

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt
```

```
In [2]: mining_data=pd.read_csv(r'Mining_Dataset.csv', decimal=',',parse_dates=[ "date"],i
```

```
In [3]: mining_data.head()
```

Out[3]:

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Flo Co C
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	...	25
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	...	24
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	...	24
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	...	25
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	...	24

5 rows × 24 columns



In [4]: `mining_data.head(15)`

Out[4]:

	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Fl C
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	...	2
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	...	2
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	...	2
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	...	2
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	...	2
5	2017-03-10 01:00:00	55.2	16.98	3079.10	564.697	396.533	10.0705	1.74	250.730	248.906	...	2
6	2017-03-10 01:00:00	55.2	16.98	3127.79	566.467	392.900	10.0713	1.74	250.313	252.202	...	2
7	2017-03-10 01:00:00	55.2	16.98	3152.93	558.777	397.002	10.0722	1.74	249.895	253.630	...	2
8	2017-03-10 01:00:00	55.2	16.98	3147.27	556.030	394.307	10.0730	1.74	250.137	251.104	...	2
9	2017-03-10 01:00:00	55.2	16.98	3142.58	565.857	393.105	10.0738	1.74	249.653	252.202	...	2
10	2017-03-10 01:00:00	55.2	16.98	3148.05	561.951	396.533	10.0746	1.74	249.236	250.818	...	2
11	2017-03-10 01:00:00	55.2	16.98	3150.39	558.472	397.852	10.0755	1.74	249.170	249.829	...	2
12	2017-03-10 01:00:00	55.2	16.98	3280.27	564.026	393.545	10.0763	1.74	249.016	249.829	...	2
13	2017-03-10 01:00:00	55.2	16.98	3411.13	567.261	394.160	10.0771	1.74	249.258	250.137	...	2
14	2017-03-10 01:00:00	55.2	16.98	3447.46	561.646	392.549	10.0779	1.74	249.390	251.191	...	2

15 rows × 24 columns

In [5]: `mining_data.describe()`

Out[5]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH
<b>count</b>	737453.000000	737453.000000	737453.000000	737453.000000	737453.000000	737453.000000
<b>mean</b>	56.294739	14.651716	2869.140569	488.144697	397.578372	9.767639
<b>std</b>	5.157744	6.807439	1215.203734	91.230534	9.699785	0.387007
<b>min</b>	42.740000	1.310000	0.002026	241.669000	376.249000	8.753340
<b>25%</b>	52.670000	8.940000	2076.320000	431.796000	394.264000	9.527360
<b>50%</b>	56.080000	13.850000	3018.430000	504.393000	399.249000	9.798100
<b>75%</b>	59.720000	19.600000	3727.730000	553.257000	402.968000	10.038000
<b>max</b>	65.780000	33.400000	6300.230000	739.538000	418.641000	10.808100

8 rows × 23 columns

In [6]: `mining_data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 737453 entries, 0 to 737452
Data columns (total 24 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   date             737453 non-null   datetime64[ns]
 1   % Iron Feed      737453 non-null   float64
 2   % Silica Feed    737453 non-null   float64
 3   Starch Flow      737453 non-null   float64
 4   Amina Flow       737453 non-null   float64
 5   Ore Pulp Flow    737453 non-null   float64
 6   Ore Pulp pH      737453 non-null   float64
 7   Ore Pulp Density 737453 non-null   float64
 8   Flotation Column 01 Air Flow 737453 non-null   float64
 9   Flotation Column 02 Air Flow 737453 non-null   float64
 10  Flotation Column 03 Air Flow 737453 non-null   float64
 11  Flotation Column 04 Air Flow 737453 non-null   float64
 12  Flotation Column 05 Air Flow 737453 non-null   float64
 13  Flotation Column 06 Air Flow 737453 non-null   float64
 14  Flotation Column 07 Air Flow 737453 non-null   float64
 15  Flotation Column 01 Level   737453 non-null   float64
 16  Flotation Column 02 Level   737453 non-null   float64
 17  Flotation Column 03 Level   737453 non-null   float64
 18  Flotation Column 04 Level   737453 non-null   float64
 19  Flotation Column 05 Level   737453 non-null   float64
 20  Flotation Column 06 Level   737453 non-null   float64
 21  Flotation Column 07 Level   737453 non-null   float64
 22  % Iron Concentrate 737453 non-null   float64
 23  % Silica Concentrate 737453 non-null   float64
dtypes: datetime64[ns](1), float64(23)
memory usage: 135.0 MB
```

```
In [7]: for cols in mining_data.columns.tolist()[1:]:
    df = mining_data.loc[mining_data[cols] > 0]
df.info()
```

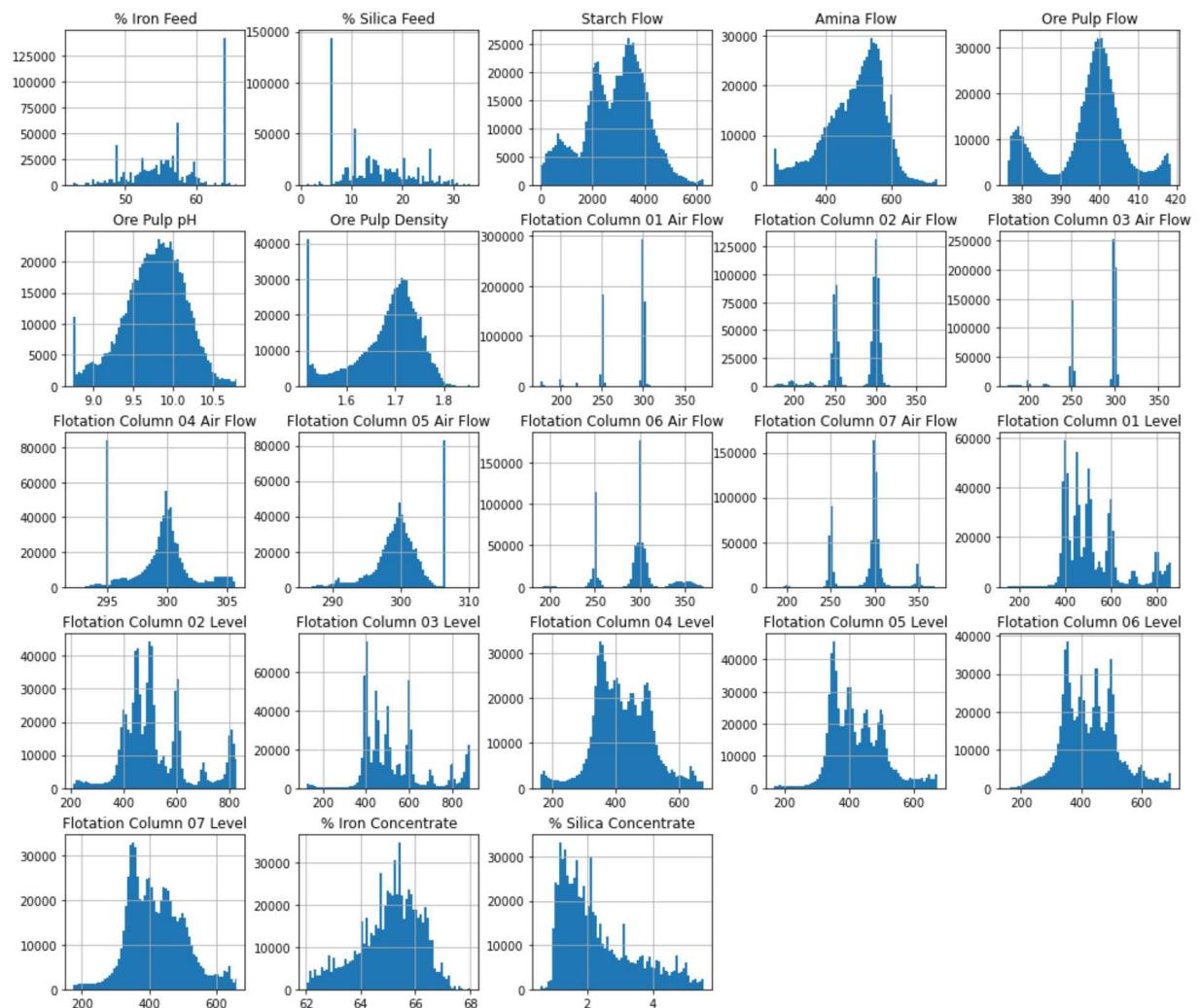
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 737453 entries, 0 to 737452
Data columns (total 24 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   date             737453 non-null   datetime64[ns]
 1   % Iron Feed     737453 non-null   float64
 2   % Silica Feed   737453 non-null   float64
 3   Starch Flow     737453 non-null   float64
 4   Amina Flow      737453 non-null   float64
 5   Ore Pulp Flow   737453 non-null   float64
 6   Ore Pulp pH     737453 non-null   float64
 7   Ore Pulp Density 737453 non-null   float64
 8   Flotation Column 01 Air Flow 737453 non-null   float64
 9   Flotation Column 02 Air Flow 737453 non-null   float64
 10  Flotation Column 03 Air Flow 737453 non-null   float64
 11  Flotation Column 04 Air Flow 737453 non-null   float64
 12  Flotation Column 05 Air Flow 737453 non-null   float64
 13  Flotation Column 06 Air Flow 737453 non-null   float64
 14  Flotation Column 07 Air Flow 737453 non-null   float64
 15  Flotation Column 01 Level   737453 non-null   float64
 16  Flotation Column 02 Level   737453 non-null   float64
 17  Flotation Column 03 Level   737453 non-null   float64
 18  Flotation Column 04 Level   737453 non-null   float64
 19  Flotation Column 05 Level   737453 non-null   float64
 20  Flotation Column 06 Level   737453 non-null   float64
 21  Flotation Column 07 Level   737453 non-null   float64
 22  % Iron Concentrate 737453 non-null   float64
 23  % Silica Concentrate 737453 non-null   float64
dtypes: datetime64[ns](1), float64(23)
memory usage: 140.7 MB
```

```
In [8]: mining_data.isnull().any()
```

```
Out[8]: date                False
% Iron Feed            False
% Silica Feed           False
Starch Flow             False
Amina Flow              False
Ore Pulp Flow           False
Ore Pulp pH              False
Ore Pulp Density         False
Flotation Column 01 Air Flow  False
Flotation Column 02 Air Flow  False
Flotation Column 03 Air Flow  False
Flotation Column 04 Air Flow  False
Flotation Column 05 Air Flow  False
Flotation Column 06 Air Flow  False
Flotation Column 07 Air Flow  False
Flotation Column 01 Level    False
Flotation Column 02 Level    False
Flotation Column 03 Level    False
Flotation Column 04 Level    False
Flotation Column 05 Level    False
Flotation Column 06 Level    False
Flotation Column 07 Level    False
% Iron Concentrate       False
% Silica Concentrate      False
dtype: bool
```

```
In [9]: df=df.set_index('date')
```

```
In [10]: import matplotlib.pyplot as plt
from matplotlib import style
df.hist(bins = 70, figsize = (17,15))
plt.show()
plt.suptitle('figure title', color='green')
```



Out[10]: Text(0.5, 0.98, 'figure title')

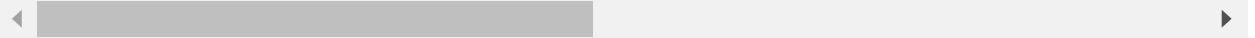
<Figure size 432x288 with 0 Axes>

In [11]: df.head()

Out[11]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow
date										
2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	250.576
2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	250.862
2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	250.313
2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	250.049
2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	249.895

5 rows × 23 columns

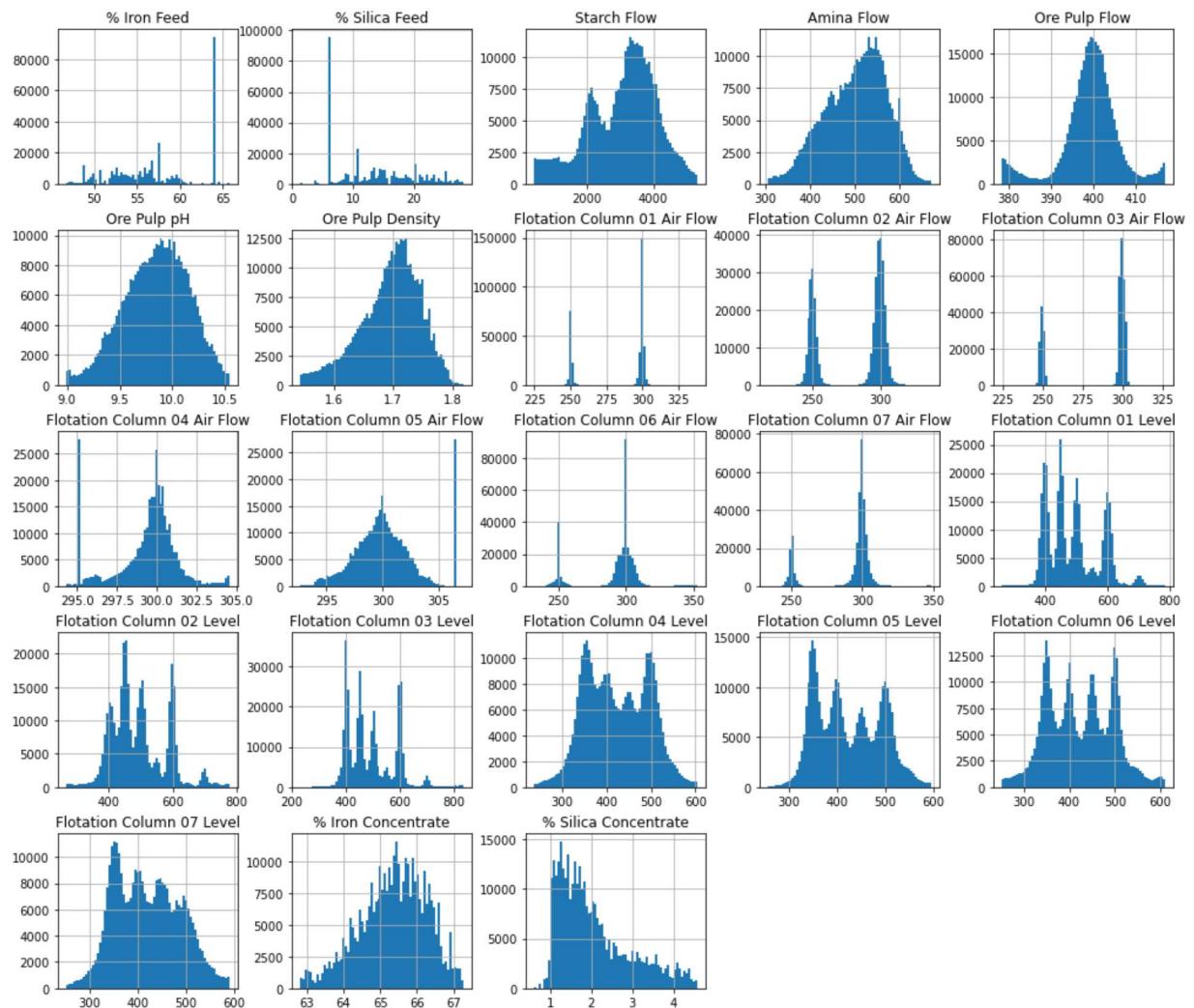


In [12]: df\_copy=df

```
from scipy import stats
df_copy=df[(np.abs(stats.zscore(df)) < 2).all(axis=1)]
df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 346475 entries, 2017-03-10 01:00:00 to 2017-09-09 22:00:00
Data columns (total 23 columns):
 #   Column           Non-Null Count   Dtype  
--- 
 0   % Iron Feed     346475 non-null    float64
 1   % Silica Feed   346475 non-null    float64
 2   Starch Flow     346475 non-null    float64
 3   Amina Flow      346475 non-null    float64
 4   Ore Pulp Flow   346475 non-null    float64
 5   Ore Pulp pH     346475 non-null    float64
 6   Ore Pulp Density 346475 non-null    float64
 7   Flotation Column 01 Air Flow 346475 non-null    float64
 8   Flotation Column 02 Air Flow 346475 non-null    float64
 9   Flotation Column 03 Air Flow 346475 non-null    float64
 10  Flotation Column 04 Air Flow 346475 non-null    float64
 11  Flotation Column 05 Air Flow 346475 non-null    float64
 12  Flotation Column 06 Air Flow 346475 non-null    float64
 13  Flotation Column 07 Air Flow 346475 non-null    float64
 14  Flotation Column 01 Level   346475 non-null    float64
 15  Flotation Column 02 Level   346475 non-null    float64
 16  Flotation Column 03 Level   346475 non-null    float64
 17  Flotation Column 04 Level   346475 non-null    float64
 18  Flotation Column 05 Level   346475 non-null    float64
 19  Flotation Column 06 Level   346475 non-null    float64
 20  Flotation Column 07 Level   346475 non-null    float64
 21  % Iron Concentrate 346475 non-null    float64
 22  % Silica Concentrate 346475 non-null    float64
dtypes: float64(23)
memory usage: 63.4 MB
```

```
In [13]: df_copy.hist(bins = 70, figsize = (17,15))
plt.show()
plt.suptitle('figure title', color='w')
df_copy.info()
```



```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 346475 entries, 2017-03-10 01:00:00 to 2017-09-09 22:00:00
Data columns (total 23 columns):
 #   Column           Non-Null Count   Dtype  
--- 
 0   % Iron Feed      346475 non-null    float64
 1   % Silica Feed    346475 non-null    float64
 2   Starch Flow      346475 non-null    float64
 3   Amina Flow       346475 non-null    float64
 4   Ore Pulp Flow    346475 non-null    float64
 5   Ore Pulp pH      346475 non-null    float64
 6   Ore Pulp Density 346475 non-null    float64
 7   Flotation Column 01 Air Flow 346475 non-null    float64
 8   Flotation Column 02 Air Flow 346475 non-null    float64
 9   Flotation Column 03 Air Flow 346475 non-null    float64
 10  Flotation Column 04 Air Flow 346475 non-null    float64
 11  Flotation Column 05 Air Flow 346475 non-null    float64
 12  Flotation Column 06 Air Flow 346475 non-null    float64
 13  Flotation Column 07 Air Flow 346475 non-null    float64
 14  Flotation Column 01 Level   346475 non-null    float64
 15  Flotation Column 02 Level   346475 non-null    float64
 16  Flotation Column 03 Level   346475 non-null    float64
 17  Flotation Column 04 Level   346475 non-null    float64
 18  Flotation Column 05 Level   346475 non-null    float64
 19  Flotation Column 06 Level   346475 non-null    float64
 20  Flotation Column 07 Level   346475 non-null    float64
 21  % Iron Concentrate 346475 non-null    float64
 22  % Silica Concentrate 346475 non-null    float64
dtypes: float64(23)
memory usage: 63.4 MB
```

```
<Figure size 432x288 with 0 Axes>
```

```
In [14]: df_copy.drop(columns=['% Iron Concentrate'], inplace=True)
df_copy.head()
```

C:\Users\AGALawe YOGESH\anaconda3\lib\site-packages\pandas\core\frame.py:4308:  
SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
    return super().drop(
```

Out[14]:

	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	Flotation Column 03 Air Flow
date										
2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	250.576
2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	250.862
2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	250.313
2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	250.049
2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	249.895

5 rows × 22 columns



In [15]: `mining_data.head()`

Out[15]:

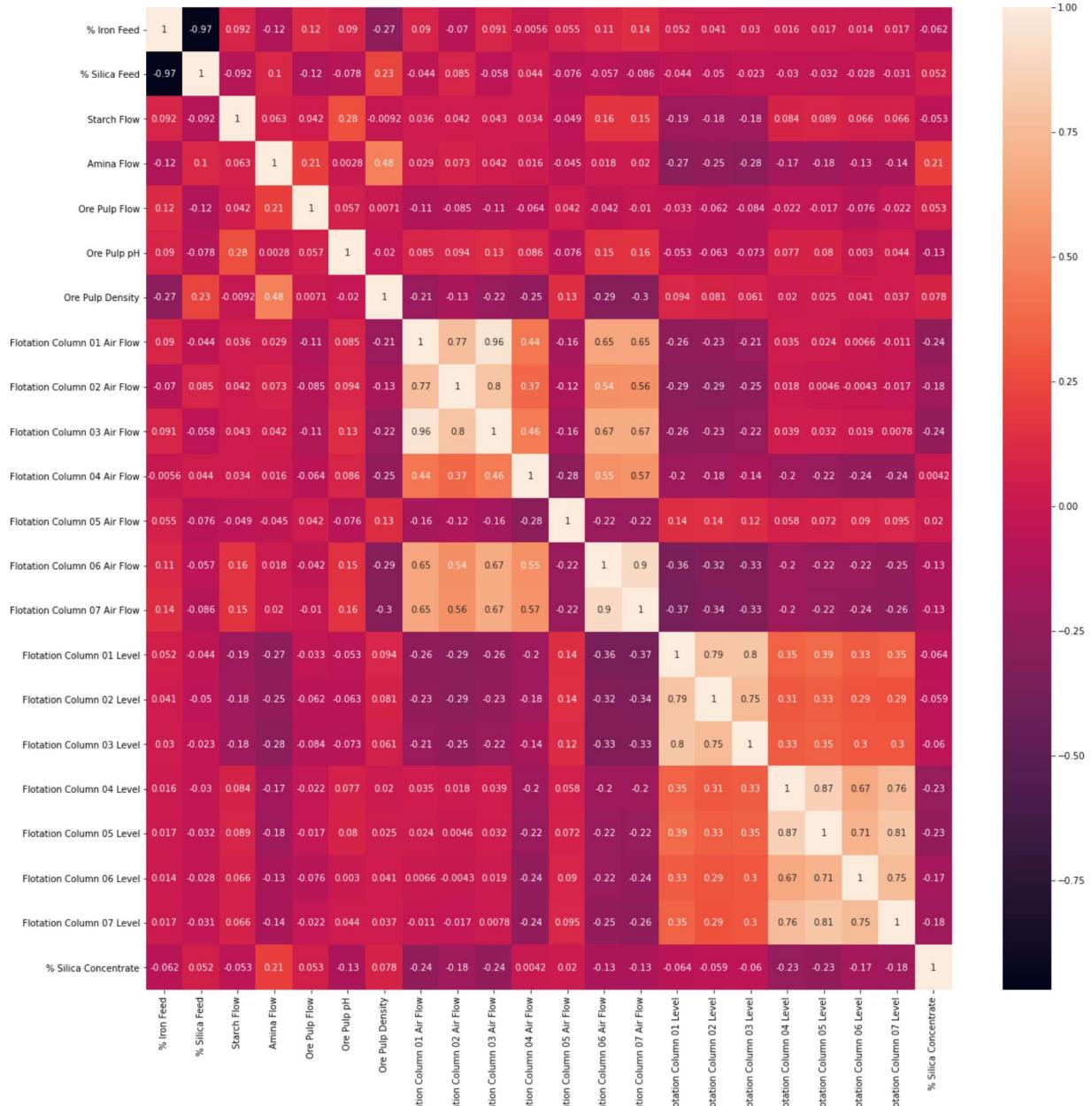
	date	% Iron Feed	% Silica Feed	Starch Flow	Amina Flow	Ore Pulp Flow	Ore Pulp pH	Ore Pulp Density	Flotation Column 01 Air Flow	Flotation Column 02 Air Flow	...	Flot Co C
0	2017-03-10 01:00:00	55.2	16.98	3019.53	557.434	395.713	10.0664	1.74	249.214	253.235	...	25
1	2017-03-10 01:00:00	55.2	16.98	3024.41	563.965	397.383	10.0672	1.74	249.719	250.532	...	24
2	2017-03-10 01:00:00	55.2	16.98	3043.46	568.054	399.668	10.0680	1.74	249.741	247.874	...	24
3	2017-03-10 01:00:00	55.2	16.98	3047.36	568.665	397.939	10.0689	1.74	249.917	254.487	...	25
4	2017-03-10 01:00:00	55.2	16.98	3033.69	558.167	400.254	10.0697	1.74	250.203	252.136	...	24

5 rows × 24 columns



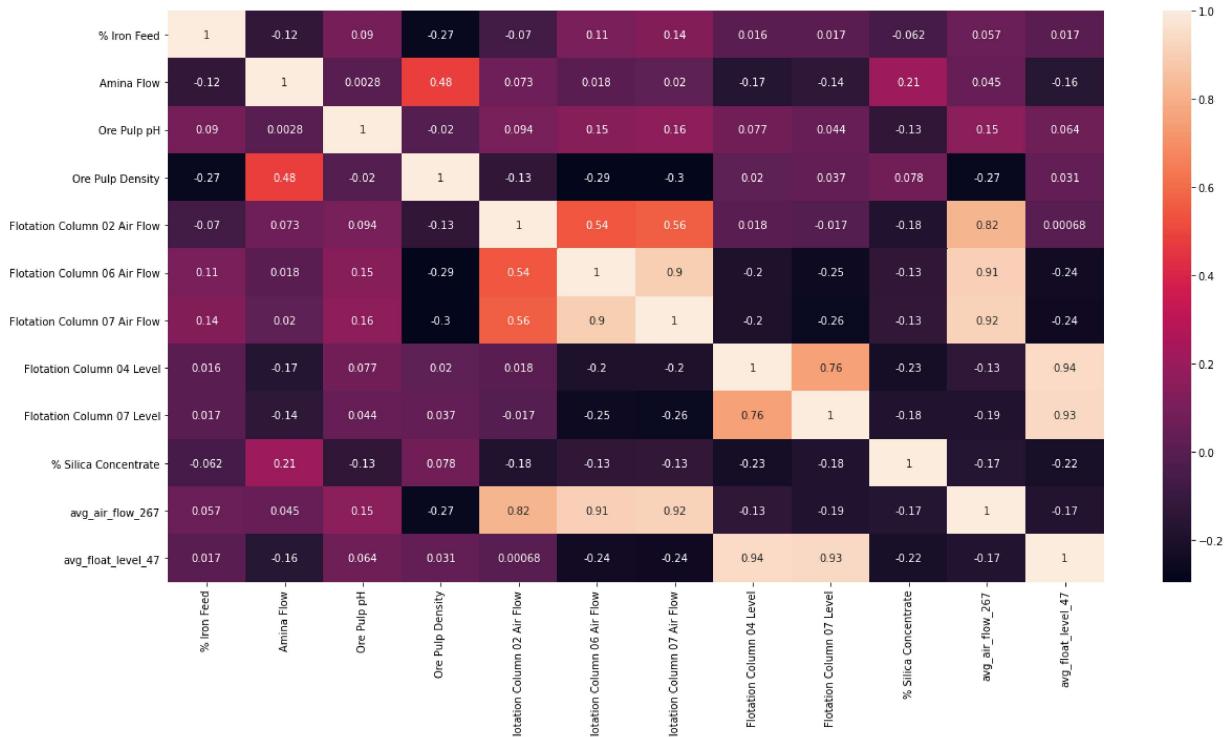
In [16]:

```
import seaborn as sns
ml_mining_data=df_copy
plt.figure(figsize=(20, 20))
p = sns.heatmap(ml_mining_data.corr(), annot=True);
```



```
In [17]: df=ml_mining_data.drop(['% Silica Feed', 'Starch Flow', 'Ore Pulp Flow','Flotation Column 01 Air Flow', 'Flotation Column 02 Air Flow', 'Flotation Column 03 Air Flow', 'Flotation Column 04 Air Flow', 'Flotation Column 05 Level', 'Flotation Column 06 Level', 'Flotation Column 07 Air Flow'], axis=1)
```

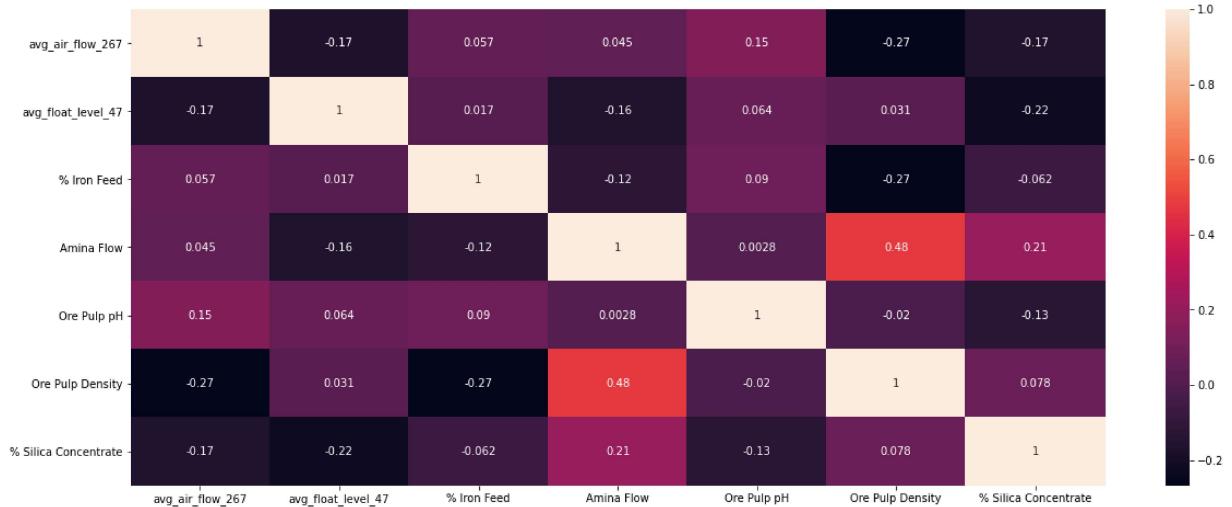
```
In [18]: df.head()
df['avg_air_flow_267'] = df[['Flotation Column 02 Air Flow','Flotation Column 06 Air Flow', 'Flotation Column 07 Air Flow']].mean(axis=1)
df['avg_float_level_47'] = df[['Flotation Column 04 Level', 'Flotation Column 07 Level']].mean(axis=1)
df.head()
plt.figure(figsize=(20, 10))
p = sns.heatmap(df.corr(), annot=True);
```

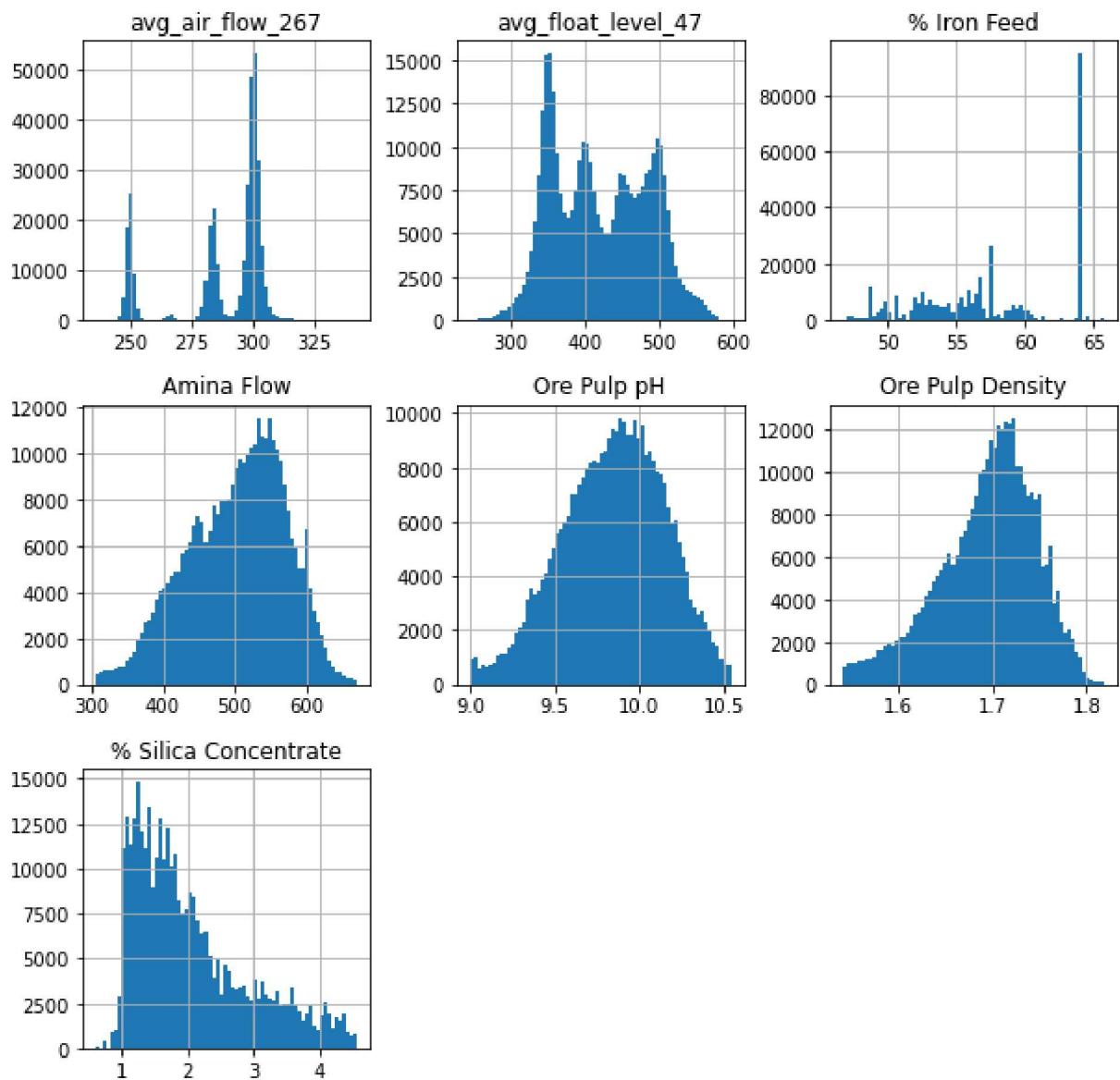


```
In [19]: df_ml=df.drop(['Flotation Column 02 Air Flow','Flotation Column 06 Air Flow','Flotation Column 04 Level', 'Flotation Column 07 Level'], axis = 1)
df_ml.head()
cols = df_ml.columns.tolist()
cols
cols = cols[-1:] + cols[:-1]
cols = cols[-1:] + cols[:-1]
df_ml = df_ml[cols]
df_ml.head()
df_ml.info()
plt.figure(figsize=(20, 8))
p = sns.heatmap(df_ml.corr(), annot=True);

df_ml.hist(bins = 70, figsize = (10,10))
plt.show()
plt.suptitle('figure title', color='w')
```

<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 346475 entries, 2017-03-10 01:00:00 to 2017-09-09 22:00:00  
Data columns (total 7 columns):  
# Column Non-Null Count Dtype  
---  
0 avg\_air\_flow\_267 346475 non-null float64  
1 avg\_float\_level\_47 346475 non-null float64  
2 % Iron Feed 346475 non-null float64  
3 Amina Flow 346475 non-null float64  
4 Ore Pulp pH 346475 non-null float64  
5 Ore Pulp Density 346475 non-null float64  
6 % Silica Concentrate 346475 non-null float64  
dtypes: float64(7)  
memory usage: 21.1 MB





Out[19]: Text(0.5, 0.98, 'figure title')

<Figure size 432x288 with 0 Axes>

In [20]: `import pandas as pd  
import numpy as np`

```
In [21]: dataset=pd.read_csv('Enhanced_Mining_dataset.csv',sep=',')
dataset=dataset.set_index('date')
dataset.head()
```

Out[21]:

	avg_air_flow_267	avg_float_level_47	% Iron Feed	Amina Flow	Ore Pulp pH	Ore Pulp Density	% Silica Concentrate
date							
2017-03-10 01:00:00	251.448000	483.4510	55.2	557.434	10.0664	1.74	1.31
2017-03-10 01:00:00	249.887667	473.0805	55.2	563.965	10.0672	1.74	1.31
2017-03-10 01:00:00	249.096667	454.1275	55.2	568.054	10.0680	1.74	1.31
2017-03-10 01:00:00	252.018667	436.9395	55.2	568.665	10.0689	1.74	1.31
2017-03-10 01:00:00	250.349000	439.6745	55.2	558.167	10.0697	1.74	1.31

```
In [22]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 346475 entries, 2017-03-10 01:00:00 to 2017-09-09 22:00:00
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   avg_air_flow_267    346475 non-null   float64
 1   avg_float_level_47  346475 non-null   float64
 2   % Iron Feed        346475 non-null   float64
 3   Amina Flow         346475 non-null   float64
 4   Ore Pulp pH        346475 non-null   float64
 5   Ore Pulp Density   346475 non-null   float64
 6   % Silica Concentrate 346475 non-null   float64
dtypes: float64(7)
memory usage: 21.1+ MB
```

In [23]: `dataset.head()`

Out[23]:

	avg_air_flow_267	avg_float_level_47	% Iron Feed	Amina Flow	Ore Pulp pH	Ore Pulp Density	% Silica Concentrate
date							
2017-03-10 01:00:00	251.448000	483.4510	55.2	557.434	10.0664	1.74	1.31
2017-03-10 01:00:00	249.8876667	473.0805	55.2	563.965	10.0672	1.74	1.31
2017-03-10 01:00:00	249.0966667	454.1275	55.2	568.054	10.0680	1.74	1.31
2017-03-10 01:00:00	252.0186667	436.9395	55.2	568.665	10.0689	1.74	1.31
2017-03-10 01:00:00	250.349000	439.6745	55.2	558.167	10.0697	1.74	1.31

In [24]: `x=dataset.iloc[:, :-1].values`  
`y=dataset.iloc[:, -1].values`

In [25]: `x`

Out[25]: `array([[251.448 , 483.451 , 55.2 , 557.434 , 10.0664 , 1.74 ,`  
`[249.8876667, 473.0805, 55.2, 563.965, 10.0672, 1.74],`  
`[249.0966667, 454.1275, 55.2, 568.054, 10.0680, 1.74],`  
`[252.0186667, 436.9395, 55.2, 568.665, 10.0689, 1.74],`  
`[250.349000, 439.6745, 55.2, 558.167, 10.0697, 1.74],`  
`[...,`  
`[321.91983333, 444.47 , 49.75 , 501.668 , 9.75384 , 1.71849],`  
`[320.522 , 434.3475 , 49.75 , 501.514 , 9.74102 , 1.71701],`  
`[319.41466667, 389.8735 , 49.75 , 483.383 , 9.73784 , 1.7091 ]])`

In [26]: `y`

Out[26]: `array([1.31, 1.31, 1.31, ..., 1.96, 1.96, 1.96])`

In [27]: `from sklearn.model_selection import train_test_split`  
`x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)`

In [28]: `from sklearn.ensemble import RandomForestRegressor`

```
In [29]: from sklearn.ensemble import RandomForestRegressor  
model=RandomForestRegressor(n_estimators=42,criterion='mse')  
model.fit(x_train,y_train)
```

```
Out[29]: RandomForestRegressor(n_estimators=42)
```

```
In [30]: y_pred=model.predict(x_test)  
y_pred
```

```
Out[30]: array([1.32095238, 1.23      , 1.88      , ..., 3.33      , 2.19100285,  
   2.8247619 ])
```

```
In [31]: from sklearn.metrics import r2_score  
r2_score(y_test,y_pred)
```

```
Out[31]: 0.9152084225696088
```

```
In [33]: import pickle  
pickle.dump(model,open('mining (1).pkl','wb'))
```

```
In [34]: model.predict([[251.448000,483.4510,55.20,557.434,10.06640,1.74000]])
```

```
Out[34]: array([1.29952381])
```

```
In [ ]:
```