```python
#importing the file using pandas

import pandas as pd

dataset = pd.read_csv('Material Compressive Strength Experimental
Data.csv')

dataset
```

```
      Material Quantity (gm)  Additive Catalyst (gm)  Ash Component
(gm)  \
0                     486.42                  180.60
21.26
1                     133.32                  260.14
185.60
2                     559.97                    2.84
111.76
3                     391.43                  351.05
76.39
4                     394.78                  352.61
194.35
...                      ...                     ...                 .
..
6134                  188.78                  162.30
142.65
6135                  349.87                  291.45
77.82
6136                  358.29                   22.70
17.99
6137                  445.25                  275.59
178.86
6138                  560.23                  266.56
167.14

      Water Mix (ml)  Plasticizer (gm)  Moderate Aggregator  \
0             201.66             16.11              1151.17
1             175.99              6.27              1090.57
2             295.23             11.95              1024.93
3             299.14             19.00              1134.88
4             235.54             17.02              1098.24
...              ...               ...                  ...
6134          163.66             15.98              1003.82
6135          188.26             25.82               925.10
6136          208.58             34.91              1081.07
6137          191.77             18.07               865.15
6138          175.49             10.63              1165.87

      Refined Aggregator  Formulation Duration (hrs)  Compression
Strength MPa
0                 708.50                      344.43
```

```
79.89
1                1010.25                       28.86
59.80
2                 810.69                      237.68
77.86
3                 881.34                      208.81
71.74
4                 781.01                      266.84
76.07
...                  ...                         ...
...
6134             1002.47                      357.91
50.61
6135             1005.31                      104.20
54.24
6136              792.44                      302.76
56.57
6137              833.10                      374.63
58.21
6138              894.53                      360.96
58.96

[6139 rows x 9 columns]
```

```
#dimesdimesnsion of the data

dataset.shape
```

```
(6139, 9)
```

```
#data description

dataset.describe()
```

```
       Material Quantity (gm)  Additive Catalyst (gm)  Ash Component
(gm)  \
count             6030.000000             6030.000000
6030.000000
mean               383.642297              196.699846
111.856252
std                149.994316              133.329220
74.241117
min                124.440000                0.000000
0.000000
25%                256.030000               78.210000
44.582500
50%                377.405000              192.320000
115.250000
75%                511.522500              307.650000
174.257500
max                658.800000              438.470000
```

```
244.120000

       Water Mix (ml)  Plasticizer (gm)  Moderate Aggregator  \
count     6030.000000       6030.000000          6030.000000
mean       224.296955         17.651085           998.669332
std         41.545751         11.687965            97.732677
min        148.600000          0.000000           821.540000
25%        190.387500          7.922500           918.437500
50%        225.700000         16.345000           997.985000
75%        257.447500         27.667500          1079.827500
max        301.340000         39.280000          1174.360000

       Refined Aggregator  Formulation Duration (hrs)  \
count         6030.000000                 6030.000000
mean           811.832398                  174.408504
std            112.813539                  112.415173
min            609.230000                   16.250000
25%            717.447500                   70.300000
50%            810.260000                  163.105000
75%            905.857500                  272.602500
max           1018.050000                  380.250000

       Compression Strength MPa
count               6139.000000
mean                  56.851430
std                   16.124932
min                    2.610000
25%                   47.085000
50%                   59.790000
75%                   69.845000
max                   92.510000
```

```
#dataset info

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6139 entries, 0 to 6138
Data columns (total 9 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Material Quantity (gm)      6030 non-null   float64
 1   Additive Catalyst (gm)      6030 non-null   float64
 2   Ash Component (gm)          6030 non-null   float64
 3   Water Mix (ml)              6030 non-null   float64
 4   Plasticizer (gm)            6030 non-null   float64
 5   Moderate Aggregator         6030 non-null   float64
 6   Refined Aggregator          6030 non-null   float64
 7   Formulation Duration (hrs)  6030 non-null   float64
 8   Compression Strength MPa    6139 non-null   float64
```

```
dtypes: float64(9)
memory usage: 431.8 KB

#finding variant columns

dataset.var()

Material Quantity (gm)        22498.294732
Additive Catalyst (gm)        17776.680972
Ash Component (gm)             5511.743471
Water Mix (ml)                 1726.049445
Plasticizer (gm)                136.608535
Moderate Aggregator            9551.676176
Refined Aggregator            12726.894650
Formulation Duration (hrs)    12637.171095
Compression Strength MPa        260.013420
dtype: float64

dataset.var().sort_values(ascending = False)

Material Quantity (gm)        22498.294732
Additive Catalyst (gm)        17776.680972
Refined Aggregator            12726.894650
Formulation Duration (hrs)    12637.171095
Moderate Aggregator            9551.676176
Ash Component (gm)             5511.743471
Water Mix (ml)                 1726.049445
Compression Strength MPa        260.013420
Plasticizer (gm)                136.608535
dtype: float64

#finding correlated columns

corr_matrix = dataset.corr()
corr_matrix

                            Material Quantity (gm)  Additive Catalyst
(gm)  \
Material Quantity (gm)                    1.000000
0.009507
Additive Catalyst (gm)                    0.009507
1.000000
Ash Component (gm)                       -0.024180
0.053598
Water Mix (ml)                            0.004640
0.029818
Plasticizer (gm)                          0.048551
0.140246
Moderate Aggregator                      -0.009366               -
0.022772
Refined Aggregator                       -0.016475
```

```
0.009807
Formulation Duration (hrs)                        0.066251
0.162214
Compression Strength MPa                          0.130875
0.180811

                            Ash Component (gm)  Water Mix (ml)  \
Material Quantity (gm)                -0.024180        0.004640
Additive Catalyst (gm)                 0.053598        0.029818
Ash Component (gm)                     1.000000       -0.006846
Water Mix (ml)                        -0.006846        1.000000
Plasticizer (gm)                       0.161667       -0.024760
Moderate Aggregator                   -0.003301       -0.029820
Refined Aggregator                     0.040000       -0.054666
Formulation Duration (hrs)             0.109820        0.031210
Compression Strength MPa               0.090961       -0.027051

                            Plasticizer (gm)  Moderate Aggregator  \
Material Quantity (gm)               0.048551            -0.009366
Additive Catalyst (gm)               0.140246            -0.022772
Ash Component (gm)                   0.161667            -0.003301
Water Mix (ml)                      -0.024760            -0.029820
Plasticizer (gm)                     1.000000            -0.020225
Moderate Aggregator                 -0.020225             1.000000
Refined Aggregator                   0.056807            -0.006605
Formulation Duration (hrs)           0.156834             0.008240
Compression Strength MPa             0.207256            -0.032151

                            Refined Aggregator  Formulation Duration
(hrs)  \
Material Quantity (gm)                -0.016475
0.066251
Additive Catalyst (gm)                 0.009807
0.162214
Ash Component (gm)                     0.040000
0.109820
Water Mix (ml)                        -0.054666
0.031210
Plasticizer (gm)                       0.056807
0.156834
Moderate Aggregator                   -0.006605
0.008240
Refined Aggregator                     1.000000
0.006408
Formulation Duration (hrs)             0.006408
1.000000
Compression Strength MPa              -0.010762
0.268032

                            Compression Strength MPa
```

```
Material Quantity (gm)                            0.130875
Additive Catalyst (gm)                            0.180811
Ash Component (gm)                                0.090961
Water Mix (ml)                                   -0.027051
Plasticizer (gm)                                  0.207256
Moderate Aggregator                              -0.032151
Refined Aggregator                               -0.010762
Formulation Duration (hrs)                        0.268032
Compression Strength MPa                          1.000000
```

```
# finding columns that carry more than 50% of the same information by
setting threshold

threshold = 0.5
correlated_columns = set()

for row in range(len(corr_matrix)):
    for col in range(row):
        if abs (corr_matrix.iloc[row][col]) > threshold:

            corr = correlated_columns.add(corr_matrix.columns[row])
            print(f'correlated column',corr)
        else:
            print('There are no correlated columns')
```

```
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
There are no correlated columns
```

OBSERVATION :

Thus, there are no columns that are related columns. All columns are mutually exclusive of each other.

```
#null values count

dataset.isnull().sum()

Material Quantity (gm)        0
Additive Catalyst (gm)        0
Ash Component (gm)            0
Water Mix (ml)                0
Plasticizer (gm)              0
Moderate Aggregator           0
Refined Aggregator            0
Formulation Duration (hrs)    0
Compression Strength MPa      0
dtype: int64

#109 null values ate present in 8 columns which is significant amount
of data to be dropped. So,filling with mean would be best here.

#filling Material Quantity (gm)

dataset['Material Quantity (gm)'].mean()

383.6422968490886

dataset['Material Quantity (gm)'] = dataset['Material Quantity
(gm)'].fillna(dataset['Material Quantity (gm)'].mean())

dataset['Material Quantity (gm)'].isnull().sum()

0
```

OBSERVATION :

The null values in 'Material Quantity (gm)' is filled with its mean value.

```python
#filling Additive Catalyst (gm)

add_mean = dataset['Additive Catalyst (gm)'].mean()

dataset['Additive Catalyst (gm)'] = dataset['Additive Catalyst (gm)'].fillna(add_mean)

dataset['Additive Catalyst (gm)'].isnull().sum()
```

```
0
```

OBSERVATION :

The null values in 'Additive Catalyst (gm)' is filled with its mean value.

```python
#filling Ash Component (gm)

ash_mean = dataset['Ash Component (gm)'].mean()
dataset['Ash Component (gm)'] = dataset['Ash Component (gm)'].fillna(ash_mean)
dataset['Ash Component (gm)'].isnull().sum()
```

```
0
```

OBSERVATION : The null values in 'Ash Component (gm)' is filled with its mean value.

```python
#filling Water Mix (ml)

wat_mean = dataset['Water Mix (ml)'].mean()
dataset['Water Mix (ml)'] = dataset['Water Mix (ml)'].fillna(wat_mean)
dataset['Water Mix (ml)'].isnull().sum()
```

```
0
```

OBSERVATION :

The null values in 'Water Mix (ml)' is filled with its mean value.

```python
#filling Plasticizer (gm)

pla_mean = dataset['Plasticizer (gm)'].mean()
dataset['Plasticizer (gm)'] = dataset['Plasticizer (gm)'].fillna(pla_mean)
dataset['Plasticizer (gm)'].isnull().sum()
```

```
0
```

OBSERVATION : The null values in 'Plasticizer (gm)' is filled with its mean value.

```python
#filling Moderate Aggregator
```

```
mod_mean = dataset['Moderate Aggregator'].mean()
dataset['Moderate Aggregator'] = dataset['Moderate
Aggregator'].fillna(mod_mean)
dataset['Moderate Aggregator'].isnull().sum()
```

```
0
```

OBSERVATION:

The null values in 'Moderate Aggregator' is filled with its mean value.

```
#filling Refined Aggregator

ref_mean = dataset['Refined Aggregator'].mean()
dataset['Refined Aggregator'] = dataset['Refined
Aggregator'].fillna(ref_mean)
dataset['Refined Aggregator'].isnull().sum()
```

```
0
```

OBSERVATION : The null values in 'Refined Aggregator' is filled with its mean value.

```
#filling Formulation Duration (hrs)

for_mean = dataset['Formulation Duration (hrs)'].mean()
dataset['Formulation Duration (hrs)'] = dataset['Formulation Duration
(hrs)'].fillna(for_mean)
dataset['Formulation Duration (hrs)'].isnull().sum()
```

```
0
```

OBSERVATION:

The null values in 'Refined Aggregator' is filled with its mean value.

```
dataset.isnull().sum()
```

```
Material Quantity (gm)        0
Additive Catalyst (gm)        0
Ash Component (gm)            0
Water Mix (ml)                0
Plasticizer (gm)              0
Moderate Aggregator           0
Refined Aggregator            0
Formulation Duration (hrs)    0
Compression Strength MPa      0
dtype: int64
```

OBSERVATION:

All the null values are filled.

```
#checking skeweness of data

dataset.skew()

Material Quantity (gm)        0.096605
Additive Catalyst (gm)        0.107584
Ash Component (gm)           -0.001224
Water Mix (ml)                0.024953
Plasticizer (gm)              0.182842
Moderate Aggregator          -0.020582
Refined Aggregator           -0.006749
Formulation Duration (hrs)    0.233290
Compression Strength MPa     -0.766954
dtype: float64
```

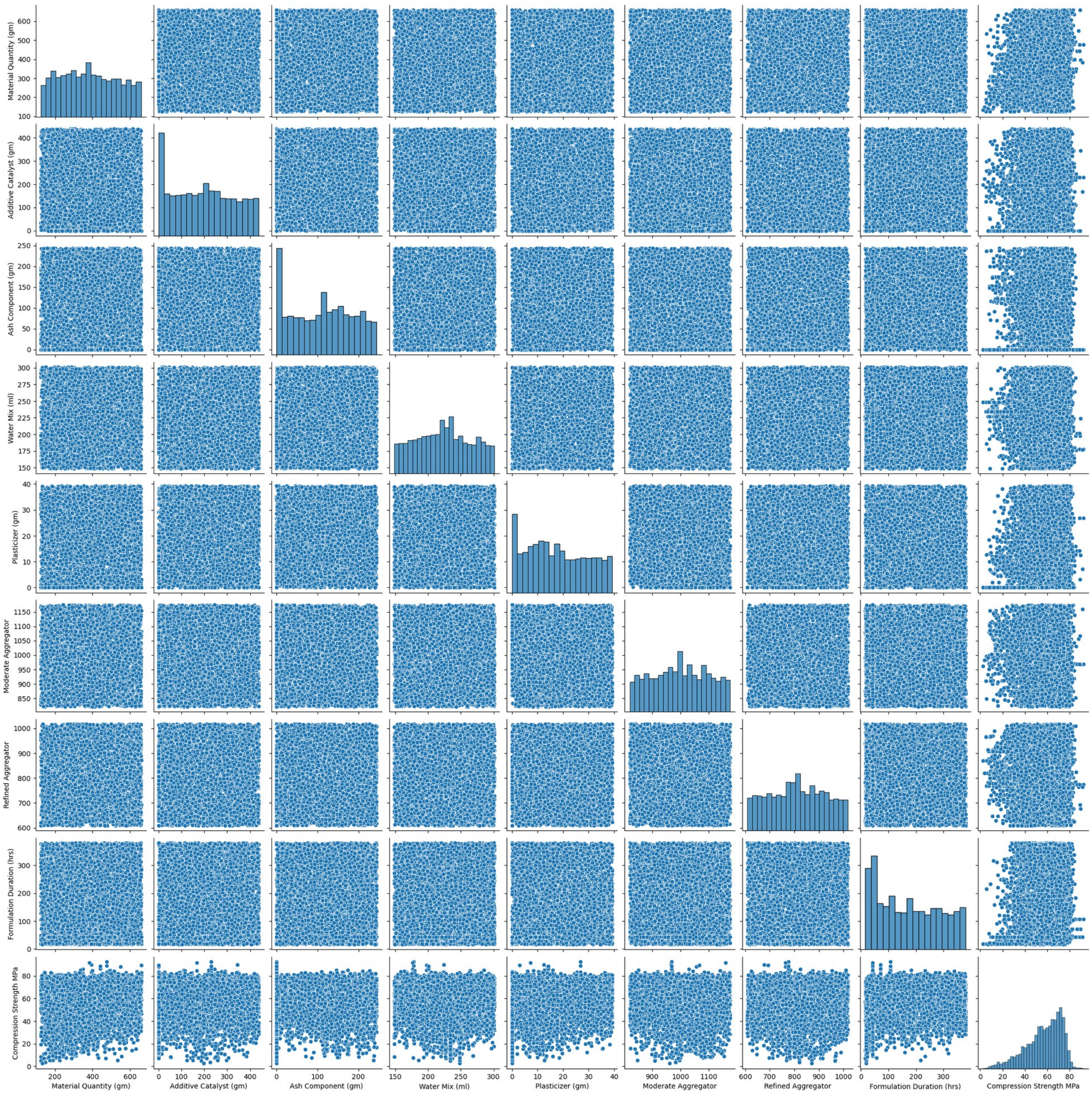
NOTES -0.5 and 0.5, the distribution of the value is almost symmetrical. -1 and -0.5, the data is negatively skewed. 0.5 to 1, the data is positively skewed.

OBSERVATION

Almost symmetrical - Compression Strength MPa Negatively skewed - Compression Strength MPa

we can say Compression Strength MPa is almost symmetrical but negatively screwed

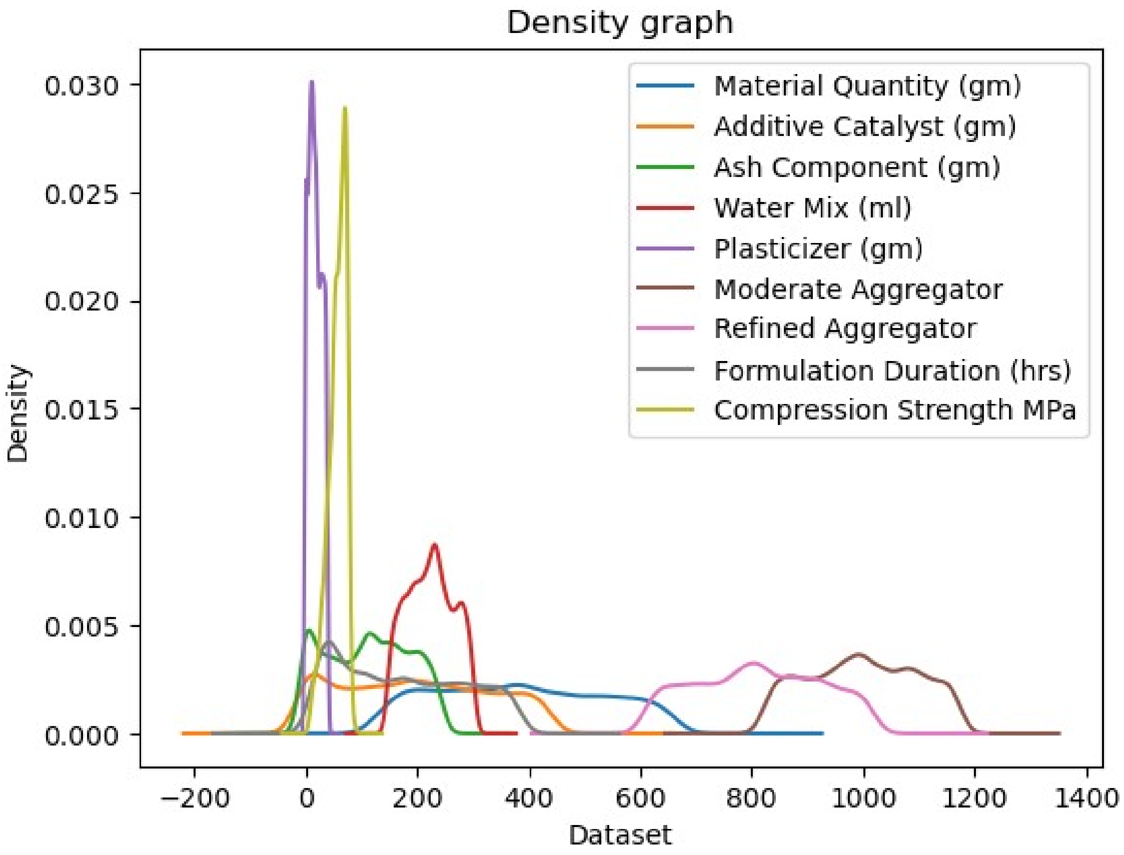
```
import seaborn as sns

sns.pairplot(dataset)

<seaborn.axisgrid.PairGrid at 0x2aabdb54d90>
```



OBSERVATION:

The above graph explains relationship between two variables

```
import matplotlib.pyplot as plt
dataset.plot(kind = 'density')
plt.title('Density graph')
plt.xlabel('Dataset')
plt.show()
```

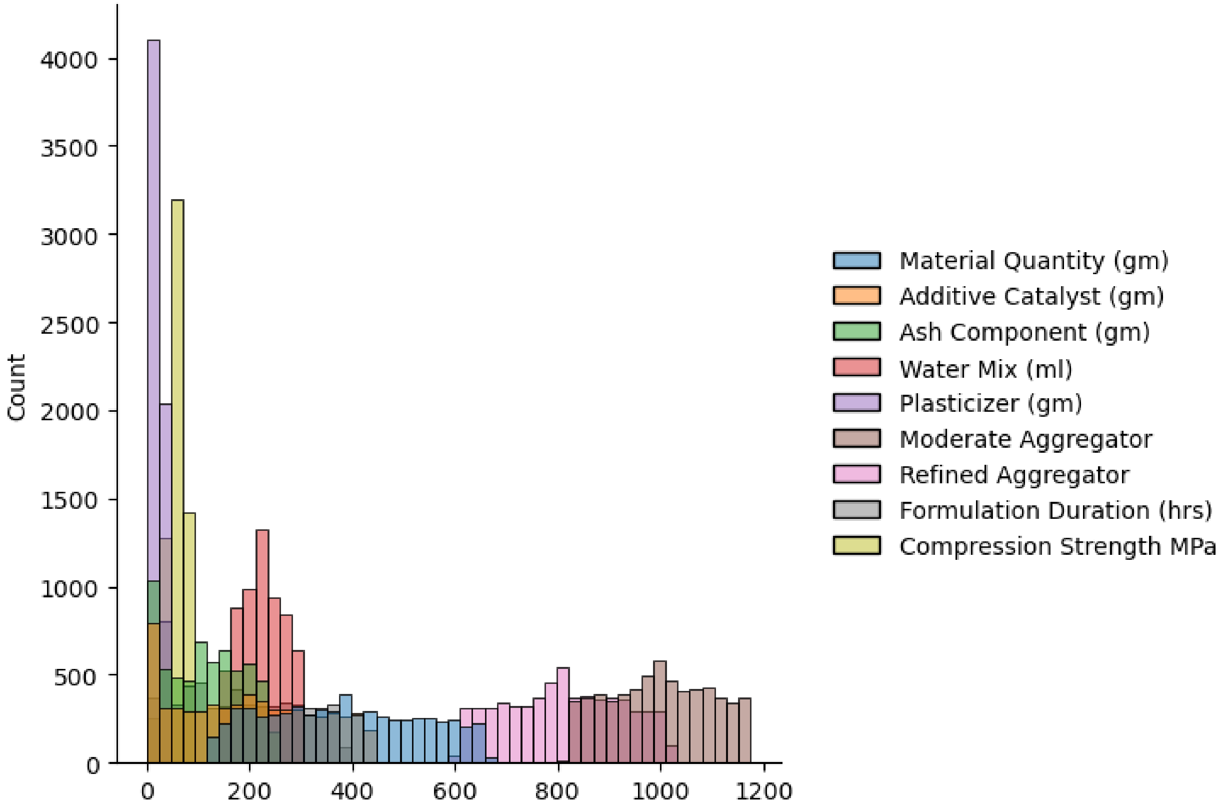


OBSERVATION:

Data is distributed highly in Plasticizer(gm) and Compression Strength Mpa. Data is distributed less in Addictive Catalyst (gm) and Material Qunatity (gm)

```
sns.displot(dataset, legend=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x2aabdb54970>
```



```
#scaling data
#NEED FOR SCALING :Making dataset normally distributed for further ML
modelling.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled_data = scaler.fit_transform(dataset)

scaled_data
```

```
array([[ 6.91433284e-01, -1.21849163e-01, -1.23137947e+00, ...,
        -9.24275536e-01,  1.52617760e+00,  1.42887096e+00],
       [-1.68403421e+00,  4.80136880e-01,  1.00232112e+00, ...,
         1.77478254e+00, -1.30649872e+00,  1.82872712e-01],
       [ 1.18623825e+00, -1.46719791e+00, -1.30825309e-03, ...,
        -1.02183880e-02,  5.67949006e-01,  1.30296869e+00],
       ...,
       [-1.70556661e-01, -1.31689060e+00, -1.27582514e+00, ...,
        -1.73458851e-01,  1.15213183e+00, -1.74545314e-02],
       [ 4.14463597e-01,  5.97067787e-01,  9.10711399e-01, ...,
         1.90231956e-01,  1.79726428e+00,  8.42596109e-02],
       [ 1.18798739e+00,  5.28725645e-01,  7.51413779e-01, ...,
         7.39703829e-01,  1.67455716e+00,  1.30775225e-01]])
```

OBSERVATION:

The data is scaled between the range -1 to 1