main_new_0_1_2_1

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
```

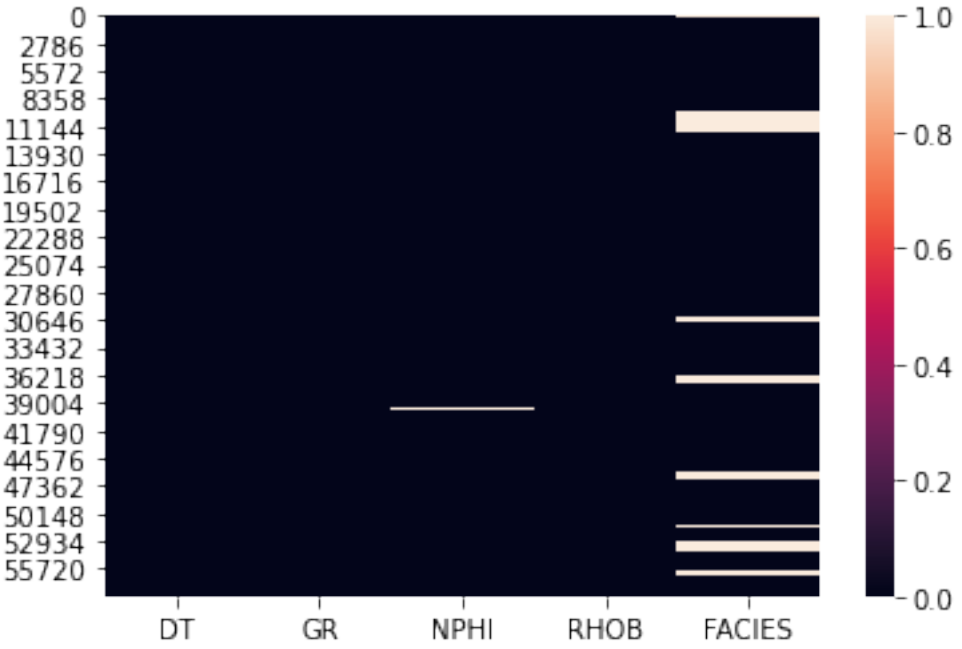
[28]:
```
(58499, 5)
```

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

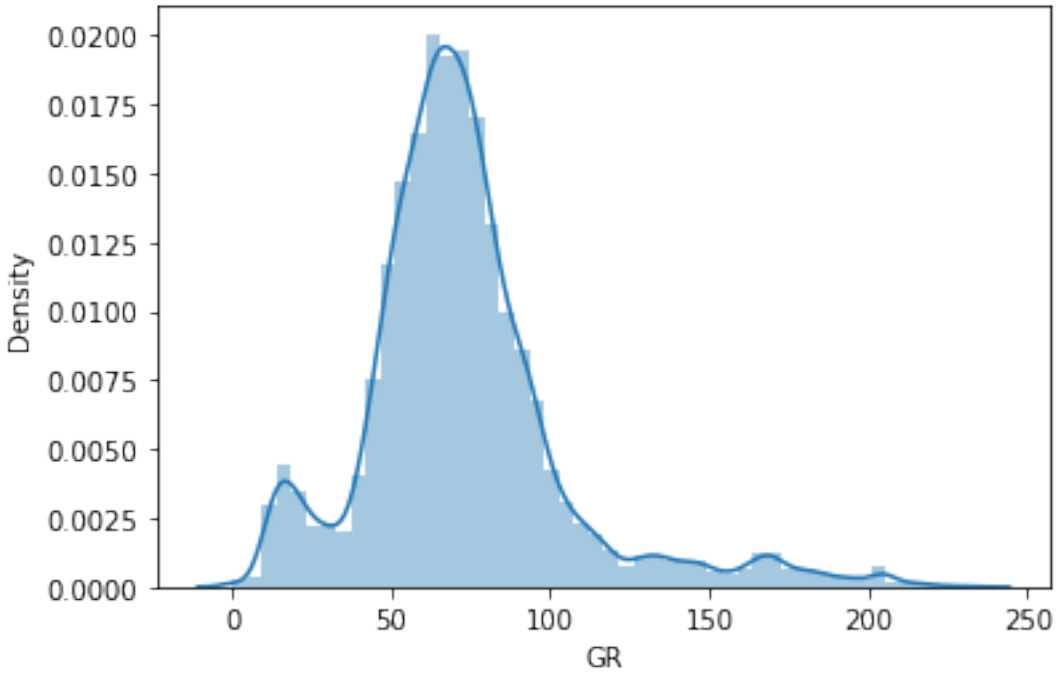
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DT      58499 non-null  float64
 1   GR      58379 non-null  float64
 2   NPHI    58172 non-null  float64
 3 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



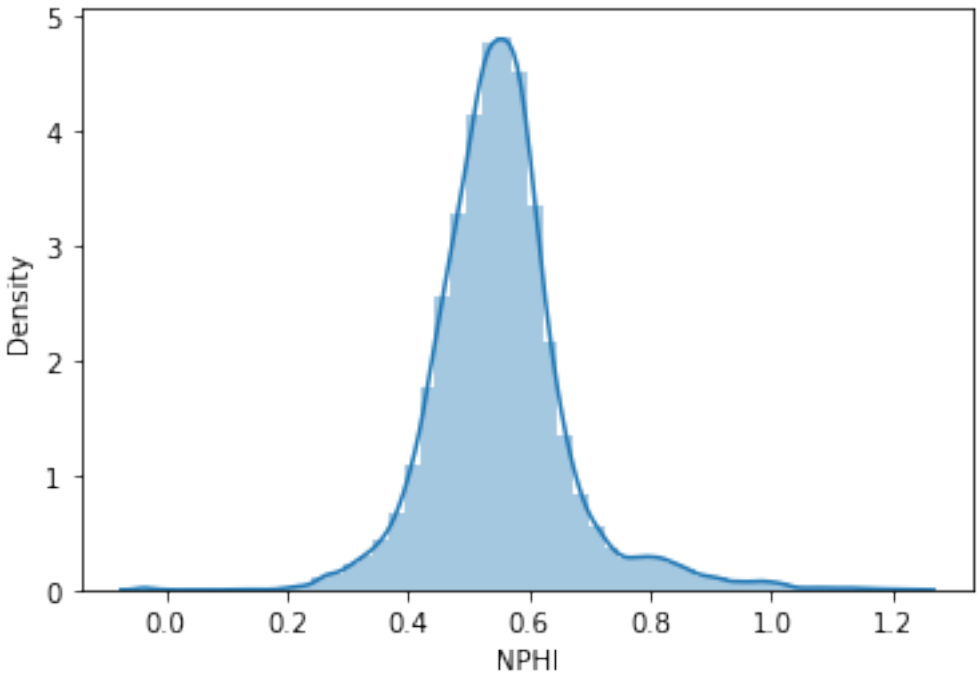
[31]:
```
null_gr = sns.distplot(df.GR.dropna())
```



[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())
```

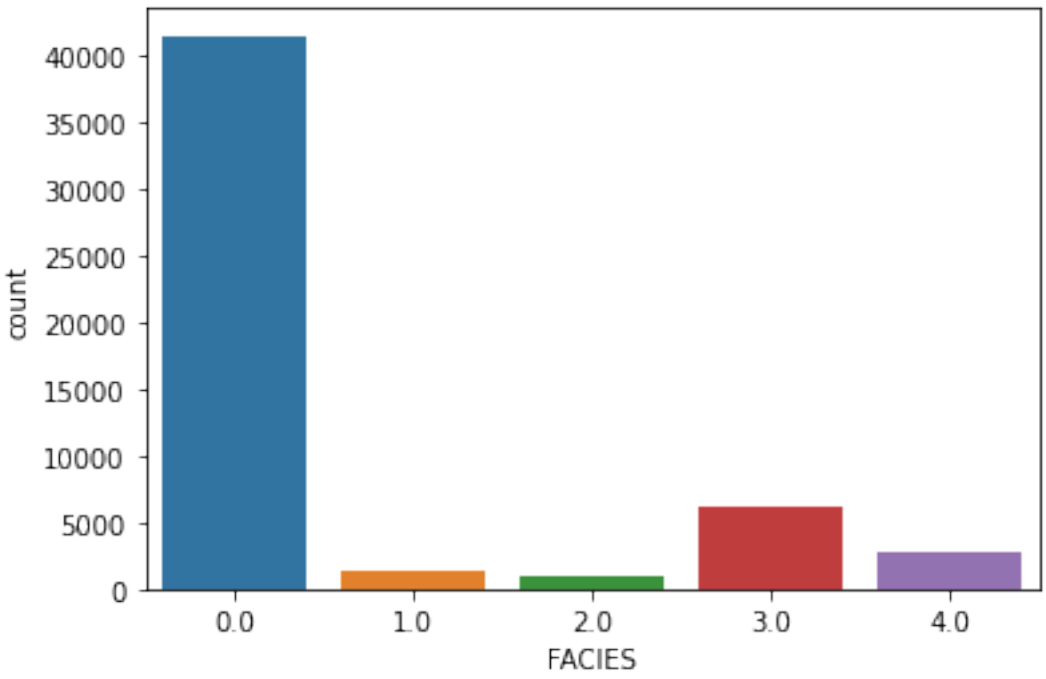


[34]:
```
df.NPHI.describe()
```

[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)
```

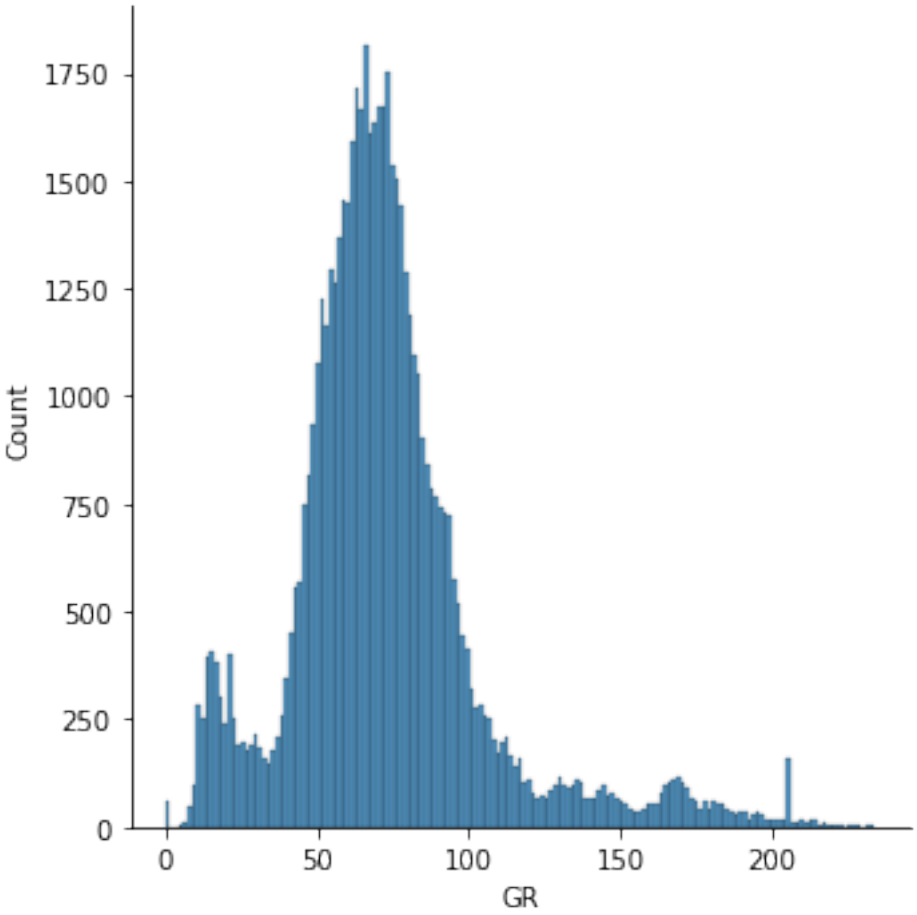
```
        df.info()
        df.FACIES.describe()
        return df

    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
```

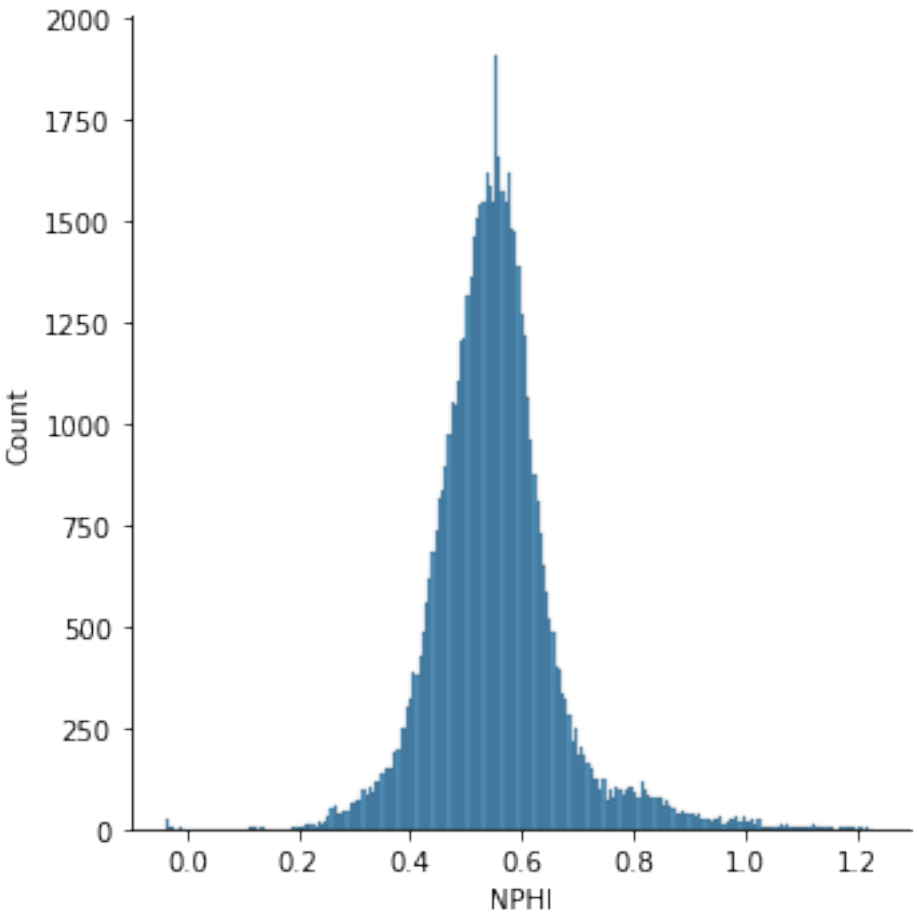
```
        knn= KNNImputer(n_neighbors=3)
        t=knn.fit_transform(df)
        df=pd.DataFrame(t)
        d=['DT','GR','NPHI','RHOB','FACIES']
        df['FACIES1'] = X.FACIES
        for i in range(5):
            df.columns.values[i]=d[i]
        df=df.drop('FACIES1',1)
        df['FACIES'] = df.FACIES.astype(np.int64)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

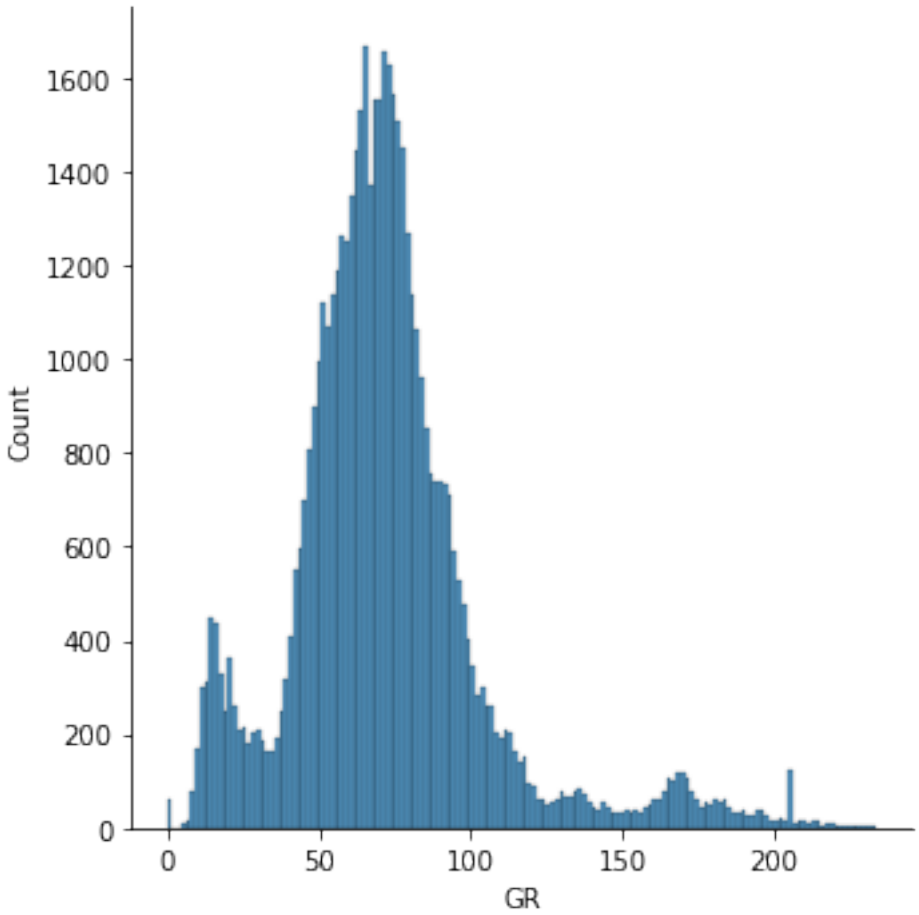
[41]:
```
df.isnull().sum()
```

[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())
```

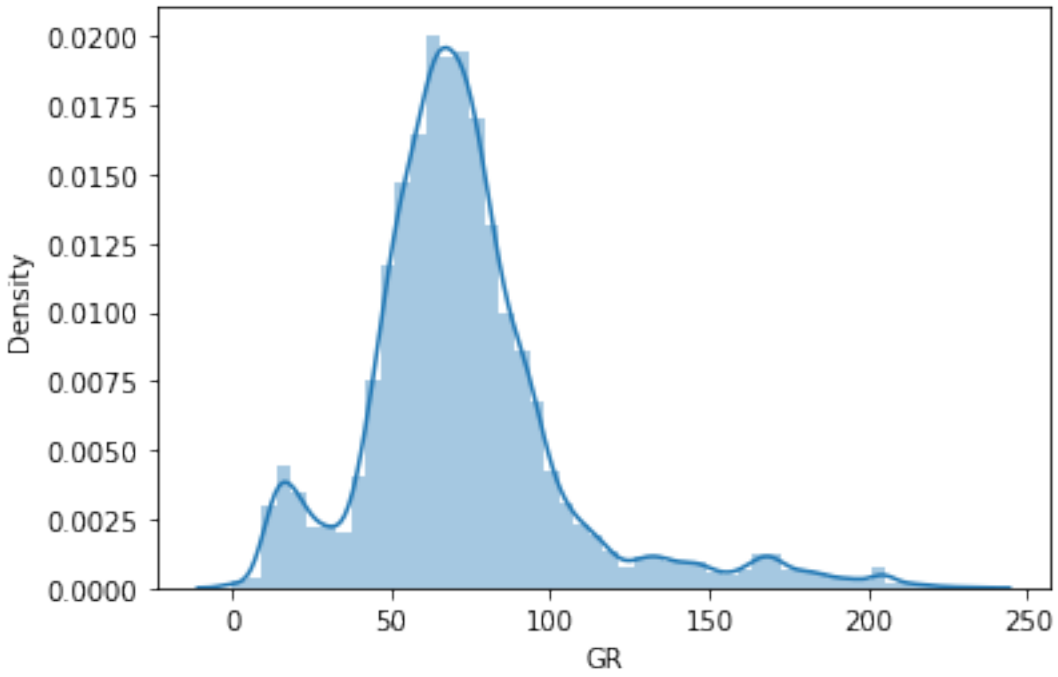
[43]:
```
<seaborn.axisgrid.FacetGrid at 0x7fcc1f03dfa0>
```



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

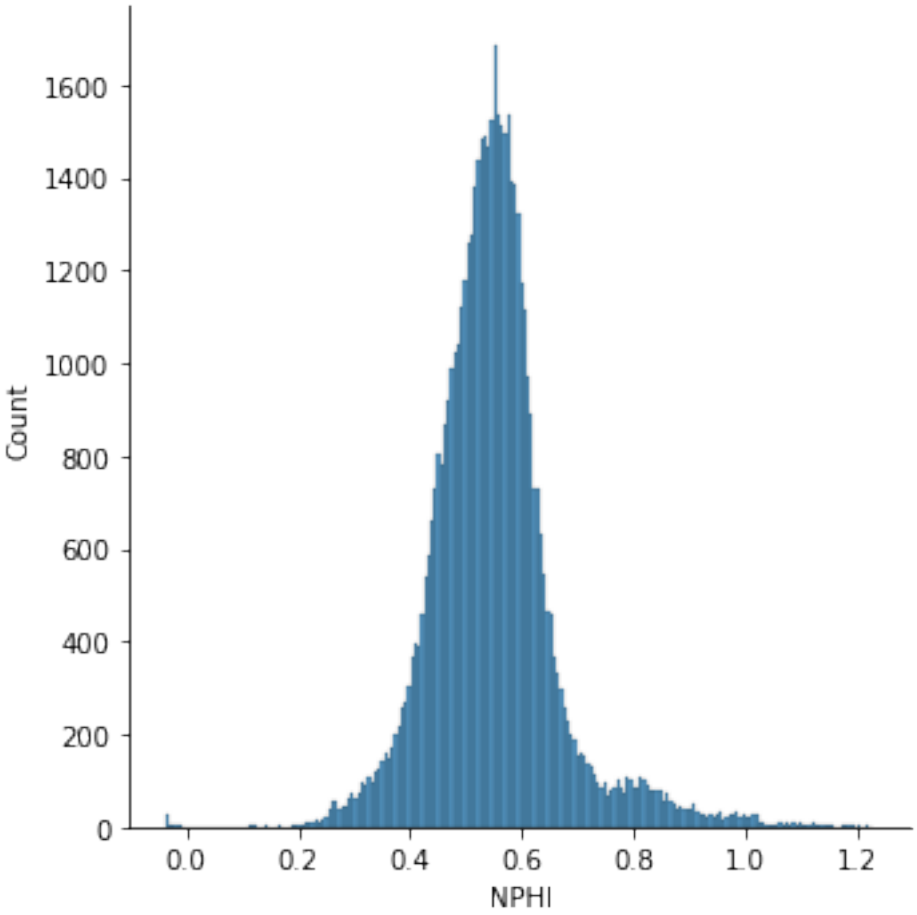
```
WHEN GR WAS NULL
```

[44]:



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

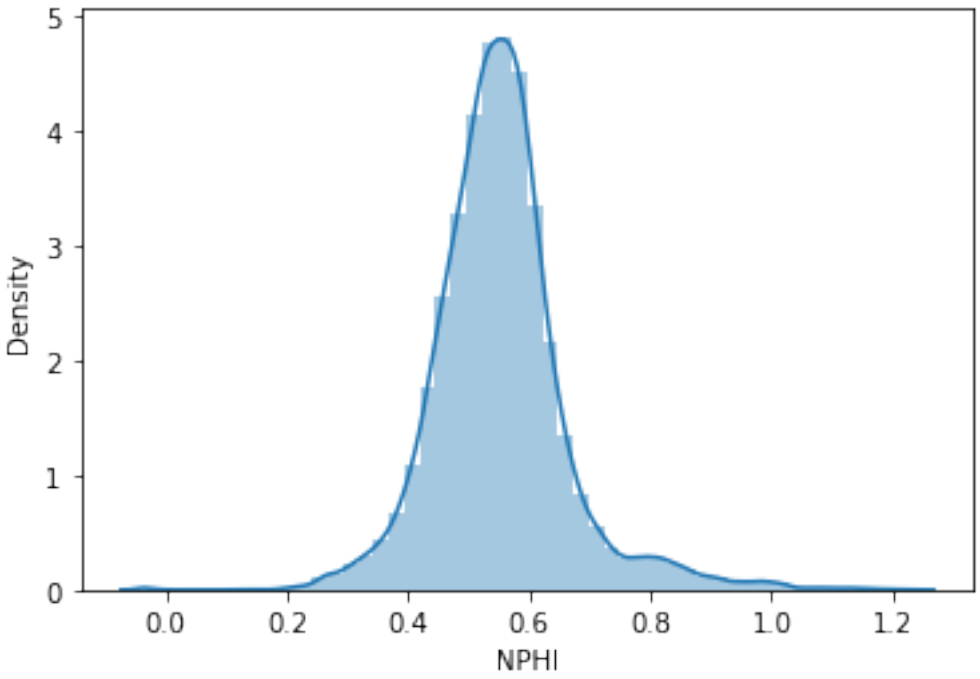
[45]:
```
<seaborn.axisgrid.FacetGrid at 0x7fcc1eb96a60>
```



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

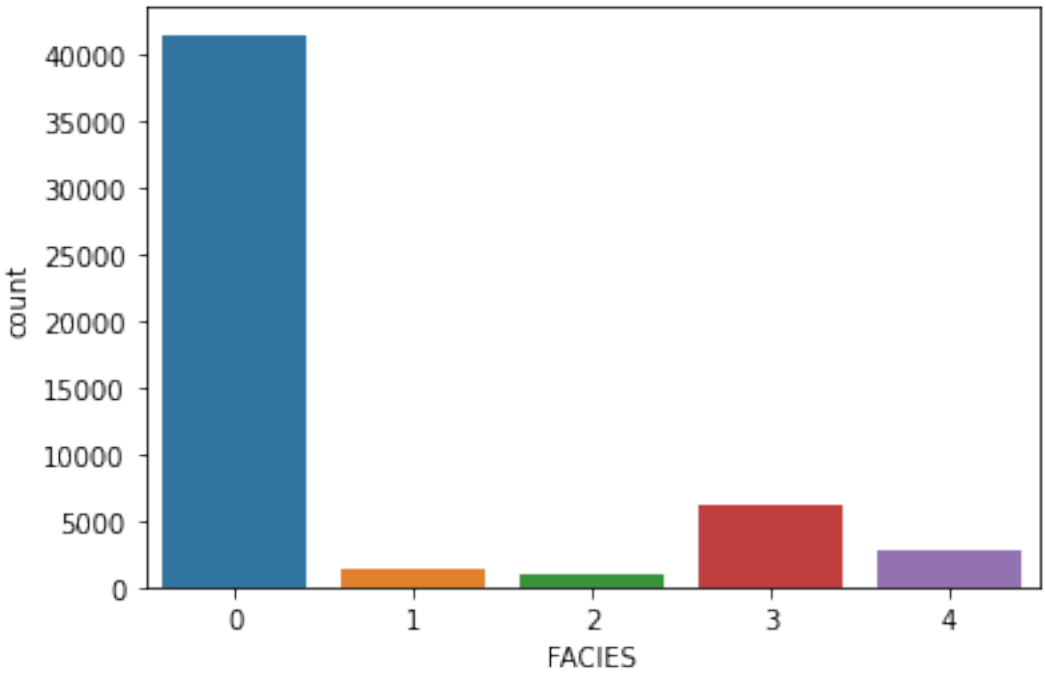
```
WHEN NPHI WAS NULL
```

[46]:



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

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



4 DATA CONDITIONING / OUTLIER REMOVAL

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

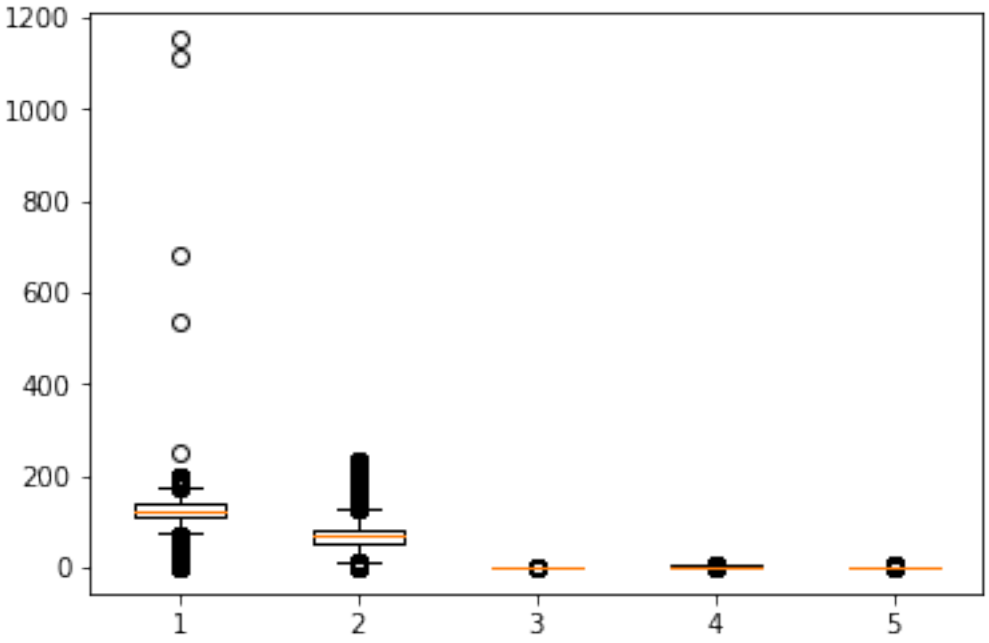
[48]:
```
<bound method NDFrame.head of              DT       GR    NPHI    RHOB  FACIES
0       51.9301  67.3725  0.5192  2.1625       0
1       49.5776  69.2251  0.5173  2.1624       0
2       48.4933  70.2807  0.5094  2.1608       0
3       48.7997  71.6177  0.4974  2.1703       0
4       49.0683  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 0x7fcc1d6dbdc0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6ec190>,
  <matplotlib.lines.Line2D at 0x7fcc1d6f7790>,
  <matplotlib.lines.Line2D at 0x7fcc1d6f7b20>,
  <matplotlib.lines.Line2D at 0x7fcc1d70c100>,
  <matplotlib.lines.Line2D at 0x7fcc1d70c490>,
  <matplotlib.lines.Line2D at 0x7fcc1d695a30>,
  <matplotlib.lines.Line2D at 0x7fcc1d695dc0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6ac3a0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6ac730>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc1d6ec520>,
  <matplotlib.lines.Line2D at 0x7fcc1d6ec8b0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6f7eb0>,
  <matplotlib.lines.Line2D at 0x7fcc1d701280>,
  <matplotlib.lines.Line2D at 0x7fcc1d70c820>,
  <matplotlib.lines.Line2D at 0x7fcc1d70cbb0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6a1190>,
  <matplotlib.lines.Line2D at 0x7fcc1d6a1520>,
  <matplotlib.lines.Line2D at 0x7fcc1d6acac0>,
  <matplotlib.lines.Line2D at 0x7fcc1d6ace50>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc1d6dba30>,
  <matplotlib.lines.Line2D at 0x7fcc1d6f7400>,
```

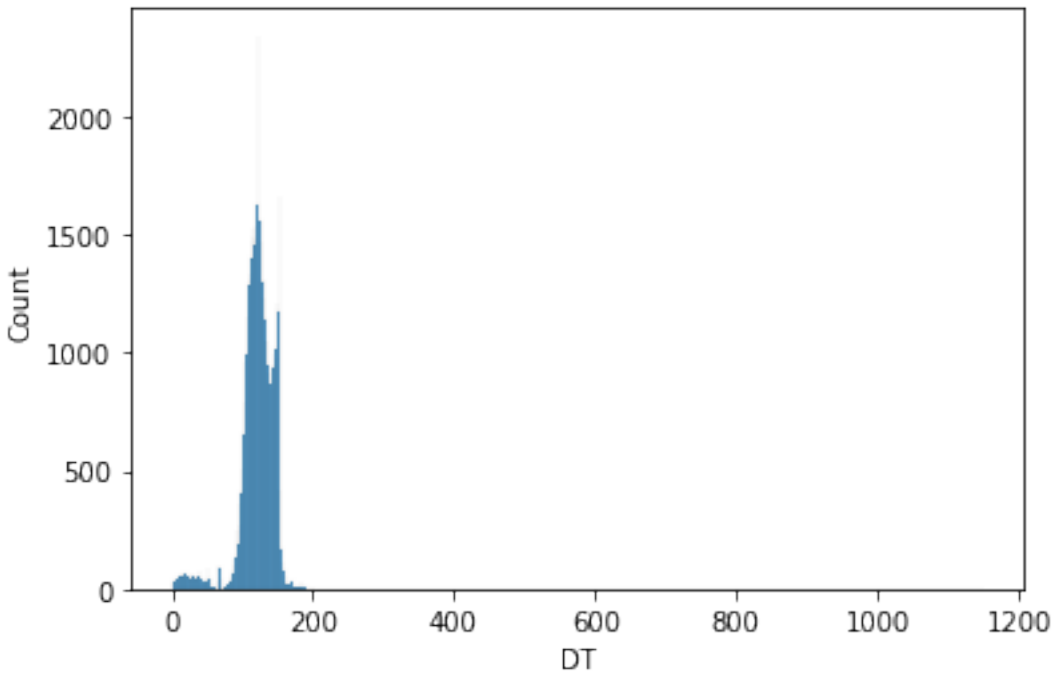
```
 <matplotlib.lines.Line2D at 0x7fcc1d701d30>,
 <matplotlib.lines.Line2D at 0x7fcc1d6956a0>,
 <matplotlib.lines.Line2D at 0x7fcc1d6a1fd0>],
'medians': [<matplotlib.lines.Line2D at 0x7fcc1d6ecc40>,
 <matplotlib.lines.Line2D at 0x7fcc1d701610>,
 <matplotlib.lines.Line2D at 0x7fcc1d70cf40>,
 <matplotlib.lines.Line2D at 0x7fcc1d6a18b0>,
 <matplotlib.lines.Line2D at 0x7fcc1d6b7220>],
'fliers': [<matplotlib.lines.Line2D at 0x7fcc1d6ecfd0>,
 <matplotlib.lines.Line2D at 0x7fcc1d7019a0>,
 <matplotlib.lines.Line2D at 0x7fcc1d695310>,
 <matplotlib.lines.Line2D at 0x7fcc1d6a1c40>,
 <matplotlib.lines.Line2D at 0x7fcc1d6b75b0>],
'means': []}
```



4.2 DT VISUALIZATION

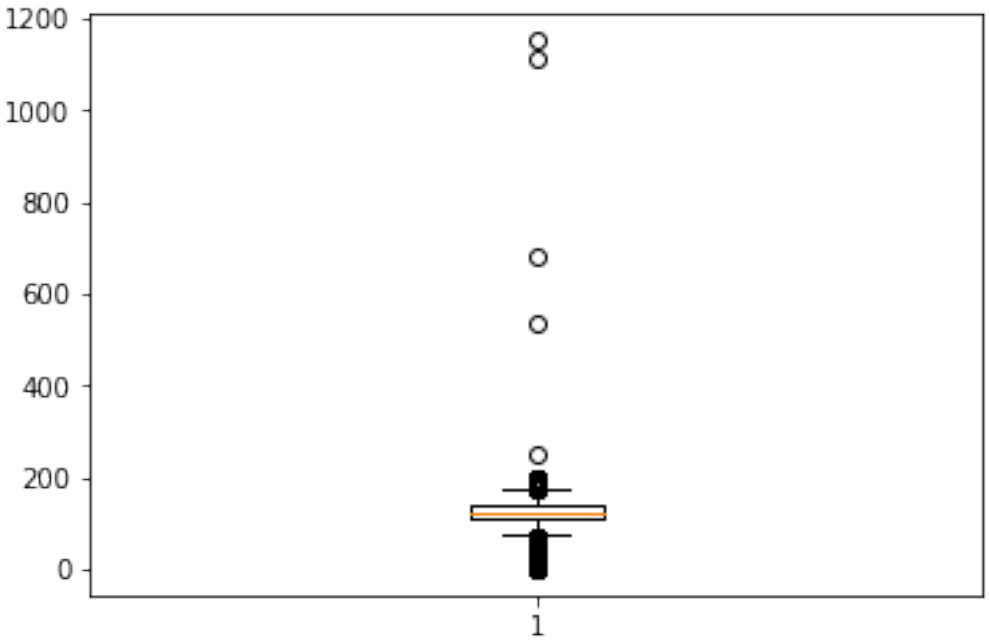
[50]:
```
sns.histplot(df.DT)
```

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



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

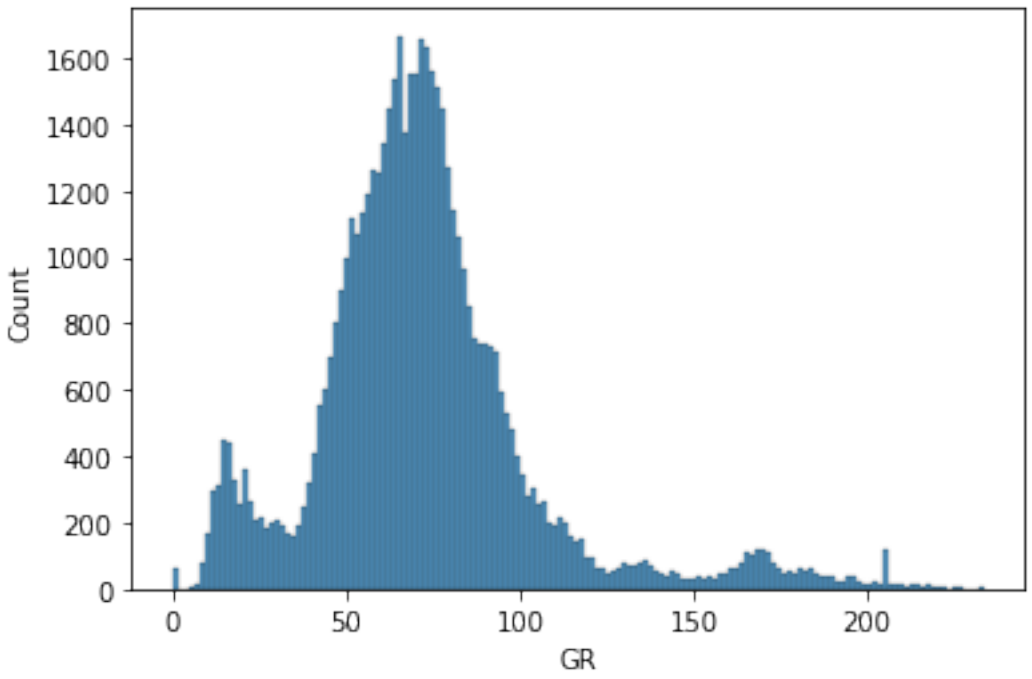
[51]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc10975880>,
  <matplotlib.lines.Line2D at 0x7fcc10975c10>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc10975fa0>,
  <matplotlib.lines.Line2D at 0x7fcc10981370>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc109754f0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc10981700>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc10981a90>],
 'means': []}
```



4.3 GR VISUALIZATION

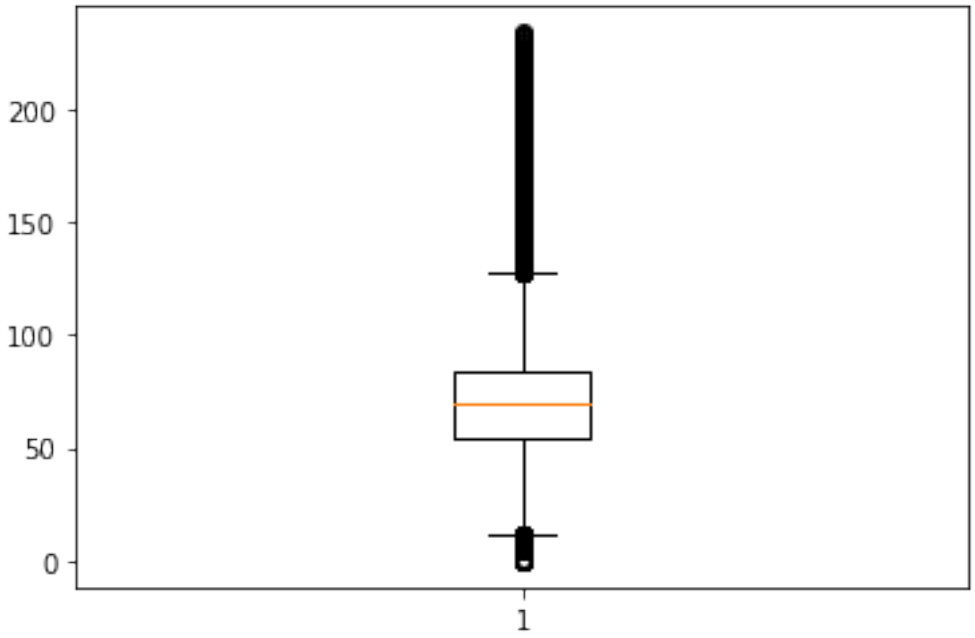
[52]:
```
sns.histplot(df.GR)
```

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



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

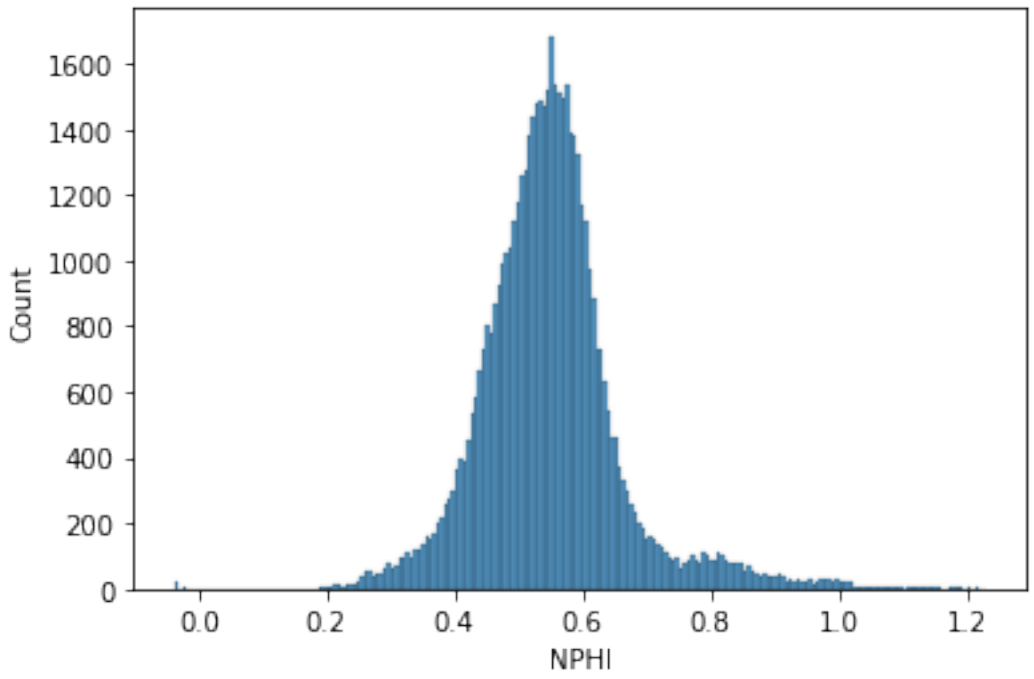
[53]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc107ba2e0>,
  <matplotlib.lines.Line2D at 0x7fcc107ba670>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc107baa00>,
  <matplotlib.lines.Line2D at 0x7fcc107bad90>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc107adf10>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc107c6160>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc107c64f0>],
 'means': []}
```



4.4 NPHI VISUALIZATION

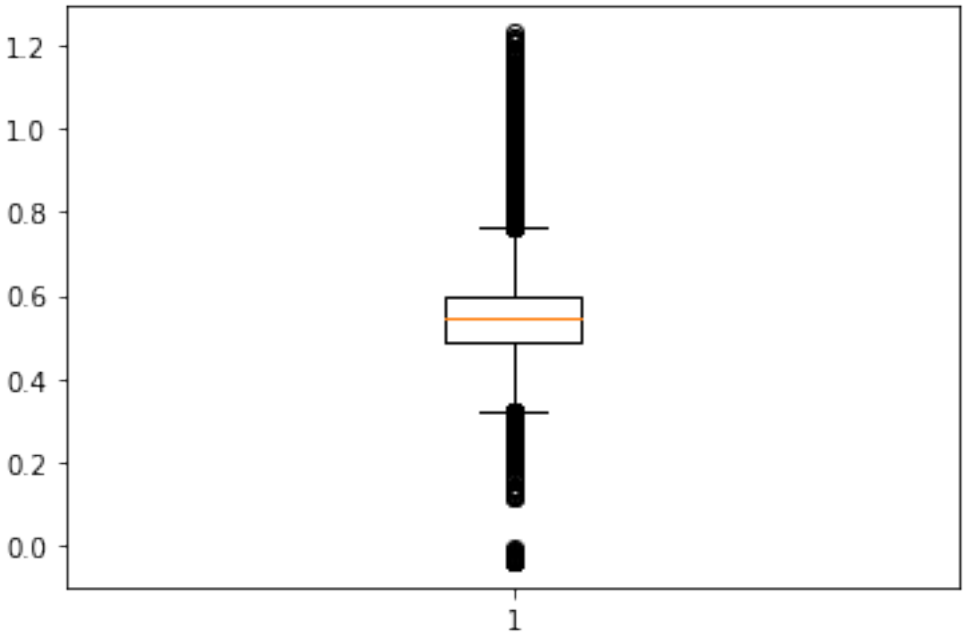
[54]:
```
sns.histplot(df.NPHI)
```

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



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

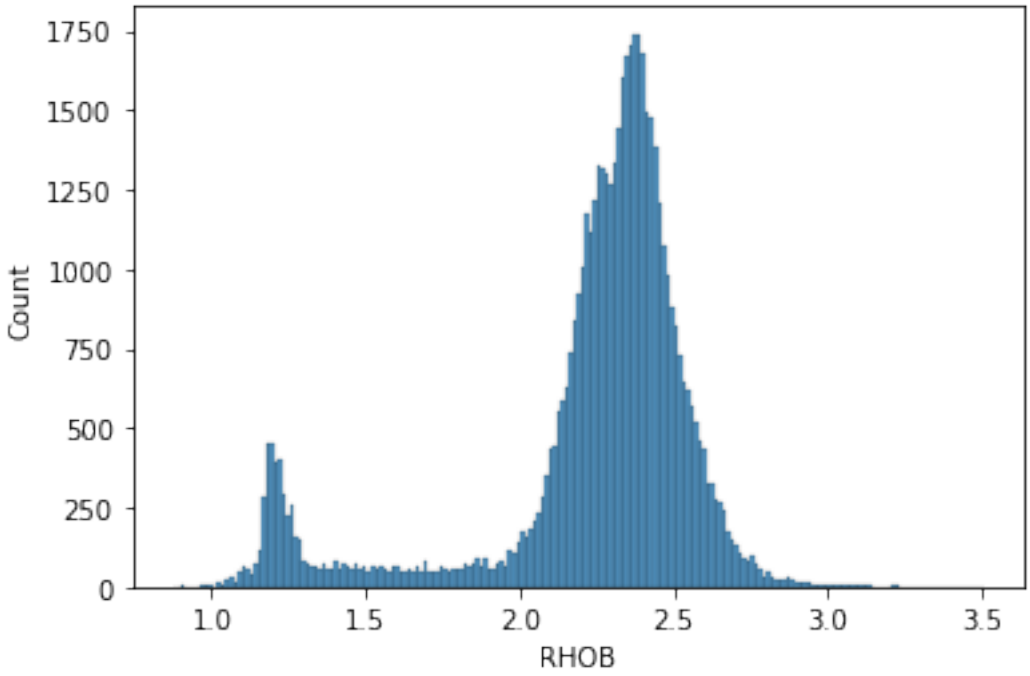
[55]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc10547040>,
  <matplotlib.lines.Line2D at 0x7fcc10547400>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc10547790>,
  <matplotlib.lines.Line2D at 0x7fcc10547b20>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc10538c70>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc10547eb0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc10554280>],
 'means': []}
```



4.5 RHOB VISUALIZATION

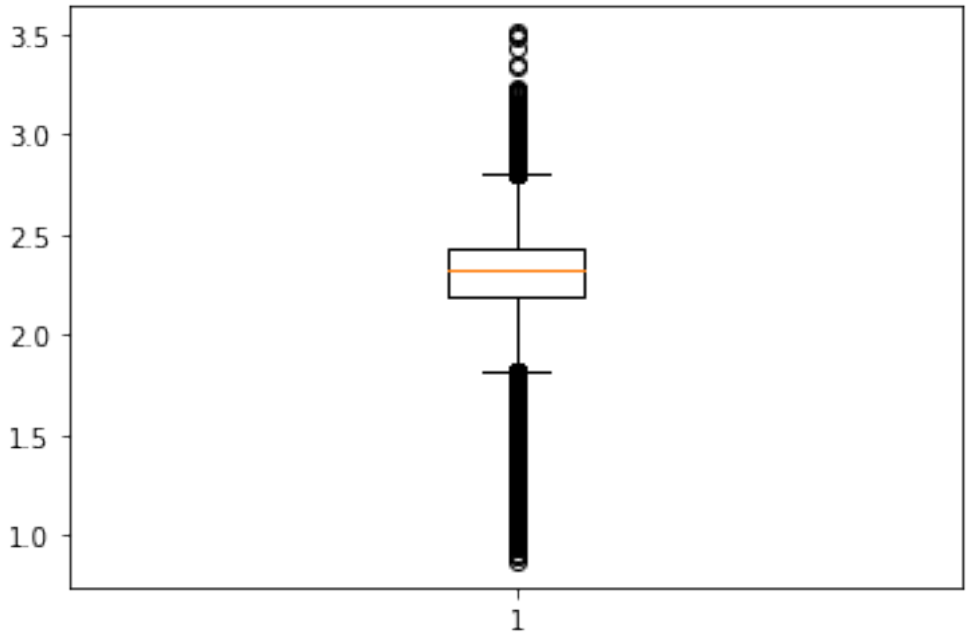
[56]:
```
sns.histplot(df.RHOB)
```

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



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

[57]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc1ebab730>,
  <matplotlib.lines.Line2D at 0x7fcc1eb81340>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc1eb81490>,
  <matplotlib.lines.Line2D at 0x7fcc1eb81670>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc1ebab310>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc1eb81eb0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc1ebcad60>],
 '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 = 1
df = outliers(DATAConditioningStrategy[optionoutlier] , df,␣
↪DATAConditioningColumns)
```

```
column DT
4 standard deviation outliers -:
              DT       GR    NPHI    RHOB  FACIES
282      15.8999  63.5563  0.3764  2.5182       0
```

```
283       8.6395  61.5439  0.3675  2.5916       0
284       3.1202  60.7632  0.3411  2.6241       0
285       4.3432  60.9371  0.3120  2.5905       0
286      10.5087  62.4516  0.3091  2.5213       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

[621 rows x 5 columns]
(621, 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

[52020 rows x 5 columns]
column GR
4 standard deviation outliers -:
             DT        GR     NPHI    RHOB  FACIES
35161  133.9539  207.7189  0.66280  2.3796       0
35162  134.4976  208.2332  0.66850  2.3742       0
35169  141.0677  205.0823  0.64220  2.3929       0
35170  130.4464  214.7148  0.56620  2.5062       0
35171  119.0552  219.3920  0.50910  2.6632       0
…          …       …       …       …      …
36494  125.9000  204.7348  0.55171  2.5171       0
36495  125.9000  204.7348  0.55171  2.5122       0
36496  125.9000  204.7348  0.55171  2.4951       0
36497  125.9000  204.7348  0.55171  2.4817       0
36498  125.9000  204.7348  0.55171  2.4790       0

[297 rows x 5 columns]
(297, 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

[51723 rows x 5 columns]
column NPHI
4 standard deviation outliers -:
             DT        GR    NPHI    RHOB  FACIES
3730   151.8671   12.5059  1.0006  1.1972       3
8414   150.7242   16.0597  1.0039  1.2529       3
8415   150.7014   16.0969  1.0071  1.2497       3
8416   150.4433   16.2727  1.0224  1.2348       3
8417   150.3887   16.3387  1.0202  1.2317       3
…         …       …      …       …      …
36120  159.6745  141.6185  1.0849  1.2767       3
36121  168.0660  141.8772  1.0714  1.3280       3
36225  153.0819  122.2618  1.0099  1.1822       3
36229  151.5947  121.3572  1.0042  1.1892       3
48606  113.3730   63.3097  0.9980  2.6148       0

[290 rows x 5 columns]
(290, 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

[51433 rows x 5 columns]
column RHOB
4 standard deviation outliers -:
Empty DataFrame
Columns: [DT, GR, NPHI, RHOB, FACIES]
Index: []
(0, 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

[51433 rows x 5 columns]
```

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

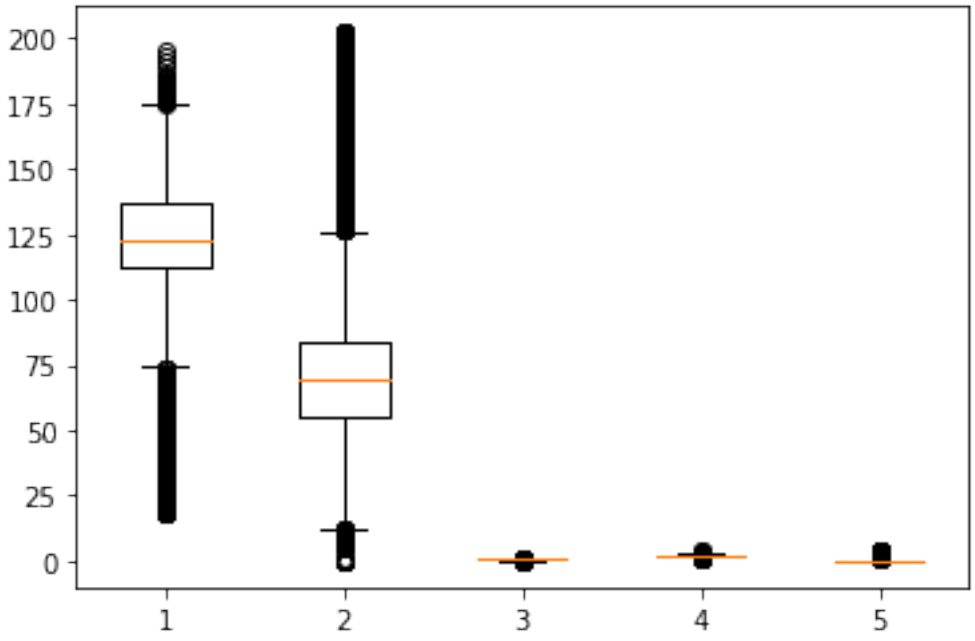
[60]:
```
(51433, 5)
```

4.6 WHOLE DATA AFTER REMOVING OUTLIERS

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

[61]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc1eb967c0>,
  <matplotlib.lines.Line2D at 0x7fcc105a9ac0>,
  <matplotlib.lines.Line2D at 0x7fcc1ebb87c0>,
  <matplotlib.lines.Line2D at 0x7fcc1ebb82e0>,
  <matplotlib.lines.Line2D at 0x7fcc1ec44a90>,
  <matplotlib.lines.Line2D at 0x7fcc1ec44d00>,
  <matplotlib.lines.Line2D at 0x7fcc1eff0d30>,
  <matplotlib.lines.Line2D at 0x7fcc1f2c3760>,
  <matplotlib.lines.Line2D at 0x7fcc1eff8970>,
  <matplotlib.lines.Line2D at 0x7fcc1eff8d00>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc10647220>,
  <matplotlib.lines.Line2D at 0x7fcc1ebca250>,
  <matplotlib.lines.Line2D at 0x7fcc1ec3a970>,
  <matplotlib.lines.Line2D at 0x7fcc1ec3adf0>,
  <matplotlib.lines.Line2D at 0x7fcc1ec44e50>,
  <matplotlib.lines.Line2D at 0x7fcc1ec74e80>,
  <matplotlib.lines.Line2D at 0x7fcc1f2c3f40>,
  <matplotlib.lines.Line2D at 0x7fcc1f2c3700>,
  <matplotlib.lines.Line2D at 0x7fcc037af0d0>,
  <matplotlib.lines.Line2D at 0x7fcc037af460>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc1eb96790>,
  <matplotlib.lines.Line2D at 0x7fcc1ebb8400>,
  <matplotlib.lines.Line2D at 0x7fcc1ec441c0>,
  <matplotlib.lines.Line2D at 0x7fcc1eff0160>,
  <matplotlib.lines.Line2D at 0x7fcc1eff85e0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc1efe9370>,
```

```
 <matplotlib.lines.Line2D at 0x7fcc1ec3a460>,
 <matplotlib.lines.Line2D at 0x7fcc1eff0700>,
 <matplotlib.lines.Line2D at 0x7fcc1f2c3df0>,
 <matplotlib.lines.Line2D at 0x7fcc037af7f0>],
'fliers': [<matplotlib.lines.Line2D at 0x7fcc1ebb8520>,
 <matplotlib.lines.Line2D at 0x7fcc1f2c9430>,
 <matplotlib.lines.Line2D at 0x7fcc1eff0e20>,
 <matplotlib.lines.Line2D at 0x7fcc1eff8250>,
 <matplotlib.lines.Line2D at 0x7fcc037afb80>],
'means': []}
```



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

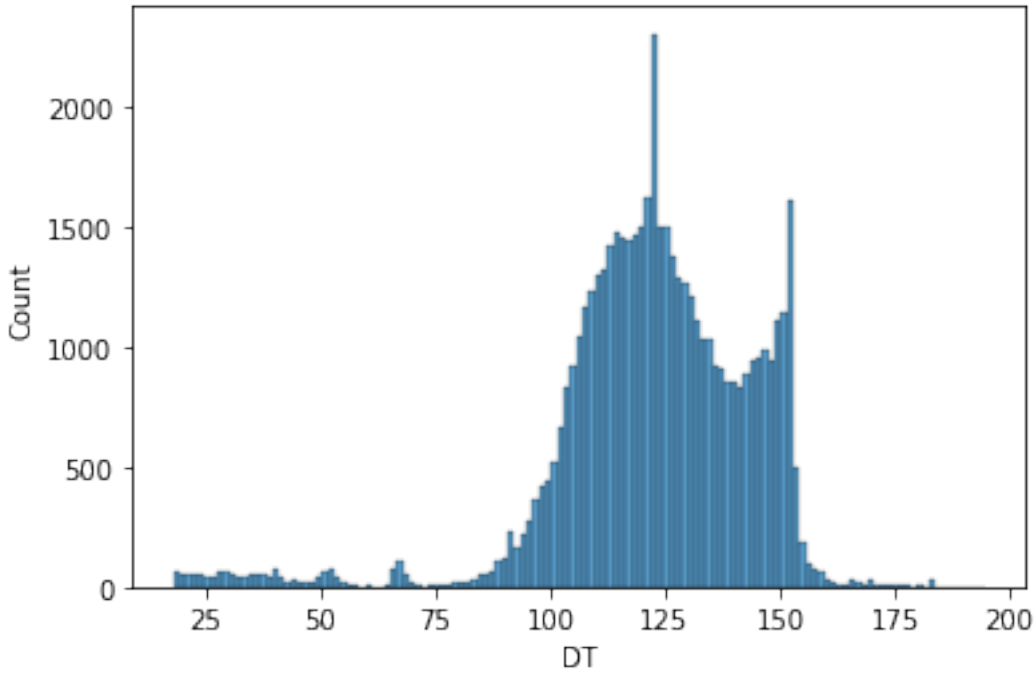
[62]:

	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

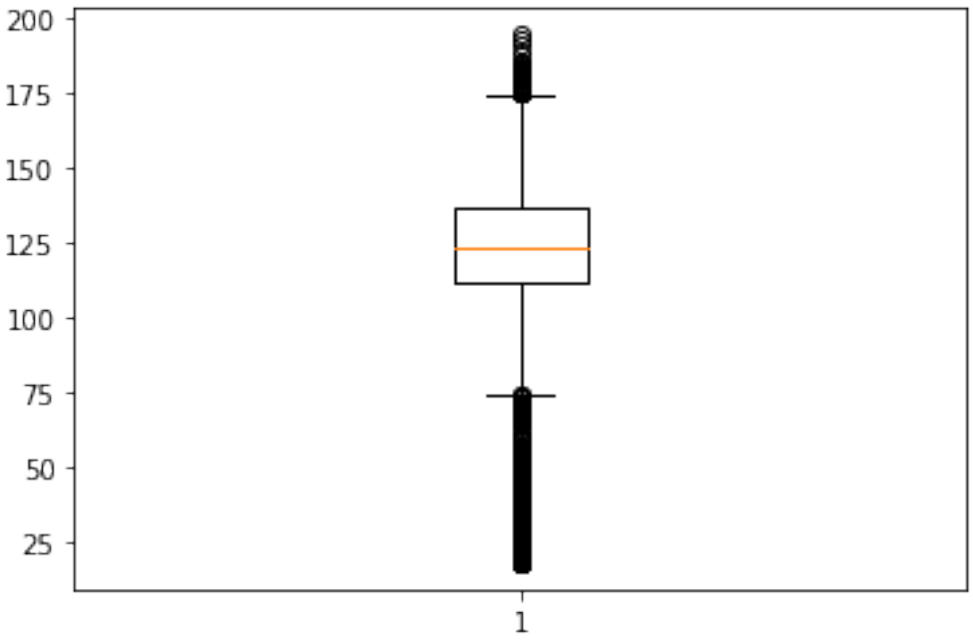
[63]:
```
sns.histplot(df.DT)
```

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



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

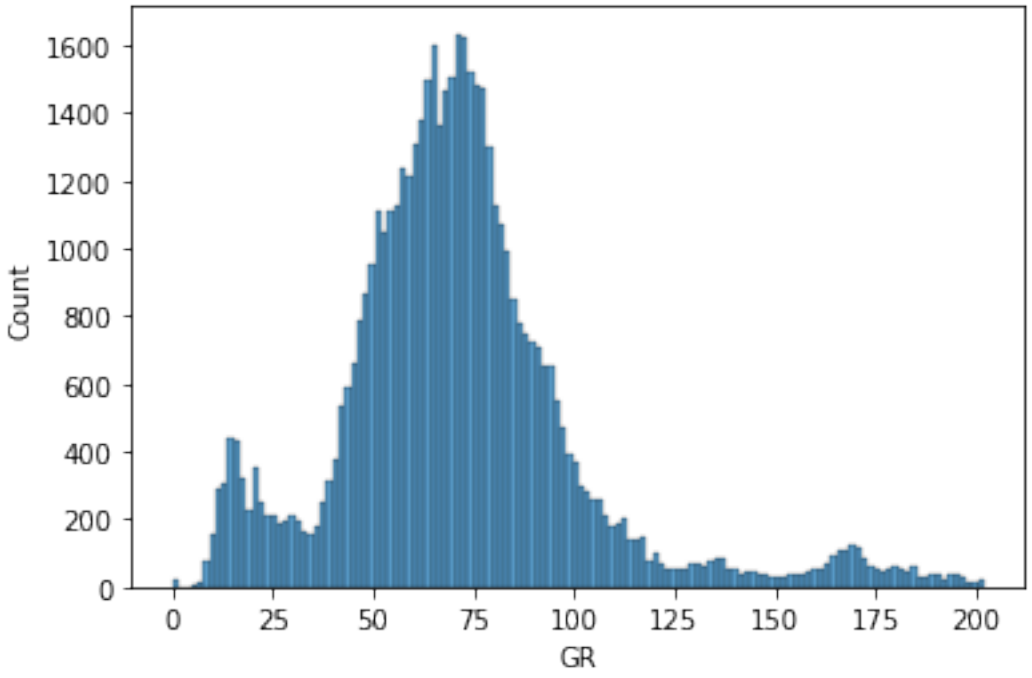
[64]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc1ec17f40>,
  <matplotlib.lines.Line2D at 0x7fcc1ebfc310>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc1ebfc6a0>,
  <matplotlib.lines.Line2D at 0x7fcc1ebfca30>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc1ec17bb0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc1ebfcdc0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc1ebf4190>],
 'means': []}
```



4.8 GR AFTER REMOVING OUTLIER

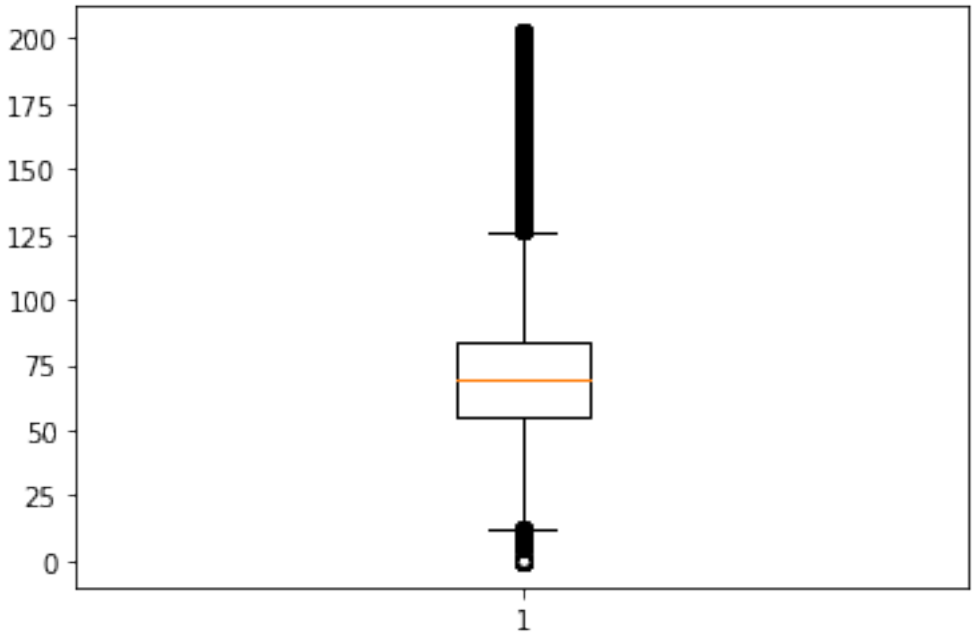
[65]:
```
sns.histplot(df.GR)
```

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



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

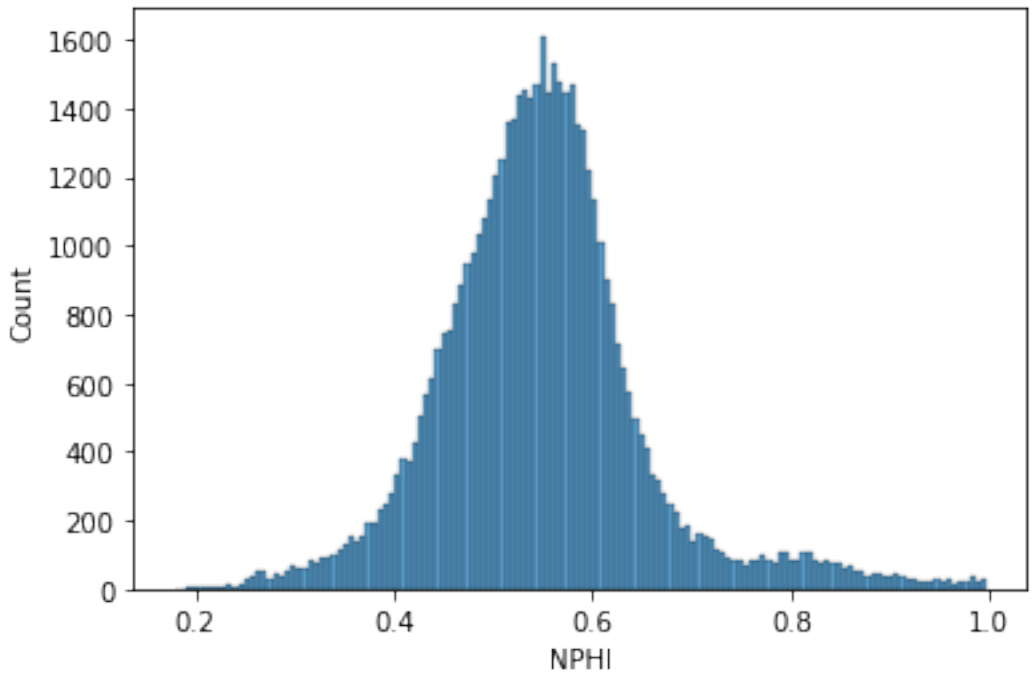
[66]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc10227dc0>,
  <matplotlib.lines.Line2D at 0x7fcc10235190>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc10235520>,
  <matplotlib.lines.Line2D at 0x7fcc102358b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc10227a30>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc10235c40>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc10235fd0>],
 'means': []}
```



4.9 NPHI AFTER REMOVING OUTLIER

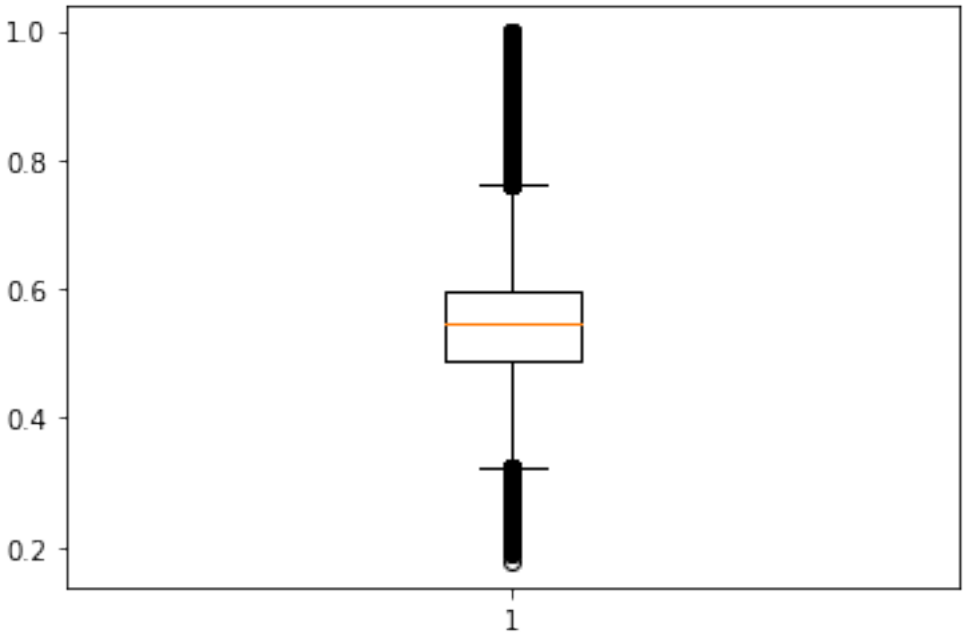
[67]:
```
sns.histplot(df.NPHI)
```

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



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

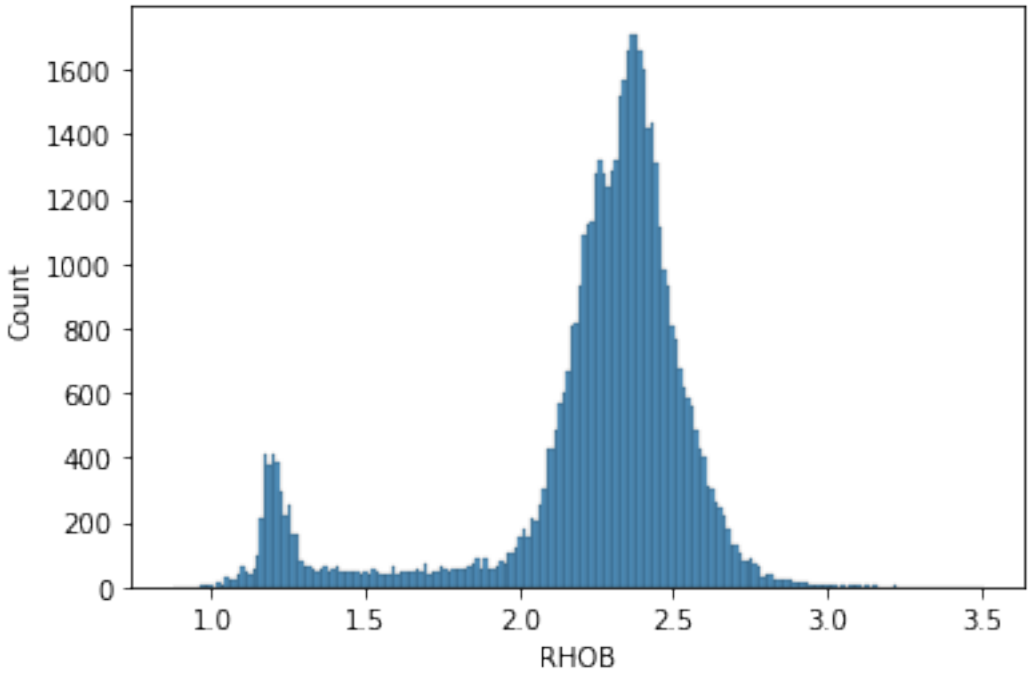
[68]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc10023b50>,
  <matplotlib.lines.Line2D at 0x7fcc10023ee0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc100302b0>,
  <matplotlib.lines.Line2D at 0x7fcc10030640>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc100237c0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc100309d0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc10030d60>],
 'means': []}
```



4.10 RHOB AFTER REMOVING OUTLIER

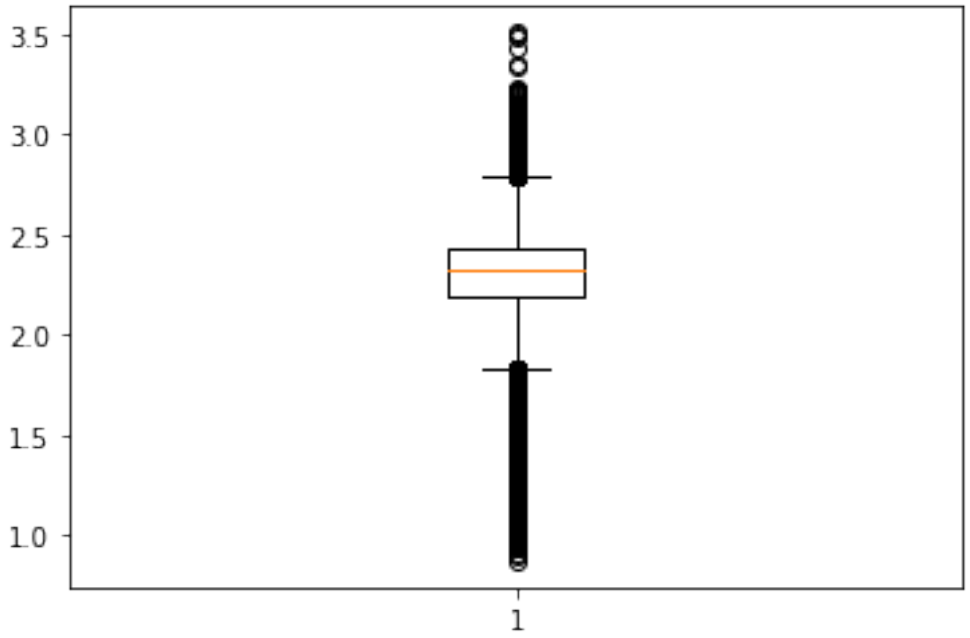
[69]:
```
sns.histplot(df.RHOB)
```

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



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

[70]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcc0fddbe50>,
  <matplotlib.lines.Line2D at 0x7fcc0fd6c220>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcc0fd6c5b0>,
  <matplotlib.lines.Line2D at 0x7fcc0fd6c940>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcc0fddbac0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcc0fd6ccd0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcc0fd760a0>],
 '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

[51433 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
9  40.9733  79.0259  0.4829  2.2439       0
```

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

[73]:
```
(51433, 5)
```

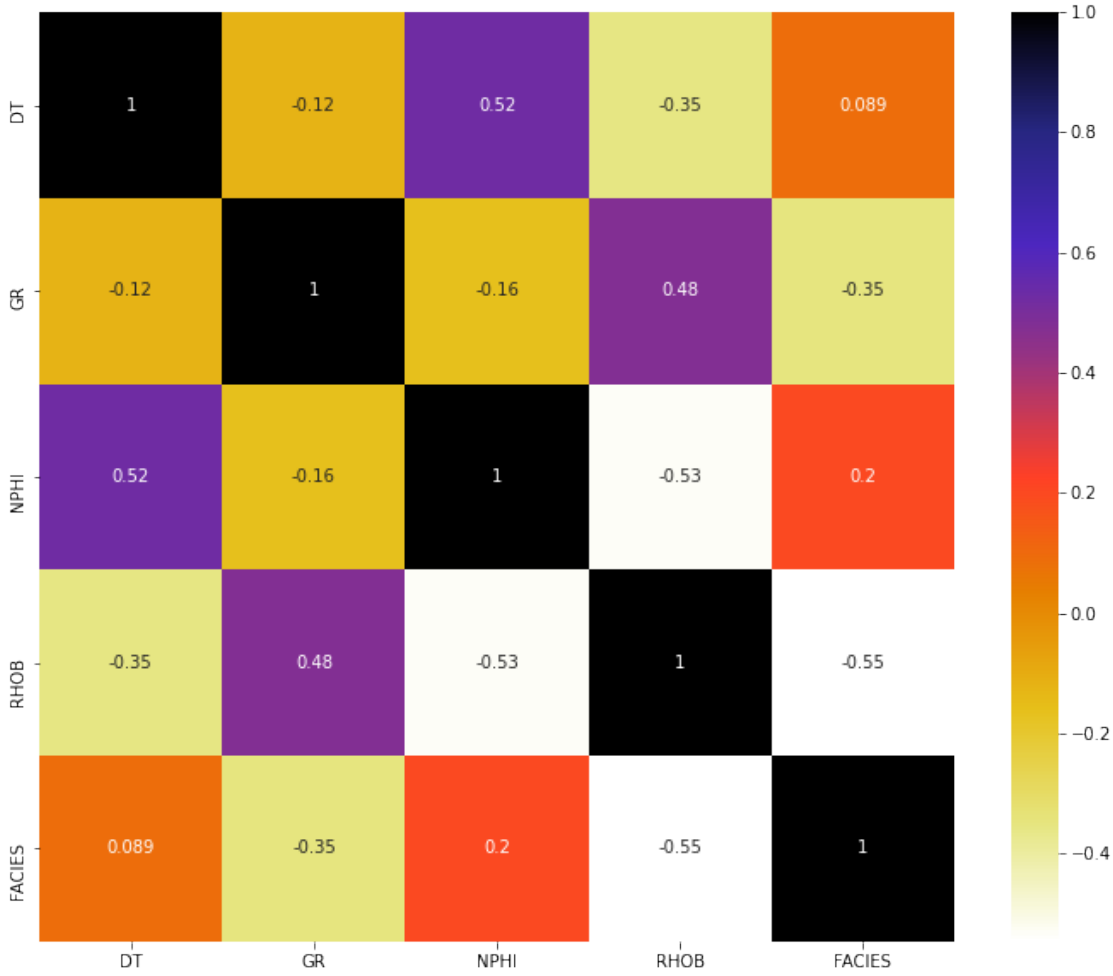
[74]:
```
features = df.shape[1]
features
```

[74]:
```
5
```

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

[75]:
```
DT        481.844221
GR        941.259816
NPHI        0.010919
RHOB        0.127575
FACIES      1.588386
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 = 2
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

[51433 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

[51433 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

[51433 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 = 1
df = data_scaling( scaling_strategy[optionscaling] , df ,␣
↪DATAConditioningColumns )
```

[85]:
```
df
```

[85]:
```
             DT        GR      NPHI      RHOB  FACIES
0      0.192151  0.333301  0.416178  0.488967       0
1      0.178847  0.342466  0.413856  0.488929       0
2      0.172715  0.347689  0.404203  0.488323       0
3      0.174447  0.354303  0.389541  0.491924       0
4      0.175966  0.359123  0.375489  0.498332       0
…           …         …         …         …       …
52636  0.513872  0.369503  0.336632  0.702646       0
52637  0.520121  0.356194  0.337488  0.687291       0
52638  0.542136  0.366741  0.365836  0.659615       0
52639  0.577473  0.390473  0.391618  0.644525       0
52640  0.594446  0.410043  0.409580  0.651729       0

[51433 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	0.192151	0.333301	0.416178	0.488967	0
1	0.178847	0.342466	0.413856	0.488929	0
2	0.172715	0.347689	0.404203	0.488323	0
3	0.174447	0.354303	0.389541	0.491924	0
4	0.175966	0.359123	0.375489	0.498332	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]:
```
(36003, 4)
```

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

[92]:
```
(15430, 4)
```

[93]:
```
X_test
```

[93]:

	DT	GR	NPHI	RHOB
8622	0.622591	0.477044	0.521139	0.576661
7270	0.503449	0.356492	0.353250	0.632772
20177	0.677466	0.302431	0.484115	0.529497
10340	0.677306	0.219004	0.559995	0.496550
22734	0.590451	0.455214	0.416422	0.591295
…	…	…	…	…
26611	0.544721	0.353916	0.500611	0.589399

```
25612  0.738721  0.224478  0.573069  0.523203
14509  0.727258  0.215182  0.472263  0.502047
14525  0.697082  0.214862  0.457967  0.477897
41823  0.617230  0.443041  0.362292  0.563353

[15430 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: 0.885343 using {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.885278 (0.002957) with: {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.885306 (0.002868) with: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
0.883917 (0.002993) with: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
0.885343 (0.002967) with: {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.885287 (0.002981) with: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
0.883676 (0.002963) with: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
0.885232 (0.003117) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'newton-cg'}
0.885241 (0.003132) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
0.881982 (0.002768) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}
0.878353 (0.002869) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
0.878353 (0.002869) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
0.869122 (0.002366) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
0.862586 (0.003053) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
0.862586 (0.003052) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
0.858206 (0.003118) 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 0.094994
0:      learn: 1.3334567        total: 55.7ms   remaining: 55.6s
1:      learn: 1.1548076        total: 62.6ms   remaining: 31.2s
2:      learn: 1.0221565        total: 69.4ms   remaining: 23.1s
3:      learn: 0.9209659        total: 76.3ms   remaining: 19s
4:      learn: 0.8386240        total: 82.8ms   remaining: 16.5s
5:      learn: 0.7710541        total: 90.1ms   remaining: 14.9s
6:      learn: 0.7147399        total: 97.8ms   remaining: 13.9s
7:      learn: 0.6664077        total: 106ms    remaining: 13.1s
8:      learn: 0.6257331        total: 113ms    remaining: 12.4s
9:      learn: 0.5908388        total: 121ms    remaining: 12s
10:     learn: 0.5608992        total: 128ms    remaining: 11.5s
11:     learn: 0.5339481        total: 136ms    remaining: 11.2s
12:     learn: 0.5116344        total: 143ms    remaining: 10.8s
13:     learn: 0.4914901        total: 150ms    remaining: 10.5s
14:     learn: 0.4743142        total: 157ms    remaining: 10.3s
```

```
15:     learn: 0.4577681        total: 165ms    remaining: 10.1s
16:     learn: 0.4427327        total: 173ms    remaining: 10s
17:     learn: 0.4305088        total: 180ms    remaining: 9.85s
18:     learn: 0.4191623        total: 188ms    remaining: 9.73s
19:     learn: 0.4083696        total: 196ms    remaining: 9.61s
20:     learn: 0.3991839        total: 204ms    remaining: 9.52s
21:     learn: 0.3911887        total: 212ms    remaining: 9.41s
22:     learn: 0.3834359        total: 219ms    remaining: 9.32s
23:     learn: 0.3768863        total: 227ms    remaining: 9.22s
24:     learn: 0.3712931        total: 235ms    remaining: 9.15s
25:     learn: 0.3659212        total: 242ms    remaining: 9.07s
26:     learn: 0.3604482        total: 250ms    remaining: 9s
27:     learn: 0.3556547        total: 257ms    remaining: 8.93s
28:     learn: 0.3511876        total: 268ms    remaining: 8.96s
29:     learn: 0.3479838        total: 274ms    remaining: 8.87s
30:     learn: 0.3456403        total: 281ms    remaining: 8.79s
31:     learn: 0.3422390        total: 289ms    remaining: 8.75s
32:     learn: 0.3392140        total: 298ms    remaining: 8.72s
33:     learn: 0.3366596        total: 306ms    remaining: 8.68s
34:     learn: 0.3342812        total: 313ms    remaining: 8.63s
35:     learn: 0.3317687        total: 320ms    remaining: 8.56s
36:     learn: 0.3291419        total: 327ms    remaining: 8.52s
37:     learn: 0.3270984        total: 334ms    remaining: 8.46s
38:     learn: 0.3252649        total: 342ms    remaining: 8.42s
39:     learn: 0.3237038        total: 349ms    remaining: 8.37s
40:     learn: 0.3219873        total: 357ms    remaining: 8.36s
41:     learn: 0.3203402        total: 364ms    remaining: 8.31s
42:     learn: 0.3189985        total: 372ms    remaining: 8.28s
43:     learn: 0.3175724        total: 380ms    remaining: 8.25s
44:     learn: 0.3161081        total: 388ms    remaining: 8.23s
45:     learn: 0.3150233        total: 396ms    remaining: 8.21s
46:     learn: 0.3135736        total: 404ms    remaining: 8.2s
47:     learn: 0.3123150        total: 412ms    remaining: 8.17s
48:     learn: 0.3110588        total: 419ms    remaining: 8.13s
49:     learn: 0.3101186        total: 427ms    remaining: 8.12s
50:     learn: 0.3086921        total: 436ms    remaining: 8.11s
51:     learn: 0.3079511        total: 444ms    remaining: 8.09s
52:     learn: 0.3069847        total: 453ms    remaining: 8.09s
53:     learn: 0.3062509        total: 460ms    remaining: 8.05s
54:     learn: 0.3052776        total: 467ms    remaining: 8.02s
55:     learn: 0.3045688        total: 475ms    remaining: 8s
56:     learn: 0.3036410        total: 483ms    remaining: 7.98s
57:     learn: 0.3028042        total: 490ms    remaining: 7.96s
58:     learn: 0.3021499        total: 498ms    remaining: 7.94s
59:     learn: 0.3016536        total: 504ms    remaining: 7.9s
60:     learn: 0.3007801        total: 511ms    remaining: 7.87s
61:     learn: 0.3001566        total: 518ms    remaining: 7.84s
62:     learn: 0.2993483        total: 525ms    remaining: 7.82s
```

```
63:     learn: 0.2987373        total: 533ms    remaining: 7.79s
64:     learn: 0.2982026        total: 540ms    remaining: 7.77s
65:     learn: 0.2972011        total: 549ms    remaining: 7.77s
66:     learn: 0.2966630        total: 556ms    remaining: 7.74s
67:     learn: 0.2962223        total: 563ms    remaining: 7.72s
68:     learn: 0.2956946        total: 571ms    remaining: 7.71s
69:     learn: 0.2952593        total: 579ms    remaining: 7.7s
70:     learn: 0.2947784        total: 586ms    remaining: 7.67s
71:     learn: 0.2939334        total: 595ms    remaining: 7.67s
72:     learn: 0.2935654        total: 602ms    remaining: 7.64s
73:     learn: 0.2930070        total: 612ms    remaining: 7.66s
74:     learn: 0.2925582        total: 619ms    remaining: 7.63s
75:     learn: 0.2921213        total: 626ms    remaining: 7.61s
76:     learn: 0.2917302        total: 633ms    remaining: 7.58s
77:     learn: 0.2912622        total: 641ms    remaining: 7.58s
78:     learn: 0.2909641        total: 648ms    remaining: 7.56s
79:     learn: 0.2905719        total: 655ms    remaining: 7.53s
80:     learn: 0.2901079        total: 662ms    remaining: 7.51s
81:     learn: 0.2898252        total: 669ms    remaining: 7.49s
82:     learn: 0.2895368        total: 676ms    remaining: 7.47s
83:     learn: 0.2890615        total: 683ms    remaining: 7.45s
84:     learn: 0.2887422        total: 691ms    remaining: 7.44s
85:     learn: 0.2884230        total: 698ms    remaining: 7.41s
86:     learn: 0.2879047        total: 705ms    remaining: 7.4s
87:     learn: 0.2874844        total: 712ms    remaining: 7.38s
88:     learn: 0.2870105        total: 720ms    remaining: 7.36s
89:     learn: 0.2865813        total: 726ms    remaining: 7.34s
90:     learn: 0.2863211        total: 733ms    remaining: 7.33s
91:     learn: 0.2859928        total: 741ms    remaining: 7.31s
92:     learn: 0.2857466        total: 747ms    remaining: 7.29s
93:     learn: 0.2855270        total: 754ms    remaining: 7.27s
94:     learn: 0.2848652        total: 762ms    remaining: 7.26s
95:     learn: 0.2846690        total: 770ms    remaining: 7.25s
96:     learn: 0.2843278        total: 778ms    remaining: 7.24s
97:     learn: 0.2841047        total: 785ms    remaining: 7.23s
98:     learn: 0.2837651        total: 792ms    remaining: 7.21s
99:     learn: 0.2833629        total: 803ms    remaining: 7.23s
100:    learn: 0.2831726        total: 809ms    remaining: 7.2s
101:    learn: 0.2827852        total: 817ms    remaining: 7.19s
102:    learn: 0.2823706        total: 824ms    remaining: 7.17s
103:    learn: 0.2822369        total: 832ms    remaining: 7.16s
104:    learn: 0.2819985        total: 838ms    remaining: 7.14s
105:    learn: 0.2814180        total: 846ms    remaining: 7.13s
106:    learn: 0.2810455        total: 852ms    remaining: 7.11s
107:    learn: 0.2807123        total: 860ms    remaining: 7.1s
108:    learn: 0.2803951        total: 867ms    remaining: 7.09s
109:    learn: 0.2801173        total: 875ms    remaining: 7.08s
110:    learn: 0.2799338        total: 882ms    remaining: 7.07s
```

```
111:    learn: 0.2795954        total: 889ms    remaining: 7.05s
112:    learn: 0.2792490        total: 897ms    remaining: 7.04s
113:    learn: 0.2791115        total: 903ms    remaining: 7.02s
114:    learn: 0.2790104        total: 911ms    remaining: 7.01s
115:    learn: 0.2787844        total: 917ms    remaining: 6.99s
116:    learn: 0.2785165        total: 925ms    remaining: 6.98s
117:    learn: 0.2782976        total: 932ms    remaining: 6.96s
118:    learn: 0.2781089        total: 940ms    remaining: 6.96s
119:    learn: 0.2777836        total: 947ms    remaining: 6.94s
120:    learn: 0.2776591        total: 954ms    remaining: 6.93s
121:    learn: 0.2774554        total: 961ms    remaining: 6.92s
122:    learn: 0.2771843        total: 969ms    remaining: 6.91s
123:    learn: 0.2769292        total: 977ms    remaining: 6.9s
124:    learn: 0.2767820        total: 984ms    remaining: 6.89s
125:    learn: 0.2765626        total: 992ms    remaining: 6.88s
126:    learn: 0.2762572        total: 999ms    remaining: 6.86s
127:    learn: 0.2761535        total: 1s       remaining: 6.85s
128:    learn: 0.2760040        total: 1.01s    remaining: 6.84s
129:    learn: 0.2756919        total: 1.02s    remaining: 6.83s
130:    learn: 0.2755119        total: 1.03s    remaining: 6.82s
131:    learn: 0.2753191        total: 1.03s    remaining: 6.8s
132:    learn: 0.2746323        total: 1.04s    remaining: 6.8s
133:    learn: 0.2743807        total: 1.05s    remaining: 6.79s
134:    learn: 0.2741562        total: 1.06s    remaining: 6.78s
135:    learn: 0.2740390        total: 1.06s    remaining: 6.77s
136:    learn: 0.2738190        total: 1.07s    remaining: 6.76s
137:    learn: 0.2737078        total: 1.08s    remaining: 6.75s
138:    learn: 0.2734538        total: 1.09s    remaining: 6.74s
139:    learn: 0.2731027        total: 1.09s    remaining: 6.73s
140:    learn: 0.2726115        total: 1.1s     remaining: 6.72s
141:    learn: 0.2723242        total: 1.11s    remaining: 6.71s
142:    learn: 0.2721948        total: 1.12s    remaining: 6.7s
143:    learn: 0.2720853        total: 1.13s    remaining: 6.69s
144:    learn: 0.2717101        total: 1.13s    remaining: 6.68s
145:    learn: 0.2714010        total: 1.14s    remaining: 6.67s
146:    learn: 0.2711495        total: 1.15s    remaining: 6.67s
147:    learn: 0.2709211        total: 1.16s    remaining: 6.66s
148:    learn: 0.2707328        total: 1.17s    remaining: 6.66s
149:    learn: 0.2706042        total: 1.17s    remaining: 6.65s
150:    learn: 0.2703143        total: 1.18s    remaining: 6.64s
151:    learn: 0.2700019        total: 1.19s    remaining: 6.63s
152:    learn: 0.2699368        total: 1.2s     remaining: 6.62s
153:    learn: 0.2697146        total: 1.2s     remaining: 6.61s
154:    learn: 0.2696512        total: 1.21s    remaining: 6.6s
155:    learn: 0.2694553        total: 1.22s    remaining: 6.59s
156:    learn: 0.2693187        total: 1.23s    remaining: 6.58s
157:    learn: 0.2692006        total: 1.23s    remaining: 6.56s
158:    learn: 0.2689792        total: 1.24s    remaining: 6.55s
```

```
159:    learn: 0.2688524        total: 1.25s    remaining: 6.54s
160:    learn: 0.2686887        total: 1.25s    remaining: 6.52s
161:    learn: 0.2684675        total: 1.26s    remaining: 6.52s
162:    learn: 0.2683252        total: 1.27s    remaining: 6.5s
163:    learn: 0.2681866        total: 1.27s    remaining: 6.49s
164:    learn: 0.2679448        total: 1.28s    remaining: 6.48s
165:    learn: 0.2677041        total: 1.29s    remaining: 6.47s
166:    learn: 0.2674756        total: 1.29s    remaining: 6.46s
167:    learn: 0.2673706        total: 1.3s     remaining: 6.45s
168:    learn: 0.2671470        total: 1.31s    remaining: 6.44s
169:    learn: 0.2670737        total: 1.31s    remaining: 6.42s
170:    learn: 0.2668053        total: 1.32s    remaining: 6.41s
171:    learn: 0.2666185        total: 1.33s    remaining: 6.4s
172:    learn: 0.2665573        total: 1.34s    remaining: 6.39s
173:    learn: 0.2663838        total: 1.34s    remaining: 6.38s
174:    learn: 0.2663025        total: 1.35s    remaining: 6.37s
175:    learn: 0.2661891        total: 1.36s    remaining: 6.36s
176:    learn: 0.2661214        total: 1.37s    remaining: 6.36s
177:    learn: 0.2659887        total: 1.38s    remaining: 6.35s
178:    learn: 0.2657668        total: 1.38s    remaining: 6.34s
179:    learn: 0.2656455        total: 1.39s    remaining: 6.33s
180:    learn: 0.2653584        total: 1.4s     remaining: 6.32s
181:    learn: 0.2651566        total: 1.4s     remaining: 6.31s
182:    learn: 0.2649469        total: 1.41s    remaining: 6.31s
183:    learn: 0.2647929        total: 1.42s    remaining: 6.3s
184:    learn: 0.2645755        total: 1.43s    remaining: 6.29s
185:    learn: 0.2644158        total: 1.43s    remaining: 6.28s
186:    learn: 0.2642309        total: 1.44s    remaining: 6.26s
187:    learn: 0.2640305        total: 1.45s    remaining: 6.26s
188:    learn: 0.2639209        total: 1.46s    remaining: 6.24s
189:    learn: 0.2637357        total: 1.46s    remaining: 6.24s
190:    learn: 0.2636666        total: 1.47s    remaining: 6.23s
191:    learn: 0.2634482        total: 1.48s    remaining: 6.22s
192:    learn: 0.2632461        total: 1.48s    remaining: 6.21s
193:    learn: 0.2630561        total: 1.49s    remaining: 6.2s
194:    learn: 0.2628950        total: 1.5s     remaining: 6.19s
195:    learn: 0.2627634        total: 1.51s    remaining: 6.18s
196:    learn: 0.2626052        total: 1.51s    remaining: 6.17s
197:    learn: 0.2625535        total: 1.52s    remaining: 6.16s
198:    learn: 0.2623547        total: 1.53s    remaining: 6.15s
199:    learn: 0.2621337        total: 1.53s    remaining: 6.14s
200:    learn: 0.2619295        total: 1.54s    remaining: 6.13s
201:    learn: 0.2617128        total: 1.55s    remaining: 6.13s
202:    learn: 0.2615307        total: 1.56s    remaining: 6.12s
203:    learn: 0.2613348        total: 1.56s    remaining: 6.11s
204:    learn: 0.2612614        total: 1.57s    remaining: 6.1s
205:    learn: 0.2610322        total: 1.58s    remaining: 6.09s
206:    learn: 0.2608341        total: 1.59s    remaining: 6.08s
```

```
207:    learn: 0.2606429        total: 1.59s    remaining: 6.07s
208:    learn: 0.2605507        total: 1.6s     remaining: 6.06s
209:    learn: 0.2603984        total: 1.61s    remaining: 6.05s
210:    learn: 0.2602309        total: 1.61s    remaining: 6.04s
211:    learn: 0.2599966        total: 1.62s    remaining: 6.03s
212:    learn: 0.2599537        total: 1.63s    remaining: 6.02s
213:    learn: 0.2597431        total: 1.64s    remaining: 6.01s
214:    learn: 0.2596399        total: 1.64s    remaining: 6s
215:    learn: 0.2595809        total: 1.65s    remaining: 5.99s
216:    learn: 0.2594158        total: 1.66s    remaining: 5.98s
217:    learn: 0.2592828        total: 1.66s    remaining: 5.97s
218:    learn: 0.2592177        total: 1.67s    remaining: 5.96s
219:    learn: 0.2591094        total: 1.68s    remaining: 5.95s
220:    learn: 0.2587730        total: 1.68s    remaining: 5.94s
221:    learn: 0.2586245        total: 1.69s    remaining: 5.93s
222:    learn: 0.2585554        total: 1.7s     remaining: 5.92s
223:    learn: 0.2584303        total: 1.71s    remaining: 5.91s
224:    learn: 0.2583277        total: 1.71s    remaining: 5.9s
225:    learn: 0.2582476        total: 1.72s    remaining: 5.89s
226:    learn: 0.2581968        total: 1.73s    remaining: 5.88s
227:    learn: 0.2580450        total: 1.74s    remaining: 5.88s
228:    learn: 0.2579457        total: 1.74s    remaining: 5.87s
229:    learn: 0.2577143        total: 1.75s    remaining: 5.87s
230:    learn: 0.2574361        total: 1.76s    remaining: 5.87s
231:    learn: 0.2573418        total: 1.77s    remaining: 5.85s
232:    learn: 0.2570080        total: 1.78s    remaining: 5.85s
233:    learn: 0.2567820        total: 1.78s    remaining: 5.84s
234:    learn: 0.2566291        total: 1.79s    remaining: 5.83s
235:    learn: 0.2565305        total: 1.8s     remaining: 5.82s
236:    learn: 0.2563652        total: 1.8s     remaining: 5.81s
237:    learn: 0.2563033        total: 1.81s    remaining: 5.8s
238:    learn: 0.2562018        total: 1.82s    remaining: 5.79s
239:    learn: 0.2561136        total: 1.82s    remaining: 5.78s
240:    learn: 0.2559499        total: 1.83s    remaining: 5.77s
241:    learn: 0.2558078        total: 1.84s    remaining: 5.76s
242:    learn: 0.2555700        total: 1.85s    remaining: 5.75s
243:    learn: 0.2554338        total: 1.85s    remaining: 5.75s
244:    learn: 0.2551703        total: 1.86s    remaining: 5.74s
245:    learn: 0.2549813        total: 1.87s    remaining: 5.73s
246:    learn: 0.2547739        total: 1.88s    remaining: 5.72s
247:    learn: 0.2546285        total: 1.88s    remaining: 5.71s
248:    learn: 0.2545615        total: 1.89s    remaining: 5.7s
249:    learn: 0.2544311        total: 1.9s     remaining: 5.69s
250:    learn: 0.2541400        total: 1.9s     remaining: 5.68s
251:    learn: 0.2540079        total: 1.91s    remaining: 5.67s
252:    learn: 0.2539466        total: 1.92s    remaining: 5.67s
253:    learn: 0.2538098        total: 1.93s    remaining: 5.66s
254:    learn: 0.2536666        total: 1.94s    remaining: 5.66s
```

```
255:    learn: 0.2534861        total: 1.94s    remaining: 5.65s
256:    learn: 0.2532984        total: 1.95s    remaining: 5.65s
257:    learn: 0.2531767        total: 1.96s    remaining: 5.64s
258:    learn: 0.2529697        total: 1.97s    remaining: 5.63s
259:    learn: 0.2528842        total: 1.98s    remaining: 5.63s
260:    learn: 0.2527317        total: 1.99s    remaining: 5.63s
261:    learn: 0.2525346        total: 2s       remaining: 5.63s
262:    learn: 0.2523806        total: 2s       remaining: 5.62s
263:    learn: 0.2519933        total: 2.01s    remaining: 5.62s
264:    learn: 0.2517428        total: 2.02s    remaining: 5.61s
265:    learn: 0.2514956        total: 2.03s    remaining: 5.61s
266:    learn: 0.2514291        total: 2.04s    remaining: 5.6s
267:    learn: 0.2511988        total: 2.05s    remaining: 5.59s
268:    learn: 0.2510766        total: 2.05s    remaining: 5.58s
269:    learn: 0.2508975        total: 2.06s    remaining: 5.57s
270:    learn: 0.2508324        total: 2.07s    remaining: 5.56s
271:    learn: 0.2507690        total: 2.08s    remaining: 5.56s
272:    learn: 0.2504477        total: 2.08s    remaining: 5.55s
273:    learn: 0.2503135        total: 2.09s    remaining: 5.54s
274:    learn: 0.2502092        total: 2.1s     remaining: 5.53s
275:    learn: 0.2500786        total: 2.11s    remaining: 5.53s
276:    learn: 0.2499886        total: 2.11s    remaining: 5.52s
277:    learn: 0.2498526        total: 2.12s    remaining: 5.51s
278:    learn: 0.2497063        total: 2.13s    remaining: 5.5s
279:    learn: 0.2495484        total: 2.14s    remaining: 5.5s
280:    learn: 0.2494960        total: 2.15s    remaining: 5.49s
281:    learn: 0.2493493        total: 2.15s    remaining: 5.48s
282:    learn: 0.2492950        total: 2.16s    remaining: 5.47s
283:    learn: 0.2492046        total: 2.17s    remaining: 5.46s
284:    learn: 0.2490772        total: 2.17s    remaining: 5.45s
285:    learn: 0.2489946        total: 2.18s    remaining: 5.44s
286:    learn: 0.2487838        total: 2.19s    remaining: 5.43s
287:    learn: 0.2486442        total: 2.19s    remaining: 5.43s
288:    learn: 0.2485748        total: 2.2s     remaining: 5.42s
289:    learn: 0.2484430        total: 2.21s    remaining: 5.41s
290:    learn: 0.2482949        total: 2.22s    remaining: 5.41s
291:    learn: 0.2481213        total: 2.23s    remaining: 5.4s
292:    learn: 0.2480382        total: 2.23s    remaining: 5.39s
293:    learn: 0.2479760        total: 2.24s    remaining: 5.38s
294:    learn: 0.2477973        total: 2.25s    remaining: 5.38s
295:    learn: 0.2477217        total: 2.25s    remaining: 5.37s
296:    learn: 0.2474727        total: 2.27s    remaining: 5.36s
297:    learn: 0.2473221        total: 2.27s    remaining: 5.35s
298:    learn: 0.2471884        total: 2.28s    remaining: 5.34s
299:    learn: 0.2470678        total: 2.29s    remaining: 5.34s
300:    learn: 0.2467319        total: 2.29s    remaining: 5.33s
301:    learn: 0.2466049        total: 2.3s     remaining: 5.32s
302:    learn: 0.2465358        total: 2.31s    remaining: 5.31s
```

```
303:    learn: 0.2464291        total: 2.32s    remaining: 5.3s
304:    learn: 0.2462402        total: 2.33s    remaining: 5.3s
305:    learn: 0.2460489        total: 2.33s    remaining: 5.29s
306:    learn: 0.2459813        total: 2.34s    remaining: 5.29s
307:    learn: 0.2458809        total: 2.35s    remaining: 5.28s
308:    learn: 0.2457630        total: 2.36s    remaining: 5.27s
309:    learn: 0.2456670        total: 2.36s    remaining: 5.26s
310:    learn: 0.2455594        total: 2.37s    remaining: 5.25s
311:    learn: 0.2454394        total: 2.38s    remaining: 5.25s
312:    learn: 0.2453137        total: 2.38s    remaining: 5.24s
313:    learn: 0.2451104        total: 2.39s    remaining: 5.23s
314:    learn: 0.2449088        total: 2.4s     remaining: 5.22s
315:    learn: 0.2447927        total: 2.41s    remaining: 5.21s
316:    learn: 0.2447452        total: 2.41s    remaining: 5.2s
317:    learn: 0.2446590        total: 2.42s    remaining: 5.19s
318:    learn: 0.2445401        total: 2.43s    remaining: 5.18s
319:    learn: 0.2444602        total: 2.44s    remaining: 5.18s
320:    learn: 0.2443429        total: 2.44s    remaining: 5.17s
321:    learn: 0.2442142        total: 2.45s    remaining: 5.16s
322:    learn: 0.2441493        total: 2.46s    remaining: 5.15s
323:    learn: 0.2440516        total: 2.47s    remaining: 5.14s
324:    learn: 0.2440068        total: 2.47s    remaining: 5.14s
325:    learn: 0.2438444        total: 2.48s    remaining: 5.13s
326:    learn: 0.2437147        total: 2.49s    remaining: 5.12s
327:    learn: 0.2435862        total: 2.49s    remaining: 5.11s
328:    learn: 0.2434039        total: 2.5s     remaining: 5.1s
329:    learn: 0.2432443        total: 2.51s    remaining: 5.1s
330:    learn: 0.2431434        total: 2.52s    remaining: 5.09s
331:    learn: 0.2430061        total: 2.52s    remaining: 5.08s
332:    learn: 0.2428811        total: 2.53s    remaining: 5.07s
333:    learn: 0.2428170        total: 2.54s    remaining: 5.07s
334:    learn: 0.2426904        total: 2.55s    remaining: 5.06s
335:    learn: 0.2426472        total: 2.56s    remaining: 5.05s
336:    learn: 0.2423173        total: 2.56s    remaining: 5.04s
337:    learn: 0.2422767        total: 2.57s    remaining: 5.04s
338:    learn: 0.2421802        total: 2.58s    remaining: 5.03s
339:    learn: 0.2421081        total: 2.58s    remaining: 5.02s
340:    learn: 0.2418888        total: 2.59s    remaining: 5.01s
341:    learn: 0.2418085        total: 2.6s     remaining: 5s
342:    learn: 0.2417015        total: 2.61s    remaining: 4.99s
343:    learn: 0.2416252        total: 2.61s    remaining: 4.99s
344:    learn: 0.2415549        total: 2.62s    remaining: 4.98s
345:    learn: 0.2414320        total: 2.63s    remaining: 4.97s
346:    learn: 0.2413219        total: 2.64s    remaining: 4.96s
347:    learn: 0.2411286        total: 2.64s    remaining: 4.96s
348:    learn: 0.2409570        total: 2.65s    remaining: 4.95s
349:    learn: 0.2407500        total: 2.66s    remaining: 4.94s
350:    learn: 0.2407000        total: 2.67s    remaining: 4.93s
```

```
351:    learn: 0.2405977        total: 2.67s    remaining: 4.92s
352:    learn: 0.2404088        total: 2.68s    remaining: 4.91s
353:    learn: 0.2402633        total: 2.69s    remaining: 4.9s
354:    learn: 0.2401168        total: 2.69s    remaining: 4.9s
355:    learn: 0.2400521        total: 2.7s     remaining: 4.89s
356:    learn: 0.2399218        total: 2.71s    remaining: 4.89s
357:    learn: 0.2398520        total: 2.72s    remaining: 4.88s
358:    learn: 0.2397879        total: 2.73s    remaining: 4.87s
359:    learn: 0.2397140        total: 2.74s    remaining: 4.87s
360:    learn: 0.2395852        total: 2.75s    remaining: 4.86s
361:    learn: 0.2394685        total: 2.75s    remaining: 4.85s
362:    learn: 0.2393925        total: 2.76s    remaining: 4.84s
363:    learn: 0.2391196        total: 2.77s    remaining: 4.83s
364:    learn: 0.2390323        total: 2.77s    remaining: 4.82s
365:    learn: 0.2389369        total: 2.78s    remaining: 4.81s
366:    learn: 0.2388050        total: 2.79s    remaining: 4.81s
367:    learn: 0.2385485        total: 2.79s    remaining: 4.8s
368:    learn: 0.2384069        total: 2.8s     remaining: 4.79s
369:    learn: 0.2382815        total: 2.81s    remaining: 4.78s
370:    learn: 0.2381706        total: 2.81s    remaining: 4.77s
371:    learn: 0.2381182        total: 2.82s    remaining: 4.77s
372:    learn: 0.2380281        total: 2.83s    remaining: 4.76s
373:    learn: 0.2379772        total: 2.84s    remaining: 4.75s
374:    learn: 0.2378570        total: 2.84s    remaining: 4.74s
375:    learn: 0.2377523        total: 2.85s    remaining: 4.73s
376:    learn: 0.2376650        total: 2.86s    remaining: 4.72s
377:    learn: 0.2375562        total: 2.87s    remaining: 4.71s
378:    learn: 0.2375101        total: 2.87s    remaining: 4.71s
379:    learn: 0.2374183        total: 2.88s    remaining: 4.7s
380:    learn: 0.2373166        total: 2.89s    remaining: 4.69s
381:    learn: 0.2372264        total: 2.89s    remaining: 4.68s
382:    learn: 0.2371620        total: 2.9s     remaining: 4.67s
383:    learn: 0.2370399        total: 2.91s    remaining: 4.67s
384:    learn: 0.2368955        total: 2.92s    remaining: 4.66s
385:    learn: 0.2367418        total: 2.93s    remaining: 4.66s
386:    learn: 0.2366443        total: 2.94s    remaining: 4.65s
387:    learn: 0.2365896        total: 2.94s    remaining: 4.64s
388:    learn: 0.2365390        total: 2.95s    remaining: 4.63s
389:    learn: 0.2364440        total: 2.96s    remaining: 4.63s
390:    learn: 0.2363401        total: 2.97s    remaining: 4.62s
391:    learn: 0.2362679        total: 2.97s    remaining: 4.61s
392:    learn: 0.2361593        total: 2.98s    remaining: 4.61s
393:    learn: 0.2361014        total: 2.99s    remaining: 4.6s
394:    learn: 0.2359888        total: 3s       remaining: 4.59s
395:    learn: 0.2358184        total: 3s       remaining: 4.58s
396:    learn: 0.2357462        total: 3.01s    remaining: 4.57s
397:    learn: 0.2356465        total: 3.02s    remaining: 4.57s
398:    learn: 0.2355257        total: 3.03s    remaining: 4.56s
```

```
399:    learn: 0.2354565        total: 3.04s    remaining: 4.55s
400:    learn: 0.2353964        total: 3.04s    remaining: 4.54s
401:    learn: 0.2353410        total: 3.05s    remaining: 4.54s
402:    learn: 0.2352811        total: 3.06s    remaining: 4.53s
403:    learn: 0.2350930        total: 3.06s    remaining: 4.52s
404:    learn: 0.2350068        total: 3.07s    remaining: 4.51s
405:    learn: 0.2349422        total: 3.08s    remaining: 4.51s
406:    learn: 0.2348809        total: 3.09s    remaining: 4.5s
407:    learn: 0.2347684        total: 3.1s     remaining: 4.5s
408:    learn: 0.2346330        total: 3.11s    remaining: 4.49s
409:    learn: 0.2345363        total: 3.11s    remaining: 4.48s
410:    learn: 0.2344374        total: 3.12s    remaining: 4.47s
411:    learn: 0.2343196        total: 3.13s    remaining: 4.47s
412:    learn: 0.2341400        total: 3.14s    remaining: 4.46s
413:    learn: 0.2339666        total: 3.15s    remaining: 4.45s
414:    learn: 0.2337494        total: 3.15s    remaining: 4.44s
415:    learn: 0.2336505        total: 3.16s    remaining: 4.43s
416:    learn: 0.2335918        total: 3.17s    remaining: 4.43s
417:    learn: 0.2334656        total: 3.17s    remaining: 4.42s
418:    learn: 0.2333640        total: 3.18s    remaining: 4.41s
419:    learn: 0.2332128        total: 3.19s    remaining: 4.4s
420:    learn: 0.2331312        total: 3.19s    remaining: 4.39s
421:    learn: 0.2330216        total: 3.2s     remaining: 4.38s
422:    learn: 0.2328705        total: 3.21s    remaining: 4.38s
423:    learn: 0.2327858        total: 3.22s    remaining: 4.37s
424:    learn: 0.2327207        total: 3.23s    remaining: 4.36s
425:    learn: 0.2325516        total: 3.23s    remaining: 4.36s
426:    learn: 0.2324806        total: 3.24s    remaining: 4.35s
427:    learn: 0.2323698        total: 3.25s    remaining: 4.34s
428:    learn: 0.2322613        total: 3.25s    remaining: 4.33s
429:    learn: 0.2321819        total: 3.26s    remaining: 4.32s
430:    learn: 0.2320607        total: 3.27s    remaining: 4.32s
431:    learn: 0.2319450        total: 3.28s    remaining: 4.31s
432:    learn: 0.2318016        total: 3.29s    remaining: 4.3s
433:    learn: 0.2317562        total: 3.29s    remaining: 4.29s
434:    learn: 0.2314860        total: 3.3s     remaining: 4.29s
435:    learn: 0.2313911        total: 3.31s    remaining: 4.28s
436:    learn: 0.2313025        total: 3.31s    remaining: 4.27s
437:    learn: 0.2312163        total: 3.32s    remaining: 4.26s
438:    learn: 0.2311335        total: 3.33s    remaining: 4.25s
439:    learn: 0.2310738        total: 3.34s    remaining: 4.25s
440:    learn: 0.2309571        total: 3.34s    remaining: 4.24s
441:    learn: 0.2308935        total: 3.35s    remaining: 4.23s
442:    learn: 0.2308111        total: 3.36s    remaining: 4.22s
443:    learn: 0.2307010        total: 3.37s    remaining: 4.21s
444:    learn: 0.2306212        total: 3.37s    remaining: 4.21s
445:    learn: 0.2305602        total: 3.38s    remaining: 4.2s
446:    learn: 0.2304547        total: 3.39s    remaining: 4.19s
```

```
447:    learn: 0.2302867        total: 3.4s     remaining: 4.18s
448:    learn: 0.2301709        total: 3.4s     remaining: 4.17s
449:    learn: 0.2300588        total: 3.41s    remaining: 4.17s
450:    learn: 0.2299361        total: 3.42s    remaining: 4.16s
451:    learn: 0.2298232        total: 3.42s    remaining: 4.15s
452:    learn: 0.2296529        total: 3.43s    remaining: 4.14s
453:    learn: 0.2295829        total: 3.44s    remaining: 4.13s
454:    learn: 0.2295044        total: 3.45s    remaining: 4.13s
455:    learn: 0.2294366        total: 3.45s    remaining: 4.12s
456:    learn: 0.2293837        total: 3.46s    remaining: 4.11s
457:    learn: 0.2292331        total: 3.47s    remaining: 4.1s
458:    learn: 0.2291076        total: 3.48s    remaining: 4.1s
459:    learn: 0.2289837        total: 3.48s    remaining: 4.09s
460:    learn: 0.2288429        total: 3.49s    remaining: 4.08s
461:    learn: 0.2287785        total: 3.5s     remaining: 4.08s
462:    learn: 0.2287393        total: 3.51s    remaining: 4.07s
463:    learn: 0.2285673        total: 3.52s    remaining: 4.06s
464:    learn: 0.2284807        total: 3.52s    remaining: 4.06s
465:    learn: 0.2282884        total: 3.53s    remaining: 4.05s
466:    learn: 0.2281528        total: 3.54s    remaining: 4.04s
467:    learn: 0.2280556        total: 3.55s    remaining: 4.03s
468:    learn: 0.2280090        total: 3.55s    remaining: 4.02s
469:    learn: 0.2278405        total: 3.56s    remaining: 4.02s
470:    learn: 0.2277539        total: 3.57s    remaining: 4.01s
471:    learn: 0.2276695        total: 3.58s    remaining: 4s
472:    learn: 0.2275735        total: 3.58s    remaining: 3.99s
473:    learn: 0.2274386        total: 3.59s    remaining: 3.98s
474:    learn: 0.2273206        total: 3.6s     remaining: 3.98s
475:    learn: 0.2272481        total: 3.61s    remaining: 3.97s
476:    learn: 0.2271142        total: 3.61s    remaining: 3.96s
477:    learn: 0.2270268        total: 3.62s    remaining: 3.96s
478:    learn: 0.2269557        total: 3.63s    remaining: 3.95s
479:    learn: 0.2269171        total: 3.64s    remaining: 3.94s
480:    learn: 0.2267782        total: 3.64s    remaining: 3.93s
481:    learn: 0.2267143        total: 3.65s    remaining: 3.92s
482:    learn: 0.2266256        total: 3.66s    remaining: 3.92s
483:    learn: 0.2265127        total: 3.66s    remaining: 3.91s
484:    learn: 0.2263948        total: 3.67s    remaining: 3.9s
485:    learn: 0.2263730        total: 3.68s    remaining: 3.89s
486:    learn: 0.2262754        total: 3.69s    remaining: 3.88s
487:    learn: 0.2261624        total: 3.69s    remaining: 3.88s
488:    learn: 0.2260744        total: 3.7s     remaining: 3.87s
489:    learn: 0.2260175        total: 3.71s    remaining: 3.86s
490:    learn: 0.2259522        total: 3.72s    remaining: 3.85s
491:    learn: 0.2258750        total: 3.72s    remaining: 3.85s
492:    learn: 0.2257361        total: 3.73s    remaining: 3.84s
493:    learn: 0.2256667        total: 3.74s    remaining: 3.83s
494:    learn: 0.2256320        total: 3.75s    remaining: 3.82s
```

```
495:    learn: 0.2255182        total: 3.75s    remaining: 3.81s
496:    learn: 0.2254560        total: 3.76s    remaining: 3.81s
497:    learn: 0.2253650        total: 3.77s    remaining: 3.8s
498:    learn: 0.2253089        total: 3.77s    remaining: 3.79s
499:    learn: 0.2252165        total: 3.78s    remaining: 3.78s
500:    learn: 0.2251426        total: 3.79s    remaining: 3.77s
501:    learn: 0.2250667        total: 3.8s     remaining: 3.77s
502:    learn: 0.2249659        total: 3.8s     remaining: 3.76s
503:    learn: 0.2248984        total: 3.81s    remaining: 3.75s
504:    learn: 0.2248114        total: 3.82s    remaining: 3.74s
505:    learn: 0.2247385        total: 3.83s    remaining: 3.73s
506:    learn: 0.2246855        total: 3.83s    remaining: 3.73s
507:    learn: 0.2245985        total: 3.84s    remaining: 3.72s
508:    learn: 0.2245572        total: 3.85s    remaining: 3.71s
509:    learn: 0.2245027        total: 3.85s    remaining: 3.7s
510:    learn: 0.2244342        total: 3.86s    remaining: 3.7s
511:    learn: 0.2243834        total: 3.87s    remaining: 3.69s
512:    learn: 0.2242484        total: 3.88s    remaining: 3.68s
513:    learn: 0.2241403        total: 3.89s    remaining: 3.68s
514:    learn: 0.2240964        total: 3.9s     remaining: 3.67s
515:    learn: 0.2240091        total: 3.91s    remaining: 3.67s
516:    learn: 0.2239555        total: 3.92s    remaining: 3.66s
517:    learn: 0.2239154        total: 3.92s    remaining: 3.65s
518:    learn: 0.2238027        total: 3.93s    remaining: 3.64s
519:    learn: 0.2237356        total: 3.94s    remaining: 3.64s
520:    learn: 0.2236134        total: 3.95s    remaining: 3.63s
521:    learn: 0.2235368        total: 3.95s    remaining: 3.62s
522:    learn: 0.2233868        total: 3.96s    remaining: 3.61s
523:    learn: 0.2233478        total: 3.97s    remaining: 3.61s
524:    learn: 0.2232885        total: 3.98s    remaining: 3.6s
525:    learn: 0.2231955        total: 3.98s    remaining: 3.59s
526:    learn: 0.2230903        total: 3.99s    remaining: 3.58s
527:    learn: 0.2230562        total: 4s       remaining: 3.58s
528:    learn: 0.2229859        total: 4.01s    remaining: 3.57s
529:    learn: 0.2228743        total: 4.01s    remaining: 3.56s
530:    learn: 0.2227998        total: 4.02s    remaining: 3.55s
531:    learn: 0.2227145        total: 4.03s    remaining: 3.55s
532:    learn: 0.2226757        total: 4.04s    remaining: 3.54s
533:    learn: 0.2225910        total: 4.05s    remaining: 3.53s
534:    learn: 0.2225609        total: 4.05s    remaining: 3.52s
535:    learn: 0.2224414        total: 4.06s    remaining: 3.51s
536:    learn: 0.2223156        total: 4.07s    remaining: 3.51s
537:    learn: 0.2222712        total: 4.08s    remaining: 3.5s
538:    learn: 0.2222277        total: 4.08s    remaining: 3.49s
539:    learn: 0.2220867        total: 4.09s    remaining: 3.48s
540:    learn: 0.2220441        total: 4.1s     remaining: 3.48s
541:    learn: 0.2219668        total: 4.11s    remaining: 3.47s
542:    learn: 0.2218736        total: 4.12s    remaining: 3.46s
```

```
543:    learn: 0.2217744        total: 4.12s    remaining: 3.46s
544:    learn: 0.2217086        total: 4.13s    remaining: 3.45s
545:    learn: 0.2215576        total: 4.14s    remaining: 3.44s
546:    learn: 0.2215211        total: 4.15s    remaining: 3.43s
547:    learn: 0.2213884        total: 4.15s    remaining: 3.43s
548:    learn: 0.2213076        total: 4.16s    remaining: 3.42s
549:    learn: 0.2212358        total: 4.17s    remaining: 3.41s
550:    learn: 0.2211706        total: 4.18s    remaining: 3.4s
551:    learn: 0.2210269        total: 4.18s    remaining: 3.4s
552:    learn: 0.2209812        total: 4.19s    remaining: 3.39s
553:    learn: 0.2209251        total: 4.2s     remaining: 3.38s
554:    learn: 0.2208601        total: 4.2s     remaining: 3.37s
555:    learn: 0.2208279        total: 4.21s    remaining: 3.36s
556:    learn: 0.2207521        total: 4.22s    remaining: 3.35s
557:    learn: 0.2206564        total: 4.23s    remaining: 3.35s
558:    learn: 0.2205526        total: 4.23s    remaining: 3.34s
559:    learn: 0.2204586        total: 4.24s    remaining: 3.33s
560:    learn: 0.2203381        total: 4.25s    remaining: 3.32s
561:    learn: 0.2202606        total: 4.25s    remaining: 3.32s
562:    learn: 0.2201241        total: 4.26s    remaining: 3.31s
563:    learn: 0.2200514        total: 4.27s    remaining: 3.3s
564:    learn: 0.2199656        total: 4.28s    remaining: 3.29s
565:    learn: 0.2199003        total: 4.29s    remaining: 3.29s
566:    learn: 0.2197658        total: 4.29s    remaining: 3.28s
567:    learn: 0.2197382        total: 4.3s     remaining: 3.27s
568:    learn: 0.2196728        total: 4.31s    remaining: 3.27s
569:    learn: 0.2195156        total: 4.32s    remaining: 3.26s
570:    learn: 0.2194661        total: 4.32s    remaining: 3.25s
571:    learn: 0.2194076        total: 4.33s    remaining: 3.24s
572:    learn: 0.2193512        total: 4.34s    remaining: 3.23s
573:    learn: 0.2192336        total: 4.35s    remaining: 3.23s
574:    learn: 0.2191472        total: 4.36s    remaining: 3.22s
575:    learn: 0.2190393        total: 4.36s    remaining: 3.21s
576:    learn: 0.2190072        total: 4.37s    remaining: 3.2s
577:    learn: 0.2189295        total: 4.38s    remaining: 3.19s
578:    learn: 0.2188572        total: 4.38s    remaining: 3.19s
579:    learn: 0.2187447        total: 4.39s    remaining: 3.18s
580:    learn: 0.2186736        total: 4.4s     remaining: 3.17s
581:    learn: 0.2186522        total: 4.41s    remaining: 3.16s
582:    learn: 0.2186131        total: 4.41s    remaining: 3.16s
583:    learn: 0.2185212        total: 4.42s    remaining: 3.15s
584:    learn: 0.2184110        total: 4.43s    remaining: 3.14s
585:    learn: 0.2182688        total: 4.43s    remaining: 3.13s
586:    learn: 0.2181922        total: 4.44s    remaining: 3.13s
587:    learn: 0.2180476        total: 4.45s    remaining: 3.12s
588:    learn: 0.2179974        total: 4.46s    remaining: 3.11s
589:    learn: 0.2179331        total: 4.47s    remaining: 3.1s
590:    learn: 0.2178931        total: 4.48s    remaining: 3.1s
```

```
591:    learn: 0.2178253        total: 4.48s    remaining: 3.09s
592:    learn: 0.2177628        total: 4.49s    remaining: 3.08s
593:    learn: 0.2176926        total: 4.5s     remaining: 3.08s
594:    learn: 0.2176083        total: 4.51s    remaining: 3.07s
595:    learn: 0.2175633        total: 4.51s    remaining: 3.06s
596:    learn: 0.2175248        total: 4.52s    remaining: 3.05s
597:    learn: 0.2174396        total: 4.53s    remaining: 3.05s
598:    learn: 0.2173452        total: 4.54s    remaining: 3.04s
599:    learn: 0.2172828        total: 4.54s    remaining: 3.03s
600:    learn: 0.2171670        total: 4.55s    remaining: 3.02s
601:    learn: 0.2170759        total: 4.56s    remaining: 3.01s
602:    learn: 0.2170365        total: 4.57s    remaining: 3.01s
603:    learn: 0.2170056        total: 4.57s    remaining: 3s
604:    learn: 0.2169319        total: 4.58s    remaining: 2.99s
605:    learn: 0.2167941        total: 4.59s    remaining: 2.98s
606:    learn: 0.2166963        total: 4.59s    remaining: 2.98s
607:    learn: 0.2166370        total: 4.6s     remaining: 2.97s
608:    learn: 0.2165870        total: 4.61s    remaining: 2.96s
609:    learn: 0.2165340        total: 4.62s    remaining: 2.95s
610:    learn: 0.2165032        total: 4.62s    remaining: 2.94s
611:    learn: 0.2164031        total: 4.63s    remaining: 2.94s
612:    learn: 0.2163563        total: 4.64s    remaining: 2.93s
613:    learn: 0.2163086        total: 4.64s    remaining: 2.92s
614:    learn: 0.2162614        total: 4.65s    remaining: 2.91s
615:    learn: 0.2161605        total: 4.66s    remaining: 2.9s
616:    learn: 0.2160910        total: 4.67s    remaining: 2.9s
617:    learn: 0.2160247        total: 4.68s    remaining: 2.89s
618:    learn: 0.2159467        total: 4.69s    remaining: 2.88s
619:    learn: 0.2158863        total: 4.7s     remaining: 2.88s
620:    learn: 0.2158367        total: 4.7s     remaining: 2.87s
621:    learn: 0.2157776        total: 4.71s    remaining: 2.86s
622:    learn: 0.2157110        total: 4.72s    remaining: 2.85s
623:    learn: 0.2156504        total: 4.72s    remaining: 2.85s
624:    learn: 0.2154907        total: 4.73s    remaining: 2.84s
625:    learn: 0.2154484        total: 4.74s    remaining: 2.83s
626:    learn: 0.2153987        total: 4.75s    remaining: 2.82s
627:    learn: 0.2152711        total: 4.75s    remaining: 2.81s
628:    learn: 0.2151045        total: 4.76s    remaining: 2.81s
629:    learn: 0.2150729        total: 4.77s    remaining: 2.8s
630:    learn: 0.2150234        total: 4.78s    remaining: 2.79s
631:    learn: 0.2149710        total: 4.78s    remaining: 2.78s
632:    learn: 0.2149293        total: 4.79s    remaining: 2.78s
633:    learn: 0.2148823        total: 4.8s     remaining: 2.77s
634:    learn: 0.2147472        total: 4.8s     remaining: 2.76s
635:    learn: 0.2146764        total: 4.81s    remaining: 2.75s
636:    learn: 0.2145452        total: 4.82s    remaining: 2.75s
637:    learn: 0.2144784        total: 4.83s    remaining: 2.74s
638:    learn: 0.2143927        total: 4.83s    remaining: 2.73s
```

```
639:    learn: 0.2143425        total: 4.84s    remaining: 2.72s
640:    learn: 0.2143227        total: 4.85s    remaining: 2.71s
641:    learn: 0.2142429        total: 4.86s    remaining: 2.71s
642:    learn: 0.2141991        total: 4.87s    remaining: 2.7s
643:    learn: 0.2141149        total: 4.87s    remaining: 2.69s
644:    learn: 0.2140528        total: 4.88s    remaining: 2.69s
645:    learn: 0.2139976        total: 4.89s    remaining: 2.68s
646:    learn: 0.2139236        total: 4.9s     remaining: 2.67s
647:    learn: 0.2138725        total: 4.91s    remaining: 2.67s
648:    learn: 0.2138010        total: 4.91s    remaining: 2.66s
649:    learn: 0.2137785        total: 4.92s    remaining: 2.65s
650:    learn: 0.2136042        total: 4.93s    remaining: 2.64s
651:    learn: 0.2135415        total: 4.94s    remaining: 2.64s
652:    learn: 0.2134389        total: 4.95s    remaining: 2.63s
653:    learn: 0.2133508        total: 4.95s    remaining: 2.62s
654:    learn: 0.2132563        total: 4.96s    remaining: 2.61s
655:    learn: 0.2132120        total: 4.97s    remaining: 2.61s
656:    learn: 0.2131378        total: 4.98s    remaining: 2.6s
657:    learn: 0.2130766        total: 4.98s    remaining: 2.59s
658:    learn: 0.2129704        total: 4.99s    remaining: 2.58s
659:    learn: 0.2129074        total: 5s       remaining: 2.58s
660:    learn: 0.2128027        total: 5.01s    remaining: 2.57s
661:    learn: 0.2127506        total: 5.01s    remaining: 2.56s
662:    learn: 0.2127038        total: 5.02s    remaining: 2.55s
663:    learn: 0.2126485        total: 5.03s    remaining: 2.54s
664:    learn: 0.2126081        total: 5.03s    remaining: 2.54s
665:    learn: 0.2125666        total: 5.04s    remaining: 2.53s
666:    learn: 0.2124447        total: 5.05s    remaining: 2.52s
667:    learn: 0.2124045        total: 5.06s    remaining: 2.51s
668:    learn: 0.2123625        total: 5.07s    remaining: 2.51s
669:    learn: 0.2122648        total: 5.07s    remaining: 2.5s
670:    learn: 0.2121857        total: 5.08s    remaining: 2.49s
671:    learn: 0.2121456        total: 5.09s    remaining: 2.48s
672:    learn: 0.2120307        total: 5.09s    remaining: 2.48s
673:    learn: 0.2120022        total: 5.1s     remaining: 2.47s
674:    learn: 0.2119633        total: 5.11s    remaining: 2.46s
675:    learn: 0.2118610        total: 5.12s    remaining: 2.45s
676:    learn: 0.2117962        total: 5.12s    remaining: 2.44s
677:    learn: 0.2116758        total: 5.13s    remaining: 2.44s
678:    learn: 0.2115914        total: 5.14s    remaining: 2.43s
679:    learn: 0.2115388        total: 5.14s    remaining: 2.42s
680:    learn: 0.2114400        total: 5.15s    remaining: 2.41s
681:    learn: 0.2113517        total: 5.16s    remaining: 2.4s
682:    learn: 0.2112321        total: 5.17s    remaining: 2.4s
683:    learn: 0.2111697        total: 5.17s    remaining: 2.39s
684:    learn: 0.2110972        total: 5.18s    remaining: 2.38s
685:    learn: 0.2110670        total: 5.19s    remaining: 2.38s
686:    learn: 0.2109786        total: 5.2s     remaining: 2.37s
```

```
687:    learn: 0.2108833        total: 5.21s    remaining: 2.36s
688:    learn: 0.2108366        total: 5.21s    remaining: 2.35s
689:    learn: 0.2107978        total: 5.22s    remaining: 2.34s
690:    learn: 0.2106933        total: 5.22s    remaining: 2.34s
691:    learn: 0.2106116        total: 5.23s    remaining: 2.33s
692:    learn: 0.2105507        total: 5.24s    remaining: 2.32s
693:    learn: 0.2104682        total: 5.25s    remaining: 2.31s
694:    learn: 0.2103703        total: 5.26s    remaining: 2.31s
695:    learn: 0.2102580        total: 5.28s    remaining: 2.31s
696:    learn: 0.2100917        total: 5.31s    remaining: 2.31s
697:    learn: 0.2100012        total: 5.32s    remaining: 2.3s
698:    learn: 0.2099290        total: 5.33s    remaining: 2.29s
699:    learn: 0.2098311        total: 5.33s    remaining: 2.29s
700:    learn: 0.2097350        total: 5.34s    remaining: 2.28s
701:    learn: 0.2096702        total: 5.35s    remaining: 2.27s
702:    learn: 0.2095403        total: 5.36s    remaining: 2.27s
703:    learn: 0.2094978        total: 5.37s    remaining: 2.26s
704:    learn: 0.2094711        total: 5.38s    remaining: 2.25s
705:    learn: 0.2094288        total: 5.38s    remaining: 2.24s
706:    learn: 0.2093817        total: 5.39s    remaining: 2.23s
707:    learn: 0.2093405        total: 5.4s     remaining: 2.23s
708:    learn: 0.2093015        total: 5.41s    remaining: 2.22s
709:    learn: 0.2092517        total: 5.41s    remaining: 2.21s
710:    learn: 0.2091766        total: 5.42s    remaining: 2.2s
711:    learn: 0.2091340        total: 5.43s    remaining: 2.19s
712:    learn: 0.2090614        total: 5.43s    remaining: 2.19s
713:    learn: 0.2089821        total: 5.44s    remaining: 2.18s
714:    learn: 0.2088649        total: 5.45s    remaining: 2.17s
715:    learn: 0.2087393        total: 5.46s    remaining: 2.17s
716:    learn: 0.2086591        total: 5.47s    remaining: 2.16s
717:    learn: 0.2085836        total: 5.47s    remaining: 2.15s
718:    learn: 0.2085592        total: 5.48s    remaining: 2.14s
719:    learn: 0.2084928        total: 5.49s    remaining: 2.13s
720:    learn: 0.2084213        total: 5.5s     remaining: 2.13s
721:    learn: 0.2083720        total: 5.5s     remaining: 2.12s
722:    learn: 0.2083181        total: 5.51s    remaining: 2.11s
723:    learn: 0.2082312        total: 5.52s    remaining: 2.1s
724:    learn: 0.2081222        total: 5.53s    remaining: 2.1s
725:    learn: 0.2080488        total: 5.53s    remaining: 2.09s
726:    learn: 0.2079925        total: 5.54s    remaining: 2.08s
727:    learn: 0.2079488        total: 5.55s    remaining: 2.07s
728:    learn: 0.2078849        total: 5.55s    remaining: 2.06s
729:    learn: 0.2077967        total: 5.56s    remaining: 2.06s
730:    learn: 0.2076761        total: 5.57s    remaining: 2.05s
731:    learn: 0.2076234        total: 5.58s    remaining: 2.04s
732:    learn: 0.2075730        total: 5.58s    remaining: 2.03s
733:    learn: 0.2074576        total: 5.59s    remaining: 2.03s
734:    learn: 0.2074234        total: 5.6s     remaining: 2.02s
```

```
735:    learn: 0.2073919        total: 5.61s    remaining: 2.01s
736:    learn: 0.2072863        total: 5.61s    remaining: 2s
737:    learn: 0.2072357        total: 5.62s    remaining: 2s
738:    learn: 0.2071631        total: 5.63s    remaining: 1.99s
739:    learn: 0.2070946        total: 5.64s    remaining: 1.98s
740:    learn: 0.2070393        total: 5.64s    remaining: 1.97s
741:    learn: 0.2069792        total: 5.65s    remaining: 1.97s
742:    learn: 0.2069120        total: 5.66s    remaining: 1.96s
743:    learn: 0.2068597        total: 5.67s    remaining: 1.95s
744:    learn: 0.2067746        total: 5.67s    remaining: 1.94s
745:    learn: 0.2067119        total: 5.68s    remaining: 1.93s
746:    learn: 0.2066711        total: 5.69s    remaining: 1.93s
747:    learn: 0.2065970        total: 5.7s     remaining: 1.92s
748:    learn: 0.2065104        total: 5.7s     remaining: 1.91s
749:    learn: 0.2064469        total: 5.71s    remaining: 1.9s
750:    learn: 0.2063806        total: 5.72s    remaining: 1.9s
751:    learn: 0.2062947        total: 5.72s    remaining: 1.89s
752:    learn: 0.2062397        total: 5.73s    remaining: 1.88s
753:    learn: 0.2061751        total: 5.74s    remaining: 1.87s
754:    learn: 0.2060543        total: 5.75s    remaining: 1.86s
755:    learn: 0.2059959        total: 5.75s    remaining: 1.86s
756:    learn: 0.2059584        total: 5.76s    remaining: 1.85s
757:    learn: 0.2058354        total: 5.77s    remaining: 1.84s
758:    learn: 0.2057380        total: 5.77s    remaining: 1.83s
759:    learn: 0.2055958        total: 5.78s    remaining: 1.83s
760:    learn: 0.2054706        total: 5.79s    remaining: 1.82s
761:    learn: 0.2054075        total: 5.8s     remaining: 1.81s
762:    learn: 0.2053573        total: 5.8s     remaining: 1.8s
763:    learn: 0.2053176        total: 5.81s    remaining: 1.79s
764:    learn: 0.2052907        total: 5.82s    remaining: 1.79s
765:    learn: 0.2052553        total: 5.83s    remaining: 1.78s
766:    learn: 0.2052179        total: 5.84s    remaining: 1.77s
767:    learn: 0.2051160        total: 5.85s    remaining: 1.77s
768:    learn: 0.2050816        total: 5.85s    remaining: 1.76s
769:    learn: 0.2050289        total: 5.86s    remaining: 1.75s
770:    learn: 0.2050018        total: 5.87s    remaining: 1.74s
771:    learn: 0.2049505        total: 5.88s    remaining: 1.74s
772:    learn: 0.2048789        total: 5.88s    remaining: 1.73s
773:    learn: 0.2048439        total: 5.89s    remaining: 1.72s
774:    learn: 0.2047550        total: 5.9s     remaining: 1.71s
775:    learn: 0.2046661        total: 5.91s    remaining: 1.71s
776:    learn: 0.2045456        total: 5.92s    remaining: 1.7s
777:    learn: 0.2044616        total: 5.92s    remaining: 1.69s
778:    learn: 0.2044068        total: 5.93s    remaining: 1.68s
779:    learn: 0.2043719        total: 5.94s    remaining: 1.67s
780:    learn: 0.2042999        total: 5.95s    remaining: 1.67s
781:    learn: 0.2042597        total: 5.95s    remaining: 1.66s
782:    learn: 0.2041890        total: 5.96s    remaining: 1.65s
```

```
783:    learn: 0.2041490        total: 5.97s    remaining: 1.64s
784:    learn: 0.2041316        total: 5.97s    remaining: 1.64s
785:    learn: 0.2040443        total: 5.98s    remaining: 1.63s
786:    learn: 0.2040232        total: 5.99s    remaining: 1.62s
787:    learn: 0.2039905        total: 5.99s    remaining: 1.61s
788:    learn: 0.2038942        total: 6s       remaining: 1.6s
789:    learn: 0.2037922        total: 6.01s    remaining: 1.6s
790:    learn: 0.2037097        total: 6.02s    remaining: 1.59s
791:    learn: 0.2036660        total: 6.03s    remaining: 1.58s
792:    learn: 0.2036133        total: 6.03s    remaining: 1.57s
793:    learn: 0.2034963        total: 6.04s    remaining: 1.57s
794:    learn: 0.2034589        total: 6.05s    remaining: 1.56s
795:    learn: 0.2033161        total: 6.06s    remaining: 1.55s
796:    learn: 0.2032725        total: 6.07s    remaining: 1.54s
797:    learn: 0.2032122        total: 6.07s    remaining: 1.54s
798:    learn: 0.2031486        total: 6.08s    remaining: 1.53s
799:    learn: 0.2031016        total: 6.09s    remaining: 1.52s
800:    learn: 0.2030150        total: 6.09s    remaining: 1.51s
801:    learn: 0.2029889        total: 6.1s     remaining: 1.51s
802:    learn: 0.2029613        total: 6.11s    remaining: 1.5s
803:    learn: 0.2029153        total: 6.12s    remaining: 1.49s
804:    learn: 0.2028177        total: 6.12s    remaining: 1.48s
805:    learn: 0.2027606        total: 6.13s    remaining: 1.48s
806:    learn: 0.2026402        total: 6.14s    remaining: 1.47s
807:    learn: 0.2025656        total: 6.15s    remaining: 1.46s
808:    learn: 0.2025148        total: 6.16s    remaining: 1.45s
809:    learn: 0.2024853        total: 6.16s    remaining: 1.45s
810:    learn: 0.2024193        total: 6.17s    remaining: 1.44s
811:    learn: 0.2023812        total: 6.17s    remaining: 1.43s
812:    learn: 0.2022887        total: 6.18s    remaining: 1.42s
813:    learn: 0.2022677        total: 6.19s    remaining: 1.41s
814:    learn: 0.2022463        total: 6.2s     remaining: 1.41s
815:    learn: 0.2021795        total: 6.21s    remaining: 1.4s
816:    learn: 0.2021576        total: 6.21s    remaining: 1.39s
817:    learn: 0.2021092        total: 6.22s    remaining: 1.38s
818:    learn: 0.2020509        total: 6.23s    remaining: 1.38s
819:    learn: 0.2019898        total: 6.24s    remaining: 1.37s
820:    learn: 0.2019309        total: 6.25s    remaining: 1.36s
821:    learn: 0.2018749        total: 6.25s    remaining: 1.35s
822:    learn: 0.2018187        total: 6.26s    remaining: 1.35s
823:    learn: 0.2017707        total: 6.27s    remaining: 1.34s
824:    learn: 0.2017286        total: 6.27s    remaining: 1.33s
825:    learn: 0.2016544        total: 6.28s    remaining: 1.32s
826:    learn: 0.2016264        total: 6.29s    remaining: 1.31s
827:    learn: 0.2015446        total: 6.3s     remaining: 1.31s
828:    learn: 0.2014997        total: 6.3s     remaining: 1.3s
829:    learn: 0.2014743        total: 6.31s    remaining: 1.29s
830:    learn: 0.2013827        total: 6.32s    remaining: 1.28s
```

```
831:    learn: 0.2013558        total: 6.32s    remaining: 1.28s
832:    learn: 0.2012587        total: 6.33s    remaining: 1.27s
833:    learn: 0.2012067        total: 6.34s    remaining: 1.26s
834:    learn: 0.2011487        total: 6.35s    remaining: 1.25s
835:    learn: 0.2010849        total: 6.35s    remaining: 1.25s
836:    learn: 0.2010500        total: 6.36s    remaining: 1.24s
837:    learn: 0.2010181        total: 6.37s    remaining: 1.23s
838:    learn: 0.2009305        total: 6.38s    remaining: 1.22s
839:    learn: 0.2008627        total: 6.38s    remaining: 1.22s
840:    learn: 0.2007853        total: 6.39s    remaining: 1.21s
841:    learn: 0.2007330        total: 6.4s     remaining: 1.2s
842:    learn: 0.2006637        total: 6.41s    remaining: 1.19s
843:    learn: 0.2006016        total: 6.41s    remaining: 1.19s
844:    learn: 0.2005597        total: 6.42s    remaining: 1.18s
845:    learn: 0.2004585        total: 6.43s    remaining: 1.17s
846:    learn: 0.2004125        total: 6.44s    remaining: 1.16s
847:    learn: 0.2003469        total: 6.45s    remaining: 1.16s
848:    learn: 0.2002632        total: 6.45s    remaining: 1.15s
849:    learn: 0.2002120        total: 6.46s    remaining: 1.14s
850:    learn: 0.2001349        total: 6.47s    remaining: 1.13s
851:    learn: 0.2000158        total: 6.47s    remaining: 1.12s
852:    learn: 0.1999519        total: 6.48s    remaining: 1.12s
853:    learn: 0.1998690        total: 6.49s    remaining: 1.11s
854:    learn: 0.1997884        total: 6.5s     remaining: 1.1s
855:    learn: 0.1997471        total: 6.5s     remaining: 1.09s
856:    learn: 0.1997237        total: 6.51s    remaining: 1.09s
857:    learn: 0.1996560        total: 6.52s    remaining: 1.08s
858:    learn: 0.1995909        total: 6.53s    remaining: 1.07s
859:    learn: 0.1995407        total: 6.53s    remaining: 1.06s
860:    learn: 0.1995003        total: 6.54s    remaining: 1.05s
861:    learn: 0.1994489        total: 6.55s    remaining: 1.05s
862:    learn: 0.1994035        total: 6.55s    remaining: 1.04s
863:    learn: 0.1993301        total: 6.56s    remaining: 1.03s
864:    learn: 0.1992481        total: 6.57s    remaining: 1.02s
865:    learn: 0.1992035        total: 6.58s    remaining: 1.02s
866:    learn: 0.1991180        total: 6.58s    remaining: 1.01s
867:    learn: 0.1990549        total: 6.59s    remaining: 1s
868:    learn: 0.1990024        total: 6.6s     remaining: 995ms
869:    learn: 0.1988013        total: 6.61s    remaining: 988ms
870:    learn: 0.1987437        total: 6.62s    remaining: 980ms
871:    learn: 0.1986990        total: 6.62s    remaining: 972ms
872:    learn: 0.1986548        total: 6.63s    remaining: 965ms
873:    learn: 0.1985892        total: 6.64s    remaining: 957ms
874:    learn: 0.1985589        total: 6.65s    remaining: 950ms
875:    learn: 0.1984908        total: 6.66s    remaining: 942ms
876:    learn: 0.1984462        total: 6.66s    remaining: 935ms
877:    learn: 0.1984235        total: 6.67s    remaining: 927ms
878:    learn: 0.1983706        total: 6.68s    remaining: 919ms
```

```
879:    learn: 0.1983518        total: 6.68s    remaining: 911ms
880:    learn: 0.1983182        total: 6.69s    remaining: 904ms
881:    learn: 0.1982720        total: 6.7s     remaining: 896ms
882:    learn: 0.1982450        total: 6.71s    remaining: 889ms
883:    learn: 0.1981981        total: 6.71s    remaining: 881ms
884:    learn: 0.1981360        total: 6.72s    remaining: 873ms
885:    learn: 0.1981182        total: 6.73s    remaining: 866ms
886:    learn: 0.1980585        total: 6.74s    remaining: 858ms
887:    learn: 0.1979916        total: 6.74s    remaining: 851ms
888:    learn: 0.1979564        total: 6.75s    remaining: 843ms
889:    learn: 0.1979081        total: 6.76s    remaining: 836ms
890:    learn: 0.1978329        total: 6.77s    remaining: 828ms
891:    learn: 0.1977351        total: 6.78s    remaining: 821ms
892:    learn: 0.1976334        total: 6.78s    remaining: 813ms
893:    learn: 0.1975760        total: 6.79s    remaining: 805ms
894:    learn: 0.1975316        total: 6.8s     remaining: 798ms
895:    learn: 0.1974367        total: 6.81s    remaining: 791ms
896:    learn: 0.1973877        total: 6.82s    remaining: 783ms
897:    learn: 0.1973228        total: 6.83s    remaining: 776ms
898:    learn: 0.1972882        total: 6.84s    remaining: 768ms
899:    learn: 0.1971984        total: 6.85s    remaining: 761ms
900:    learn: 0.1971837        total: 6.85s    remaining: 753ms
901:    learn: 0.1970955        total: 6.86s    remaining: 745ms
902:    learn: 0.1970241        total: 6.87s    remaining: 738ms
903:    learn: 0.1969886        total: 6.88s    remaining: 730ms
904:    learn: 0.1969294        total: 6.88s    remaining: 723ms
905:    learn: 0.1968640        total: 6.89s    remaining: 715ms
906:    learn: 0.1967675        total: 6.9s     remaining: 707ms
907:    learn: 0.1967067        total: 6.9s     remaining: 700ms
908:    learn: 0.1966383        total: 6.91s    remaining: 692ms
909:    learn: 0.1965532        total: 6.92s    remaining: 684ms
910:    learn: 0.1964947        total: 6.93s    remaining: 677ms
911:    learn: 0.1964201        total: 6.93s    remaining: 669ms
912:    learn: 0.1963713        total: 6.94s    remaining: 661ms
913:    learn: 0.1962946        total: 6.95s    remaining: 654ms
914:    learn: 0.1962399        total: 6.96s    remaining: 646ms
915:    learn: 0.1962034        total: 6.96s    remaining: 638ms
916:    learn: 0.1961358        total: 6.97s    remaining: 631ms
917:    learn: 0.1960888        total: 6.98s    remaining: 623ms
918:    learn: 0.1960658        total: 6.98s    remaining: 616ms
919:    learn: 0.1959992        total: 6.99s    remaining: 608ms
920:    learn: 0.1959438        total: 7s       remaining: 600ms
921:    learn: 0.1959064        total: 7.01s    remaining: 593ms
922:    learn: 0.1958504        total: 7.01s    remaining: 585ms
923:    learn: 0.1957574        total: 7.02s    remaining: 578ms
924:    learn: 0.1956765        total: 7.03s    remaining: 570ms
925:    learn: 0.1955849        total: 7.04s    remaining: 563ms
926:    learn: 0.1955103        total: 7.05s    remaining: 555ms
```

```
927:    learn: 0.1954856        total: 7.05s    remaining: 547ms
928:    learn: 0.1954225        total: 7.06s    remaining: 540ms
929:    learn: 0.1953750        total: 7.07s    remaining: 532ms
930:    learn: 0.1953372        total: 7.08s    remaining: 524ms
931:    learn: 0.1952841        total: 7.08s    remaining: 517ms
932:    learn: 0.1952177        total: 7.09s    remaining: 509ms
933:    learn: 0.1952035        total: 7.1s     remaining: 502ms
934:    learn: 0.1951587        total: 7.1s     remaining: 494ms
935:    learn: 0.1951192        total: 7.11s    remaining: 486ms
936:    learn: 0.1950938        total: 7.12s    remaining: 479ms
937:    learn: 0.1950349        total: 7.13s    remaining: 471ms
938:    learn: 0.1949769        total: 7.13s    remaining: 463ms
939:    learn: 0.1949355        total: 7.14s    remaining: 456ms
940:    learn: 0.1948807        total: 7.15s    remaining: 448ms
941:    learn: 0.1948386        total: 7.15s    remaining: 441ms
942:    learn: 0.1947642        total: 7.16s    remaining: 433ms
943:    learn: 0.1947347        total: 7.17s    remaining: 425ms
944:    learn: 0.1946423        total: 7.18s    remaining: 418ms
945:    learn: 0.1945500        total: 7.18s    remaining: 410ms
946:    learn: 0.1945337        total: 7.19s    remaining: 403ms
947:    learn: 0.1944702        total: 7.2s     remaining: 395ms
948:    learn: 0.1944289        total: 7.21s    remaining: 387ms
949:    learn: 0.1943834        total: 7.22s    remaining: 380ms
950:    learn: 0.1943278        total: 7.22s    remaining: 372ms
951:    learn: 0.1942464        total: 7.23s    remaining: 365ms
952:    learn: 0.1941656        total: 7.24s    remaining: 357ms
953:    learn: 0.1940940        total: 7.25s    remaining: 349ms
954:    learn: 0.1939997        total: 7.25s    remaining: 342ms
955:    learn: 0.1939533        total: 7.26s    remaining: 334ms
956:    learn: 0.1938963        total: 7.27s    remaining: 327ms
957:    learn: 0.1938772        total: 7.28s    remaining: 319ms
958:    learn: 0.1937952        total: 7.28s    remaining: 311ms
959:    learn: 0.1937003        total: 7.29s    remaining: 304ms
960:    learn: 0.1936736        total: 7.3s     remaining: 296ms
961:    learn: 0.1935899        total: 7.31s    remaining: 289ms
962:    learn: 0.1935624        total: 7.31s    remaining: 281ms
963:    learn: 0.1934844        total: 7.32s    remaining: 273ms
964:    learn: 0.1933919        total: 7.33s    remaining: 266ms
965:    learn: 0.1933742        total: 7.33s    remaining: 258ms
966:    learn: 0.1933223        total: 7.34s    remaining: 251ms
967:    learn: 0.1932505        total: 7.35s    remaining: 243ms
968:    learn: 0.1931841        total: 7.36s    remaining: 235ms
969:    learn: 0.1930994        total: 7.36s    remaining: 228ms
970:    learn: 0.1930722        total: 7.38s    remaining: 220ms
971:    learn: 0.1930198        total: 7.38s    remaining: 213ms
972:    learn: 0.1929849        total: 7.39s    remaining: 205ms
973:    learn: 0.1929237        total: 7.4s     remaining: 197ms
974:    learn: 0.1928613        total: 7.41s    remaining: 190ms
```

```
975:    learn: 0.1928279        total: 7.41s    remaining: 182ms
976:    learn: 0.1927571        total: 7.42s    remaining: 175ms
977:    learn: 0.1926480        total: 7.43s    remaining: 167ms
978:    learn: 0.1926139        total: 7.43s    remaining: 160ms
979:    learn: 0.1925261        total: 7.44s    remaining: 152ms
980:    learn: 0.1924614        total: 7.45s    remaining: 144ms
981:    learn: 0.1924206        total: 7.46s    remaining: 137ms
982:    learn: 0.1924062        total: 7.46s    remaining: 129ms
983:    learn: 0.1923406        total: 7.47s    remaining: 121ms
984:    learn: 0.1922904        total: 7.48s    remaining: 114ms
985:    learn: 0.1922338        total: 7.48s    remaining: 106ms
986:    learn: 0.1921696        total: 7.49s    remaining: 98.7ms
987:    learn: 0.1920920        total: 7.5s     remaining: 91.1ms
988:    learn: 0.1920267        total: 7.51s    remaining: 83.5ms
989:    learn: 0.1919298        total: 7.52s    remaining: 75.9ms
990:    learn: 0.1919051        total: 7.52s    remaining: 68.3ms
991:    learn: 0.1918527        total: 7.53s    remaining: 60.7ms
992:    learn: 0.1917804        total: 7.54s    remaining: 53.1ms
993:    learn: 0.1916893        total: 7.55s    remaining: 45.6ms
994:    learn: 0.1916735        total: 7.55s    remaining: 38ms
995:    learn: 0.1916198        total: 7.56s    remaining: 30.4ms
996:    learn: 0.1915641        total: 7.57s    remaining: 22.8ms
997:    learn: 0.1915181        total: 7.58s    remaining: 15.2ms
998:    learn: 0.1914961        total: 7.58s    remaining: 7.59ms
999:    learn: 0.1914472        total: 7.59s    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')
```

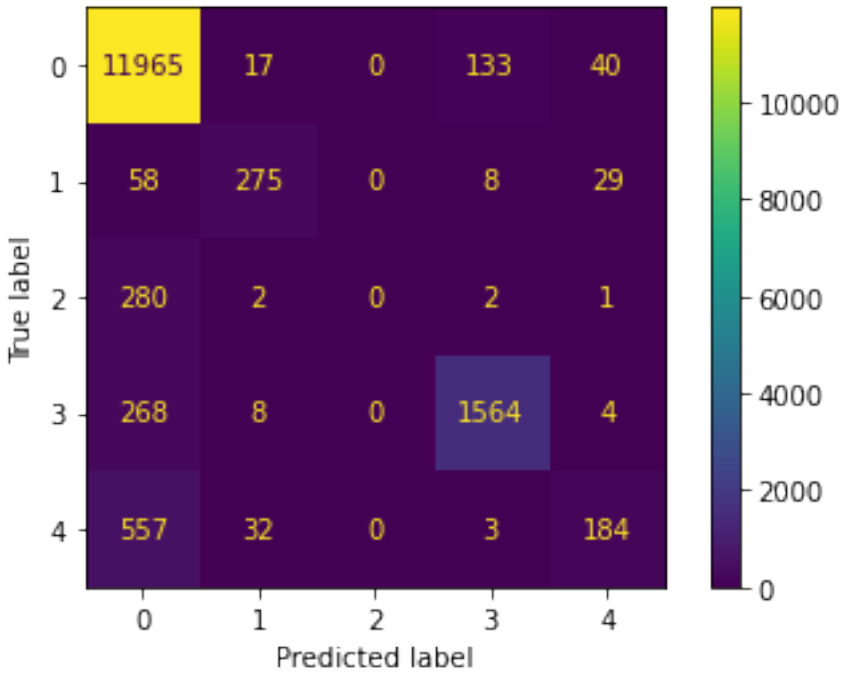
[105]:
```
y_pred = vot_soft.predict(X_test)
```

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

[106]:
```
90.6545690213869
```

[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
0x7fcc0f6a4dc0>
```



[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
```

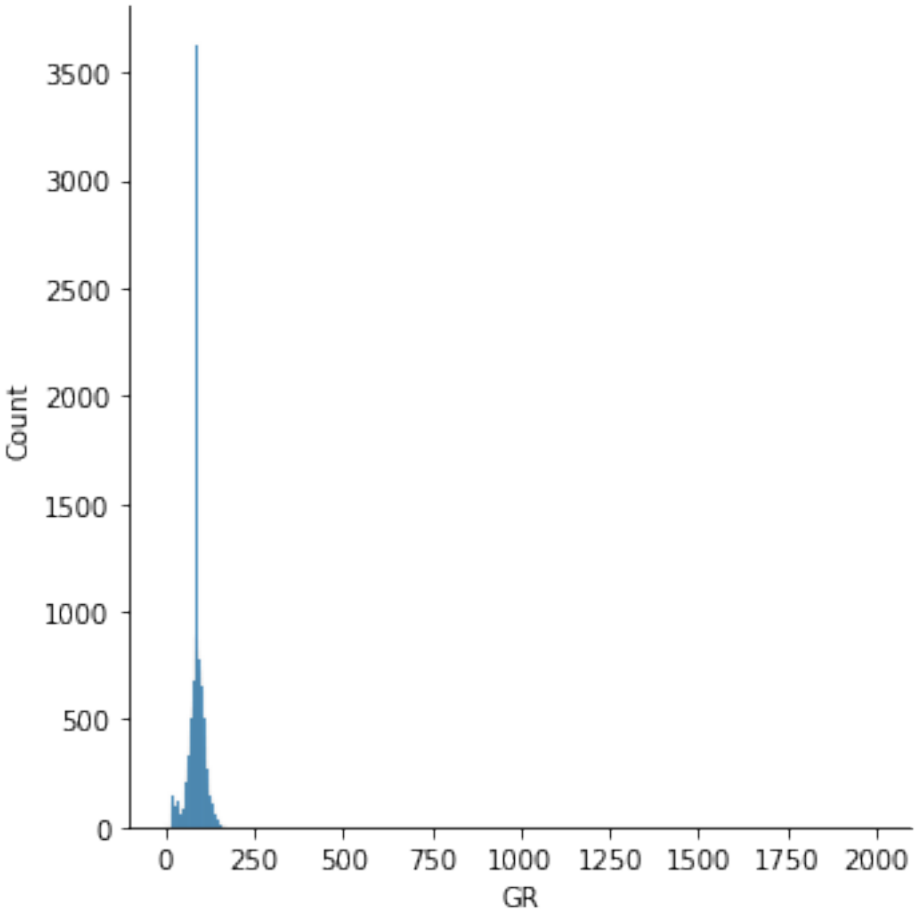
```
0
Graph (GR) after filling null values with mean
Graph (NPHI) after filling null values with mean
0
DT        0
GR        0
NPHI      0
RHOB      0
FACIES    0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
Int64Index: 25801 entries, 643 to 29515
Data columns (total 5 columns):
 #   Column  Non-Null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

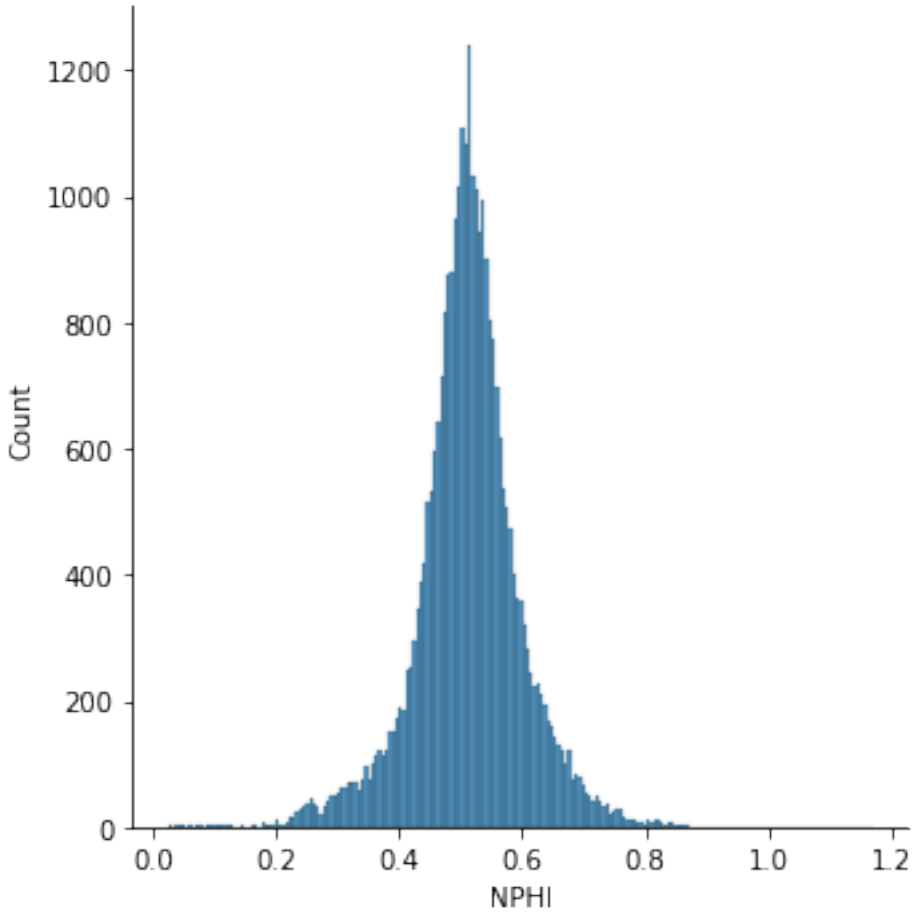
```
---  ------  --------------  -----
 0   DT      25801 non-null  float64
 1   GR      25801 non-null  float64
 2   NPHI    25801 non-null  float64
 3   RHOB    25801 non-null  float64
 4   FACIES  25801 non-null  int64
dtypes: float64(4), int64(1)
memory usage: 1.2 MB
```

[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
4 standard deviation outliers -:
Empty DataFrame
Columns: [DT, GR, NPHI, RHOB, FACIES]
Index: []
(0, 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

[25801 rows x 5 columns]
column GR
4 standard deviation outliers -:
Empty DataFrame
Columns: [DT, GR, NPHI, RHOB, FACIES]
Index: []
(0, 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

[25801 rows x 5 columns]
column NPHI
4 standard deviation outliers -:
             DT       GR    NPHI    RHOB  FACIES
3668   112.0577  57.4443  0.1480  1.8899       1
3669   106.4163  53.5238  0.1198  1.8785       1
3670   101.4661  52.0916  0.0936  1.8735       1
3671    99.3440  51.7385  0.0687  1.8693       1
3672    99.3754  51.6659  0.0494  1.8639       1
…         …       …      …       …      …
25371  109.8243  55.4493  0.0941  2.0305       1
25372  111.2239  52.5198  0.0989  2.0335       1
25373  112.9419  53.3644  0.1088  2.0729       1
25374  114.6335  58.9418  0.1227  2.1418       1
25375  115.8208  69.8713  0.1452  2.2079       1

[63 rows x 5 columns]
(63, 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

[25738 rows x 5 columns]
column RHOB
4 standard deviation outliers -:
Empty DataFrame
Columns: [DT, GR, NPHI, RHOB, FACIES]
Index: []
(0, 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

[25738 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      0.698565  0.380866  0.697426  0.485305       0
1      0.693807  0.386310  0.740892  0.450650       0
2      0.706670  0.388284  0.733014  0.376259       3
3      0.729742  0.350357  0.728654  0.276872       3
4      0.746233  0.280139  0.763539  0.193509       3
…          …         …         …         …      …
25733  0.497633  0.507280  0.446336  0.562580       0
25734  0.505571  0.491298  0.436208  0.563084       0
25735  0.517405  0.472802  0.430018  0.566151       0
```

```
25736  0.536043  0.456455  0.454213  0.571461       0
25737  0.546172  0.444417  0.497398  0.578465       0

[25738 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      0.698565  0.380866  0.697426  0.485305
1      0.693807  0.386310  0.740892  0.450650
2      0.706670  0.388284  0.733014  0.376259
3      0.729742  0.350357  0.728654  0.276872
4      0.746233  0.280139  0.763539  0.193509
…         …         …         …         …
25733  0.497633  0.507280  0.446336  0.562580
25734  0.505571  0.491298  0.436208  0.563084
25735  0.517405  0.472802  0.430018  0.566151
25736  0.536043  0.456455  0.454213  0.571461
25737  0.546172  0.444417  0.497398  0.578465

[25738 rows x 4 columns]
```

[135]:
```
y_testing
```

[135]:
```
       FACIES
0           0
1           0
2           3
3           3
4           3
…         …
25733       0
25734       0
```

```
25735       0
25736       0
25737       0

[25738 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
```

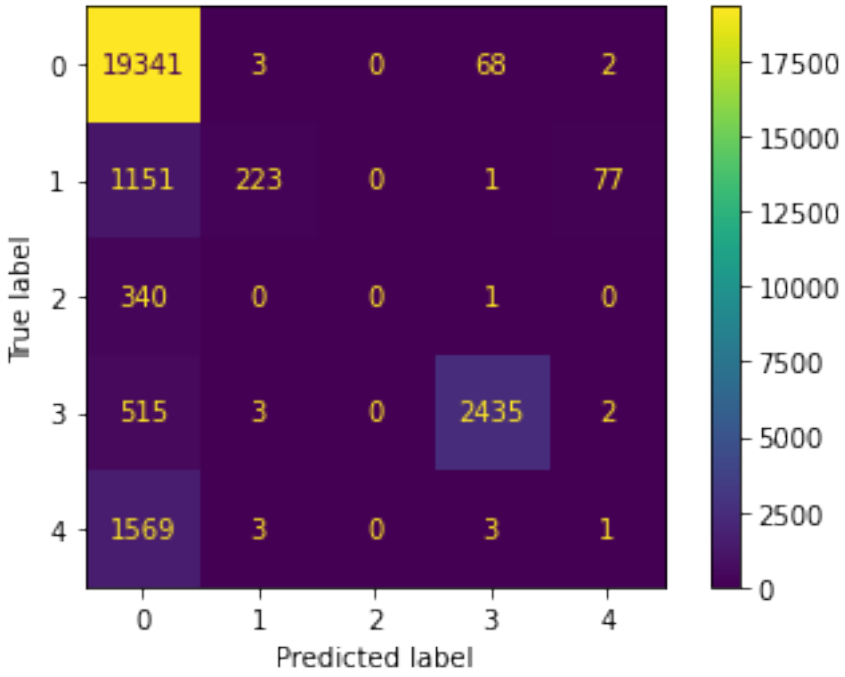
[138]:
```
85.47672701841636
```

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

[139]:
```
array([[19341,     3,     0,    68,     2],
       [ 1151,   223,     0,     1,    77],
       [  340,     0,     0,     1,     0],
       [  515,     3,     0,  2435,     2],
       [ 1569,     3,     0,     3,     1]])
```

[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
0x7fcc037a4c70>
```



[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)
```

[ ]:
```

```