main_new_0_0_0_0

August 28, 2021

1 IMPORTANT LIBRARIES

[1]:
```
# Warning Libraries :
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
# Scientific and Data Manipulation Libraries :
import pandas as pd
import numpy as np
from numpy import percentile
import math
import os
from sklearn.model_selection import train_test_split
```

[3]:
```
# Data Visualization Libraries :
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
```

[4]:
```
#pip install lasio
```

[5]:
```
#Libraries to convert .las files to .csv and merge

import lasio
from sys import stdout
import glob  ##For merging csv files
```

[6]:
```
#DATA IMPUTATION LIBRARY
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.impute import KNNImputer
from sklearn.linear_model import LinearRegression
```

[7]:
```
#Feature Selection Libraries
from sklearn.feature_selection import VarianceThreshold
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import SelectKBest
```

[8]:
```
#SCALING LIBRARIES
from sklearn.preprocessing import StandardScaler, MinMaxScaler, Normalizer,␣
↪RobustScaler, MaxAbsScaler
```

[9]:
```
#pip install catboost
```

[10]:
```
#MODEL TRAINING LIBRARIES
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from catboost import CatBoostClassifier
from sklearn.svm import OneClassSVM
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import RandomForestClassifier
```

[11]:
```
#MODEL ACCURACY LIBRARIES
from sklearn import metrics
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

[12]:
```
#grid searching key hyperparametres for logistic regression
from sklearn.datasets import make_blobs
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV
```

[13]:
```
path='/media/mr-robot/Local Disk/summer_training/Train'
os.chdir(path)
```

2 LAS TO CSV

[14]:
```
# Converting all las files in csv by iterating using lasio
for file in os.listdir():
    if file.endswith(".las"):
        file_path = f"{path}/{file}"
        las=lasio.read(file_path)
        size=len(file_path)
        filepath1=file_path[:size-3]
        las.to_csv(filepath1+'csv', units=False)
```

[15]:
```
# Adding Well name to easily identify
for file in os.listdir():
    if file.endswith(".csv"):
        s=pd.read_csv(file)
        size=len(file)
        dict=[]
        filename= file[:size-4]
```

```
        t=s.shape[0]
        for i in range(t):
            dict.append(filename)
        s['WELL']=dict
        s.to_csv(filename+'.csv',index=False)
```

[16]:
```
## To avoid furthur merging data and redundancy
if(os.path.isfile('./merged_data.csv') ):
    os.remove("merged_data.csv")

if(os.path.isfile('./FACIES_imputed.csv')):
    os.remove("FACIES_imputed.csv")

if(os.path.isfile('./FACIES_TRAIN.csv')):
    os.remove("FACIES_TRAIN.csv")
```

[17]:
```
# Merging all Well Log using Glob
filenames = glob.glob(path + "/*.csv")
dfs = []
for filename in filenames:
    dfs.append(pd.read_csv(filename))
big_frame = pd.concat(dfs, ignore_index=True)
big_frame.to_csv('merged_data.csv',index=False)
```

3 IMPUTATION

[18]:
```
df = pd.read_csv('merged_data.csv')
df
```

[18]:

	DEPTH	ACOUSTICIMPEDANCE1	AI	AVG_PIGN	CALI	\
0	1275.0552	12875.0811	12875081.0	NaN	9.7141	
1	1275.2076	12854.2256	12854226.0	NaN	9.7848	
2	1275.3600	13024.1377	13024138.0	NaN	9.8300	
3	1275.5124	13093.3428	13093343.0	NaN	9.8587	
4	1275.6648	13169.9307	13169931.0	NaN	9.8756	
…	…	…	…	…	…	
58494	1622.6028	6069.1309	6069130.5	NaN	8.5257	
58495	1622.7552	6067.8120	6067812.0	NaN	8.5282	
58496	1622.9076	6105.7729	6105773.0	NaN	8.5313	
58497	1623.0600	6152.9897	6152977.5	NaN	8.5331	
58498	1623.2124	6157.8291	6157829.5	NaN	8.5338	

	CALI[DERIVED]1	DT	FACIES	FLD1	GR	…	CALI_1	NPHI_1	\
0	9.7141	50.2544	NaN	NaN	50.2128	…	NaN	NaN	
1	9.7848	50.3881	NaN	NaN	49.7509	…	NaN	NaN	
2	9.8300	49.8852	NaN	NaN	48.2513	…	NaN	NaN	
3	9.8587	49.9032	NaN	NaN	46.8212	…	NaN	NaN	

```
4              9.8756   50.0157     NaN   NaN  45.3463  …     NaN     NaN
…               …        …      …    …       …  …      …      …
58494             NaN  123.7404     NaN   NaN       NaN  …     NaN  0.4993
58495             NaN  123.8728     NaN   NaN       NaN  …     NaN  0.5313
58496             NaN  123.3722     NaN   NaN       NaN  …     NaN  0.5448
58497             NaN  122.6038     NaN   NaN       NaN  …     NaN  0.5364
58498             NaN  122.3045     NaN   NaN       NaN  …     NaN  0.5331

       ZCOR  RHOB_1  RXO  SPDH      DTDS    M2R1  TH   U
0       NaN     NaN  NaN   NaN       NaN     NaN NaN NaN
1       NaN     NaN  NaN   NaN       NaN     NaN NaN NaN
2       NaN     NaN  NaN   NaN       NaN     NaN NaN NaN
3       NaN     NaN  NaN   NaN       NaN     NaN NaN NaN
4       NaN     NaN  NaN   NaN       NaN     NaN NaN NaN
…     …      …   …   …       …     …   ..  ..
58494   NaN  2.4639  NaN   NaN  123.7404  1.5970 NaN NaN
58495   NaN  2.4660  NaN   NaN  123.8728  1.6128 NaN NaN
58496   NaN  2.4714  NaN   NaN  123.3722  1.7043 NaN NaN
58497   NaN  2.4750  NaN   NaN  122.6038  1.8375 NaN NaN
58498   NaN  2.4709  NaN   NaN  122.3045  1.9363 NaN NaN

[58499 rows x 67 columns]
```

[19]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 67 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   DEPTH                58499 non-null  float64
 1   ACOUSTICIMPEDANCE1   58499 non-null  float64
 2   AI                   55259 non-null  float64
 3   AVG_PIGN             323 non-null    float64
 4   CALI                 54981 non-null  float64
 5   CALI[DERIVED]1       44090 non-null  float64
 6   DT                   58499 non-null  float64
 7   FACIES               52641 non-null  float64
 8   FLD1                 3963 non-null   float64
 9   GR                   58379 non-null  float64
 10  LLD                  44942 non-null  float64
 11  LLS                  27394 non-null  float64
 12  DEPTH_1              50885 non-null  float64
 13  NPHI                 58172 non-null  float64
 14  ONE-WAYTIME1         15713 non-null  float64
 15  PIGN_MODELLING       51101 non-null  float64
 16  PIMP                 55259 non-null  float64
 17  RHOB                 58499 non-null  float64
```

```
18  RT_MODELLING        53629 non-null  float64
19  SP                  55992 non-null  float64
20  SUWI_MODELLING      51099 non-null  float64
21  TDVSS               58437 non-null  float64
22  ZLT                 44562 non-null  float64
23  WELL                58499 non-null  object
24  DFL                 23458 non-null  float64
25  HDRS                26951 non-null  float64
26  HMRS                26951 non-null  float64
27  PERF_INT            1569 non-null   float64
28  PERMEABILITY        28149 non-null  float64
29  PIGN                46949 non-null  float64
30  RT_POWER            51379 non-null  float64
31  SUWI                46947 non-null  float64
32  VCL                 46947 non-null  float64
33  WATER_VOL           43735 non-null  float64
34  LL3                 12373 non-null  float64
35  BS                  6706 non-null   float64
36  CALI1               2389 non-null   float64
37  DEVI                10283 non-null  float64
38  DT1                 6130 non-null   float64
39  PHIT                16532 non-null  float64
40  PIGE                5245 non-null   float64
41  LLD_1               9518 non-null   float64
42  SXWI                27938 non-null  float64
43  PEF                 19419 non-null  float64
44  AZI1                2487 non-null   float64
45  TEMP                14514 non-null  float64
46  DRES                2765 non-null   float64
47  DT2                 2765 non-null   float64
48  DT4P                5854 non-null   float64
49  GR_EDTC             2765 non-null   float64
50  M2R2                8568 non-null   float64
51  LLS_1               238 non-null    float64
52  MSFL                2765 non-null   float64
53  PR                  2757 non-null   float64
54  TENS                2765 non-null   float64
55  VPVS                2757 non-null   float64
56  BIT                 5553 non-null   float64
57  CALI_1              2999 non-null   float64
58  NPHI_1              10811 non-null  float64
59  ZCOR                2998 non-null   float64
60  RHOB_1              10899 non-null  float64
61  RXO                 1552 non-null   float64
62  SPDH                3069 non-null   float64
63  DTDS                2546 non-null   float64
64  M2R1                2546 non-null   float64
65  TH                  2509 non-null   float64
```

```
 66  U                    2509 non-null   float64
dtypes: float64(66), object(1)
memory usage: 29.9+ MB
```

[20]:
```
df.shape[1]
```

[20]: 67

[21]:
```
obj = df.isnull().sum()
for key,value in obj.iteritems():
    print(key,",",value)
```

```
DEPTH , 0
ACOUSTICIMPEDANCE1 , 0
AI , 3240
AVG_PIGN , 58176
CALI , 3518
CALI[DERIVED]1 , 14409
DT , 0
FACIES , 5858
FLD1 , 54536
GR , 120
LLD , 13557
LLS , 31105
DEPTH_1 , 7614
NPHI , 327
ONE-WAYTIME1 , 42786
PIGN_MODELLING , 7398
PIMP , 3240
RHOB , 0
RT_MODELLING , 4870
SP , 2507
SUWI_MODELLING , 7400
TDVSS , 62
ZLT , 13937
WELL , 0
DFL , 35041
HDRS , 31548
HMRS , 31548
PERF_INT , 56930
PERMEABILITY , 30350
PIGN , 11550
RT_POWER , 7120
SUWI , 11552
VCL , 11552
WATER_VOL , 14764
LL3 , 46126
BS , 51793
```

```
CALI1 , 56110
DEVI , 48216
DT1 , 52369
PHIT , 41967
PIGE , 53254
LLD_1 , 48981
SXWI , 30561
PEF , 39080
AZI1 , 56012
TEMP , 43985
DRES , 55734
DT2 , 55734
DT4P , 52645
GR_EDTC , 55734
M2R2 , 49931
LLS_1 , 58261
MSFL , 55734
PR , 55742
TENS , 55734
VPVS , 55742
BIT , 52946
CALI_1 , 55500
NPHI_1 , 47688
ZCOR , 55501
RHOB_1 , 47600
RXO , 56947
SPDH , 55430
DTDS , 55953
M2R1 , 55953
TH , 55990
U , 55990
```

[22]:
```
#Selecting required feature
df=df[["DT","GR","NPHI","RHOB","FACIES"]]
```

[23]:
```
df
```

[23]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.2128	0.5340	2.1228	NaN
1	50.3881	49.7509	0.5316	2.1250	NaN
2	49.8852	48.2513	0.5126	2.1316	NaN
3	49.9032	46.8212	0.5137	2.1437	NaN
4	50.0157	45.3463	0.5472	2.1611	NaN
…	…	…	…	…	…
58494	123.7404	NaN	0.4993	2.4639	NaN
58495	123.8728	NaN	0.5313	2.4660	NaN
58496	123.3722	NaN	0.5448	2.4714	NaN

```
58497  122.6038      NaN  0.5364  2.4750     NaN
58498  122.3045      NaN  0.5331  2.4709     NaN

[58499 rows x 5 columns]
```

[24]:
```
df.isnull().sum()
```

[24]:
```
DT           0
GR         120
NPHI       327
RHOB         0
FACIES    5858
dtype: int64
```

[25]:
```
#Exporting required features to csv
df.to_csv("FACIES_TRAIN.csv",index=False)
```

[26]:
```
df=pd.read_csv("FACIES_TRAIN.csv")
```

[27]:
```
df.head(20)
```

[27]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.2128	0.5340	2.1228	NaN
1	50.3881	49.7509	0.5316	2.1250	NaN
2	49.8852	48.2513	0.5126	2.1316	NaN
3	49.9032	46.8212	0.5137	2.1437	NaN
4	50.0157	45.3463	0.5472	2.1611	NaN
5	50.6831	44.0819	0.5550	2.1740	NaN
6	51.4311	43.6654	0.5612	2.1707	NaN
7	52.1678	43.3915	0.5566	2.1595	NaN
8	52.2883	44.1249	0.5390	2.1534	NaN
9	51.5991	46.1805	0.5245	2.1551	NaN
10	50.6185	48.6156	0.5152	2.1542	NaN
11	50.5171	49.6999	0.5152	2.1535	NaN
12	50.1209	49.4600	0.5180	2.1586	NaN
13	50.0558	48.3665	0.5156	2.1662	NaN
14	49.4216	46.8647	0.5070	2.1705	NaN
15	47.9804	45.7345	0.4913	2.1702	NaN
16	46.3324	45.5512	0.4696	2.1657	NaN
17	45.1378	45.9222	0.4570	2.1579	NaN
18	45.2291	46.4844	0.4654	2.1533	NaN
19	45.6106	49.6481	0.4952	2.1526	NaN

[28]:
```
df.shape
```

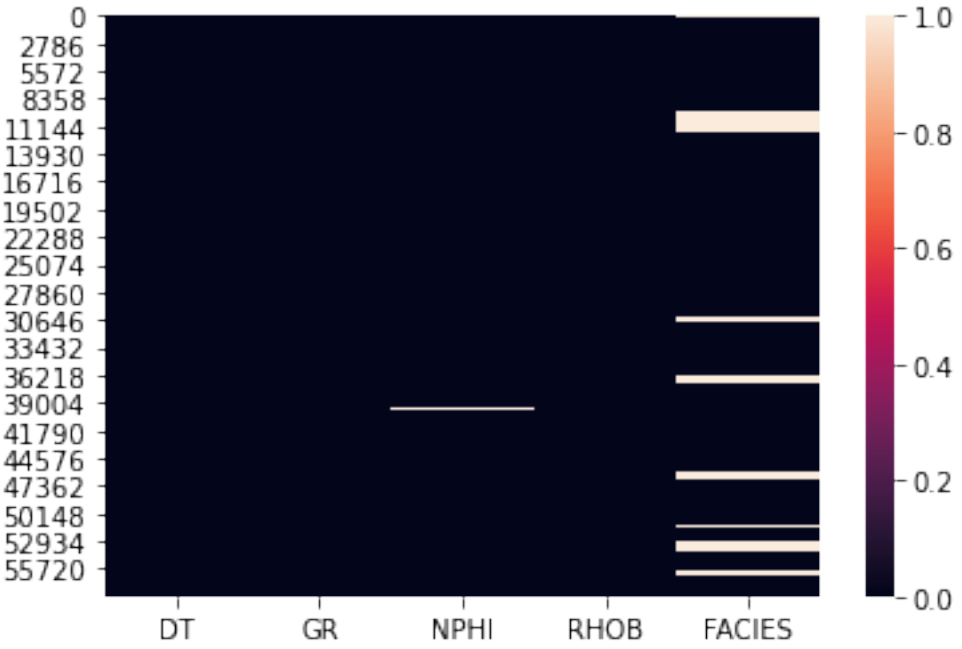
[28]:
```
(58499, 5)
```

[29]:
```
df.info()
```

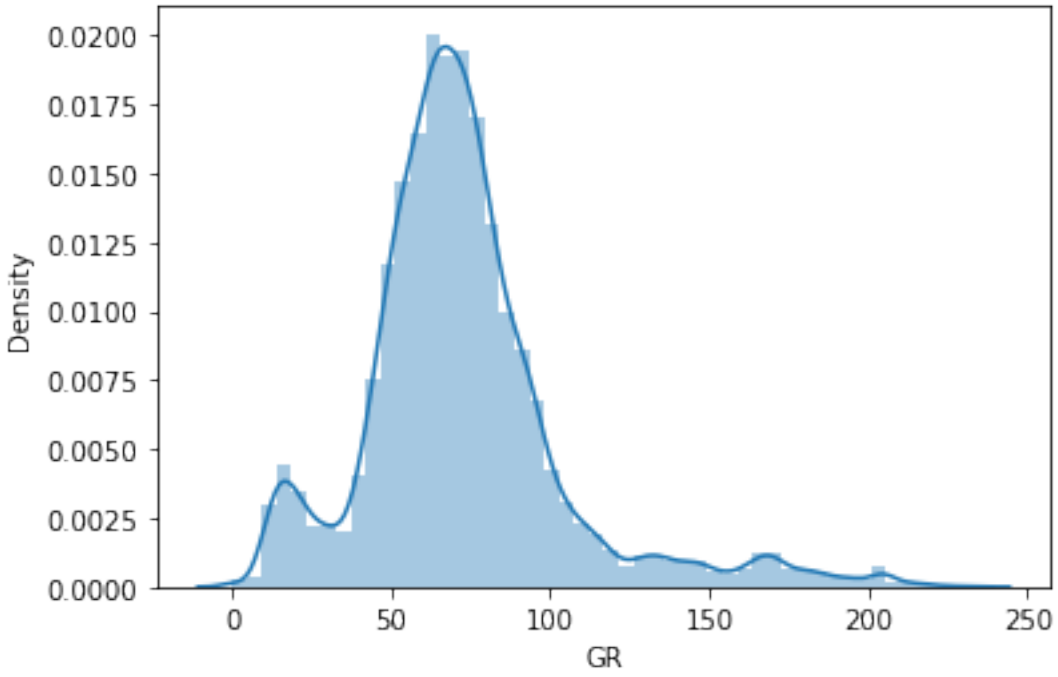
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DT      58499 non-null  float64
 1   GR      58379 non-null  float64
 2   NPHI    58172 non-null  float64
 3   RHOB    58499 non-null  float64
 4   FACIES  52641 non-null  float64
dtypes: float64(5)
memory usage: 2.2 MB
```

[30]:
```
sns.heatmap(df.isnull())
```

[30]: <AxesSubplot:>



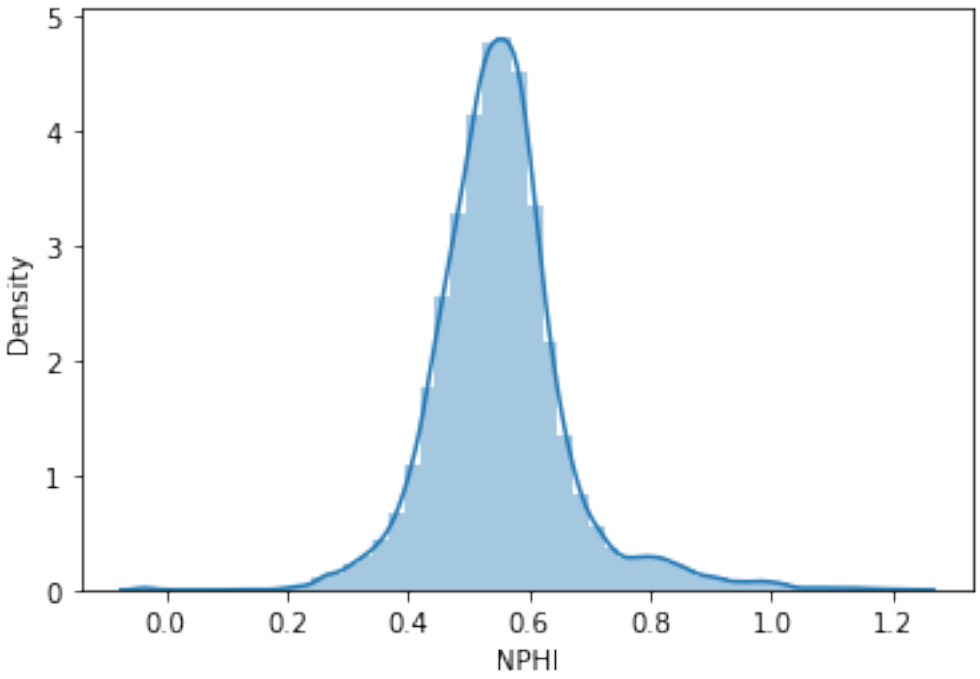
[31]:
```
null_gr = sns.distplot(df.GR.dropna())
```



[32]:
```
df.GR.describe()
```

[32]:
```
count    58379.000000
mean        72.610942
std         32.140407
min          0.000000
25%         55.340300
50%         68.939700
75%         83.758300
max        233.707400
Name: GR, dtype: float64
```

[33]:
```
null_nphi=sns.distplot(df.NPHI.dropna())
```

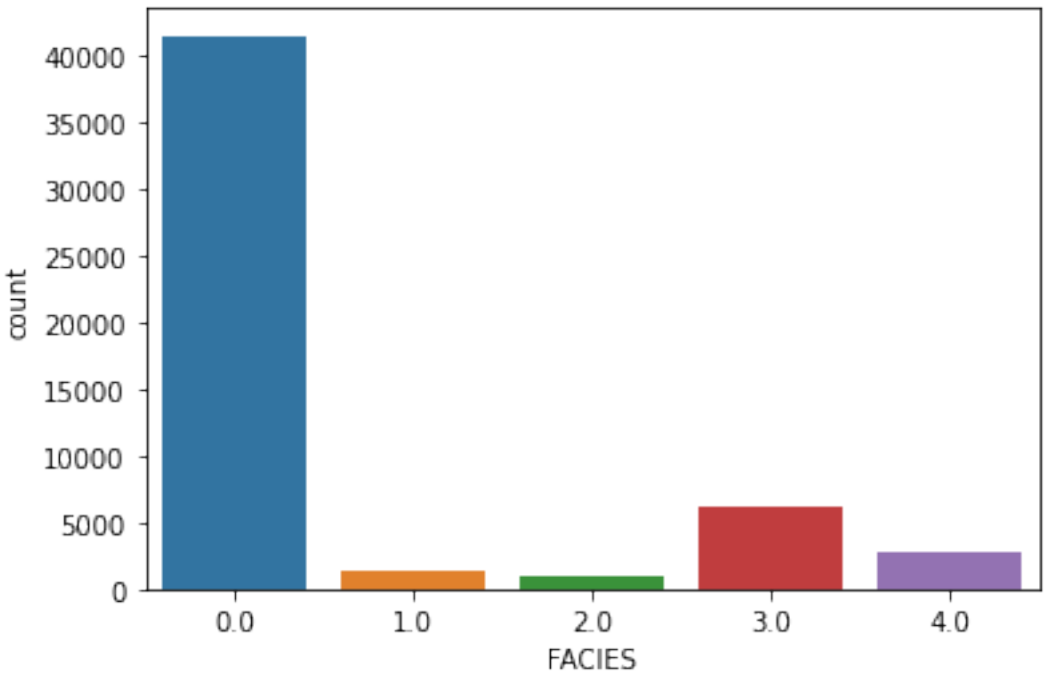


[34]:
```
df.NPHI.describe()
```

[34]:
```
count    58172.000000
mean         0.551710
std          0.109983
min         -0.038000
25%          0.489275
50%          0.546600
75%          0.600500
max          1.231200
Name: NPHI, dtype: float64
```

[35]:
```
sns.countplot(x="FACIES",data=df)
```

[35]:
```
<AxesSubplot:xlabel='FACIES', ylabel='count'>
```



[36]:
```
df.FACIES.value_counts(dropna=False)
```

[36]:
```
0.0    41514
3.0     6138
NaN     5858
4.0     2798
1.0     1281
2.0      910
Name: FACIES, dtype: int64
```

[37]:
```
def imputing(imputation_strategy,imputing_data):
    df=imputing_data
    if imputation_strategy == "Mean":
        df.GR.fillna(df.GR.mean(),inplace=True)
        print( df.GR.isnull().sum())
        print("Graph (GR) after filling null values with mean")
        sns.displot(df.GR.dropna())
        df.NPHI.fillna(df.NPHI.mean(),inplace=True)
        print("Graph (NPHI) after filling null values with mean")
        print(df.NPHI.isnull().sum())
        sns.displot(df.NPHI.dropna())
        #dropping FACIES rows with null
        df.dropna(axis=0,inplace=True)
        print(df.isnull().sum())
        df['FACIES'] = df.FACIES.astype(np.int64)
```

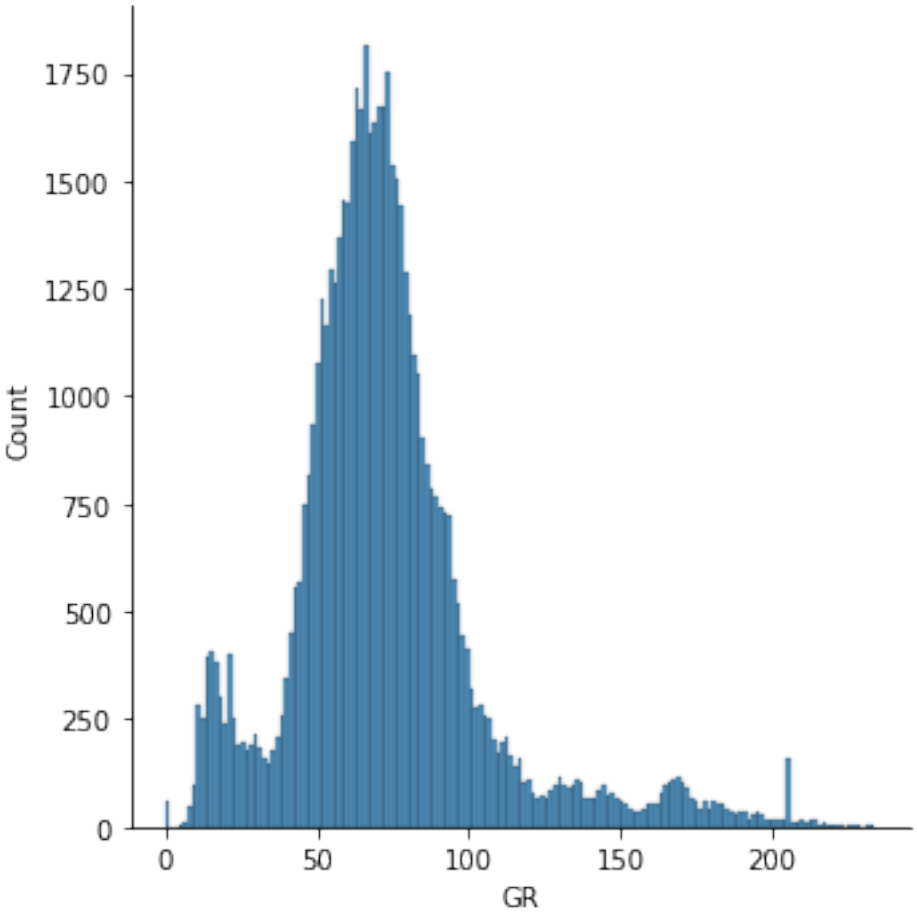
```
        df.info()
        df.FACIES.describe()
        return df

    elif imputation_strategy == "bffill":
        df = df.ffill(axis = 0)
        df = df.bfill(axis = 0)
        df['FACIES'] = df.FACIES.astype(np.int64)
        print(df.isnull().sum())
        return df

    elif imputation_strategy == "KNNImputer":
        knn= KNNImputer(n_neighbors=3)
        X=df.drop('FACIES',1)
        t=knn.fit_transform(X)
        X=pd.DataFrame(t)
        Y=df['FACIES']
        Y=Y.ffill(axis=0)
        Y=Y.bfill(axis=0)
        X['FACIES']=Y
        df=X
        df['FACIES'] = df.FACIES.astype(np.int64)
        d=['DT','GR','NPHI','RHOB']
        for i in range(4):
            df.columns.values[i]=d[i]
        return df

    elif imputation_strategy == "IterativeImputer":
        lr=LinearRegression()    #can use other regressions too. / default is␣
↪beysian
        imp=IterativeImputer(max_iter=3)
        X=df.drop('FACIES',1)
        t=imp.fit_transform(X)
        X=pd.DataFrame(t)
        Y=df['FACIES']
        Y=Y.ffill(axis=0)
        Y=Y.bfill(axis=0)
        X['FACIES']=Y
        df=X
        df['FACIES'] = df.FACIES.astype(np.int64)
        d=['DT','GR','NPHI','RHOB']
        for i in range(4):
            df.columns.values[i]=d[i]
        return df

    elif imputation_strategy == "KNNimputer_floor" :
        X=df
```

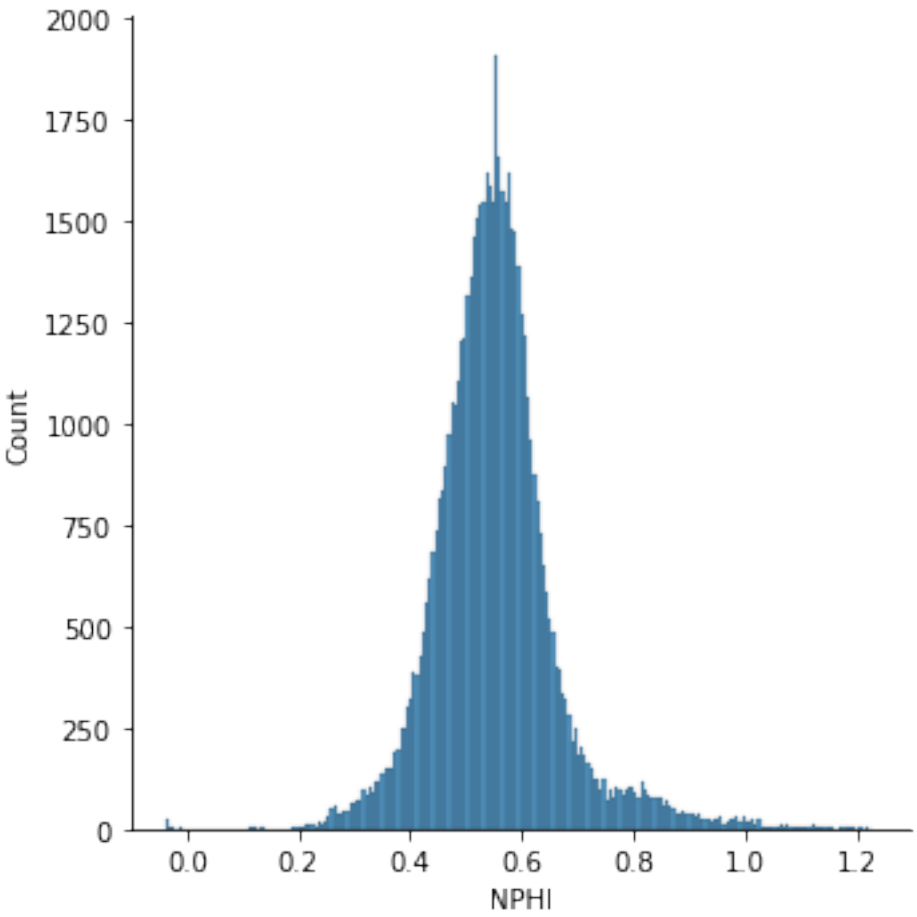
```
        knn= KNNImputer(n_neighbors=3)
        t=knn.fit_transform(df)
        df=pd.DataFrame(t)
        d=['DT','GR','NPHI','RHOB','FACIES']
        df['FACIES1'] = X.FACIES
        for i in range(5):
            df.columns.values[i]=d[i]
        df=df.drop('FACIES1',1)
        df['FACIES'] = df.FACIES.astype(np.int64)
        return df

    elif imputation_strategy == "IterativeImputer_floor" :
        X=df
        lr=LinearRegression()
        imp= IterativeImputer(max_iter=3)
        t=imp.fit_transform(df)
        df=pd.DataFrame(t)
        d=['DT','GR','NPHI','RHOB','FACIES']
        df['FACIES1'] = X.FACIES
        for i in range(5):
            df.columns.values[i]=d[i]
        df=df.drop('FACIES1',1)
        df['FACIES'] = df.FACIES.astype(np.int64)
        return df

    elif imputation_strategy == "KNNBinning" :
        X=df
        knn= KNNImputer(n_neighbors=3)
        t=knn.fit_transform(df)
        df=pd.DataFrame(t)
        d=['DT','GR','NPHI','RHOB','FACIES']
        df['FACIES1'] = X.FACIES
        for i in range(5):
            df.columns.values[i]=d[i]
        df=df.drop('FACIES1',1)
        #df['FACIES'] = pd.cut(x=df['FACIES'],bins=[0,0.5,1.5,2.5,3.5,4.0],␣
↪labels=['0','1','2','3','4'])
        return df

    elif imputation_strategy == "dropna":
        df=df.dropna(axis=0)
        return df
```

[38]:
```
imputation_strategy = ["Mean" , "bffill" , "KNNImputer" , "IterativeImputer" ,␣
↪"KNNimputer_floor" , "IterativeImputer_floor" , "KNNBinning","dropna"]
#select option from 0-7 (6 is experimental)
optionimputation=0
```

```
df=imputing(imputation_strategy[optionimputation],df)
```

```
0
Graph (GR) after filling null values with mean
Graph (NPHI) after filling null values with mean
0
DT        0
GR        0
NPHI      0
RHOB      0
FACIES    0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
Int64Index: 52641 entries, 271 to 58447
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DT      52641 non-null  float64
 1   GR      52641 non-null  float64
 2   NPHI    52641 non-null  float64
 3   RHOB    52641 non-null  float64
 4   FACIES  52641 non-null  int64
dtypes: float64(4), int64(1)
memory usage: 2.4 MB
```


1750
1500
1250
1000
750
500
250
0
Count
0
50
100
150
200
GR



[39]:
```
#if option==6:
#    df['FACIES'] = pd.cut(x=df['FACIES'],bins=[0.0,0.5,1.5,2.5,3.5,4.0],␣
 ↪labels=['0','1','2','3','4'])
```

[40]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 52641 entries, 271 to 58447
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DT      52641 non-null  float64
 1   GR      52641 non-null  float64
 2   NPHI    52641 non-null  float64
 3   RHOB    52641 non-null  float64
 4   FACIES  52641 non-null  int64
dtypes: float64(4), int64(1)
```

```
memory usage: 2.4 MB
```

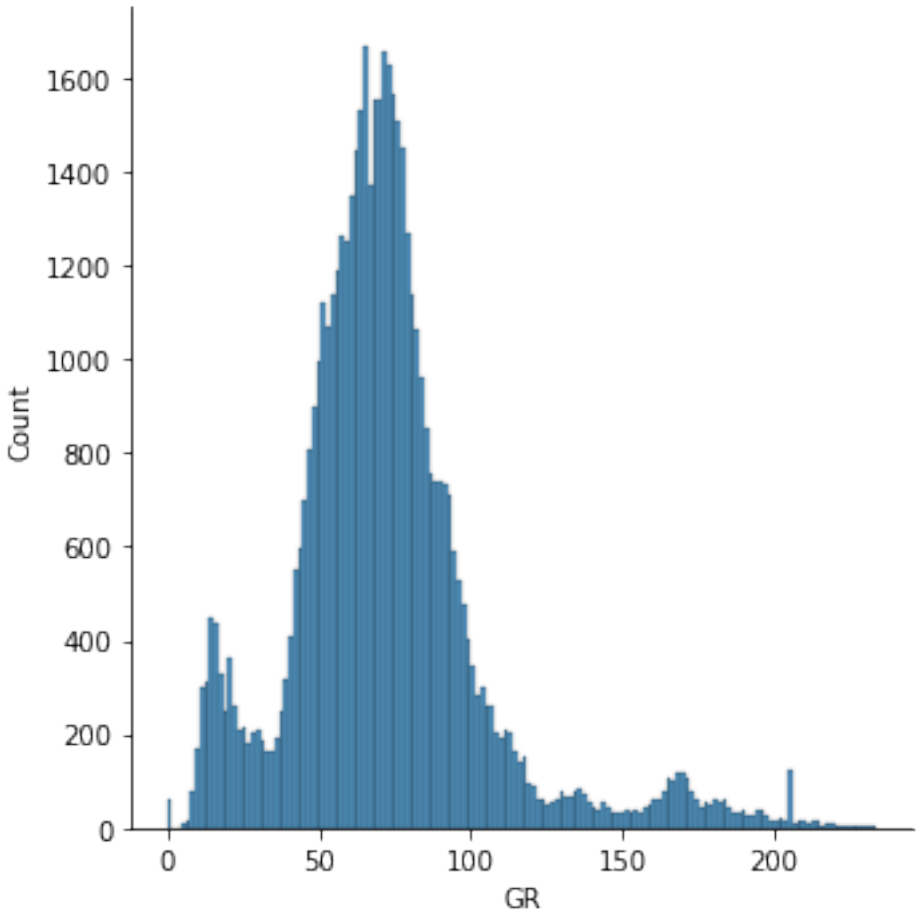
[41]:
```
df.isnull().sum()
```

[41]:
```
DT        0
GR        0
NPHI      0
RHOB      0
FACIES    0
dtype: int64
```

[42]:
```
df.to_csv("FACIES_imputed.csv",index=False)
df=pd.read_csv("FACIES_imputed.csv")
```

[43]:
```
sns.displot(df.GR.dropna())
```

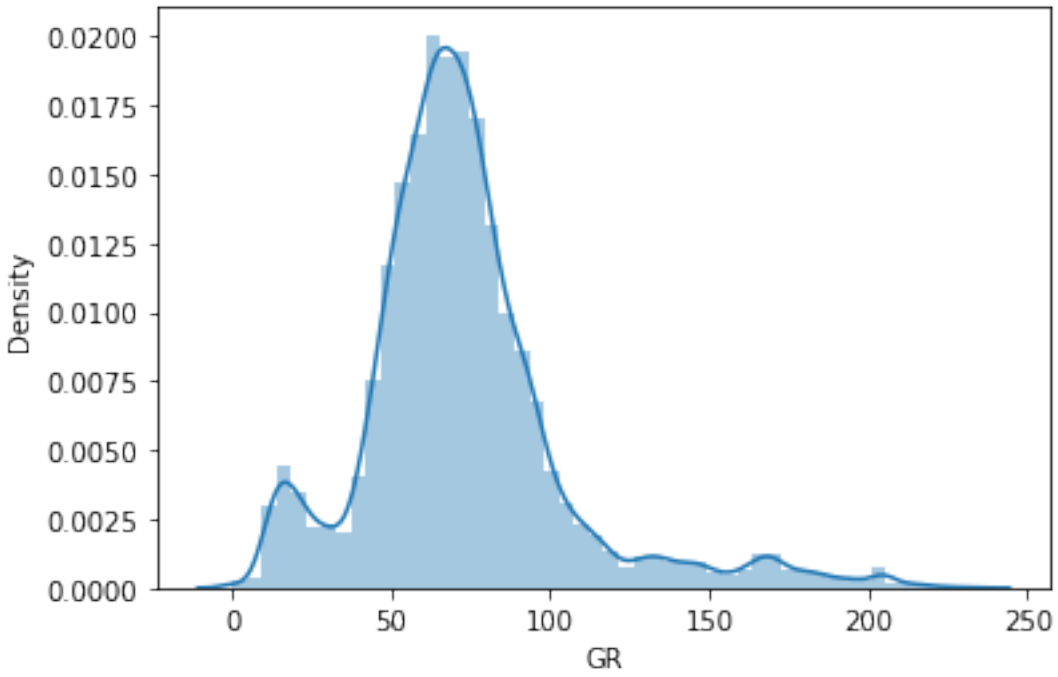
[43]:
```
<seaborn.axisgrid.FacetGrid at 0x7f94e2ee7fd0>
```



[44]:
```
print("WHEN GR WAS NULL")
null_gr.figure
```

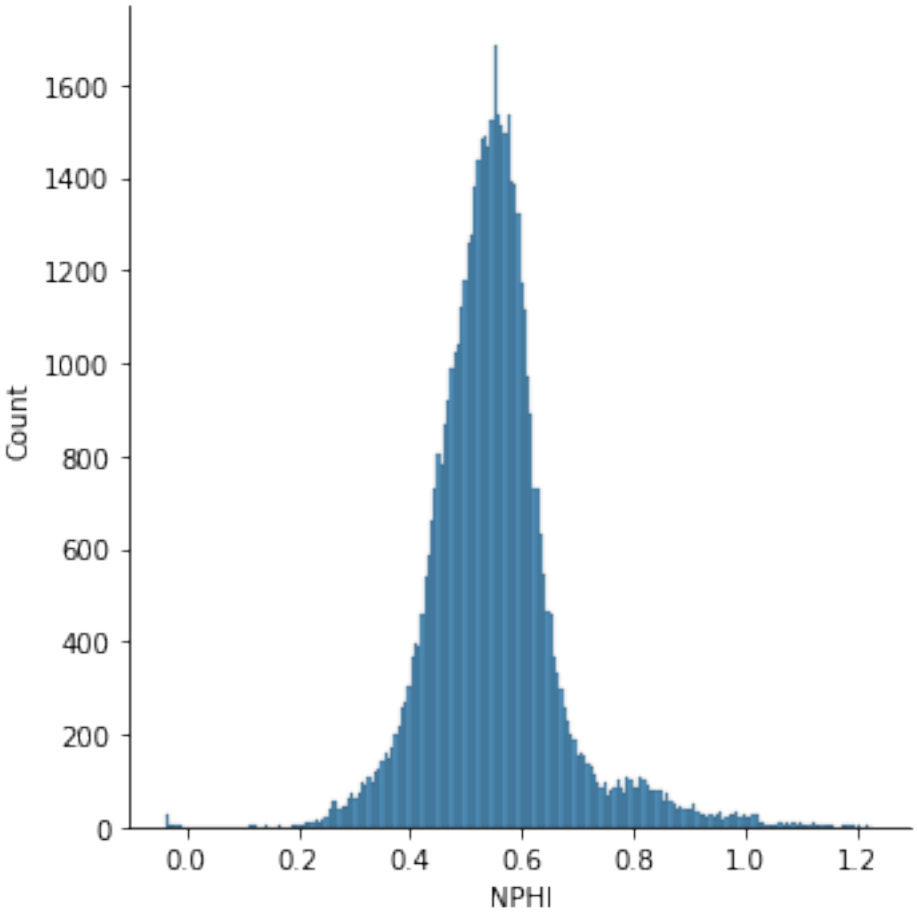
```
WHEN GR WAS NULL
```

[44]:



[45]:
```
sns.displot(df.NPHI.dropna())
```

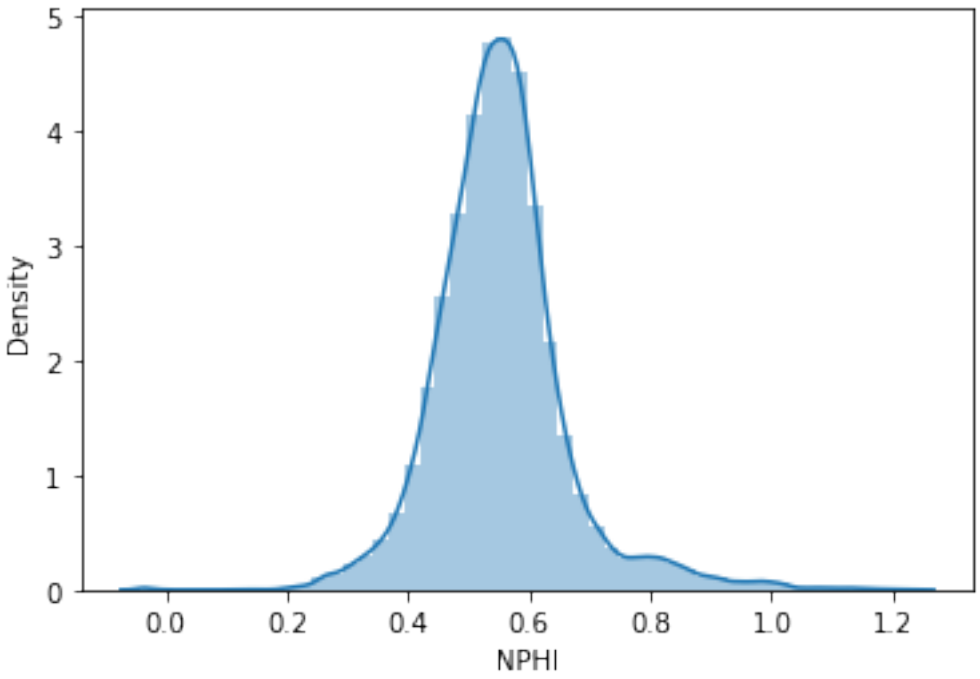
[45]:
```
<seaborn.axisgrid.FacetGrid at 0x7f9405021910>
```



[46]:
```
print("WHEN NPHI WAS NULL")
null_nphi.figure
```

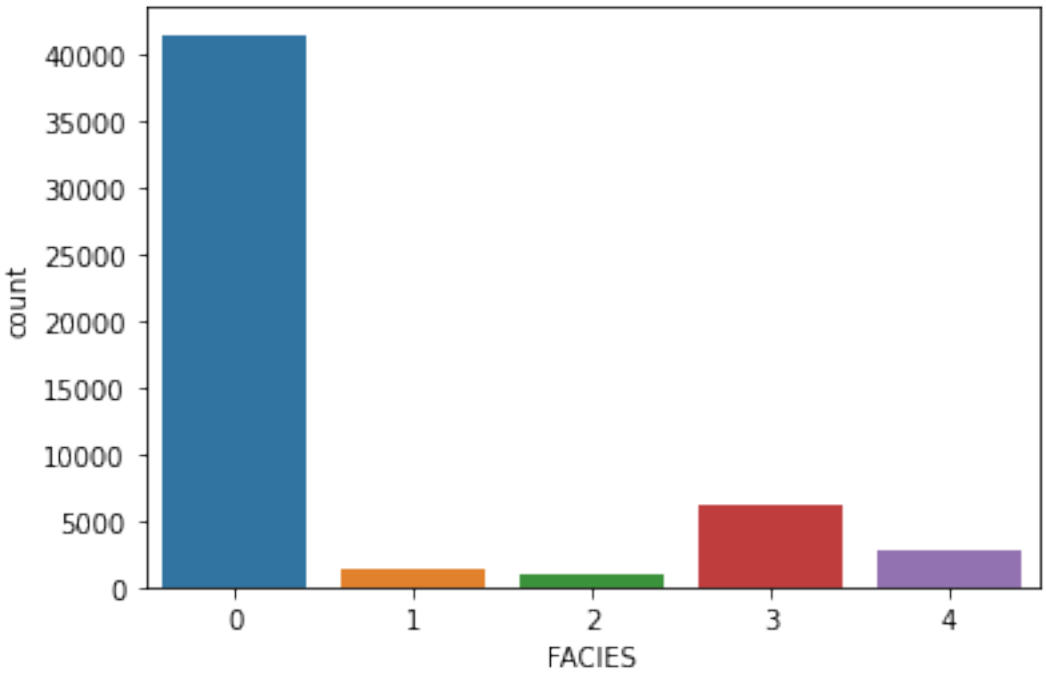
```
WHEN NPHI WAS NULL
```

[46]:



[47]:
```
sns.countplot(x="FACIES",data=df)
```

[47]: <AxesSubplot:xlabel='FACIES', ylabel='count'>



4 DATA CONDITIONING / OUTLIER REMOVAL

[48]:
```
df.head
```

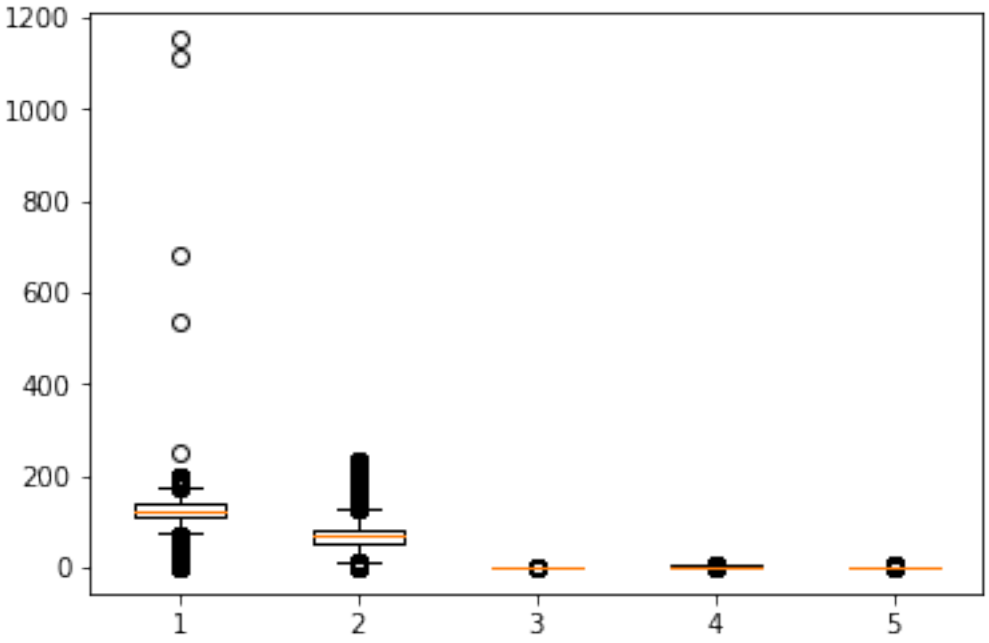
[48]:
```
<bound method NDFrame.head of              DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  72.5921  0.4859  2.1872       0
…           …        …       …       …       …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[52641 rows x 5 columns]>
```

4.1 WHOLE DATA OUTLIER VISUALIZATION

[49]:
```
plt.boxplot(df)
```

[49]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f9403435dc0>,
  <matplotlib.lines.Line2D at 0x7f9403446190>,
  <matplotlib.lines.Line2D at 0x7f9403450760>,
  <matplotlib.lines.Line2D at 0x7f9403450af0>,
  <matplotlib.lines.Line2D at 0x7f94034660d0>,
  <matplotlib.lines.Line2D at 0x7f9403466460>,
  <matplotlib.lines.Line2D at 0x7f94033efa00>,
  <matplotlib.lines.Line2D at 0x7f94033efd90>,
  <matplotlib.lines.Line2D at 0x7f9403406370>,
  <matplotlib.lines.Line2D at 0x7f9403406700>],
 'caps': [<matplotlib.lines.Line2D at 0x7f9403446520>,
  <matplotlib.lines.Line2D at 0x7f94034468b0>,
  <matplotlib.lines.Line2D at 0x7f9403450e80>,
  <matplotlib.lines.Line2D at 0x7f940345b250>,
  <matplotlib.lines.Line2D at 0x7f94034667f0>,
  <matplotlib.lines.Line2D at 0x7f9403466b80>,
  <matplotlib.lines.Line2D at 0x7f94033fb160>,
  <matplotlib.lines.Line2D at 0x7f94033fb4f0>,
  <matplotlib.lines.Line2D at 0x7f9403406a90>,
  <matplotlib.lines.Line2D at 0x7f9403406e20>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f9403435a30>,
  <matplotlib.lines.Line2D at 0x7f94034503d0>,
```

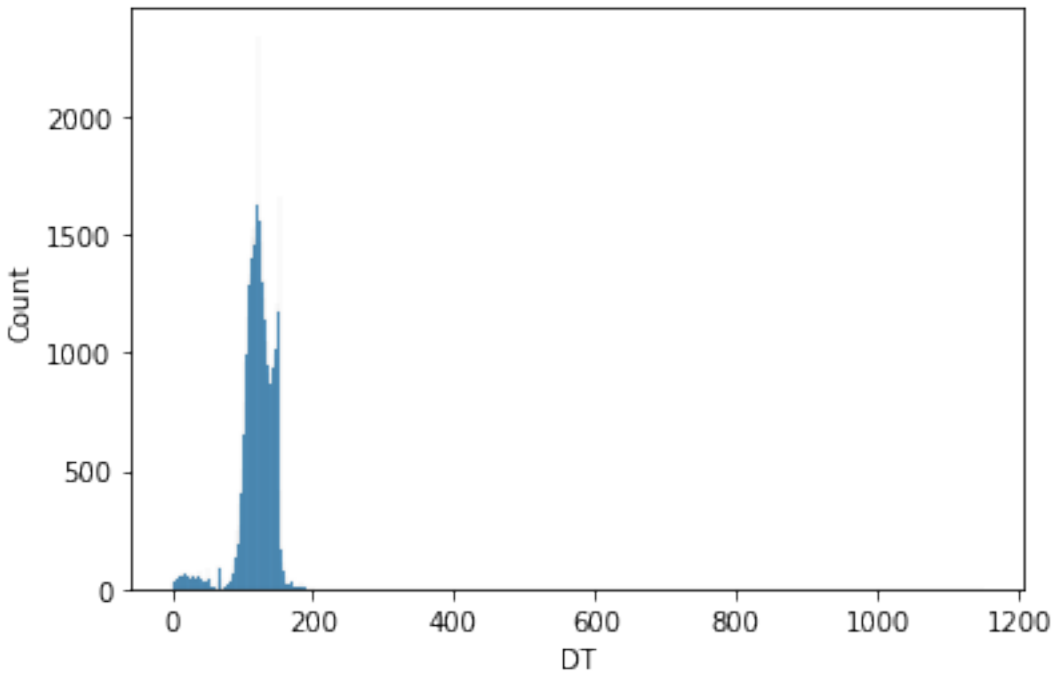
```
 <matplotlib.lines.Line2D at 0x7f940345bd00>,
 <matplotlib.lines.Line2D at 0x7f94033ef670>,
 <matplotlib.lines.Line2D at 0x7f94033fbfa0>],
'medians': [<matplotlib.lines.Line2D at 0x7f9403446c40>,
 <matplotlib.lines.Line2D at 0x7f940345b5e0>,
 <matplotlib.lines.Line2D at 0x7f9403466f10>,
 <matplotlib.lines.Line2D at 0x7f94033fb880>,
 <matplotlib.lines.Line2D at 0x7f94034111f0>],
'fliers': [<matplotlib.lines.Line2D at 0x7f9403446fd0>,
 <matplotlib.lines.Line2D at 0x7f940345b970>,
 <matplotlib.lines.Line2D at 0x7f94033ef2e0>,
 <matplotlib.lines.Line2D at 0x7f94033fbc10>,
 <matplotlib.lines.Line2D at 0x7f9403411580>],
'means': []}
```



4.2 DT VISUALIZATION

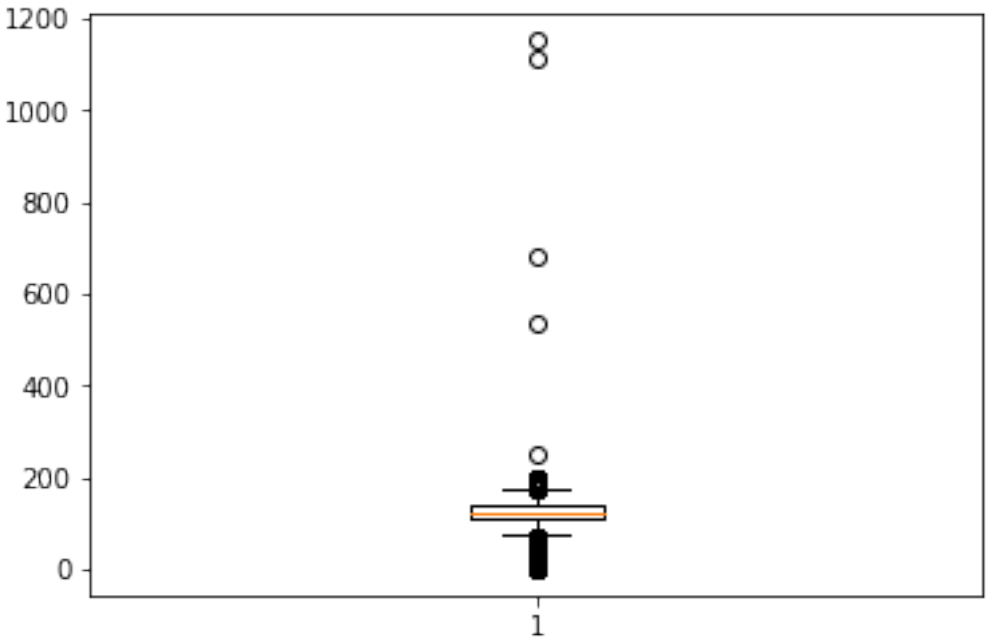
[50]:
```
sns.histplot(df.DT)
```

[50]:
```
<AxesSubplot:xlabel='DT', ylabel='Count'>
```



[51]:
```
plt.boxplot(df["DT"])
```

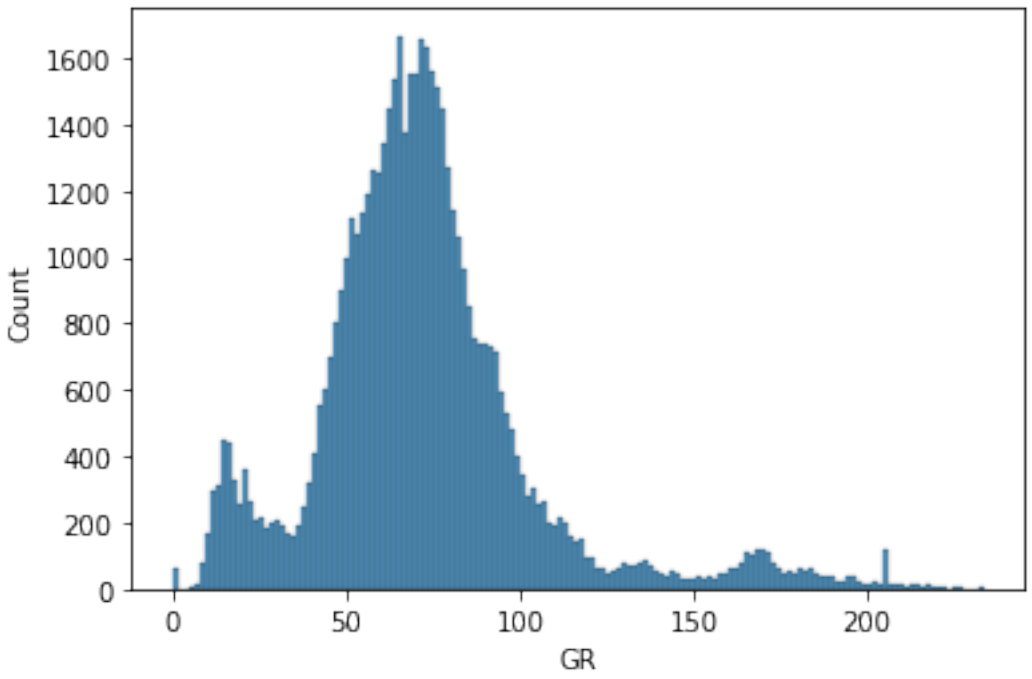
[51]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f67048e0>,
  <matplotlib.lines.Line2D at 0x7f93f6704c70>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f6713040>,
  <matplotlib.lines.Line2D at 0x7f93f67133d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f6704550>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f6713760>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f6713af0>],
 'means': []}
```



4.3 GR VISUALIZATION

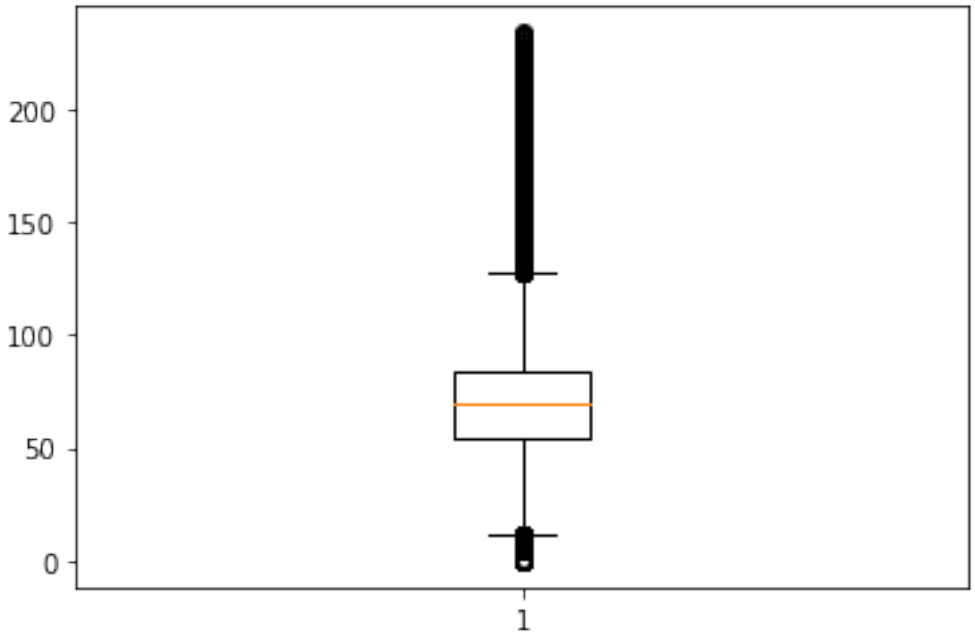
[52]:
```
sns.histplot(df.GR)
```

[52]:
```
<AxesSubplot:xlabel='GR', ylabel='Count'>
```



[53]:
```
plt.boxplot(df.GR)
```

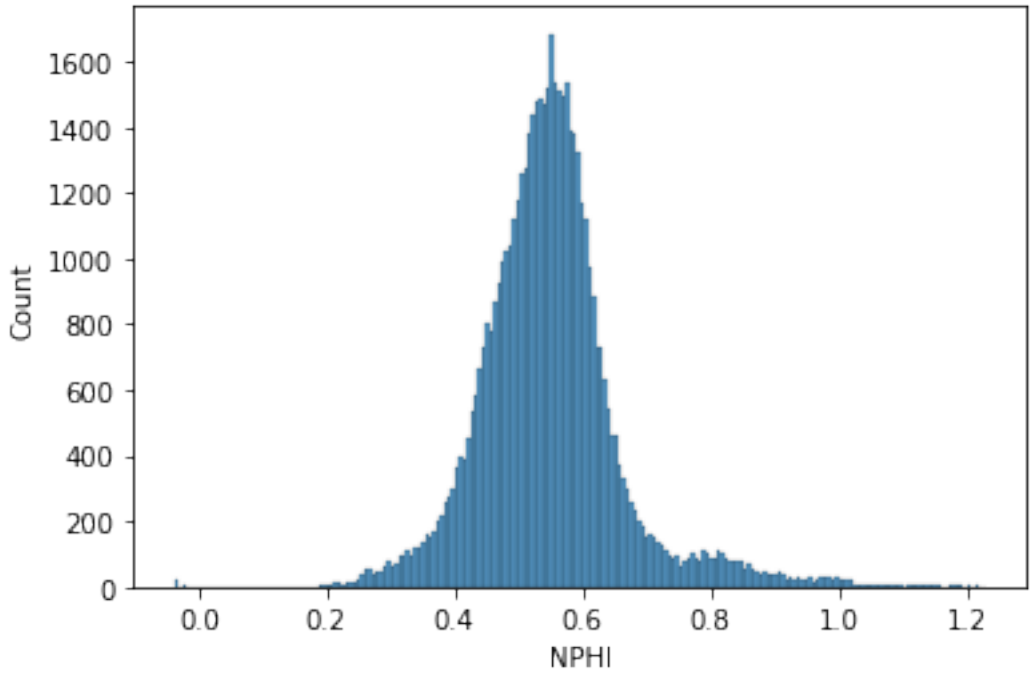
[53]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f6514340>,
  <matplotlib.lines.Line2D at 0x7f93f65146d0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f6514a60>,
  <matplotlib.lines.Line2D at 0x7f93f6514df0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f6507f70>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f65201c0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f6520550>],
 'means': []}
```



4.4 NPHI VISUALIZATION

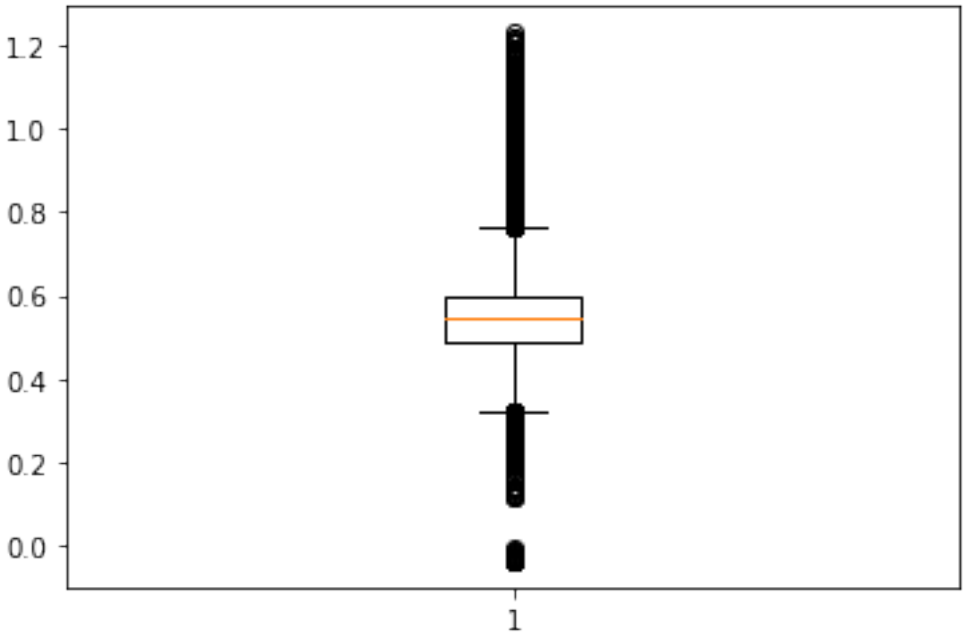
[54]:
```
sns.histplot(df.NPHI)
```

[54]:
```
<AxesSubplot:xlabel='NPHI', ylabel='Count'>
```



[55]:
```
plt.boxplot(df.NPHI)
```

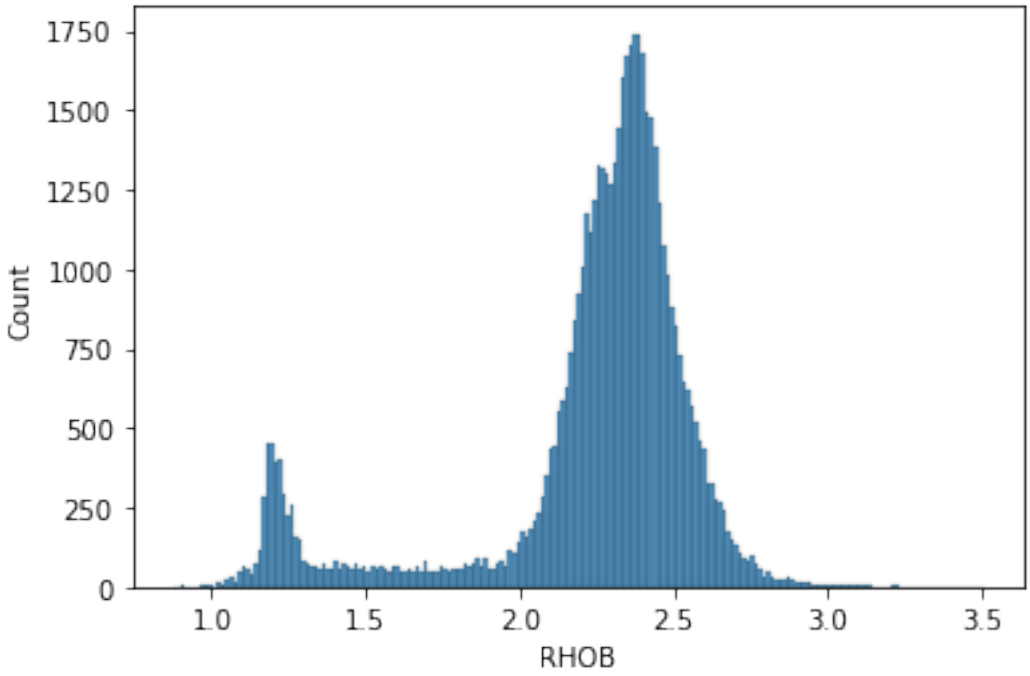
[55]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f629f100>,
  <matplotlib.lines.Line2D at 0x7f93f629f4c0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f629f850>,
  <matplotlib.lines.Line2D at 0x7f93f629fbe0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f6291d30>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f629ff70>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f62ac340>],
 'means': []}
```



4.5 RHOB VISUALIZATION

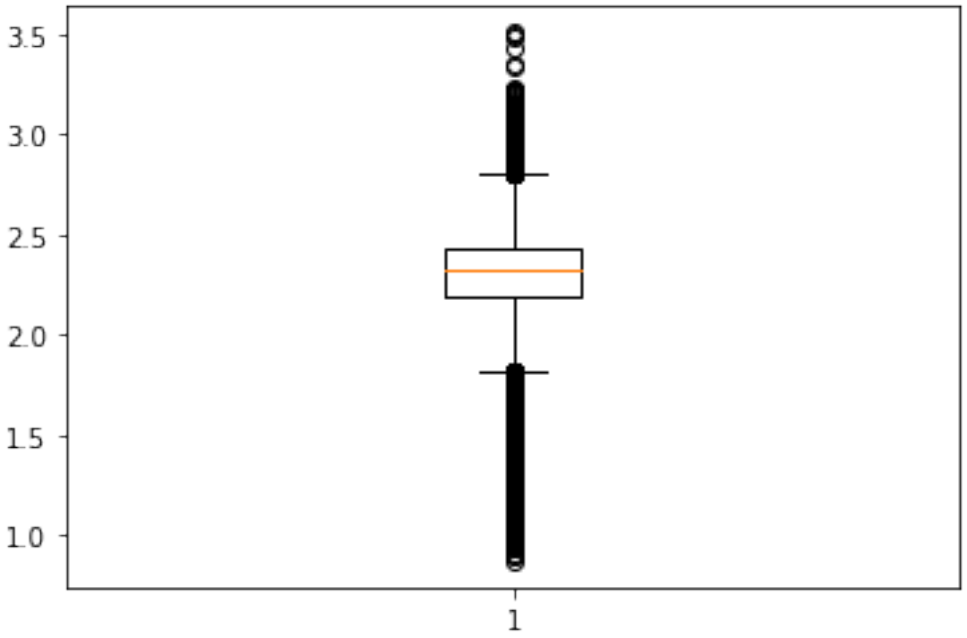
[56]:
```
sns.histplot(df.RHOB)
```

[56]:
```
<AxesSubplot:xlabel='RHOB', ylabel='Count'>
```



[57]:
```
plt.boxplot(df.RHOB)
```

[57]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f94048f6670>,
  <matplotlib.lines.Line2D at 0x7f94048da040>],
 'caps': [<matplotlib.lines.Line2D at 0x7f94048da730>,
  <matplotlib.lines.Line2D at 0x7f94048dabb0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f94048f60a0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f94048daca0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f9404d4d250>],
 'means': []}
```



[58]:
```
def outliers(dataConditioningStrategy,dataframe, dataconditioningcolumns):
    df=dataframe
    if dataConditioningStrategy == "3_Standard_Deviation":
        for column in dataconditioningcolumns:
            print("column",column )
            upperlimit = df[column].mean() + 3*df[column].std()
            lowerlimit = df[column].mean() - 3*df[column].std()

            print("3 standard deviation outliers -:")
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)])
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)].
 ↪shape)
```

```
            df= df[(df[column] < upperlimit) & (df[column] > lowerlimit)]
            print(df)

    elif dataConditioningStrategy == "4_Standard_Deviation":
        for column in dataconditioningcolumns:
            print("column",column )
            upperlimit = df[column].mean() + 4*df[column].std()
            lowerlimit = df[column].mean() - 4*df[column].std()

            print("4 standard deviation outliers -:")
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)])
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)].
↪shape)
            df= df[(df[column] < upperlimit) & (df[column] > lowerlimit)]
            print(df)

    elif dataConditioningStrategy == "InterquartileRange":
        for column in dataconditioningcolumns:
            print("column",column )
            q25, q75 = percentile(df[column], 25), percentile(df[column], 75)
            iqr = q75 - q25
            print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75,␣
↪iqr))
            cut_off = iqr * 1.5
            lowerlimit, upperlimit = q25 - cut_off, q75 + cut_off

            print("InterQuartile Range Outliers-:")
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)])
            print(df[(df[column] > upperlimit) | (df[column] < lowerlimit)].
↪shape)
            df= df[(df[column] < upperlimit) & (df[column] > lowerlimit)]
            print(df)


    return df
```

[59]:
```
DATAConditioningStrategy =␣
↪["3_Standard_Deviation","4_Standard_Deviation","InterquartileRange"]
DATAConditioningColumns = ["DT","GR","NPHI","RHOB"]
optionoutlier = 0
df = outliers(DATAConditioningStrategy[optionoutlier] , df,␣
↪DATAConditioningColumns)
```

```
column DT
3 standard deviation outliers -:
              DT       GR    NPHI    RHOB  FACIES
9        40.9733  79.0259  0.4829  2.2439       0
```

```
10       38.8346  78.1270  0.4736  2.2566       0
11       36.6448  76.3777  0.4676  2.2692       0
12       35.7336  74.5414  0.4657  2.2810       0
13       36.8537  73.1677  0.4648  2.2859       0
…           …       …      …      …      …
4156   1150.8206  93.6033  0.6520  1.8355       0
4157   1109.5558  93.6033  0.6349  1.8700       0
4158    535.0460  93.6033  0.6083  1.8946       0
40167     6.1411  76.3710  0.5301  2.0501       0
40245    14.3304  34.9702  0.5064  2.1492       0

[1610 rows x 5 columns]
(1610, 5)
             DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  72.5921  0.4859  2.1872       0
…          …       …      …      …      …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[51031 rows x 5 columns]
column GR
3 standard deviation outliers -:
             DT        GR     NPHI    RHOB  FACIES
34053  137.1805  171.2380  0.66770  2.2121       0
34054  136.7823  172.2366  0.65630  2.2035       0
34120  132.5928  170.7881  0.64200  2.3634       0
34121  137.5844  171.1272  0.64430  2.3787       0
34122  140.2098  171.6047  0.65090  2.3800       0
…         …        …       …      …      …
36498  125.9000  204.7348  0.55171  2.4790       0
44966  123.8192  174.4802  0.50670  2.4969       0
44967  124.4092  179.8847  0.48940  2.5020       0
44968  120.9927  178.2829  0.47030  2.5231       0
44969  118.3554  173.5724  0.46300  2.5394       0

[1222 rows x 5 columns]
(1222, 5)
             DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
```

```
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  72.5921  0.4859  2.1872       0
…         …       …      …       …      …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[49809 rows x 5 columns]
column NPHI
3 standard deviation outliers -:
             DT       GR    NPHI    RHOB  FACIES
2764   148.2634  14.6269  0.9038  1.1987       3
2765   147.1349  13.6548  0.9360  1.2014       3
2766   146.6394  12.8142  0.9320  1.2041       3
2767   147.0011  12.5580  0.9024  1.2211       3
3025   147.1638  20.4708  0.9567  1.1724       3
…         …       …      …       …      …
48602  113.3730  63.3097  0.9885  2.3216       0
48603  113.3730  63.3097  0.9874  2.3732       0
48604  113.3730  63.3097  0.9888  2.4878       0
48605  113.3730  63.3097  0.9949  2.5784       0
48606  113.3730  63.3097  0.9980  2.6148       0

[873 rows x 5 columns]
(873, 5)
             DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  72.5921  0.4859  2.1872       0
…         …       …      …       …      …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[48936 rows x 5 columns]
column RHOB
3 standard deviation outliers -:
             DT       GR    NPHI    RHOB  FACIES
1389    47.3848  63.7414  0.5042  1.0230       0
1657    45.4202  52.1434  0.6049  0.9053       3
1665    44.8027  56.8584  0.5339  1.0916       3
1666    47.4086  55.1926  0.5349  1.1008       0
```

```
1667    47.2308  52.8494  0.5337  1.0965       0
…          …       …      …       …      …
52517  154.1245  19.2778  0.6490  1.0912       3
52518  153.4207  18.0953  0.6286  1.0916       3
52519  153.3652  17.5801  0.6027  1.0998       3
52520  153.7527  18.1221  0.6070  1.1148       3
52521  152.9592  20.9677  0.6362  1.1651       3

[2061 rows x 5 columns]
(2061, 5)
             DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  72.5921  0.4859  2.1872       0
…          …       …      …       …      …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[46875 rows x 5 columns]
```

[60]:
```
df.shape
```

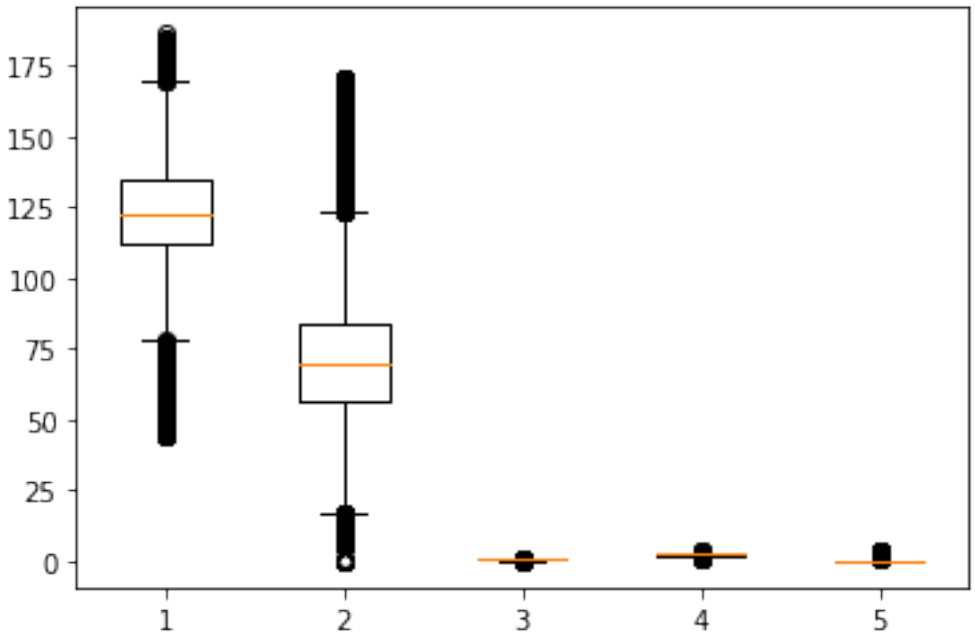
[60]:
```
(46875, 5)
```

4.6 WHOLE DATA AFTER REMOVING OUTLIERS

[61]:
```
plt.boxplot(df)
```

[61]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f9404ecbbb0>,
  <matplotlib.lines.Line2D at 0x7f93f624e100>,
  <matplotlib.lines.Line2D at 0x7f9404d93f10>,
  <matplotlib.lines.Line2D at 0x7f9404f75e50>,
  <matplotlib.lines.Line2D at 0x7f9404f7c6a0>,
  <matplotlib.lines.Line2D at 0x7f9404f7ca30>,
  <matplotlib.lines.Line2D at 0x7f9404f68fd0>,
  <matplotlib.lines.Line2D at 0x7f9404f5e3a0>,
  <matplotlib.lines.Line2D at 0x7f9404f41940>,
  <matplotlib.lines.Line2D at 0x7f9404f41cd0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f62aca90>,
  <matplotlib.lines.Line2D at 0x7f9404f6ad90>,
  <matplotlib.lines.Line2D at 0x7f9404f756a0>,
  <matplotlib.lines.Line2D at 0x7f9404f75940>,
  <matplotlib.lines.Line2D at 0x7f9404f7cdc0>,
```

```
 <matplotlib.lines.Line2D at 0x7f9404f68190>,
 <matplotlib.lines.Line2D at 0x7f9404f5e730>,
 <matplotlib.lines.Line2D at 0x7f9404f5eac0>,
 <matplotlib.lines.Line2D at 0x7f9404ef70a0>,
 <matplotlib.lines.Line2D at 0x7f9404ef7430>],
'boxes': [<matplotlib.lines.Line2D at 0x7f94049098b0>,
 <matplotlib.lines.Line2D at 0x7f9404d80e20>,
 <matplotlib.lines.Line2D at 0x7f9404f7c310>,
 <matplotlib.lines.Line2D at 0x7f9404f68c40>,
 <matplotlib.lines.Line2D at 0x7f9404f415b0>],
'medians': [<matplotlib.lines.Line2D at 0x7f9404d80340>,
 <matplotlib.lines.Line2D at 0x7f9404f75610>,
 <matplotlib.lines.Line2D at 0x7f9404f68520>,
 <matplotlib.lines.Line2D at 0x7f9404f5ee50>,
 <matplotlib.lines.Line2D at 0x7f9404ef77c0>],
'fliers': [<matplotlib.lines.Line2D at 0x7f9404d80130>,
 <matplotlib.lines.Line2D at 0x7f9404f759a0>,
 <matplotlib.lines.Line2D at 0x7f9404f688b0>,
 <matplotlib.lines.Line2D at 0x7f9404f41220>,
 <matplotlib.lines.Line2D at 0x7f9404ef7b50>],
'means': []}
```



[62]:
```
df.head(5)
```

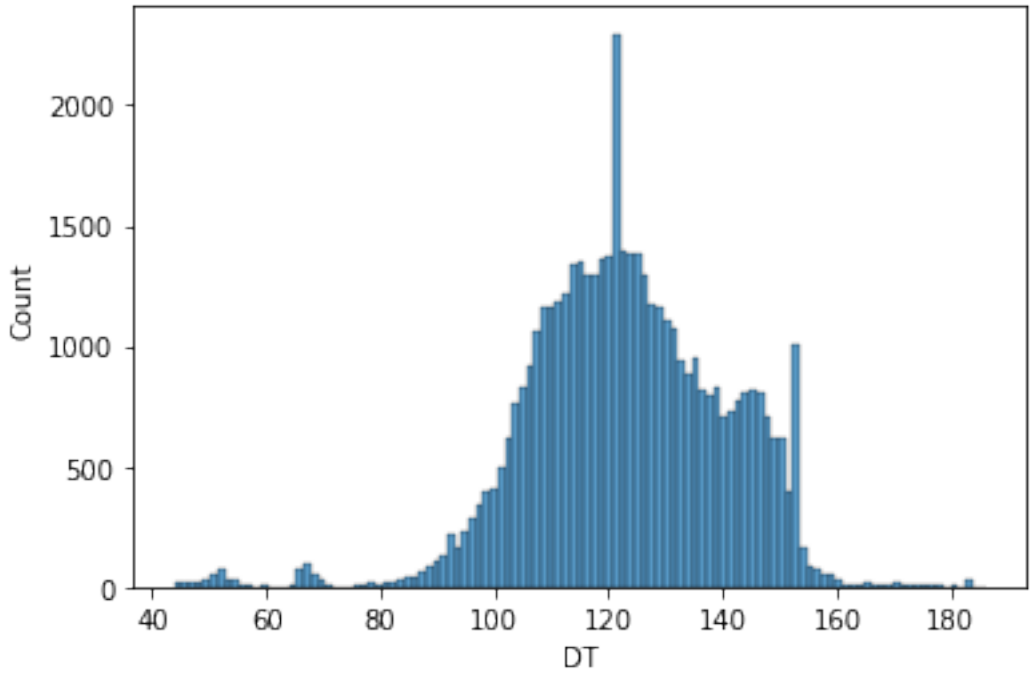
[62]:

	DT	GR	NPHI	RHOB	FACIES
0	51.9301	67.3725	0.5192	2.1625	0
1	49.5776	69.2251	0.5173	2.1624	0
2	48.4933	70.2807	0.5094	2.1608	0
3	48.7997	71.6177	0.4974	2.1703	0
4	49.0683	72.5921	0.4859	2.1872	0

4.7 DT AFTER REMOVING OUTLIER

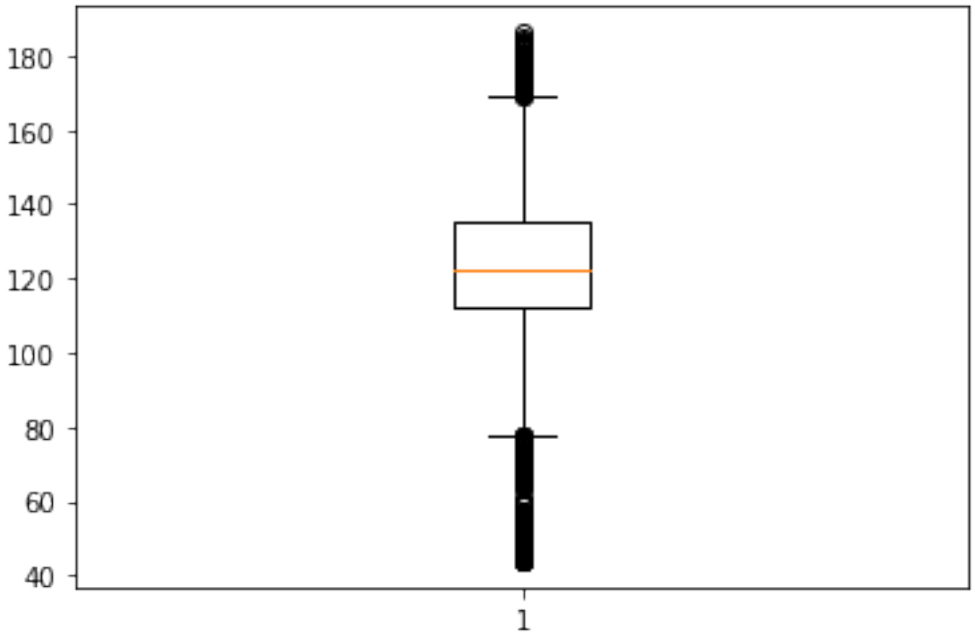
[63]:
```
sns.histplot(df.DT)
```

[63]:
```
<AxesSubplot:xlabel='DT', ylabel='Count'>
```



[64]:
```
plt.boxplot(df["DT"])
```

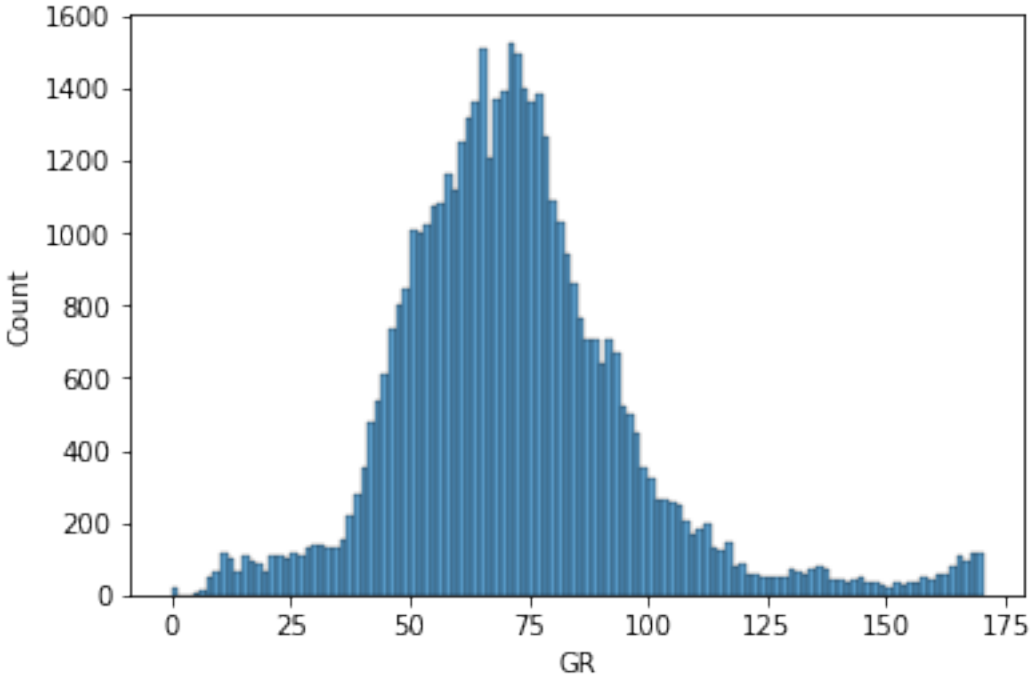
[64]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f9404ccd370>,
  <matplotlib.lines.Line2D at 0x7f9404ccd700>],
 'caps': [<matplotlib.lines.Line2D at 0x7f9404ccda90>,
  <matplotlib.lines.Line2D at 0x7f9404ccde20>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f9404cd8fa0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f9404cc11f0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f9404cc1580>],
 'means': []}
```



4.8 GR AFTER REMOVING OUTLIER

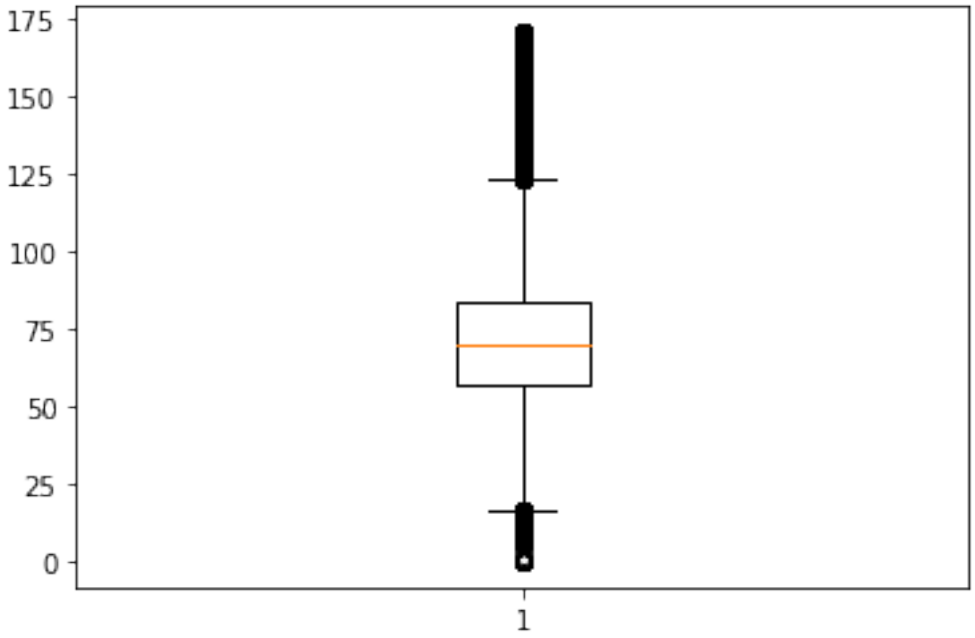
[65]:
```
sns.histplot(df.GR)
```

[65]: <AxesSubplot:xlabel='GR', ylabel='Count'>



[66]:
```
plt.boxplot(df.GR)
```

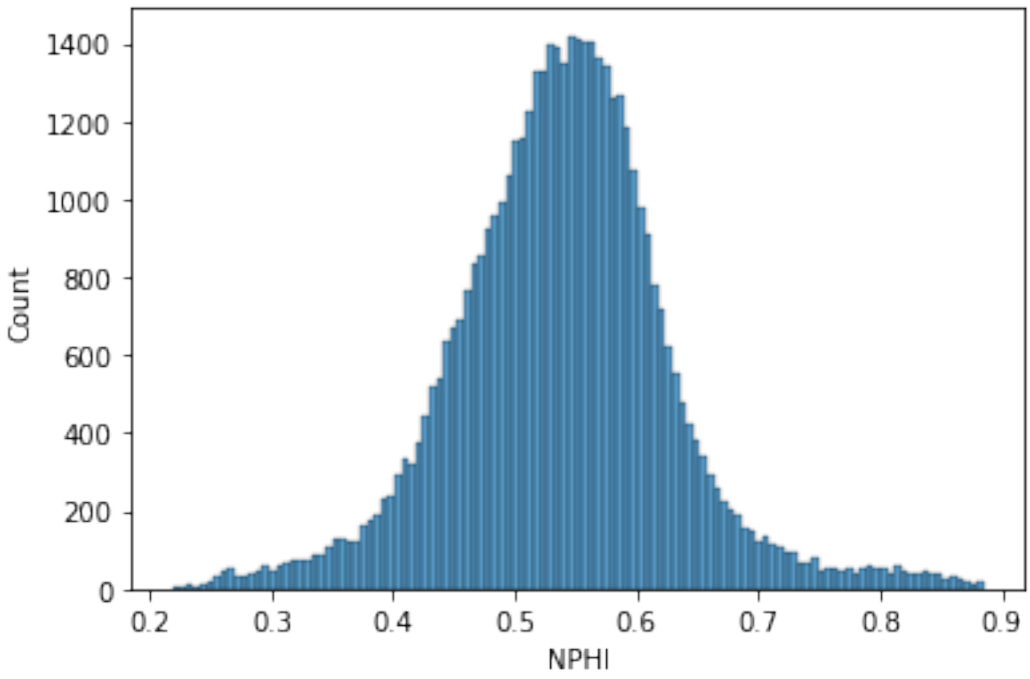
[66]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f5fb6a60>,
  <matplotlib.lines.Line2D at 0x7f93f5fb6df0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f5fc31c0>,
  <matplotlib.lines.Line2D at 0x7f93f5fc3550>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f5fb66d0>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f5fc38e0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f5fc3c70>],
 'means': []}
```



4.9 NPHI AFTER REMOVING OUTLIER

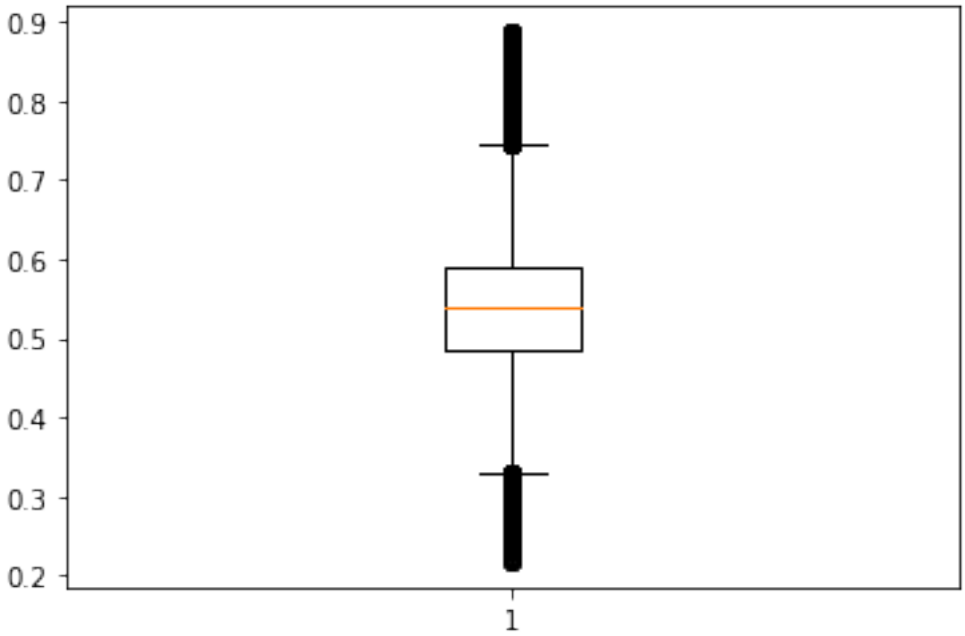
[67]:
```
sns.histplot(df.NPHI)
```

[67]:
```
<AxesSubplot:xlabel='NPHI', ylabel='Count'>
```



[68]:
```
plt.boxplot(df.NPHI)
```

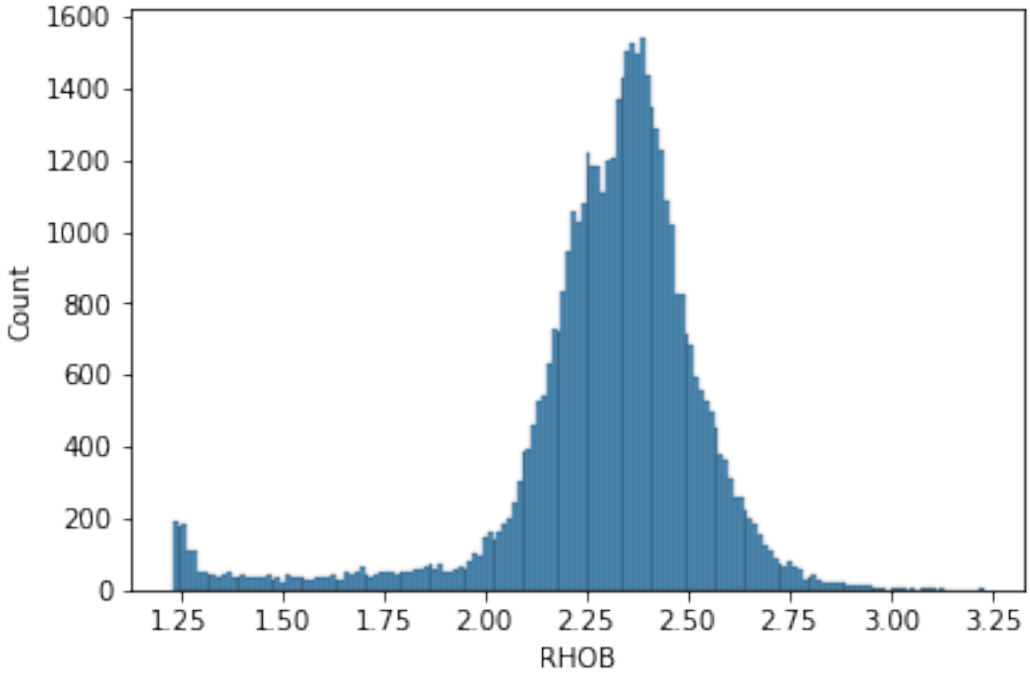
[68]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f5dbfcd0>,
  <matplotlib.lines.Line2D at 0x7f93f5dcc0a0>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f5dcc430>,
  <matplotlib.lines.Line2D at 0x7f93f5dcc7c0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f5dbf940>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f5dccb50>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f5dccee0>],
 'means': []}
```



4.10 RHOB AFTER REMOVING OUTLIER

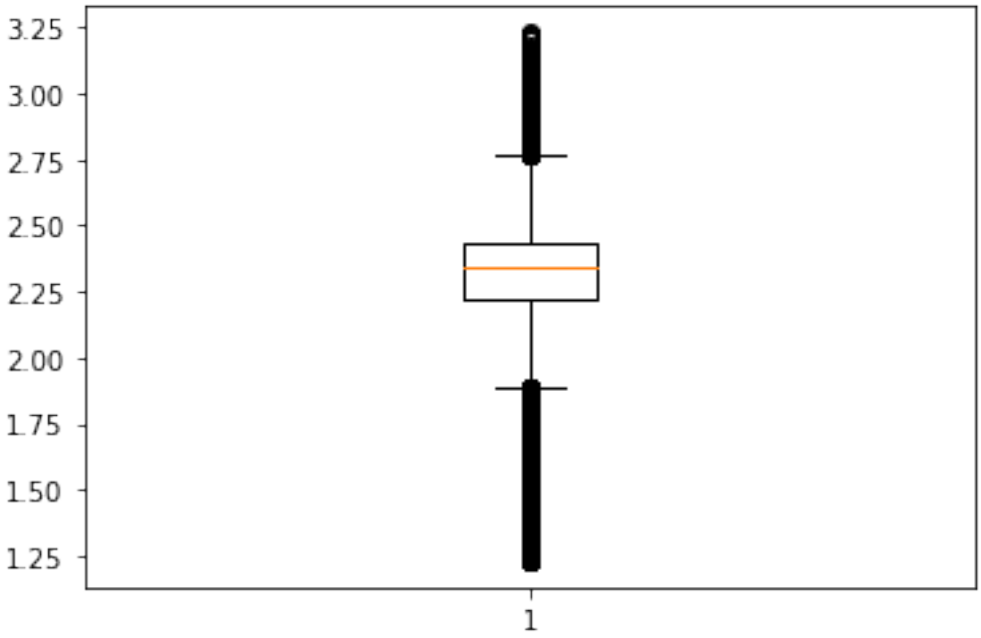
[69]:
```
sns.histplot(df.RHOB)
```

[69]:
```
<AxesSubplot:xlabel='RHOB', ylabel='Count'>
```



[70]:
```
plt.boxplot(df.RHOB)
```

[70]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f93f5b830d0>,
  <matplotlib.lines.Line2D at 0x7f93f5b83460>],
 'caps': [<matplotlib.lines.Line2D at 0x7f93f5b837f0>,
  <matplotlib.lines.Line2D at 0x7f93f5b83b80>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f93f5bf5d00>],
 'medians': [<matplotlib.lines.Line2D at 0x7f93f5b83f10>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f93f5b8d2e0>],
 'means': []}
```



[71]:
```
df
```

[71]:

	DT	GR	NPHI	RHOB	FACIES
0	51.9301	67.3725	0.5192	2.1625	0
1	49.5776	69.2251	0.5173	2.1624	0
2	48.4933	70.2807	0.5094	2.1608	0
3	48.7997	71.6177	0.4974	2.1703	0
4	49.0683	72.5921	0.4859	2.1872	0
…	…	…	…	…	…
52636	108.8188	74.6901	0.4541	2.7261	0
52637	109.9238	72.0000	0.4548	2.6856	0
52638	113.8166	74.1318	0.4780	2.6126	0

```
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[46875 rows x 5 columns]
```

5 FEATURE SELECTION

[72]:
```
df.head(10)
```

[72]:
```
          DT       GR    NPHI    RHOB  FACIES
0   51.9301  67.3725  0.5192  2.1625       0
1   49.5776  69.2251  0.5173  2.1624       0
2   48.4933  70.2807  0.5094  2.1608       0
3   48.7997  71.6177  0.4974  2.1703       0
4   49.0683  72.5921  0.4859  2.1872       0
5   49.2140  73.8317  0.4825  2.2036       0
6   48.4738  75.8763  0.4864  2.2170       0
7   46.6610  78.3465  0.4904  2.2253       0
8   43.9641  79.4059  0.4898  2.2329       0
56  45.4016  69.1220  0.4346  2.2947       0
```

[73]:
```
df.shape
```

[73]:
```
(46875, 5)
```

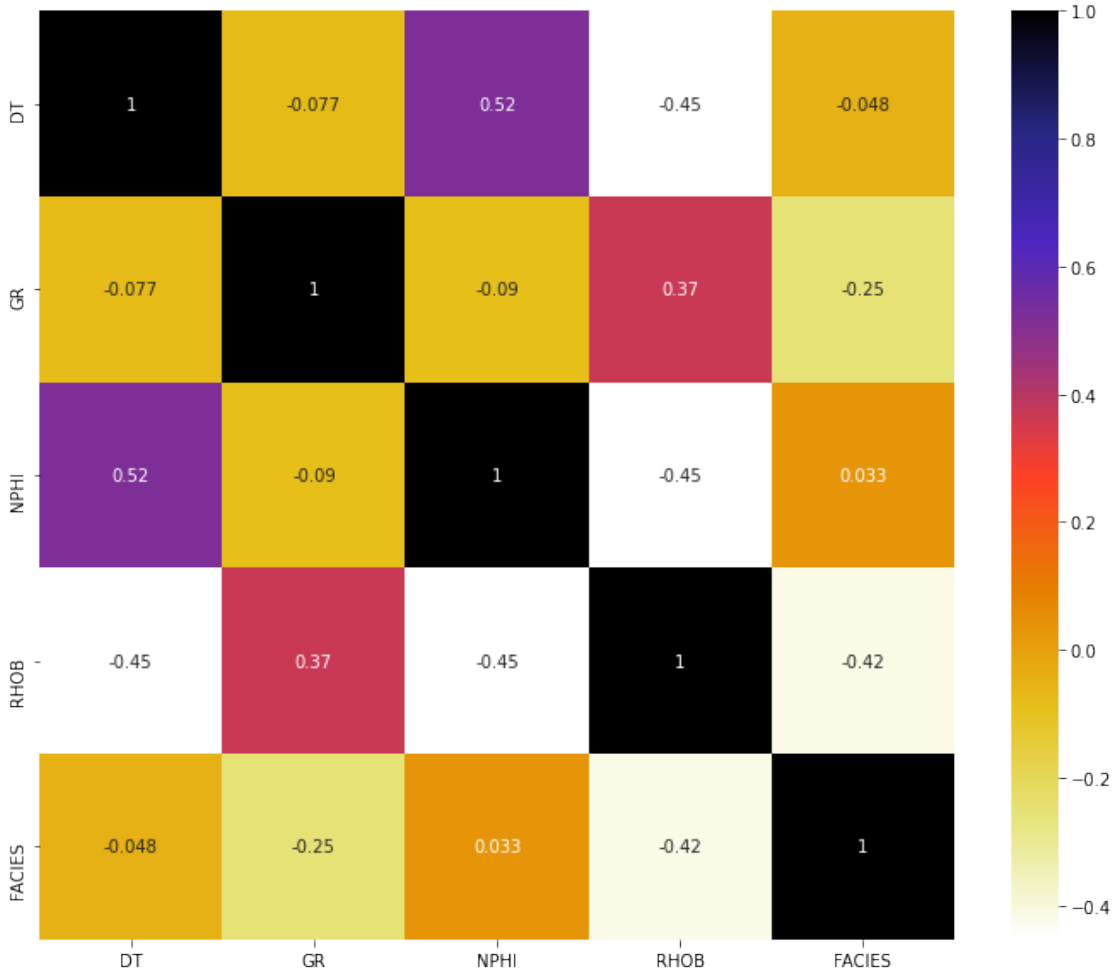
[74]:
```
features = df.shape[1]
features
```

[74]:
```
5
```

[75]:
```
df.var()
```

[75]:
```
DT        298.787525
GR        627.603951
NPHI        0.008011
RHOB        0.067956
FACIES      1.392844
dtype: float64
```

[76]:
```
plt.figure(figsize=(12,10))
cor = df.corr()
sns.heatmap(cor , annot=True , cmap=plt.cm.CMRmap_r)
plt.show()
```



[77]:
```
def FeatureSelection(FeatureSelectionStrategy,dataframe):
    df=dataframe

    if(FeatureSelectionStrategy=="Variance_Threshold"):
        var_thres=VarianceThreshold(threshold=0.0)
        var_thres.fit(df)
        df.columns[var_thres.get_support()]
        cols = [column for column in df.columns
                    if column not in df.columns[var_thres.get_support()]]
        print(cols)
        df = df.drop(cols,axis=1)
        return df

    if(FeatureSelectionStrategy=="Absolute_Correlation"):
        threshold = 0.6
        col_corr = set()
```

```
        corr_matrix = df.corr()
        for i in range(len(corr_matrix.columns)):
            for j in range(i):
                if abs(corr_matrix.iloc[i,j]) > threshold :
                    colname = corr_matrix.columns[i]
                    print(colname)
                    col_corr.add(colname)
        df = df.drop(col_corr,axis=1)
        return df

    if(FeatureSelectionStrategy=="Correlation"):
        threshold = 0.6
        col_corr = set()
        corr_matrix = df.corr()
        for i in range(len(corr_matrix.columns)):
            for j in range(i):
                if (corr_matrix.iloc[i,j]) > threshold :
                    colname = corr_matrix.columns[i]
                    print(colname)
                    col_corr.add(colname)
        df = df.drop(col_corr,axis=1)
        return df

    if(FeatureSelectionStrategy == "SelectKBest"):
        mutual_info = mutual_info_classif(df)
        print(mutual_info)
        mutual_info=pd.Series(mutual_info)
        mutual_info.sort_values(ascending=False)
        mutual_info.sort_values(ascending=False).plot.bar(figsize=(20,8))
        select_col = SelectKBest(mutual_info_classif,k=1)
        select_col.fit(df)
        column1 = df.columns[select_col.get_support()]
        df = df.drop(column1,axis=1)
        return df

    if(FeatureSelectionStrategy == "Mutual_Info_Class"):
        mutual_info = mutual_info_classif(df)
        print(mutual_info)
        mutual_info=pd.Series(mutual_info)
        mutual_info.sort_values(ascending=False)
        mutual_info.sort_values(ascending=False).plot.bar(figsize=(20,8))
        return df
```

[78]:
```
FeatureSelectionStrategy=["Variance_Threshold","Absolute_Correlation","Correlation","SelectKBe
optionfeature = 0
df=FeatureSelection(FeatureSelectionStrategy[optionfeature],df)
```

```
[]
```

[79]:
```
print("Deleted feature(s) = " + str(features-df.shape[1]))
```

```
Deleted feature(s) = 0
```

[80]:
```
df
```

[80]:

	DT	GR	NPHI	RHOB	FACIES
0	51.9301	67.3725	0.5192	2.1625	0
1	49.5776	69.2251	0.5173	2.1624	0
2	48.4933	70.2807	0.5094	2.1608	0
3	48.7997	71.6177	0.4974	2.1703	0
4	49.0683	72.5921	0.4859	2.1872	0
…	…	…	…	…	…
52636	108.8188	74.6901	0.4541	2.7261	0
52637	109.9238	72.0000	0.4548	2.6856	0
52638	113.8166	74.1318	0.4780	2.6126	0
52639	120.0651	78.9290	0.4991	2.5728	0
52640	123.0664	82.8848	0.5138	2.5918	0

[46875 rows x 5 columns]

[81]:
```
df
```

[81]:

	DT	GR	NPHI	RHOB	FACIES
0	51.9301	67.3725	0.5192	2.1625	0
1	49.5776	69.2251	0.5173	2.1624	0
2	48.4933	70.2807	0.5094	2.1608	0
3	48.7997	71.6177	0.4974	2.1703	0
4	49.0683	72.5921	0.4859	2.1872	0
…	…	…	…	…	…
52636	108.8188	74.6901	0.4541	2.7261	0
52637	109.9238	72.0000	0.4548	2.6856	0
52638	113.8166	74.1318	0.4780	2.6126	0
52639	120.0651	78.9290	0.4991	2.5728	0
52640	123.0664	82.8848	0.5138	2.5918	0

[46875 rows x 5 columns]

6 SCALING DATA

[82]:
```
df
```

[82]:

	DT	GR	NPHI	RHOB	FACIES
0	51.9301	67.3725	0.5192	2.1625	0
1	49.5776	69.2251	0.5173	2.1624	0
2	48.4933	70.2807	0.5094	2.1608	0
3	48.7997	71.6177	0.4974	2.1703	0

```
4       49.0683  72.5921  0.4859  2.1872       0
…          …       …       …       …          …
52636  108.8188  74.6901  0.4541  2.7261       0
52637  109.9238  72.0000  0.4548  2.6856       0
52638  113.8166  74.1318  0.4780  2.6126       0
52639  120.0651  78.9290  0.4991  2.5728       0
52640  123.0664  82.8848  0.5138  2.5918       0

[46875 rows x 5 columns]
```

[83]:
```
def data_scaling( scaling_strategy , scaling_data , scaling_columns ):

    if scaling_strategy =="RobustScaler" :
        scaling_data[scaling_columns] = RobustScaler().
↪fit_transform(scaling_data[scaling_columns])

    elif scaling_strategy =="MinMaxScaler" :
        scaling_data[scaling_columns] = MinMaxScaler().
↪fit_transform(scaling_data[scaling_columns])

    else : # If any other scaling send by mistake still perform Robust Scalar
        scaling_data[scaling_columns] = RobustScaler().
↪fit_transform(scaling_data[scaling_columns])

    return scaling_data
```

[84]:
```
scaling_strategy = ["RobustScaler","MinMaxScaler"]
optionscaling = 0
df = data_scaling( scaling_strategy[optionscaling] , df ,␣
↪DATAConditioningColumns )
```

[85]:
```
df
```

[85]:
```
             DT        GR      NPHI      RHOB  FACIES
0     -3.084459 -0.101928 -0.207429 -0.799544       0
1     -3.187408 -0.032520 -0.225760 -0.800000       0
2     -3.234859  0.007028 -0.301978 -0.807289       0
3     -3.221450  0.057119 -0.417752 -0.764009       0
4     -3.209696  0.093625 -0.528702 -0.687016       0
…           …         …        …         …        …
52636 -0.594932  0.172227 -0.835504  1.768109       0
52637 -0.546575  0.071442 -0.828751  1.583599       0
52638 -0.376221  0.151310 -0.604920  1.251025       0
52639 -0.102778  0.331038 -0.401351  1.069704       0
52640  0.028563  0.479242 -0.259527  1.156264       0

[46875 rows x 5 columns]
```

[86]:
```
df.to_csv("Preprocessed_data.csv",index=False)
```

7 SPLITTING DATA USING TRAIN_TEST_SPLIT

[87]:
```
df=pd.read_csv('Preprocessed_data.csv')
```

[88]:
```
df.head()
```

[88]:

	DT	GR	NPHI	RHOB	FACIES
0	-3.084459	-0.101928	-0.207429	-0.799544	0
1	-3.187408	-0.032520	-0.225760	-0.800000	0
2	-3.234859	0.007028	-0.301978	-0.807289	0
3	-3.221450	0.057119	-0.417752	-0.764009	0
4	-3.209696	0.093625	-0.528702	-0.687016	0

[89]:
```
df.isnull().sum()
```

[89]:
```
DT        0
GR        0
NPHI      0
RHOB      0
FACIES    0
dtype: int64
```

[90]:
```
x = df.drop("FACIES",1)
y = df["FACIES"]
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,␣
 ↪random_state=8)
```

[91]:
```
X_train.shape
```

[91]:
```
(32812, 4)
```

[92]:
```
X_test.shape
```

[92]:
```
(14063, 4)
```

[93]:
```
X_test
```

[93]:

	DT	GR	NPHI	RHOB
33256	-0.493335	-0.075586	-0.523878	0.646925
11725	0.072027	0.188663	0.171732	-1.109339
20519	-0.568745	0.056490	-0.410034	0.482460
26253	0.549245	0.060551	1.249397	-1.707973
9984	0.552277	0.012034	0.251809	-0.532574
…	…	…	…	…
8602	-0.585234	0.356278	0.312590	0.591800

```
5048  -0.295162 -0.082929  0.149542  0.357631
45920 -0.332328  0.303628 -0.834539  0.459681
45651 -0.695381  1.186891 -1.849493  0.602278
23181  0.942213 -0.995309  0.414858 -0.217768

[14063 rows x 4 columns]
```

8 MODEL TRAINING

[94]:
```
estimator=[]
```

[95]:
```
gnb = GaussianNB()
```

[96]:
```
model = LogisticRegression()
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values = [100, 10, 1.0, 0.1, 0.01]

grid = {'solver':solvers,'penalty':penalty,'C':c_values}
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
grid_search = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1, cv=cv,␣
 ↪scoring='accuracy',error_score=0)
grid_result = grid_search.fit(X_train, y_train)

print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("%f (%f) with: %r" % (mean, stdev, param))
```

```
Best: 0.883345 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.883345 (0.003631) with: {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.883335 (0.003642) with: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
0.882462 (0.003285) with: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
0.883345 (0.003631) with: {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.883345 (0.003631) with: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
0.882462 (0.003300) with: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
0.883295 (0.003577) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'newton-cg'}
0.883295 (0.003577) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
0.882482 (0.003294) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}
0.883213 (0.003527) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
0.883203 (0.003530) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
0.882319 (0.003180) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
0.881730 (0.003148) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
0.881730 (0.003148) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
0.880562 (0.003009) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
```

[97]:
```
dtclf = DecisionTreeClassifier(max_depth=5)
```

[98]:
```
cat = CatBoostClassifier()
```

[99]:
```
xgb= XGBClassifier(learning_rate =0.09,
n_estimators=494,
max_depth=5,
subsample = 0.70,
verbosity = 0,)
```

[100]:
```
lgbm=LGBMClassifier(importance_type = "gain",
verbosity = -1,
max_bin = 60,
num_leaves=300,
boosting_type = 'dart',
learning_rate=0.1,
n_estimators=494,
max_depth=5, )
```

[101]:
```
rdmclf = RandomForestClassifier(n_estimators=494,max_depth=5)
```

[102]:
```
estimator.append(('gaussian',gnb))
estimator.append(('Gridlogistic',grid_search))
estimator.append(('catboost_classifier',cat))
estimator.append(('decision_tree',dtclf))
estimator.append(('xgbclassifier',xgb))
estimator.append(('LGBMclassifier',lgbm))
```

[103]:
```
vot_soft = VotingClassifier(estimators = estimator, voting ='soft')
```

[104]:
```
vot_soft.fit(X_train,y_train)
```

```
Learning rate set to 0.094546
0:      learn: 1.3401739        total: 57.4ms   remaining: 57.4s
1:      learn: 1.1634978        total: 63.8ms   remaining: 31.8s
2:      learn: 1.0311001        total: 70.5ms   remaining: 23.4s
3:      learn: 0.9295236        total: 76.7ms   remaining: 19.1s
4:      learn: 0.8481459        total: 82.6ms   remaining: 16.4s
5:      learn: 0.7817915        total: 89.1ms   remaining: 14.8s
6:      learn: 0.7250768        total: 95ms     remaining: 13.5s
7:      learn: 0.6770088        total: 103ms    remaining: 12.7s
8:      learn: 0.6366800        total: 109ms    remaining: 12s
9:      learn: 0.6013783        total: 116ms    remaining: 11.5s
10:     learn: 0.5719576        total: 123ms    remaining: 11.1s
11:     learn: 0.5461442        total: 129ms    remaining: 10.6s
12:     learn: 0.5229368        total: 136ms    remaining: 10.3s
13:     learn: 0.5031472        total: 142ms    remaining: 10s
14:     learn: 0.4856426        total: 149ms    remaining: 9.8s
```

```
15:     learn: 0.4695554        total: 156ms    remaining: 9.58s
16:     learn: 0.4551825        total: 163ms    remaining: 9.44s
17:     learn: 0.4429784        total: 170ms    remaining: 9.26s
18:     learn: 0.4315076        total: 176ms    remaining: 9.1s
19:     learn: 0.4215592        total: 183ms    remaining: 8.96s
20:     learn: 0.4122966        total: 190ms    remaining: 8.86s
21:     learn: 0.4038830        total: 197ms    remaining: 8.76s
22:     learn: 0.3968199        total: 204ms    remaining: 8.67s
23:     learn: 0.3895913        total: 212ms    remaining: 8.62s
24:     learn: 0.3842812        total: 219ms    remaining: 8.54s
25:     learn: 0.3790941        total: 227ms    remaining: 8.49s
26:     learn: 0.3736402        total: 234ms    remaining: 8.44s
27:     learn: 0.3691355        total: 241ms    remaining: 8.37s
28:     learn: 0.3647643        total: 248ms    remaining: 8.31s
29:     learn: 0.3608892        total: 255ms    remaining: 8.25s
30:     learn: 0.3577580        total: 266ms    remaining: 8.32s
31:     learn: 0.3544054        total: 273ms    remaining: 8.27s
32:     learn: 0.3512266        total: 282ms    remaining: 8.25s
33:     learn: 0.3482574        total: 289ms    remaining: 8.21s
34:     learn: 0.3457946        total: 297ms    remaining: 8.18s
35:     learn: 0.3431511        total: 305ms    remaining: 8.15s
36:     learn: 0.3402250        total: 312ms    remaining: 8.12s
37:     learn: 0.3383668        total: 319ms    remaining: 8.08s
38:     learn: 0.3364603        total: 326ms    remaining: 8.03s
39:     learn: 0.3350029        total: 332ms    remaining: 7.97s
40:     learn: 0.3336726        total: 349ms    remaining: 8.15s
41:     learn: 0.3311375        total: 361ms    remaining: 8.23s
42:     learn: 0.3298983        total: 370ms    remaining: 8.23s
43:     learn: 0.3288313        total: 377ms    remaining: 8.2s
44:     learn: 0.3276480        total: 385ms    remaining: 8.18s
45:     learn: 0.3261373        total: 393ms    remaining: 8.14s
46:     learn: 0.3250302        total: 400ms    remaining: 8.11s
47:     learn: 0.3243349        total: 407ms    remaining: 8.07s
48:     learn: 0.3225773        total: 416ms    remaining: 8.07s
49:     learn: 0.3216461        total: 423ms    remaining: 8.04s
50:     learn: 0.3206880        total: 433ms    remaining: 8.06s
51:     learn: 0.3197236        total: 439ms    remaining: 8.01s
52:     learn: 0.3186838        total: 446ms    remaining: 7.97s
53:     learn: 0.3178860        total: 452ms    remaining: 7.92s
54:     learn: 0.3170903        total: 459ms    remaining: 7.88s
55:     learn: 0.3166773        total: 464ms    remaining: 7.83s
56:     learn: 0.3157539        total: 471ms    remaining: 7.79s
57:     learn: 0.3148677        total: 478ms    remaining: 7.76s
58:     learn: 0.3142872        total: 485ms    remaining: 7.73s
59:     learn: 0.3138748        total: 492ms    remaining: 7.7s
60:     learn: 0.3133571        total: 500ms    remaining: 7.69s
61:     learn: 0.3126248        total: 508ms    remaining: 7.69s
62:     learn: 0.3119687        total: 515ms    remaining: 7.65s
```

```
63:     learn: 0.3112866        total: 523ms    remaining: 7.66s
64:     learn: 0.3105467        total: 530ms    remaining: 7.63s
65:     learn: 0.3095473        total: 537ms    remaining: 7.6s
66:     learn: 0.3090181        total: 544ms    remaining: 7.58s
67:     learn: 0.3084832        total: 551ms    remaining: 7.55s
68:     learn: 0.3079598        total: 557ms    remaining: 7.52s
69:     learn: 0.3074483        total: 564ms    remaining: 7.49s
70:     learn: 0.3065998        total: 572ms    remaining: 7.49s
71:     learn: 0.3062533        total: 578ms    remaining: 7.45s
72:     learn: 0.3057903        total: 585ms    remaining: 7.43s
73:     learn: 0.3052990        total: 593ms    remaining: 7.42s
74:     learn: 0.3049843        total: 602ms    remaining: 7.42s
75:     learn: 0.3046648        total: 609ms    remaining: 7.4s
76:     learn: 0.3041125        total: 616ms    remaining: 7.38s
77:     learn: 0.3037677        total: 622ms    remaining: 7.35s
78:     learn: 0.3032730        total: 628ms    remaining: 7.33s
79:     learn: 0.3027455        total: 635ms    remaining: 7.3s
80:     learn: 0.3021777        total: 642ms    remaining: 7.28s
81:     learn: 0.3019483        total: 648ms    remaining: 7.25s
82:     learn: 0.3015734        total: 654ms    remaining: 7.23s
83:     learn: 0.3011265        total: 660ms    remaining: 7.2s
84:     learn: 0.3007856        total: 668ms    remaining: 7.19s
85:     learn: 0.3002689        total: 674ms    remaining: 7.17s
86:     learn: 0.2999763        total: 681ms    remaining: 7.15s
87:     learn: 0.2996514        total: 688ms    remaining: 7.13s
88:     learn: 0.2991584        total: 695ms    remaining: 7.12s
89:     learn: 0.2990282        total: 701ms    remaining: 7.09s
90:     learn: 0.2987097        total: 707ms    remaining: 7.07s
91:     learn: 0.2984794        total: 716ms    remaining: 7.06s
92:     learn: 0.2981086        total: 722ms    remaining: 7.04s
93:     learn: 0.2978772        total: 729ms    remaining: 7.03s
94:     learn: 0.2975576        total: 735ms    remaining: 7s
95:     learn: 0.2972347        total: 742ms    remaining: 6.99s
96:     learn: 0.2969863        total: 749ms    remaining: 6.97s
97:     learn: 0.2966607        total: 756ms    remaining: 6.96s
98:     learn: 0.2963559        total: 762ms    remaining: 6.94s
99:     learn: 0.2958982        total: 772ms    remaining: 6.95s
100:    learn: 0.2956104        total: 780ms    remaining: 6.94s
101:    learn: 0.2953080        total: 789ms    remaining: 6.95s
102:    learn: 0.2948782        total: 795ms    remaining: 6.93s
103:    learn: 0.2945869        total: 805ms    remaining: 6.93s
104:    learn: 0.2941477        total: 812ms    remaining: 6.92s
105:    learn: 0.2938808        total: 820ms    remaining: 6.92s
106:    learn: 0.2936097        total: 827ms    remaining: 6.9s
107:    learn: 0.2933482        total: 834ms    remaining: 6.89s
108:    learn: 0.2930932        total: 841ms    remaining: 6.87s
109:    learn: 0.2925500        total: 848ms    remaining: 6.86s
110:    learn: 0.2924003        total: 855ms    remaining: 6.84s
```

```
111:    learn: 0.2919587        total: 862ms    remaining: 6.83s
112:    learn: 0.2918404        total: 869ms    remaining: 6.82s
113:    learn: 0.2914256        total: 876ms    remaining: 6.8s
114:    learn: 0.2911982        total: 883ms    remaining: 6.79s
115:    learn: 0.2906196        total: 889ms    remaining: 6.78s
116:    learn: 0.2902632        total: 896ms    remaining: 6.76s
117:    learn: 0.2897877        total: 904ms    remaining: 6.75s
118:    learn: 0.2895782        total: 911ms    remaining: 6.74s
119:    learn: 0.2893189        total: 918ms    remaining: 6.73s
120:    learn: 0.2889450        total: 925ms    remaining: 6.72s
121:    learn: 0.2885059        total: 932ms    remaining: 6.71s
122:    learn: 0.2882340        total: 938ms    remaining: 6.69s
123:    learn: 0.2877928        total: 945ms    remaining: 6.68s
124:    learn: 0.2875885        total: 952ms    remaining: 6.66s
125:    learn: 0.2874480        total: 958ms    remaining: 6.65s
126:    learn: 0.2871405        total: 965ms    remaining: 6.63s
127:    learn: 0.2868500        total: 974ms    remaining: 6.63s
128:    learn: 0.2865663        total: 981ms    remaining: 6.62s
129:    learn: 0.2863615        total: 987ms    remaining: 6.6s
130:    learn: 0.2861019        total: 997ms    remaining: 6.61s
131:    learn: 0.2859269        total: 1s       remaining: 6.6s
132:    learn: 0.2857520        total: 1.01s    remaining: 6.58s
133:    learn: 0.2854524        total: 1.02s    remaining: 6.57s
134:    learn: 0.2853199        total: 1.02s    remaining: 6.56s
135:    learn: 0.2850572        total: 1.03s    remaining: 6.54s
136:    learn: 0.2848730        total: 1.04s    remaining: 6.53s
137:    learn: 0.2846787        total: 1.04s    remaining: 6.51s
138:    learn: 0.2844674        total: 1.05s    remaining: 6.5s
139:    learn: 0.2841928        total: 1.06s    remaining: 6.49s
140:    learn: 0.2839001        total: 1.06s    remaining: 6.47s
141:    learn: 0.2835915        total: 1.07s    remaining: 6.46s
142:    learn: 0.2834072        total: 1.07s    remaining: 6.44s
143:    learn: 0.2831629        total: 1.08s    remaining: 6.43s
144:    learn: 0.2830300        total: 1.09s    remaining: 6.42s
145:    learn: 0.2828072        total: 1.09s    remaining: 6.4s
146:    learn: 0.2825957        total: 1.1s     remaining: 6.39s
147:    learn: 0.2823629        total: 1.11s    remaining: 6.38s
148:    learn: 0.2821761        total: 1.11s    remaining: 6.37s
149:    learn: 0.2819308        total: 1.12s    remaining: 6.35s
150:    learn: 0.2817125        total: 1.13s    remaining: 6.34s
151:    learn: 0.2815271        total: 1.13s    remaining: 6.33s
152:    learn: 0.2813567        total: 1.14s    remaining: 6.31s
153:    learn: 0.2810586        total: 1.15s    remaining: 6.3s
154:    learn: 0.2809616        total: 1.15s    remaining: 6.29s
155:    learn: 0.2806671        total: 1.16s    remaining: 6.28s
156:    learn: 0.2804890        total: 1.17s    remaining: 6.27s
157:    learn: 0.2802647        total: 1.18s    remaining: 6.26s
158:    learn: 0.2801127        total: 1.18s    remaining: 6.25s
```

```
159:    learn: 0.2799527        total: 1.19s    remaining: 6.24s
160:    learn: 0.2797509        total: 1.2s     remaining: 6.23s
161:    learn: 0.2794940        total: 1.2s     remaining: 6.22s
162:    learn: 0.2793055        total: 1.21s    remaining: 6.21s
163:    learn: 0.2791484        total: 1.22s    remaining: 6.2s
164:    learn: 0.2788063        total: 1.22s    remaining: 6.18s
165:    learn: 0.2786260        total: 1.23s    remaining: 6.17s
166:    learn: 0.2784517        total: 1.23s    remaining: 6.16s
167:    learn: 0.2783023        total: 1.24s    remaining: 6.14s
168:    learn: 0.2779348        total: 1.25s    remaining: 6.13s
169:    learn: 0.2777710        total: 1.25s    remaining: 6.12s
170:    learn: 0.2776180        total: 1.26s    remaining: 6.11s
171:    learn: 0.2773704        total: 1.27s    remaining: 6.1s
172:    learn: 0.2771728        total: 1.27s    remaining: 6.1s
173:    learn: 0.2770214        total: 1.28s    remaining: 6.09s
174:    learn: 0.2768671        total: 1.29s    remaining: 6.08s
175:    learn: 0.2766528        total: 1.29s    remaining: 6.07s
176:    learn: 0.2765488        total: 1.3s     remaining: 6.05s
177:    learn: 0.2764767        total: 1.31s    remaining: 6.04s
178:    learn: 0.2763391        total: 1.31s    remaining: 6.03s
179:    learn: 0.2761439        total: 1.32s    remaining: 6.02s
180:    learn: 0.2760841        total: 1.33s    remaining: 6.01s
181:    learn: 0.2759772        total: 1.33s    remaining: 6s
182:    learn: 0.2759107        total: 1.34s    remaining: 5.99s
183:    learn: 0.2756863        total: 1.35s    remaining: 5.98s
184:    learn: 0.2755069        total: 1.35s    remaining: 5.97s
185:    learn: 0.2753581        total: 1.36s    remaining: 5.96s
186:    learn: 0.2751378        total: 1.37s    remaining: 5.96s
187:    learn: 0.2749145        total: 1.38s    remaining: 5.95s
188:    learn: 0.2747698        total: 1.38s    remaining: 5.94s
189:    learn: 0.2745010        total: 1.39s    remaining: 5.93s
190:    learn: 0.2741518        total: 1.4s     remaining: 5.92s
191:    learn: 0.2740163        total: 1.41s    remaining: 5.91s
192:    learn: 0.2738899        total: 1.41s    remaining: 5.9s
193:    learn: 0.2737032        total: 1.42s    remaining: 5.89s
194:    learn: 0.2735783        total: 1.42s    remaining: 5.88s
195:    learn: 0.2734757        total: 1.43s    remaining: 5.87s
196:    learn: 0.2732287        total: 1.44s    remaining: 5.86s
197:    learn: 0.2731049        total: 1.44s    remaining: 5.85s
198:    learn: 0.2727692        total: 1.45s    remaining: 5.84s
199:    learn: 0.2726049        total: 1.46s    remaining: 5.83s
200:    learn: 0.2724737        total: 1.46s    remaining: 5.82s
201:    learn: 0.2722946        total: 1.47s    remaining: 5.81s
202:    learn: 0.2720950        total: 1.48s    remaining: 5.8s
203:    learn: 0.2719205        total: 1.48s    remaining: 5.79s
204:    learn: 0.2717669        total: 1.49s    remaining: 5.78s
205:    learn: 0.2716000        total: 1.5s     remaining: 5.78s
206:    learn: 0.2713129        total: 1.5s     remaining: 5.76s
```

```
207:    learn: 0.2711515        total: 1.51s    remaining: 5.75s
208:    learn: 0.2709708        total: 1.52s    remaining: 5.74s
209:    learn: 0.2707511        total: 1.52s    remaining: 5.73s
210:    learn: 0.2705600        total: 1.53s    remaining: 5.72s
211:    learn: 0.2703129        total: 1.54s    remaining: 5.71s
212:    learn: 0.2702358        total: 1.54s    remaining: 5.7s
213:    learn: 0.2699204        total: 1.55s    remaining: 5.7s
214:    learn: 0.2697254        total: 1.56s    remaining: 5.69s
215:    learn: 0.2695640        total: 1.56s    remaining: 5.68s
216:    learn: 0.2693328        total: 1.57s    remaining: 5.68s
217:    learn: 0.2691509        total: 1.58s    remaining: 5.67s
218:    learn: 0.2690504        total: 1.59s    remaining: 5.66s
219:    learn: 0.2687297        total: 1.59s    remaining: 5.65s
220:    learn: 0.2685505        total: 1.6s     remaining: 5.64s
221:    learn: 0.2684626        total: 1.61s    remaining: 5.63s
222:    learn: 0.2683507        total: 1.61s    remaining: 5.62s
223:    learn: 0.2681310        total: 1.62s    remaining: 5.61s
224:    learn: 0.2679856        total: 1.63s    remaining: 5.6s
225:    learn: 0.2678327        total: 1.63s    remaining: 5.59s
226:    learn: 0.2677328        total: 1.64s    remaining: 5.58s
227:    learn: 0.2675679        total: 1.65s    remaining: 5.57s
228:    learn: 0.2675089        total: 1.65s    remaining: 5.56s
229:    learn: 0.2673398        total: 1.66s    remaining: 5.55s
230:    learn: 0.2672542        total: 1.67s    remaining: 5.54s
231:    learn: 0.2670798        total: 1.67s    remaining: 5.53s
232:    learn: 0.2670172        total: 1.68s    remaining: 5.52s
233:    learn: 0.2668790        total: 1.68s    remaining: 5.51s
234:    learn: 0.2666330        total: 1.69s    remaining: 5.5s
235:    learn: 0.2665352        total: 1.7s     remaining: 5.5s
236:    learn: 0.2664430        total: 1.7s     remaining: 5.49s
237:    learn: 0.2663074        total: 1.71s    remaining: 5.47s
238:    learn: 0.2661058        total: 1.72s    remaining: 5.47s
239:    learn: 0.2659102        total: 1.72s    remaining: 5.46s
240:    learn: 0.2657153        total: 1.73s    remaining: 5.45s
241:    learn: 0.2656345        total: 1.74s    remaining: 5.44s
242:    learn: 0.2655788        total: 1.74s    remaining: 5.43s
243:    learn: 0.2654141        total: 1.75s    remaining: 5.43s
244:    learn: 0.2652077        total: 1.76s    remaining: 5.43s
245:    learn: 0.2651237        total: 1.77s    remaining: 5.42s
246:    learn: 0.2650135        total: 1.78s    remaining: 5.42s
247:    learn: 0.2648546        total: 1.78s    remaining: 5.41s
248:    learn: 0.2647120        total: 1.79s    remaining: 5.4s
249:    learn: 0.2646321        total: 1.8s     remaining: 5.39s
250:    learn: 0.2645084        total: 1.8s     remaining: 5.38s
251:    learn: 0.2642439        total: 1.81s    remaining: 5.38s
252:    learn: 0.2641141        total: 1.82s    remaining: 5.37s
253:    learn: 0.2639512        total: 1.82s    remaining: 5.36s
254:    learn: 0.2638013        total: 1.83s    remaining: 5.35s
```

```
255:    learn: 0.2636394        total: 1.84s    remaining: 5.34s
256:    learn: 0.2635295        total: 1.85s    remaining: 5.34s
257:    learn: 0.2634381        total: 1.85s    remaining: 5.33s
258:    learn: 0.2632915        total: 1.86s    remaining: 5.32s
259:    learn: 0.2631816        total: 1.87s    remaining: 5.31s
260:    learn: 0.2630627        total: 1.87s    remaining: 5.31s
261:    learn: 0.2629332        total: 1.88s    remaining: 5.3s
262:    learn: 0.2627458        total: 1.89s    remaining: 5.29s
263:    learn: 0.2625885        total: 1.89s    remaining: 5.28s
264:    learn: 0.2623263        total: 1.9s     remaining: 5.28s
265:    learn: 0.2621748        total: 1.91s    remaining: 5.27s
266:    learn: 0.2620294        total: 1.92s    remaining: 5.26s
267:    learn: 0.2618224        total: 1.92s    remaining: 5.25s
268:    learn: 0.2617166        total: 1.93s    remaining: 5.24s
269:    learn: 0.2615084        total: 1.94s    remaining: 5.24s
270:    learn: 0.2613742        total: 1.94s    remaining: 5.23s
271:    learn: 0.2612696        total: 1.95s    remaining: 5.23s
272:    learn: 0.2612063        total: 1.96s    remaining: 5.22s
273:    learn: 0.2610977        total: 1.97s    remaining: 5.21s
274:    learn: 0.2610502        total: 1.97s    remaining: 5.2s
275:    learn: 0.2608756        total: 1.98s    remaining: 5.19s
276:    learn: 0.2607833        total: 1.99s    remaining: 5.18s
277:    learn: 0.2605631        total: 1.99s    remaining: 5.17s
278:    learn: 0.2604485        total: 2s       remaining: 5.17s
279:    learn: 0.2602771        total: 2s       remaining: 5.16s
280:    learn: 0.2601206        total: 2.01s    remaining: 5.15s
281:    learn: 0.2599849        total: 2.02s    remaining: 5.14s
282:    learn: 0.2598402        total: 2.02s    remaining: 5.13s
283:    learn: 0.2596791        total: 2.03s    remaining: 5.12s
284:    learn: 0.2595510        total: 2.04s    remaining: 5.12s
285:    learn: 0.2594367        total: 2.05s    remaining: 5.11s
286:    learn: 0.2592892        total: 2.05s    remaining: 5.1s
287:    learn: 0.2590495        total: 2.06s    remaining: 5.09s
288:    learn: 0.2589590        total: 2.07s    remaining: 5.08s
289:    learn: 0.2588324        total: 2.07s    remaining: 5.07s
290:    learn: 0.2587036        total: 2.08s    remaining: 5.07s
291:    learn: 0.2586173        total: 2.09s    remaining: 5.06s
292:    learn: 0.2584292        total: 2.09s    remaining: 5.05s
293:    learn: 0.2583495        total: 2.1s     remaining: 5.04s
294:    learn: 0.2582624        total: 2.11s    remaining: 5.03s
295:    learn: 0.2581133        total: 2.11s    remaining: 5.03s
296:    learn: 0.2579693        total: 2.12s    remaining: 5.02s
297:    learn: 0.2577717        total: 2.13s    remaining: 5.01s
298:    learn: 0.2575583        total: 2.14s    remaining: 5.01s
299:    learn: 0.2574493        total: 2.14s    remaining: 5s
300:    learn: 0.2573396        total: 2.15s    remaining: 5s
301:    learn: 0.2572854        total: 2.16s    remaining: 4.99s
302:    learn: 0.2571665        total: 2.17s    remaining: 4.99s
```

```
303:    learn: 0.2569914        total: 2.17s    remaining: 4.98s
304:    learn: 0.2569582        total: 2.18s    remaining: 4.97s
305:    learn: 0.2569127        total: 2.19s    remaining: 4.96s
306:    learn: 0.2568093        total: 2.19s    remaining: 4.95s
307:    learn: 0.2567142        total: 2.2s     remaining: 4.94s
308:    learn: 0.2566495        total: 2.21s    remaining: 4.94s
309:    learn: 0.2565959        total: 2.21s    remaining: 4.93s
310:    learn: 0.2564857        total: 2.22s    remaining: 4.92s
311:    learn: 0.2563354        total: 2.23s    remaining: 4.92s
312:    learn: 0.2562207        total: 2.24s    remaining: 4.91s
313:    learn: 0.2560835        total: 2.25s    remaining: 4.92s
314:    learn: 0.2559922        total: 2.26s    remaining: 4.91s
315:    learn: 0.2558174        total: 2.27s    remaining: 4.91s
316:    learn: 0.2556417        total: 2.27s    remaining: 4.9s
317:    learn: 0.2554609        total: 2.28s    remaining: 4.9s
318:    learn: 0.2553370        total: 2.29s    remaining: 4.89s
319:    learn: 0.2552525        total: 2.3s     remaining: 4.89s
320:    learn: 0.2551663        total: 2.31s    remaining: 4.89s
321:    learn: 0.2551063        total: 2.32s    remaining: 4.88s
322:    learn: 0.2550123        total: 2.33s    remaining: 4.88s
323:    learn: 0.2548105        total: 2.33s    remaining: 4.87s
324:    learn: 0.2546762        total: 2.34s    remaining: 4.87s
325:    learn: 0.2545700        total: 2.35s    remaining: 4.86s
326:    learn: 0.2545043        total: 2.37s    remaining: 4.87s
327:    learn: 0.2544325        total: 2.38s    remaining: 4.87s
328:    learn: 0.2543470        total: 2.38s    remaining: 4.86s
329:    learn: 0.2540798        total: 2.4s     remaining: 4.87s
330:    learn: 0.2539366        total: 2.41s    remaining: 4.87s
331:    learn: 0.2537998        total: 2.42s    remaining: 4.86s
332:    learn: 0.2536280        total: 2.43s    remaining: 4.86s
333:    learn: 0.2535319        total: 2.43s    remaining: 4.85s
334:    learn: 0.2534334        total: 2.44s    remaining: 4.85s
335:    learn: 0.2533541        total: 2.45s    remaining: 4.84s
336:    learn: 0.2532279        total: 2.46s    remaining: 4.84s
337:    learn: 0.2531536        total: 2.47s    remaining: 4.84s
338:    learn: 0.2530635        total: 2.48s    remaining: 4.83s
339:    learn: 0.2528590        total: 2.49s    remaining: 4.83s
340:    learn: 0.2526574        total: 2.5s     remaining: 4.82s
341:    learn: 0.2525343        total: 2.5s     remaining: 4.82s
342:    learn: 0.2524801        total: 2.51s    remaining: 4.81s
343:    learn: 0.2522730        total: 2.52s    remaining: 4.81s
344:    learn: 0.2521947        total: 2.54s    remaining: 4.82s
345:    learn: 0.2520102        total: 2.56s    remaining: 4.85s
346:    learn: 0.2517677        total: 2.58s    remaining: 4.85s
347:    learn: 0.2516486        total: 2.58s    remaining: 4.84s
348:    learn: 0.2515390        total: 2.59s    remaining: 4.84s
349:    learn: 0.2513395        total: 2.6s     remaining: 4.83s
350:    learn: 0.2512586        total: 2.61s    remaining: 4.82s
```

```
351:    learn: 0.2511912        total: 2.62s    remaining: 4.82s
352:    learn: 0.2511189        total: 2.62s    remaining: 4.81s
353:    learn: 0.2510233        total: 2.63s    remaining: 4.81s
354:    learn: 0.2509355        total: 2.64s    remaining: 4.8s
355:    learn: 0.2508595        total: 2.65s    remaining: 4.8s
356:    learn: 0.2507577        total: 2.66s    remaining: 4.79s
357:    learn: 0.2506796        total: 2.67s    remaining: 4.79s
358:    learn: 0.2505693        total: 2.68s    remaining: 4.78s
359:    learn: 0.2504043        total: 2.69s    remaining: 4.78s
360:    learn: 0.2503271        total: 2.69s    remaining: 4.77s
361:    learn: 0.2502966        total: 2.71s    remaining: 4.77s
362:    learn: 0.2501693        total: 2.72s    remaining: 4.77s
363:    learn: 0.2500923        total: 2.73s    remaining: 4.76s
364:    learn: 0.2499820        total: 2.73s    remaining: 4.76s
365:    learn: 0.2498756        total: 2.74s    remaining: 4.75s
366:    learn: 0.2498119        total: 2.75s    remaining: 4.75s
367:    learn: 0.2496107        total: 2.76s    remaining: 4.74s
368:    learn: 0.2494825        total: 2.77s    remaining: 4.74s
369:    learn: 0.2494190        total: 2.78s    remaining: 4.73s
370:    learn: 0.2492703        total: 2.79s    remaining: 4.73s
371:    learn: 0.2492111        total: 2.8s     remaining: 4.72s
372:    learn: 0.2490851        total: 2.81s    remaining: 4.72s
373:    learn: 0.2489012        total: 2.81s    remaining: 4.71s
374:    learn: 0.2486909        total: 2.82s    remaining: 4.71s
375:    learn: 0.2484517        total: 2.83s    remaining: 4.7s
376:    learn: 0.2483050        total: 2.84s    remaining: 4.69s
377:    learn: 0.2481726        total: 2.85s    remaining: 4.69s
378:    learn: 0.2480810        total: 2.86s    remaining: 4.69s
379:    learn: 0.2480074        total: 2.87s    remaining: 4.68s
380:    learn: 0.2478954        total: 2.88s    remaining: 4.67s
381:    learn: 0.2477283        total: 2.88s    remaining: 4.67s
382:    learn: 0.2476136        total: 2.89s    remaining: 4.66s
383:    learn: 0.2475434        total: 2.9s     remaining: 4.66s
384:    learn: 0.2474423        total: 2.91s    remaining: 4.65s
385:    learn: 0.2473654        total: 2.92s    remaining: 4.64s
386:    learn: 0.2473118        total: 2.93s    remaining: 4.64s
387:    learn: 0.2472375        total: 2.94s    remaining: 4.63s
388:    learn: 0.2471173        total: 2.94s    remaining: 4.63s
389:    learn: 0.2470254        total: 2.95s    remaining: 4.62s
390:    learn: 0.2469452        total: 2.96s    remaining: 4.61s
391:    learn: 0.2468535        total: 2.97s    remaining: 4.6s
392:    learn: 0.2467912        total: 2.97s    remaining: 4.59s
393:    learn: 0.2466927        total: 2.98s    remaining: 4.59s
394:    learn: 0.2465479        total: 2.99s    remaining: 4.58s
395:    learn: 0.2463352        total: 3s       remaining: 4.57s
396:    learn: 0.2462179        total: 3.01s    remaining: 4.57s
397:    learn: 0.2460917        total: 3.01s    remaining: 4.56s
398:    learn: 0.2459411        total: 3.02s    remaining: 4.55s
```

```
399:    learn: 0.2458841        total: 3.03s    remaining: 4.54s
400:    learn: 0.2458081        total: 3.04s    remaining: 4.54s
401:    learn: 0.2457452        total: 3.04s    remaining: 4.53s
402:    learn: 0.2456750        total: 3.05s    remaining: 4.52s
403:    learn: 0.2455607        total: 3.06s    remaining: 4.51s
404:    learn: 0.2454089        total: 3.07s    remaining: 4.5s
405:    learn: 0.2453208        total: 3.07s    remaining: 4.5s
406:    learn: 0.2452392        total: 3.08s    remaining: 4.49s
407:    learn: 0.2451543        total: 3.09s    remaining: 4.48s
408:    learn: 0.2450538        total: 3.09s    remaining: 4.47s
409:    learn: 0.2449136        total: 3.1s     remaining: 4.46s
410:    learn: 0.2448178        total: 3.11s    remaining: 4.46s
411:    learn: 0.2446322        total: 3.12s    remaining: 4.45s
412:    learn: 0.2445567        total: 3.12s    remaining: 4.44s
413:    learn: 0.2444239        total: 3.13s    remaining: 4.43s
414:    learn: 0.2443230        total: 3.14s    remaining: 4.42s
415:    learn: 0.2442513        total: 3.14s    remaining: 4.41s
416:    learn: 0.2441210        total: 3.15s    remaining: 4.41s
417:    learn: 0.2440332        total: 3.16s    remaining: 4.4s
418:    learn: 0.2439961        total: 3.16s    remaining: 4.39s
419:    learn: 0.2437823        total: 3.17s    remaining: 4.38s
420:    learn: 0.2436803        total: 3.18s    remaining: 4.37s
421:    learn: 0.2435702        total: 3.18s    remaining: 4.36s
422:    learn: 0.2435219        total: 3.19s    remaining: 4.35s
423:    learn: 0.2434607        total: 3.2s     remaining: 4.34s
424:    learn: 0.2433484        total: 3.2s     remaining: 4.33s
425:    learn: 0.2432564        total: 3.21s    remaining: 4.33s
426:    learn: 0.2431710        total: 3.22s    remaining: 4.32s
427:    learn: 0.2431107        total: 3.22s    remaining: 4.31s
428:    learn: 0.2430586        total: 3.23s    remaining: 4.3s
429:    learn: 0.2430176        total: 3.24s    remaining: 4.29s
430:    learn: 0.2428598        total: 3.24s    remaining: 4.28s
431:    learn: 0.2427883        total: 3.25s    remaining: 4.28s
432:    learn: 0.2427088        total: 3.26s    remaining: 4.27s
433:    learn: 0.2425881        total: 3.27s    remaining: 4.26s
434:    learn: 0.2424907        total: 3.27s    remaining: 4.25s
435:    learn: 0.2424262        total: 3.28s    remaining: 4.25s
436:    learn: 0.2422920        total: 3.29s    remaining: 4.24s
437:    learn: 0.2421690        total: 3.3s     remaining: 4.24s
438:    learn: 0.2421377        total: 3.31s    remaining: 4.23s
439:    learn: 0.2420341        total: 3.32s    remaining: 4.22s
440:    learn: 0.2419517        total: 3.33s    remaining: 4.21s
441:    learn: 0.2418692        total: 3.33s    remaining: 4.21s
442:    learn: 0.2417533        total: 3.34s    remaining: 4.2s
443:    learn: 0.2416754        total: 3.35s    remaining: 4.2s
444:    learn: 0.2415760        total: 3.36s    remaining: 4.19s
445:    learn: 0.2414568        total: 3.37s    remaining: 4.19s
446:    learn: 0.2413805        total: 3.38s    remaining: 4.18s
```

```
447:    learn: 0.2412710        total: 3.39s    remaining: 4.17s
448:    learn: 0.2412166        total: 3.4s     remaining: 4.17s
449:    learn: 0.2411769        total: 3.4s     remaining: 4.16s
450:    learn: 0.2411315        total: 3.41s    remaining: 4.15s
451:    learn: 0.2409872        total: 3.42s    remaining: 4.14s
452:    learn: 0.2408737        total: 3.43s    remaining: 4.14s
453:    learn: 0.2408011        total: 3.44s    remaining: 4.13s
454:    learn: 0.2407050        total: 3.45s    remaining: 4.13s
455:    learn: 0.2406112        total: 3.46s    remaining: 4.13s
456:    learn: 0.2405135        total: 3.47s    remaining: 4.12s
457:    learn: 0.2403933        total: 3.48s    remaining: 4.12s
458:    learn: 0.2403263        total: 3.49s    remaining: 4.11s
459:    learn: 0.2402245        total: 3.5s     remaining: 4.11s
460:    learn: 0.2401176        total: 3.51s    remaining: 4.1s
461:    learn: 0.2400628        total: 3.51s    remaining: 4.09s
462:    learn: 0.2399675        total: 3.52s    remaining: 4.08s
463:    learn: 0.2398718        total: 3.53s    remaining: 4.08s
464:    learn: 0.2398111        total: 3.54s    remaining: 4.07s
465:    learn: 0.2397225        total: 3.54s    remaining: 4.06s
466:    learn: 0.2396186        total: 3.55s    remaining: 4.05s
467:    learn: 0.2395598        total: 3.56s    remaining: 4.05s
468:    learn: 0.2394441        total: 3.57s    remaining: 4.04s
469:    learn: 0.2393482        total: 3.57s    remaining: 4.03s
470:    learn: 0.2392658        total: 3.58s    remaining: 4.02s
471:    learn: 0.2392279        total: 3.59s    remaining: 4.01s
472:    learn: 0.2391144        total: 3.6s     remaining: 4s
473:    learn: 0.2390368        total: 3.6s     remaining: 4s
474:    learn: 0.2389403        total: 3.61s    remaining: 3.99s
475:    learn: 0.2388955        total: 3.61s    remaining: 3.98s
476:    learn: 0.2388040        total: 3.62s    remaining: 3.97s
477:    learn: 0.2386951        total: 3.63s    remaining: 3.96s
478:    learn: 0.2385888        total: 3.63s    remaining: 3.95s
479:    learn: 0.2385213        total: 3.64s    remaining: 3.94s
480:    learn: 0.2384362        total: 3.65s    remaining: 3.94s
481:    learn: 0.2383377        total: 3.66s    remaining: 3.93s
482:    learn: 0.2382454        total: 3.66s    remaining: 3.92s
483:    learn: 0.2381806        total: 3.67s    remaining: 3.91s
484:    learn: 0.2381234        total: 3.68s    remaining: 3.9s
485:    learn: 0.2380210        total: 3.68s    remaining: 3.9s
486:    learn: 0.2379709        total: 3.69s    remaining: 3.89s
487:    learn: 0.2378573        total: 3.7s     remaining: 3.88s
488:    learn: 0.2377750        total: 3.71s    remaining: 3.88s
489:    learn: 0.2376907        total: 3.72s    remaining: 3.87s
490:    learn: 0.2376052        total: 3.73s    remaining: 3.86s
491:    learn: 0.2375363        total: 3.73s    remaining: 3.85s
492:    learn: 0.2374885        total: 3.74s    remaining: 3.85s
493:    learn: 0.2373798        total: 3.75s    remaining: 3.84s
494:    learn: 0.2372918        total: 3.75s    remaining: 3.83s
```

```
495:    learn: 0.2372014        total: 3.76s    remaining: 3.82s
496:    learn: 0.2371190        total: 3.77s    remaining: 3.81s
497:    learn: 0.2369891        total: 3.78s    remaining: 3.81s
498:    learn: 0.2369449        total: 3.78s    remaining: 3.8s
499:    learn: 0.2368201        total: 3.79s    remaining: 3.79s
500:    learn: 0.2366439        total: 3.79s    remaining: 3.78s
501:    learn: 0.2365447        total: 3.8s     remaining: 3.77s
502:    learn: 0.2365085        total: 3.81s    remaining: 3.76s
503:    learn: 0.2363954        total: 3.81s    remaining: 3.75s
504:    learn: 0.2363089        total: 3.82s    remaining: 3.75s
505:    learn: 0.2362400        total: 3.83s    remaining: 3.74s
506:    learn: 0.2362133        total: 3.83s    remaining: 3.73s
507:    learn: 0.2361606        total: 3.84s    remaining: 3.72s
508:    learn: 0.2359589        total: 3.85s    remaining: 3.71s
509:    learn: 0.2358921        total: 3.85s    remaining: 3.7s
510:    learn: 0.2357758        total: 3.86s    remaining: 3.69s
511:    learn: 0.2356624        total: 3.87s    remaining: 3.69s
512:    learn: 0.2355938        total: 3.87s    remaining: 3.68s
513:    learn: 0.2354221        total: 3.88s    remaining: 3.67s
514:    learn: 0.2353705        total: 3.89s    remaining: 3.66s
515:    learn: 0.2352950        total: 3.89s    remaining: 3.65s
516:    learn: 0.2351944        total: 3.9s     remaining: 3.65s
517:    learn: 0.2350758        total: 3.91s    remaining: 3.64s
518:    learn: 0.2349332        total: 3.92s    remaining: 3.63s
519:    learn: 0.2349104        total: 3.92s    remaining: 3.62s
520:    learn: 0.2348828        total: 3.93s    remaining: 3.61s
521:    learn: 0.2348005        total: 3.94s    remaining: 3.6s
522:    learn: 0.2347139        total: 3.94s    remaining: 3.6s
523:    learn: 0.2345907        total: 3.95s    remaining: 3.59s
524:    learn: 0.2343715        total: 3.96s    remaining: 3.58s
525:    learn: 0.2342514        total: 3.96s    remaining: 3.57s
526:    learn: 0.2341866        total: 3.97s    remaining: 3.56s
527:    learn: 0.2340830        total: 3.98s    remaining: 3.55s
528:    learn: 0.2340194        total: 3.98s    remaining: 3.54s
529:    learn: 0.2338708        total: 3.99s    remaining: 3.54s
530:    learn: 0.2337823        total: 4s       remaining: 3.53s
531:    learn: 0.2336724        total: 4s       remaining: 3.52s
532:    learn: 0.2336355        total: 4.01s    remaining: 3.51s
533:    learn: 0.2334123        total: 4.02s    remaining: 3.51s
534:    learn: 0.2333637        total: 4.02s    remaining: 3.5s
535:    learn: 0.2332831        total: 4.03s    remaining: 3.49s
536:    learn: 0.2332152        total: 4.04s    remaining: 3.48s
537:    learn: 0.2330987        total: 4.04s    remaining: 3.47s
538:    learn: 0.2330492        total: 4.05s    remaining: 3.46s
539:    learn: 0.2330067        total: 4.06s    remaining: 3.46s
540:    learn: 0.2329408        total: 4.06s    remaining: 3.45s
541:    learn: 0.2328675        total: 4.07s    remaining: 3.44s
542:    learn: 0.2328051        total: 4.08s    remaining: 3.43s
```

```
543:    learn: 0.2327398        total: 4.08s    remaining: 3.42s
544:    learn: 0.2326481        total: 4.09s    remaining: 3.41s
545:    learn: 0.2325698        total: 4.1s     remaining: 3.41s
546:    learn: 0.2325409        total: 4.1s     remaining: 3.4s
547:    learn: 0.2324602        total: 4.11s    remaining: 3.39s
548:    learn: 0.2323701        total: 4.12s    remaining: 3.38s
549:    learn: 0.2322487        total: 4.13s    remaining: 3.38s
550:    learn: 0.2321561        total: 4.13s    remaining: 3.37s
551:    learn: 0.2320686        total: 4.14s    remaining: 3.36s
552:    learn: 0.2319534        total: 4.14s    remaining: 3.35s
553:    learn: 0.2318556        total: 4.15s    remaining: 3.34s
554:    learn: 0.2317785        total: 4.16s    remaining: 3.33s
555:    learn: 0.2317383        total: 4.16s    remaining: 3.32s
556:    learn: 0.2316309        total: 4.17s    remaining: 3.32s
557:    learn: 0.2315543        total: 4.18s    remaining: 3.31s
558:    learn: 0.2314691        total: 4.18s    remaining: 3.3s
559:    learn: 0.2313966        total: 4.19s    remaining: 3.29s
560:    learn: 0.2313101        total: 4.2s     remaining: 3.28s
561:    learn: 0.2312846        total: 4.2s     remaining: 3.27s
562:    learn: 0.2312108        total: 4.21s    remaining: 3.27s
563:    learn: 0.2311382        total: 4.21s    remaining: 3.26s
564:    learn: 0.2310464        total: 4.22s    remaining: 3.25s
565:    learn: 0.2309615        total: 4.23s    remaining: 3.24s
566:    learn: 0.2308791        total: 4.23s    remaining: 3.23s
567:    learn: 0.2308162        total: 4.24s    remaining: 3.23s
568:    learn: 0.2307032        total: 4.25s    remaining: 3.22s
569:    learn: 0.2306071        total: 4.25s    remaining: 3.21s
570:    learn: 0.2305441        total: 4.26s    remaining: 3.2s
571:    learn: 0.2304864        total: 4.26s    remaining: 3.19s
572:    learn: 0.2304367        total: 4.27s    remaining: 3.18s
573:    learn: 0.2303616        total: 4.28s    remaining: 3.17s
574:    learn: 0.2303132        total: 4.29s    remaining: 3.17s
575:    learn: 0.2301102        total: 4.29s    remaining: 3.16s
576:    learn: 0.2300070        total: 4.3s     remaining: 3.15s
577:    learn: 0.2298970        total: 4.31s    remaining: 3.15s
578:    learn: 0.2297866        total: 4.32s    remaining: 3.14s
579:    learn: 0.2296978        total: 4.32s    remaining: 3.13s
580:    learn: 0.2296523        total: 4.33s    remaining: 3.12s
581:    learn: 0.2295582        total: 4.34s    remaining: 3.11s
582:    learn: 0.2294388        total: 4.34s    remaining: 3.11s
583:    learn: 0.2293839        total: 4.35s    remaining: 3.1s
584:    learn: 0.2293178        total: 4.35s    remaining: 3.09s
585:    learn: 0.2292084        total: 4.36s    remaining: 3.08s
586:    learn: 0.2291417        total: 4.37s    remaining: 3.07s
587:    learn: 0.2290473        total: 4.37s    remaining: 3.06s
588:    learn: 0.2290282        total: 4.38s    remaining: 3.06s
589:    learn: 0.2289542        total: 4.39s    remaining: 3.05s
590:    learn: 0.2288354        total: 4.39s    remaining: 3.04s
```

```
591:    learn: 0.2287734        total: 4.4s     remaining: 3.03s
592:    learn: 0.2286932        total: 4.41s    remaining: 3.02s
593:    learn: 0.2286191        total: 4.41s    remaining: 3.02s
594:    learn: 0.2285411        total: 4.42s    remaining: 3.01s
595:    learn: 0.2284742        total: 4.43s    remaining: 3s
596:    learn: 0.2283725        total: 4.43s    remaining: 2.99s
597:    learn: 0.2283152        total: 4.44s    remaining: 2.98s
598:    learn: 0.2281917        total: 4.44s    remaining: 2.98s
599:    learn: 0.2281476        total: 4.45s    remaining: 2.97s
600:    learn: 0.2280218        total: 4.46s    remaining: 2.96s
601:    learn: 0.2279444        total: 4.46s    remaining: 2.95s
602:    learn: 0.2278107        total: 4.47s    remaining: 2.94s
603:    learn: 0.2277435        total: 4.48s    remaining: 2.94s
604:    learn: 0.2276865        total: 4.49s    remaining: 2.93s
605:    learn: 0.2275265        total: 4.49s    remaining: 2.92s
606:    learn: 0.2273750        total: 4.5s     remaining: 2.91s
607:    learn: 0.2272734        total: 4.51s    remaining: 2.91s
608:    learn: 0.2271752        total: 4.51s    remaining: 2.9s
609:    learn: 0.2270513        total: 4.52s    remaining: 2.89s
610:    learn: 0.2269745        total: 4.53s    remaining: 2.88s
611:    learn: 0.2268952        total: 4.53s    remaining: 2.87s
612:    learn: 0.2268209        total: 4.54s    remaining: 2.87s
613:    learn: 0.2266685        total: 4.55s    remaining: 2.86s
614:    learn: 0.2266132        total: 4.55s    remaining: 2.85s
615:    learn: 0.2265536        total: 4.56s    remaining: 2.84s
616:    learn: 0.2264320        total: 4.57s    remaining: 2.83s
617:    learn: 0.2263561        total: 4.57s    remaining: 2.83s
618:    learn: 0.2262750        total: 4.58s    remaining: 2.82s
619:    learn: 0.2261826        total: 4.59s    remaining: 2.81s
620:    learn: 0.2260958        total: 4.59s    remaining: 2.8s
621:    learn: 0.2260096        total: 4.6s     remaining: 2.79s
622:    learn: 0.2259338        total: 4.61s    remaining: 2.79s
623:    learn: 0.2259203        total: 4.61s    remaining: 2.78s
624:    learn: 0.2258315        total: 4.62s    remaining: 2.77s
625:    learn: 0.2257777        total: 4.63s    remaining: 2.76s
626:    learn: 0.2255963        total: 4.63s    remaining: 2.76s
627:    learn: 0.2255530        total: 4.64s    remaining: 2.75s
628:    learn: 0.2254733        total: 4.65s    remaining: 2.74s
629:    learn: 0.2254179        total: 4.66s    remaining: 2.73s
630:    learn: 0.2252961        total: 4.66s    remaining: 2.73s
631:    learn: 0.2252181        total: 4.67s    remaining: 2.72s
632:    learn: 0.2251207        total: 4.68s    remaining: 2.71s
633:    learn: 0.2249987        total: 4.68s    remaining: 2.7s
634:    learn: 0.2249561        total: 4.69s    remaining: 2.7s
635:    learn: 0.2248620        total: 4.7s     remaining: 2.69s
636:    learn: 0.2248074        total: 4.71s    remaining: 2.68s
637:    learn: 0.2247124        total: 4.71s    remaining: 2.67s
638:    learn: 0.2246600        total: 4.72s    remaining: 2.67s
```

```
639:    learn: 0.2245868        total: 4.73s    remaining: 2.66s
640:    learn: 0.2245486        total: 4.73s    remaining: 2.65s
641:    learn: 0.2244430        total: 4.74s    remaining: 2.64s
642:    learn: 0.2244136        total: 4.75s    remaining: 2.63s
643:    learn: 0.2243012        total: 4.75s    remaining: 2.63s
644:    learn: 0.2242103        total: 4.76s    remaining: 2.62s
645:    learn: 0.2241624        total: 4.76s    remaining: 2.61s
646:    learn: 0.2240465        total: 4.77s    remaining: 2.6s
647:    learn: 0.2239896        total: 4.78s    remaining: 2.6s
648:    learn: 0.2239490        total: 4.78s    remaining: 2.59s
649:    learn: 0.2238632        total: 4.79s    remaining: 2.58s
650:    learn: 0.2238351        total: 4.8s     remaining: 2.57s
651:    learn: 0.2237155        total: 4.8s     remaining: 2.56s
652:    learn: 0.2236351        total: 4.81s    remaining: 2.56s
653:    learn: 0.2236086        total: 4.81s    remaining: 2.55s
654:    learn: 0.2235782        total: 4.82s    remaining: 2.54s
655:    learn: 0.2234977        total: 4.83s    remaining: 2.53s
656:    learn: 0.2234455        total: 4.83s    remaining: 2.52s
657:    learn: 0.2233456        total: 4.84s    remaining: 2.52s
658:    learn: 0.2232818        total: 4.85s    remaining: 2.51s
659:    learn: 0.2232522        total: 4.86s    remaining: 2.5s
660:    learn: 0.2231503        total: 4.86s    remaining: 2.49s
661:    learn: 0.2230821        total: 4.87s    remaining: 2.49s
662:    learn: 0.2230465        total: 4.88s    remaining: 2.48s
663:    learn: 0.2229884        total: 4.88s    remaining: 2.47s
664:    learn: 0.2229312        total: 4.89s    remaining: 2.46s
665:    learn: 0.2228304        total: 4.9s     remaining: 2.46s
666:    learn: 0.2227812        total: 4.9s     remaining: 2.45s
667:    learn: 0.2227297        total: 4.91s    remaining: 2.44s
668:    learn: 0.2226735        total: 4.92s    remaining: 2.43s
669:    learn: 0.2226116        total: 4.92s    remaining: 2.42s
670:    learn: 0.2225599        total: 4.93s    remaining: 2.42s
671:    learn: 0.2224786        total: 4.94s    remaining: 2.41s
672:    learn: 0.2224078        total: 4.94s    remaining: 2.4s
673:    learn: 0.2223445        total: 4.95s    remaining: 2.39s
674:    learn: 0.2222712        total: 4.96s    remaining: 2.39s
675:    learn: 0.2222367        total: 4.96s    remaining: 2.38s
676:    learn: 0.2221321        total: 4.97s    remaining: 2.37s
677:    learn: 0.2220190        total: 4.98s    remaining: 2.36s
678:    learn: 0.2219676        total: 4.98s    remaining: 2.35s
679:    learn: 0.2219169        total: 4.99s    remaining: 2.35s
680:    learn: 0.2218689        total: 5s       remaining: 2.34s
681:    learn: 0.2218370        total: 5s       remaining: 2.33s
682:    learn: 0.2217454        total: 5.01s    remaining: 2.32s
683:    learn: 0.2216599        total: 5.01s    remaining: 2.32s
684:    learn: 0.2215378        total: 5.02s    remaining: 2.31s
685:    learn: 0.2214376        total: 5.03s    remaining: 2.3s
686:    learn: 0.2213975        total: 5.03s    remaining: 2.29s
```

```
687:    learn: 0.2213292        total: 5.04s    remaining: 2.29s
688:    learn: 0.2212721        total: 5.05s    remaining: 2.28s
689:    learn: 0.2212110        total: 5.06s    remaining: 2.27s
690:    learn: 0.2211598        total: 5.06s    remaining: 2.26s
691:    learn: 0.2211142        total: 5.07s    remaining: 2.26s
692:    learn: 0.2210669        total: 5.08s    remaining: 2.25s
693:    learn: 0.2210069        total: 5.08s    remaining: 2.24s
694:    learn: 0.2209183        total: 5.09s    remaining: 2.23s
695:    learn: 0.2208963        total: 5.09s    remaining: 2.23s
696:    learn: 0.2208444        total: 5.1s     remaining: 2.22s
697:    learn: 0.2207980        total: 5.11s    remaining: 2.21s
698:    learn: 0.2206758        total: 5.11s    remaining: 2.2s
699:    learn: 0.2205903        total: 5.12s    remaining: 2.19s
700:    learn: 0.2205043        total: 5.13s    remaining: 2.19s
701:    learn: 0.2204194        total: 5.13s    remaining: 2.18s
702:    learn: 0.2203744        total: 5.14s    remaining: 2.17s
703:    learn: 0.2203193        total: 5.14s    remaining: 2.16s
704:    learn: 0.2202513        total: 5.15s    remaining: 2.15s
705:    learn: 0.2201954        total: 5.16s    remaining: 2.15s
706:    learn: 0.2201351        total: 5.17s    remaining: 2.14s
707:    learn: 0.2200494        total: 5.17s    remaining: 2.13s
708:    learn: 0.2199444        total: 5.18s    remaining: 2.13s
709:    learn: 0.2198646        total: 5.18s    remaining: 2.12s
710:    learn: 0.2197414        total: 5.19s    remaining: 2.11s
711:    learn: 0.2196565        total: 5.2s     remaining: 2.1s
712:    learn: 0.2196060        total: 5.21s    remaining: 2.1s
713:    learn: 0.2195434        total: 5.21s    remaining: 2.09s
714:    learn: 0.2194537        total: 5.22s    remaining: 2.08s
715:    learn: 0.2194222        total: 5.22s    remaining: 2.07s
716:    learn: 0.2193432        total: 5.23s    remaining: 2.06s
717:    learn: 0.2192443        total: 5.24s    remaining: 2.06s
718:    learn: 0.2191434        total: 5.25s    remaining: 2.05s
719:    learn: 0.2190962        total: 5.25s    remaining: 2.04s
720:    learn: 0.2189546        total: 5.26s    remaining: 2.04s
721:    learn: 0.2189082        total: 5.27s    remaining: 2.03s
722:    learn: 0.2188569        total: 5.28s    remaining: 2.02s
723:    learn: 0.2188171        total: 5.28s    remaining: 2.01s
724:    learn: 0.2187779        total: 5.29s    remaining: 2.01s
725:    learn: 0.2187466        total: 5.29s    remaining: 2s
726:    learn: 0.2186752        total: 5.3s     remaining: 1.99s
727:    learn: 0.2185675        total: 5.31s    remaining: 1.98s
728:    learn: 0.2185424        total: 5.31s    remaining: 1.98s
729:    learn: 0.2184930        total: 5.32s    remaining: 1.97s
730:    learn: 0.2184496        total: 5.33s    remaining: 1.96s
731:    learn: 0.2183827        total: 5.33s    remaining: 1.95s
732:    learn: 0.2183290        total: 5.34s    remaining: 1.95s
733:    learn: 0.2182483        total: 5.35s    remaining: 1.94s
734:    learn: 0.2182161        total: 5.35s    remaining: 1.93s
```

```
735:    learn: 0.2181494        total: 5.36s    remaining: 1.92s
736:    learn: 0.2180817        total: 5.37s    remaining: 1.91s
737:    learn: 0.2180049        total: 5.37s    remaining: 1.91s
738:    learn: 0.2179710        total: 5.38s    remaining: 1.9s
739:    learn: 0.2179012        total: 5.38s    remaining: 1.89s
740:    learn: 0.2178036        total: 5.39s    remaining: 1.88s
741:    learn: 0.2177439        total: 5.4s     remaining: 1.88s
742:    learn: 0.2176152        total: 5.4s     remaining: 1.87s
743:    learn: 0.2175377        total: 5.41s    remaining: 1.86s
744:    learn: 0.2174860        total: 5.42s    remaining: 1.85s
745:    learn: 0.2174450        total: 5.42s    remaining: 1.85s
746:    learn: 0.2173471        total: 5.43s    remaining: 1.84s
747:    learn: 0.2173141        total: 5.44s    remaining: 1.83s
748:    learn: 0.2172256        total: 5.45s    remaining: 1.83s
749:    learn: 0.2171288        total: 5.46s    remaining: 1.82s
750:    learn: 0.2170987        total: 5.46s    remaining: 1.81s
751:    learn: 0.2170493        total: 5.47s    remaining: 1.8s
752:    learn: 0.2169640        total: 5.47s    remaining: 1.79s
753:    learn: 0.2168961        total: 5.48s    remaining: 1.79s
754:    learn: 0.2168489        total: 5.49s    remaining: 1.78s
755:    learn: 0.2167777        total: 5.49s    remaining: 1.77s
756:    learn: 0.2166660        total: 5.5s     remaining: 1.76s
757:    learn: 0.2165891        total: 5.51s    remaining: 1.76s
758:    learn: 0.2164904        total: 5.51s    remaining: 1.75s
759:    learn: 0.2163905        total: 5.52s    remaining: 1.74s
760:    learn: 0.2163299        total: 5.53s    remaining: 1.74s
761:    learn: 0.2162183        total: 5.53s    remaining: 1.73s
762:    learn: 0.2161617        total: 5.54s    remaining: 1.72s
763:    learn: 0.2160619        total: 5.55s    remaining: 1.71s
764:    learn: 0.2160146        total: 5.55s    remaining: 1.71s
765:    learn: 0.2159820        total: 5.56s    remaining: 1.7s
766:    learn: 0.2159262        total: 5.57s    remaining: 1.69s
767:    learn: 0.2158032        total: 5.58s    remaining: 1.68s
768:    learn: 0.2157841        total: 5.58s    remaining: 1.68s
769:    learn: 0.2157353        total: 5.59s    remaining: 1.67s
770:    learn: 0.2156409        total: 5.6s     remaining: 1.66s
771:    learn: 0.2155545        total: 5.6s     remaining: 1.65s
772:    learn: 0.2155038        total: 5.61s    remaining: 1.65s
773:    learn: 0.2154249        total: 5.62s    remaining: 1.64s
774:    learn: 0.2152980        total: 5.63s    remaining: 1.63s
775:    learn: 0.2152451        total: 5.63s    remaining: 1.63s
776:    learn: 0.2151989        total: 5.64s    remaining: 1.62s
777:    learn: 0.2151439        total: 5.65s    remaining: 1.61s
778:    learn: 0.2151294        total: 5.66s    remaining: 1.6s
779:    learn: 0.2150510        total: 5.66s    remaining: 1.6s
780:    learn: 0.2149460        total: 5.67s    remaining: 1.59s
781:    learn: 0.2148635        total: 5.68s    remaining: 1.58s
782:    learn: 0.2148134        total: 5.68s    remaining: 1.57s
```

```
783:    learn: 0.2147621        total: 5.69s    remaining: 1.57s
784:    learn: 0.2146921        total: 5.7s     remaining: 1.56s
785:    learn: 0.2146173        total: 5.7s     remaining: 1.55s
786:    learn: 0.2145807        total: 5.71s    remaining: 1.54s
787:    learn: 0.2145516        total: 5.71s    remaining: 1.54s
788:    learn: 0.2144763        total: 5.72s    remaining: 1.53s
789:    learn: 0.2143402        total: 5.73s    remaining: 1.52s
790:    learn: 0.2142761        total: 5.73s    remaining: 1.51s
791:    learn: 0.2141205        total: 5.74s    remaining: 1.51s
792:    learn: 0.2140544        total: 5.75s    remaining: 1.5s
793:    learn: 0.2139586        total: 5.75s    remaining: 1.49s
794:    learn: 0.2138685        total: 5.76s    remaining: 1.49s
795:    learn: 0.2138339        total: 5.76s    remaining: 1.48s
796:    learn: 0.2137315        total: 5.77s    remaining: 1.47s
797:    learn: 0.2136777        total: 5.78s    remaining: 1.46s
798:    learn: 0.2136334        total: 5.79s    remaining: 1.46s
799:    learn: 0.2135114        total: 5.79s    remaining: 1.45s
800:    learn: 0.2134744        total: 5.8s     remaining: 1.44s
801:    learn: 0.2133855        total: 5.81s    remaining: 1.43s
802:    learn: 0.2133243        total: 5.81s    remaining: 1.43s
803:    learn: 0.2132675        total: 5.82s    remaining: 1.42s
804:    learn: 0.2132198        total: 5.83s    remaining: 1.41s
805:    learn: 0.2131744        total: 5.83s    remaining: 1.4s
806:    learn: 0.2131371        total: 5.84s    remaining: 1.4s
807:    learn: 0.2130215        total: 5.85s    remaining: 1.39s
808:    learn: 0.2129453        total: 5.86s    remaining: 1.38s
809:    learn: 0.2128993        total: 5.86s    remaining: 1.38s
810:    learn: 0.2128079        total: 5.87s    remaining: 1.37s
811:    learn: 0.2127103        total: 5.88s    remaining: 1.36s
812:    learn: 0.2126604        total: 5.88s    remaining: 1.35s
813:    learn: 0.2126273        total: 5.89s    remaining: 1.34s
814:    learn: 0.2125361        total: 5.9s     remaining: 1.34s
815:    learn: 0.2124484        total: 5.9s     remaining: 1.33s
816:    learn: 0.2123577        total: 5.91s    remaining: 1.32s
817:    learn: 0.2122587        total: 5.92s    remaining: 1.32s
818:    learn: 0.2121657        total: 5.92s    remaining: 1.31s
819:    learn: 0.2120887        total: 5.93s    remaining: 1.3s
820:    learn: 0.2120243        total: 5.94s    remaining: 1.29s
821:    learn: 0.2119729        total: 5.94s    remaining: 1.29s
822:    learn: 0.2119416        total: 5.95s    remaining: 1.28s
823:    learn: 0.2119208        total: 5.95s    remaining: 1.27s
824:    learn: 0.2118615        total: 5.96s    remaining: 1.26s
825:    learn: 0.2118159        total: 5.97s    remaining: 1.26s
826:    learn: 0.2117179        total: 5.97s    remaining: 1.25s
827:    learn: 0.2116493        total: 5.98s    remaining: 1.24s
828:    learn: 0.2115506        total: 5.99s    remaining: 1.23s
829:    learn: 0.2114755        total: 5.99s    remaining: 1.23s
830:    learn: 0.2113746        total: 6s       remaining: 1.22s
```

```
831:    learn: 0.2113144        total: 6.01s    remaining: 1.21s
832:    learn: 0.2112480        total: 6.01s    remaining: 1.21s
833:    learn: 0.2111552        total: 6.02s    remaining: 1.2s
834:    learn: 0.2111200        total: 6.03s    remaining: 1.19s
835:    learn: 0.2110287        total: 6.03s    remaining: 1.18s
836:    learn: 0.2109524        total: 6.04s    remaining: 1.18s
837:    learn: 0.2109231        total: 6.05s    remaining: 1.17s
838:    learn: 0.2108539        total: 6.05s    remaining: 1.16s
839:    learn: 0.2108313        total: 6.06s    remaining: 1.15s
840:    learn: 0.2108055        total: 6.07s    remaining: 1.15s
841:    learn: 0.2107755        total: 6.07s    remaining: 1.14s
842:    learn: 0.2107020        total: 6.08s    remaining: 1.13s
843:    learn: 0.2106112        total: 6.09s    remaining: 1.13s
844:    learn: 0.2105455        total: 6.09s    remaining: 1.12s
845:    learn: 0.2104612        total: 6.1s     remaining: 1.11s
846:    learn: 0.2103983        total: 6.11s    remaining: 1.1s
847:    learn: 0.2103240        total: 6.11s    remaining: 1.09s
848:    learn: 0.2102703        total: 6.12s    remaining: 1.09s
849:    learn: 0.2102389        total: 6.13s    remaining: 1.08s
850:    learn: 0.2101870        total: 6.13s    remaining: 1.07s
851:    learn: 0.2101313        total: 6.14s    remaining: 1.07s
852:    learn: 0.2100830        total: 6.15s    remaining: 1.06s
853:    learn: 0.2099996        total: 6.15s    remaining: 1.05s
854:    learn: 0.2099623        total: 6.16s    remaining: 1.04s
855:    learn: 0.2099237        total: 6.16s    remaining: 1.04s
856:    learn: 0.2098622        total: 6.17s    remaining: 1.03s
857:    learn: 0.2098137        total: 6.18s    remaining: 1.02s
858:    learn: 0.2097764        total: 6.18s    remaining: 1.01s
859:    learn: 0.2096838        total: 6.19s    remaining: 1.01s
860:    learn: 0.2096157        total: 6.2s     remaining: 1s
861:    learn: 0.2094792        total: 6.2s     remaining: 993ms
862:    learn: 0.2093587        total: 6.21s    remaining: 986ms
863:    learn: 0.2092820        total: 6.22s    remaining: 979ms
864:    learn: 0.2092428        total: 6.22s    remaining: 972ms
865:    learn: 0.2091995        total: 6.23s    remaining: 964ms
866:    learn: 0.2090642        total: 6.24s    remaining: 957ms
867:    learn: 0.2089781        total: 6.25s    remaining: 950ms
868:    learn: 0.2089027        total: 6.25s    remaining: 943ms
869:    learn: 0.2088297        total: 6.26s    remaining: 936ms
870:    learn: 0.2087519        total: 6.27s    remaining: 928ms
871:    learn: 0.2086939        total: 6.27s    remaining: 921ms
872:    learn: 0.2085155        total: 6.28s    remaining: 914ms
873:    learn: 0.2084402        total: 6.29s    remaining: 906ms
874:    learn: 0.2083635        total: 6.29s    remaining: 899ms
875:    learn: 0.2082730        total: 6.3s     remaining: 892ms
876:    learn: 0.2081898        total: 6.31s    remaining: 884ms
877:    learn: 0.2081237        total: 6.31s    remaining: 877ms
878:    learn: 0.2080641        total: 6.32s    remaining: 870ms
```

```
879:    learn: 0.2079973        total: 6.33s    remaining: 863ms
880:    learn: 0.2079346        total: 6.33s    remaining: 855ms
881:    learn: 0.2079012        total: 6.34s    remaining: 848ms
882:    learn: 0.2078279        total: 6.34s    remaining: 841ms
883:    learn: 0.2077146        total: 6.35s    remaining: 833ms
884:    learn: 0.2076680        total: 6.36s    remaining: 826ms
885:    learn: 0.2076089        total: 6.36s    remaining: 819ms
886:    learn: 0.2074882        total: 6.37s    remaining: 811ms
887:    learn: 0.2073932        total: 6.38s    remaining: 804ms
888:    learn: 0.2073032        total: 6.38s    remaining: 797ms
889:    learn: 0.2072535        total: 6.39s    remaining: 790ms
890:    learn: 0.2071752        total: 6.4s     remaining: 783ms
891:    learn: 0.2070802        total: 6.41s    remaining: 775ms
892:    learn: 0.2069873        total: 6.41s    remaining: 768ms
893:    learn: 0.2069236        total: 6.42s    remaining: 761ms
894:    learn: 0.2069048        total: 6.43s    remaining: 754ms
895:    learn: 0.2068350        total: 6.43s    remaining: 747ms
896:    learn: 0.2067929        total: 6.44s    remaining: 739ms
897:    learn: 0.2067379        total: 6.45s    remaining: 732ms
898:    learn: 0.2066103        total: 6.45s    remaining: 725ms
899:    learn: 0.2065210        total: 6.46s    remaining: 718ms
900:    learn: 0.2064406        total: 6.47s    remaining: 710ms
901:    learn: 0.2063355        total: 6.47s    remaining: 703ms
902:    learn: 0.2062793        total: 6.48s    remaining: 696ms
903:    learn: 0.2062264        total: 6.49s    remaining: 689ms
904:    learn: 0.2060988        total: 6.5s     remaining: 682ms
905:    learn: 0.2060553        total: 6.51s    remaining: 675ms
906:    learn: 0.2060111        total: 6.51s    remaining: 668ms
907:    learn: 0.2059688        total: 6.52s    remaining: 661ms
908:    learn: 0.2058764        total: 6.53s    remaining: 654ms
909:    learn: 0.2058013        total: 6.54s    remaining: 646ms
910:    learn: 0.2057514        total: 6.54s    remaining: 639ms
911:    learn: 0.2056684        total: 6.55s    remaining: 632ms
912:    learn: 0.2056085        total: 6.55s    remaining: 625ms
913:    learn: 0.2055341        total: 6.56s    remaining: 617ms
914:    learn: 0.2055046        total: 6.57s    remaining: 610ms
915:    learn: 0.2054337        total: 6.58s    remaining: 603ms
916:    learn: 0.2053615        total: 6.58s    remaining: 596ms
917:    learn: 0.2052547        total: 6.59s    remaining: 589ms
918:    learn: 0.2052307        total: 6.6s     remaining: 582ms
919:    learn: 0.2051881        total: 6.61s    remaining: 575ms
920:    learn: 0.2051262        total: 6.62s    remaining: 567ms
921:    learn: 0.2050871        total: 6.62s    remaining: 560ms
922:    learn: 0.2050590        total: 6.63s    remaining: 553ms
923:    learn: 0.2050180        total: 6.63s    remaining: 546ms
924:    learn: 0.2049462        total: 6.64s    remaining: 539ms
925:    learn: 0.2048782        total: 6.65s    remaining: 531ms
926:    learn: 0.2048353        total: 6.66s    remaining: 524ms
```

```
927:    learn: 0.2047640        total: 6.66s    remaining: 517ms
928:    learn: 0.2047128        total: 6.67s    remaining: 510ms
929:    learn: 0.2046680        total: 6.67s    remaining: 502ms
930:    learn: 0.2046134        total: 6.68s    remaining: 495ms
931:    learn: 0.2045273        total: 6.69s    remaining: 488ms
932:    learn: 0.2044200        total: 6.69s    remaining: 481ms
933:    learn: 0.2043629        total: 6.7s     remaining: 473ms
934:    learn: 0.2043372        total: 6.71s    remaining: 466ms
935:    learn: 0.2042850        total: 6.71s    remaining: 459ms
936:    learn: 0.2042199        total: 6.72s    remaining: 452ms
937:    learn: 0.2041759        total: 6.73s    remaining: 445ms
938:    learn: 0.2041073        total: 6.73s    remaining: 437ms
939:    learn: 0.2040490        total: 6.74s    remaining: 430ms
940:    learn: 0.2040236        total: 6.75s    remaining: 423ms
941:    learn: 0.2039005        total: 6.75s    remaining: 416ms
942:    learn: 0.2038657        total: 6.76s    remaining: 409ms
943:    learn: 0.2038092        total: 6.76s    remaining: 401ms
944:    learn: 0.2037890        total: 6.77s    remaining: 394ms
945:    learn: 0.2037325        total: 6.78s    remaining: 387ms
946:    learn: 0.2036182        total: 6.79s    remaining: 380ms
947:    learn: 0.2035864        total: 6.8s     remaining: 373ms
948:    learn: 0.2035657        total: 6.81s    remaining: 366ms
949:    learn: 0.2034695        total: 6.81s    remaining: 359ms
950:    learn: 0.2034182        total: 6.82s    remaining: 351ms
951:    learn: 0.2033275        total: 6.83s    remaining: 344ms
952:    learn: 0.2032569        total: 6.83s    remaining: 337ms
953:    learn: 0.2031720        total: 6.84s    remaining: 330ms
954:    learn: 0.2031104        total: 6.85s    remaining: 323ms
955:    learn: 0.2030515        total: 6.86s    remaining: 316ms
956:    learn: 0.2029990        total: 6.86s    remaining: 308ms
957:    learn: 0.2029536        total: 6.87s    remaining: 301ms
958:    learn: 0.2028872        total: 6.88s    remaining: 294ms
959:    learn: 0.2027813        total: 6.88s    remaining: 287ms
960:    learn: 0.2026951        total: 6.89s    remaining: 280ms
961:    learn: 0.2025985        total: 6.9s     remaining: 273ms
962:    learn: 0.2025522        total: 6.91s    remaining: 265ms
963:    learn: 0.2025087        total: 6.91s    remaining: 258ms
964:    learn: 0.2024433        total: 6.92s    remaining: 251ms
965:    learn: 0.2023609        total: 6.92s    remaining: 244ms
966:    learn: 0.2023044        total: 6.93s    remaining: 237ms
967:    learn: 0.2022114        total: 6.94s    remaining: 229ms
968:    learn: 0.2021352        total: 6.95s    remaining: 222ms
969:    learn: 0.2020644        total: 6.95s    remaining: 215ms
970:    learn: 0.2020159        total: 6.96s    remaining: 208ms
971:    learn: 0.2019918        total: 6.96s    remaining: 201ms
972:    learn: 0.2019202        total: 6.97s    remaining: 193ms
973:    learn: 0.2018359        total: 6.98s    remaining: 186ms
974:    learn: 0.2018242        total: 6.99s    remaining: 179ms
```

```
975:    learn: 0.2017678        total: 6.99s    remaining: 172ms
976:    learn: 0.2016973        total: 7s       remaining: 165ms
977:    learn: 0.2016479        total: 7.01s    remaining: 158ms
978:    learn: 0.2015503        total: 7.02s    remaining: 151ms
979:    learn: 0.2015067        total: 7.02s    remaining: 143ms
980:    learn: 0.2014804        total: 7.03s    remaining: 136ms
981:    learn: 0.2014377        total: 7.04s    remaining: 129ms
982:    learn: 0.2013627        total: 7.04s    remaining: 122ms
983:    learn: 0.2012774        total: 7.05s    remaining: 115ms
984:    learn: 0.2012208        total: 7.05s    remaining: 107ms
985:    learn: 0.2011773        total: 7.06s    remaining: 100ms
986:    learn: 0.2011058        total: 7.07s    remaining: 93.1ms
987:    learn: 0.2010677        total: 7.07s    remaining: 85.9ms
988:    learn: 0.2009902        total: 7.08s    remaining: 78.8ms
989:    learn: 0.2009337        total: 7.09s    remaining: 71.6ms
990:    learn: 0.2008757        total: 7.09s    remaining: 64.4ms
991:    learn: 0.2008352        total: 7.1s     remaining: 57.3ms
992:    learn: 0.2007729        total: 7.11s    remaining: 50.1ms
993:    learn: 0.2007398        total: 7.11s    remaining: 42.9ms
994:    learn: 0.2006813        total: 7.12s    remaining: 35.8ms
995:    learn: 0.2006076        total: 7.13s    remaining: 28.6ms
996:    learn: 0.2005498        total: 7.13s    remaining: 21.5ms
997:    learn: 0.2004633        total: 7.14s    remaining: 14.3ms
998:    learn: 0.2003937        total: 7.14s    remaining: 7.15ms
999:    learn: 0.2003577        total: 7.15s    remaining: 0us
```

```
[104]: VotingClassifier(estimators=[('gaussian', GaussianNB()),
                                    ('Gridlogistic',
       GridSearchCV(cv=RepeatedStratifiedKFold(n_repeats=3, n_splits=10,
       random_state=1),
                                                 error_score=0,
                                                 estimator=LogisticRegression(),
                                                 n_jobs=-1,
                                                 param_grid={'C': [100, 10, 1.0, 0.1,
                                                                   0.01],
                                                             'penalty': ['l2'],
                                                             'solver': ['newton-cg',
                                                                        'lbfgs',
                                                                        'liblinear']},
                                                 scoring='accuracy')),
                                   ('catboost_classifier',
                                    <…
                                                  n_estimators=494, n_jobs=None,
                                                  num_parallel_tree=None,
                                                  random_state=None, reg_alpha=None,
                                                  reg_lambda=None,
                                                  scale_pos_weight=None,
```

```
                                                subsample=0.7, tree_method=None,
                                                validate_parameters=None,
                                                verbosity=0)),
                                       ('LGBMclassifier',
                                        LGBMClassifier(boosting_type='dart',
                                                       importance_type='gain', max_bin=60,
                                                       max_depth=5, n_estimators=494,
                                                       num_leaves=300, verbosity=-1))],
                           voting='soft')
```

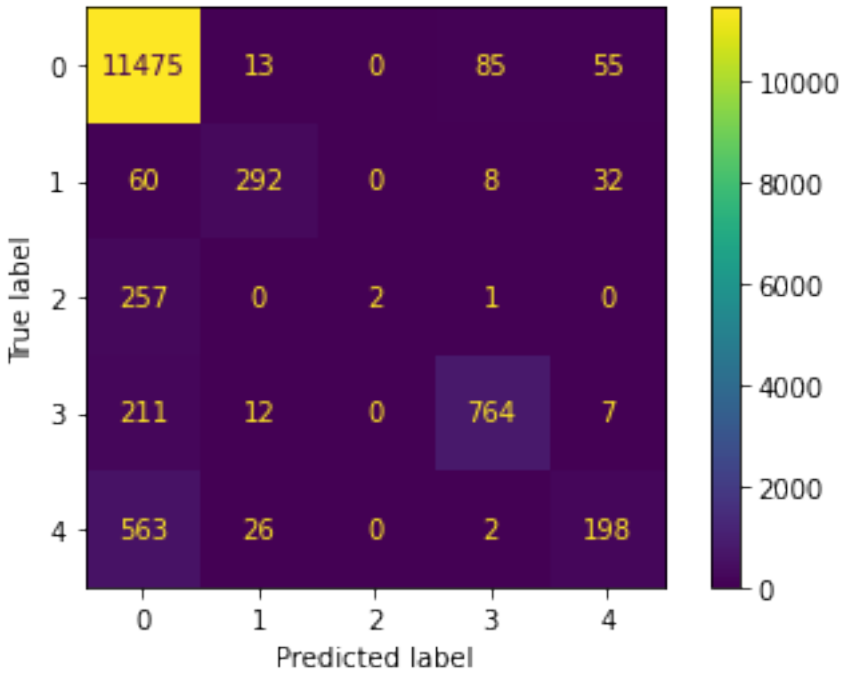
[105]:
```
y_pred = vot_soft.predict(X_test)
```

[106]:
```
metrics.accuracy_score(y_test, y_pred)*100
```

[106]:
```
90.52833677024817
```

[107]:
```
t = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= vot_soft.
 ↪classes_)
disp.plot()
```

[107]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f94c2532190>
```



[108]:
```
#metrics.accuracy_score(y_test, y_pred_gnb)*100
```

[109]:
```
#confusion_matrix(y_test, y_pred_gnb)
```

[110]:
```
#t = confusion_matrix(y_test, y_pred_gnb)
#disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= gnb.
 ↪classes_)
```

[111]:
```
#disp.plot()
```

[112]:
```
#metrics.accuracy_score(y_test, y_pred_log)*100
```

[113]:
```
#t = confusion_matrix(y_test, y_pred_log)
#disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= grid_search.
 ↪classes_)
#disp.plot()
```

[114]:
```
#metrics.accuracy_score(y_test, y_pred_cat)*100
```

[115]:
```
#t = confusion_matrix(y_test, y_pred_cat)
#disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= cat.
 ↪classes_)
#disp.plot()
```

[116]:
```
#metrics.accuracy_score(y_test, y_pred_dt)*100
```

[117]:
```
#t = confusion_matrix(y_test, y_pred_dt)
#disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= dtclf.
 ↪classes_)
#disp.plot()
```

9 TESTING DATA

[118]:
```
path = '/media/mr-robot/Local Disk/summer_training/test'
os.chdir(path)
```

[119]:
```
# Converting all las files in csv by iterating using lasio
for file in os.listdir():
    if file.endswith(".las"):
        file_path = f"{path}/{file}"
        las=lasio.read(file_path)
        size=len(file_path)
        filepath1=file_path[:size-3]
        las.to_csv(filepath1+'csv', units=False)
```

[120]:
```
## To avoid furthur merging data and redundancy
if(os.path.isfile('./merged_data.csv') ):
    os.remove("merged_data.csv")
```

```
if(os.path.isfile('./FACIES_imputed.csv')):
    os.remove("FACIES_imputed.csv")

if(os.path.isfile('./FACIES_TRAIN.csv')):
    os.remove("FACIES_TRAIN.csv")
```

[121]:
```
# Merging all Well Log using Glob
filenames = glob.glob(path + "/*.csv")
dfs = []
for filename in filenames:
    dfs.append(pd.read_csv(filename))
big_frame = pd.concat(dfs, ignore_index=True)
big_frame.to_csv('merged_data.csv',index=False)
```

[122]:
```
df = pd.read_csv('merged_data.csv')
df
```

[122]:

	DEPTH	ACOUSTICIMPEDANCE1	AI	AVG_PIGN	BIT	CALI	\
0	1197.4072	5252.3882	5252388.0	NaN	0.2159	8.9012	
1	1197.5596	5289.7070	5289707.0	NaN	0.2159	8.9005	
2	1197.7120	5245.4429	5245443.0	NaN	0.2159	8.8957	
3	1197.8644	5181.9023	5181902.5	NaN	0.2159	8.8932	
4	1198.0168	5131.1343	5131134.5	NaN	0.2159	8.8980	
…	…	…	…	…	…	…	
29560	1689.5065	6013.4722	6013472.5	NaN	0.2159	NaN	
29561	1689.6589	5953.0059	5953006.0	NaN	0.2159	NaN	
29562	1689.8113	5954.4824	5954482.0	NaN	0.2159	NaN	
29563	1689.9637	5911.3301	5911330.0	NaN	0.2159	NaN	
29564	1690.1161	5930.9585	5930958.5	NaN	0.2159	NaN	

	NPHI	DT	FACIES	FLD1	…	SPSD	ZCOR	BS	CALI[DERIVED]1	\
0	0.4682	133.4417	NaN	NaN	…	NaN	NaN	NaN	NaN	
1	0.4585	132.4196	NaN	NaN	…	NaN	NaN	NaN	NaN	
2	0.4543	133.3569	NaN	NaN	…	NaN	NaN	NaN	NaN	
3	0.4827	134.7392	NaN	NaN	…	NaN	NaN	NaN	NaN	
4	0.5361	135.7694	NaN	NaN	…	NaN	NaN	NaN	NaN	
…	…	…	…	…	…	…	…	..	…	
29560	NaN	126.6800	NaN	NaN	…	NaN	NaN	NaN	NaN	
29561	NaN	127.9872	NaN	NaN	…	NaN	NaN	NaN	NaN	
29562	NaN	127.9657	NaN	NaN	…	NaN	NaN	NaN	NaN	
29563	NaN	128.9050	NaN	NaN	…	NaN	NaN	NaN	NaN	
29564	NaN	128.4784	NaN	NaN	…	NaN	NaN	NaN	NaN	

	DFL	GRCO	HDRS	HMRS	PHIT	TEMP1
0	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN

```
3      NaN   NaN   NaN   NaN   NaN    NaN
4      NaN   NaN   NaN   NaN   NaN    NaN
…    …   …   …   …   …    …
29560  NaN   NaN   NaN   NaN   NaN    NaN
29561  NaN   NaN   NaN   NaN   NaN    NaN
29562  NaN   NaN   NaN   NaN   NaN    NaN
29563  NaN   NaN   NaN   NaN   NaN    NaN
29564  NaN   NaN   NaN   NaN   NaN    NaN

[29565 rows x 55 columns]
```

[123]:
```
#Selecting required feature
df=df[["DT","GR","NPHI","RHOB","FACIES"]]
```

[124]:
```
df
```

[124]:
```
              DT       GR    NPHI    RHOB  FACIES
0       133.4417  87.3154  0.4682  2.2995     NaN
1       132.4196  88.5412  0.4585  2.2981     NaN
2       133.3569  87.5764  0.4543  2.2950     NaN
3       134.7392  86.0361  0.4827  2.2907     NaN
4       135.7694  85.0393  0.5361  2.2856     NaN
…          …       …      …       …      …
29560   126.6800      NaN     NaN  2.4993     NaN
29561   127.9872      NaN     NaN  2.4997     NaN
29562   127.9657      NaN     NaN  2.4999     NaN
29563   128.9050      NaN     NaN  2.5000     NaN
29564   128.4784      NaN     NaN  2.5000     NaN

[29565 rows x 5 columns]
```

[125]:
```
df=imputing(imputation_strategy[optionimputation],df)
df
```

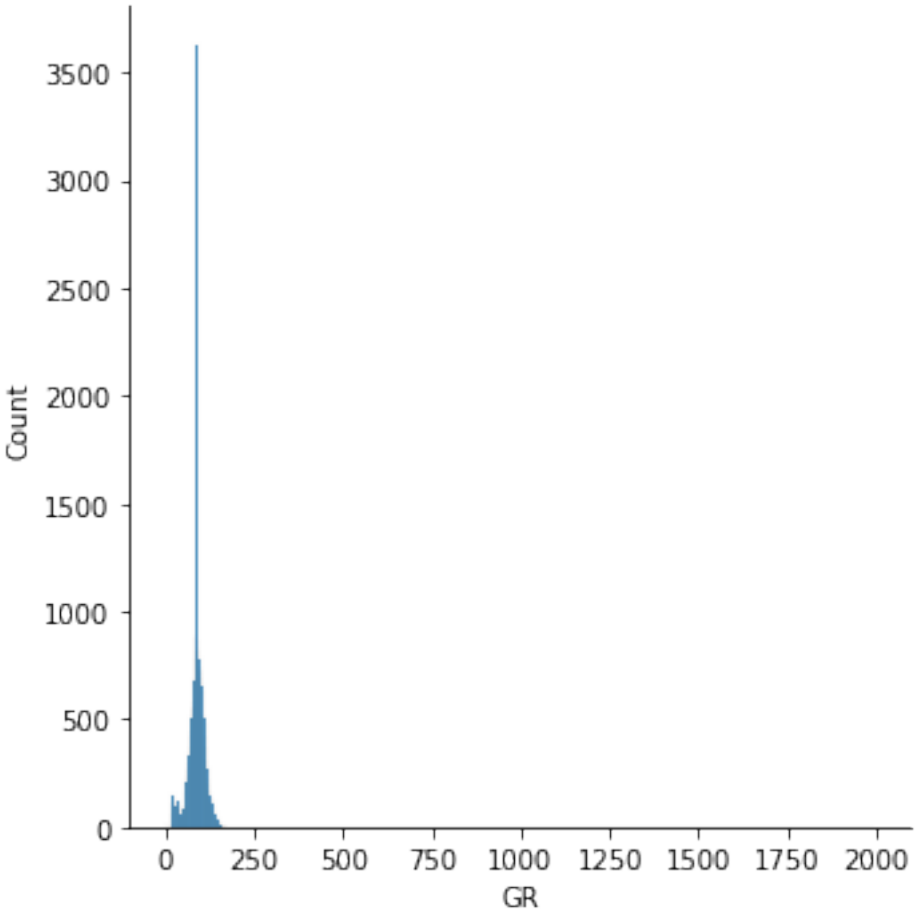
```
0
Graph (GR) after filling null values with mean
Graph (NPHI) after filling null values with mean
0
DT        0
GR        0
NPHI      0
RHOB      0
FACIES    0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25801 entries, 643 to 29515
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
```

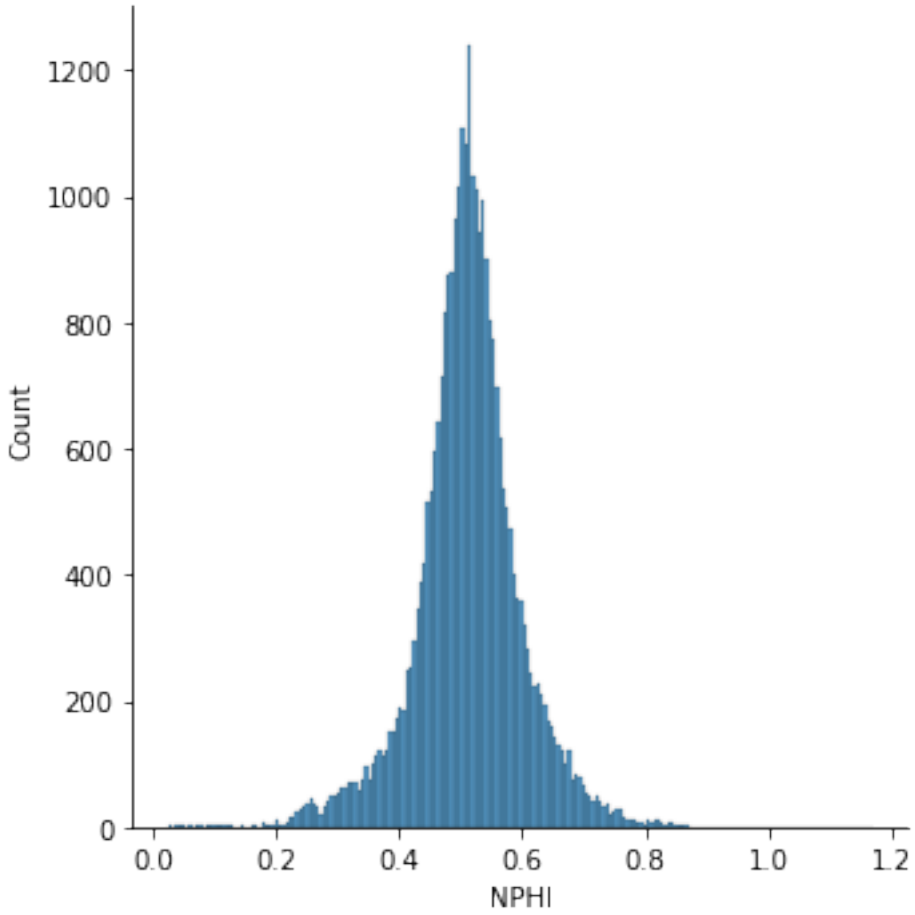
```
---  ------  --------------  -----
 0   DT      25801 non-null  float64
 1   GR      25801 non-null  float64
 2   NPHI    25801 non-null  float64
 3   RHOB    25801 non-null  float64
 4   FACIES  25801 non-null  int64
dtypes: float64(4), int64(1)
memory usage: 1.2 MB
```

[125]:

	DT	GR	NPHI	RHOB	FACIES
643	143.7439	77.1611	0.6496	2.0121	0
644	143.2483	78.0601	0.6805	1.9364	0
645	144.5881	78.3862	0.6749	1.7739	3
646	146.9913	72.1231	0.6718	1.5568	3
647	148.7089	60.5277	0.6966	1.3747	3
…	…	…	…	…	…
29511	122.8153	98.0364	0.4711	2.1809	0
29512	123.6421	95.3973	0.4639	2.1820	0
29513	124.8747	92.3430	0.4595	2.1887	0
29514	126.8160	89.6435	0.4767	2.2003	0
29515	127.8710	87.6556	0.5074	2.2156	0

[25801 rows x 5 columns]


3500
3000
2500
2000
1500
1000
500
0
Count
0
250
500
750
1000
1250
1500
1750
2000
GR



[126]:
```
df = outliers(DATAConditioningStrategy[optionoutlier] , df,␣
↪DATAConditioningColumns)
```

```
column DT
3 standard deviation outliers -:
             DT         GR    NPHI    RHOB  FACIES
3981    75.3027  73.368300  0.5153  2.5090       0
3982    73.6734  73.261800  0.5041  2.4475       0
6110    76.3801  96.356900  0.3313  2.7004       0
13939   70.9828  95.723000  0.4255  3.0317       0
13940   75.5917  94.711500  0.4245  2.9428       0
15678  172.0652  88.702839  0.5173  2.3382       0
15679  175.1408  88.702839  0.5044  2.3501       0
15680  173.8879  88.702839  0.4875  2.3948       0
15706  172.7409  88.702839  0.5074  2.4185       0
15707  174.8540  88.702839  0.4967  2.4147       0
15708  172.7833  88.702839  0.4784  2.4165       0
16123   76.3119  88.702839  0.3927  3.0026       0
```

```
16907  173.0850  88.702839  0.6734  1.8918       0
23404   72.9019  86.674800  0.3879  2.6145       0
23405   73.6668  86.070200  0.3612  2.5231       0
28926   78.1889  66.276900  0.4540  2.9479       0
(16, 5)
             DT       GR    NPHI    RHOB  FACIES
643    143.7439  77.1611  0.6496  2.0121       0
644    143.2483  78.0601  0.6805  1.9364       0
645    144.5881  78.3862  0.6749  1.7739       3
646    146.9913  72.1231  0.6718  1.5568       3
647    148.7089  60.5277  0.6966  1.3747       3
…         …       …     …      …     …
29511  122.8153  98.0364  0.4711  2.1809       0
29512  123.6421  95.3973  0.4639  2.1820       0
29513  124.8747  92.3430  0.4595  2.1887       0
29514  126.8160  89.6435  0.4767  2.2003       0
29515  127.8710  87.6556  0.5074  2.2156       0

[25785 rows x 5 columns]
column GR
3 standard deviation outliers -:
             DT        GR    NPHI    RHOB  FACIES
6713   106.0169  171.0279  0.4576  2.3747       0
6714   103.2064  177.2435  0.4532  2.4002       0
6715    97.3051  169.6761  0.4546  2.4704       0
8992   119.9756  170.0602  0.5277  2.3908       0
8993   124.2488  169.7567  0.5524  2.3856       0
8994   128.2512  163.4437  0.5622  2.3528       0
9173   115.0370  162.8367  0.5044  2.2922       0
9174   112.9930  162.9981  0.5193  2.2804       0
11798  149.0447   16.4786  0.7822  1.0974       3
11799  149.1794   16.0217  0.7487  1.0968       3
11800  149.7072   16.5116  0.7227  1.0936       3
11801  150.5438   17.0523  0.7356  1.0916       3
11802  151.6902   17.0036  0.7567  1.0959       3
11803  151.8816   17.0432  0.7523  1.1044       3
11807  150.7018   16.7745  0.7656  1.0894       3
14068  114.9280  167.3294  0.4747  2.4798       0
14069  116.1878  170.4165  0.4722  2.4701       0
14085  104.8841  166.8226  0.4847  2.5464       0
14086  110.8359  173.5260  0.4654  2.5244       0
14087  108.1552  168.0866  0.4556  2.5867       0
14090  102.4340  162.4829  0.4683  2.5099       0
14091  106.9375  173.8383  0.4934  2.4539       0
14092  109.4050  177.5311  0.5145  2.4617       0
14093  105.8563  168.5443  0.5259  2.5342       0
20873  144.7165   16.7429  0.6031  1.0840       3
20874  144.7407   16.8696  0.5880  1.0766       3
```

```
21174  145.7040   16.9913  0.7395  1.0769       3
21825  149.9749   15.9072  0.6041  1.1148       3
21826  150.0421   14.7489  0.6211  1.1160       3
21827  150.1681   14.2670  0.6517  1.1162       3
21828  150.3558   14.3121  0.6625  1.1168       3
21829  150.6390   14.5554  0.6335  1.1187       3
21830  150.8810   14.8455  0.6132  1.1211       3
21831  151.0431   15.2466  0.6475  1.1226       3
21832  151.1666   15.8596  0.7098  1.1221       3
21833  151.1997   16.7785  0.7470  1.1199       3
23534  102.8643  162.7138  0.4767  2.4589       0
23630  113.8739  165.7769  0.5210  2.4317       0
23643  120.1768  162.9347  0.5397  2.3526       0
23644  130.3486  171.4542  0.5593  2.2894       0
23645  132.3959  170.9262  0.5376  2.2852       0
23646  130.5374  162.9399  0.5158  2.3263       0
23654  125.4795  166.8859  0.5280  2.4625       0
23655  119.4731  169.0622  0.4950  2.5335       0
23660  123.5364  172.4396  0.5018  2.4531       0
23661  125.7776  176.4603  0.4890  2.4261       0
23662  119.6145  170.7566  0.4569  2.4612       0
23677  107.9535  165.4008  0.5177  2.4557       0
23678  103.9874  168.2264  0.5091  2.5137       0
23682  115.3539  170.3146  0.5415  2.5352       0
23683  125.3887  179.4016  0.5361  2.4888       0
23684  126.7310  174.8743  0.5220  2.5545       0
23848  113.4049  163.3054  0.4610  2.5142       0
24530  132.3120  165.7702  0.5849  2.4287       0
24531  132.2990  169.5260  0.5889  2.4042       0
24532  132.6524  168.5493  0.5897  2.3819       0
24635  133.1989  162.6861  0.4445  2.3513       0
(57, 5)
             DT       GR    NPHI    RHOB  FACIES
643    143.7439  77.1611  0.6496  2.0121       0
644    143.2483  78.0601  0.6805  1.9364       0
645    144.5881  78.3862  0.6749  1.7739       3
646    146.9913  72.1231  0.6718  1.5568       3
647    148.7089  60.5277  0.6966  1.3747       3
…         …       …     …      …     …
29511  122.8153  98.0364  0.4711  2.1809       0
29512  123.6421  95.3973  0.4639  2.1820       0
29513  124.8747  92.3430  0.4595  2.1887       0
29514  126.8160  89.6435  0.4767  2.2003       0
29515  127.8710  87.6556  0.5074  2.2156       0

[25728 rows x 5 columns]
column NPHI
3 standard deviation outliers -:
```

```
             DT       GR    NPHI    RHOB  FACIES
2106   150.9081  24.1867  0.8250  1.0699       3
2107   150.4199  26.6826  0.8147  1.1273       3
3666   116.6654  72.7995  0.2374  1.9826       1
3667   115.3509  64.1530  0.1863  1.9223       1
3668   112.0577  57.4443  0.1480  1.8899       1
…           …       …       …       …       …
25377  117.9592  96.2781  0.2103  2.2595       1
25378  119.1010  99.0550  0.2309  2.2574       1
25515  115.9660  63.3382  0.2259  2.1641       1
25538  110.9418  72.4180  0.2338  2.1608       1
25539  113.6031  72.6727  0.2261  2.1554       1

[313 rows x 5 columns]
(313, 5)
             DT       GR    NPHI    RHOB  FACIES
643    143.7439  77.1611  0.6496  2.0121       0
644    143.2483  78.0601  0.6805  1.9364       0
645    144.5881  78.3862  0.6749  1.7739       3
646    146.9913  72.1231  0.6718  1.5568       3
647    148.7089  60.5277  0.6966  1.3747       3
…           …       …       …       …       …
29511  122.8153  98.0364  0.4711  2.1809       0
29512  123.6421  95.3973  0.4639  2.1820       0
29513  124.8747  92.3430  0.4595  2.1887       0
29514  126.8160  89.6435  0.4767  2.2003       0
29515  127.8710  87.6556  0.5074  2.2156       0

[25415 rows x 5 columns]
column RHOB
3 standard deviation outliers -:
             DT       GR    NPHI    RHOB  FACIES
1515   114.5461  41.0112  0.6470  1.1092       3
1516   115.6053  37.5339  0.6715  1.1197       3
1625   149.5008  36.3442  0.6133  1.1143       3
1626   150.9417  29.3642  0.6122  1.0951       3
1627   149.6250  27.0870  0.6129  1.1136       3
…           …       …       …       …       …
28973  154.5691  32.9489  0.6210  1.1474       3
28974  152.7028  39.2345  0.6226  1.1453       3
28975  150.7755  43.9511  0.6209  1.1472       3
28976  149.7247  45.9729  0.6084  1.1460       3
28977  150.0203  46.1717  0.6030  1.1502       3

[894 rows x 5 columns]
(894, 5)
             DT       GR    NPHI    RHOB  FACIES
643    143.7439  77.1611  0.6496  2.0121       0
```

```
644    143.2483  78.0601  0.6805  1.9364       0
645    144.5881  78.3862  0.6749  1.7739       3
646    146.9913  72.1231  0.6718  1.5568       3
647    148.7089  60.5277  0.6966  1.3747       3
…           …       …       …       …       …
29511  122.8153  98.0364  0.4711  2.1809       0
29512  123.6421  95.3973  0.4639  2.1820       0
29513  124.8747  92.3430  0.4595  2.1887       0
29514  126.8160  89.6435  0.4767  2.2003       0
29515  127.8710  87.6556  0.5074  2.2156       0

[24521 rows x 5 columns]
```

[127]:
```
df = data_scaling( scaling_strategy[optionscaling] , df ,␣
↪DATAConditioningColumns )
```

[128]:
```
df.to_csv("testing_preprocessed.csv",index=False)
```

[129]:
```
df=pd.read_csv('testing_preprocessed.csv')
```

[130]:
```
df
```

[130]:
```
              DT        GR      NPHI      RHOB  FACIES
0       1.017560 -0.623055  1.691765 -1.325945       0
1       0.993879 -0.582175  2.055294 -1.654933       0
2       1.057899 -0.567346  1.989412 -2.361147       3
3       1.172731 -0.852149  1.952941 -3.304650       3
4       1.254803 -1.379428  2.244706 -4.096045       3
…            …       …        …       …       …
24516   0.017527  0.326211 -0.408235 -0.592351       0
24517   0.057034  0.206203 -0.492941 -0.587571       0
24518   0.115931  0.067314 -0.544706 -0.558453       0
24519   0.208693 -0.055441 -0.342353 -0.508040       0
24520   0.259104 -0.145837  0.018824 -0.441547       0

[24521 rows x 5 columns]
```

[131]:
```
X_testing=df[["DT","GR","NPHI","RHOB"]]
y_testing=df[["FACIES"]]
```

[132]:
```
X_testing.isnull().sum()
```

[132]:
```
DT      0
GR      0
NPHI    0
RHOB    0
dtype: int64
```

[133]:
```
#X_testing=FeatureSelection(FeatureSelectionStrategy[optionfeature],X_testing,y_testing)
```

[ ]:
```

```

[134]:
```
X_testing
```

[134]:
```
              DT        GR      NPHI      RHOB
0       1.017560 -0.623055  1.691765 -1.325945
1       0.993879 -0.582175  2.055294 -1.654933
2       1.057899 -0.567346  1.989412 -2.361147
3       1.172731 -0.852149  1.952941 -3.304650
4       1.254803 -1.379428  2.244706 -4.096045
…            …        …        …        …
24516   0.017527  0.326211 -0.408235 -0.592351
24517   0.057034  0.206203 -0.492941 -0.587571
24518   0.115931  0.067314 -0.544706 -0.558453
24519   0.208693 -0.055441 -0.342353 -0.508040
24520   0.259104 -0.145837  0.018824 -0.441547

[24521 rows x 4 columns]
```

[135]:
```
y_testing
```

[135]:
```
       FACIES
0           0
1           0
2           3
3           3
4           3
…         …
24516       0
24517       0
24518       0
24519       0
24520       0

[24521 rows x 1 columns]
```

[136]:
```
y_predicted = vot_soft.predict(X_testing)
```

[137]:
```
y_predicted
```

[137]:
```
array([0, 0, 3, …, 0, 0, 0])
```

[138]:
```
metrics.accuracy_score(y_testing, y_predicted)*100
```

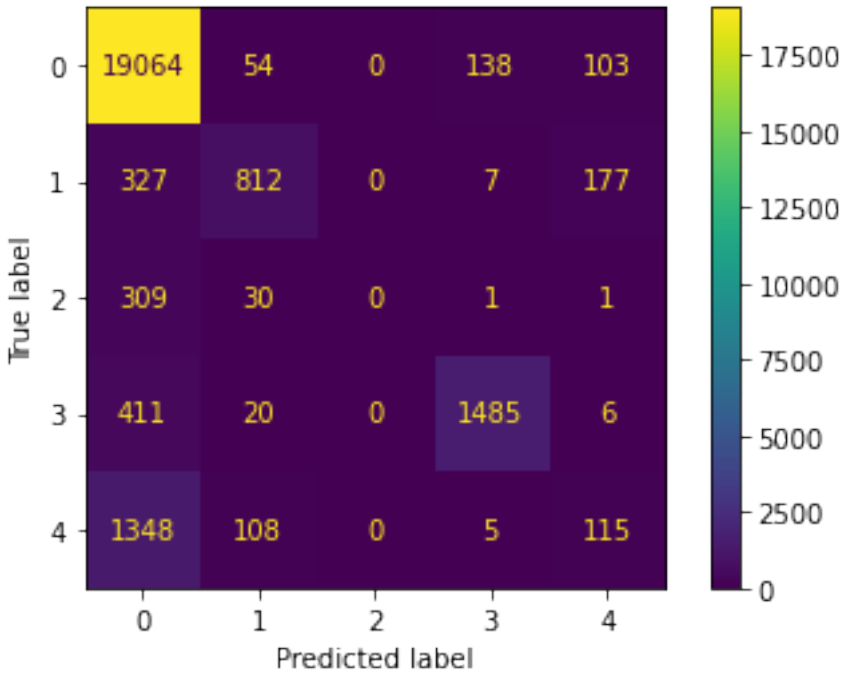
[138]:
```
87.58207250927776
```

[139]:
```
confusion_matrix(y_testing, y_predicted)
```

[139]:
```
array([[19064,    54,     0,   138,   103],
       [  327,   812,     0,     7,   177],
       [  309,    30,     0,     1,     1],
       [  411,    20,     0,  1485,     6],
       [ 1348,   108,     0,     5,   115]])
```

[140]:
```
t = confusion_matrix(y_testing, y_predicted)
disp = ConfusionMatrixDisplay(confusion_matrix= t, display_labels= vot_soft.
 ↪classes_)
disp.plot()
```

[140]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7f93f5949c40>
```



[141]:
```
t1=pd.DataFrame(y_testing)
```

[142]:
```
t1.to_csv('y_given.csv',index=False)
```

[143]:
```
t2=pd.DataFrame(y_predicted)
```

[144]:
```
t2.to_csv('y_predicted.csv',index=False)
```

[ ]:
```

```