4_b_level  5290 non-null   float64
 47  secondary_cleaner.state.floatbank5_a_air    5290 non-null   float64
 48  secondary_cleaner.state.floatbank5_a_level  5290 non-null   float64
 49  secondary_cleaner.state.floatbank5_b_air    5290 non-null   float64
 50  secondary_cleaner.state.floatbank5_b_level  5290 non-null   float64
 51  secondary_cleaner.state.floatbank6_a_air    5290 non-null   float64
 52  secondary_cleaner.state.floatbank6_a_level  5290 non-null   float64
dtypes: float64(52), object(1)
memory usage: 2.1+ MB
None
df_full
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19439 entries, 0 to 19438
Data columns (total 87 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   date                                                19439 non-null  object
 1   final.output.concentrate_ag                         19438 non-null  float64
 2   final.output.concentrate_pb                         19438 non-null  float64
 3   final.output.concentrate_sol                        19228 non-null  float64
 4   final.output.concentrate_au                         19439 non-null  float64
 5   final.output.recovery                               19439 non-null  float64
 6   final.output.tail_ag                                19438 non-null  float64
 7   final.output.tail_pb                                19338 non-null  float64
 8   final.output.tail_sol                               19433 non-null  float64
 9   final.output.tail_au                                19439 non-null  float64
 10  primary_cleaner.input.sulfate                       19415 non-null  float64
 11  primary_cleaner.input.depressant                    19402 non-null  float64
 12  primary_cleaner.input.feed_size                     19439 non-null  float64
 13  primary_cleaner.input.xanthate                      19335 non-null  float64
 14  primary_cleaner.output.concentrate_ag               19439 non-null  float64
 15  primary_cleaner.output.concentrate_pb               19323 non-null  float64
```

```
16  primary_cleaner.output.concentrate_sol              19069 non-null  float64
17  primary_cleaner.output.concentrate_au               19439 non-null  float64
18  primary_cleaner.output.tail_ag                      19435 non-null  float64
19  primary_cleaner.output.tail_pb                      19418 non-null  float64
20  primary_cleaner.output.tail_sol                     19377 non-null  float64
21  primary_cleaner.output.tail_au                      19439 non-null  float64
22  primary_cleaner.state.floatbank8_a_air              19435 non-null  float64
23  primary_cleaner.state.floatbank8_a_level            19438 non-null  float64
24  primary_cleaner.state.floatbank8_b_air              19435 non-null  float64
25  primary_cleaner.state.floatbank8_b_level            19438 non-null  float64
26  primary_cleaner.state.floatbank8_c_air              19437 non-null  float64
27  primary_cleaner.state.floatbank8_c_level            19438 non-null  float64
28  primary_cleaner.state.floatbank8_d_air              19436 non-null  float64
29  primary_cleaner.state.floatbank8_d_level            19438 non-null  float64
30  rougher.calculation.sulfate_to_au_concentrate       19437 non-null  float64
31  rougher.calculation.floatbank10_sulfate_to_au_feed  19437 non-null  float64
32  rougher.calculation.floatbank11_sulfate_to_au_feed  19437 non-null  float64
33  rougher.calculation.au_pb_ratio                     19439 non-null  float64
34  rougher.input.feed_ag                               19439 non-null  float64
35  rougher.input.feed_pb                               19339 non-null  float64
36  rougher.input.feed_rate                             19428 non-null  float64
37  rougher.input.feed_size                             19294 non-null  float64
38  rougher.input.feed_sol                              19340 non-null  float64
39  rougher.input.feed_au                               19439 non-null  float64
40  rougher.input.floatbank10_sulfate                   19405 non-null  float64
41  rougher.input.floatbank10_xanthate                  19431 non-null  float64
42  rougher.input.floatbank11_sulfate                   19395 non-null  float64
43  rougher.input.floatbank11_xanthate                  18986 non-null  float64
44  rougher.output.concentrate_ag                       19439 non-null  float64
45  rougher.output.concentrate_pb                       19439 non-null  float64
46  rougher.output.concentrate_sol                      19416 non-null  float64
47  rougher.output.concentrate_au                       19439 non-null  float64
48  rougher.output.recovery                             19439 non-null  float64
49  rougher.output.tail_ag                              19438 non-null  float64
50  rougher.output.tail_pb                              19439 non-null  float64
51  rougher.output.tail_sol                             19439 non-null  float64
52  rougher.output.tail_au                              19439 non-null  float64
53  rougher.state.floatbank10_a_air                     19438 non-null  float64
54  rougher.state.floatbank10_a_level                   19438 non-null  float64
55  rougher.state.floatbank10_b_air                     19438 non-null  float64
56  rougher.state.floatbank10_b_level                   19438 non-null  float64
57  rougher.state.floatbank10_c_air                     19438 non-null  float64
58  rougher.state.floatbank10_c_level                   19438 non-null  float64
59  rougher.state.floatbank10_d_air                     19439 non-null  float64
60  rougher.state.floatbank10_d_level                   19439 non-null  float64
61  rougher.state.floatbank10_e_air                     19003 non-null  float64
62  rougher.state.floatbank10_e_level                   19439 non-null  float64
63  rougher.state.floatbank10_f_air                     19439 non-null  float64
64  rougher.state.floatbank10_f_level                   19439 non-null  float64
65  secondary_cleaner.output.tail_ag                    19437 non-null  float64
66  secondary_cleaner.output.tail_pb                    19427 non-null  float64
67  secondary_cleaner.output.tail_sol                   17691 non-null  float64
68  secondary_cleaner.output.tail_au                    19439 non-null  float64
69  secondary_cleaner.state.floatbank2_a_air            19219 non-null  float64
70  secondary_cleaner.state.floatbank2_a_level          19438 non-null  float64
71  secondary_cleaner.state.floatbank2_b_air            19416 non-null  float64
72  secondary_cleaner.state.floatbank2_b_level          19438 non-null  float64
73  secondary_cleaner.state.floatbank3_a_air            19426 non-null  float64
74  secondary_cleaner.state.floatbank3_a_level          19438 non-null  float64
75  secondary_cleaner.state.floatbank3_b_air            19438 non-null  float64
76  secondary_cleaner.state.floatbank3_b_level          19438 non-null  float64
77  secondary_cleaner.state.floatbank4_a_air            19433 non-null  float64
78  secondary_cleaner.state.floatbank4_a_level          19438 non-null  float64
79  secondary_cleaner.state.floatbank4_b_air            19438 non-null  float64
80  secondary_cleaner.state.floatbank4_b_level          19438 non-null  float64
81  secondary_cleaner.state.floatbank5_a_air            19438 non-null  float64
```

```
 82  secondary_cleaner.state.floatbank5_a_level          19438 non-null  float64
 83  secondary_cleaner.state.floatbank5_b_air            19438 non-null  float64
 84  secondary_cleaner.state.floatbank5_b_level          19438 non-null  float64
 85  secondary_cleaner.state.floatbank6_a_air            19437 non-null  float64
 86  secondary_cleaner.state.floatbank6_a_level          19438 non-null  float64
dtypes: float64(86), object(1)
memory usage: 12.9+ MB
None
```

In [8]:

```
display(df_train.isnull().mean())
display(df_test.isnull().mean())
display(df_full.isnull().mean())
```

```
date                                          0.000000
final.output.concentrate_ag                   0.000071
final.output.concentrate_pb                   0.000071
final.output.concentrate_sol                  0.014913
final.output.concentrate_au                   0.000000
                                                ...
secondary_cleaner.state.floatbank5_a_level    0.000071
secondary_cleaner.state.floatbank5_b_air      0.000071
secondary_cleaner.state.floatbank5_b_level    0.000071
secondary_cleaner.state.floatbank6_a_air      0.000141
secondary_cleaner.state.floatbank6_a_level    0.000071
Length: 87, dtype: float64
date                                          0.000000
primary_cleaner.input.sulfate                 0.000756
primary_cleaner.input.depressant              0.000945
primary_cleaner.input.feed_size               0.000000
primary_cleaner.input.xanthate                0.000756
primary_cleaner.state.floatbank8_a_air        0.000000
primary_cleaner.state.floatbank8_a_level      0.000000
primary_cleaner.state.floatbank8_b_air        0.000000
primary_cleaner.state.floatbank8_b_level      0.000000
primary_cleaner.state.floatbank8_c_air        0.000000
primary_cleaner.state.floatbank8_c_level      0.000000
primary_cleaner.state.floatbank8_d_air        0.000000
primary_cleaner.state.floatbank8_d_level      0.000000
rougher.input.feed_ag                         0.000000
rougher.input.feed_pb                         0.000000
rougher.input.feed_rate                       0.000567
rougher.input.feed_size                       0.000189
rougher.input.feed_sol                        0.003970
rougher.input.feed_au                         0.000000
rougher.input.floatbank10_sulfate             0.000945
rougher.input.floatbank10_xanthate            0.000000
rougher.input.floatbank11_sulfate             0.001512
rougher.input.floatbank11_xanthate            0.004726
rougher.state.floatbank10_a_air               0.000000
rougher.state.floatbank10_a_level             0.000000
rougher.state.floatbank10_b_air               0.000000
rougher.state.floatbank10_b_level             0.000000
rougher.state.floatbank10_c_air               0.000000
rougher.state.floatbank10_c_level             0.000000
rougher.state.floatbank10_d_air               0.000000
rougher.state.floatbank10_d_level             0.000000
rougher.state.floatbank10_e_air               0.000000
rougher.state.floatbank10_e_level             0.000000
rougher.state.floatbank10_f_air               0.000000
rougher.state.floatbank10_f_level             0.000000
secondary_cleaner.state.floatbank2_a_air      0.000567
secondary_cleaner.state.floatbank2_a_level    0.000000
secondary_cleaner.state.floatbank2_b_air      0.000378
secondary_cleaner.state.floatbank2_b_level    0.000000
secondary_cleaner.state.floatbank3_a_air      0.001701
```

```
secondary_cleaner.state.floatbank3_a_level    0.000000
secondary_cleaner.state.floatbank3_b_air      0.000000
secondary_cleaner.state.floatbank3_b_level    0.000000
secondary_cleaner.state.floatbank4_a_air      0.000000
secondary_cleaner.state.floatbank4_a_level    0.000000
secondary_cleaner.state.floatbank4_b_air      0.000000
secondary_cleaner.state.floatbank4_b_level    0.000000
secondary_cleaner.state.floatbank5_a_air      0.000000
secondary_cleaner.state.floatbank5_a_level    0.000000
secondary_cleaner.state.floatbank5_b_air      0.000000
secondary_cleaner.state.floatbank5_b_level    0.000000
secondary_cleaner.state.floatbank6_a_air      0.000000
secondary_cleaner.state.floatbank6_a_level    0.000000
dtype: float64
date                                          0.000000
final.output.concentrate_ag                   0.000051
final.output.concentrate_pb                   0.000051
final.output.concentrate_sol                  0.010854
final.output.concentrate_au                   0.000000
                                                ...
secondary_cleaner.state.floatbank5_a_level    0.000051
secondary_cleaner.state.floatbank5_b_air      0.000051
secondary_cleaner.state.floatbank5_b_level    0.000051
secondary_cleaner.state.floatbank6_a_air      0.000103
secondary_cleaner.state.floatbank6_a_level    0.000051
Length: 87, dtype: float64
```

Расчет эффективности обогащения

In [9]:
```python
def recovery(row):
    c = row['rougher.output.concentrate_au']
    f = row['rougher.input.feed_au']
    t = row['rougher.output.tail_au']

    result = ((c * (f - t)) / (f * (c - t))) * 100

    return result
```

In [10]:
```python
recovery_calc = df_train.apply(recovery, axis=1)
```

In [11]:
```python
recovery_calc
```

Out[11]:
```
0        87.107763
1        86.843261
2        86.842308
3        87.226430
4        86.688794
             ...
14144    89.574376
14145    87.724007
14146    88.890579
14147    89.858126
14148    89.514960
Length: 14149, dtype: float64
```

In [12]:
```python
recovery_true = df_train['rougher.output.recovery']
```

In [13]:
```python
recovery_true
```

Out[13]:
```
0        87.107763
1        86.843261
2        86.842308
3        87.226430
4        86.688794
           ...
14144    89.574376
14145    87.724007
14146    88.890579
14147    89.858126
14148    89.514960
Name: rougher.output.recovery, Length: 14149, dtype: float64
```

In [14]:
```
mae(y_true=recovery_true, y_pred=recovery_calc)
```

Out[14]:
```
9.73512347450521e-15
```

Вывод

Если смотреть "глазами" и сравнивать значения, то вычисления совпадают, но не должна ли MAE в таком случае выдавать ноль?

Может в df_train есть пропущенные значения, а в расчетах нет, поэтому значение больше нуля. А не проверить ли

In [15]:
```
df_testing = df_train.dropna()
```

In [16]:
```
recovery_calc_testing = df_testing.apply(recovery, axis=1)
```

In [17]:
```
recovery_true_testing = df_testing['rougher.output.recovery']
```

In [18]:
```
mae(recovery_true_testing, recovery_calc_testing)
```

Out[18]:
```
9.82970122149377e-15
```

Видимо дело было не в этом. Значит в датасете изначально расчеты были не верны

In [19]:
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5290 entries, 0 to 5289
Data columns (total 53 columns):
 #   Column                                      Non-Null Count  Dtype  
---  ------                                      --------------  -----  
 0   date                                        5290 non-null   object 
 1   primary_cleaner.input.sulfate               5286 non-null   float64
 2   primary_cleaner.input.depressant            5285 non-null   float64
 3   primary_cleaner.input.feed_size             5290 non-null   float64
 4   primary_cleaner.input.xanthate              5286 non-null   float64
 5   primary_cleaner.state.floatbank8_a_air      5290 non-null   float64
 6   primary_cleaner.state.floatbank8_a_level    5290 non-null   float64
 7   primary_cleaner.state.floatbank8_b_air      5290 non-null   float64
 8   primary_cleaner.state.floatbank8_b_level    5290 non-null   float64
 9   primary_cleaner.state.floatbank8_c_air      5290 non-null   float64
 10  primary_cleaner.state.floatbank8_c_level    5290 non-null   float64
 11  primary_cleaner.state.floatbank8_d_air      5290 non-null   float64
 12  primary_cleaner.state.floatbank8_d_level    5290 non-null   float64
```

```
 13  rougher.input.feed_ag                       5290 non-null   float64
 14  rougher.input.feed_pb                       5290 non-null   float64
 15  rougher.input.feed_rate                     5287 non-null   float64
 16  rougher.input.feed_size                     5289 non-null   float64
 17  rougher.input.feed_sol                      5269 non-null   float64
 18  rougher.input.feed_au                       5290 non-null   float64
 19  rougher.input.floatbank10_sulfate           5285 non-null   float64
 20  rougher.input.floatbank10_xanthate          5290 non-null   float64
 21  rougher.input.floatbank11_sulfate           5282 non-null   float64
 22  rougher.input.floatbank11_xanthate          5265 non-null   float64
 23  rougher.state.floatbank10_a_air             5290 non-null   float64
 24  rougher.state.floatbank10_a_level           5290 non-null   float64
 25  rougher.state.floatbank10_b_air             5290 non-null   float64
 26  rougher.state.floatbank10_b_level           5290 non-null   float64
 27  rougher.state.floatbank10_c_air             5290 non-null   float64
 28  rougher.state.floatbank10_c_level           5290 non-null   float64
 29  rougher.state.floatbank10_d_air             5290 non-null   float64
 30  rougher.state.floatbank10_d_level           5290 non-null   float64
 31  rougher.state.floatbank10_e_air             5290 non-null   float64
 32  rougher.state.floatbank10_e_level           5290 non-null   float64
 33  rougher.state.floatbank10_f_air             5290 non-null   float64
 34  rougher.state.floatbank10_f_level           5290 non-null   float64
 35  secondary_cleaner.state.floatbank2_a_air    5287 non-null   float64
 36  secondary_cleaner.state.floatbank2_a_level  5290 non-null   float64
 37  secondary_cleaner.state.floatbank2_b_air    5288 non-null   float64
 38  secondary_cleaner.state.floatbank2_b_level  5290 non-null   float64
 39  secondary_cleaner.state.floatbank3_a_air    5281 non-null   float64
 40  secondary_cleaner.state.floatbank3_a_level  5290 non-null   float64
 41  secondary_cleaner.state.floatbank3_b_air    5290 non-null   float64
 42  secondary_cleaner.state.floatbank3_b_level  5290 non-null   float64
 43  secondary_cleaner.state.floatbank4_a_air    5290 non-null   float64
 44  secondary_cleaner.state.floatbank4_a_level  5290 non-null   float64
 45  secondary_cleaner.state.floatbank4_b_air    5290 non-null   float64
 46  secondary_cleaner.state.floatbank4_b_level  5290 non-null   float64
 47  secondary_cleaner.state.floatbank5_a_air    5290 non-null   float64
 48  secondary_cleaner.state.floatbank5_a_level  5290 non-null   float64
 49  secondary_cleaner.state.floatbank5_b_air    5290 non-null   float64
 50  secondary_cleaner.state.floatbank5_b_level  5290 non-null   float64
 51  secondary_cleaner.state.floatbank6_a_air    5290 non-null   float64
 52  secondary_cleaner.state.floatbank6_a_level  5290 non-null   float64
dtypes: float64(52), object(1)
memory usage: 2.1+ MB
```

In [20]:

```
df_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19439 entries, 0 to 19438
Data columns (total 87 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   date                                                19439 non-null  object
 1   final.output.concentrate_ag                         19438 non-null  float64
 2   final.output.concentrate_pb                         19438 non-null  float64
 3   final.output.concentrate_sol                        19228 non-null  float64
 4   final.output.concentrate_au                         19439 non-null  float64
 5   final.output.recovery                               19439 non-null  float64
 6   final.output.tail_ag                                19438 non-null  float64
 7   final.output.tail_pb                                19338 non-null  float64
 8   final.output.tail_sol                               19433 non-null  float64
 9   final.output.tail_au                                19439 non-null  float64
 10  primary_cleaner.input.sulfate                       19415 non-null  float64
 11  primary_cleaner.input.depressant                    19402 non-null  float64
 12  primary_cleaner.input.feed_size                     19439 non-null  float64
 13  primary_cleaner.input.xanthate                      19335 non-null  float64
 14  primary_cleaner.output.concentrate_ag               19439 non-null  float64
```

```
15  primary_cleaner.output.concentrate_pb                19323 non-null  float64
16  primary_cleaner.output.concentrate_sol               19069 non-null  float64
17  primary_cleaner.output.concentrate_au                19439 non-null  float64
18  primary_cleaner.output.tail_ag                       19435 non-null  float64
19  primary_cleaner.output.tail_pb                       19418 non-null  float64
20  primary_cleaner.output.tail_sol                      19377 non-null  float64
21  primary_cleaner.output.tail_au                       19439 non-null  float64
22  primary_cleaner.state.floatbank8_a_air               19435 non-null  float64
23  primary_cleaner.state.floatbank8_a_level             19438 non-null  float64
24  primary_cleaner.state.floatbank8_b_air               19435 non-null  float64
25  primary_cleaner.state.floatbank8_b_level             19438 non-null  float64
26  primary_cleaner.state.floatbank8_c_air               19437 non-null  float64
27  primary_cleaner.state.floatbank8_c_level             19438 non-null  float64
28  primary_cleaner.state.floatbank8_d_air               19436 non-null  float64
29  primary_cleaner.state.floatbank8_d_level             19438 non-null  float64
30  rougher.calculation.sulfate_to_au_concentrate        19437 non-null  float64
31  rougher.calculation.floatbank10_sulfate_to_au_feed   19437 non-null  float64
32  rougher.calculation.floatbank11_sulfate_to_au_feed   19437 non-null  float64
33  rougher.calculation.au_pb_ratio                      19439 non-null  float64
34  rougher.input.feed_ag                                19439 non-null  float64
35  rougher.input.feed_pb                                19339 non-null  float64
36  rougher.input.feed_rate                              19428 non-null  float64
37  rougher.input.feed_size                              19294 non-null  float64
38  rougher.input.feed_sol                               19340 non-null  float64
39  rougher.input.feed_au                                19439 non-null  float64
40  rougher.input.floatbank10_sulfate                    19405 non-null  float64
41  rougher.input.floatbank10_xanthate                   19431 non-null  float64
42  rougher.input.floatbank11_sulfate                    19395 non-null  float64
43  rougher.input.floatbank11_xanthate                   18986 non-null  float64
44  rougher.output.concentrate_ag                        19439 non-null  float64
45  rougher.output.concentrate_pb                        19439 non-null  float64
46  rougher.output.concentrate_sol                       19416 non-null  float64
47  rougher.output.concentrate_au                        19439 non-null  float64
48  rougher.output.recovery                              19439 non-null  float64
49  rougher.output.tail_ag                               19438 non-null  float64
50  rougher.output.tail_pb                               19439 non-null  float64
51  rougher.output.tail_sol                              19439 non-null  float64
52  rougher.output.tail_au                               19439 non-null  float64
53  rougher.state.floatbank10_a_air                      19438 non-null  float64
54  rougher.state.floatbank10_a_level                    19438 non-null  float64
55  rougher.state.floatbank10_b_air                      19438 non-null  float64
56  rougher.state.floatbank10_b_level                    19438 non-null  float64
57  rougher.state.floatbank10_c_air                      19438 non-null  float64
58  rougher.state.floatbank10_c_level                    19438 non-null  float64
59  rougher.state.floatbank10_d_air                      19439 non-null  float64
60  rougher.state.floatbank10_d_level                    19439 non-null  float64
61  rougher.state.floatbank10_e_air                      19003 non-null  float64
62  rougher.state.floatbank10_e_level                    19439 non-null  float64
63  rougher.state.floatbank10_f_air                      19439 non-null  float64
64  rougher.state.floatbank10_f_level                    19439 non-null  float64
65  secondary_cleaner.output.tail_ag                     19437 non-null  float64
66  secondary_cleaner.output.tail_pb                     19427 non-null  float64
67  secondary_cleaner.output.tail_sol                    17691 non-null  float64
68  secondary_cleaner.output.tail_au                     19439 non-null  float64
69  secondary_cleaner.state.floatbank2_a_air             19219 non-null  float64
70  secondary_cleaner.state.floatbank2_a_level           19438 non-null  float64
71  secondary_cleaner.state.floatbank2_b_air             19416 non-null  float64
72  secondary_cleaner.state.floatbank2_b_level           19438 non-null  float64
73  secondary_cleaner.state.floatbank3_a_air             19426 non-null  float64
74  secondary_cleaner.state.floatbank3_a_level           19438 non-null  float64
75  secondary_cleaner.state.floatbank3_b_air             19438 non-null  float64
76  secondary_cleaner.state.floatbank3_b_level           19438 non-null  float64
77  secondary_cleaner.state.floatbank4_a_air             19433 non-null  float64
78  secondary_cleaner.state.floatbank4_a_level           19438 non-null  float64
79  secondary_cleaner.state.floatbank4_b_air             19438 non-null  float64
80  secondary_cleaner.state.floatbank4_b_level           19438 non-null  float64
```

```
 81  secondary_cleaner.state.floatbank5_a_air            19438 non-null  float64
 82  secondary_cleaner.state.floatbank5_a_level          19438 non-null  float64
 83  secondary_cleaner.state.floatbank5_b_air            19438 non-null  float64
 84  secondary_cleaner.state.floatbank5_b_level          19438 non-null  float64
 85  secondary_cleaner.state.floatbank6_a_air            19437 non-null  float64
 86  secondary_cleaner.state.floatbank6_a_level          19438 non-null  float64
dtypes: float64(86), object(1)
memory usage: 12.9+ MB
```

Вывод

В тестовой выборке отсутствуют параметры `output` (параметры продукта) и `calculation` (расчётные характеристики)

Предобработка данных

In [21]:
```
df_test = df_test.fillna(method='ffill')
df_train = df_train.fillna(method='ffill')
df_full = df_full.fillna(method='ffill')
```

In [22]:
```
df_test = df_test.set_index('date')
df_train = df_train.set_index('date')
df_full = df_full.set_index('date')
```

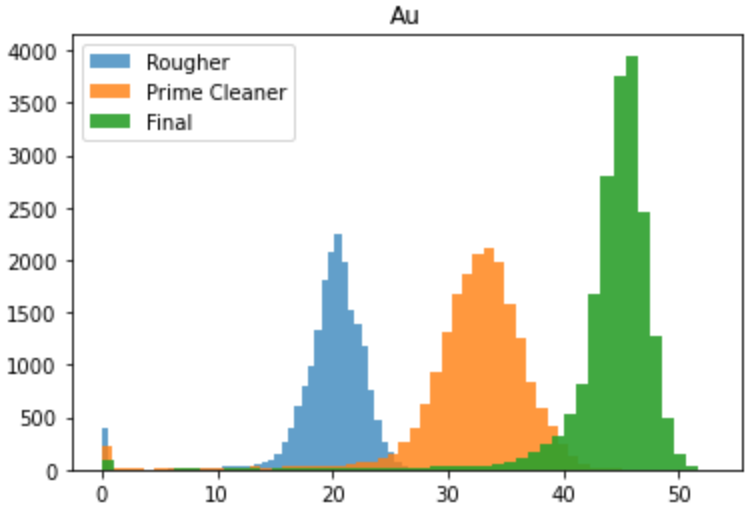
In [23]:
```
rougher_recovery_test = pd.Series(df_full['rougher.output.recovery'], index=df_test.index)
final_recovery_test = pd.Series(df_full['final.output.recovery'], index=df_test.index)

df_test['rougher.output.recovery'] = rougher_recovery_test
df_test['final.output.recovery'] = final_recovery_test
```

Вывод

- Так-как данные которые находятся рядом(по времени) не сильно отличаются, то решил заполнить пропуски значениями из соседних строк
- Добавил в тестовую выборку целевые признаки

Анализ данных

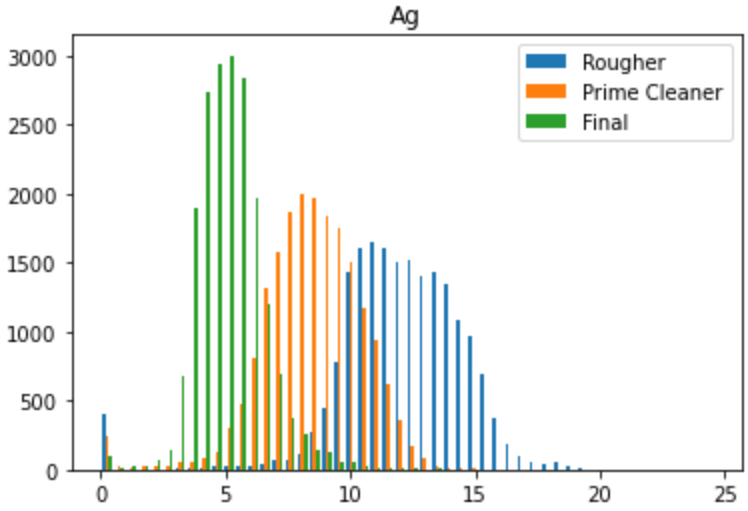
In [24]:
```
rougher_au = df_full['rougher.output.concentrate_au']
plt.hist(rougher_au,  bins=50, alpha=0.7)

prime_clean_au = df_full['primary_cleaner.output.concentrate_au']
plt.hist(prime_clean_au,  bins=50, alpha=0.8)

final_au = df_full['final.output.concentrate_au']
plt.hist(final_au, bins=50, alpha=0.9)
plt.title('Au')
plt.legend(['Rougher', 'Prime Cleaner', 'Final'])
plt.show()
```



Вывод

Концетрация золота увеличивается на каждом этапе отчистки

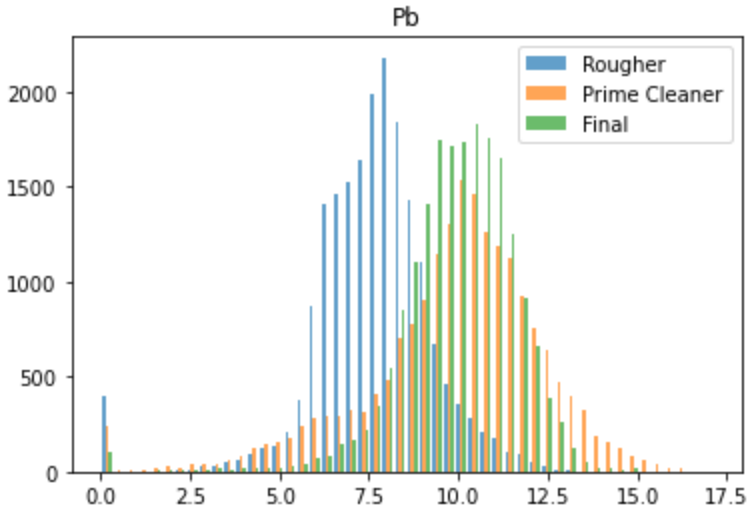
In [25]:
```
rougher_ag = df_full['rougher.output.concentrate_ag']
prime_clean_ag = df_full['primary_cleaner.output.concentrate_ag']
final_ag = df_full['final.output.concentrate_ag']
plt.hist([rougher_ag, prime_clean_ag, final_ag], bins=50)
plt.title('Ag')
plt.legend(['Rougher', 'Prime Cleaner', 'Final'])
plt.show()
```



Вывод

Концетрация серебра падает переходя с флотации на первичную отчистку и примерно такая же на финальной отчистке

In [26]:
```
rougher_pb = df_full['rougher.output.concentrate_pb']
prime_clean_pb = df_full['primary_cleaner.output.concentrate_pb']
final_pb = df_full['final.output.concentrate_pb']

plt.hist([rougher_pb, prime_clean_pb, final_pb], bins=50, alpha=0.7)
plt.title('Pb')
plt.legend(['Rougher', 'Prime Cleaner', 'Final'])
plt.show()
```

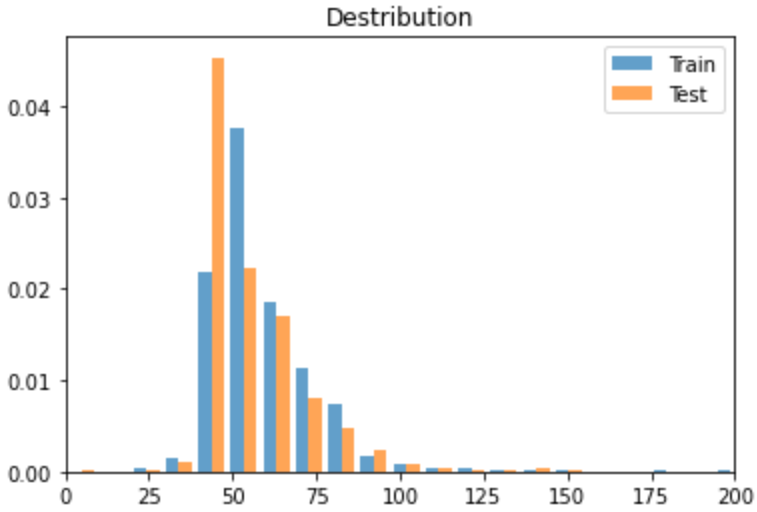


Вывод

Концетрация свинца переходя с флотации на первичную отчистку возрастает и остается примерно такой же на финальной отчистке

Сравнение распределения размеров гранул сырья на обучающей и тестовой выборке

In [27]:
```
feed_size_train = df_train['rougher.input.feed_size']
feed_size_test = df_test['rougher.input.feed_size']

plt.hist([feed_size_train, feed_size_test], density=True, bins=50, alpha=0.7)
plt.title('Destribution')
plt.legend(['Train', 'Test'])
plt.xlim(0, 200)
plt.show()
```



Вывод

Распределение на двух выборка примерно одинаковое

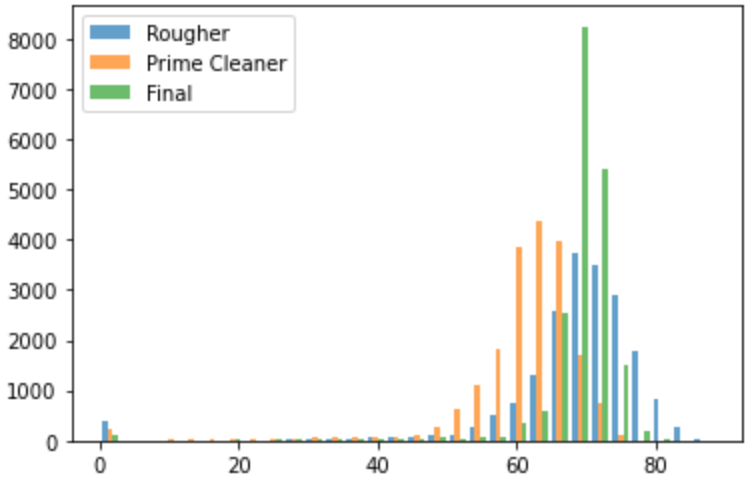
Суммарная концетрация всех веществ на разных стадиях

In [28]:
```
rougher = (df_full[['rougher.output.concentrate_au',
                    'rougher.output.concentrate_pb',
                    'rougher.output.concentrate_sol',
                    'rougher.output.concentrate_ag']])
```

```
prime_clean = df_full[['primary_cleaner.output.concentrate_au',
                       'primary_cleaner.output.concentrate_pb',
                       'primary_cleaner.output.concentrate_sol',
                       'primary_cleaner.output.concentrate_ag']]

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

In [29]:
```
rougher = rougher.apply(sum, axis=1)
prime = prime_clean.apply(sum, axis=1)
final = final.apply(sum, axis=1)
```

In [30]:
```
plt.hist([rougher, prime, final], bins=30, alpha=.7)
plt.legend(['Rougher', 'Prime Cleaner', 'Final'])
plt.show()
```



Модель

In [31]:
```
def smape(a, f):
    smape = 1/len(a) * np.sum(2 * np.abs(f-a) / (np.abs(a) + np.abs(f)) * 100)
    final_smape = 0.25 * smape[0] + 0.75 * smape[1]
    return final_smape
```

In [32]:
```
smape_score = make_scorer(smape, greater_is_better=False)
```

In [33]:
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 14149 entries, 2016-01-15 00:00:00 to 2018-08-18 10:59:59
Data columns (total 86 columns):
 #   Column                                              Non-Null Count  Dtype  
---  ------                                              --------------  -----  
 0   final.output.concentrate_ag                         14149 non-null  float64
 1   final.output.concentrate_pb                         14149 non-null  float64
 2   final.output.concentrate_sol                        14149 non-null  float64
 3   final.output.concentrate_au                         14149 non-null  float64
 4   final.output.recovery                               14149 non-null  float64
 5   final.output.tail_ag                                14149 non-null  float64
```

6	final.output.tail_pb	14149	non-null	float64
7	final.output.tail_sol	14149	non-null	float64
8	final.output.tail_au	14149	non-null	float64
9	primary_cleaner.input.sulfate	14149	non-null	float64
10	primary_cleaner.input.depressant	14149	non-null	float64
11	primary_cleaner.input.feed_size	14149	non-null	float64
12	primary_cleaner.input.xanthate	14149	non-null	float64
13	primary_cleaner.output.concentrate_ag	14149	non-null	float64
14	primary_cleaner.output.concentrate_pb	14149	non-null	float64
15	primary_cleaner.output.concentrate_sol	14149	non-null	float64
16	primary_cleaner.output.concentrate_au	14149	non-null	float64
17	primary_cleaner.output.tail_ag	14149	non-null	float64
18	primary_cleaner.output.tail_pb	14149	non-null	float64
19	primary_cleaner.output.tail_sol	14149	non-null	float64
20	primary_cleaner.output.tail_au	14149	non-null	float64
21	primary_cleaner.state.floatbank8_a_air	14149	non-null	float64
22	primary_cleaner.state.floatbank8_a_level	14149	non-null	float64
23	primary_cleaner.state.floatbank8_b_air	14149	non-null	float64
24	primary_cleaner.state.floatbank8_b_level	14149	non-null	float64
25	primary_cleaner.state.floatbank8_c_air	14149	non-null	float64
26	primary_cleaner.state.floatbank8_c_level	14149	non-null	float64
27	primary_cleaner.state.floatbank8_d_air	14149	non-null	float64
28	primary_cleaner.state.floatbank8_d_level	14149	non-null	float64
29	rougher.calculation.sulfate_to_au_concentrate	14149	non-null	float64
30	rougher.calculation.floatbank10_sulfate_to_au_feed	14149	non-null	float64
31	rougher.calculation.floatbank11_sulfate_to_au_feed	14149	non-null	float64
32	rougher.calculation.au_pb_ratio	14149	non-null	float64
33	rougher.input.feed_ag	14149	non-null	float64
34	rougher.input.feed_pb	14149	non-null	float64
35	rougher.input.feed_rate	14149	non-null	float64
36	rougher.input.feed_size	14149	non-null	float64
37	rougher.input.feed_sol	14149	non-null	float64
38	rougher.input.feed_au	14149	non-null	float64
39	rougher.input.floatbank10_sulfate	14149	non-null	float64
40	rougher.input.floatbank10_xanthate	14149	non-null	float64
41	rougher.input.floatbank11_sulfate	14149	non-null	float64
42	rougher.input.floatbank11_xanthate	14149	non-null	float64
43	rougher.output.concentrate_ag	14149	non-null	float64
44	rougher.output.concentrate_pb	14149	non-null	float64
45	rougher.output.concentrate_sol	14149	non-null	float64
46	rougher.output.concentrate_au	14149	non-null	float64
47	rougher.output.recovery	14149	non-null	float64
48	rougher.output.tail_ag	14149	non-null	float64
49	rougher.output.tail_pb	14149	non-null	float64
50	rougher.output.tail_sol	14149	non-null	float64
51	rougher.output.tail_au	14149	non-null	float64
52	rougher.state.floatbank10_a_air	14149	non-null	float64
53	rougher.state.floatbank10_a_level	14149	non-null	float64
54	rougher.state.floatbank10_b_air	14149	non-null	float64
55	rougher.state.floatbank10_b_level	14149	non-null	float64
56	rougher.state.floatbank10_c_air	14149	non-null	float64
57	rougher.state.floatbank10_c_level	14149	non-null	float64
58	rougher.state.floatbank10_d_air	14149	non-null	float64
59	rougher.state.floatbank10_d_level	14149	non-null	float64
60	rougher.state.floatbank10_e_air	14149	non-null	float64
61	rougher.state.floatbank10_e_level	14149	non-null	float64
62	rougher.state.floatbank10_f_air	14149	non-null	float64
63	rougher.state.floatbank10_f_level	14149	non-null	float64
64	secondary_cleaner.output.tail_ag	14149	non-null	float64
65	secondary_cleaner.output.tail_pb	14149	non-null	float64
66	secondary_cleaner.output.tail_sol	14149	non-null	float64
67	secondary_cleaner.output.tail_au	14149	non-null	float64
68	secondary_cleaner.state.floatbank2_a_air	14149	non-null	float64
69	secondary_cleaner.state.floatbank2_a_level	14149	non-null	float64
70	secondary_cleaner.state.floatbank2_b_air	14149	non-null	float64
71	secondary_cleaner.state.floatbank2_b_level	14149	non-null	float64

```
 72  secondary_cleaner.state.floatbank3_a_air            14149 non-null  float64
 73  secondary_cleaner.state.floatbank3_a_level          14149 non-null  float64
 74  secondary_cleaner.state.floatbank3_b_air            14149 non-null  float64
 75  secondary_cleaner.state.floatbank3_b_level          14149 non-null  float64
 76  secondary_cleaner.state.floatbank4_a_air            14149 non-null  float64
 77  secondary_cleaner.state.floatbank4_a_level          14149 non-null  float64
 78  secondary_cleaner.state.floatbank4_b_air            14149 non-null  float64
 79  secondary_cleaner.state.floatbank4_b_level          14149 non-null  float64
 80  secondary_cleaner.state.floatbank5_a_air            14149 non-null  float64
 81  secondary_cleaner.state.floatbank5_a_level          14149 non-null  float64
 82  secondary_cleaner.state.floatbank5_b_air            14149 non-null  float64
 83  secondary_cleaner.state.floatbank5_b_level          14149 non-null  float64
 84  secondary_cleaner.state.floatbank6_a_air            14149 non-null  float64
 85  secondary_cleaner.state.floatbank6_a_level          14149 non-null  float64
dtypes: float64(86)
memory usage: 9.4+ MB
```

In [34]:
```
features = df_train.drop(['rougher.output.recovery', 'final.output.recovery'], axis=1)
target = df_train[['rougher.output.recovery', 'final.output.recovery']]
```

In [35]:
```
features_train, features_valid, target_train, target_valid = train_test_split(
    features, target, test_size=0.20, random_state=12345
)
```

DecisionTreeRegressor

In [36]:
```
grid_tree = {
    'max_depth': list(range(1, 20))
}

model_dtr = DecisionTreeRegressor(random_state=12345)

grid_search = GridSearchCV(model_dtr, grid_tree, cv=5, scoring=smape_score)

grid_search.fit(features_train, target_train)
```

Out[36]:
```
GridSearchCV(cv=5, estimator=DecisionTreeRegressor(random_state=12345),
             param_grid={'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
                                       13, 14, 15, 16, 17, 18, 19]},
             scoring=make_scorer(smape, greater_is_better=False))
```

In [37]:
```
grid_search.best_params_
```

Out[37]:
```
{'max_depth': 18}
```

In [38]:
```
model_dtr = DecisionTreeRegressor(random_state=12345, max_depth=18)
model_dtr.fit(features_train, target_train)
predicted_dtr = model_dtr.predict(features_valid)
```

In [39]:
```
smape(target_valid, predicted_dtr)
```

Out[39]:
```
2.6393106843244203
```

In [40]:
```
score_dtr = cross_val_score(model_dtr, features, target)
print(sum(score_dtr) / len(score_dtr))
```

```
0.7012566763256516
```

RandomForestRegressor

In [41]:
```
#grid_forest = {'n_estimators': list(range(1, 11, 2)),

#              'max_depth': list(range(1, 11, 2))}
#model_rfr = RandomForestRegressor(random_state=12345)

#grid_search = GridSearchCV(model_rfr, grid_forest, cv=5, scoring=smape_score)

#grid_search.fit(features_train, target_train)
```

In [42]:
```
#grid_search.best_params_
```

In [43]:
```
%%time
best_smape = 0
best_est = 0
best_depth = 0


for est in range(10, 51, 10):
    for depth in range(1, 11):
        model = RandomForestRegressor(random_state=12345, n_estimators=est, max_depth=dept
        model.fit(features_train, target_train)
        predictions_valid = model.predict(features_valid)
        smape_rfr = smape(target_valid, predictions_valid)
        if smape_rfr < best_smape:
            best_smape = smape_rfr
            best_est = est
            best_depth = depth

print('est:', est, 'max_depth:', depth, 'SMAPE:', smape_rfr)
```

```
est: 50 max_depth: 10 SMAPE: 3.0792673006316553
```

LinearRegression

In [44]:
```
model_lreg = LinearRegression()
model_lreg.fit(features_train, target_train)
predicted_lreg = model_lreg.predict(features_valid)
```

In [45]:
```
smape_lreg = smape(target_valid, predicted_lreg)
```

In [46]:
```
print('Final sMAPE:', smape_lreg)

print()

score_lreg = cross_val_score(model_lreg, features, target, scoring=smape_score)
print(score_lreg)
```

```
Final sMAPE: 4.642843825575602

[-5.68270836 -5.40862852 -8.44645366 -7.54842327 -5.61390862]
```

Общий вывод

Модель случайного леса показывает лучшие результаты

Проверка модели на тестовой выборке

In [47]:
```
features_train_test = features_train[df_test.columns.drop(['final.output.recovery', 'rough
```

Питон ругался что кол-во колонок не совпадает. Я додумался только до такого решения. Либо я не правильно понял задачу и надо обучить модель так же на тестовой выборке

In [48]:
```
target_test = df_test[['rougher.output.recovery', 'final.output.recovery']]
```

In [49]:
```
target_test.shape
```

Out[49]:
```
(5290, 2)
```

In [50]:
```
features_train_test.shape
```

Out[50]:
```
(11319, 52)
```

In [51]:
```
model_test = RandomForestRegressor(random_state=12345, n_estimators=30, max_depth=10)
model_test.fit(features_train_test, target_train)
```

Out[51]:
```
RandomForestRegressor(max_depth=10, n_estimators=30, random_state=12345)
```

In [52]:
```
features_test = df_test.drop(['final.output.recovery', 'rougher.output.recovery'], axis=1)
```

In [53]:
```
predictions_test = model_test.predict(features_test)
```

In [54]:
```
smape_rfr_test = smape(target_test, predictions_test)

print('Final sMAPE:', smape_rfr_test)
```

```
Final sMAPE: 10.028032578881785
```

Общий вывод

Подготовка данных

- Заполнил пропуски в строках значениями из соседних строк.
- Использвал столбец с датами как индекс.
- Добавил в тестовую выборку целевые признаки из полного датасета.

Расчет эффективности обогащения

- Расчет в "ручную" показал, что изначальные значения в данных корректны.

Анализ данных

- Концетрация золота увеличивается на каждом этапе отчистки.
- Концетрация серебра падает переходя с флотации на первичную отчистку и примерно такая же на финальной отчистке.

- Концетрация свинца переходя с флотации на первичную отчистку возрастает и остается примерно такой же на финальной отчистке.

Сравнение распределения размеров гранул сырья на обучающей и тестовой выборке

- На тестовой и обучающей выборке распределение размеров гранул примерно одинаковое.

Суммарная концетрация всех веществ на разных стадиях

- Суммарная концетрация возрастает переходя с флотации до финальной отчистки.

Проверка и выбор моделей

- Выбрал для этого проекта модели дерева решений, случайного леса и ленейной регрессии. Из них лучшие результаты показывает модель случайного леса.

Чек-лист готовности проекта

- [x] Jupyter Notebook открыт
- [x] Весь код выполняется без ошибок
- [x] Ячейки с кодом расположены в порядке выполнения
- [x] Выполнен шаг 1: данные подготовлены
  - [x] Проверена формула вычисления эффективности обогащения
  - [x] Проанализированы признаки, недоступные в тестовой выборке
  - [x] Проведена предобработка данных
- [x] Выполнен шаг 2: данные проанализированы
  - [x] Исследовано изменение концентрации элементов на каждом этапе
  - [x] Проанализированы распределения размеров гранул на обучающей и тестовой выборках
  - [x] Исследованы суммарные концентрации
- [x] Выполнен шаг 3: построена модель прогнозирования
  - [x] Написана функция для вычисления итогового *sMAPE*
  - [x] Обучено и проверено несколько моделей
  - [x] Выбрана лучшая модель, её качество проверено на тестовой выборке