

1. Explore each independent feature w.r.t dependent variable(Compression Strength)

```
import pandas as pd
import matplotlib
from matplotlib import pyplot as plt
import seaborn as sns
df = pd.read_csv("Material Compressive Strength Experimental Data
(1).csv")

df.isnull().sum()

Material_Quantity      109
Additive Catalyst (gm)  109
Ash Component (gm)      109
Water Mix (ml)          109
Plasticizer (gm)        109
Moderate Aggregator     109
Refined Aggregator      109
Formulation Duration (hrs) 109
Compression Strength MPa    0
dtype: int64

df['Material_Quantity']
df["Material_Quantity"].mean()
Material_Quantity_mean = df["Material_Quantity"].mean()
Material_Quantity_mean
df['Material_Quantity'] =
df['Material_Quantity'].fillna(Material_Quantity_mean)
df['Additive Catalyst (gm)']
df["Additive Catalyst (gm)"].mean()
Additive_Catalyst_mean = df["Additive Catalyst (gm)"].mean()
Additive_Catalyst_mean
df['Additive Catalyst (gm)'] = df['Additive Catalyst
(gm)'].fillna(Additive_Catalyst_mean)
df['Ash Component (gm)']
df["Ash Component (gm)"].mean()
Ash_Component_mean = df["Ash Component (gm)"].mean()
Ash_Component_mean
df['Ash Component (gm)'] = df['Ash Component
(gm)'].fillna(Ash_Component_mean)
df['Water Mix (ml)']
df["Water Mix (ml)"].mean()
Water_Mix_mean = df["Water Mix (ml)"].mean()
Water_Mix_mean
df['Water Mix (ml)'] = df['Water Mix (ml)'].fillna(Water_Mix_mean)
df['Plasticizer (gm)']
```

```

df["Plasticizer (gm)"].mean()
Plasticizer_mean = df["Plasticizer (gm)"].mean()
Plasticizer_mean
df['Plasticizer (gm)'] = df['Plasticizer
(gm)'].fillna(Plasticizer_mean)
df['Moderate Aggregator']
df["Moderate Aggregator"].mean()
Moderate_Aggregator_mean = df["Moderate Aggregator"].mean()
Moderate_Aggregator_mean
df['Moderate Aggregator'] = df['Moderate
Aggregator'].fillna(Moderate_Aggregator_mean)
df['Refined Aggregator']
df["Refined Aggregator"].mean()
Refined_Aggregator_mean = df["Refined Aggregator"].mean()
Refined_Aggregator_mean
df['Refined Aggregator'] = df['Refined
Aggregator'].fillna(Refined_Aggregator_mean)
df['Formulation Duration (hrs)']
df["Formulation Duration (hrs)"].mean()
Formulation_Duration_mean = df["Formulation Duration (hrs)"].mean()
Formulation_Duration_mean
df['Formulation Duration (hrs)'] = df['Formulation Duration
(hrs)'].fillna(Formulation_Duration_mean)

```

```
df.isnull().sum()
```

```

Material_Quantity      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

```

```

plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Material_Quantity', y='Compression
Strength MPa')
plt.title('Scatter Plot of Material_Quantity vs. Compression
Strength')
plt.xlabel('Material_Quantity')
plt.ylabel('Compression Strength MPa')
plt.show()

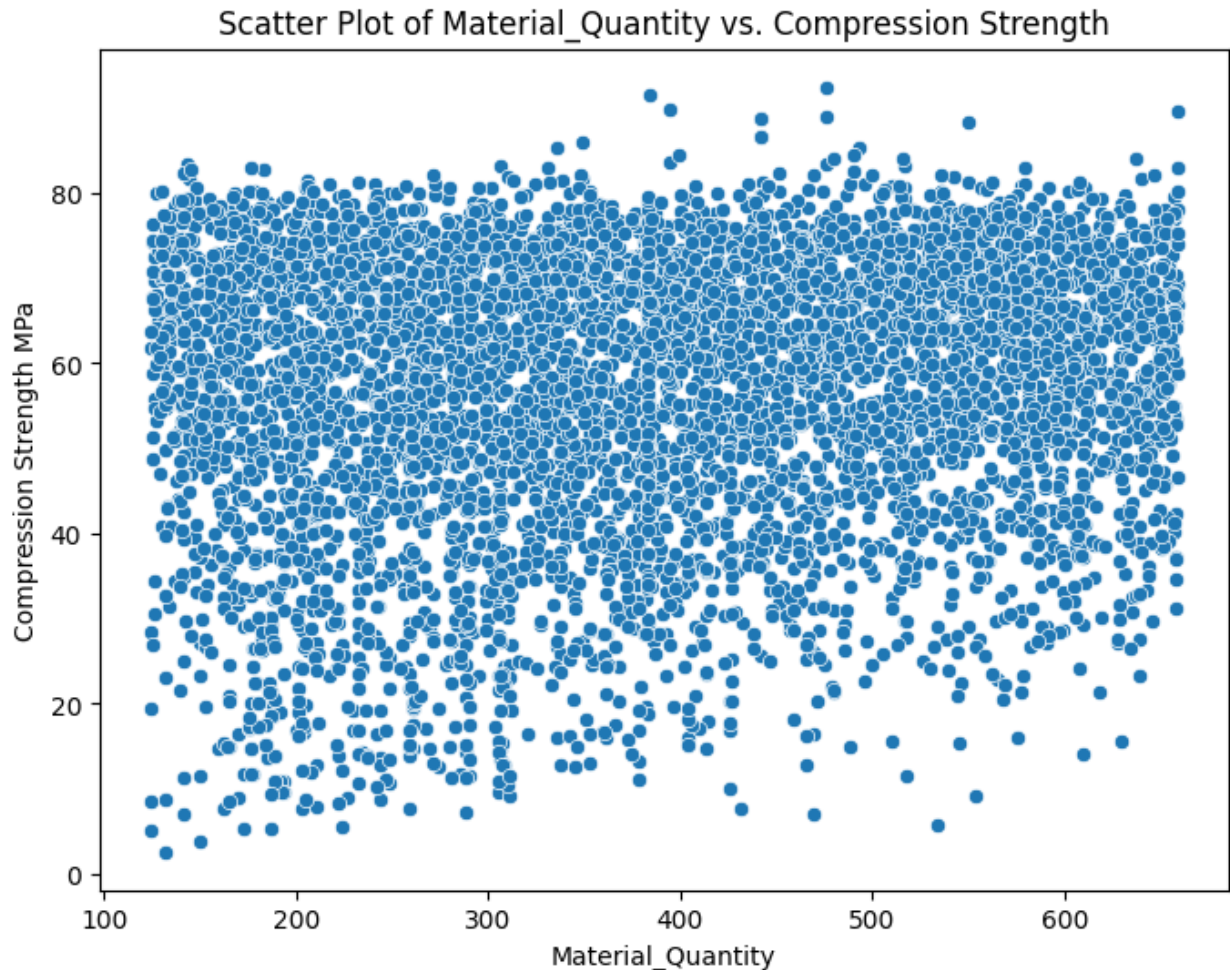
```

```

C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):

```

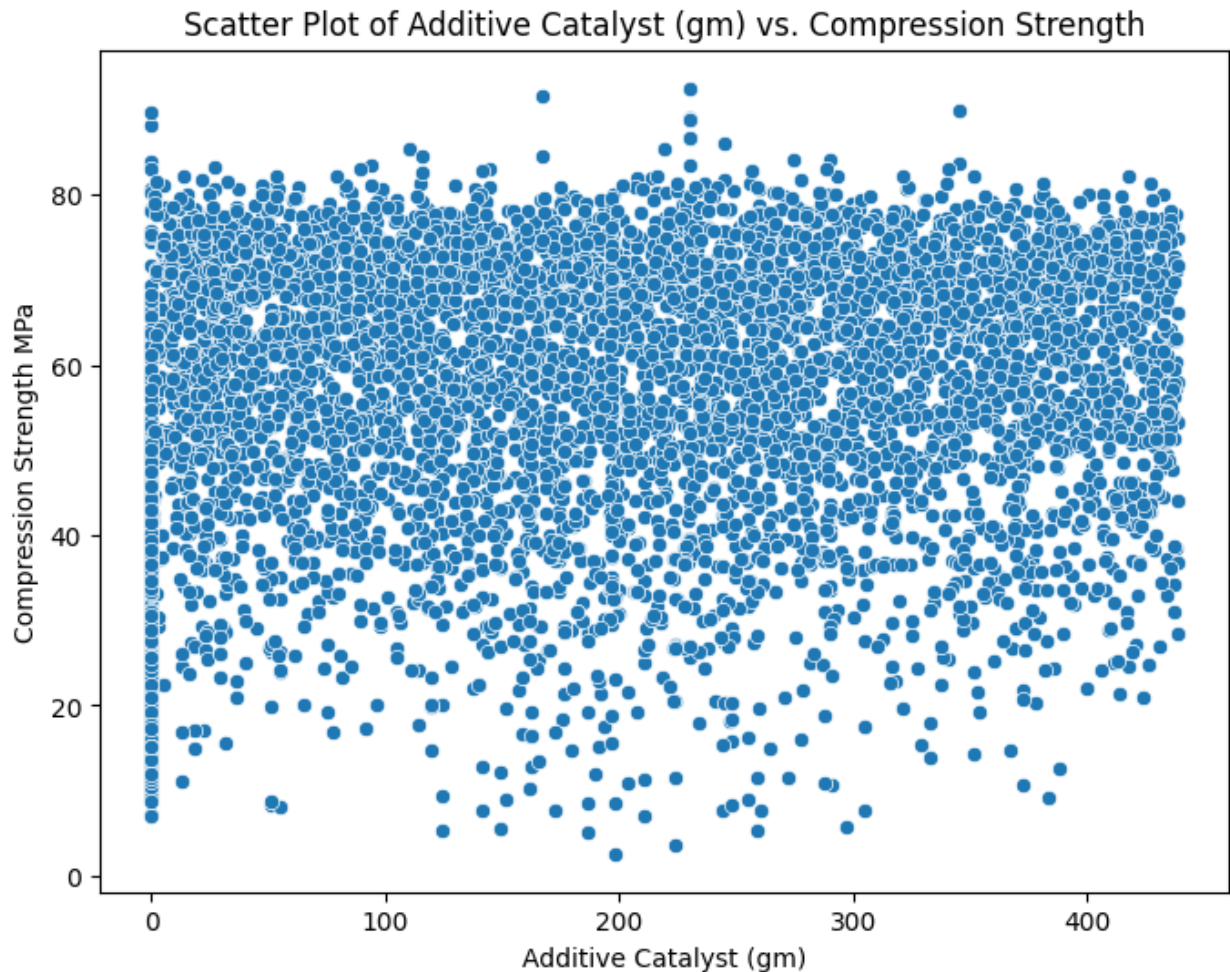
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```



```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Additive Catalyst (gm)', y='Compression
Strength MPa')
plt.title('Scatter Plot of Additive Catalyst (gm) vs. Compression
Strength')
plt.xlabel('Additive Catalyst (gm)')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
```

```
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```

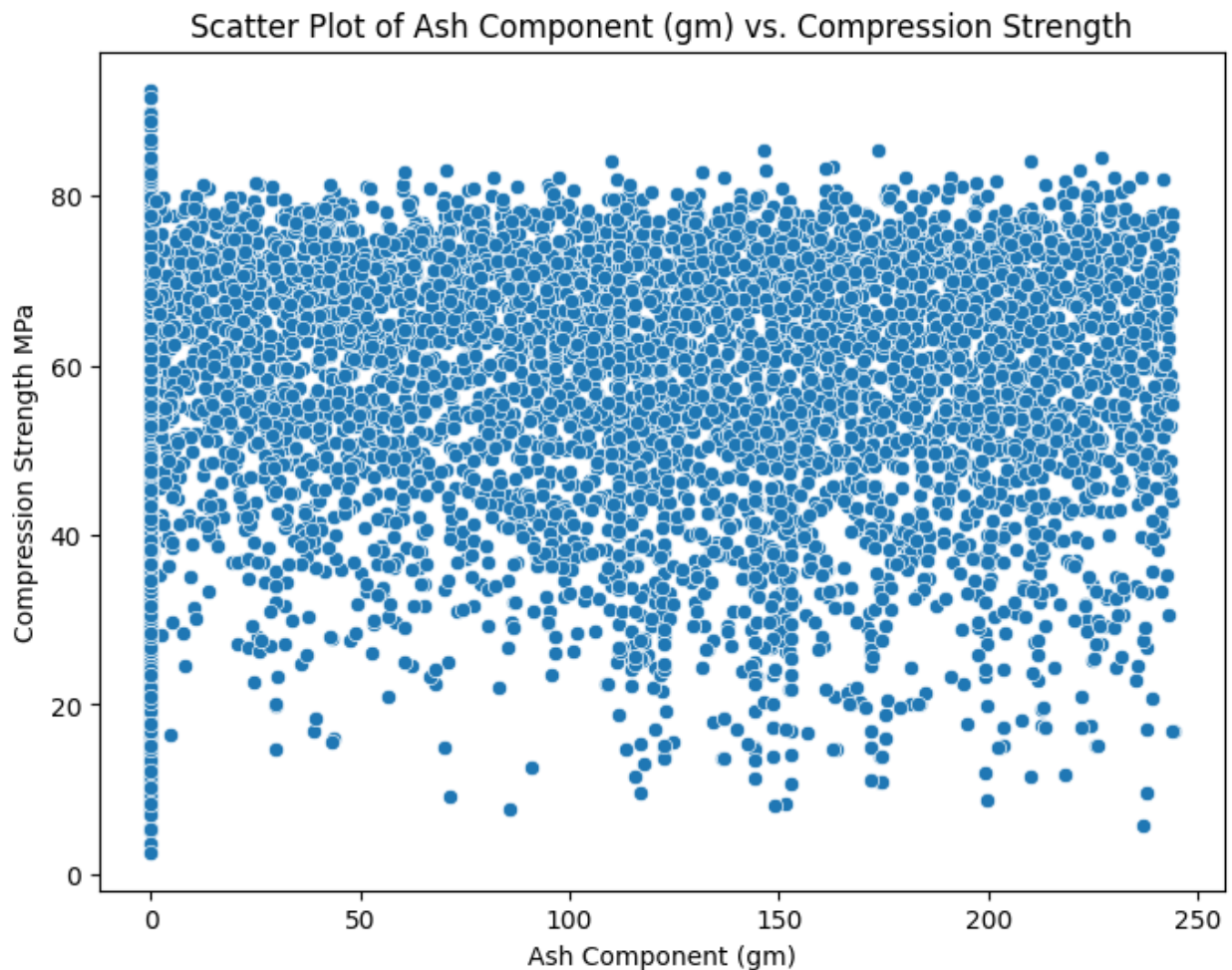


```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Ash Component (gm)', y='Compression
Strength MPa')
plt.title('Scatter Plot of Ash Component (gm) vs. Compression
Strength')
plt.xlabel('Ash Component (gm)')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
```

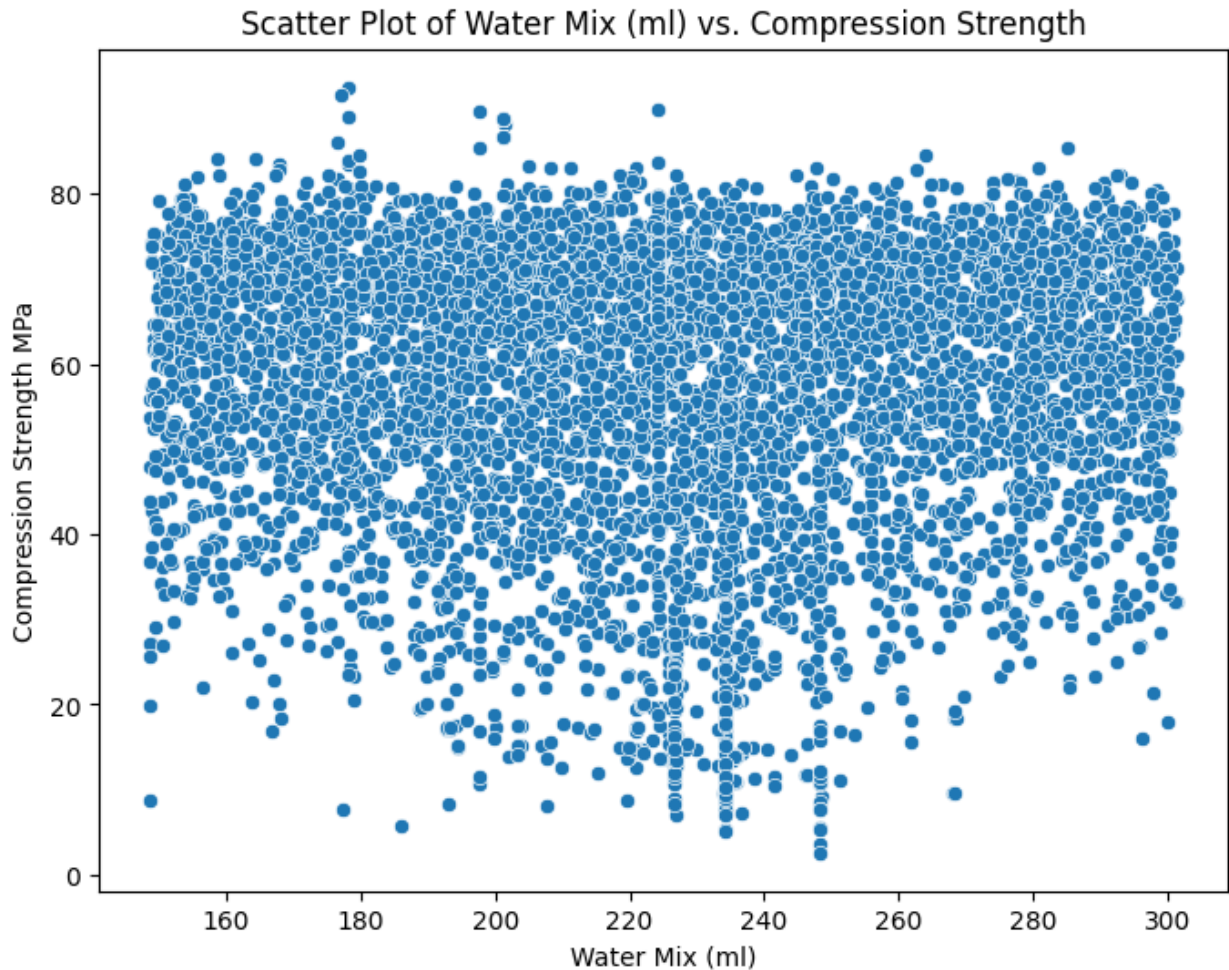


```
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```



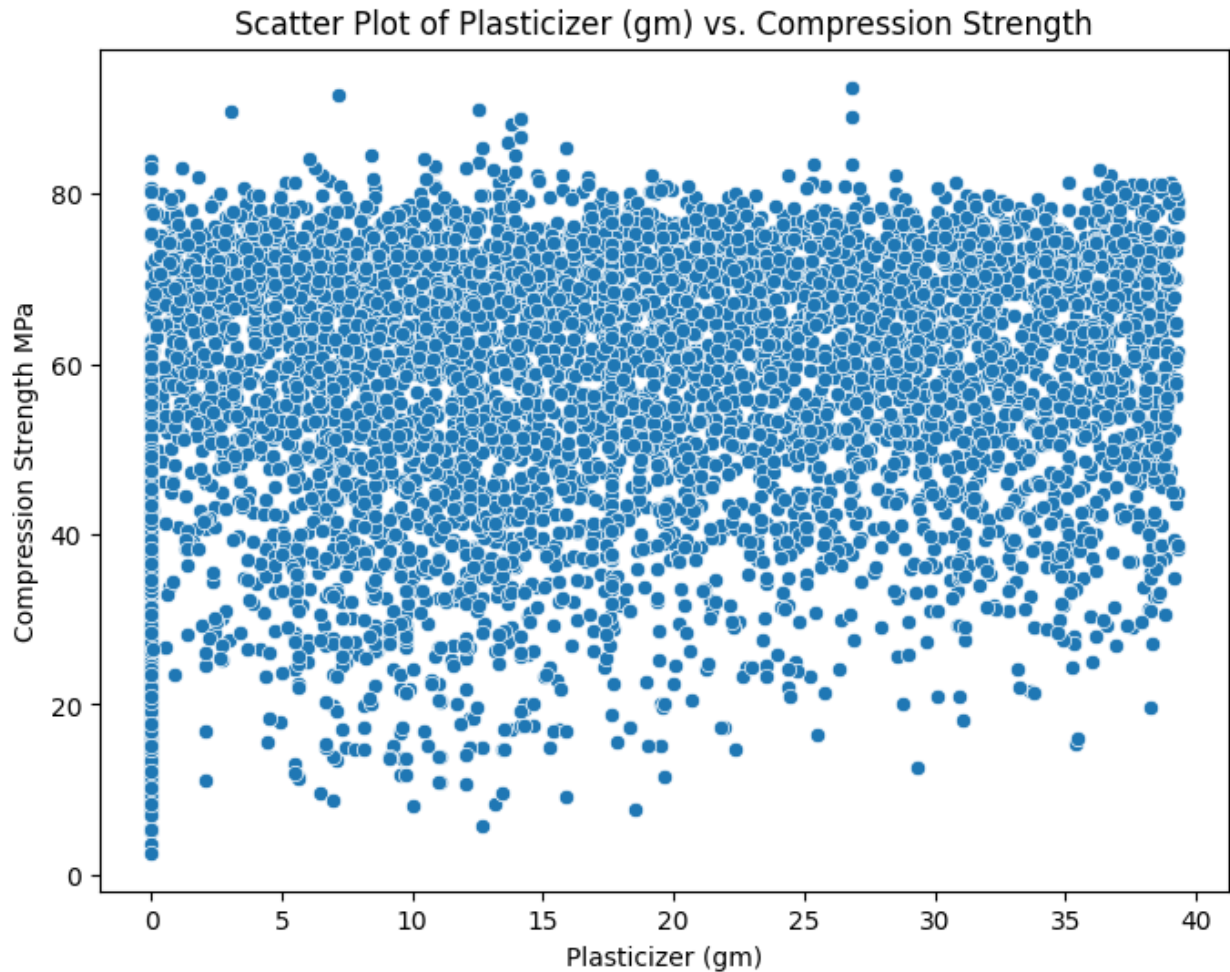
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Water Mix (ml)', y='Compression Strength
MPa')
plt.title('Scatter Plot of Water Mix (ml) vs. Compression Strength')
plt.xlabel('Water Mix (ml)')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```



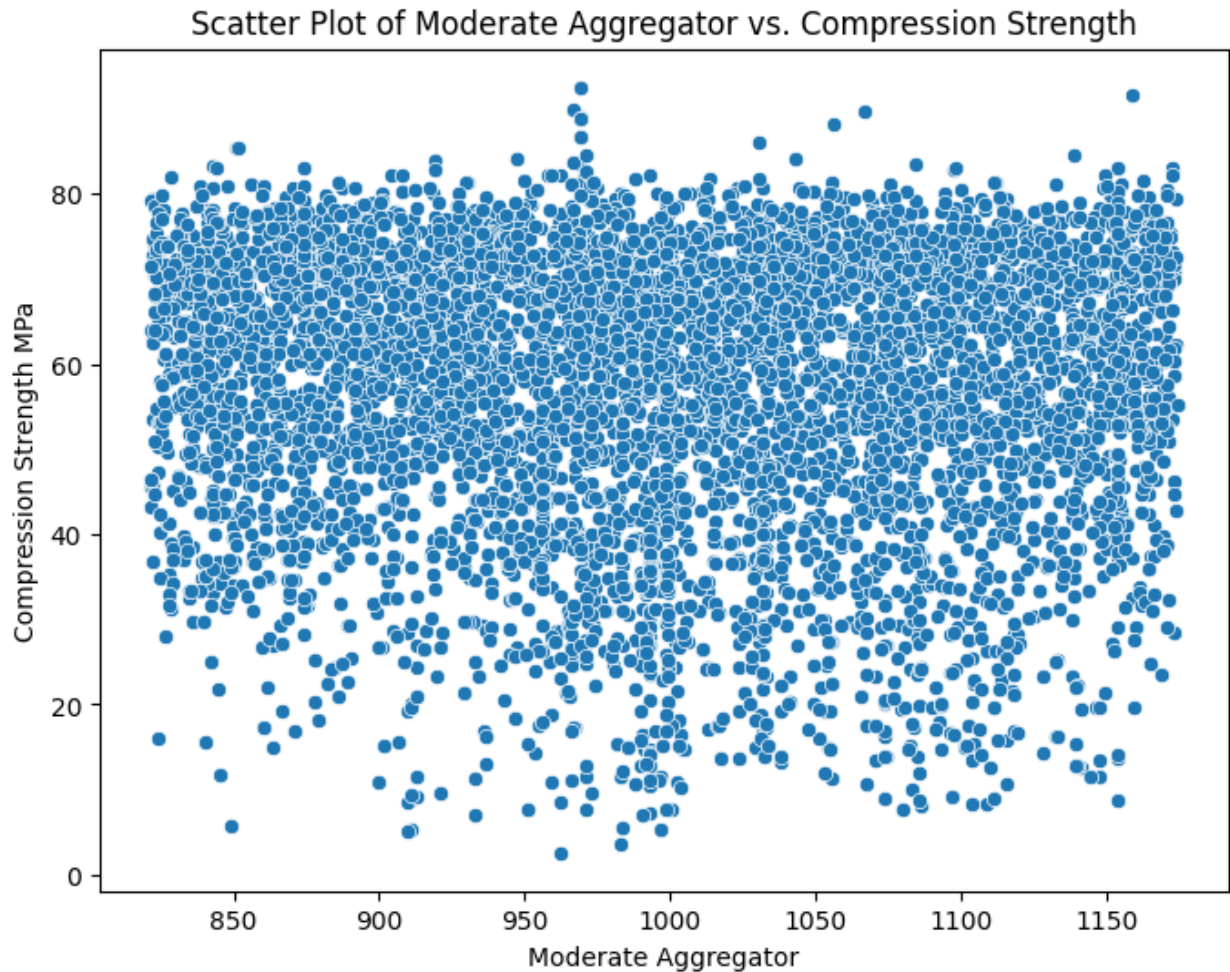
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Plasticizer (gm)', y='Compression Strength MPa')
plt.title('Scatter Plot of Plasticizer (gm) vs. Compression Strength')
plt.xlabel('Plasticizer (gm)')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



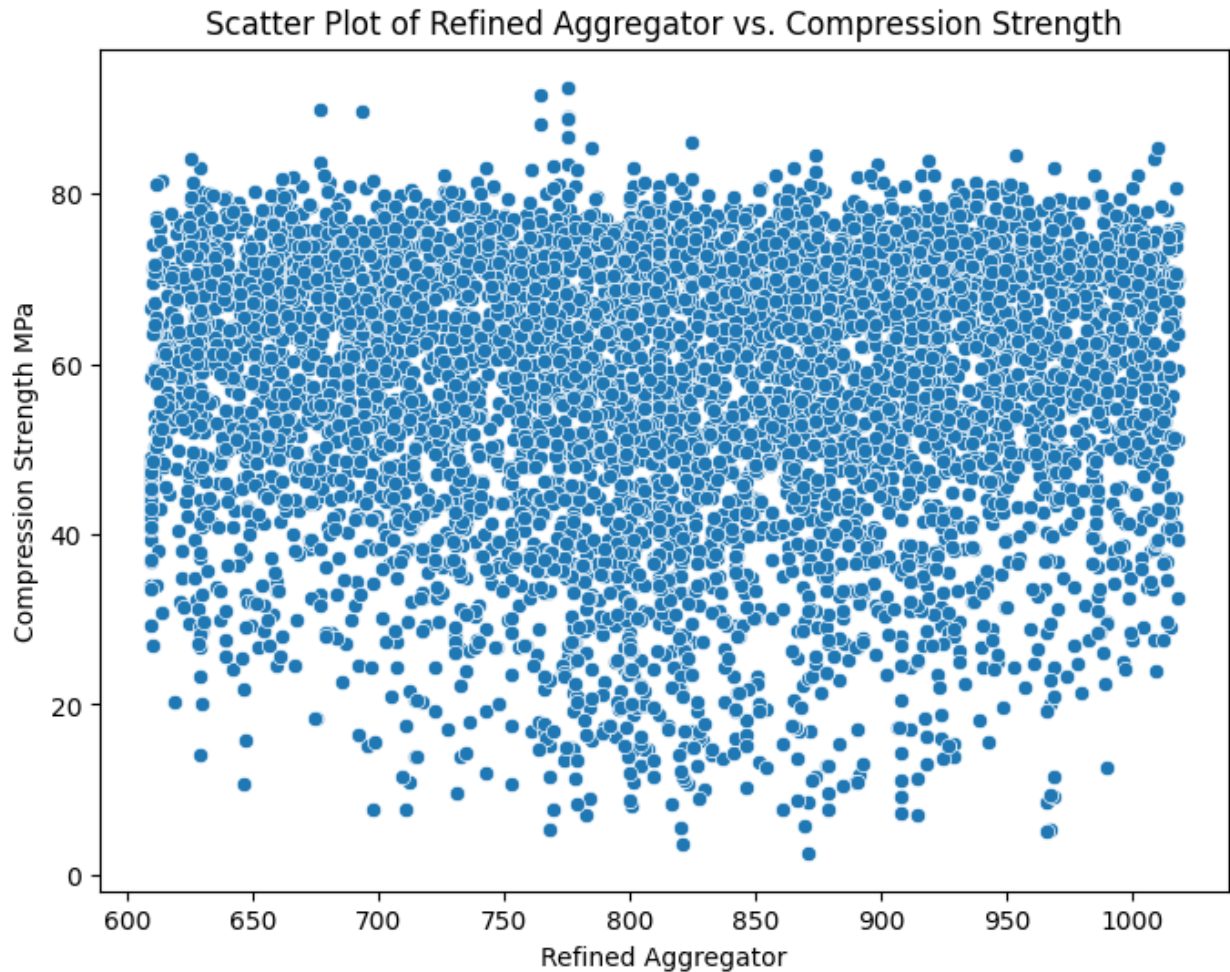
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Moderate Aggregator', y='Compression Strength MPa')
plt.title('Scatter Plot of Moderate Aggregator vs. Compression Strength')
plt.xlabel('Moderate Aggregator')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```

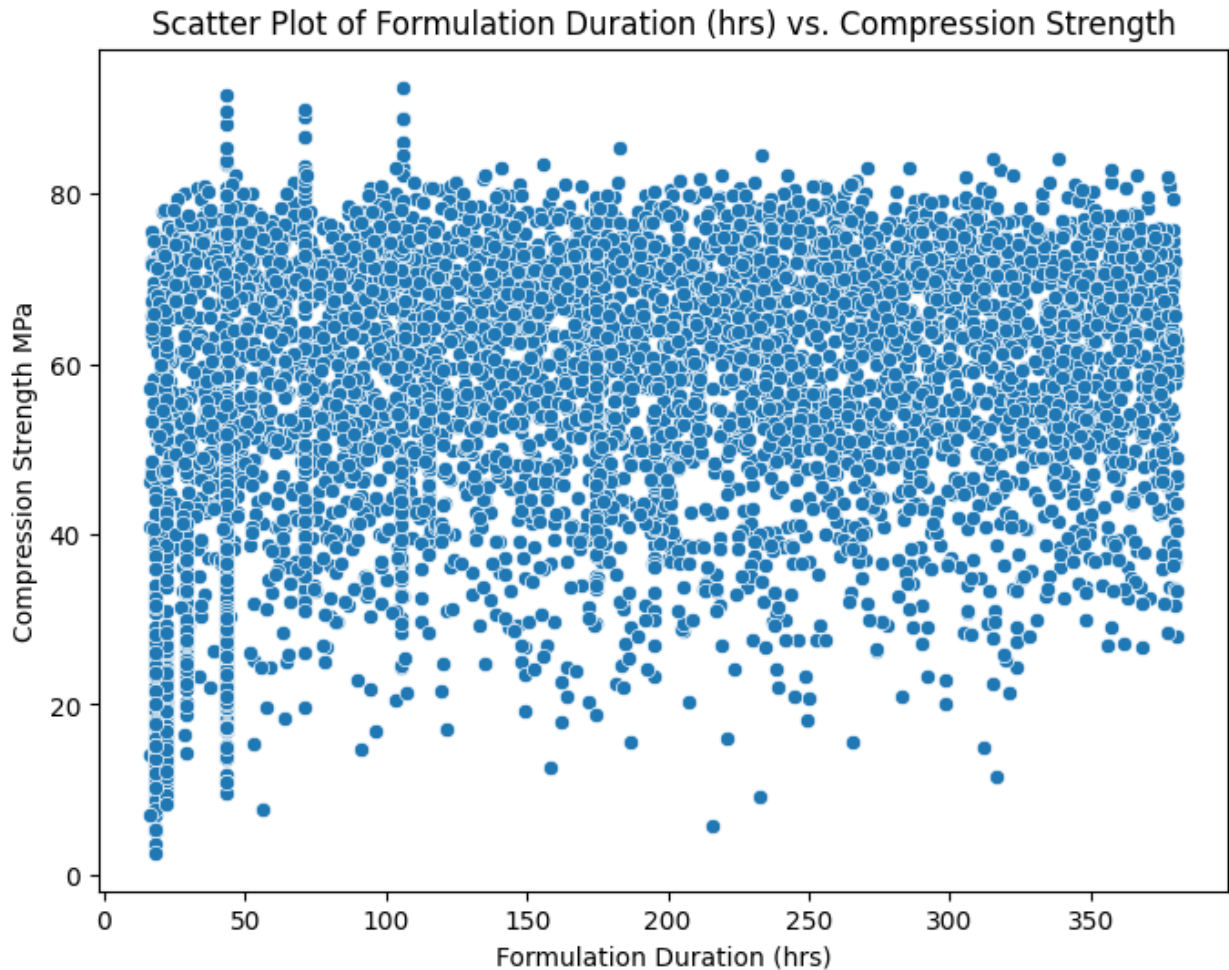
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Refined Aggregator', y='Compression
Strength MPa')
plt.title('Scatter Plot of Refined Aggregator vs. Compression
Strength')
plt.xlabel('Refined Aggregator')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Formulation Duration (hrs)',
y='Compression Strength MPa')
plt.title('Scatter Plot of Formulation Duration (hrs) vs. Compression
Strength')
plt.xlabel('Formulation Duration (hrs)')
plt.ylabel('Compression Strength MPa')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```



After handling with null values, here I checked the relation with each independent and dependent variable

2. Find the co-relation between dependent and independent variable.

```
correlation_coefficient = df['Compression Strength  
MPa'].corr(df['Material_Quantity'])  
correlation_coefficient
```

```
0.13007529709309232
```

```
correlation_coefficient = df['Compression Strength  
MPa'].corr(df['Additive Catalyst (gm)'])  
correlation_coefficient
```

```
0.1797059886553599
```

```

correlation_coefficient = df['Compression Strength MPa'].corr(df['Ash
Component (gm)'])
correlation_coefficient

0.09040508042374193

correlation_coefficient = df['Compression Strength
MPa'].corr(df['Water Mix (ml)'])
correlation_coefficient

-0.026886167045915132

correlation_coefficient = df['Compression Strength
MPa'].corr(df['Plasticizer (gm)'])
correlation_coefficient

0.20599022156208402

correlation_coefficient = df['Compression Strength
MPa'].corr(df['Moderate Aggregator'])
correlation_coefficient

-0.031954423119964596

correlation_coefficient = df['Compression Strength
MPa'].corr(df['Refined Aggregator'])
correlation_coefficient

-0.010696132475569099

correlation_coefficient = df['Compression Strength
MPa'].corr(df['Formulation Duration (hrs)'])
correlation_coefficient

0.26639438555286893

correlation_data = {
    'Feature': ['Material_Quantity', 'Additive Catalyst (gm)', 'Ash
Component (gm)', 'Water Mix (ml)',
                'Plasticizer (gm)', 'Moderate Aggregator', 'Refined
Aggregator', 'Formulation Duration (hrs)'],
    'Correlation': [0.13007529709309232, 0.1797059886553599,
0.09040508042374193, -0.026886167045915132,
0.20599022156208402, 0.031954423119964596,
0.010696132475569099, 0.26639438555286893]
}

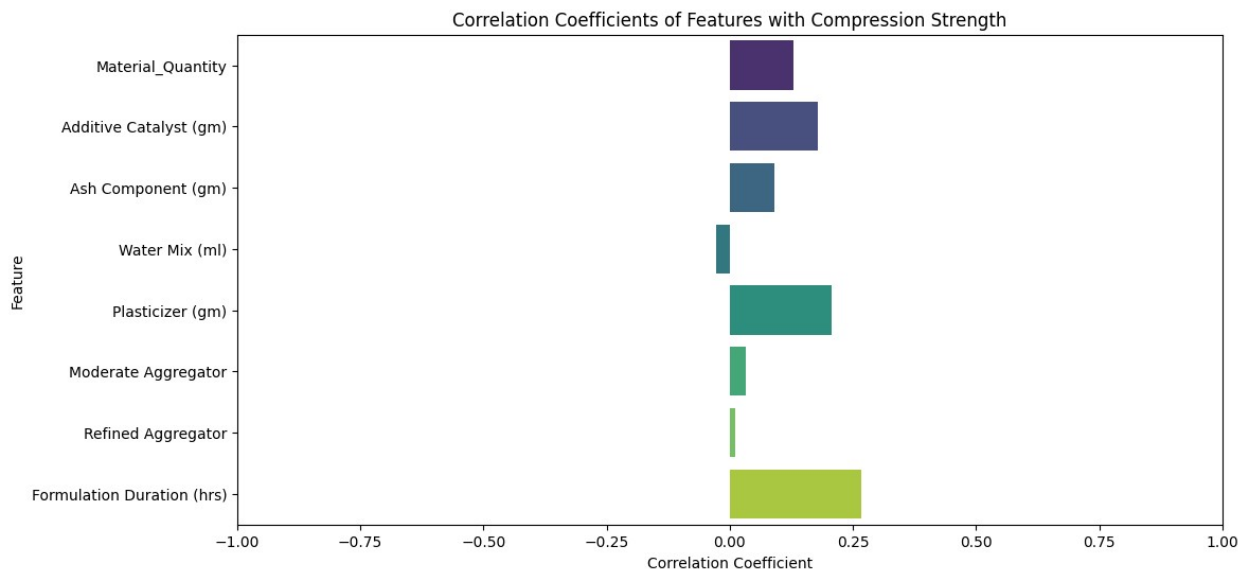
```

Above correlation_data shows that some columns are postively correlated and some are negatively.

3. Plot the co-relation of other features with Compression Strength.

```
correlation_df = pd.DataFrame(correlation_data)
plt.figure(figsize=(12, 6))
sns.barplot(x='Correlation', y='Feature', data=correlation_df,
palette='viridis')
plt.title('Correlation Coefficients of Features with Compression
Strength')
plt.xlabel('Correlation Coefficient')
plt.ylabel('Feature')
plt.xlim(-1, 1)
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```



Above plot showing how the features are correlated with the dependent variable Compression Strength

4. Check the outliers in the dataset using Boxplot and various techniques.

```
df.sample(5)
```

	Material_Quantity	Additive Catalyst (gm)	Ash Component (gm)	\
453	650.300000	70.820000	151.840000	
1772	348.780000	274.910000	67.080000	
850	487.170000	212.490000	200.030000	
1760	383.642297	196.699846	111.856252	
389	346.560000	171.630000	138.580000	

	Water Mix (ml)	Plasticizer (gm)	Moderate Aggregator	\
453	175.390000	20.810000	837.830000	
1772	262.730000	8.990000	1018.070000	
850	151.810000	31.990000	1128.990000	
1760	224.296955	17.651085	998.669332	
389	151.470000	13.500000	977.310000	

	Refined Aggregator	Formulation Duration (hrs)	Compression Strength MPa
453	860.130000	222.720000	61.56
1772	676.340000	18.160000	66.76
850	883.900000	194.220000	36.63
1760	811.832398	174.408504	68.95
389	970.090000	298.780000	71.49

```
sns.distplot(df["Material_Quantity"]) # Normal Distribution
```

C:\Users\dewan\AppData\Local\Temp\ipykernel_17004\3391140938.py:1:
UserWarning:

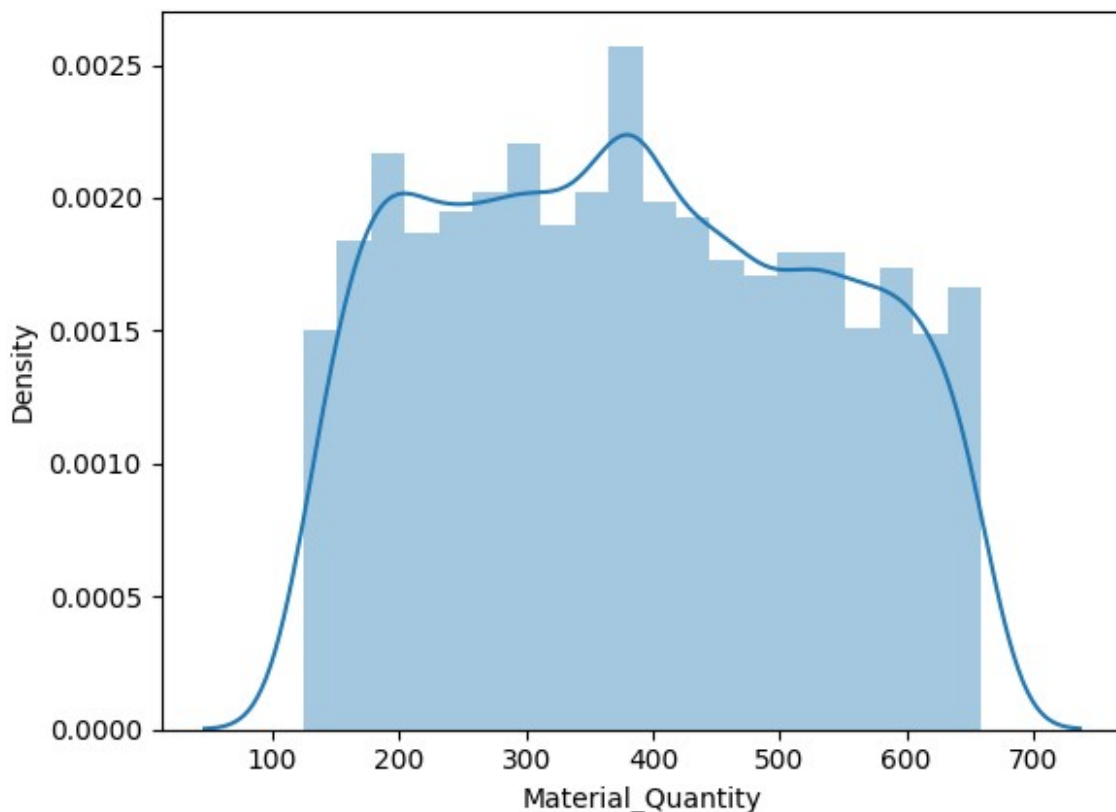
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(df["Material_Quantity"]) # Normal Distribution
```

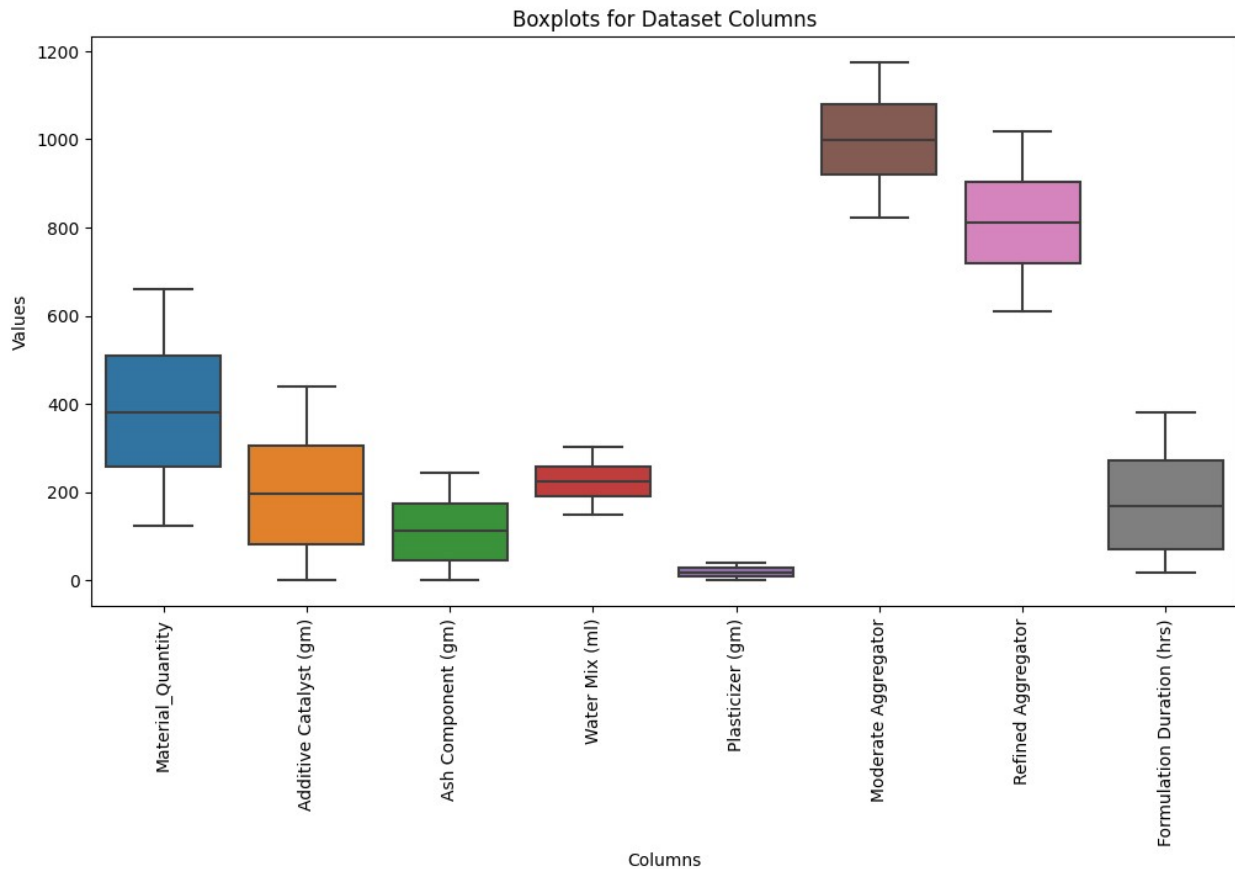
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
<Axes: xlabel='Material_Quantity', ylabel='Density'>
```



```
plt.figure(figsize=(12, 6))
df.drop(columns=['Compression Strength MPa'], inplace=True)
sns.boxplot(data=df, orient="v") # vertical orientation
plt.xticks(rotation=90)
plt.title("Boxplots for Dataset Columns")
plt.xlabel("Columns")
plt.ylabel("Values")
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
```

```
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
```



Above plot clearly shows that dataset has no outliers

```
df.Material_Quantity.mean()  
383.6422968490879  
df.Material_Quantity.std()  
148.65653331503697
```

(1) Outlier detection and removal using 3 standard deviation

```
upper_limit = df.Material_Quantity.mean() +  
3*df.Material_Quantity.std()  
upper_limit  
829.6118967941989
```



```

lower_limit = df.Material_Quantity.mean() -
3*df.Material_Quantity.std()
lower_limit

-62.327303096022945

df[(df.Material_Quantity>upper_limit) |
(df.Material_Quantity<lower_limit)]

Empty DataFrame
Columns: [Material_Quantity, Additive Catalyst (gm), Ash Component
(gm), Water Mix (ml), Plasticizer (gm), Moderate Aggregator, Refined
Aggregator, Formulation Duration (hrs)]
Index: []

df_no_outlier_std_dev =df[(df.Material_Quantity>upper_limit) |
(df.Material_Quantity<lower_limit)]
df_no_outlier_std_dev.head()

Empty DataFrame
Columns: [Material_Quantity, Additive Catalyst (gm), Ash Component
(gm), Water Mix (ml), Plasticizer (gm), Moderate Aggregator, Refined
Aggregator, Formulation Duration (hrs)]
Index: []

df_no_outlier_std_dev.shape

(0, 8)

df.shape

(6139, 8)

```

(2) Outlier detection and removal using Z Score

```

df['zscore'] = ( df.Material_Quantity - df.Material_Quantity.mean() )
/ df.Material_Quantity.std()
df.head(5)

```

	Material_Quantity	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	

	Water Mix (ml)	Plasticizer (gm)	Moderate Aggregator	Refined Aggregator	\
0	201.66	16.11	1151.17		
	708.50				

1	175.99	6.27	1090.57
1010.25			
2	295.23	11.95	1024.93
810.69			
3	299.14	19.00	1134.88
881.34			
4	235.54	17.02	1098.24
781.01			

	Formulation Duration (hrs)	zscore
0	344.43	0.691377
1	28.86	-1.683897
2	237.68	1.186142
3	208.81	0.052387
4	266.84	0.074922

Above for first record with Material_Quantity 486.42, z score is 0.68. This means 486.42 is 0.68 standard deviation away from mean

```
df[df['zscore']>3]
```

Empty DataFrame

Columns: [Material_Quantity, Additive Catalyst (gm), Ash Component (gm), Water Mix (mL), Plasticizer (gm), Moderate Aggregator, Refined Aggregator, Formulation Duration (hrs), zscore]

Index: []

```
df[df['zscore']<-3]
```

Empty DataFrame

Columns: [Material_Quantity, Additive Catalyst (gm), Ash Component (gm), Water Mix (mL), Plasticizer (gm), Moderate Aggregator, Refined Aggregator, Formulation Duration (hrs), zscore]

Index: []

```
df[(df.zscore<-3) | (df.zscore>3)]
```

Empty DataFrame

Columns: [Material_Quantity, Additive Catalyst (gm), Ash Component (gm), Water Mix (mL), Plasticizer (gm), Moderate Aggregator, Refined Aggregator, Formulation Duration (hrs), zscore]

Index: []

```
df_no_outliers = df[(df.zscore>-3) & (df.zscore<3)]
```

```
df_no_outliers.head()
```

	Material_Quantity	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	

	Water Mix (ml)	Plasticizer (gm)	Moderate Aggregator	Refined
4	394.78		352.61	194.35
Aggregator \				
0	201.66	16.11	1151.17	
708.50				
1	175.99	6.27	1090.57	
1010.25				
2	295.23	11.95	1024.93	
810.69				
3	299.14	19.00	1134.88	
881.34				
4	235.54	17.02	1098.24	
781.01				

	Formulation Duration (hrs)	zscore
0	344.43	0.691377
1	28.86	-1.683897
2	237.68	1.186142
3	208.81	0.052387
4	266.84	0.074922


```
df_no_outliers.shape
(6139, 9)
df.shape
(6139, 9)
```

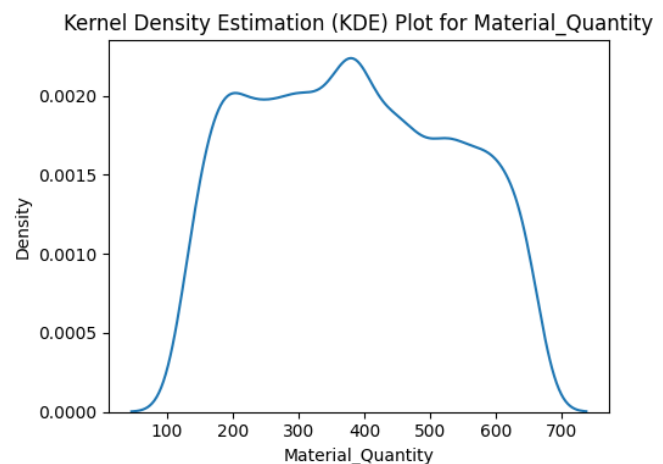
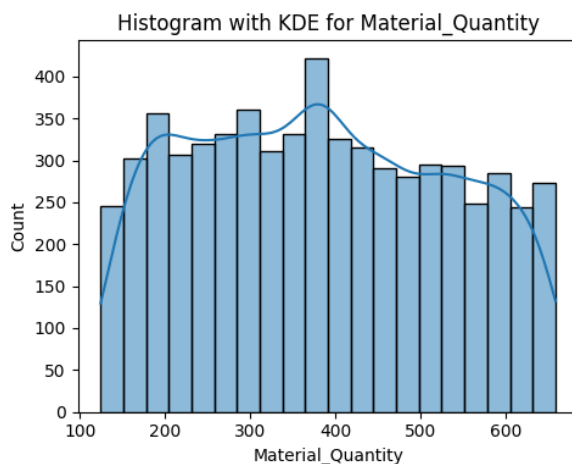
Above shows original dataframe data 6139 data points. Out of that we removed 109 outliers (i.e. 6139-6030)

5. With the help of graph check the skeweness of the dataset

```
df.drop(columns=['zscore'], inplace=True)
for column in df.columns:
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    sns.histplot(df[column], kde=True) #Histogram with KDE
    plt.title(f"Histogram with KDE for {column}")
    plt.subplot(1, 2, 2)
    sns.kdeplot(df[column]) # KDE plot
    plt.title(f"Kernel Density Estimation (KDE) Plot for {column}")
    plt.tight_layout()
    plt.show()
```

```
skewness = df[column].skew()
print(f"Skewness for {column}: {skewness}")
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
```



Skewness for Material_Quantity: 0.09660470518509962

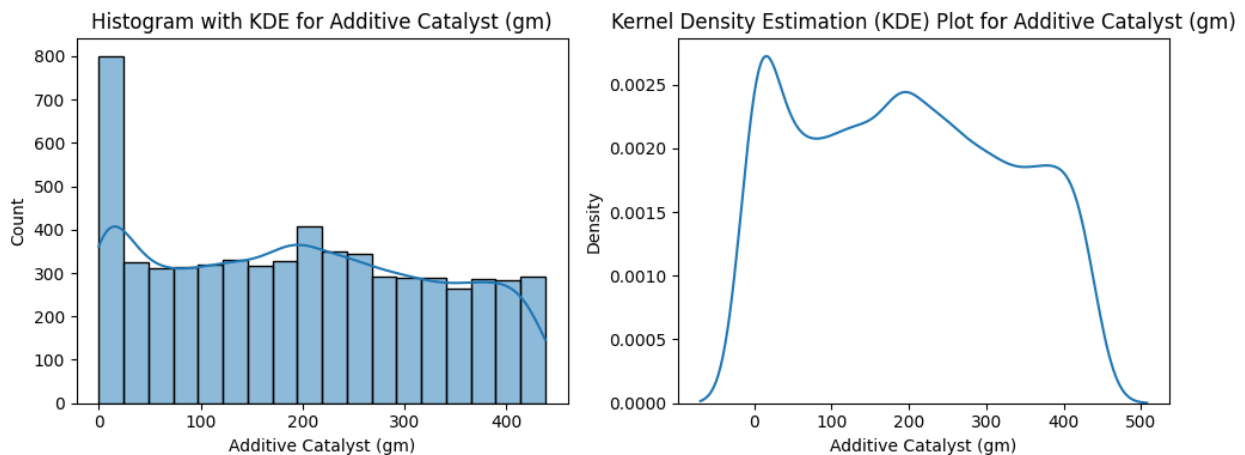
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
```



```

in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):

```

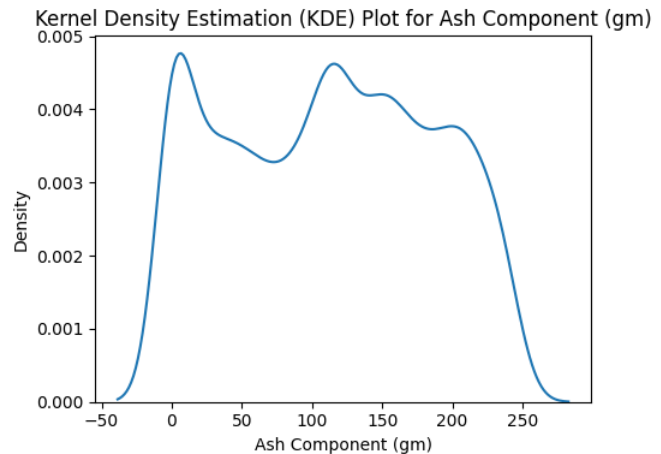
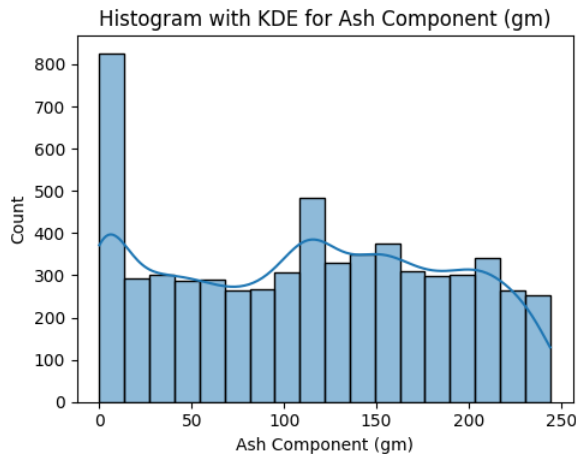


Skewness for Additive Catalyst (gm): 0.10758430815567124

```

C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):

```



Skewness for Ash Component (gm): -0.0012241862673132268

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```

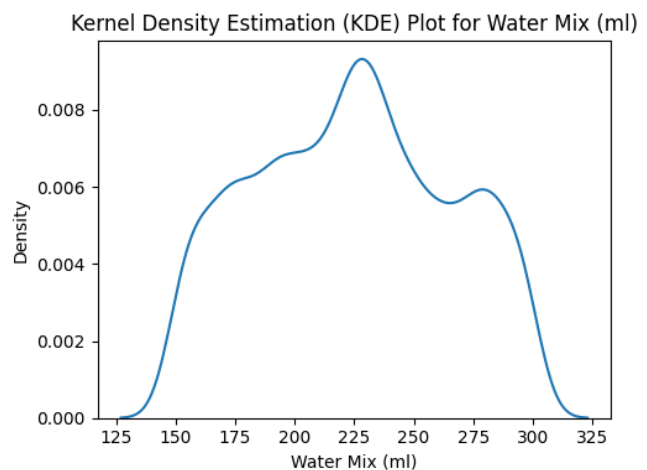
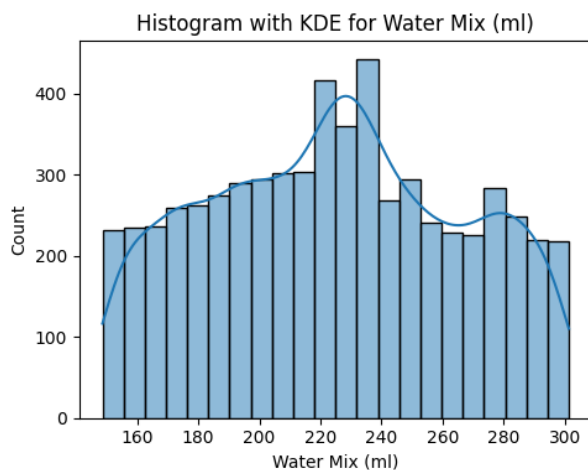
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
```

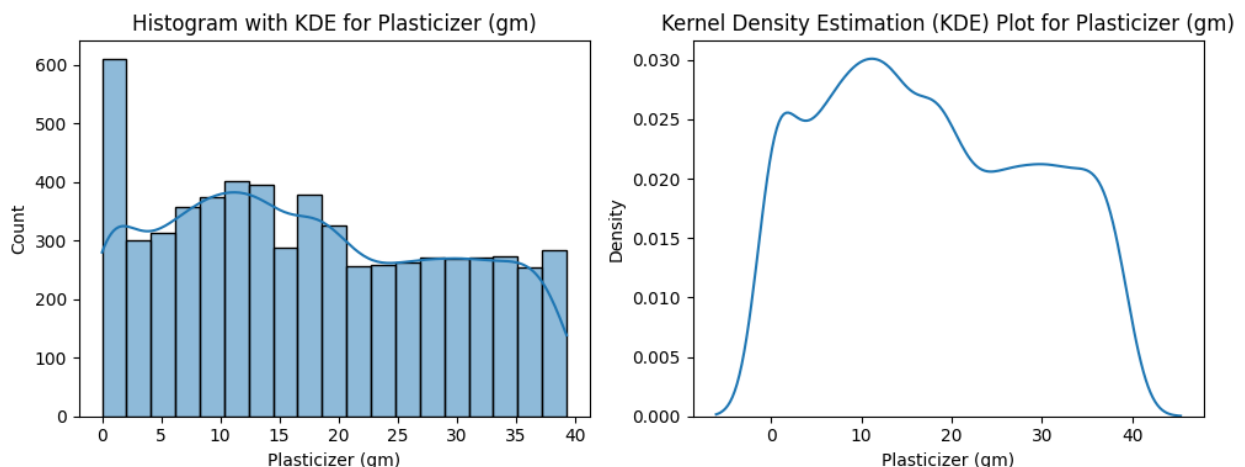
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):
```



Skewness for Water Mix (ml): 0.02495322530123565

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
```



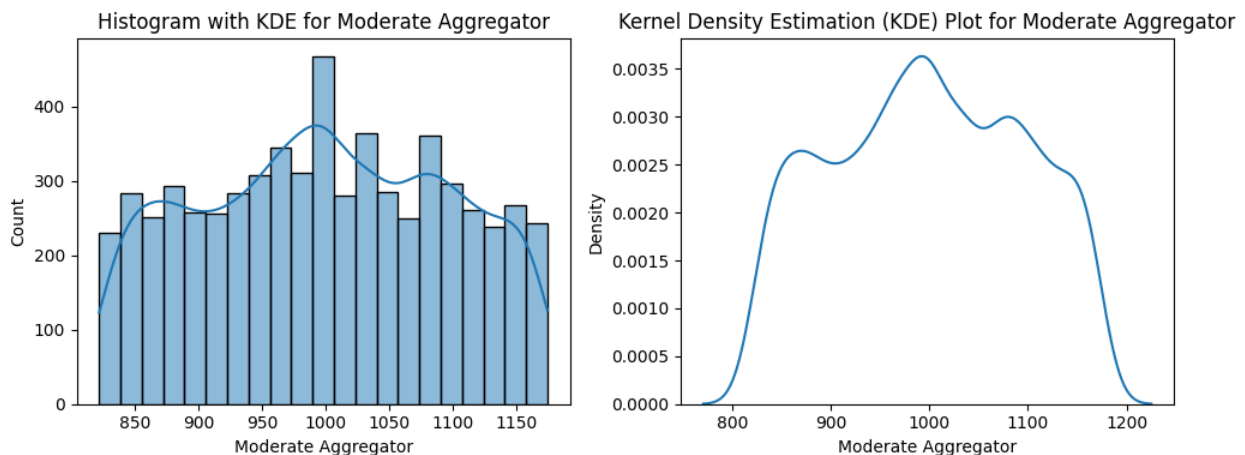
Skewness for Plasticizer (gm): 0.18284172041829327

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
```

```

in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):

```

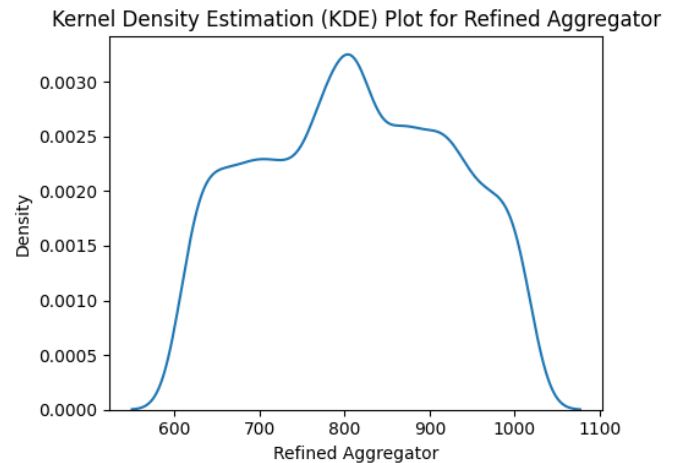
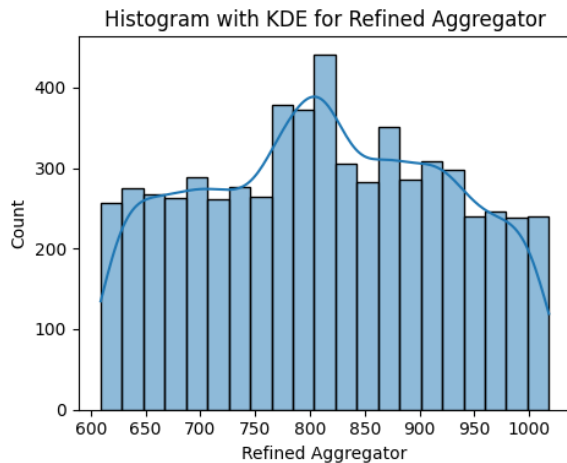


Skewness for Moderate Aggregator: -0.02058214455764159

```

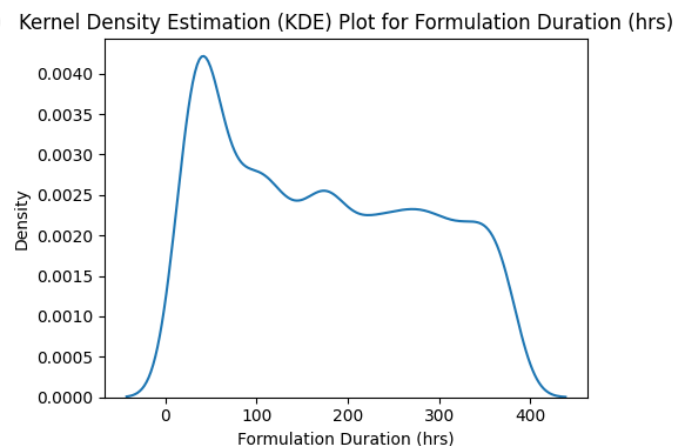
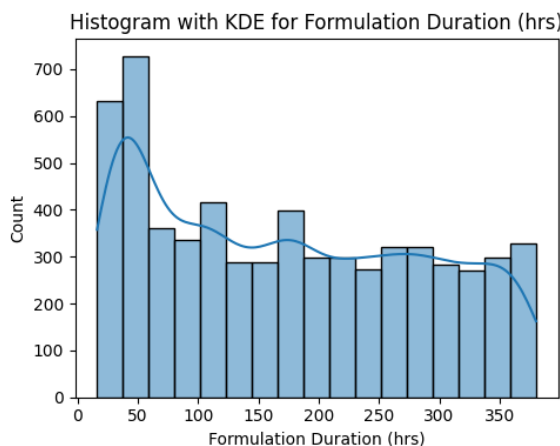
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):

```

Skewness for Refined Aggregator: -0.006749187304419871

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```

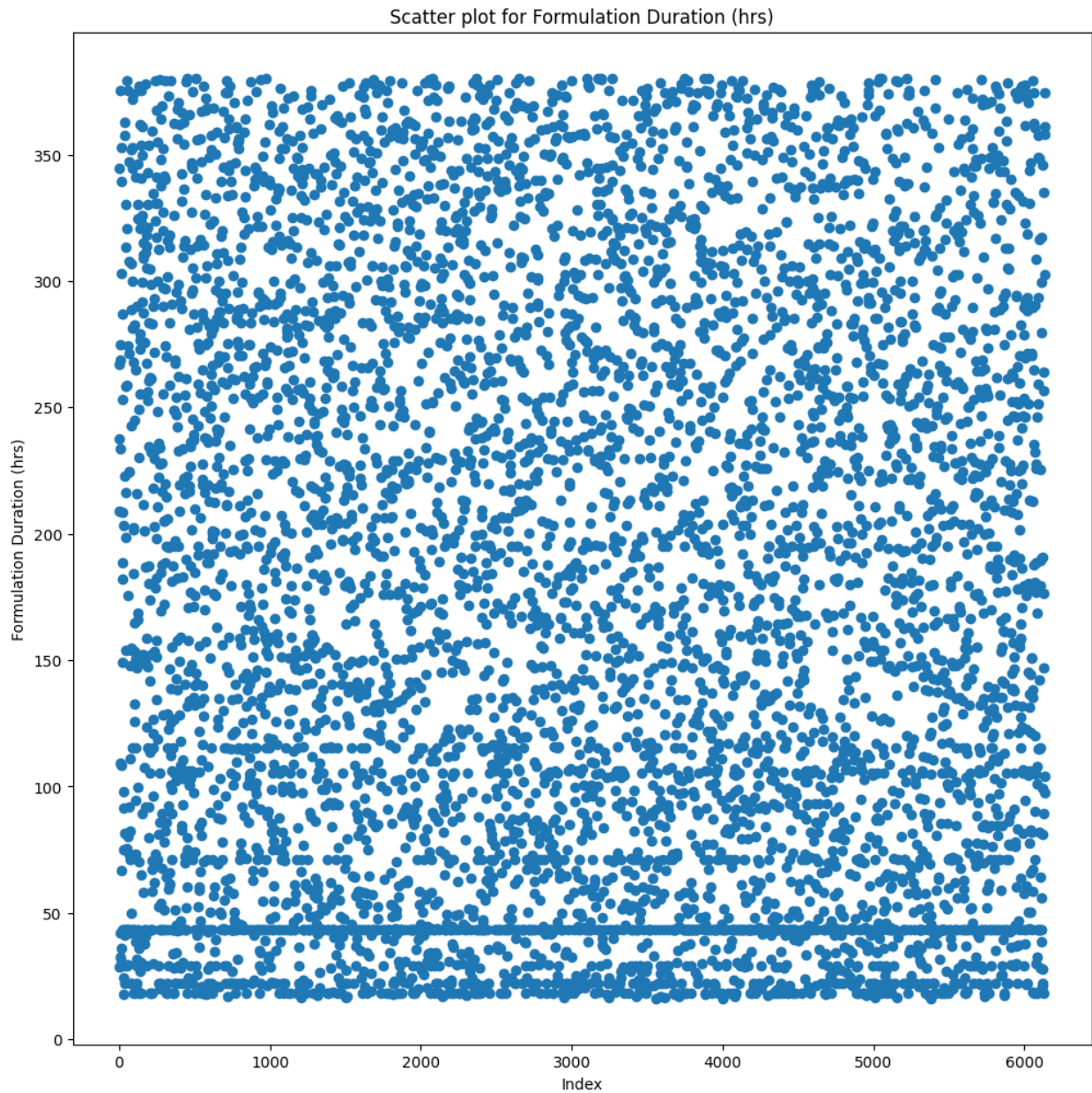


Skewness for Formulation Duration (hrs): 0.23329048470835131

Graph showing some features are Left skewed(Negatively Skewed) and some are Right skewed(Positively skewed)

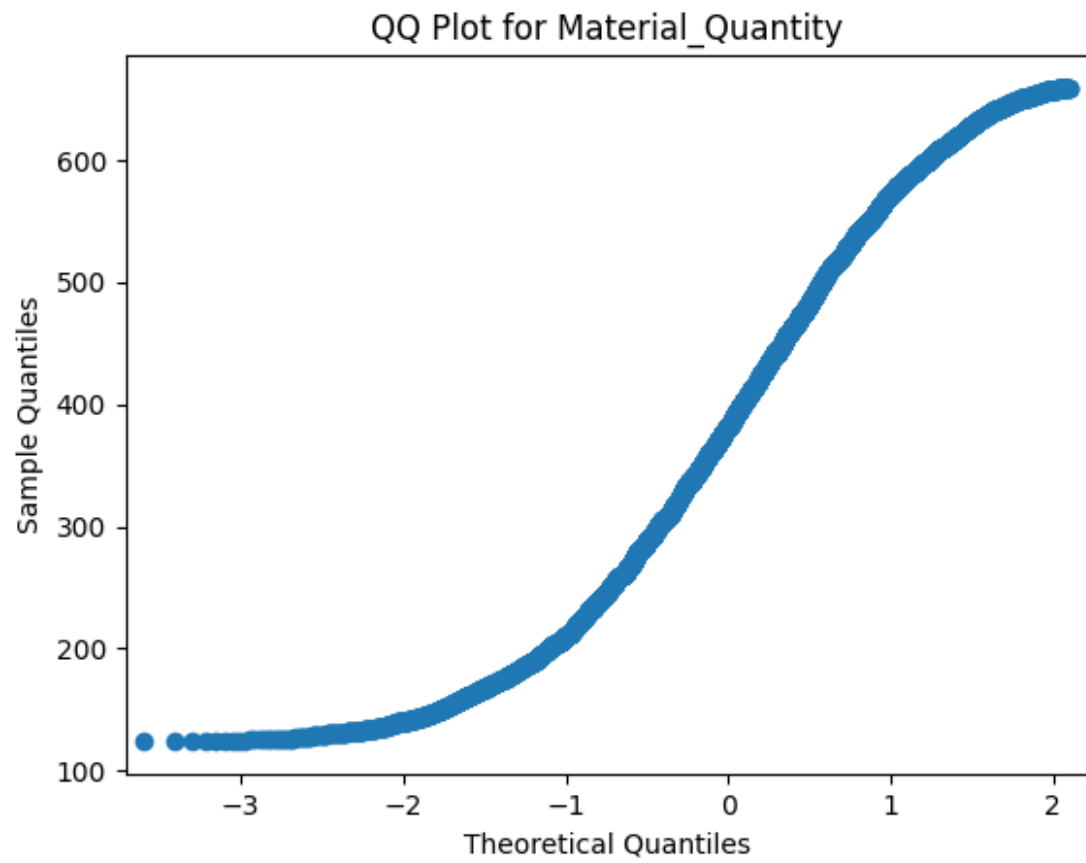
6. Check distribution of each independent variables using scatter plot and also using QQ plot to understand column distribution.

```
#Using Scatter plot
#for column in df.columns[:-1]:
df = pd.read_csv("Material Compressive Strength Experimental Data
(1).csv")
plt.figure(figsize=(12, 12))
plt.scatter(df.index, df[column])
plt.xlabel("Index")
plt.ylabel(column)
plt.title(f'Scatter plot for {column}')
plt.show()
```

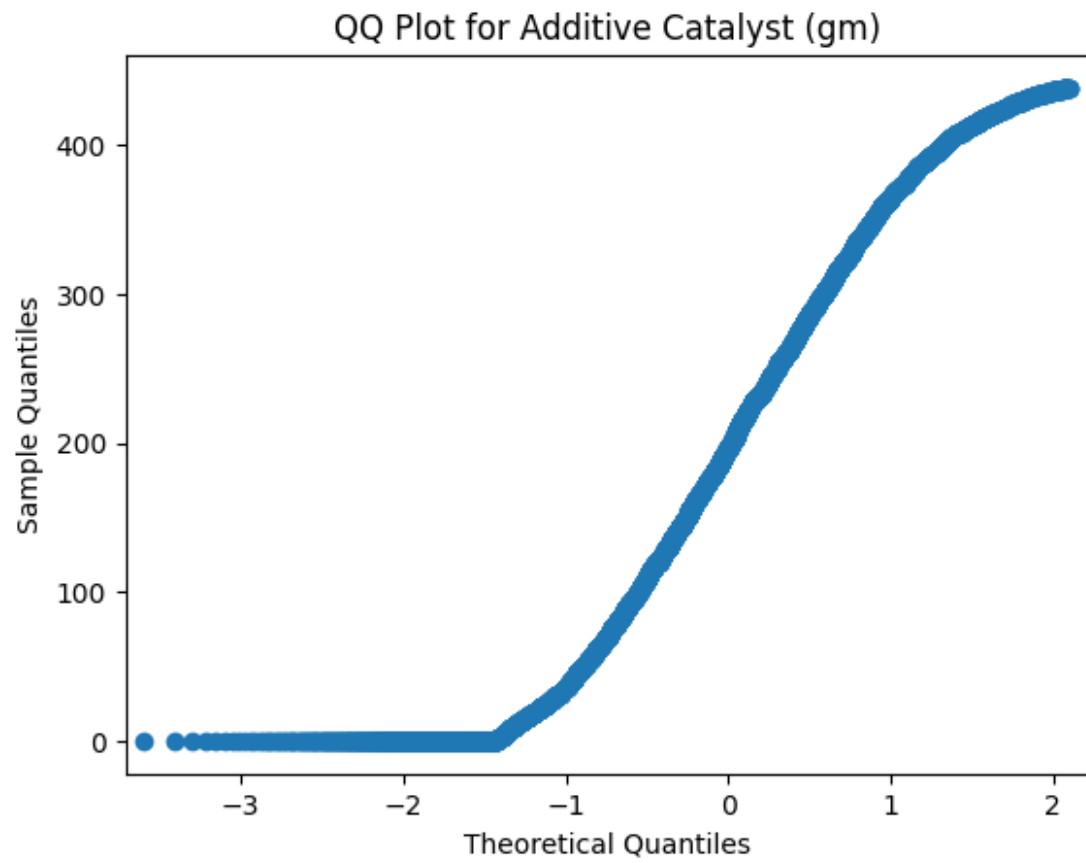


```
#Using QQ plot
import statsmodels.api as sm
for column in df.columns[:-1]:
    plt.figure(figsize=(8, 4))
    sm.qqplot(df[column], line='s')
    plt.title(f'QQ Plot for {column}')
    plt.show()
```

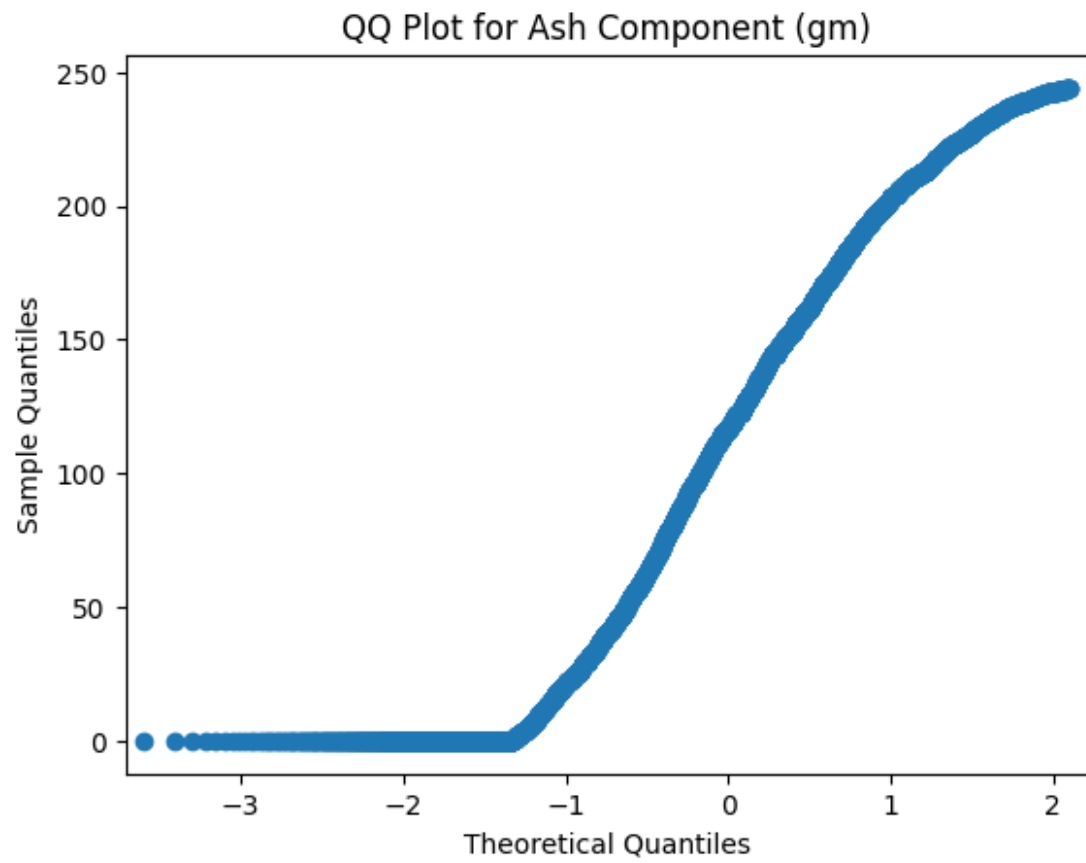
<Figure size 800x400 with 0 Axes>



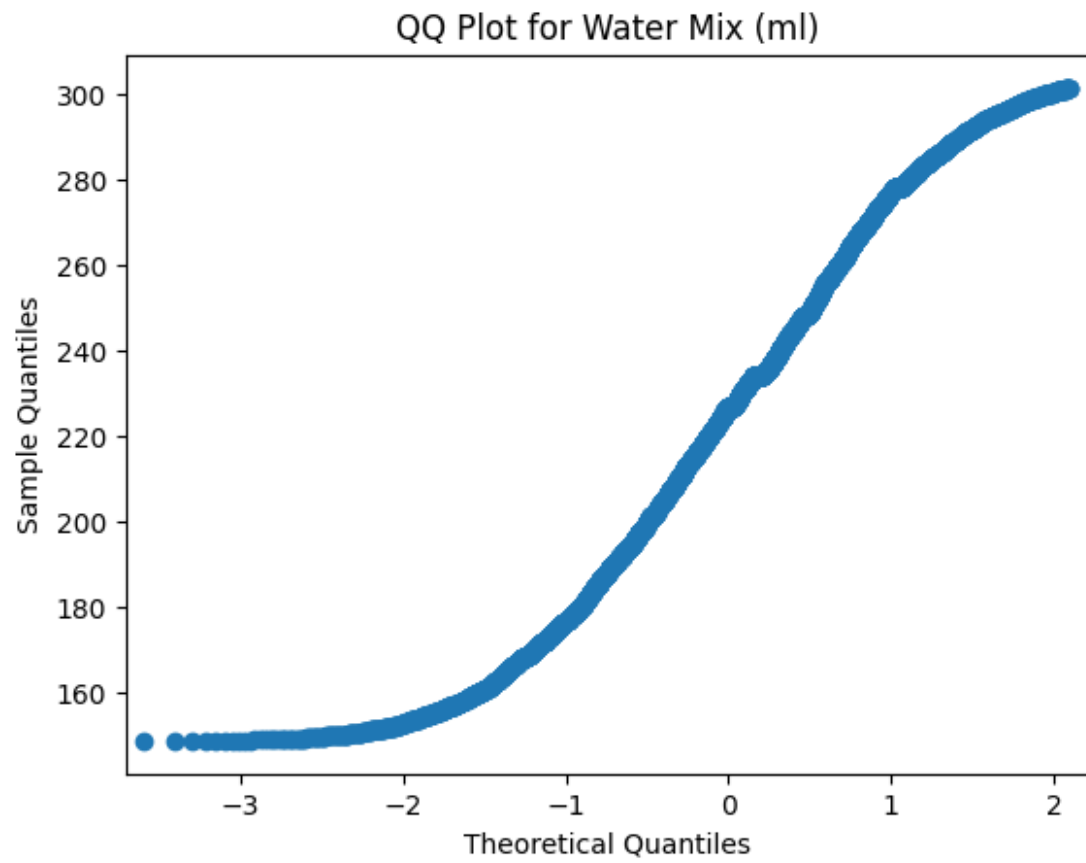
<Figure size 800x400 with 0 Axes>



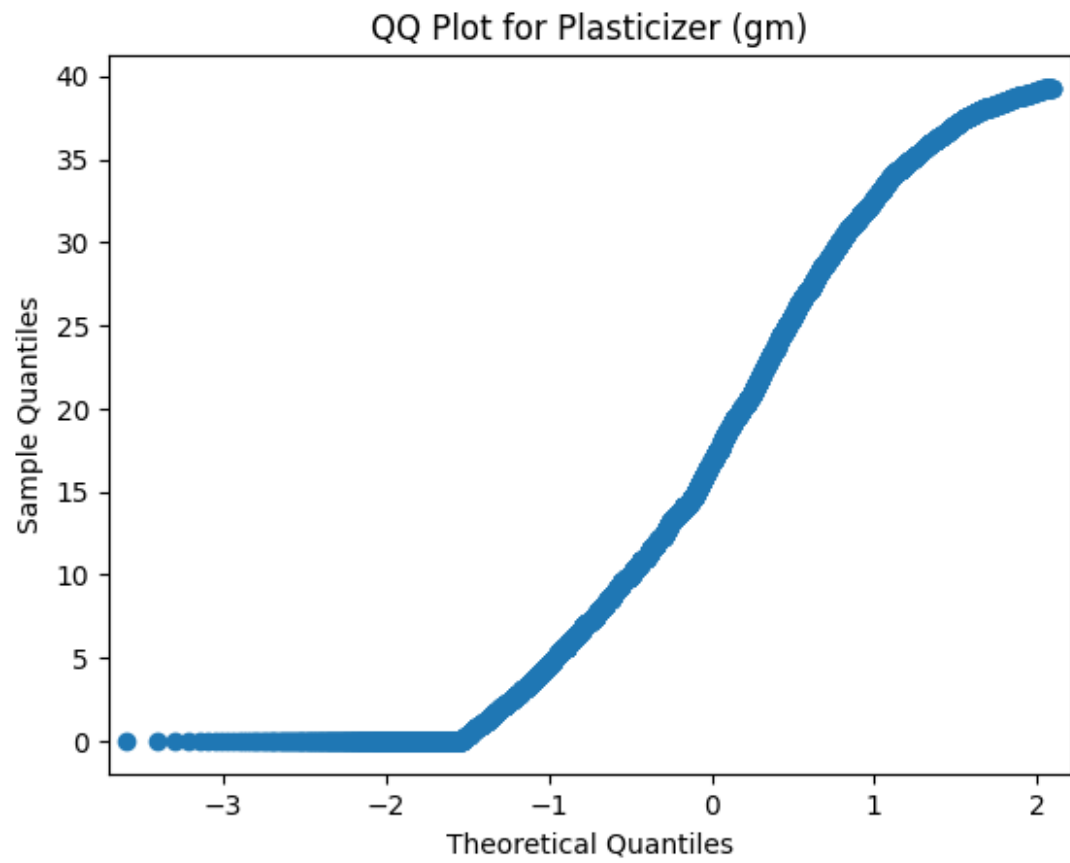
<Figure size 800x400 with 0 Axes>



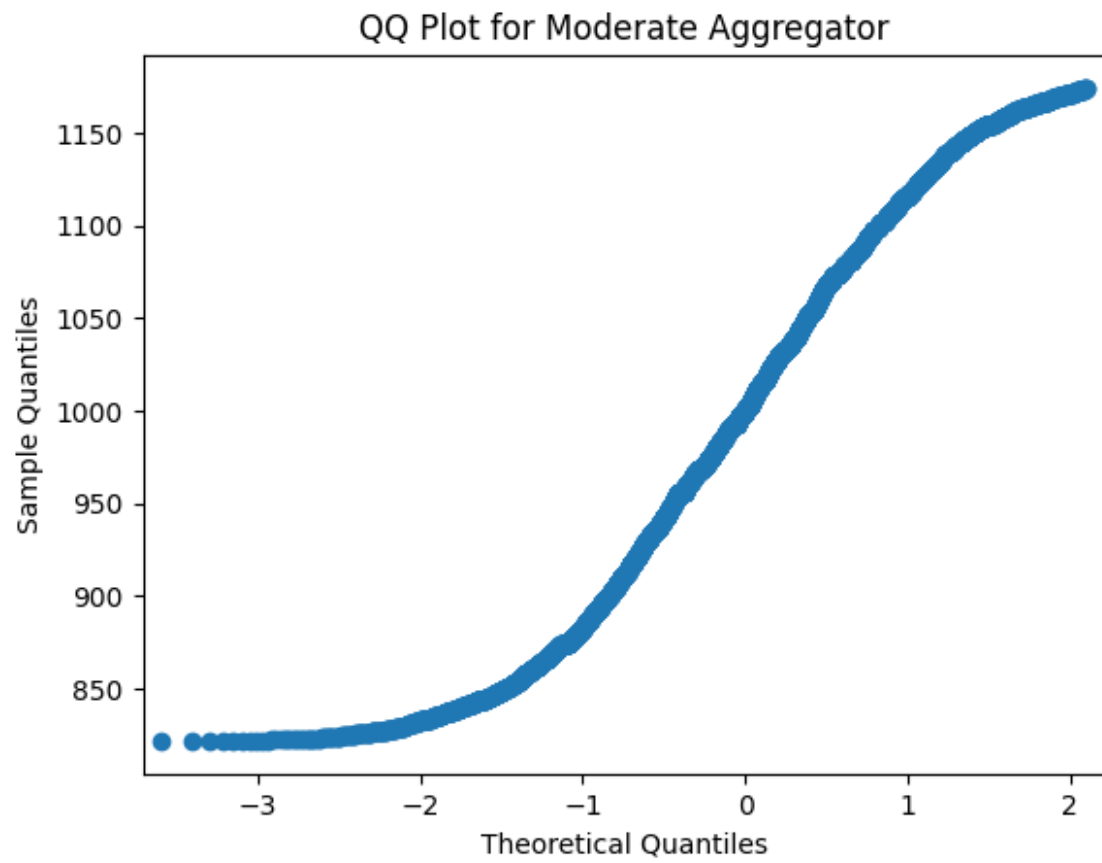
<Figure size 800x400 with 0 Axes>



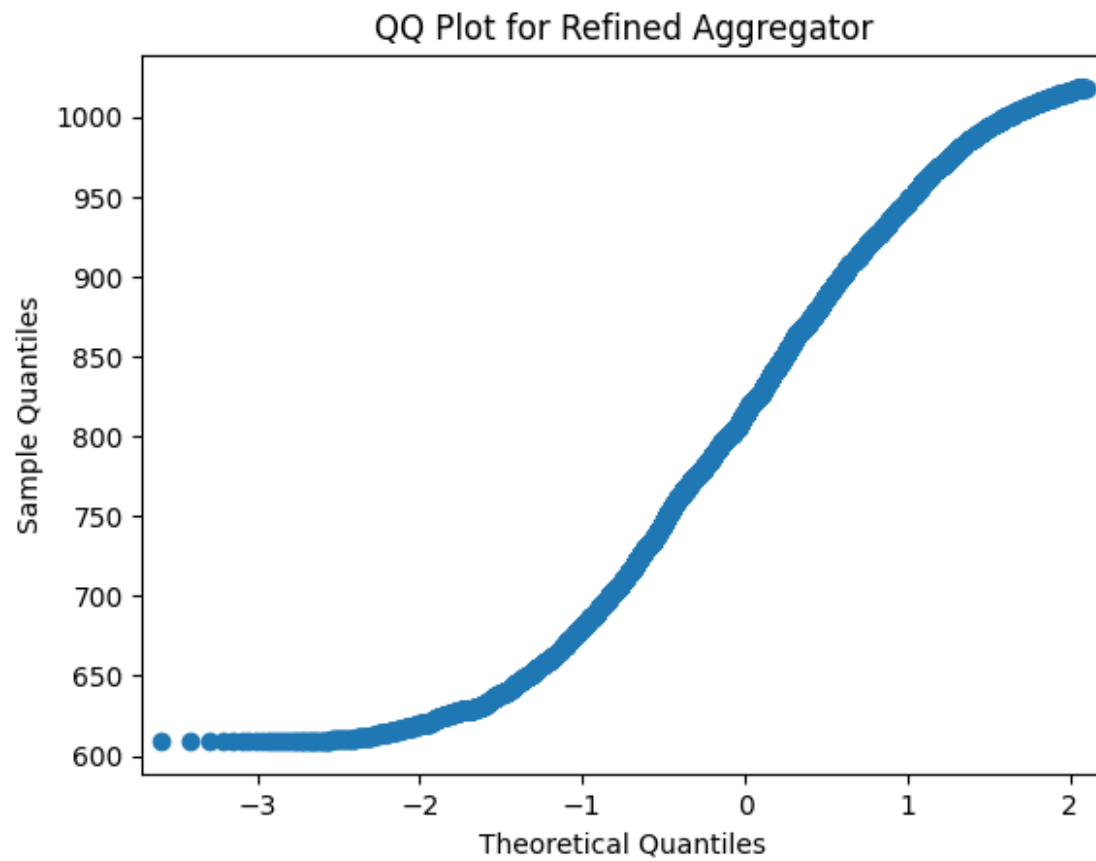
<Figure size 800x400 with 0 Axes>



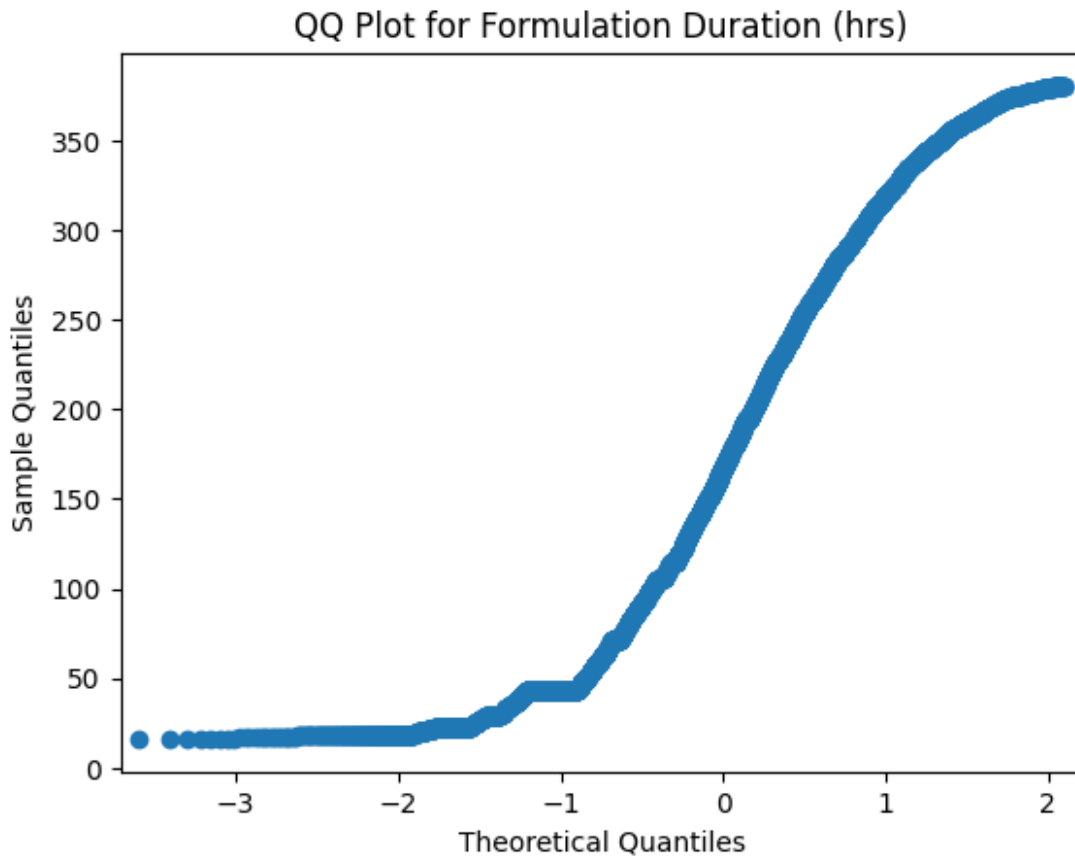
<Figure size 800x400 with 0 Axes>



<Figure size 800x400 with 0 Axes>



<Figure size 800x400 with 0 Axes>



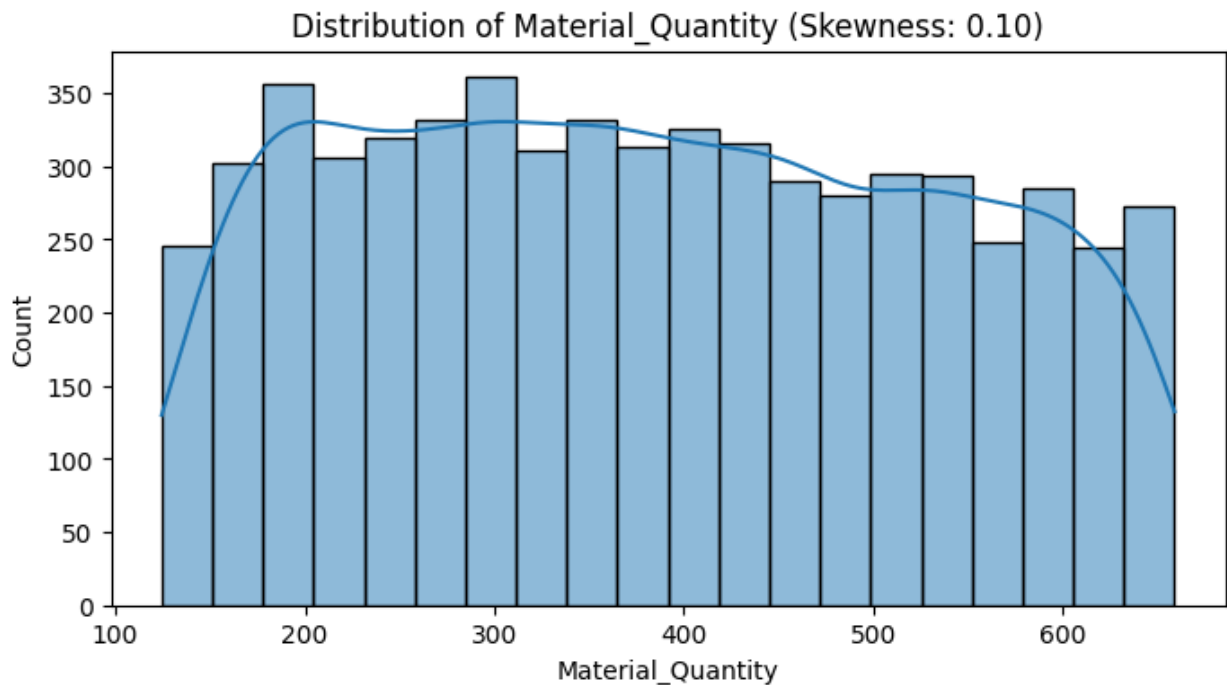
Plots showing almost all the features are exponentially increases.

7. Plotting the Dataset Distribution to check every density and skewness.

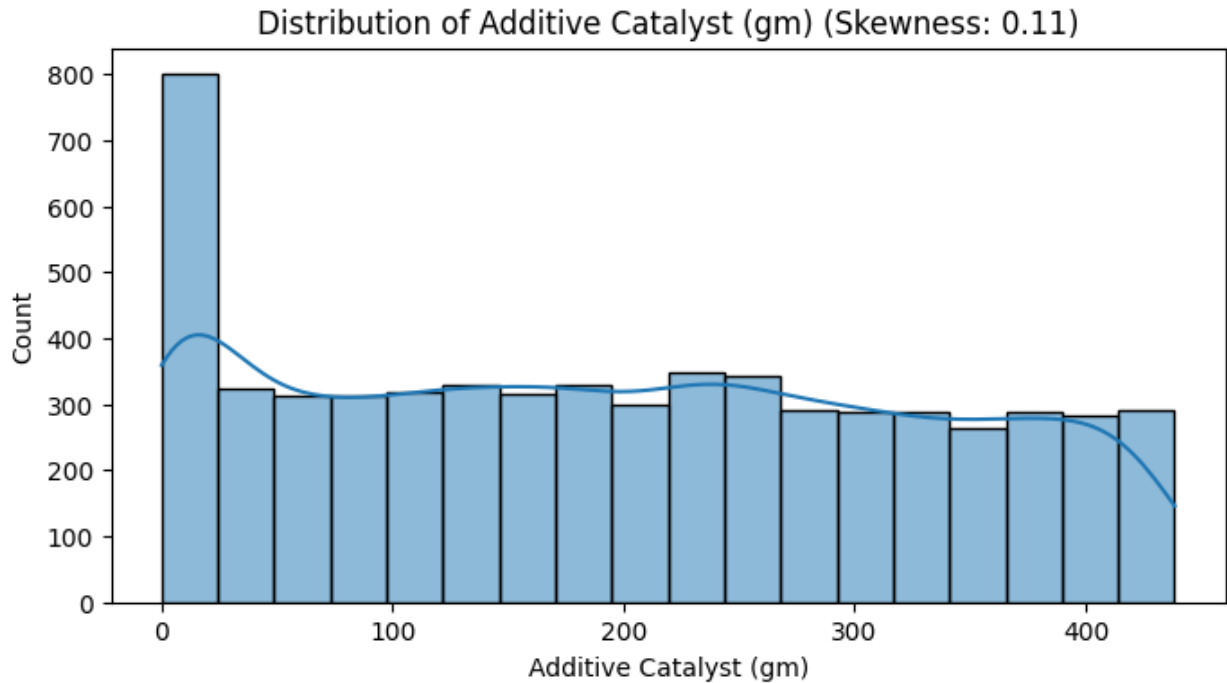
```
independent_vars = df.columns[:-1]
for col in independent_vars:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], kde=True)
    skewness = df[col].skew()
    plt.title(f'Distribution of {col} (Skewness: {skewness:.2f})')
    plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
```

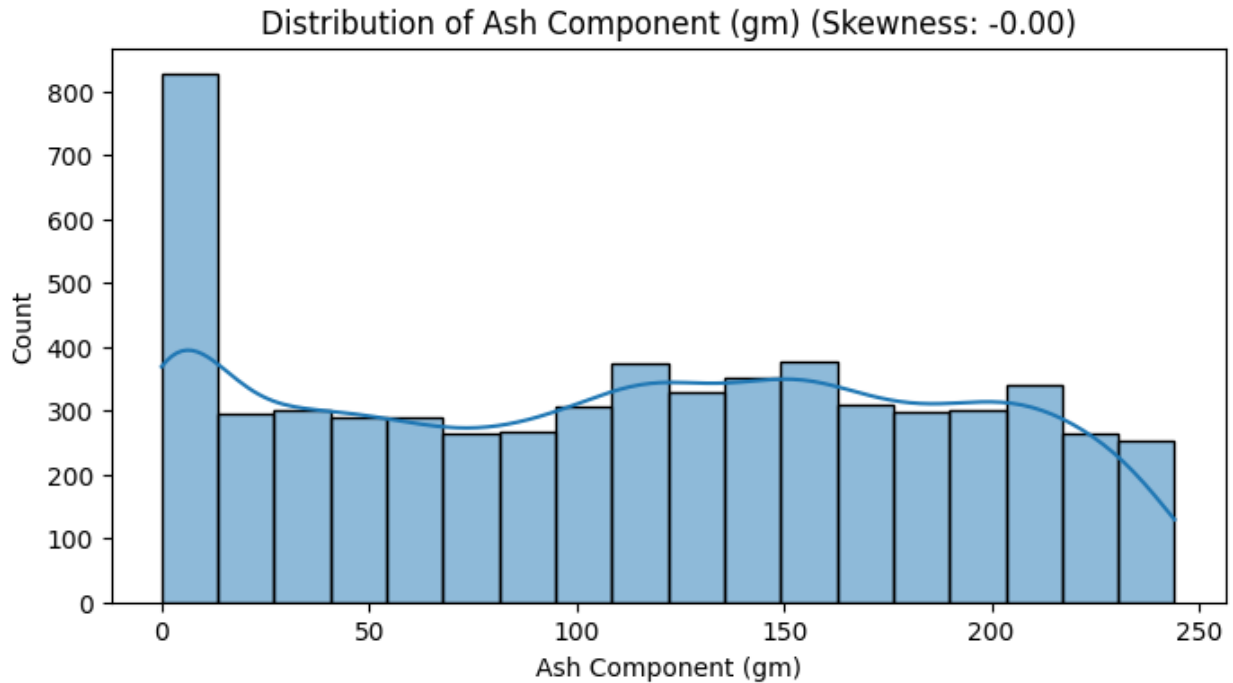
```
instead.  
with pd.option_context('mode.use_inf_as_na', True):
```



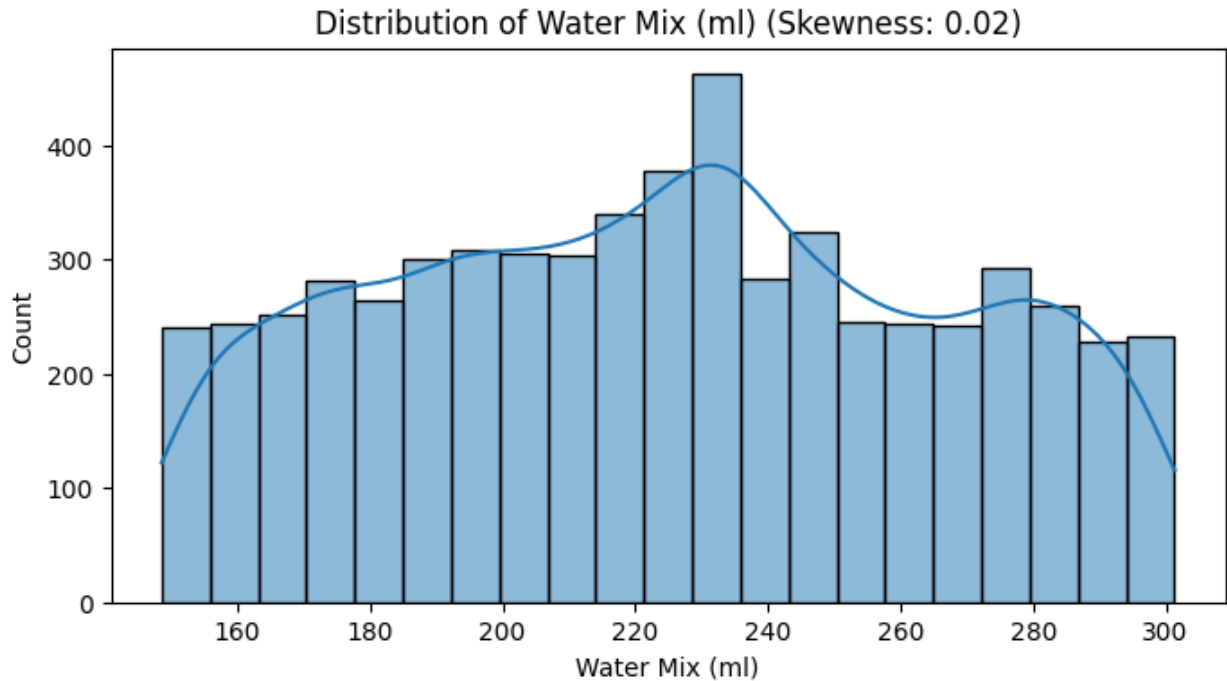
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:  
FutureWarning: is_categorical_dtype is deprecated and will be removed  
in a future version. Use isinstance(dtype, CategoricalDtype) instead  
if pd.api.types.is_categorical_dtype(vector):  
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:  
FutureWarning: use_inf_as_na option is deprecated and will be removed  
in a future version. Convert inf values to NaN before operating  
instead.  
with pd.option_context('mode.use_inf_as_na', True):
```



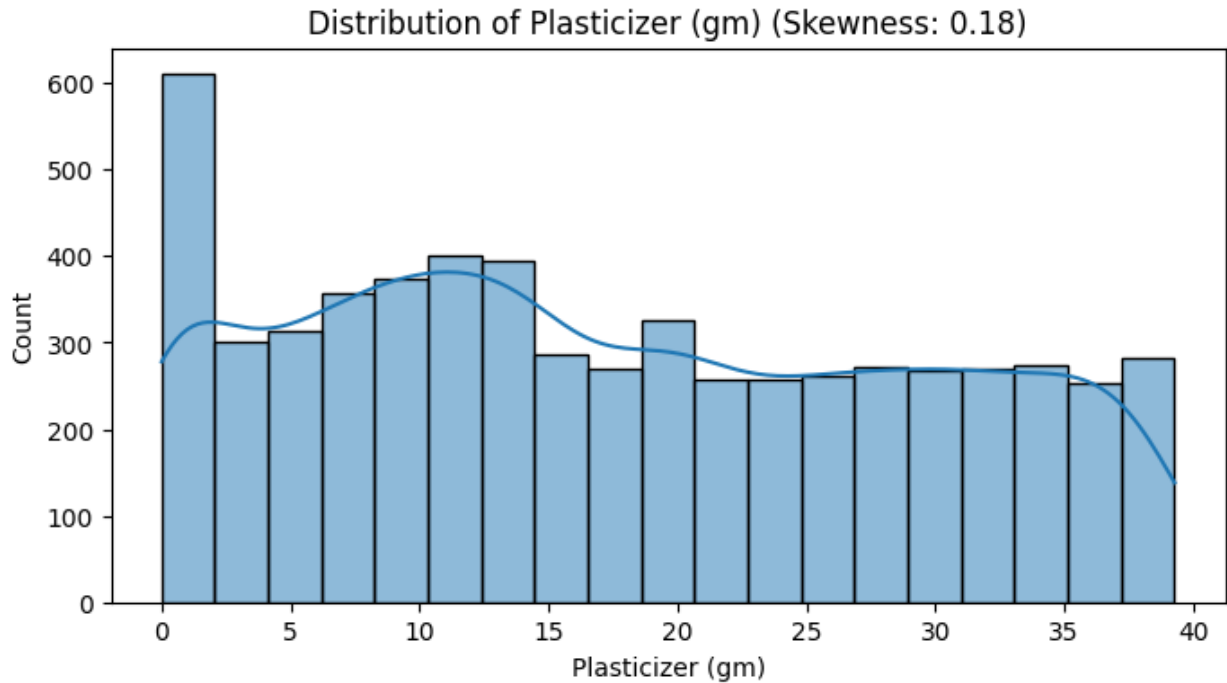
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```



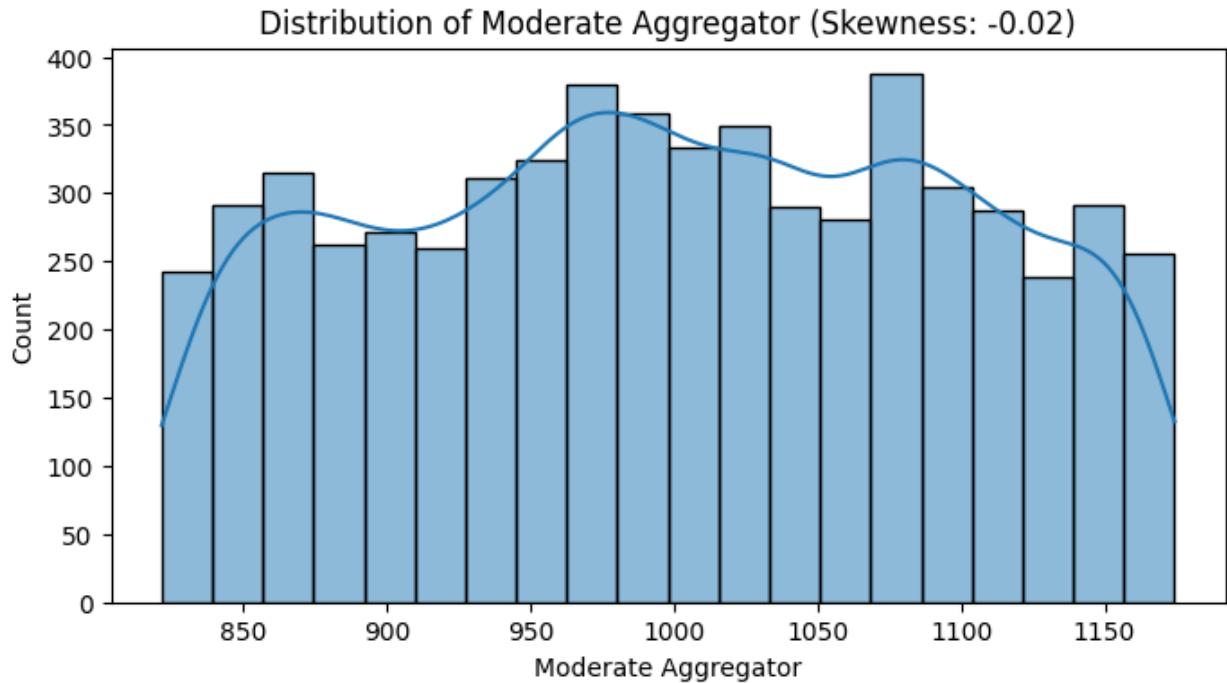
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```



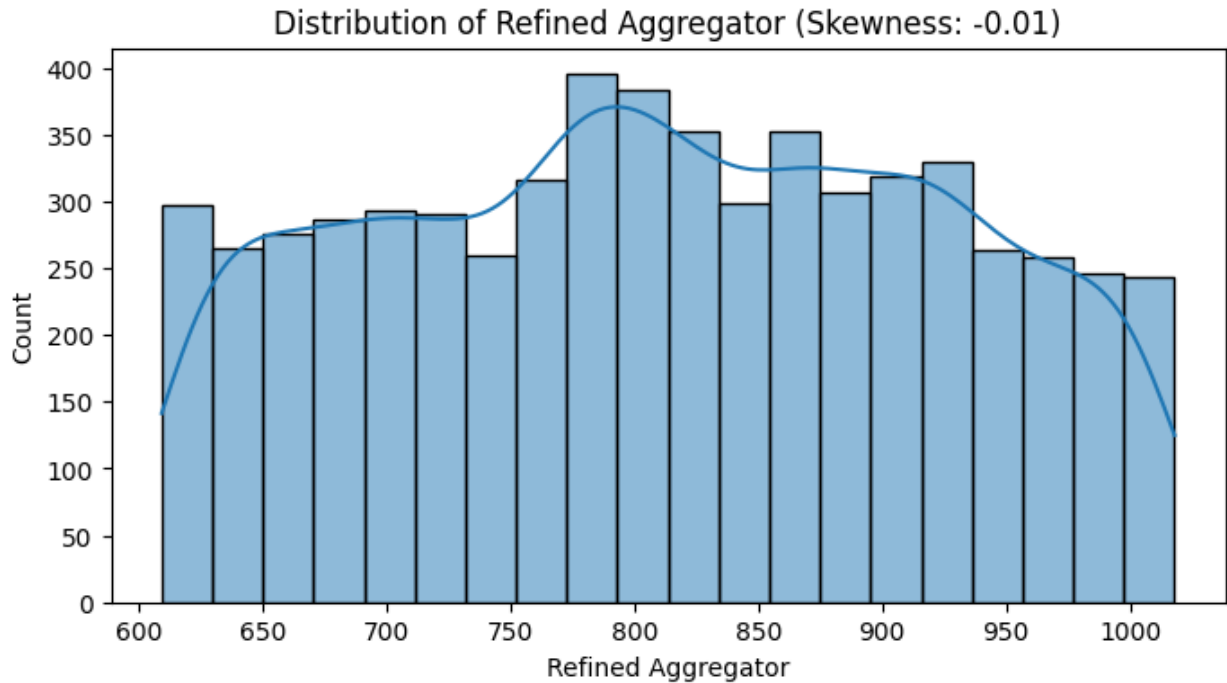
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```

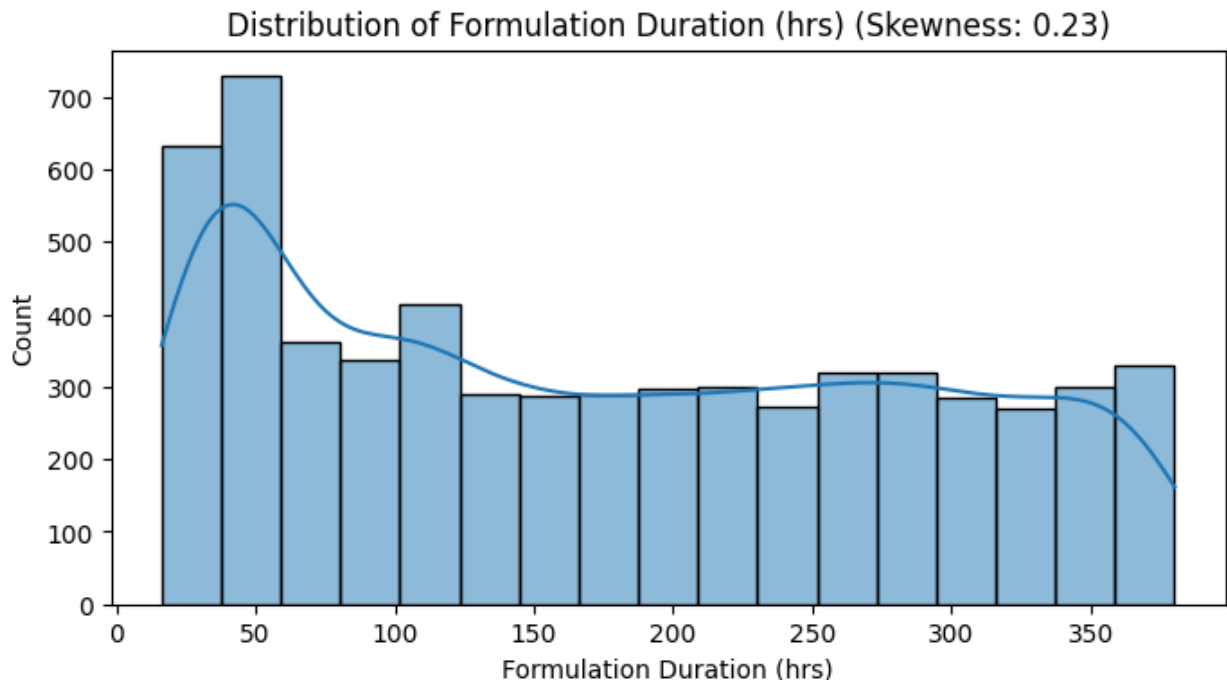
```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```



```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```



```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
with pd.option_context('mode.use_inf_as_na', True):
```



Plot showing how much is the density of any datapoints of all the independent variable on which how skewness lies

8. Scaling the Dataset.

```
from sklearn.preprocessing import StandardScaler
features_to_scale = df.columns
# Initializing the StandardScaler
scaler = StandardScaler()
# Scaling happening
df[features_to_scale] = scaler.fit_transform(df[features_to_scale])
```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```
if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():
```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```
if is_sparse(pd_dtype):
```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:767: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

```
df[features_to_scale] # Scaled columns
```

	Material_Quantity	Additive Catalyst (gm)	Ash Component (gm)	\
0	0.685267	-0.120763	-1.220399	
1	-1.669017	0.475855	0.993383	
2	1.175660	-1.454114	-0.001297	
3	0.051924	1.157758	-0.477757	
4	0.074260	1.169459	1.111252	
...	
6134	-1.299239	-0.258028	0.414815	
6135	-0.225176	0.710707	-0.458494	
6136	-0.169036	-1.305147	-1.264448	
6137	0.410768	0.591743	0.902590	
6138	1.177394	0.524011	0.744713	

	Water Mix (ml)	Plasticizer (gm)	Moderate Aggregator	\
0	-0.544913	-0.131863	1.560515	
1	-1.162838	-0.973825	0.940405	
2	1.707489	-0.487814	0.268721	
3	1.801610	0.115420	1.393822	
4	0.270641	-0.053999	1.018891	
...	
6134	-1.459643	-0.142987	0.052706	
6135	-0.867476	0.698975	-0.752823	
6136	-0.378336	1.476762	0.843193	
6137	-0.782984	0.035845	-1.366282	
6138	-1.174874	-0.600760	1.710938	

	Refined Aggregator	Formulation Duration (hrs)	Compression Strength MPa
0	-0.916033	1.512568	
1.428871			

1	1.758956	-1.294848	
0.182873			
2	-0.010127	0.562884	
1.302969			
3	0.616179	0.306047	
0.923401			
4	-0.273238	0.822301	
1.191951			
...	
...			
6134	1.689987	1.632491	-
0.387099			
6135	1.715163	-0.624598	-
0.161963			
6136	-0.171912	1.141858	-
0.017455			
6137	0.188536	1.781237	
0.084260			
6138	0.733108	1.659624	
0.130775			

[6139 rows x 9 columns]

Now the dataset is all scaled to get in all same range/ scale.

9. If columns are not gaussian distributed then make it normal distribution.

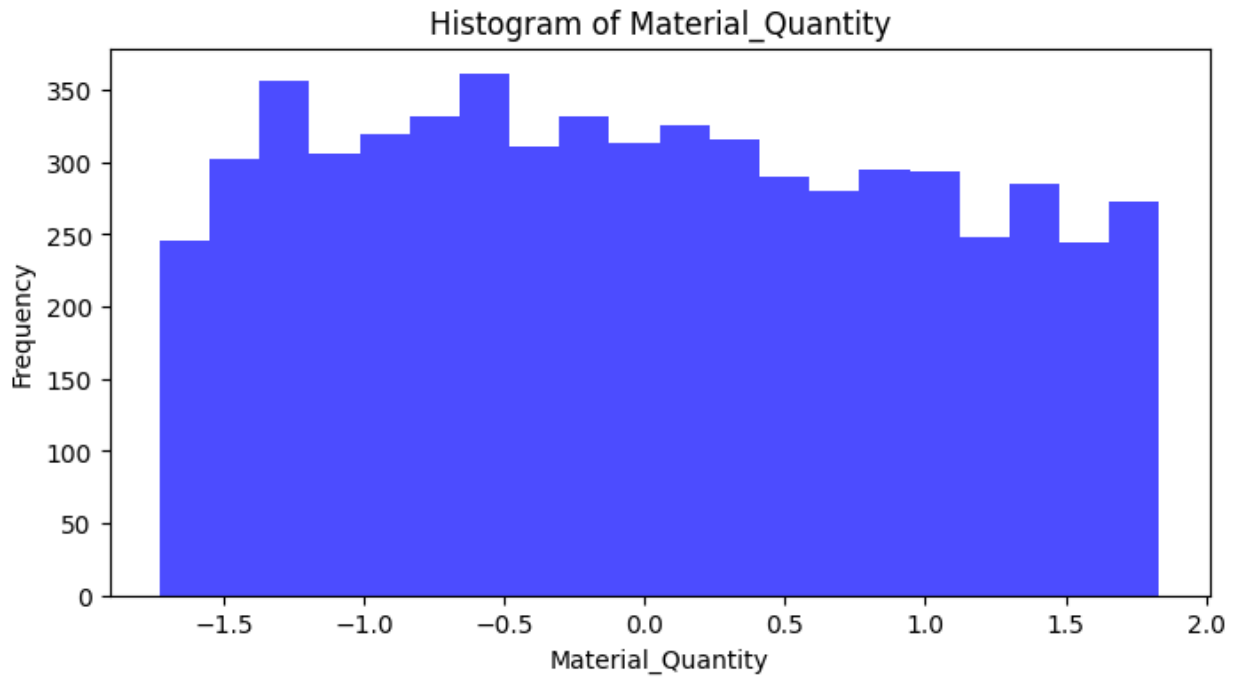
```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
for feature in features_to_scale:
    data = df[feature]
    stat, p = stats.shapiro(data)
    alpha = 0.05
    if p > alpha:
        print(f'{feature} is normally distributed (p-value = {p:.4f})')
    else:
        print(f'{feature} is not normally distributed (p-value = {p:.4f})')
plt.figure(figsize=(8, 4))
plt.hist(data, bins=20, color='blue', alpha=0.7)
plt.title(f'Histogram of {feature}')
plt.xlabel(feature)
plt.ylabel('Frequency')
plt.show()
```



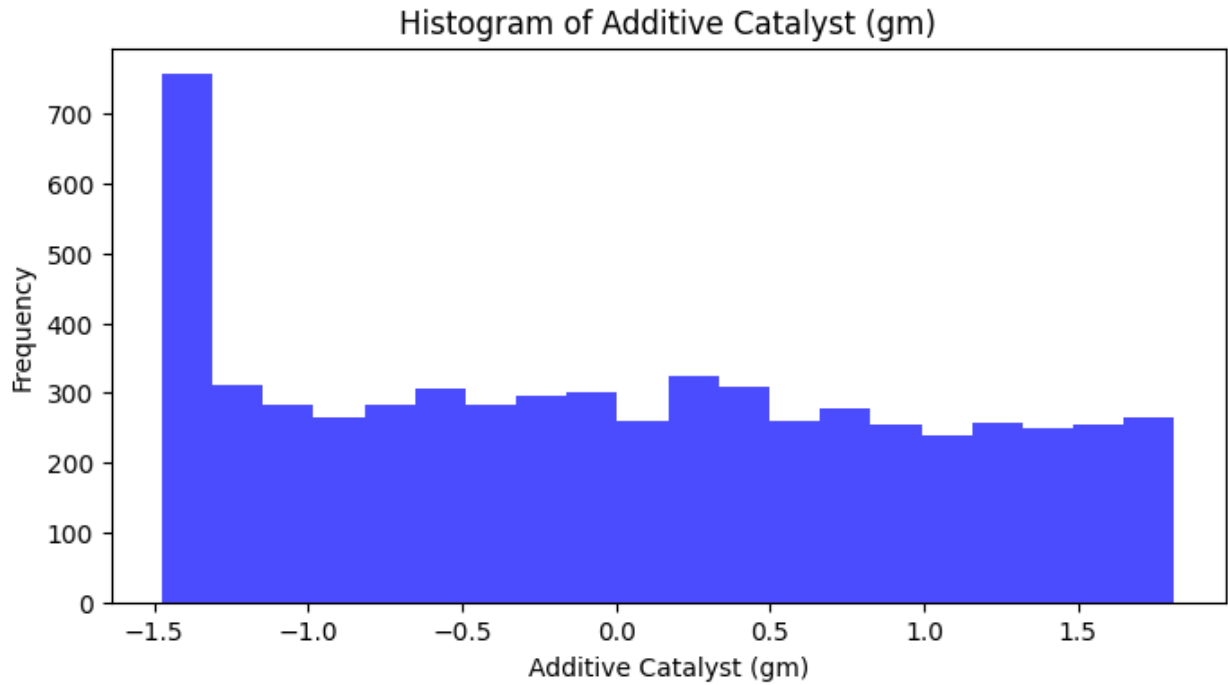
```
C:\Users\dewan\anaconda3\lib\site-packages\scipy\stats\
_morestats.py:1816: UserWarning: p-value may not be accurate for N >
5000.
```

```
warnings.warn("p-value may not be accurate for N > 5000.")
```

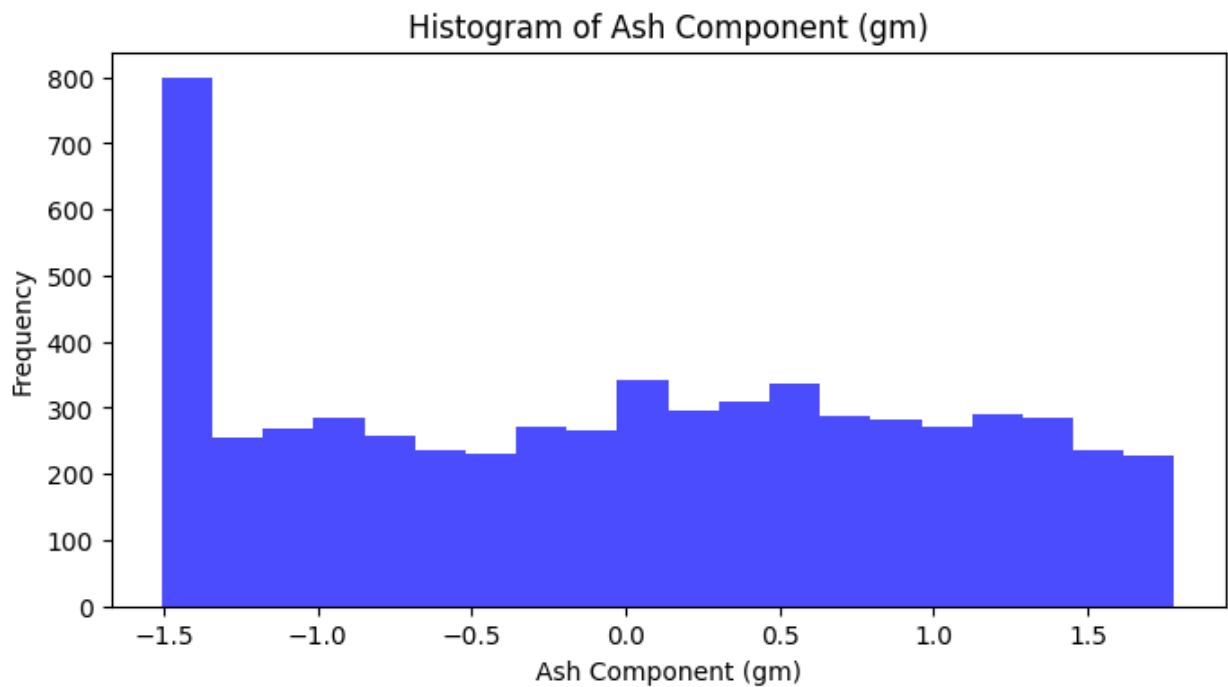
```
Material_Quantity is normally distributed (p-value = 1.0000)
```



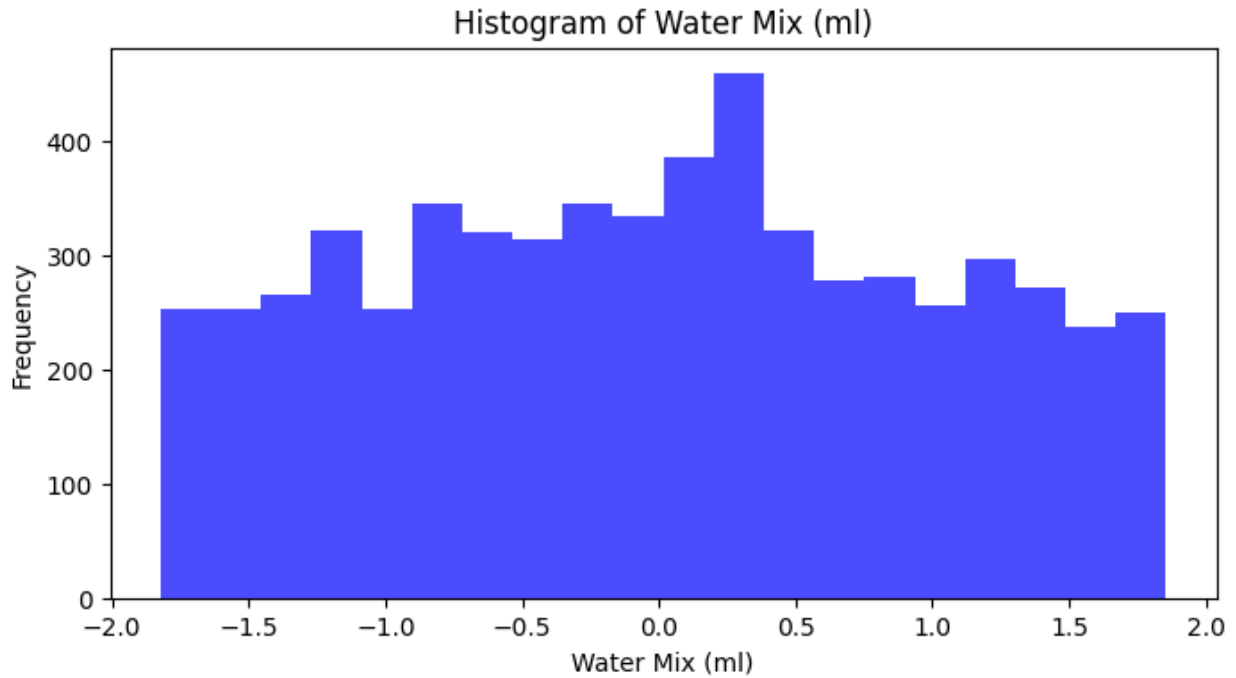
```
Additive Catalyst (gm) is normally distributed (p-value = 1.0000)
```



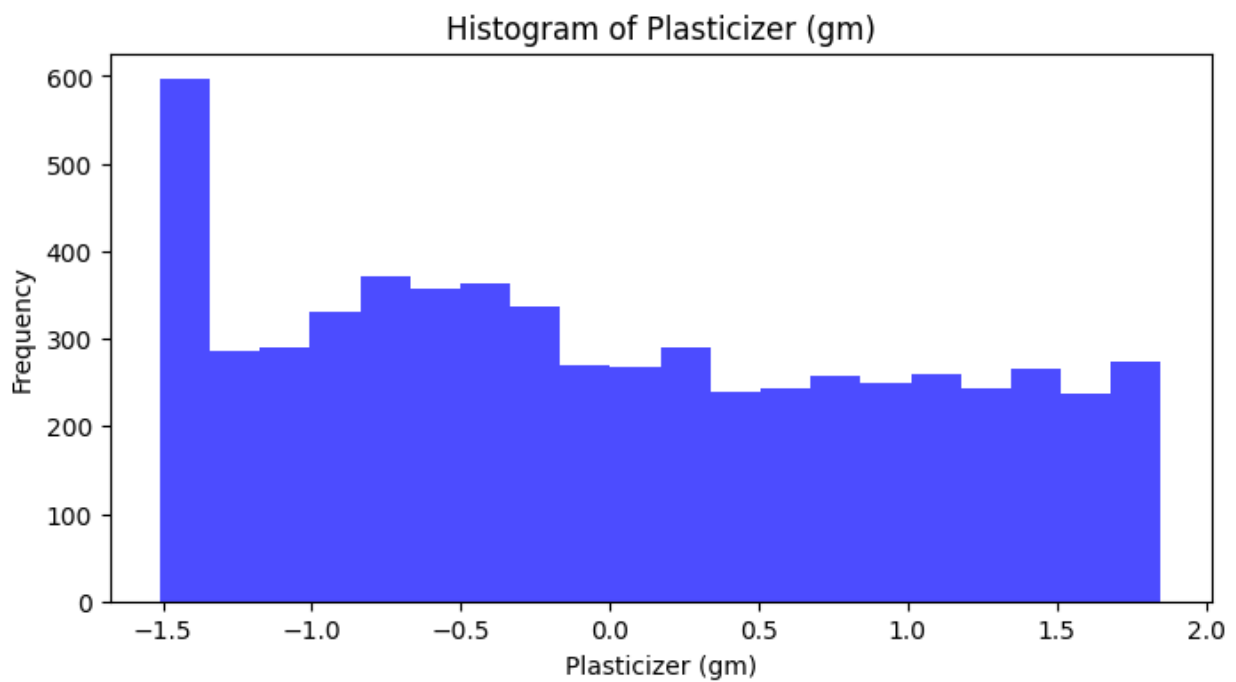
Ash Component (gm) is normally distributed (p-value = 1.0000)



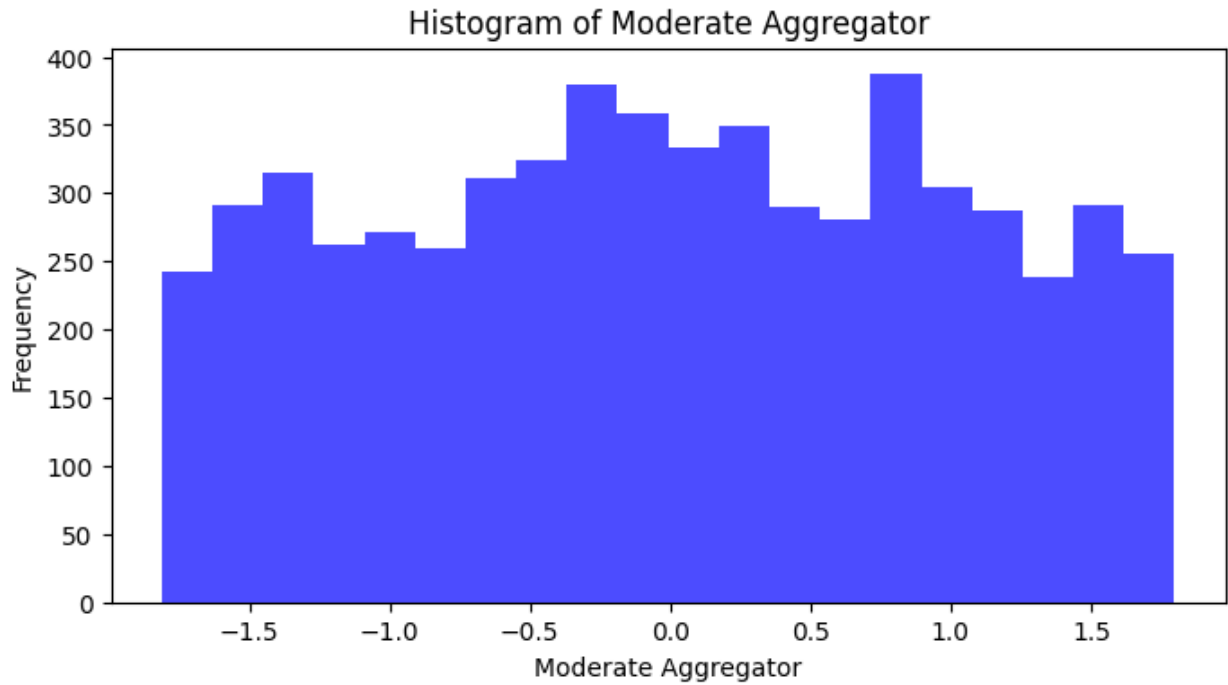
Water Mix (ml) is normally distributed (p-value = 1.0000)



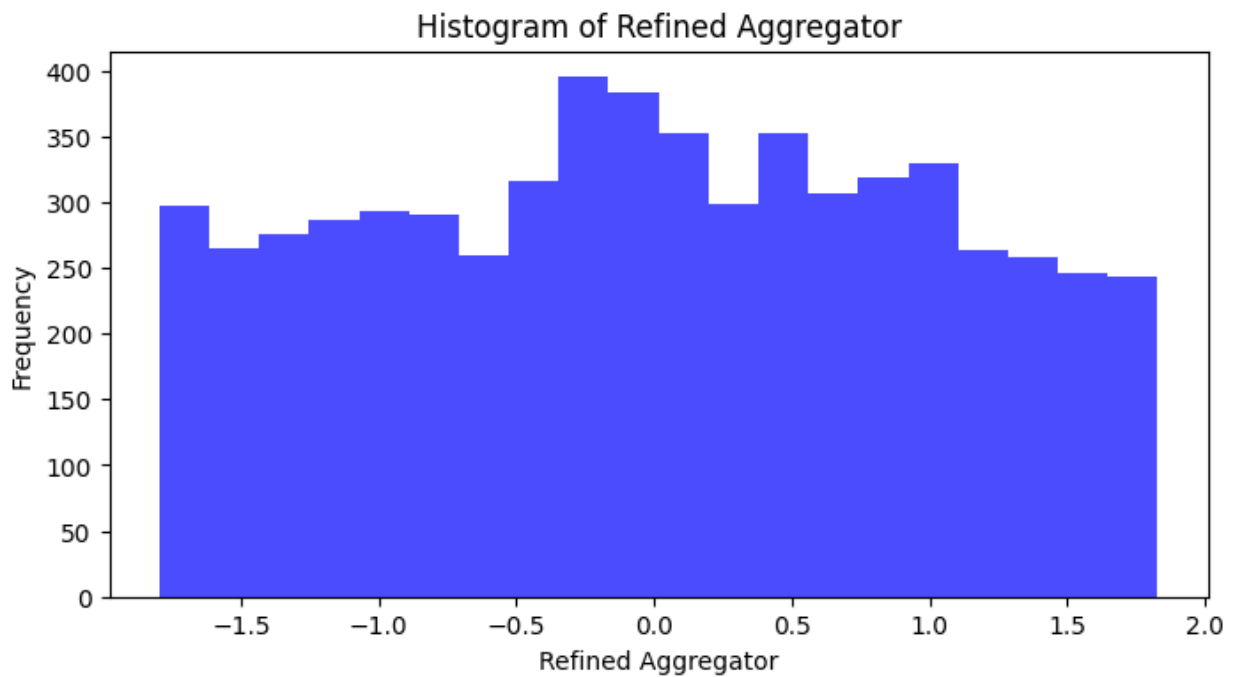
Plasticizer (gm) is normally distributed (p-value = 1.0000)



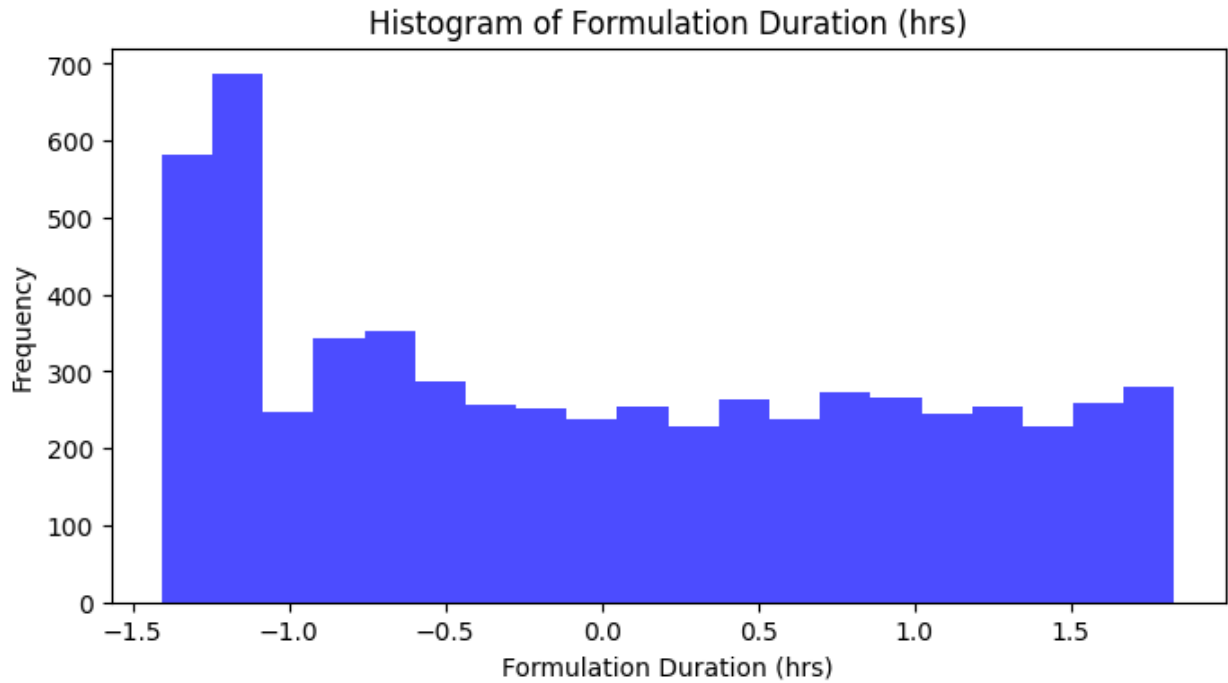
Moderate Aggregator is normally distributed (p-value = 1.0000)



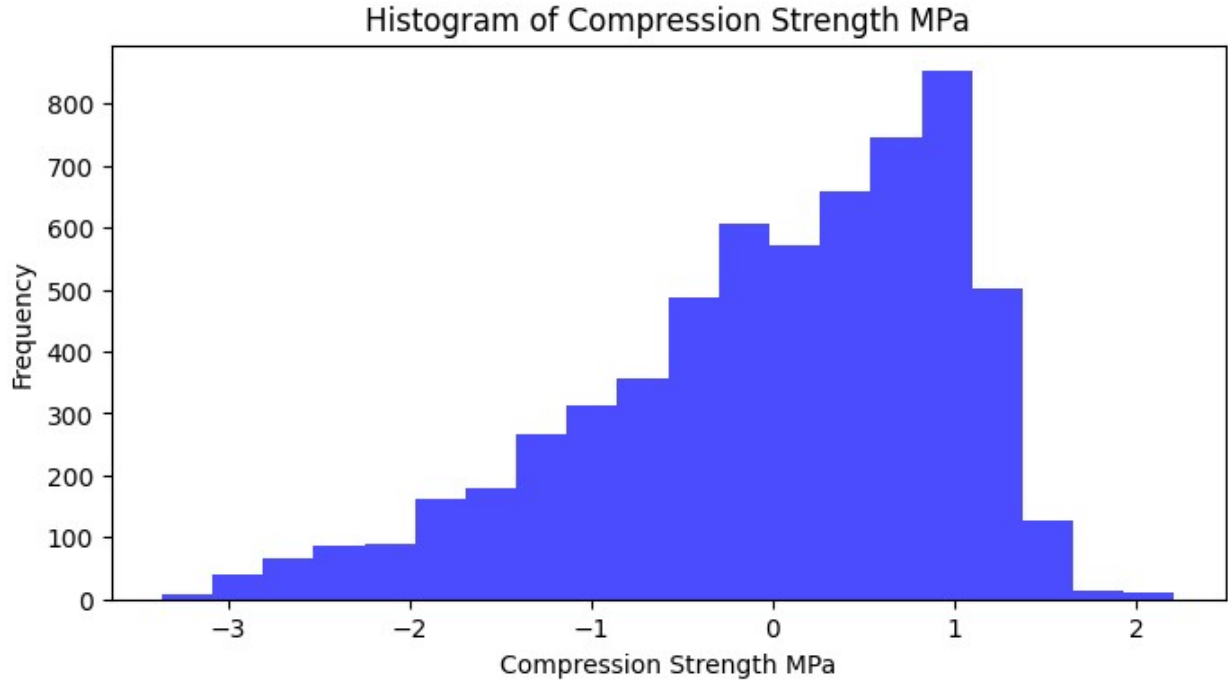
Refined Aggregator is normally distributed (p-value = 1.0000)



Formulation Duration (hrs) is normally distributed (p-value = 1.0000)



Compression Strength MPa is not normally distributed (p-value = 0.0000)

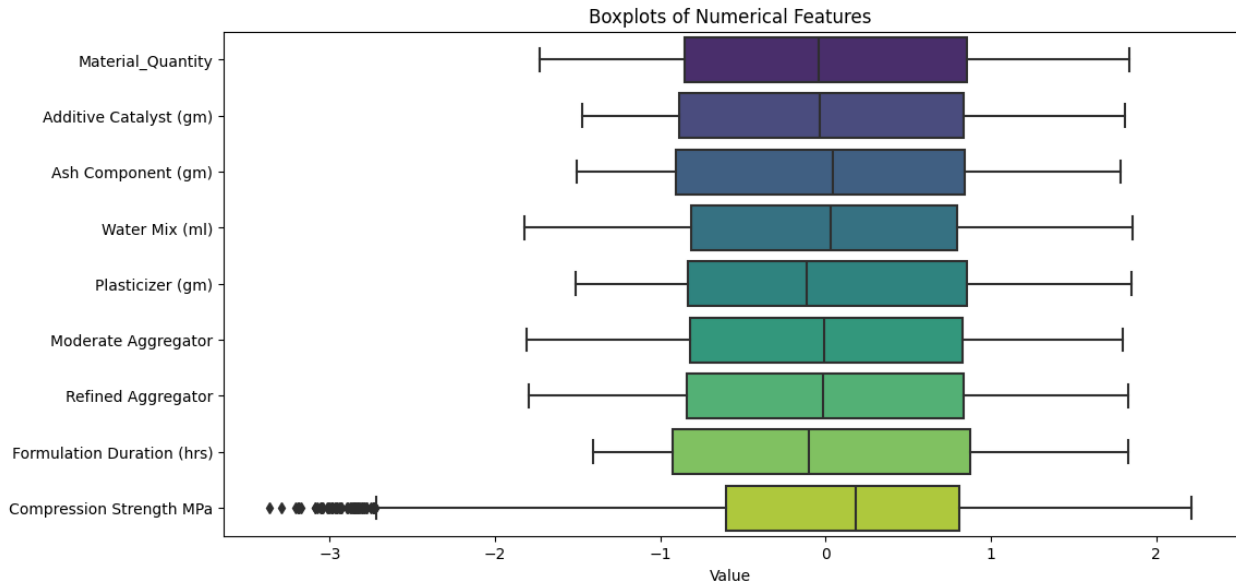


Output showing that columns are Normally Distributed

10. If dataset is too noisy then apply power transformer (Yeo-Jhonson)

```
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, orient='h', palette='viridis')
plt.title('Boxplots of Numerical Features')
plt.xlabel('Value')
plt.show()
```

```
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
```



Boxplots are helpful for identifying outliers and extreme values in the data. Outliers can be indicative of noise. Here it is showing there are no outliers present in the dataset

11. Once obtained the cleaned and normally distributed dataset. 12. Select the important feature for modelling.

After all the visualization, it is concluded that we will select import features for modelling: 1. Additive Catalyst 2. Material Quantity 3. Ash Component

Feature Engineering

1. Feature Normalization.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, StandardScaler
features_to_normalize = ['Additive Catalyst (gm)',
'Material_Quantity', 'Ash Component (gm)'] # selected features
selected_features_df = df[features_to_normalize]
# Min-Max Scaling (Normalization)
min_max_scaler = MinMaxScaler()
df_minmax =
```



```

pd.DataFrame(min_max_scaler.fit_transform(selected_features_df),
columns=features_to_normalize)
plt.figure(figsize=(12, 6))
#before normalization
for i, feature in enumerate(features_to_normalize):
    plt.subplot(2, len(features_to_normalize), i + 1)
    plt.title(f'Before Normalization: {feature}')
    sns.histplot(selected_features_df[feature], kde=True)
#after Min-Max scaling
for i, feature in enumerate(features_to_normalize):
    plt.subplot(2, len(features_to_normalize),
len(features_to_normalize) + i + 1)
    plt.title(f'After Min-Max Scaling: {feature}')
    sns.histplot(df_minmax[feature], kde=True)
plt.tight_layout()
plt.show()

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:767: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype):

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:767: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype):

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

```

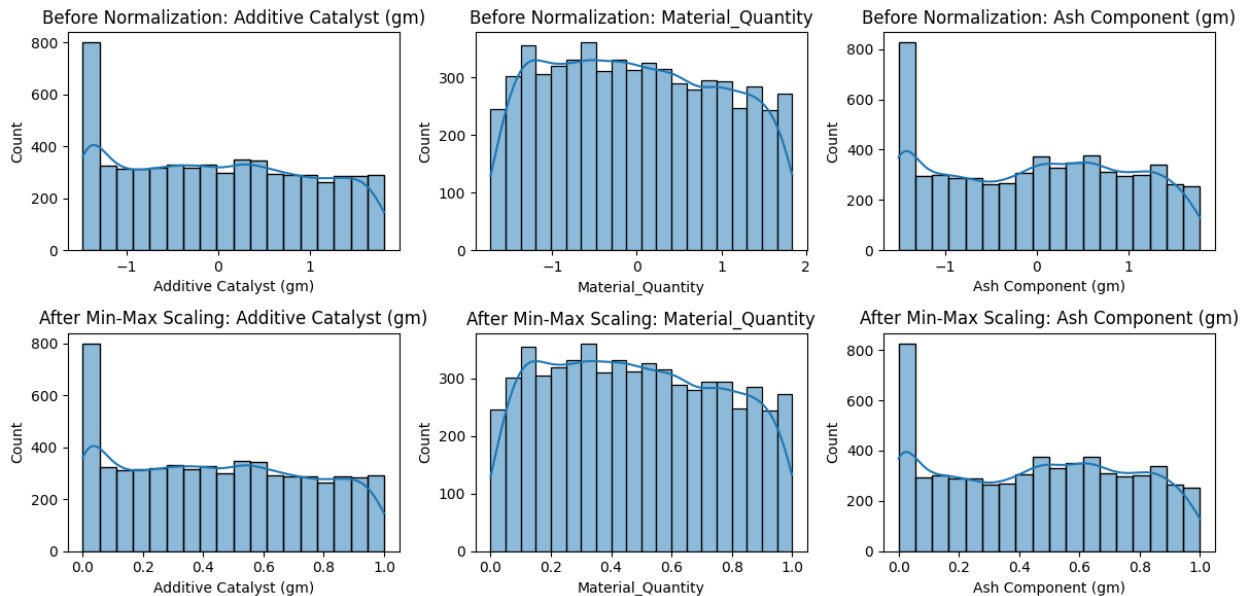
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:

```

```
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
    if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
```

FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.

```
with pd.option_context('mode.use_inf_as_na', True):
```



```
mean_std_before_normalization = {}
for feature in features_to_normalize:
    mean = selected_features_df[feature].mean()
    std = selected_features_df[feature].std()
    mean_std_before_normalization[feature] = {'Mean': mean, 'Std
Deviation': std}
```

```
print("Mean and Standard Deviation before Normalization:")
print(mean_std_before_normalization)
```

```
Mean and Standard Deviation before Normalization:
{'Additive Catalyst (gm)': {'Mean': -2.828030789094926e-17, 'Std
Deviation': 1.0000829290543316}, 'Material_Quantity': {'Mean':
1.5554169340022095e-16, 'Std Deviation': 1.000082929054337}, 'Ash
Component (gm)': {'Mean': 4.4777154160669664e-17, 'Std Deviation':
1.0000829290543338}}
```

2. Feature Standardization.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
features_to_standardize = ['Additive Catalyst (gm)',
```

```

'Material_Quantity', 'Ash Component (gm)'] # selected features
selected_features_df = df[features_to_standardize]
standard_scaler = StandardScaler()
standardized_features =
standard_scaler.fit_transform(selected_features_df)
df_standardized = pd.DataFrame(standardized_features,
columns=features_to_standardize)
plt.figure(figsize=(12, 6))
for i, feature in enumerate(features_to_standardize):
    plt.subplot(2, len(features_to_standardize),
len(features_to_standardize) + i + 1)
    plt.title(f'Standardization: {feature}')
    sns.histplot(df_standardized[feature], kde=True)
plt.tight_layout()
plt.show()

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if is_sparse(pd_dtype):

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:767: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if not hasattr(array, "sparse") and
array.dtypes.apply(is_sparse).any():

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:605: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if is_sparse(pd_dtype):

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\validation.py:614: FutureWarning: is_sparse is deprecated and will be removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.

```

    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

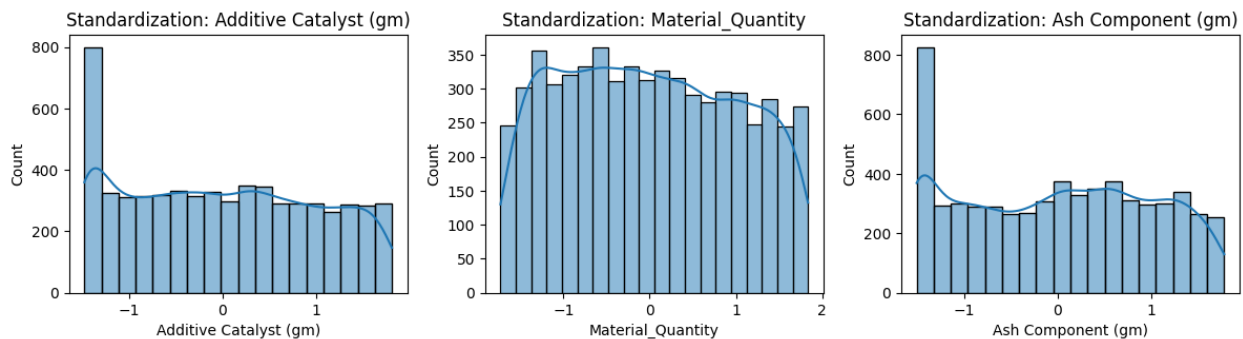
```

C:\Users\dewan\anaconda3\lib\site-packages\seaborn_oldcore.py:1498: FutureWarning: is_categorical_dtype is deprecated and will be removed

```

in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
    with pd.option_context('mode.use_inf_as_na', True):

```



3. Creating new features from the existing features.

```

# Combination 1
df['Sum_Features_1'] = (
    df['Additive Catalyst (gm)'] +
    df['Material_Quantity'] +
    df['Ash Component (gm)']
)
# Combination 2

```

```

df['Product_Features_2'] = (
    df['Additive Catalyst (gm)'] *
    df['Material_Quantity'] *
    df['Ash Component (gm)']
)
# Combination 3
df['Difference_Features_3'] = (
    df['Material_Quantity'] -
    df['Ash Component (gm)']
)
print(df)
plt.figure(figsize=(15, 5))
# Histogram for 'Sum_Features_1'
plt.subplot(1, 3, 1)
sns.histplot(df['Sum_Features_1'], kde=True)
plt.title('Histogram of Sum_Features_1')
# Histogram for 'Product_Features_2'
plt.subplot(1, 3, 2)
sns.histplot(df['Product_Features_2'], kde=True)
plt.title('Histogram of Product_Features_2')
# Histogram for 'Difference_Features_3'
plt.subplot(1, 3, 3)
sns.histplot(df['Difference_Features_3'], kde=True)
plt.title('Histogram of Difference_Features_3')
plt.tight_layout()
plt.show()

```

	Material_Quantity	Additive Catalyst (gm)	Ash Component (gm)	\
0	0.685267	-0.120763	-1.220399	
1	-1.669017	0.475855	0.993383	
2	1.175660	-1.454114	-0.001297	
3	0.051924	1.157758	-0.477757	
4	0.074260	1.169459	1.111252	
...	
6134	-1.299239	-0.258028	0.414815	
6135	-0.225176	0.710707	-0.458494	
6136	-0.169036	-1.305147	-1.264448	
6137	0.410768	0.591743	0.902590	
6138	1.177394	0.524011	0.744713	

	Water Mix (ml)	Plasticizer (gm)	Moderate Aggregator	\
0	-0.544913	-0.131863	1.560515	
1	-1.162838	-0.973825	0.940405	
2	1.707489	-0.487814	0.268721	
3	1.801610	0.115420	1.393822	
4	0.270641	-0.053999	1.018891	
...	
6134	-1.459643	-0.142987	0.052706	
6135	-0.867476	0.698975	-0.752823	
6136	-0.378336	1.476762	0.843193	

6137	-0.782984	0.035845	-1.366282
6138	-1.174874	-0.600760	1.710938

	Refined Aggregator	Formulation Duration (hrs)	\
0	-0.916033	1.512568	
1	1.758956	-1.294848	
2	-0.010127	0.562884	
3	0.616179	0.306047	
4	-0.273238	0.822301	
...	
6134	1.689987	1.632491	
6135	1.715163	-0.624598	
6136	-0.171912	1.141858	
6137	0.188536	1.781237	
6138	0.733108	1.659624	

	Compression Strength MPa	Sum_Features_1	Product_Features_2	\
0	1.428871	-0.655894	0.100994	
1	0.182873	-0.199779	-0.788955	
2	1.302969	-0.279751	0.002217	
3	0.923401	0.731925	-0.028721	
4	1.191951	2.354972	0.096506	
...	
6134	-0.387099	-1.142453	0.139063	
6135	-0.161963	0.027038	0.073375	
6136	-0.017455	-2.738631	-0.278958	
6137	0.084260	1.905101	0.219392	
6138	0.130775	2.446117	0.459463	

	Difference_Features_3
0	1.905666
1	-2.662400
2	1.176957
3	0.529681
4	-1.036992
...	...
6134	-1.714054
6135	0.233318
6136	1.095412
6137	-0.491823
6138	0.432680

[6139 rows x 12 columns]

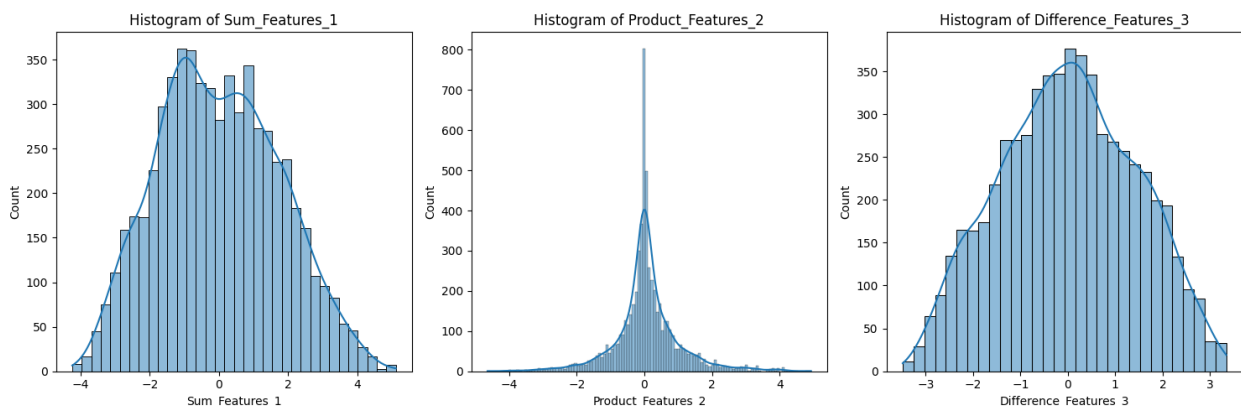
C:\Users\dewan\anaconda3\lib\site-packages\seaborn_oldcore.py:1498:
FutureWarning: is_categorical_dtype is deprecated and will be removed
in a future version. Use isinstance(dtype, CategoricalDtype) instead
if pd.api.types.is_categorical_dtype(vector):
C:\Users\dewan\anaconda3\lib\site-packages\seaborn_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed

in a future version. Convert inf values to NaN before operating instead.

```
with pd.option_context('mode.use_inf_as_na', True):  
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:  
FutureWarning: is_categorical_dtype is deprecated and will be removed  
in a future version. Use isinstance(dtype, CategoricalDtype) instead  
if pd.api.types.is_categorical_dtype(vector):  
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:  
FutureWarning: use_inf_as_na option is deprecated and will be removed  
in a future version. Convert inf values to NaN before operating  
instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):  
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1498:  
FutureWarning: is_categorical_dtype is deprecated and will be removed  
in a future version. Use isinstance(dtype, CategoricalDtype) instead  
if pd.api.types.is_categorical_dtype(vector):  
C:\Users\dewan\anaconda3\lib\site-packages\seaborn\_oldcore.py:1119:  
FutureWarning: use_inf_as_na option is deprecated and will be removed  
in a future version. Convert inf values to NaN before operating  
instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):
```



4. Feature selection and dropping features based on co-relation with dependent features, etc

```
X = df[['Additive Catalyst (gm)', 'Material_Quantity', 'Ash Component (gm)']]  
y = df['Compression Strength MPa']  
# Calculate correlation coefficients with the dependent variable  
correlations = X.corrwith(y)  
# Set a threshold for feature selection  
correlation_threshold = 0.3 # Adjust this threshold as needed
```



```

# Select features with correlations above the threshold
selected_features = X.columns[abs(correlations) >
correlation_threshold]
plt.figure(figsize=(12, 4))
for i, feature in enumerate(selected_features):
    plt.subplot(1, len(selected_features), i + 1)
    plt.scatter(df[feature], df['Compression Strength MPa'])
    plt.title(f'{feature} vs. Compression Strength')
    plt.xlabel(feature)
    plt.ylabel('Compression Strength MPa')
plt.tight_layout()
plt.show()

<Figure size 1200x400 with 0 Axes>

```

Predictive Models

```

df.columns

Index(['Material_Quantity', 'Additive Catalyst (gm)', 'Ash Component
(gm)',
      'Water Mix (ml)', 'Plasticizer (gm)', 'Moderate Aggregator',
      'Refined Aggregator', 'Formulation Duration (hrs)',
      'Compression Strength MPa', 'Sum_Features_1',
      'Product_Features_2',
      'Difference_Features_3'],
      dtype='object')

# required libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.impute import SimpleImputer

features = ['Additive Catalyst (gm)', 'Material_Quantity', 'Ash
Component (gm)']
target = 'Compression Strength MPa'
X = df[features]
y = df[target]
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y,
test_size=0.2, random_state=42)

# Make predictions on the train data for each model
from sklearn.metrics import mean_squared_error, r2_score

```

```

linear_train_predictions = linear_model.predict(X_train)
lasso_train_predictions = lasso_model.predict(X_train)
svr_train_predictions = svr_model.predict(X_train)
decision_tree_train_predictions = decision_tree_model.predict(X_train)
random_forest_train_predictions = random_forest_model.predict(X_train)

# Calculate performance metrics on the train data for each model
linear_train_mse = mean_squared_error(y_train,
linear_train_predictions)
linear_train_r2 = r2_score(y_train, linear_train_predictions)

lasso_train_mse = mean_squared_error(y_train, lasso_train_predictions)
lasso_train_r2 = r2_score(y_train, lasso_train_predictions)

svr_train_mse = mean_squared_error(y_train, svr_train_predictions)
svr_train_r2 = r2_score(y_train, svr_train_predictions)

decision_tree_train_mse = mean_squared_error(y_train,
decision_tree_train_predictions)
decision_tree_train_r2 = r2_score(y_train,
decision_tree_train_predictions)

random_forest_train_mse = mean_squared_error(y_train,
random_forest_train_predictions)
random_forest_train_r2 = r2_score(y_train,
random_forest_train_predictions)

# Print the performance metrics on the train data
print("Linear Regression - Train MSE:", linear_train_mse)
print("Linear Regression - Train R-squared (R2) Score:",
linear_train_r2)

print("\nLasso Regression - Train MSE:", lasso_train_mse)
print("Lasso Regression - Train R-squared (R2) Score:",
lasso_train_r2)

print("\nSupport Vector Regressor (SVR) - Train MSE:", svr_train_mse)
print("Support Vector Regressor (SVR) - Train R-squared (R2) Score:",
svr_train_r2)

print("\nDecision Tree - Train MSE:", decision_tree_train_mse)
print("Decision Tree - Train R-squared (R2) Score:",
decision_tree_train_r2)

print("\nRandom Forest - Train MSE:", random_forest_train_mse)
print("Random Forest - Train R-squared (R2) Score:",
random_forest_train_r2)

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:767: FutureWarning: is_sparse is deprecated and will be

```

```
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)` instead.
```

```
    if not hasattr(array, "sparse") and  
array.dtypes.apply(is_sparse).any():
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:767: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if not hasattr(array, "sparse") and  
array.dtypes.apply(is_sparse).any():
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
Linear Regression - Train MSE: 0.9517089676181134
```

```
Linear Regression - Train R-squared (R2) Score: 0.05378752837140366
```

```
Lasso Regression - Train MSE: 0.9520088380944287
```

```
Lasso Regression - Train R-squared (R2) Score: 0.05348938976578266
```

```
Support Vector Regressor (SVR) - Train MSE: 0.9719300592901996
```

```
Support Vector Regressor (SVR) - Train R-squared (R2) Score:  
0.033683221507552985
```

```
Decision Tree - Train MSE: 0.08636570026023646
```

```
Decision Tree - Train R-squared (R2) Score: 0.9141330958436821
```

```
Random Forest - Train MSE: 0.17540748762491995
```

```
Random Forest - Train R-squared (R2) Score: 0.825605560045183
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
```

```

instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be

```

```

removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

Applying Hyperparameter Tuning

```
from sklearn.model_selection import GridSearchCV

# Define a grid of hyperparameters
param_grid = {
    'alpha': [0.001, 0.01, 0.1, 1.0]
}

# Perform Grid Search for Lasso Regression
lasso_grid = GridSearchCV(Lasso(), param_grid, cv=5)
lasso_grid.fit(X_train, y_train)

# Get the best hyperparameters
best_lasso = lasso_grid.best_estimator_

# Print the best hyperparameters
print("Best Lasso Regression Hyperparameters:",
      best_lasso.get_params())
```

```
Best Lasso Regression Hyperparameters: {'alpha': 0.001, 'copy_X':
True, 'fit_intercept': True, 'max_iter': 1000, 'positive': False,
'precompute': False, 'random_state': None, 'selection': 'cyclic',
'tol': 0.0001, 'warm_start': False}
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
```

```
    if is_sparse(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
```

```
    if is_sparse(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
```

```
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
```

```
    if is_sparse(pd_dtype):
```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```

    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```



```

instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be

```

removed in a future version. Check ``isinstance(dtype, pd.SparseDtype)`` instead.

```
    if is_sparse(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:605: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  
validation.py:614: FutureWarning: is_sparse is deprecated and will be  
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`  
instead.
```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):  
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\  

```

```

validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

```

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.

```

```
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
```

```

instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

```

Based on the hyperparameter tuning for Lasso Regression, the best hyperparameters include a low regularization strength ($\alpha=0.001$). These settings aim to provide a good balance between model complexity and data fit for your specific dataset.

Checkig the model is Underfitting or Overfitting

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import learning_curve

def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None,
n_jobs=None, train_sizes=np.linspace(.1, 1.0, 5)):
    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Score")

    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, n_jobs=n_jobs,
        train_sizes=train_sizes)

    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)

    plt.grid()

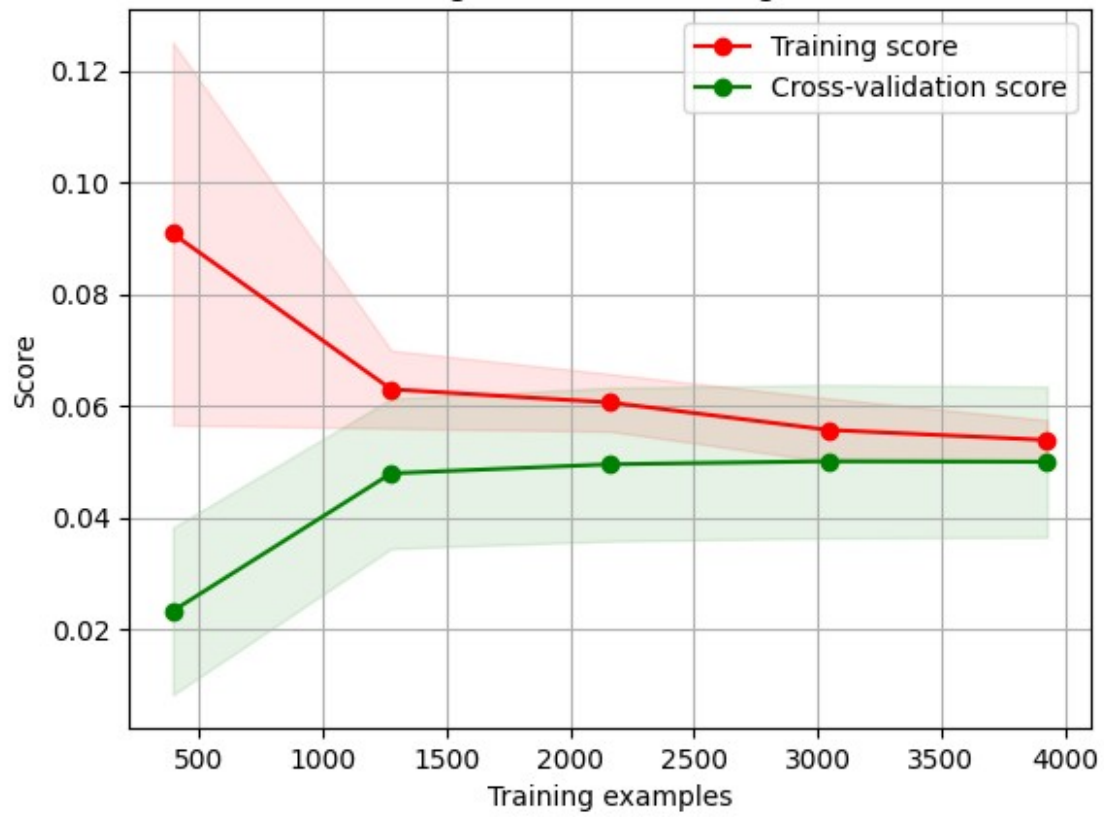
    plt.fill_between(train_sizes, train_scores_mean -
train_scores_std, train_scores_mean + train_scores_std, alpha=0.1,
color="r")
    plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
test_scores_mean + test_scores_std, alpha=0.1, color="g")
    plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
label="Training score")
    plt.plot(train_sizes, test_scores_mean, 'o-', color="g",
label="Cross-validation score")

    plt.legend(loc="best")
    return plt

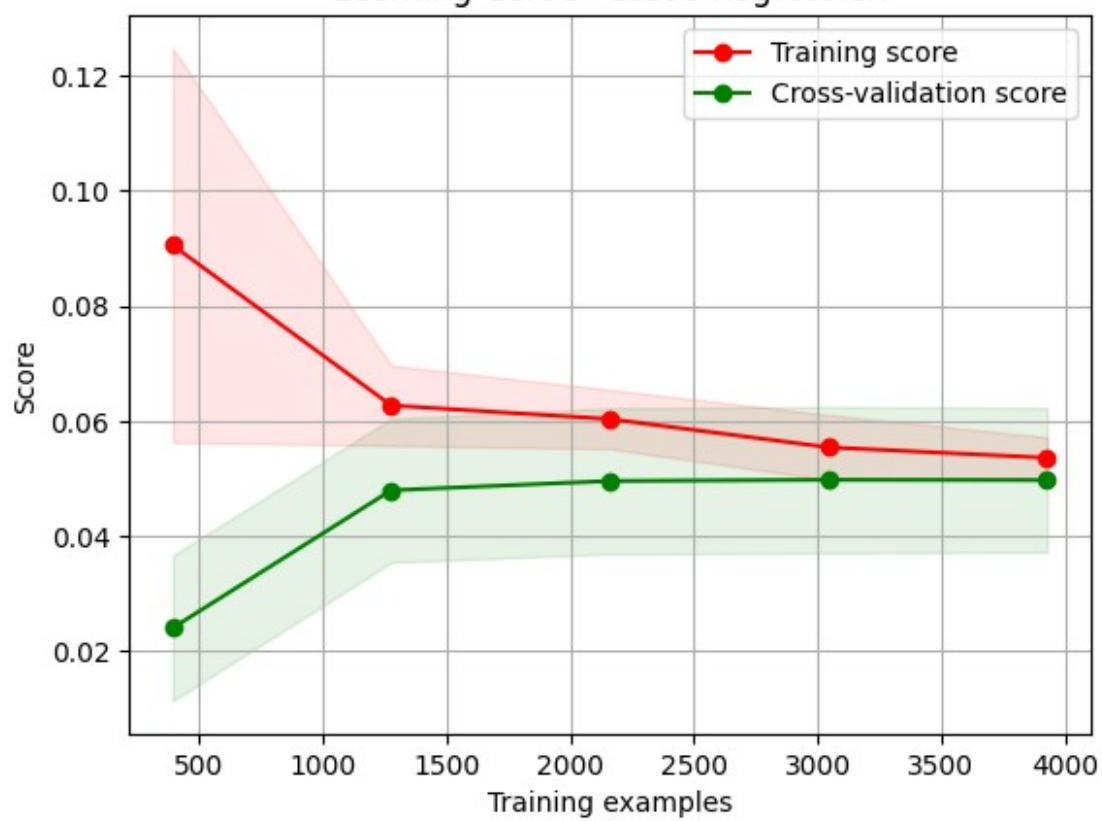
# Create learning curves for each model
models = [linear_model, lasso_model, svr_model, decision_tree_model,
random_forest_model]
model_names = ["Linear Regression", "Lasso Regression", "SVR",
"Decision Tree", "Random Forest"]

for model, model_name in zip(models, model_names):
    plot_learning_curve(model, f"Learning Curve - {model_name}",
X_train, y_train, cv=5, n_jobs=-1)
```

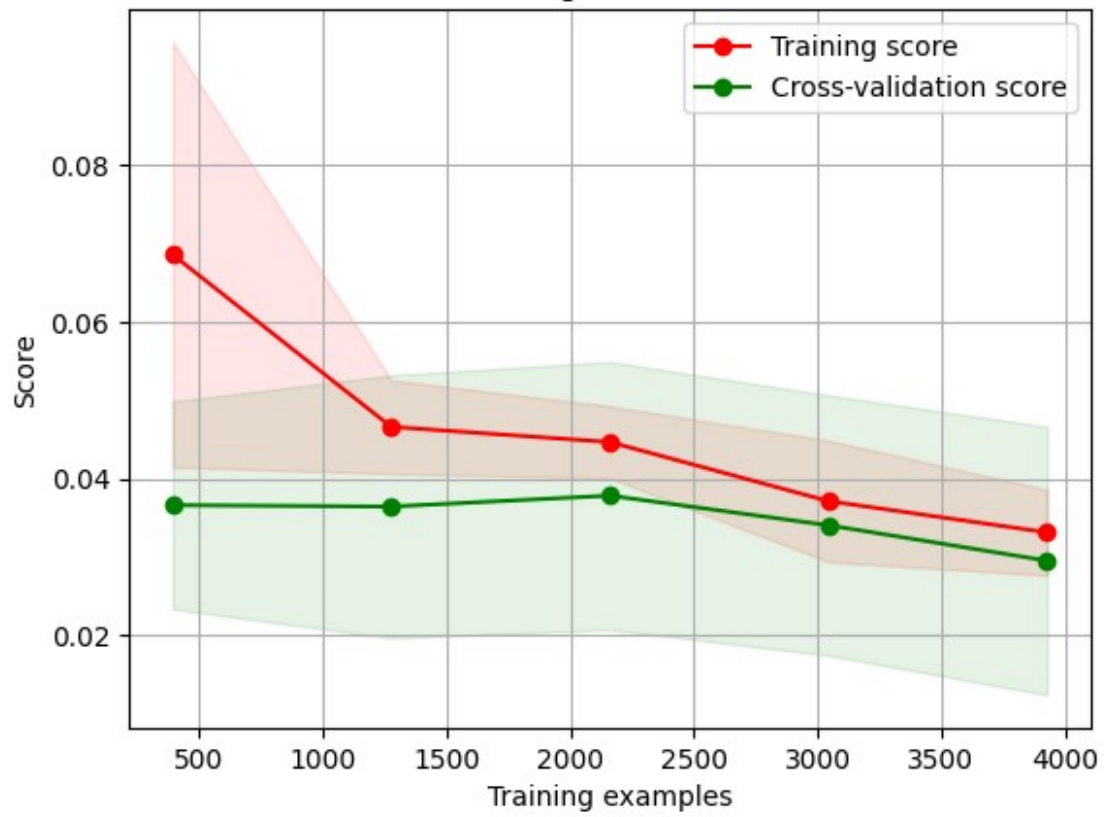
Learning Curve - Linear Regression



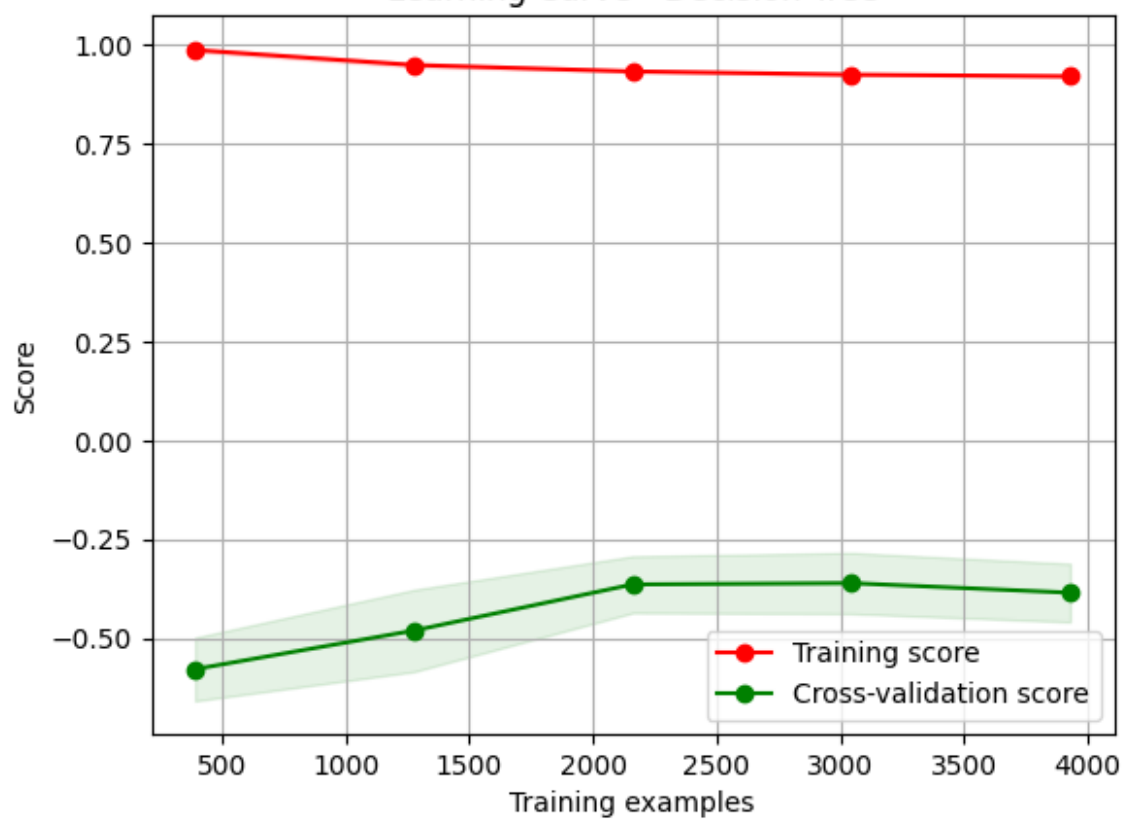
Learning Curve - Lasso Regression

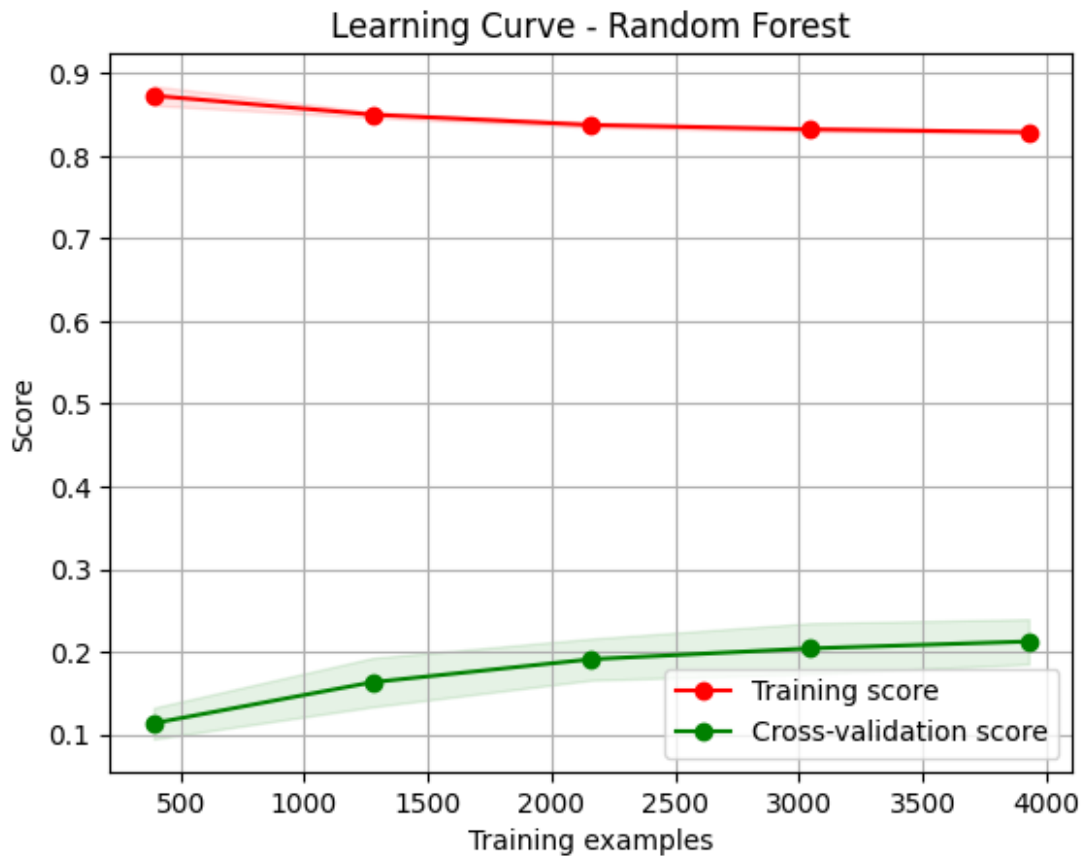


Learning Curve - SVR



Learning Curve - Decision Tree





The training score is everywhere higher than the validation score. This is generally the case: the model will be a better fit to data it has seen than to data it has not seen. For very low model complexity (a high-bias model), the training data is under-fit, which means that the model is a poor predictor both for the training data and for any previously unseen data. For very high model complexity (a high-variance model), the training data is over-fit, which means that the model predicts the training data very well, but fails for any previously unseen data. For some intermediate value, the validation curve has a maximum. This level of complexity indicates a suitable trade-off between bias and variance.

Create a Flask Application and API

```
values_to_predict = [[900, 900, 900]]
prediction = linear_model.predict(values_to_predict)
print("Predicted Value:", prediction)
```

```
Predicted Value: [348.07556929]
```

```
import pickle
from sklearn.linear_model import LinearRegression
import pickle
```

```
# Create and train a Linear Regression model
```

```

model = LinearRegression()
model.fit(X_train, y_train)

# Now you can pickle the trained model
with open('model_pickle', 'wb') as f:
    pickle.dump(model, f)

C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:605: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype):
C:\Users\dewan\anaconda3\lib\site-packages\sklearn\utils\
validation.py:614: FutureWarning: is_sparse is deprecated and will be
removed in a future version. Check `isinstance(dtype, pd.SparseDtype)`
instead.
    if is_sparse(pd_dtype) or not is_extension_array_dtype(pd_dtype):

with open('model_pickle', 'rb') as f:
    mp = pickle.load(f)

values_to_predict = [[900, 900, 900]]
prediction = mp.predict(values_to_predict)
print("Predicted Value:", prediction)

Predicted Value: [348.07556929]

from flask import Flask

app = Flask(__name__)

@app.route('/home')
def hello_world():
    return "Hello, World!"

if __name__ == '__main__':
    app.run(debug=True)

* Serving Flask app '__main__'
* Debug mode: on

WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)

An exception has occurred, use %tb to see the full traceback.

SystemExit: 1

```

```
C:\Users\dewan\anaconda3\lib\site-packages\IPython\core\
interactiveshell.py:3468: UserWarning: To exit: use 'exit', 'quit', or
Ctrl-D.
    warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```