main_new_5_1_0_0

August 28, 2021

1 IMPORTANT LIBRARIES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2 LAS TO CSV

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

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

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

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

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

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

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

3 IMPUTATION

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

[18]:

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

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

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

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

[58499 rows x 67 columns]
```

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

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

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

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

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

[20]: 67

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

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

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

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

[23]:
```
df
```

[23]:

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

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

[58499 rows x 5 columns]
```

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

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

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

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

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

[27]:

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

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

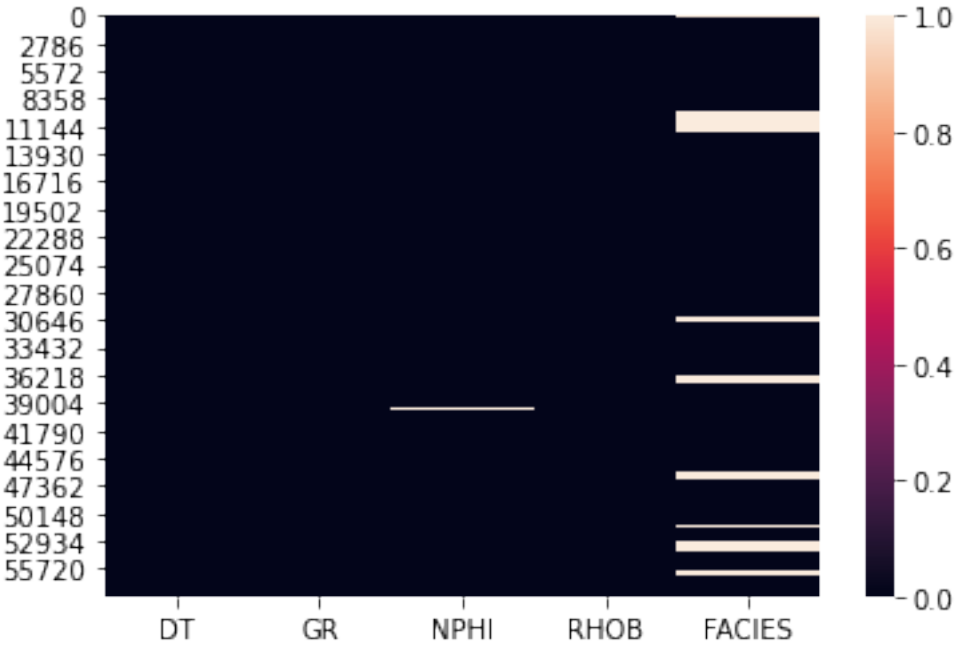
[28]:
```
(58499, 5)
```

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

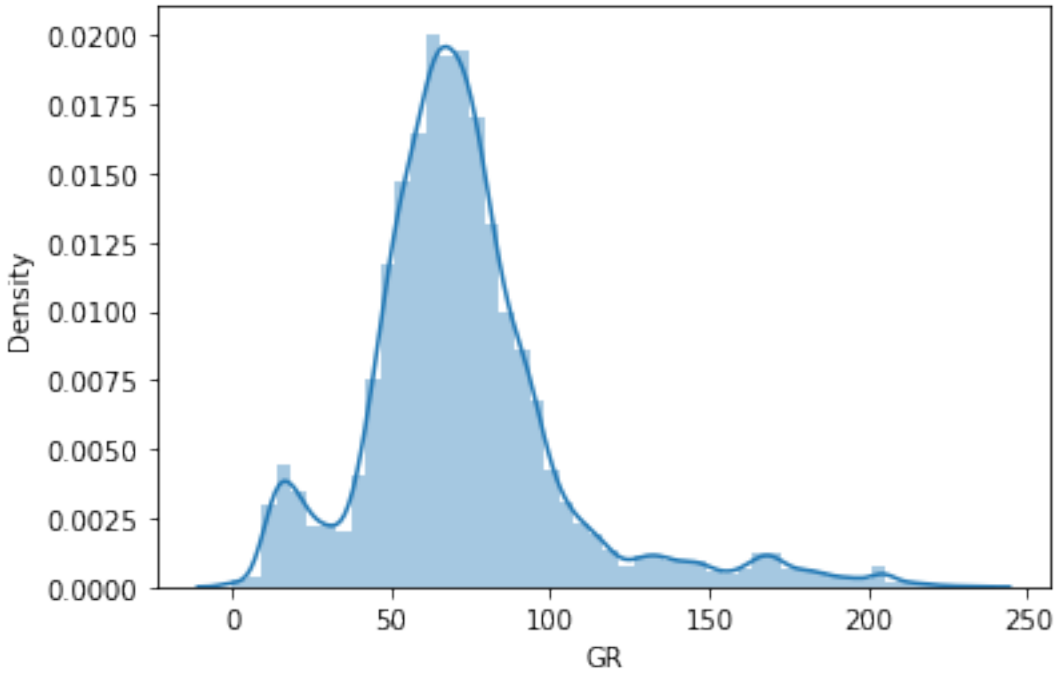
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DT      58499 non-null  float64
 1   GR      58379 non-null  float64
 2   NPHI    58172 non-null  float64
 3 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

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

[30]: <AxesSubplot:>



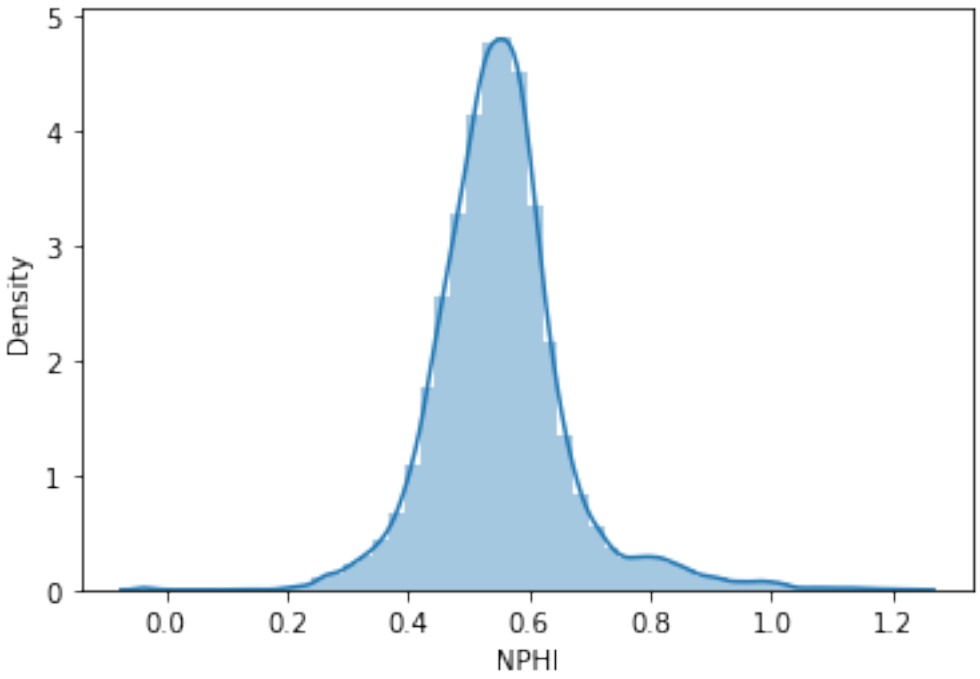
[31]:
```
null_gr = sns.distplot(df.GR.dropna())
```



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

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

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

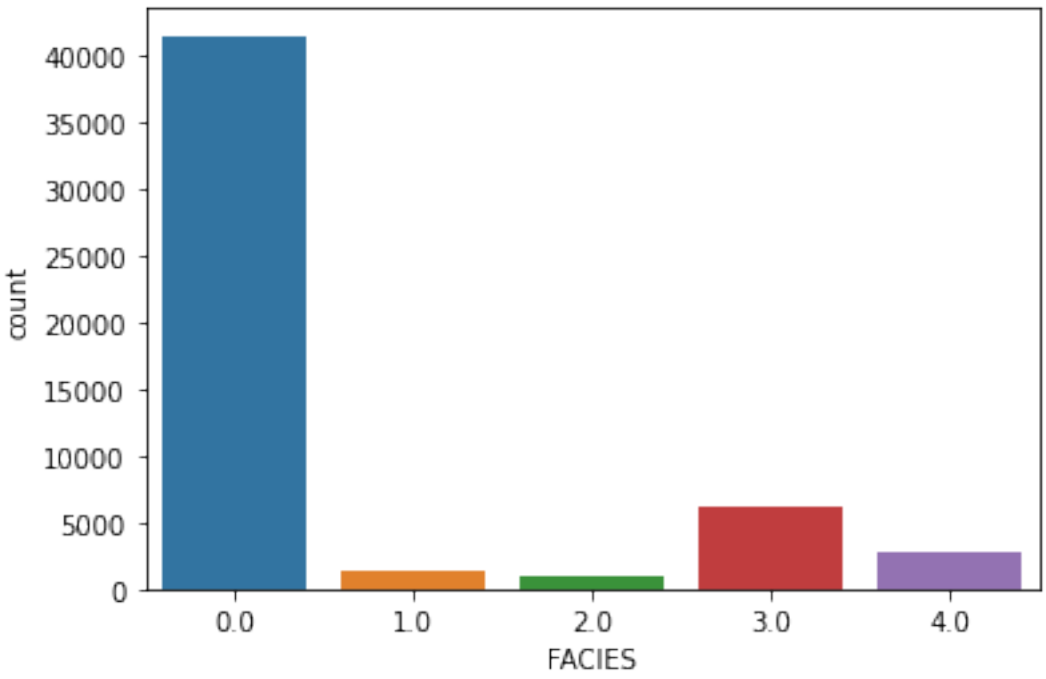


[34]:
```
df.NPHI.describe()
```

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

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

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



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

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

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

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

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

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

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

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

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

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

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

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

[38]:
```
imputation_strategy = ["Mean" , "bffill" , "KNNImputer" , "IterativeImputer" ,␣
↪"KNNimputer_floor" , "IterativeImputer_floor" , "KNNBinning","dropna"]
#select option from 0-7 (6 is experimental)
optionimputation=5
```

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

[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'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   DT      58499 non-null  float64
 1   GR      58499 non-null  float64
 2   NPHI    58499 non-null  float64
 3   RHOB    58499 non-null  float64
 4   FACIES  58499 non-null  int64
dtypes: float64(4), int64(1)
memory usage: 2.2 MB
```

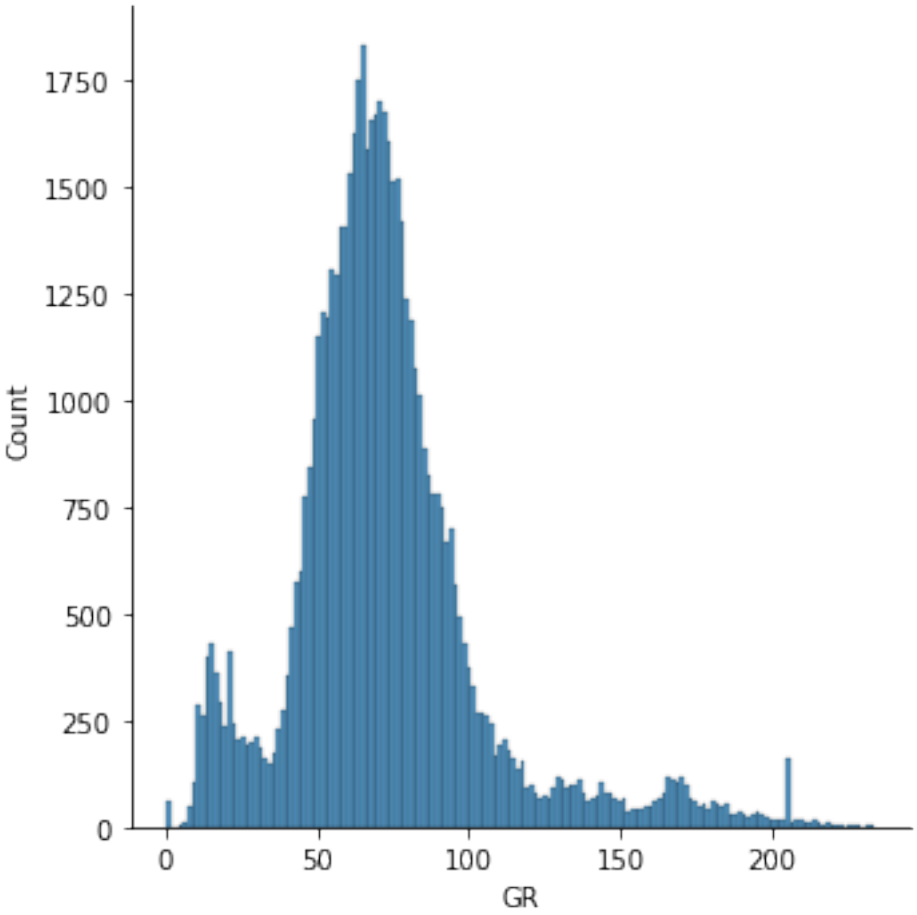
[41]:
```
df.isnull().sum()
```

[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())
```

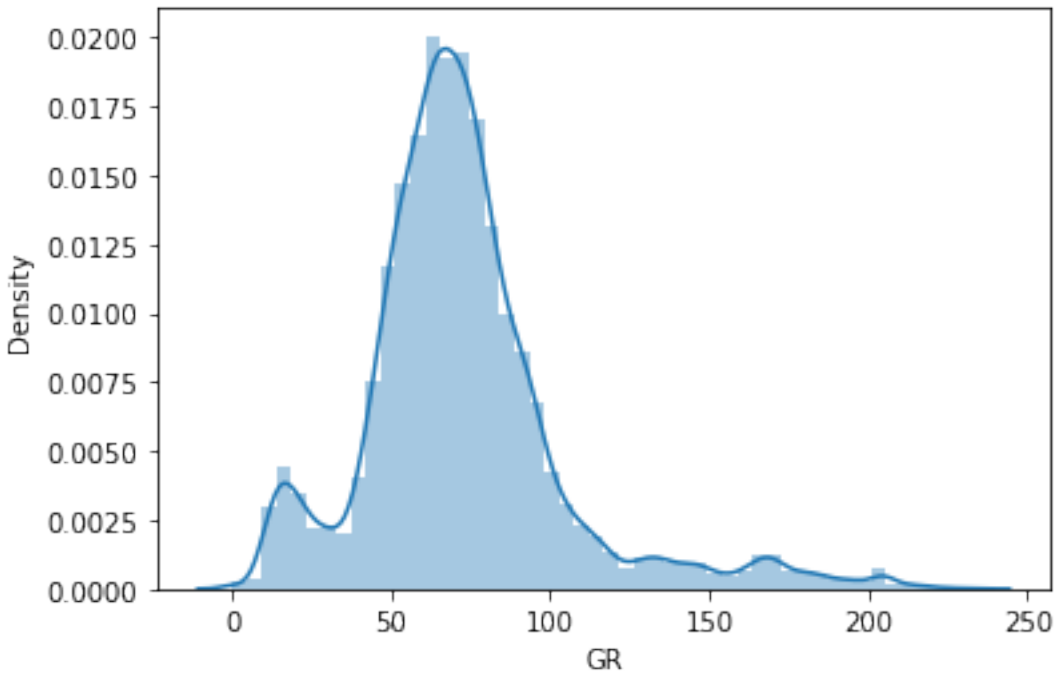
[43]:
```
<seaborn.axisgrid.FacetGrid at 0x7fcaa3237bb0>
```



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

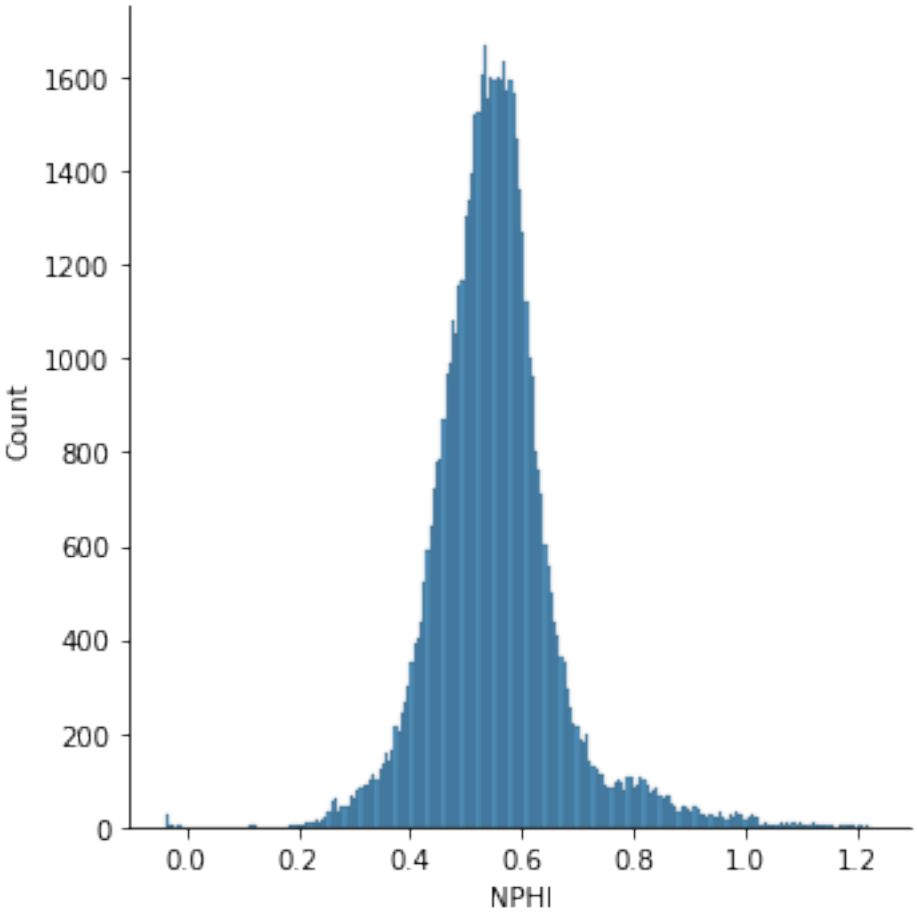
```
WHEN GR WAS NULL
```

[44]:



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

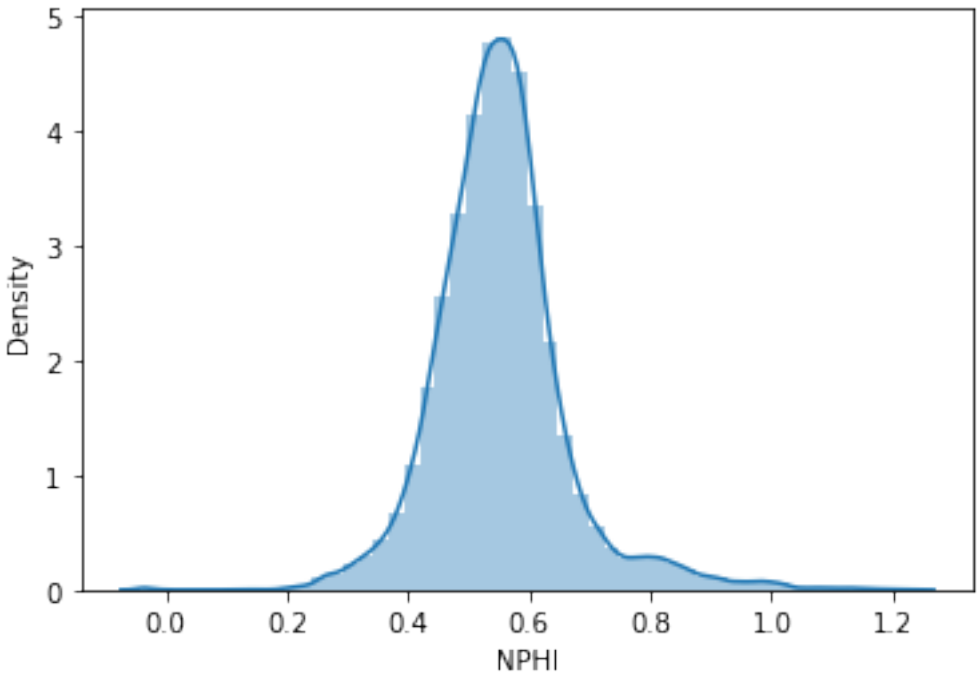
[45]: <seaborn.axisgrid.FacetGrid at 0x7fcaa2e09850>



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

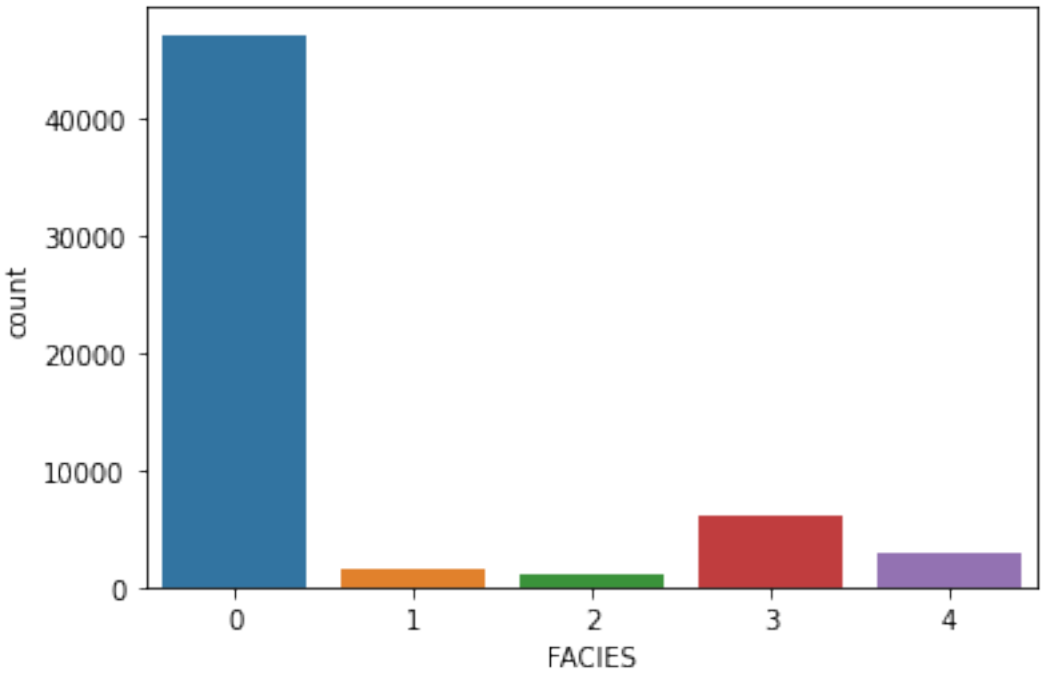
```
WHEN NPHI WAS NULL
```

[46]:



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

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



4 DATA CONDITIONING / OUTLIER REMOVAL

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

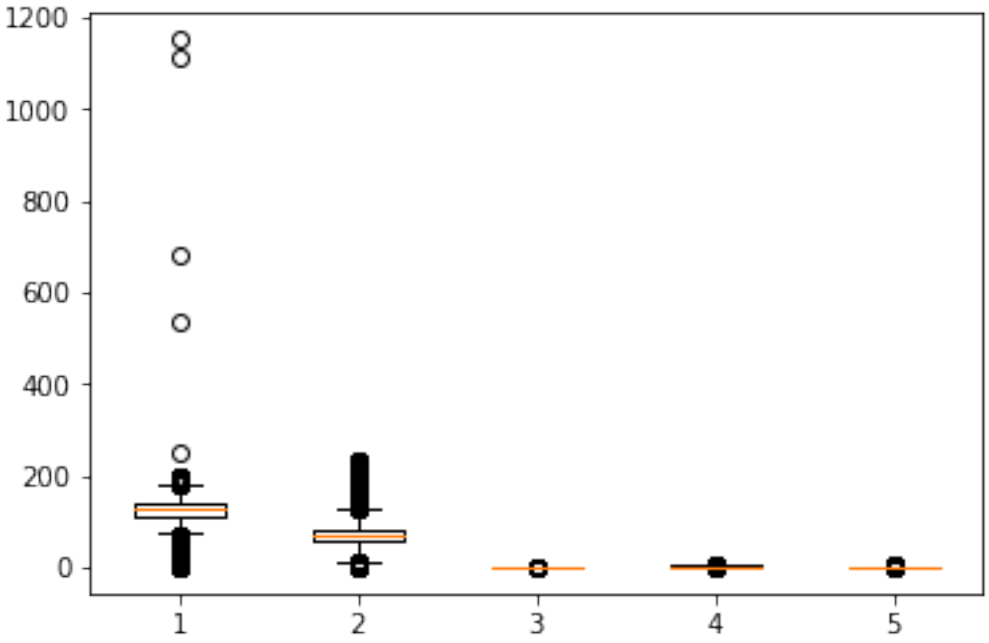
[48]:
```
<bound method NDFrame.head of              DT         GR    NPHI    RHOB  FACIES
0       50.2544  50.212800  0.5340  2.1228       1
1       50.3881  49.750900  0.5316  2.1250       1
2       49.8852  48.251300  0.5126  2.1316       1
3       49.9032  46.821200  0.5137  2.1437       1
4       50.0157  45.346300  0.5472  2.1611       1
…          …        …       …       …       …
58494  123.7404  80.913653  0.4993  2.4639       0
58495  123.8728  82.952576  0.5313  2.4660       0
58496  123.3722  84.044079  0.5448  2.4714       0
58497  122.6038  83.725389  0.5364  2.4750       0
58498  122.3045  83.329152  0.5331  2.4709       0

[58499 rows x 5 columns]>
```

4.1 WHOLE DATA OUTLIER VISUALIZATION

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

[49]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcaa27cc220>,
  <matplotlib.lines.Line2D at 0x7fcaa27cc5b0>,
  <matplotlib.lines.Line2D at 0x7fcaa27d8b80>,
  <matplotlib.lines.Line2D at 0x7fcaa27d8f10>,
  <matplotlib.lines.Line2D at 0x7fcaa27f04f0>,
  <matplotlib.lines.Line2D at 0x7fcaa27f0880>,
  <matplotlib.lines.Line2D at 0x7fcaa27fae20>,
  <matplotlib.lines.Line2D at 0x7fcaa27431f0>,
  <matplotlib.lines.Line2D at 0x7fcaa274f790>,
  <matplotlib.lines.Line2D at 0x7fcaa274fb20>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcaa27cc940>,
  <matplotlib.lines.Line2D at 0x7fcaa27cccd0>,
  <matplotlib.lines.Line2D at 0x7fcaa27e32e0>,
  <matplotlib.lines.Line2D at 0x7fcaa27e3670>,
  <matplotlib.lines.Line2D at 0x7fcaa27f0c10>,
  <matplotlib.lines.Line2D at 0x7fcaa27f0fa0>,
  <matplotlib.lines.Line2D at 0x7fcaa2743580>,
  <matplotlib.lines.Line2D at 0x7fcaa2743910>,
  <matplotlib.lines.Line2D at 0x7fcaa274feb0>,
  <matplotlib.lines.Line2D at 0x7fcaa275a280>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcaa287de50>,
  <matplotlib.lines.Line2D at 0x7fcaa27d87f0>,
```

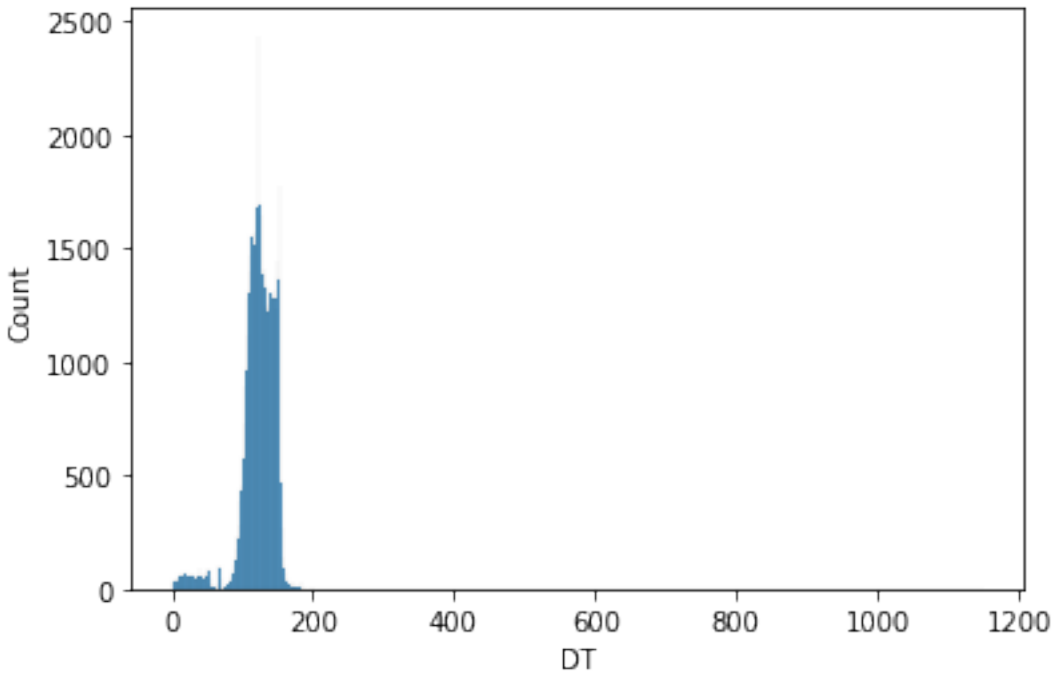
```
 <matplotlib.lines.Line2D at 0x7fcaa27f0160>,
 <matplotlib.lines.Line2D at 0x7fcaa27faa90>,
 <matplotlib.lines.Line2D at 0x7fcaa274f400>],
'medians': [<matplotlib.lines.Line2D at 0x7fcaa27d80a0>,
 <matplotlib.lines.Line2D at 0x7fcaa27e3a00>,
 <matplotlib.lines.Line2D at 0x7fcaa27fa370>,
 <matplotlib.lines.Line2D at 0x7fcaa2743ca0>,
 <matplotlib.lines.Line2D at 0x7fcaa275a610>],
'fliers': [<matplotlib.lines.Line2D at 0x7fcaa27d8430>,
 <matplotlib.lines.Line2D at 0x7fcaa27e3d90>,
 <matplotlib.lines.Line2D at 0x7fcaa27fa700>,
 <matplotlib.lines.Line2D at 0x7fcaa274f070>,
 <matplotlib.lines.Line2D at 0x7fcaa275a9d0>],
'means': []}
```



4.2 DT VISUALIZATION

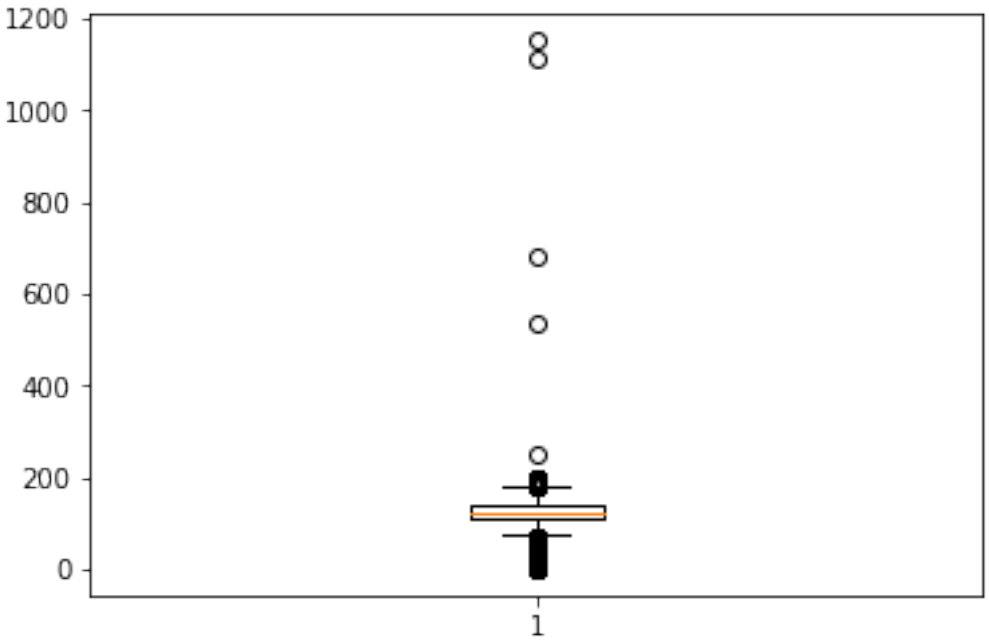
[50]:
```
sns.histplot(df.DT)
```

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



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

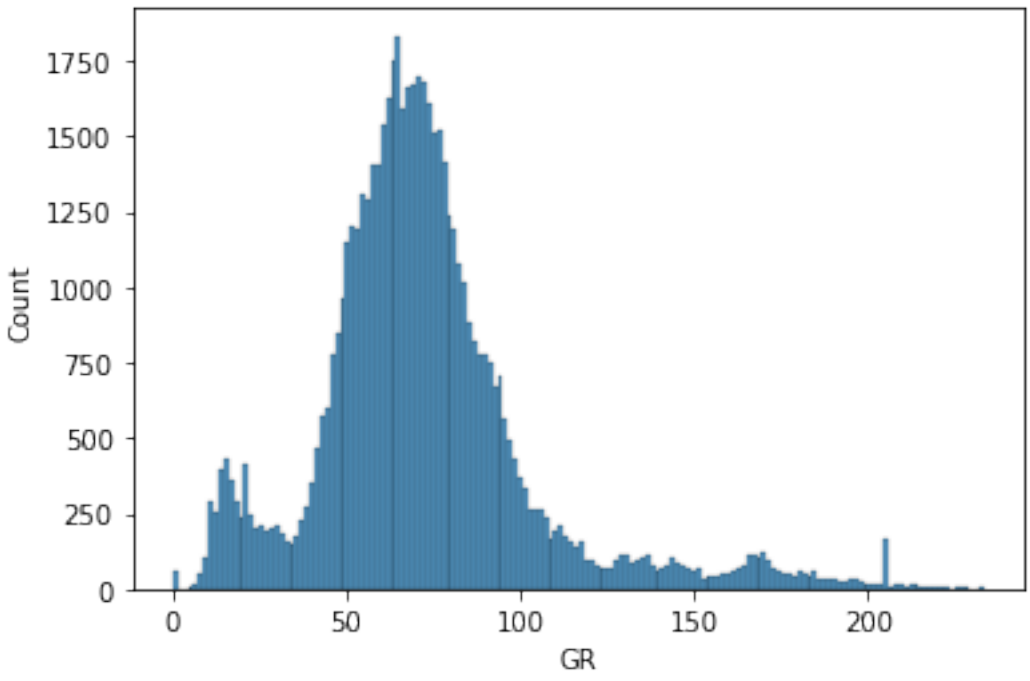
[51]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fcaa0f6b1c0>,
  <matplotlib.lines.Line2D at 0x7fcaa0f6b550>],
 'caps': [<matplotlib.lines.Line2D at 0x7fcaa0f6b8e0>,
  <matplotlib.lines.Line2D at 0x7fcaa0f6bc70>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fcaa0f60df0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fcaa0f76040>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fcaa0f763d0>],
 'means': []}
```



4.3 GR VISUALIZATION

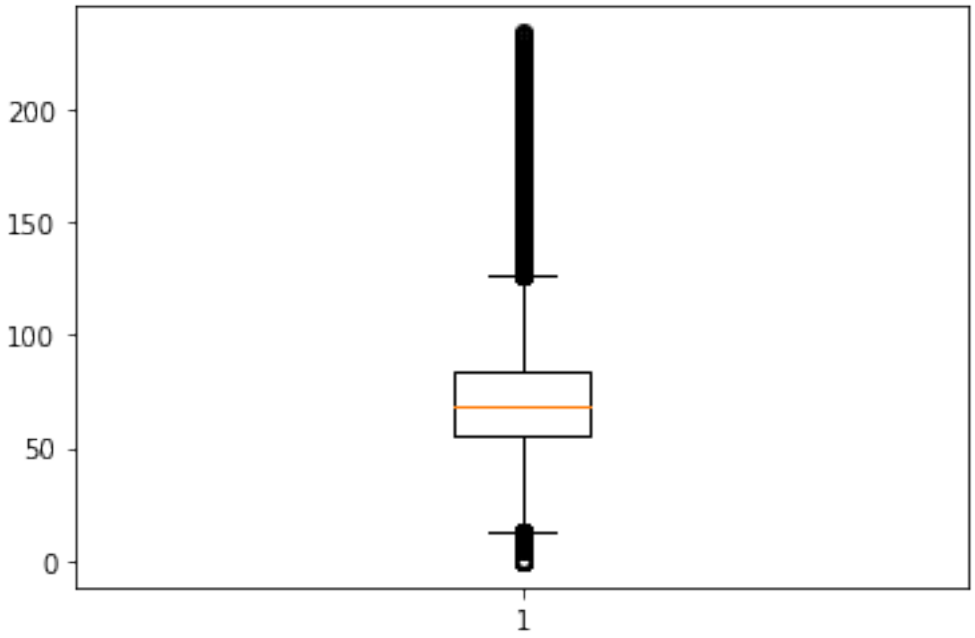
[52]:
```
sns.histplot(df.GR)
```

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



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

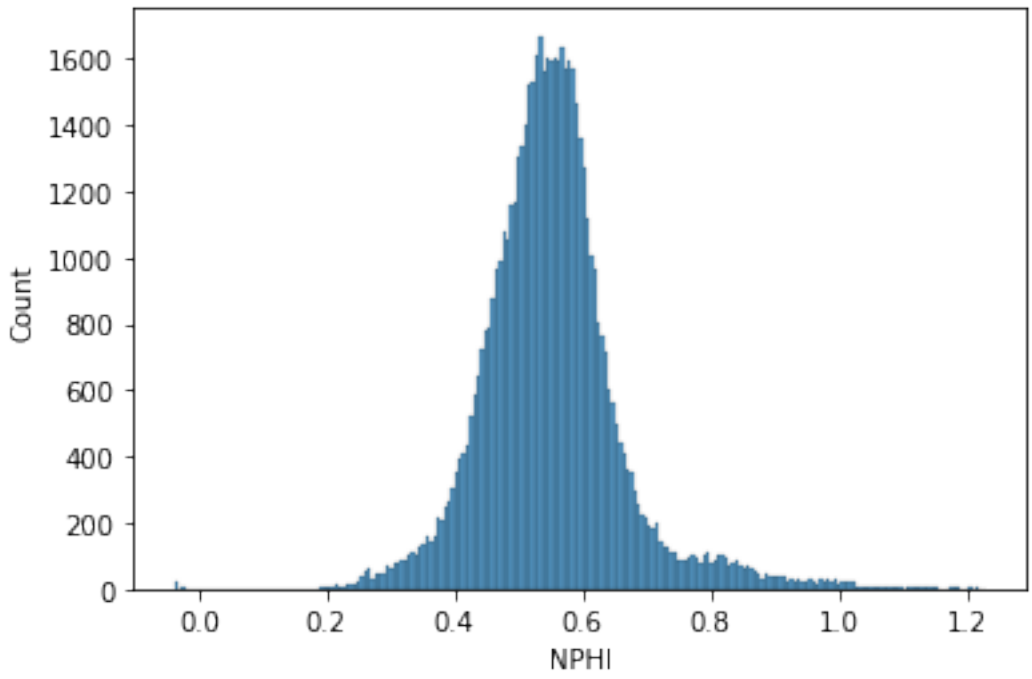
[53]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca948e44f0>,
  <matplotlib.lines.Line2D at 0x7fca948e4880>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca948e4c10>,
  <matplotlib.lines.Line2D at 0x7fca948e4fa0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca948e4160>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca948ef370>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca948ef700>],
 'means': []}
```



4.4 NPHI VISUALIZATION

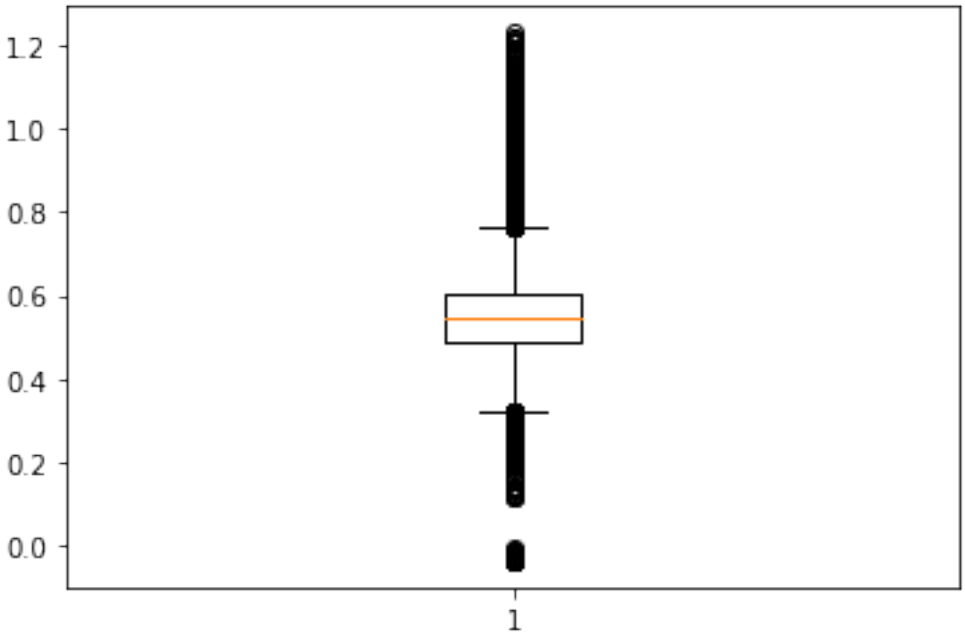
[54]:
```
sns.histplot(df.NPHI)
```

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



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

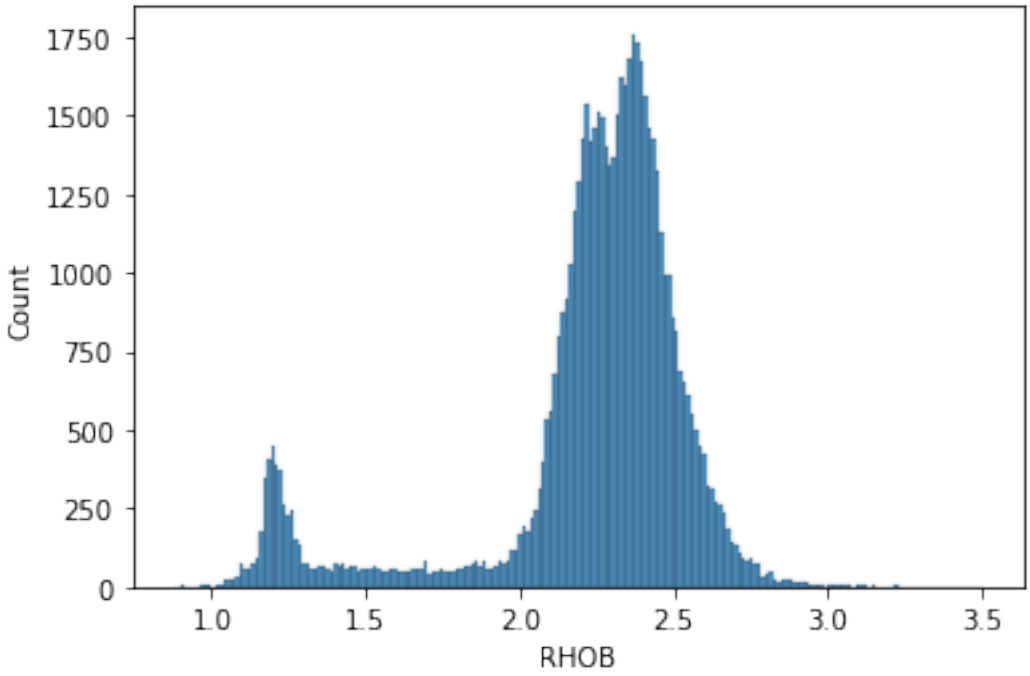
[55]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca945ca3a0>,
  <matplotlib.lines.Line2D at 0x7fca945ca730>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca945caac0>,
  <matplotlib.lines.Line2D at 0x7fca945cae50>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca945bbfd0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca945d3220>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca945d35b0>],
 'means': []}
```



4.5 RHOB VISUALIZATION

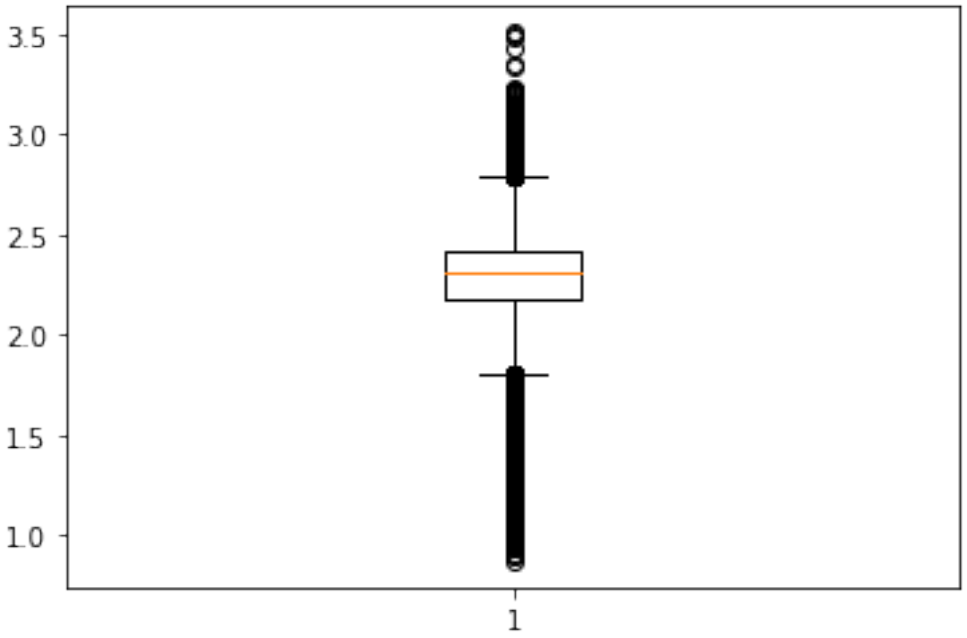
[56]:
```
sns.histplot(df.RHOB)
```

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



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

[57]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca943053d0>,
  <matplotlib.lines.Line2D at 0x7fca94305760>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca94305af0>,
  <matplotlib.lines.Line2D at 0x7fca94305e80>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca94305040>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca94310250>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca943105e0>],
 '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) & (df.
↪FACIES >= 0) & (df.FACIES <= 4)]
            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) & (df.
↪FACIES >= 0) & (df.FACIES <= 4)]
            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) & (df.
↪FACIES >= 0) & (df.FACIES <= 4)]
            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
532      18.8077  68.6271  0.3279  2.3455       0
553      15.8999  63.5563  0.3764  2.5182       0
554       8.6395  61.5439  0.3675  2.5916       0
555       3.1202  60.7632  0.3411  2.6241       0
556       4.3432  60.9371  0.3120  2.5905       0
…            …       …       …       …     …
4460   1150.8206  93.6033  0.6520  1.8355       0
4461   1109.5558  93.6033  0.6349  1.8700       0
4462    535.0460  93.6033  0.6083  1.8946       0
43457     6.1411  76.3710  0.5301  2.0501       0
43535    14.3304  34.9702  0.5064  2.1492       0

[685 rows x 5 columns]
(685, 5)
             DT         GR    NPHI    RHOB  FACIES
0       50.2544  50.212800  0.5340  2.1228       1
1       50.3881  49.750900  0.5316  2.1250       1
2       49.8852  48.251300  0.5126  2.1316       1
3       49.9032  46.821200  0.5137  2.1437       1
4       50.0157  45.346300  0.5472  2.1611       1
…          …        …       …       …     …
58494  123.7404  80.913653  0.4993  2.4639       0
58495  123.8728  82.952576  0.5313  2.4660       0
58496  123.3722  84.044079  0.5448  2.4714       0
58497  122.6038  83.725389  0.5364  2.4750       0
58498  122.3045  83.329152  0.5331  2.4709       0

[57814 rows x 5 columns]
column GR
4 standard deviation outliers -:
             DT        GR      NPHI    RHOB  FACIES
38403  133.9539  207.7189  0.662800  2.3796       0
38404  134.4976  208.2332  0.668500  2.3742       0
38405  136.0232  202.0813  0.643600  2.3345       0
38411  141.0677  205.0823  0.642200  2.3929       0
38412  130.4464  214.7148  0.566200  2.5062       0
…         …        …        …       …     …
39776  125.9000  204.7348  0.589430  2.4665       0
39777  125.9000  204.7348  0.593017  2.4427       0
39778  125.9000  204.7348  0.594706  2.4315       0
39779  125.9000  204.7348  0.595143  2.4286       0
39780  125.9000  204.7348  0.594826  2.4307       0

[345 rows x 5 columns]
(345, 5)
             DT         GR    NPHI    RHOB  FACIES
```

```
0       50.2544  50.212800  0.5340  2.1228       1
1       50.3881  49.750900  0.5316  2.1250       1
2       49.8852  48.251300  0.5126  2.1316       1
3       49.9032  46.821200  0.5137  2.1437       1
4       50.0157  45.346300  0.5472  2.1611       1
…          …        …      …       …     …
58494  123.7404  80.913653  0.4993  2.4639       0
58495  123.8728  82.952576  0.5313  2.4660       0
58496  123.3722  84.044079  0.5448  2.4714       0
58497  122.6038  83.725389  0.5364  2.4750       0
58498  122.3045  83.329152  0.5331  2.4709       0

[57469 rows x 5 columns]
column NPHI
4 standard deviation outliers -:
             DT       GR    NPHI    RHOB  FACIES
4032   151.5302  12.4220  0.9888  1.2064       3
4033   151.8671  12.5059  1.0006  1.1972       3
4227   152.9710  14.5097  0.9899  1.1861       0
4228   152.9596  14.3802  0.9912  1.1828       0
8721   150.7242  16.0597  1.0039  1.2529       3
…         …       …      …       …     …
52857  113.3730  63.3097  0.9897  2.3121       0
52860  113.3730  63.3097  0.9888  2.4878       0
52861  113.3730  63.3097  0.9949  2.5784       0
52862  113.3730  63.3097  0.9980  2.6148       0
52863  113.3730  63.3097  0.9951  2.6281       0

[330 rows x 5 columns]
(330, 5)
             DT         GR    NPHI    RHOB  FACIES
0       50.2544  50.212800  0.5340  2.1228       1
1       50.3881  49.750900  0.5316  2.1250       1
2       49.8852  48.251300  0.5126  2.1316       1
3       49.9032  46.821200  0.5137  2.1437       1
4       50.0157  45.346300  0.5472  2.1611       1
…          …        …      …       …     …
58494  123.7404  80.913653  0.4993  2.4639       0
58495  123.8728  82.952576  0.5313  2.4660       0
58496  123.3722  84.044079  0.5448  2.4714       0
58497  122.6038  83.725389  0.5364  2.4750       0
58498  122.3045  83.329152  0.5331  2.4709       0

[57139 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	50.2544	50.212800	0.5340	2.1228	1
1	50.3881	49.750900	0.5316	2.1250	1
2	49.8852	48.251300	0.5126	2.1316	1
3	49.9032	46.821200	0.5137	2.1437	1
4	50.0157	45.346300	0.5472	2.1611	1
…	…	…	…	…	…
58494	123.7404	80.913653	0.4993	2.4639	0
58495	123.8728	82.952576	0.5313	2.4660	0
58496	123.3722	84.044079	0.5448	2.4714	0
58497	122.6038	83.725389	0.5364	2.4750	0
58498	122.3045	83.329152	0.5331	2.4709	0

```
[57139 rows x 5 columns]
```

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

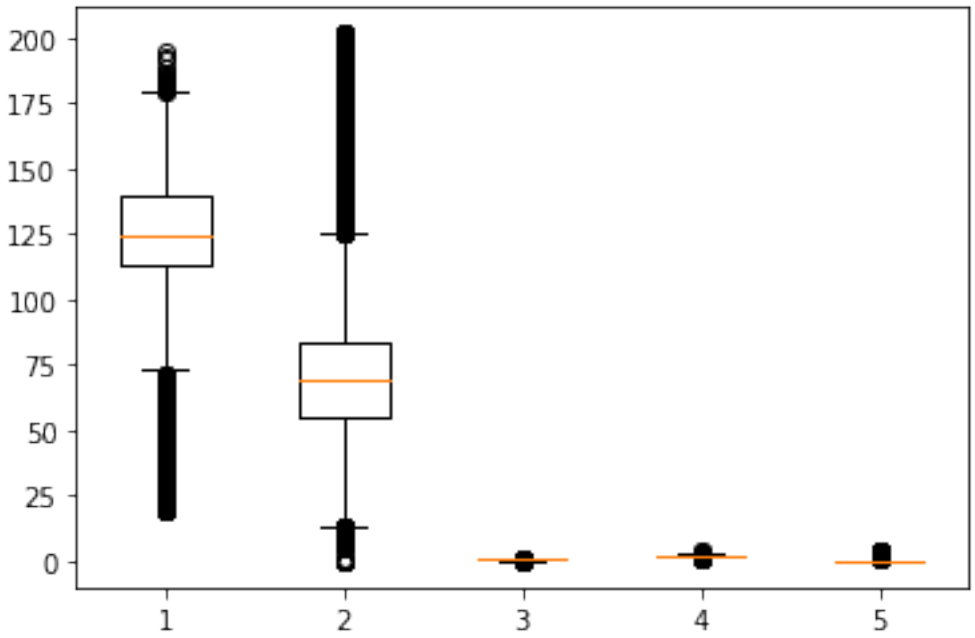
[60]:
```
(57139, 5)
```

4.6 WHOLE DATA AFTER REMOVING OUTLIERS

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

[61]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca942ecc70>,
  <matplotlib.lines.Line2D at 0x7fca93ebc040>,
  <matplotlib.lines.Line2D at 0x7fca93ec7610>,
  <matplotlib.lines.Line2D at 0x7fca93ec79a0>,
  <matplotlib.lines.Line2D at 0x7fca93ed2f40>,
  <matplotlib.lines.Line2D at 0x7fca93edd310>,
  <matplotlib.lines.Line2D at 0x7fca93ee98b0>,
  <matplotlib.lines.Line2D at 0x7fca93ee9c40>,
  <matplotlib.lines.Line2D at 0x7fca94280220>,
  <matplotlib.lines.Line2D at 0x7fca942805b0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca93ebc400>,
  <matplotlib.lines.Line2D at 0x7fca93ebc790>,
  <matplotlib.lines.Line2D at 0x7fca93ec7d30>,
  <matplotlib.lines.Line2D at 0x7fca93ed2100>,
  <matplotlib.lines.Line2D at 0x7fca93edd6a0>,
  <matplotlib.lines.Line2D at 0x7fca93edda30>,
  <matplotlib.lines.Line2D at 0x7fca93ee9fd0>,
  <matplotlib.lines.Line2D at 0x7fca93ef23a0>,
  <matplotlib.lines.Line2D at 0x7fca94280940>,
  <matplotlib.lines.Line2D at 0x7fca94280cd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca942ec8e0>,
  <matplotlib.lines.Line2D at 0x7fca93ec7280>,
  <matplotlib.lines.Line2D at 0x7fca93ed2bb0>,
```

```
 <matplotlib.lines.Line2D at 0x7fca93ee9520>,
 <matplotlib.lines.Line2D at 0x7fca93ef2e50>],
'medians': [<matplotlib.lines.Line2D at 0x7fca93ebcb20>,
 <matplotlib.lines.Line2D at 0x7fca93ed2490>,
 <matplotlib.lines.Line2D at 0x7fca93edddc0>,
 <matplotlib.lines.Line2D at 0x7fca93ef2730>,
 <matplotlib.lines.Line2D at 0x7fca942890a0>],
'fliers': [<matplotlib.lines.Line2D at 0x7fca93ebceb0>,
 <matplotlib.lines.Line2D at 0x7fca93ed2820>,
 <matplotlib.lines.Line2D at 0x7fca93ee9190>,
 <matplotlib.lines.Line2D at 0x7fca93ef2ac0>,
 <matplotlib.lines.Line2D at 0x7fca94289430>],
'means': []}
```



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

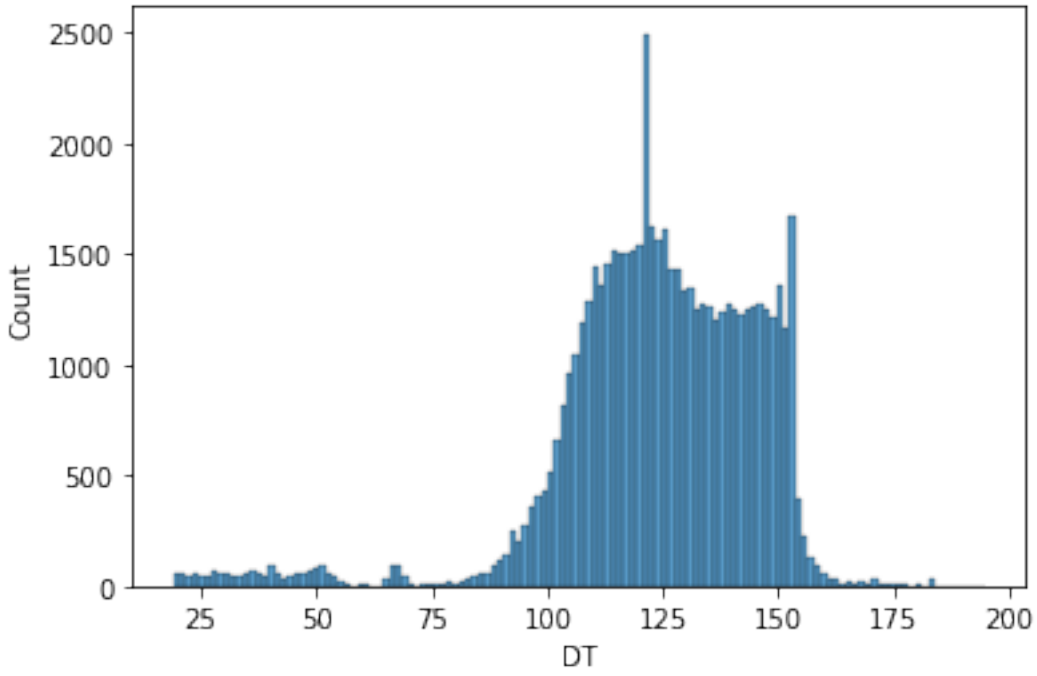
[62]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.2128	0.5340	2.1228	1
1	50.3881	49.7509	0.5316	2.1250	1
2	49.8852	48.2513	0.5126	2.1316	1
3	49.9032	46.8212	0.5137	2.1437	1
4	50.0157	45.3463	0.5472	2.1611	1

4.7 DT AFTER REMOVING OUTLIER

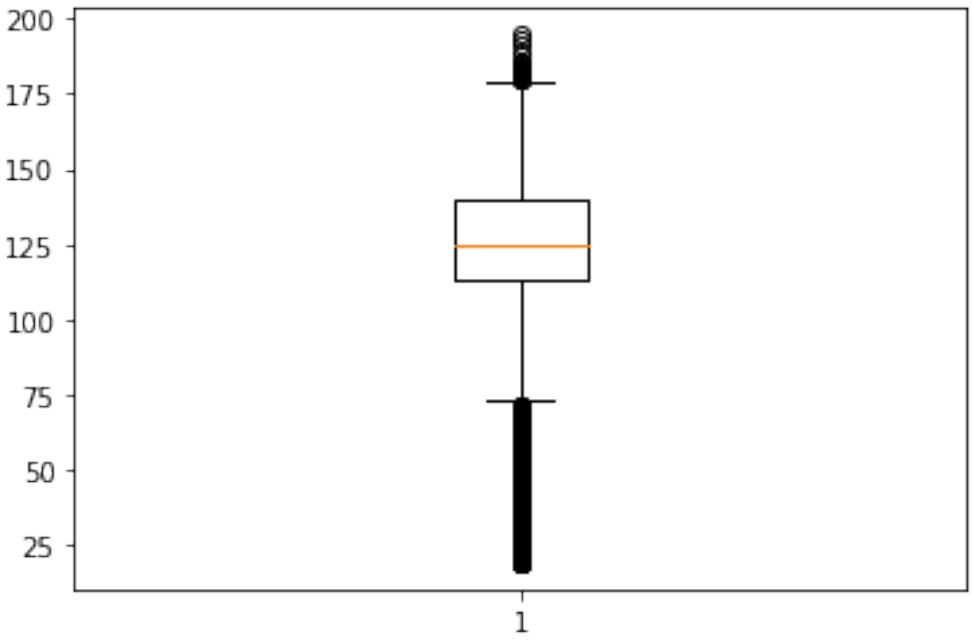
[63]:
```
sns.histplot(df.DT)
```

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



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

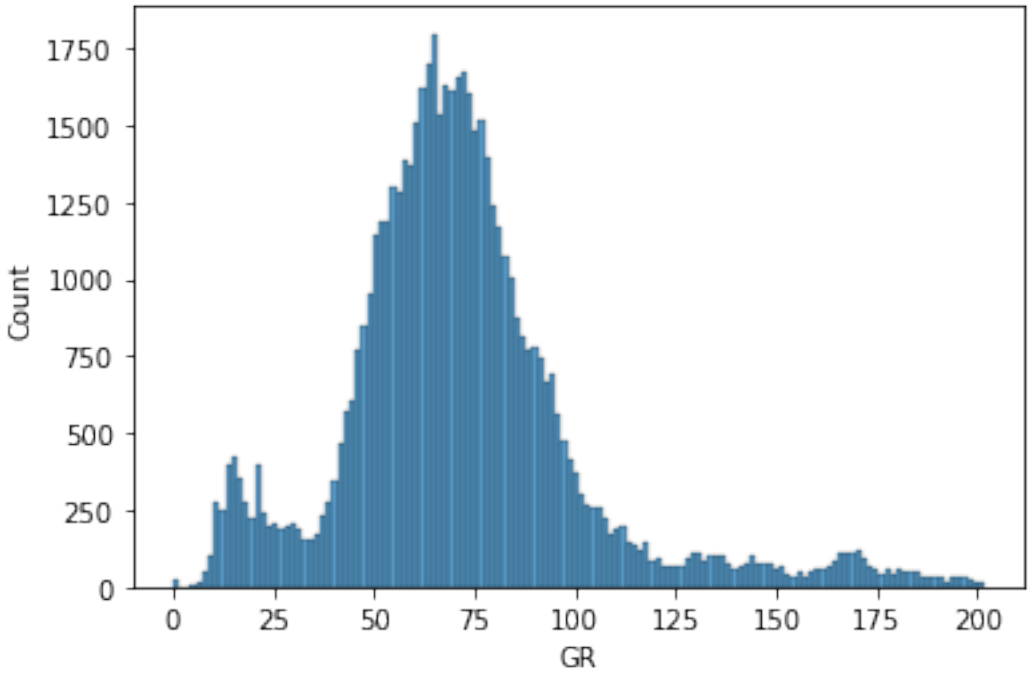
[64]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca941667c0>,
  <matplotlib.lines.Line2D at 0x7fca94166b50>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca94166ee0>,
  <matplotlib.lines.Line2D at 0x7fca941702b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca94166430>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca94170640>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca941709d0>],
 'means': []}
```



4.8 GR AFTER REMOVING OUTLIER

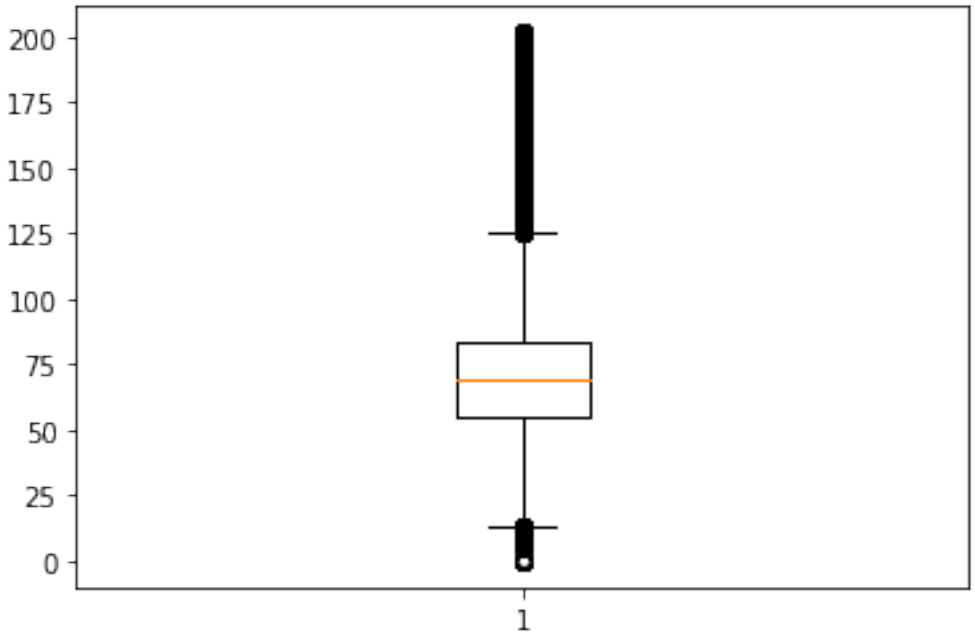
[65]:
```
sns.histplot(df.GR)
```

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



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

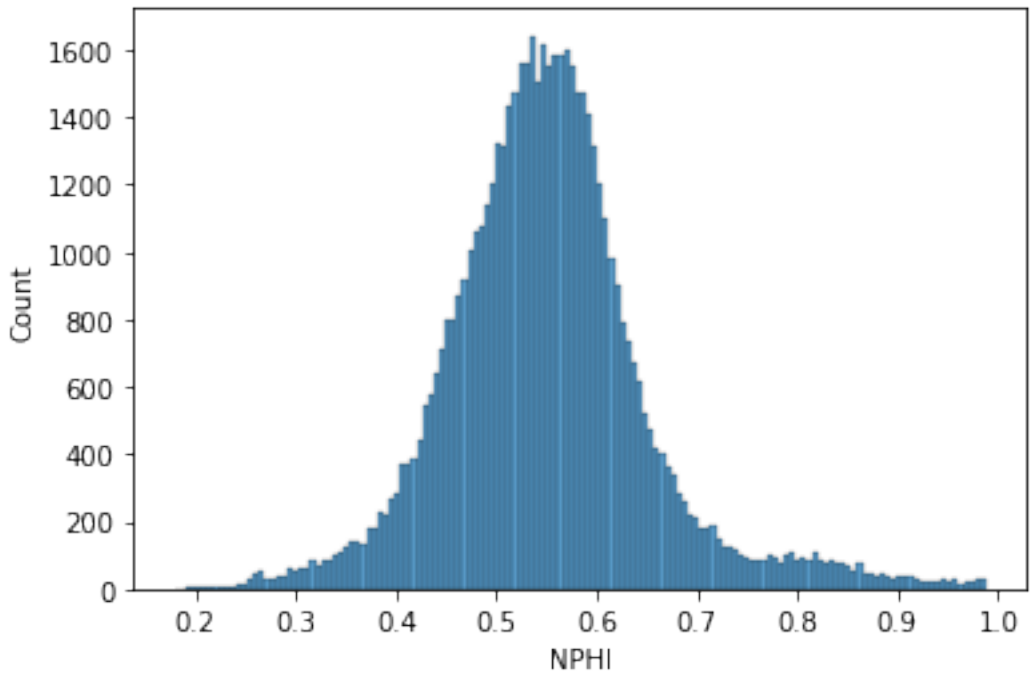
[66]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca93faa280>,
  <matplotlib.lines.Line2D at 0x7fca93faa610>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca93faa9a0>,
  <matplotlib.lines.Line2D at 0x7fca93faad30>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca93fa1eb0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca93fb1100>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca93fb1490>],
 'means': []}
```



4.9 NPHI AFTER REMOVING OUTLIER

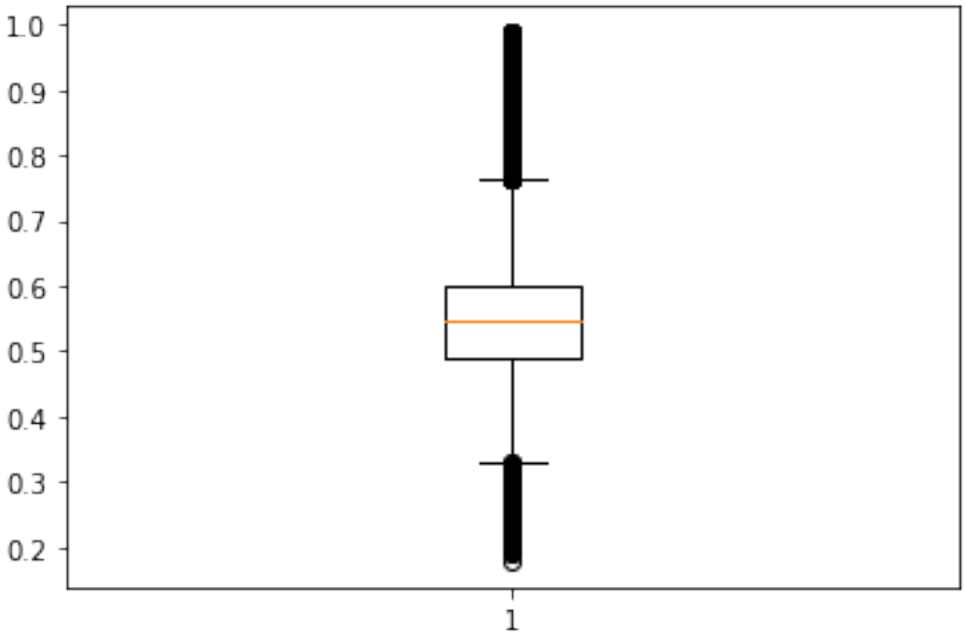
[67]:
```
sns.histplot(df.NPHI)
```

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



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

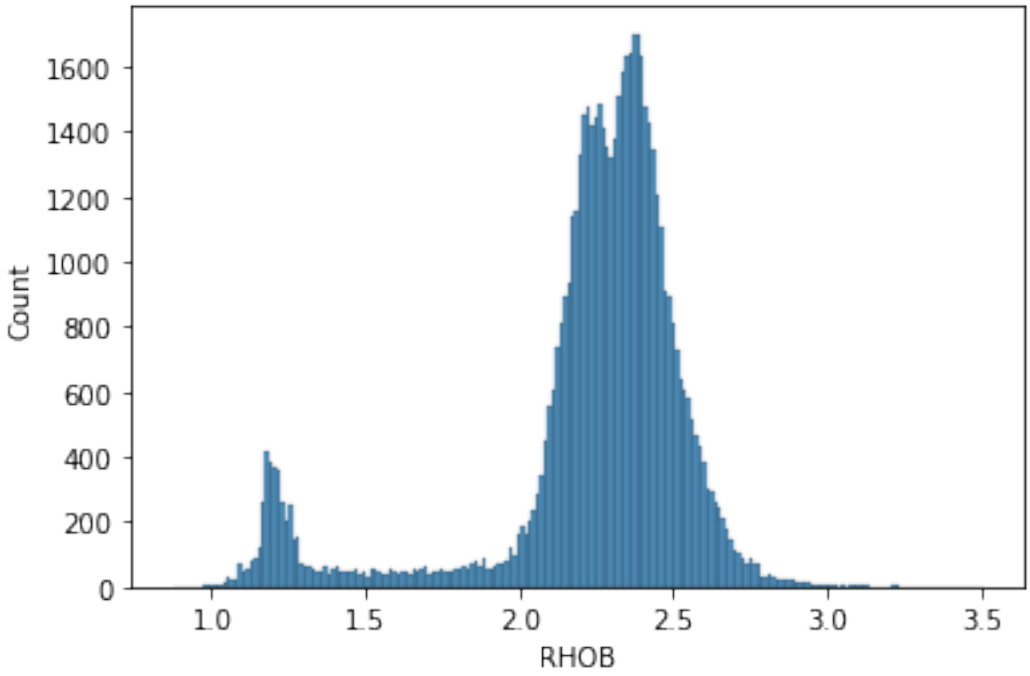
[68]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca93d5efd0>,
  <matplotlib.lines.Line2D at 0x7fca93d6b3a0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca93d6b730>,
  <matplotlib.lines.Line2D at 0x7fca93d6bac0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca93d5ec40>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca93d6be50>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca93cb5220>],
 'means': []}
```



4.10 RHOB AFTER REMOVING OUTLIER

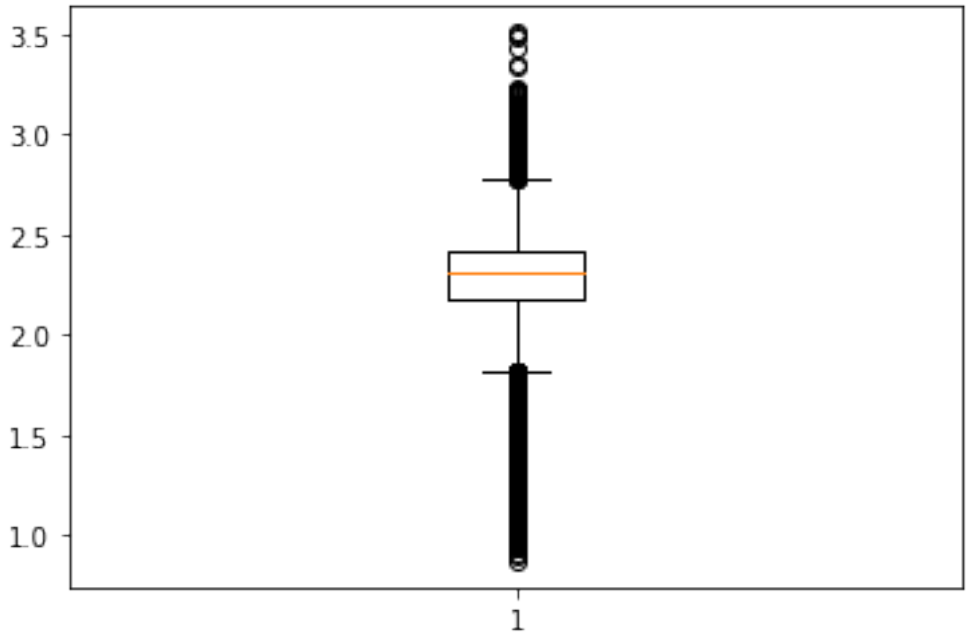
[69]:
```
sns.histplot(df.RHOB)
```

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



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

[70]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fca939f6820>,
  <matplotlib.lines.Line2D at 0x7fca939f6bb0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fca939f6f40>,
  <matplotlib.lines.Line2D at 0x7fca93a06310>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fca939f6490>],
 'medians': [<matplotlib.lines.Line2D at 0x7fca93a066a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fca93a06a30>],
 'means': []}
```



[71]:
```
df
```

[71]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	1
1	50.3881	49.750900	0.5316	2.1250	1
2	49.8852	48.251300	0.5126	2.1316	1
3	49.9032	46.821200	0.5137	2.1437	1
4	50.0157	45.346300	0.5472	2.1611	1
…	…	…	…	…	…
58494	123.7404	80.913653	0.4993	2.4639	0
58495	123.8728	82.952576	0.5313	2.4660	0
58496	123.3722	84.044079	0.5448	2.4714	0

```
58497  122.6038  83.725389  0.5364  2.4750       0
58498  122.3045  83.329152  0.5331  2.4709       0

[57139 rows x 5 columns]
```

5 FEATURE SELECTION

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

[72]:
```
        DT       GR    NPHI    RHOB  FACIES
0  50.2544  50.2128  0.5340  2.1228       1
1  50.3881  49.7509  0.5316  2.1250       1
2  49.8852  48.2513  0.5126  2.1316       1
3  49.9032  46.8212  0.5137  2.1437       1
4  50.0157  45.3463  0.5472  2.1611       1
5  50.6831  44.0819  0.5550  2.1740       1
6  51.4311  43.6654  0.5612  2.1707       1
7  52.1678  43.3915  0.5566  2.1595       1
8  52.2883  44.1249  0.5390  2.1534       1
9  51.5991  46.1805  0.5245  2.1551       1
```

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

[73]:
```
(57139, 5)
```

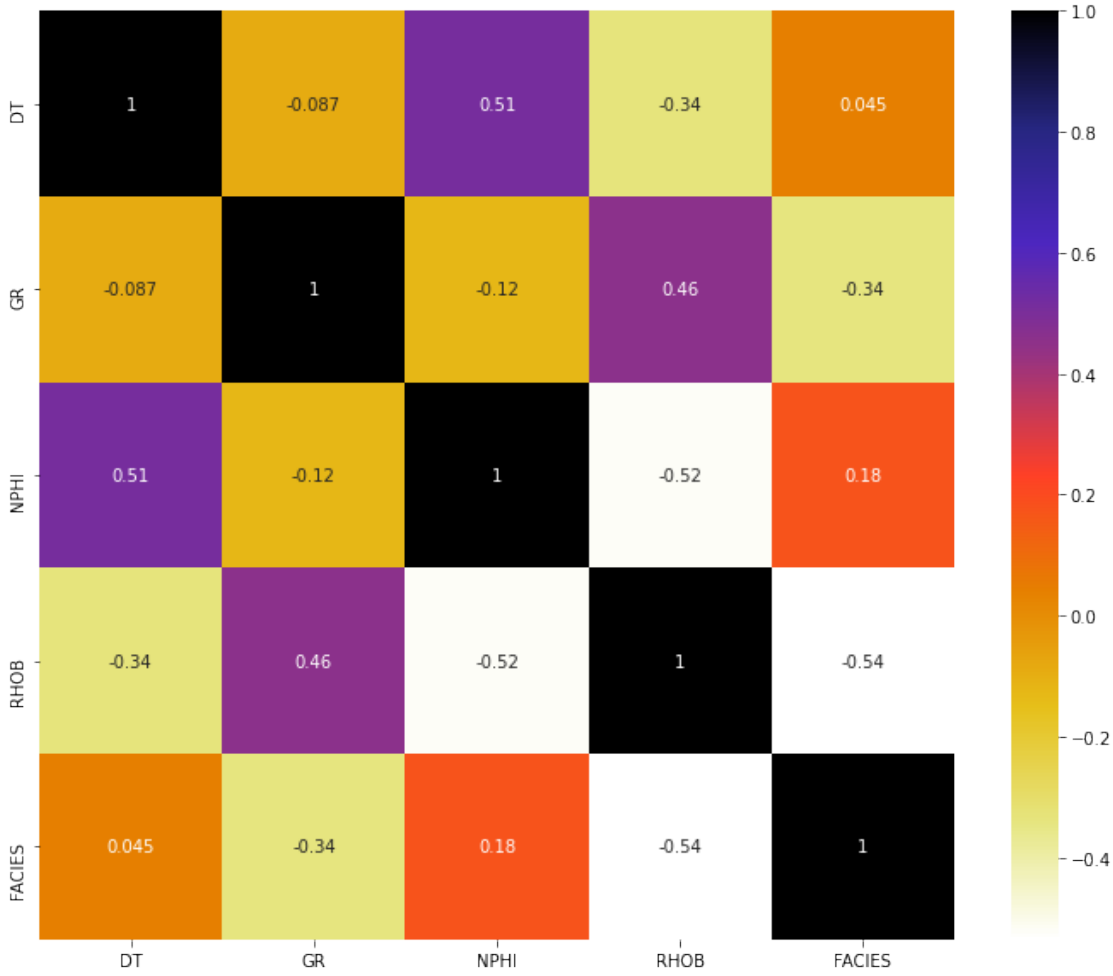
[74]:
```
features = df.shape[1]
features
```

[74]:
```
5
```

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

[75]:
```
DT        489.545094
GR        925.624347
NPHI        0.010355
RHOB        0.116722
FACIES      1.461299
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"):
        x = df.drop("FACIES",1)
        y = df["FACIES"]
        mutual_info = mutual_info_classif(x,y)
        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(x,y)
        column1 = df.columns[select_col.get_support()]
        df = df.drop(column1,axis=1)
        return df

    if(FeatureSelectionStrategy == "Mutual_Info_Class"):
        x = df.drop("FACIES",1)
        y = df["FACIES"]
        mutual_info = mutual_info_classif(x,y)
        print(mutual_info)
        mutual_info=pd.Series(mutual_info)
        mutual_info.sort_values(ascending=False)
        mutual_info.sort_values(ascending=False).plot.bar(figsize=(20,8))
        return df
```

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

```
[]
```

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

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

[80]:
```
df
```

[80]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	1
1	50.3881	49.750900	0.5316	2.1250	1
2	49.8852	48.251300	0.5126	2.1316	1
3	49.9032	46.821200	0.5137	2.1437	1
4	50.0157	45.346300	0.5472	2.1611	1
…	…	…	…	…	…
58494	123.7404	80.913653	0.4993	2.4639	0
58495	123.8728	82.952576	0.5313	2.4660	0
58496	123.3722	84.044079	0.5448	2.4714	0
58497	122.6038	83.725389	0.5364	2.4750	0
58498	122.3045	83.329152	0.5331	2.4709	0

[57139 rows x 5 columns]

[81]:
```
df
```

[81]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	1
1	50.3881	49.750900	0.5316	2.1250	1
2	49.8852	48.251300	0.5126	2.1316	1
3	49.9032	46.821200	0.5137	2.1437	1
4	50.0157	45.346300	0.5472	2.1611	1
…	…	…	…	…	…
58494	123.7404	80.913653	0.4993	2.4639	0
58495	123.8728	82.952576	0.5313	2.4660	0
58496	123.3722	84.044079	0.5448	2.4714	0
58497	122.6038	83.725389	0.5364	2.4750	0
58498	122.3045	83.329152	0.5331	2.4709	0

[57139 rows x 5 columns]

6 SCALING DATA

[82]:
```
df
```

[82]:
```
              DT         GR    NPHI    RHOB  FACIES
0        50.2544  50.212800  0.5340  2.1228       1
1        50.3881  49.750900  0.5316  2.1250       1
2        49.8852  48.251300  0.5126  2.1316       1
3        49.9032  46.821200  0.5137  2.1437       1
4        50.0157  45.346300  0.5472  2.1611       1
…            …        …      …       …      …
58494   123.7404  80.913653  0.4993  2.4639       0
58495   123.8728  82.952576  0.5313  2.4660       0
58496   123.3722  84.044079  0.5448  2.4714       0
58497   122.6038  83.725389  0.5364  2.4750       0
58498   122.3045  83.329152  0.5331  2.4709       0

[57139 rows x 5 columns]
```

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

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

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

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

    return scaling_data
```

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

[85]:
```
df
```

[85]:
```
              DT        GR      NPHI      RHOB  FACIES
0     -2.803542 -0.661900 -0.119485 -0.763591       1
1     -2.798523 -0.678299 -0.141544 -0.754530       1
2     -2.817399 -0.731541 -0.316176 -0.727348       1
3     -2.816723 -0.782315 -0.306066 -0.677512       1
```

```
4     -2.812501 -0.834680  0.001838 -0.605848       1
…          …       …        …        …      …
58494 -0.045374  0.428101 -0.438419  0.641269       0
58495 -0.040405  0.500491 -0.144301  0.649918       0
58496 -0.059194  0.539244 -0.020221  0.672158       0
58497 -0.088034  0.527929 -0.097426  0.686985       0
58498 -0.099268  0.513861 -0.127757  0.670099       0

[57139 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 -2.803542 -0.661900 -0.119485 -0.763591       1
1 -2.798523 -0.678299 -0.141544 -0.754530       1
2 -2.817399 -0.731541 -0.316176 -0.727348       1
3 -2.816723 -0.782315 -0.306066 -0.677512       1
4 -2.812501 -0.834680  0.001838 -0.605848       1
```

[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]:
```
(39997, 4)
```

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

[92]:
```
(17142, 4)
```

[93]:
```
X_test
```

[93]:

	DT	GR	NPHI	RHOB
42650	-0.575471	-0.004683	-1.307904	0.233526
54805	0.528453	-0.145445	-0.431985	-0.069193
54922	0.665783	-0.368779	-0.153493	-0.519357
34974	-0.089806	1.200366	-0.352941	0.518534
26096	-0.840182	0.323512	-0.114890	1.163509
…	…	…	…	…
49847	0.124456	0.538616	-0.288603	-0.135091
13010	0.716322	-0.412335	0.527574	-0.888797
44789	0.692875	0.688716	0.249081	-0.347611
46674	-0.861283	1.021238	-1.046875	1.209638
18267	-0.214364	-0.162345	-0.022059	0.247117

[17142 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: 0.891275 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.891275 (0.002559) with: {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.891275 (0.002559) with: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
0.888975 (0.002352) with: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
0.891275 (0.002559) with: {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.891275 (0.002559) with: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
```

```
0.888975 (0.002352) with: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
0.891234 (0.002557) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'newton-cg'}
0.891234 (0.002557) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
0.888983 (0.002358) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}
0.891100 (0.002462) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
0.891100 (0.002462) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
0.888758 (0.002282) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
0.889825 (0.002423) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
0.889825 (0.002423) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
0.887492 (0.002348) 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 0.095505
0:      learn: 1.3276154        total: 55.3ms   remaining: 55.3s
1:      learn: 1.1457638        total: 62.8ms   remaining: 31.3s
2:      learn: 1.0110403        total: 69.9ms   remaining: 23.2s
3:      learn: 0.9077536        total: 76.9ms   remaining: 19.2s
```

```
4:      learn: 0.8238180        total: 83.8ms   remaining: 16.7s
5:      learn: 0.7553297        total: 91.3ms   remaining: 15.1s
6:      learn: 0.6987072        total: 98.9ms   remaining: 14s
7:      learn: 0.6509033        total: 106ms    remaining: 13.2s
8:      learn: 0.6097222        total: 113ms    remaining: 12.5s
9:      learn: 0.5749712        total: 121ms    remaining: 11.9s
10:     learn: 0.5450739        total: 128ms    remaining: 11.5s
11:     learn: 0.5178381        total: 135ms    remaining: 11.1s
12:     learn: 0.4950173        total: 142ms    remaining: 10.7s
13:     learn: 0.4748612        total: 148ms    remaining: 10.4s
14:     learn: 0.4569448        total: 155ms    remaining: 10.2s
15:     learn: 0.4403616        total: 163ms    remaining: 10s
16:     learn: 0.4262248        total: 170ms    remaining: 9.8s
17:     learn: 0.4135618        total: 177ms    remaining: 9.66s
18:     learn: 0.4027381        total: 185ms    remaining: 9.54s
19:     learn: 0.3923539        total: 193ms    remaining: 9.44s
20:     learn: 0.3822236        total: 201ms    remaining: 9.37s
21:     learn: 0.3746566        total: 208ms    remaining: 9.25s
22:     learn: 0.3669686        total: 217ms    remaining: 9.21s
23:     learn: 0.3600197        total: 225ms    remaining: 9.15s
24:     learn: 0.3532062        total: 233ms    remaining: 9.09s
25:     learn: 0.3486258        total: 241ms    remaining: 9.02s
26:     learn: 0.3432447        total: 249ms    remaining: 8.96s
27:     learn: 0.3387909        total: 256ms    remaining: 8.89s
28:     learn: 0.3343990        total: 263ms    remaining: 8.82s
29:     learn: 0.3303402        total: 271ms    remaining: 8.77s
30:     learn: 0.3273032        total: 279ms    remaining: 8.71s
31:     learn: 0.3243001        total: 287ms    remaining: 8.67s
32:     learn: 0.3213942        total: 294ms    remaining: 8.61s
33:     learn: 0.3183091        total: 302ms    remaining: 8.57s
34:     learn: 0.3161602        total: 309ms    remaining: 8.53s
35:     learn: 0.3133099        total: 318ms    remaining: 8.5s
36:     learn: 0.3111500        total: 325ms    remaining: 8.46s
37:     learn: 0.3090502        total: 333ms    remaining: 8.43s
38:     learn: 0.3073102        total: 341ms    remaining: 8.39s
39:     learn: 0.3059064        total: 349ms    remaining: 8.37s
40:     learn: 0.3039099        total: 356ms    remaining: 8.33s
41:     learn: 0.3025671        total: 364ms    remaining: 8.3s
42:     learn: 0.3013146        total: 371ms    remaining: 8.26s
43:     learn: 0.2999086        total: 379ms    remaining: 8.24s
44:     learn: 0.2985966        total: 387ms    remaining: 8.22s
45:     learn: 0.2974635        total: 395ms    remaining: 8.19s
46:     learn: 0.2965798        total: 402ms    remaining: 8.14s
47:     learn: 0.2955385        total: 409ms    remaining: 8.11s
48:     learn: 0.2947338        total: 416ms    remaining: 8.07s
49:     learn: 0.2938785        total: 423ms    remaining: 8.04s
50:     learn: 0.2930301        total: 431ms    remaining: 8.02s
51:     learn: 0.2920700        total: 438ms    remaining: 7.99s
```

```
52:     learn: 0.2914279        total: 445ms    remaining: 7.95s
53:     learn: 0.2905416        total: 452ms    remaining: 7.92s
54:     learn: 0.2898759        total: 459ms    remaining: 7.89s
55:     learn: 0.2886715        total: 467ms    remaining: 7.88s
56:     learn: 0.2874995        total: 475ms    remaining: 7.86s
57:     learn: 0.2868766        total: 483ms    remaining: 7.84s
58:     learn: 0.2860048        total: 491ms    remaining: 7.83s
59:     learn: 0.2853461        total: 498ms    remaining: 7.8s
60:     learn: 0.2848691        total: 505ms    remaining: 7.78s
61:     learn: 0.2838477        total: 513ms    remaining: 7.76s
62:     learn: 0.2830013        total: 520ms    remaining: 7.73s
63:     learn: 0.2825095        total: 527ms    remaining: 7.71s
64:     learn: 0.2818150        total: 535ms    remaining: 7.69s
65:     learn: 0.2812549        total: 542ms    remaining: 7.67s
66:     learn: 0.2807495        total: 549ms    remaining: 7.64s
67:     learn: 0.2801847        total: 556ms    remaining: 7.62s
68:     learn: 0.2797215        total: 563ms    remaining: 7.6s
69:     learn: 0.2793590        total: 571ms    remaining: 7.58s
70:     learn: 0.2790460        total: 578ms    remaining: 7.56s
71:     learn: 0.2787033        total: 586ms    remaining: 7.55s
72:     learn: 0.2782438        total: 593ms    remaining: 7.53s
73:     learn: 0.2779217        total: 600ms    remaining: 7.51s
74:     learn: 0.2775558        total: 607ms    remaining: 7.49s
75:     learn: 0.2771658        total: 614ms    remaining: 7.47s
76:     learn: 0.2767214        total: 622ms    remaining: 7.45s
77:     learn: 0.2762432        total: 630ms    remaining: 7.45s
78:     learn: 0.2757996        total: 637ms    remaining: 7.42s
79:     learn: 0.2753853        total: 644ms    remaining: 7.41s
80:     learn: 0.2750767        total: 652ms    remaining: 7.4s
81:     learn: 0.2748140        total: 660ms    remaining: 7.38s
82:     learn: 0.2743457        total: 667ms    remaining: 7.37s
83:     learn: 0.2739430        total: 675ms    remaining: 7.36s
84:     learn: 0.2736141        total: 682ms    remaining: 7.34s
85:     learn: 0.2732808        total: 690ms    remaining: 7.33s
86:     learn: 0.2728967        total: 697ms    remaining: 7.32s
87:     learn: 0.2726153        total: 704ms    remaining: 7.3s
88:     learn: 0.2721895        total: 712ms    remaining: 7.29s
89:     learn: 0.2716820        total: 719ms    remaining: 7.27s
90:     learn: 0.2714916        total: 726ms    remaining: 7.25s
91:     learn: 0.2711811        total: 734ms    remaining: 7.24s
92:     learn: 0.2708701        total: 741ms    remaining: 7.23s
93:     learn: 0.2706099        total: 748ms    remaining: 7.21s
94:     learn: 0.2703891        total: 755ms    remaining: 7.2s
95:     learn: 0.2699719        total: 763ms    remaining: 7.19s
96:     learn: 0.2697404        total: 771ms    remaining: 7.18s
97:     learn: 0.2695521        total: 778ms    remaining: 7.16s
98:     learn: 0.2691656        total: 787ms    remaining: 7.16s
99:     learn: 0.2689878        total: 794ms    remaining: 7.14s
```

```
100:    learn: 0.2688336        total: 801ms    remaining: 7.13s
101:    learn: 0.2685892        total: 807ms    remaining: 7.11s
102:    learn: 0.2684132        total: 814ms    remaining: 7.09s
103:    learn: 0.2681264        total: 821ms    remaining: 7.08s
104:    learn: 0.2678581        total: 829ms    remaining: 7.06s
105:    learn: 0.2675943        total: 836ms    remaining: 7.05s
106:    learn: 0.2671050        total: 844ms    remaining: 7.04s
107:    learn: 0.2668319        total: 851ms    remaining: 7.03s
108:    learn: 0.2665926        total: 858ms    remaining: 7.02s
109:    learn: 0.2664066        total: 865ms    remaining: 7s
110:    learn: 0.2661561        total: 873ms    remaining: 6.99s
111:    learn: 0.2655979        total: 880ms    remaining: 6.98s
112:    learn: 0.2651464        total: 887ms    remaining: 6.96s
113:    learn: 0.2649383        total: 894ms    remaining: 6.95s
114:    learn: 0.2646639        total: 901ms    remaining: 6.93s
115:    learn: 0.2643307        total: 909ms    remaining: 6.92s
116:    learn: 0.2640763        total: 916ms    remaining: 6.91s
117:    learn: 0.2637041        total: 923ms    remaining: 6.9s
118:    learn: 0.2634001        total: 931ms    remaining: 6.89s
119:    learn: 0.2630758        total: 938ms    remaining: 6.88s
120:    learn: 0.2628740        total: 946ms    remaining: 6.87s
121:    learn: 0.2626441        total: 952ms    remaining: 6.85s
122:    learn: 0.2623667        total: 961ms    remaining: 6.85s
123:    learn: 0.2621217        total: 970ms    remaining: 6.85s
124:    learn: 0.2619156        total: 977ms    remaining: 6.84s
125:    learn: 0.2617724        total: 984ms    remaining: 6.83s
126:    learn: 0.2616200        total: 991ms    remaining: 6.81s
127:    learn: 0.2613448        total: 999ms    remaining: 6.8s
128:    learn: 0.2610293        total: 1.01s    remaining: 6.79s
129:    learn: 0.2609144        total: 1.01s    remaining: 6.78s
130:    learn: 0.2607880        total: 1.02s    remaining: 6.77s
131:    learn: 0.2605698        total: 1.03s    remaining: 6.75s
132:    learn: 0.2603043        total: 1.03s    remaining: 6.74s
133:    learn: 0.2599586        total: 1.04s    remaining: 6.73s
134:    learn: 0.2598233        total: 1.05s    remaining: 6.72s
135:    learn: 0.2595862        total: 1.06s    remaining: 6.71s
136:    learn: 0.2593513        total: 1.06s    remaining: 6.7s
137:    learn: 0.2590927        total: 1.07s    remaining: 6.69s
138:    learn: 0.2590009        total: 1.08s    remaining: 6.68s
139:    learn: 0.2589021        total: 1.08s    remaining: 6.67s
140:    learn: 0.2588204        total: 1.09s    remaining: 6.66s
141:    learn: 0.2586302        total: 1.1s     remaining: 6.64s
142:    learn: 0.2584362        total: 1.11s    remaining: 6.63s
143:    learn: 0.2582853        total: 1.11s    remaining: 6.62s
144:    learn: 0.2581116        total: 1.12s    remaining: 6.61s
145:    learn: 0.2578721        total: 1.13s    remaining: 6.6s
146:    learn: 0.2576681        total: 1.14s    remaining: 6.59s
147:    learn: 0.2575430        total: 1.14s    remaining: 6.58s
```

```
148:    learn: 0.2574022        total: 1.15s    remaining: 6.57s
149:    learn: 0.2568709        total: 1.16s    remaining: 6.56s
150:    learn: 0.2566938        total: 1.17s    remaining: 6.56s
151:    learn: 0.2563110        total: 1.17s    remaining: 6.55s
152:    learn: 0.2561192        total: 1.18s    remaining: 6.54s
153:    learn: 0.2560112        total: 1.19s    remaining: 6.54s
154:    learn: 0.2558910        total: 1.2s     remaining: 6.53s
155:    learn: 0.2556751        total: 1.21s    remaining: 6.52s
156:    learn: 0.2554589        total: 1.21s    remaining: 6.51s
157:    learn: 0.2552240        total: 1.22s    remaining: 6.5s
158:    learn: 0.2550307        total: 1.23s    remaining: 6.5s
159:    learn: 0.2548409        total: 1.24s    remaining: 6.49s
160:    learn: 0.2545776        total: 1.24s    remaining: 6.48s
161:    learn: 0.2544270        total: 1.25s    remaining: 6.47s
162:    learn: 0.2542530        total: 1.26s    remaining: 6.46s
163:    learn: 0.2539503        total: 1.27s    remaining: 6.46s
164:    learn: 0.2537300        total: 1.27s    remaining: 6.45s
165:    learn: 0.2536511        total: 1.28s    remaining: 6.44s
166:    learn: 0.2534897        total: 1.29s    remaining: 6.43s
167:    learn: 0.2533036        total: 1.3s     remaining: 6.42s
168:    learn: 0.2531296        total: 1.3s     remaining: 6.42s
169:    learn: 0.2529004        total: 1.31s    remaining: 6.41s
170:    learn: 0.2525462        total: 1.32s    remaining: 6.4s
171:    learn: 0.2523578        total: 1.33s    remaining: 6.39s
172:    learn: 0.2520687        total: 1.33s    remaining: 6.38s
173:    learn: 0.2518048        total: 1.34s    remaining: 6.37s
174:    learn: 0.2515422        total: 1.35s    remaining: 6.36s
175:    learn: 0.2514109        total: 1.36s    remaining: 6.35s
176:    learn: 0.2512306        total: 1.36s    remaining: 6.35s
177:    learn: 0.2511099        total: 1.37s    remaining: 6.34s
178:    learn: 0.2509568        total: 1.38s    remaining: 6.33s
179:    learn: 0.2508628        total: 1.39s    remaining: 6.31s
180:    learn: 0.2507455        total: 1.39s    remaining: 6.31s
181:    learn: 0.2505700        total: 1.4s     remaining: 6.3s
182:    learn: 0.2503963        total: 1.41s    remaining: 6.29s
183:    learn: 0.2503195        total: 1.41s    remaining: 6.27s
184:    learn: 0.2501071        total: 1.42s    remaining: 6.27s
185:    learn: 0.2499116        total: 1.43s    remaining: 6.26s
186:    learn: 0.2498489        total: 1.44s    remaining: 6.24s
187:    learn: 0.2497756        total: 1.44s    remaining: 6.24s
188:    learn: 0.2495971        total: 1.45s    remaining: 6.23s
189:    learn: 0.2494303        total: 1.46s    remaining: 6.22s
190:    learn: 0.2492188        total: 1.47s    remaining: 6.21s
191:    learn: 0.2490639        total: 1.47s    remaining: 6.2s
192:    learn: 0.2488807        total: 1.48s    remaining: 6.19s
193:    learn: 0.2487479        total: 1.49s    remaining: 6.18s
194:    learn: 0.2485529        total: 1.49s    remaining: 6.17s
195:    learn: 0.2484392        total: 1.5s     remaining: 6.16s
```

```
196:    learn: 0.2482936        total: 1.51s    remaining: 6.15s
197:    learn: 0.2482147        total: 1.52s    remaining: 6.14s
198:    learn: 0.2481404        total: 1.52s    remaining: 6.13s
199:    learn: 0.2477609        total: 1.53s    remaining: 6.13s
200:    learn: 0.2475457        total: 1.54s    remaining: 6.12s
201:    learn: 0.2474909        total: 1.55s    remaining: 6.11s
202:    learn: 0.2473681        total: 1.55s    remaining: 6.1s
203:    learn: 0.2472798        total: 1.56s    remaining: 6.1s
204:    learn: 0.2470633        total: 1.57s    remaining: 6.09s
205:    learn: 0.2469369        total: 1.58s    remaining: 6.08s
206:    learn: 0.2468060        total: 1.58s    remaining: 6.07s
207:    learn: 0.2467327        total: 1.59s    remaining: 6.06s
208:    learn: 0.2463749        total: 1.6s     remaining: 6.05s
209:    learn: 0.2462022        total: 1.61s    remaining: 6.05s
210:    learn: 0.2460486        total: 1.61s    remaining: 6.04s
211:    learn: 0.2458837        total: 1.62s    remaining: 6.03s
212:    learn: 0.2456564        total: 1.63s    remaining: 6.02s
213:    learn: 0.2454627        total: 1.64s    remaining: 6.02s
214:    learn: 0.2452511        total: 1.65s    remaining: 6.01s
215:    learn: 0.2450926        total: 1.65s    remaining: 6s
216:    learn: 0.2449357        total: 1.66s    remaining: 6s
217:    learn: 0.2448126        total: 1.67s    remaining: 5.99s
218:    learn: 0.2447616        total: 1.68s    remaining: 5.97s
219:    learn: 0.2446769        total: 1.68s    remaining: 5.97s
220:    learn: 0.2445485        total: 1.69s    remaining: 5.96s
221:    learn: 0.2444246        total: 1.7s     remaining: 5.95s
222:    learn: 0.2443437        total: 1.71s    remaining: 5.94s
223:    learn: 0.2442433        total: 1.71s    remaining: 5.93s
224:    learn: 0.2440256        total: 1.72s    remaining: 5.92s
225:    learn: 0.2438349        total: 1.73s    remaining: 5.91s
226:    learn: 0.2436935        total: 1.73s    remaining: 5.9s
227:    learn: 0.2435896        total: 1.74s    remaining: 5.9s
228:    learn: 0.2434702        total: 1.75s    remaining: 5.89s
229:    learn: 0.2432927        total: 1.76s    remaining: 5.88s
230:    learn: 0.2432226        total: 1.76s    remaining: 5.87s
231:    learn: 0.2431590        total: 1.77s    remaining: 5.86s
232:    learn: 0.2429567        total: 1.78s    remaining: 5.86s
233:    learn: 0.2428265        total: 1.79s    remaining: 5.85s
234:    learn: 0.2427151        total: 1.79s    remaining: 5.84s
235:    learn: 0.2426057        total: 1.8s     remaining: 5.83s
236:    learn: 0.2423756        total: 1.81s    remaining: 5.82s
237:    learn: 0.2421899        total: 1.81s    remaining: 5.81s
238:    learn: 0.2420344        total: 1.82s    remaining: 5.8s
239:    learn: 0.2419247        total: 1.83s    remaining: 5.8s
240:    learn: 0.2416771        total: 1.84s    remaining: 5.79s
241:    learn: 0.2415613        total: 1.84s    remaining: 5.78s
242:    learn: 0.2413250        total: 1.85s    remaining: 5.77s
243:    learn: 0.2411449        total: 1.86s    remaining: 5.76s
```

```
244:    learn: 0.2410919        total: 1.87s    remaining: 5.75s
245:    learn: 0.2409225        total: 1.87s    remaining: 5.75s
246:    learn: 0.2407865        total: 1.88s    remaining: 5.74s
247:    learn: 0.2406869        total: 1.89s    remaining: 5.73s
248:    learn: 0.2406110        total: 1.9s     remaining: 5.72s
249:    learn: 0.2403800        total: 1.9s     remaining: 5.71s
250:    learn: 0.2403064        total: 1.91s    remaining: 5.7s
251:    learn: 0.2401923        total: 1.92s    remaining: 5.69s
252:    learn: 0.2401191        total: 1.93s    remaining: 5.69s
253:    learn: 0.2399966        total: 1.93s    remaining: 5.68s
254:    learn: 0.2399196        total: 1.94s    remaining: 5.67s
255:    learn: 0.2397425        total: 1.95s    remaining: 5.67s
256:    learn: 0.2396325        total: 1.96s    remaining: 5.66s
257:    learn: 0.2395763        total: 1.96s    remaining: 5.65s
258:    learn: 0.2394735        total: 1.97s    remaining: 5.64s
259:    learn: 0.2393206        total: 1.98s    remaining: 5.63s
260:    learn: 0.2392814        total: 1.99s    remaining: 5.62s
261:    learn: 0.2391557        total: 1.99s    remaining: 5.61s
262:    learn: 0.2389695        total: 2s       remaining: 5.61s
263:    learn: 0.2387294        total: 2.01s    remaining: 5.6s
264:    learn: 0.2386396        total: 2.02s    remaining: 5.59s
265:    learn: 0.2384775        total: 2.02s    remaining: 5.58s
266:    learn: 0.2383245        total: 2.03s    remaining: 5.57s
267:    learn: 0.2381626        total: 2.04s    remaining: 5.57s
268:    learn: 0.2380120        total: 2.04s    remaining: 5.56s
269:    learn: 0.2379148        total: 2.05s    remaining: 5.55s
270:    learn: 0.2376015        total: 2.06s    remaining: 5.54s
271:    learn: 0.2374820        total: 2.07s    remaining: 5.54s
272:    learn: 0.2373164        total: 2.08s    remaining: 5.53s
273:    learn: 0.2372364        total: 2.08s    remaining: 5.52s
274:    learn: 0.2369775        total: 2.09s    remaining: 5.51s
275:    learn: 0.2368355        total: 2.1s     remaining: 5.5s
276:    learn: 0.2367114        total: 2.11s    remaining: 5.5s
277:    learn: 0.2365761        total: 2.12s    remaining: 5.49s
278:    learn: 0.2365321        total: 2.12s    remaining: 5.48s
279:    learn: 0.2363326        total: 2.13s    remaining: 5.48s
280:    learn: 0.2362459        total: 2.14s    remaining: 5.47s
281:    learn: 0.2360973        total: 2.15s    remaining: 5.47s
282:    learn: 0.2359020        total: 2.15s    remaining: 5.46s
283:    learn: 0.2358196        total: 2.16s    remaining: 5.45s
284:    learn: 0.2357326        total: 2.17s    remaining: 5.44s
285:    learn: 0.2356752        total: 2.18s    remaining: 5.44s
286:    learn: 0.2355591        total: 2.19s    remaining: 5.43s
287:    learn: 0.2354112        total: 2.19s    remaining: 5.42s
288:    learn: 0.2352415        total: 2.2s     remaining: 5.41s
289:    learn: 0.2351593        total: 2.21s    remaining: 5.41s
290:    learn: 0.2350416        total: 2.22s    remaining: 5.4s
291:    learn: 0.2349146        total: 2.22s    remaining: 5.39s
```

```
292:    learn: 0.2348338        total: 2.23s    remaining: 5.38s
293:    learn: 0.2348048        total: 2.24s    remaining: 5.38s
294:    learn: 0.2346400        total: 2.25s    remaining: 5.37s
295:    learn: 0.2345485        total: 2.25s    remaining: 5.36s
296:    learn: 0.2344266        total: 2.26s    remaining: 5.35s
297:    learn: 0.2342469        total: 2.27s    remaining: 5.35s
298:    learn: 0.2342040        total: 2.28s    remaining: 5.34s
299:    learn: 0.2341012        total: 2.28s    remaining: 5.33s
300:    learn: 0.2340191        total: 2.29s    remaining: 5.32s
301:    learn: 0.2339774        total: 2.3s     remaining: 5.31s
302:    learn: 0.2338827        total: 2.31s    remaining: 5.3s
303:    learn: 0.2338192        total: 2.31s    remaining: 5.3s
304:    learn: 0.2337448        total: 2.32s    remaining: 5.29s
305:    learn: 0.2336089        total: 2.33s    remaining: 5.28s
306:    learn: 0.2335116        total: 2.34s    remaining: 5.27s
307:    learn: 0.2334436        total: 2.34s    remaining: 5.27s
308:    learn: 0.2333042        total: 2.35s    remaining: 5.26s
309:    learn: 0.2332225        total: 2.36s    remaining: 5.25s
310:    learn: 0.2331160        total: 2.37s    remaining: 5.24s
311:    learn: 0.2330736        total: 2.37s    remaining: 5.23s
312:    learn: 0.2330304        total: 2.38s    remaining: 5.22s
313:    learn: 0.2329620        total: 2.39s    remaining: 5.21s
314:    learn: 0.2328958        total: 2.39s    remaining: 5.21s
315:    learn: 0.2328195        total: 2.4s     remaining: 5.2s
316:    learn: 0.2327427        total: 2.41s    remaining: 5.19s
317:    learn: 0.2326397        total: 2.42s    remaining: 5.18s
318:    learn: 0.2324198        total: 2.42s    remaining: 5.17s
319:    learn: 0.2323518        total: 2.43s    remaining: 5.17s
320:    learn: 0.2322999        total: 2.44s    remaining: 5.16s
321:    learn: 0.2321954        total: 2.44s    remaining: 5.15s
322:    learn: 0.2321032        total: 2.45s    remaining: 5.14s
323:    learn: 0.2319366        total: 2.46s    remaining: 5.13s
324:    learn: 0.2317726        total: 2.47s    remaining: 5.12s
325:    learn: 0.2316409        total: 2.47s    remaining: 5.12s
326:    learn: 0.2316005        total: 2.48s    remaining: 5.11s
327:    learn: 0.2315380        total: 2.49s    remaining: 5.1s
328:    learn: 0.2314516        total: 2.5s     remaining: 5.09s
329:    learn: 0.2313703        total: 2.5s     remaining: 5.08s
330:    learn: 0.2311573        total: 2.51s    remaining: 5.08s
331:    learn: 0.2309581        total: 2.52s    remaining: 5.07s
332:    learn: 0.2308490        total: 2.53s    remaining: 5.06s
333:    learn: 0.2307857        total: 2.53s    remaining: 5.05s
334:    learn: 0.2307403        total: 2.54s    remaining: 5.04s
335:    learn: 0.2305437        total: 2.55s    remaining: 5.04s
336:    learn: 0.2305183        total: 2.56s    remaining: 5.03s
337:    learn: 0.2304748        total: 2.56s    remaining: 5.02s
338:    learn: 0.2303887        total: 2.57s    remaining: 5.01s
339:    learn: 0.2302773        total: 2.58s    remaining: 5s
```

```
340:    learn: 0.2302005        total: 2.58s    remaining: 4.99s
341:    learn: 0.2300929        total: 2.59s    remaining: 4.99s
342:    learn: 0.2299361        total: 2.6s     remaining: 4.98s
343:    learn: 0.2298116        total: 2.61s    remaining: 4.97s
344:    learn: 0.2297172        total: 2.61s    remaining: 4.96s
345:    learn: 0.2295931        total: 2.62s    remaining: 4.96s
346:    learn: 0.2295538        total: 2.63s    remaining: 4.95s
347:    learn: 0.2293618        total: 2.64s    remaining: 4.94s
348:    learn: 0.2292877        total: 2.64s    remaining: 4.93s
349:    learn: 0.2290974        total: 2.65s    remaining: 4.92s
350:    learn: 0.2290170        total: 2.66s    remaining: 4.92s
351:    learn: 0.2289719        total: 2.67s    remaining: 4.91s
352:    learn: 0.2288601        total: 2.67s    remaining: 4.9s
353:    learn: 0.2287043        total: 2.68s    remaining: 4.89s
354:    learn: 0.2286398        total: 2.69s    remaining: 4.88s
355:    learn: 0.2285758        total: 2.7s     remaining: 4.88s
356:    learn: 0.2284350        total: 2.7s     remaining: 4.87s
357:    learn: 0.2283766        total: 2.71s    remaining: 4.86s
358:    learn: 0.2282720        total: 2.72s    remaining: 4.86s
359:    learn: 0.2282232        total: 2.73s    remaining: 4.85s
360:    learn: 0.2281169        total: 2.73s    remaining: 4.84s
361:    learn: 0.2279885        total: 2.74s    remaining: 4.83s
362:    learn: 0.2279104        total: 2.75s    remaining: 4.83s
363:    learn: 0.2277190        total: 2.76s    remaining: 4.82s
364:    learn: 0.2276282        total: 2.77s    remaining: 4.81s
365:    learn: 0.2275950        total: 2.77s    remaining: 4.8s
366:    learn: 0.2275428        total: 2.78s    remaining: 4.79s
367:    learn: 0.2274124        total: 2.79s    remaining: 4.79s
368:    learn: 0.2273591        total: 2.79s    remaining: 4.78s
369:    learn: 0.2272263        total: 2.8s     remaining: 4.77s
370:    learn: 0.2270708        total: 2.81s    remaining: 4.76s
371:    learn: 0.2269721        total: 2.82s    remaining: 4.75s
372:    learn: 0.2268998        total: 2.82s    remaining: 4.75s
373:    learn: 0.2267782        total: 2.83s    remaining: 4.74s
374:    learn: 0.2266599        total: 2.84s    remaining: 4.73s
375:    learn: 0.2265663        total: 2.85s    remaining: 4.72s
376:    learn: 0.2264524        total: 2.85s    remaining: 4.71s
377:    learn: 0.2263512        total: 2.86s    remaining: 4.71s
378:    learn: 0.2262818        total: 2.87s    remaining: 4.7s
379:    learn: 0.2262334        total: 2.88s    remaining: 4.69s
380:    learn: 0.2260968        total: 2.88s    remaining: 4.68s
381:    learn: 0.2259774        total: 2.89s    remaining: 4.67s
382:    learn: 0.2258764        total: 2.9s     remaining: 4.67s
383:    learn: 0.2257626        total: 2.9s     remaining: 4.66s
384:    learn: 0.2257237        total: 2.91s    remaining: 4.65s
385:    learn: 0.2256449        total: 2.92s    remaining: 4.65s
386:    learn: 0.2255876        total: 2.93s    remaining: 4.64s
387:    learn: 0.2254537        total: 2.94s    remaining: 4.63s
```

```
388:    learn: 0.2252619        total: 2.94s    remaining: 4.63s
389:    learn: 0.2251036        total: 2.95s    remaining: 4.62s
390:    learn: 0.2250677        total: 2.96s    remaining: 4.61s
391:    learn: 0.2249844        total: 2.97s    remaining: 4.6s
392:    learn: 0.2248788        total: 2.97s    remaining: 4.59s
393:    learn: 0.2247836        total: 2.98s    remaining: 4.58s
394:    learn: 0.2247323        total: 2.99s    remaining: 4.58s
395:    learn: 0.2245737        total: 3s       remaining: 4.57s
396:    learn: 0.2244970        total: 3s       remaining: 4.56s
397:    learn: 0.2243579        total: 3.01s    remaining: 4.55s
398:    learn: 0.2243232        total: 3.02s    remaining: 4.54s
399:    learn: 0.2242507        total: 3.02s    remaining: 4.54s
400:    learn: 0.2242226        total: 3.03s    remaining: 4.53s
401:    learn: 0.2241653        total: 3.04s    remaining: 4.52s
402:    learn: 0.2241309        total: 3.04s    remaining: 4.51s
403:    learn: 0.2240677        total: 3.05s    remaining: 4.5s
404:    learn: 0.2239700        total: 3.06s    remaining: 4.5s
405:    learn: 0.2238444        total: 3.07s    remaining: 4.49s
406:    learn: 0.2237352        total: 3.08s    remaining: 4.48s
407:    learn: 0.2236688        total: 3.08s    remaining: 4.47s
408:    learn: 0.2235846        total: 3.09s    remaining: 4.47s
409:    learn: 0.2235432        total: 3.1s     remaining: 4.46s
410:    learn: 0.2234111        total: 3.11s    remaining: 4.46s
411:    learn: 0.2233647        total: 3.12s    remaining: 4.45s
412:    learn: 0.2233165        total: 3.12s    remaining: 4.44s
413:    learn: 0.2232650        total: 3.13s    remaining: 4.43s
414:    learn: 0.2231919        total: 3.14s    remaining: 4.42s
415:    learn: 0.2231073        total: 3.15s    remaining: 4.42s
416:    learn: 0.2229961        total: 3.15s    remaining: 4.41s
417:    learn: 0.2228271        total: 3.16s    remaining: 4.4s
418:    learn: 0.2227242        total: 3.17s    remaining: 4.4s
419:    learn: 0.2225947        total: 3.18s    remaining: 4.39s
420:    learn: 0.2224864        total: 3.19s    remaining: 4.38s
421:    learn: 0.2224188        total: 3.19s    remaining: 4.38s
422:    learn: 0.2223086        total: 3.2s     remaining: 4.37s
423:    learn: 0.2222458        total: 3.21s    remaining: 4.36s
424:    learn: 0.2222007        total: 3.22s    remaining: 4.35s
425:    learn: 0.2220674        total: 3.23s    remaining: 4.35s
426:    learn: 0.2220083        total: 3.23s    remaining: 4.34s
427:    learn: 0.2219676        total: 3.24s    remaining: 4.33s
428:    learn: 0.2218429        total: 3.25s    remaining: 4.32s
429:    learn: 0.2217797        total: 3.25s    remaining: 4.31s
430:    learn: 0.2217200        total: 3.26s    remaining: 4.31s
431:    learn: 0.2216614        total: 3.27s    remaining: 4.3s
432:    learn: 0.2215572        total: 3.28s    remaining: 4.29s
433:    learn: 0.2214659        total: 3.28s    remaining: 4.28s
434:    learn: 0.2214066        total: 3.29s    remaining: 4.28s
435:    learn: 0.2212857        total: 3.3s     remaining: 4.27s
```

```
436:    learn: 0.2211846        total: 3.31s    remaining: 4.26s
437:    learn: 0.2210528        total: 3.32s    remaining: 4.25s
438:    learn: 0.2209282        total: 3.32s    remaining: 4.25s
439:    learn: 0.2208486        total: 3.33s    remaining: 4.24s
440:    learn: 0.2207569        total: 3.34s    remaining: 4.23s
441:    learn: 0.2206157        total: 3.35s    remaining: 4.22s
442:    learn: 0.2205697        total: 3.35s    remaining: 4.22s
443:    learn: 0.2205382        total: 3.36s    remaining: 4.21s
444:    learn: 0.2204356        total: 3.37s    remaining: 4.2s
445:    learn: 0.2203897        total: 3.38s    remaining: 4.19s
446:    learn: 0.2203011        total: 3.38s    remaining: 4.18s
447:    learn: 0.2202312        total: 3.39s    remaining: 4.18s
448:    learn: 0.2201528        total: 3.4s     remaining: 4.17s
449:    learn: 0.2200231        total: 3.4s     remaining: 4.16s
450:    learn: 0.2199559        total: 3.41s    remaining: 4.15s
451:    learn: 0.2198754        total: 3.42s    remaining: 4.14s
452:    learn: 0.2197869        total: 3.43s    remaining: 4.14s
453:    learn: 0.2196918        total: 3.43s    remaining: 4.13s
454:    learn: 0.2196355        total: 3.44s    remaining: 4.12s
455:    learn: 0.2195629        total: 3.45s    remaining: 4.11s
456:    learn: 0.2194322        total: 3.46s    remaining: 4.11s
457:    learn: 0.2193404        total: 3.46s    remaining: 4.1s
458:    learn: 0.2193211        total: 3.47s    remaining: 4.09s
459:    learn: 0.2192088        total: 3.48s    remaining: 4.08s
460:    learn: 0.2190625        total: 3.49s    remaining: 4.08s
461:    learn: 0.2189671        total: 3.5s     remaining: 4.07s
462:    learn: 0.2188978        total: 3.5s     remaining: 4.06s
463:    learn: 0.2187971        total: 3.51s    remaining: 4.05s
464:    learn: 0.2187408        total: 3.52s    remaining: 4.05s
465:    learn: 0.2186729        total: 3.52s    remaining: 4.04s
466:    learn: 0.2184503        total: 3.53s    remaining: 4.03s
467:    learn: 0.2183845        total: 3.54s    remaining: 4.02s
468:    learn: 0.2182597        total: 3.55s    remaining: 4.02s
469:    learn: 0.2182069        total: 3.55s    remaining: 4.01s
470:    learn: 0.2181071        total: 3.56s    remaining: 4s
471:    learn: 0.2180163        total: 3.57s    remaining: 3.99s
472:    learn: 0.2178752        total: 3.58s    remaining: 3.98s
473:    learn: 0.2177189        total: 3.58s    remaining: 3.98s
474:    learn: 0.2176958        total: 3.59s    remaining: 3.97s
475:    learn: 0.2176358        total: 3.6s     remaining: 3.96s
476:    learn: 0.2175782        total: 3.61s    remaining: 3.95s
477:    learn: 0.2174843        total: 3.61s    remaining: 3.95s
478:    learn: 0.2172988        total: 3.62s    remaining: 3.94s
479:    learn: 0.2172524        total: 3.63s    remaining: 3.93s
480:    learn: 0.2170269        total: 3.64s    remaining: 3.92s
481:    learn: 0.2169538        total: 3.64s    remaining: 3.92s
482:    learn: 0.2168545        total: 3.65s    remaining: 3.91s
483:    learn: 0.2167904        total: 3.66s    remaining: 3.9s
```

```
484:    learn: 0.2167127        total: 3.66s    remaining: 3.89s
485:    learn: 0.2166282        total: 3.67s    remaining: 3.88s
486:    learn: 0.2165550        total: 3.68s    remaining: 3.87s
487:    learn: 0.2164512        total: 3.69s    remaining: 3.87s
488:    learn: 0.2164008        total: 3.69s    remaining: 3.86s
489:    learn: 0.2163583        total: 3.7s     remaining: 3.85s
490:    learn: 0.2162744        total: 3.71s    remaining: 3.85s
491:    learn: 0.2162284        total: 3.72s    remaining: 3.84s
492:    learn: 0.2162073        total: 3.72s    remaining: 3.83s
493:    learn: 0.2161554        total: 3.73s    remaining: 3.82s
494:    learn: 0.2160619        total: 3.74s    remaining: 3.81s
495:    learn: 0.2160146        total: 3.75s    remaining: 3.81s
496:    learn: 0.2159707        total: 3.75s    remaining: 3.8s
497:    learn: 0.2158674        total: 3.76s    remaining: 3.79s
498:    learn: 0.2157750        total: 3.77s    remaining: 3.78s
499:    learn: 0.2157325        total: 3.77s    remaining: 3.77s
500:    learn: 0.2155779        total: 3.78s    remaining: 3.77s
501:    learn: 0.2154474        total: 3.79s    remaining: 3.76s
502:    learn: 0.2153558        total: 3.8s     remaining: 3.75s
503:    learn: 0.2152838        total: 3.8s     remaining: 3.74s
504:    learn: 0.2151942        total: 3.81s    remaining: 3.74s
505:    learn: 0.2150877        total: 3.82s    remaining: 3.73s
506:    learn: 0.2149908        total: 3.83s    remaining: 3.72s
507:    learn: 0.2149258        total: 3.83s    remaining: 3.71s
508:    learn: 0.2148639        total: 3.84s    remaining: 3.71s
509:    learn: 0.2147340        total: 3.85s    remaining: 3.7s
510:    learn: 0.2145914        total: 3.86s    remaining: 3.69s
511:    learn: 0.2145000        total: 3.86s    remaining: 3.68s
512:    learn: 0.2144317        total: 3.87s    remaining: 3.67s
513:    learn: 0.2143712        total: 3.88s    remaining: 3.67s
514:    learn: 0.2143240        total: 3.89s    remaining: 3.66s
515:    learn: 0.2142572        total: 3.9s     remaining: 3.65s
516:    learn: 0.2141910        total: 3.9s     remaining: 3.65s
517:    learn: 0.2140794        total: 3.91s    remaining: 3.64s
518:    learn: 0.2138832        total: 3.92s    remaining: 3.63s
519:    learn: 0.2137814        total: 3.93s    remaining: 3.62s
520:    learn: 0.2136931        total: 3.93s    remaining: 3.62s
521:    learn: 0.2136579        total: 3.94s    remaining: 3.61s
522:    learn: 0.2135845        total: 3.95s    remaining: 3.6s
523:    learn: 0.2135323        total: 3.96s    remaining: 3.59s
524:    learn: 0.2134343        total: 3.96s    remaining: 3.59s
525:    learn: 0.2133472        total: 3.97s    remaining: 3.58s
526:    learn: 0.2133158        total: 3.98s    remaining: 3.57s
527:    learn: 0.2132324        total: 3.98s    remaining: 3.56s
528:    learn: 0.2131530        total: 3.99s    remaining: 3.56s
529:    learn: 0.2131089        total: 4s       remaining: 3.55s
530:    learn: 0.2130275        total: 4.01s    remaining: 3.54s
531:    learn: 0.2129845        total: 4.01s    remaining: 3.53s
```

```
532:    learn: 0.2128825        total: 4.02s    remaining: 3.52s
533:    learn: 0.2128088        total: 4.03s    remaining: 3.52s
534:    learn: 0.2127647        total: 4.04s    remaining: 3.51s
535:    learn: 0.2127082        total: 4.05s    remaining: 3.5s
536:    learn: 0.2126624        total: 4.05s    remaining: 3.5s
537:    learn: 0.2125906        total: 4.06s    remaining: 3.49s
538:    learn: 0.2124875        total: 4.07s    remaining: 3.48s
539:    learn: 0.2124304        total: 4.08s    remaining: 3.48s
540:    learn: 0.2123290        total: 4.09s    remaining: 3.47s
541:    learn: 0.2122656        total: 4.1s     remaining: 3.46s
542:    learn: 0.2122148        total: 4.11s    remaining: 3.46s
543:    learn: 0.2121301        total: 4.11s    remaining: 3.45s
544:    learn: 0.2120572        total: 4.12s    remaining: 3.44s
545:    learn: 0.2119512        total: 4.13s    remaining: 3.43s
546:    learn: 0.2119015        total: 4.14s    remaining: 3.42s
547:    learn: 0.2118600        total: 4.14s    remaining: 3.42s
548:    learn: 0.2117717        total: 4.15s    remaining: 3.41s
549:    learn: 0.2116894        total: 4.16s    remaining: 3.4s
550:    learn: 0.2116062        total: 4.17s    remaining: 3.4s
551:    learn: 0.2115501        total: 4.17s    remaining: 3.39s
552:    learn: 0.2114691        total: 4.18s    remaining: 3.38s
553:    learn: 0.2113682        total: 4.19s    remaining: 3.37s
554:    learn: 0.2112850        total: 4.2s     remaining: 3.36s
555:    learn: 0.2112574        total: 4.2s     remaining: 3.36s
556:    learn: 0.2111914        total: 4.21s    remaining: 3.35s
557:    learn: 0.2111365        total: 4.22s    remaining: 3.34s
558:    learn: 0.2110787        total: 4.22s    remaining: 3.33s
559:    learn: 0.2109727        total: 4.23s    remaining: 3.33s
560:    learn: 0.2108773        total: 4.24s    remaining: 3.32s
561:    learn: 0.2107779        total: 4.25s    remaining: 3.31s
562:    learn: 0.2106571        total: 4.25s    remaining: 3.3s
563:    learn: 0.2105995        total: 4.26s    remaining: 3.3s
564:    learn: 0.2105078        total: 4.27s    remaining: 3.29s
565:    learn: 0.2104625        total: 4.28s    remaining: 3.28s
566:    learn: 0.2104028        total: 4.29s    remaining: 3.27s
567:    learn: 0.2103366        total: 4.29s    remaining: 3.27s
568:    learn: 0.2102812        total: 4.3s     remaining: 3.26s
569:    learn: 0.2102057        total: 4.31s    remaining: 3.25s
570:    learn: 0.2101556        total: 4.32s    remaining: 3.24s
571:    learn: 0.2100525        total: 4.33s    remaining: 3.24s
572:    learn: 0.2099822        total: 4.33s    remaining: 3.23s
573:    learn: 0.2098505        total: 4.34s    remaining: 3.22s
574:    learn: 0.2097860        total: 4.35s    remaining: 3.21s
575:    learn: 0.2097330        total: 4.35s    remaining: 3.21s
576:    learn: 0.2096586        total: 4.36s    remaining: 3.2s
577:    learn: 0.2095808        total: 4.37s    remaining: 3.19s
578:    learn: 0.2095614        total: 4.38s    remaining: 3.18s
579:    learn: 0.2094197        total: 4.38s    remaining: 3.17s
```

```
580:    learn: 0.2093089        total: 4.39s    remaining: 3.17s
581:    learn: 0.2091560        total: 4.4s     remaining: 3.16s
582:    learn: 0.2090873        total: 4.41s    remaining: 3.15s
583:    learn: 0.2090253        total: 4.42s    remaining: 3.15s
584:    learn: 0.2089739        total: 4.42s    remaining: 3.14s
585:    learn: 0.2089173        total: 4.43s    remaining: 3.13s
586:    learn: 0.2088275        total: 4.44s    remaining: 3.12s
587:    learn: 0.2087367        total: 4.45s    remaining: 3.11s
588:    learn: 0.2086395        total: 4.45s    remaining: 3.11s
589:    learn: 0.2085492        total: 4.46s    remaining: 3.1s
590:    learn: 0.2084922        total: 4.47s    remaining: 3.09s
591:    learn: 0.2084704        total: 4.47s    remaining: 3.08s
592:    learn: 0.2084381        total: 4.48s    remaining: 3.08s
593:    learn: 0.2083451        total: 4.49s    remaining: 3.07s
594:    learn: 0.2083086        total: 4.5s     remaining: 3.06s
595:    learn: 0.2082664        total: 4.5s     remaining: 3.05s
596:    learn: 0.2082191        total: 4.51s    remaining: 3.04s
597:    learn: 0.2081259        total: 4.52s    remaining: 3.04s
598:    learn: 0.2080776        total: 4.53s    remaining: 3.03s
599:    learn: 0.2080208        total: 4.53s    remaining: 3.02s
600:    learn: 0.2079586        total: 4.54s    remaining: 3.01s
601:    learn: 0.2078378        total: 4.55s    remaining: 3.01s
602:    learn: 0.2077653        total: 4.56s    remaining: 3s
603:    learn: 0.2076538        total: 4.56s    remaining: 2.99s
604:    learn: 0.2076220        total: 4.57s    remaining: 2.98s
605:    learn: 0.2075544        total: 4.58s    remaining: 2.98s
606:    learn: 0.2074818        total: 4.59s    remaining: 2.97s
607:    learn: 0.2074265        total: 4.59s    remaining: 2.96s
608:    learn: 0.2073601        total: 4.6s     remaining: 2.95s
609:    learn: 0.2072802        total: 4.61s    remaining: 2.95s
610:    learn: 0.2072294        total: 4.62s    remaining: 2.94s
611:    learn: 0.2071438        total: 4.62s    remaining: 2.93s
612:    learn: 0.2070929        total: 4.63s    remaining: 2.92s
613:    learn: 0.2070441        total: 4.64s    remaining: 2.92s
614:    learn: 0.2069790        total: 4.64s    remaining: 2.91s
615:    learn: 0.2068998        total: 4.65s    remaining: 2.9s
616:    learn: 0.2068329        total: 4.66s    remaining: 2.89s
617:    learn: 0.2067300        total: 4.67s    remaining: 2.89s
618:    learn: 0.2067058        total: 4.68s    remaining: 2.88s
619:    learn: 0.2066500        total: 4.68s    remaining: 2.87s
620:    learn: 0.2066243        total: 4.69s    remaining: 2.86s
621:    learn: 0.2064804        total: 4.7s     remaining: 2.85s
622:    learn: 0.2063718        total: 4.71s    remaining: 2.85s
623:    learn: 0.2063429        total: 4.71s    remaining: 2.84s
624:    learn: 0.2062949        total: 4.72s    remaining: 2.83s
625:    learn: 0.2062084        total: 4.73s    remaining: 2.82s
626:    learn: 0.2061837        total: 4.73s    remaining: 2.82s
627:    learn: 0.2060983        total: 4.74s    remaining: 2.81s
```

```
628:    learn: 0.2060141        total: 4.75s    remaining: 2.8s
629:    learn: 0.2058776        total: 4.76s    remaining: 2.79s
630:    learn: 0.2058381        total: 4.76s    remaining: 2.79s
631:    learn: 0.2057889        total: 4.77s    remaining: 2.78s
632:    learn: 0.2056944        total: 4.78s    remaining: 2.77s
633:    learn: 0.2055986        total: 4.79s    remaining: 2.76s
634:    learn: 0.2055535        total: 4.79s    remaining: 2.75s
635:    learn: 0.2055170        total: 4.8s     remaining: 2.75s
636:    learn: 0.2054721        total: 4.81s    remaining: 2.74s
637:    learn: 0.2054137        total: 4.81s    remaining: 2.73s
638:    learn: 0.2053815        total: 4.82s    remaining: 2.72s
639:    learn: 0.2053029        total: 4.83s    remaining: 2.72s
640:    learn: 0.2052294        total: 4.84s    remaining: 2.71s
641:    learn: 0.2051917        total: 4.84s    remaining: 2.7s
642:    learn: 0.2051274        total: 4.85s    remaining: 2.69s
643:    learn: 0.2050544        total: 4.86s    remaining: 2.69s
644:    learn: 0.2049930        total: 4.87s    remaining: 2.68s
645:    learn: 0.2049042        total: 4.88s    remaining: 2.67s
646:    learn: 0.2048560        total: 4.88s    remaining: 2.66s
647:    learn: 0.2047780        total: 4.89s    remaining: 2.66s
648:    learn: 0.2047438        total: 4.9s     remaining: 2.65s
649:    learn: 0.2046579        total: 4.9s     remaining: 2.64s
650:    learn: 0.2045901        total: 4.91s    remaining: 2.63s
651:    learn: 0.2045122        total: 4.92s    remaining: 2.63s
652:    learn: 0.2044526        total: 4.93s    remaining: 2.62s
653:    learn: 0.2043807        total: 4.93s    remaining: 2.61s
654:    learn: 0.2043198        total: 4.94s    remaining: 2.6s
655:    learn: 0.2042801        total: 4.95s    remaining: 2.6s
656:    learn: 0.2041363        total: 4.96s    remaining: 2.59s
657:    learn: 0.2040231        total: 4.96s    remaining: 2.58s
658:    learn: 0.2039576        total: 4.97s    remaining: 2.57s
659:    learn: 0.2038764        total: 4.98s    remaining: 2.56s
660:    learn: 0.2037908        total: 4.99s    remaining: 2.56s
661:    learn: 0.2037704        total: 5s       remaining: 2.55s
662:    learn: 0.2037407        total: 5s       remaining: 2.54s
663:    learn: 0.2036968        total: 5.01s    remaining: 2.54s
664:    learn: 0.2035788        total: 5.02s    remaining: 2.53s
665:    learn: 0.2035348        total: 5.03s    remaining: 2.52s
666:    learn: 0.2034750        total: 5.03s    remaining: 2.51s
667:    learn: 0.2033759        total: 5.04s    remaining: 2.51s
668:    learn: 0.2033245        total: 5.05s    remaining: 2.5s
669:    learn: 0.2032323        total: 5.06s    remaining: 2.49s
670:    learn: 0.2031865        total: 5.07s    remaining: 2.48s
671:    learn: 0.2031721        total: 5.07s    remaining: 2.48s
672:    learn: 0.2030403        total: 5.08s    remaining: 2.47s
673:    learn: 0.2030016        total: 5.09s    remaining: 2.46s
674:    learn: 0.2029468        total: 5.1s     remaining: 2.45s
675:    learn: 0.2028498        total: 5.11s    remaining: 2.45s
```

```
676:    learn: 0.2027913        total: 5.11s    remaining: 2.44s
677:    learn: 0.2027096        total: 5.12s    remaining: 2.43s
678:    learn: 0.2026506        total: 5.13s    remaining: 2.42s
679:    learn: 0.2025858        total: 5.13s    remaining: 2.42s
680:    learn: 0.2024874        total: 5.14s    remaining: 2.41s
681:    learn: 0.2024590        total: 5.15s    remaining: 2.4s
682:    learn: 0.2023826        total: 5.16s    remaining: 2.39s
683:    learn: 0.2023222        total: 5.16s    remaining: 2.39s
684:    learn: 0.2022535        total: 5.17s    remaining: 2.38s
685:    learn: 0.2021936        total: 5.18s    remaining: 2.37s
686:    learn: 0.2021278        total: 5.19s    remaining: 2.36s
687:    learn: 0.2020417        total: 5.2s     remaining: 2.35s
688:    learn: 0.2019385        total: 5.2s     remaining: 2.35s
689:    learn: 0.2019124        total: 5.21s    remaining: 2.34s
690:    learn: 0.2018633        total: 5.22s    remaining: 2.33s
691:    learn: 0.2018092        total: 5.22s    remaining: 2.33s
692:    learn: 0.2017258        total: 5.23s    remaining: 2.32s
693:    learn: 0.2016646        total: 5.24s    remaining: 2.31s
694:    learn: 0.2015796        total: 5.25s    remaining: 2.3s
695:    learn: 0.2015004        total: 5.26s    remaining: 2.29s
696:    learn: 0.2014756        total: 5.26s    remaining: 2.29s
697:    learn: 0.2014274        total: 5.27s    remaining: 2.28s
698:    learn: 0.2013633        total: 5.28s    remaining: 2.27s
699:    learn: 0.2012688        total: 5.29s    remaining: 2.27s
700:    learn: 0.2011836        total: 5.29s    remaining: 2.26s
701:    learn: 0.2011301        total: 5.3s     remaining: 2.25s
702:    learn: 0.2010328        total: 5.31s    remaining: 2.24s
703:    learn: 0.2009835        total: 5.32s    remaining: 2.23s
704:    learn: 0.2009097        total: 5.32s    remaining: 2.23s
705:    learn: 0.2008703        total: 5.33s    remaining: 2.22s
706:    learn: 0.2007976        total: 5.34s    remaining: 2.21s
707:    learn: 0.2007623        total: 5.34s    remaining: 2.2s
708:    learn: 0.2007035        total: 5.35s    remaining: 2.2s
709:    learn: 0.2006596        total: 5.36s    remaining: 2.19s
710:    learn: 0.2005877        total: 5.37s    remaining: 2.18s
711:    learn: 0.2005293        total: 5.37s    remaining: 2.17s
712:    learn: 0.2004479        total: 5.38s    remaining: 2.17s
713:    learn: 0.2003834        total: 5.39s    remaining: 2.16s
714:    learn: 0.2002354        total: 5.4s     remaining: 2.15s
715:    learn: 0.2001774        total: 5.41s    remaining: 2.14s
716:    learn: 0.2000368        total: 5.41s    remaining: 2.14s
717:    learn: 0.1999563        total: 5.42s    remaining: 2.13s
718:    learn: 0.1999111        total: 5.43s    remaining: 2.12s
719:    learn: 0.1998470        total: 5.44s    remaining: 2.12s
720:    learn: 0.1997855        total: 5.45s    remaining: 2.11s
721:    learn: 0.1997461        total: 5.45s    remaining: 2.1s
722:    learn: 0.1997061        total: 5.46s    remaining: 2.09s
723:    learn: 0.1995973        total: 5.47s    remaining: 2.08s
```

```
724:    learn: 0.1995265        total: 5.47s    remaining: 2.08s
725:    learn: 0.1994535        total: 5.48s    remaining: 2.07s
726:    learn: 0.1994012        total: 5.49s    remaining: 2.06s
727:    learn: 0.1993399        total: 5.5s     remaining: 2.05s
728:    learn: 0.1992756        total: 5.51s    remaining: 2.05s
729:    learn: 0.1992022        total: 5.51s    remaining: 2.04s
730:    learn: 0.1991639        total: 5.52s    remaining: 2.03s
731:    learn: 0.1991187        total: 5.53s    remaining: 2.02s
732:    learn: 0.1989726        total: 5.54s    remaining: 2.02s
733:    learn: 0.1989069        total: 5.54s    remaining: 2.01s
734:    learn: 0.1988103        total: 5.55s    remaining: 2s
735:    learn: 0.1987492        total: 5.56s    remaining: 1.99s
736:    learn: 0.1987144        total: 5.57s    remaining: 1.99s
737:    learn: 0.1986753        total: 5.57s    remaining: 1.98s
738:    learn: 0.1986054        total: 5.58s    remaining: 1.97s
739:    learn: 0.1985146        total: 5.59s    remaining: 1.96s
740:    learn: 0.1984474        total: 5.59s    remaining: 1.96s
741:    learn: 0.1983590        total: 5.6s     remaining: 1.95s
742:    learn: 0.1982303        total: 5.61s    remaining: 1.94s
743:    learn: 0.1981979        total: 5.62s    remaining: 1.93s
744:    learn: 0.1981290        total: 5.62s    remaining: 1.92s
745:    learn: 0.1980912        total: 5.63s    remaining: 1.92s
746:    learn: 0.1980311        total: 5.64s    remaining: 1.91s
747:    learn: 0.1979270        total: 5.65s    remaining: 1.9s
748:    learn: 0.1978335        total: 5.65s    remaining: 1.89s
749:    learn: 0.1977793        total: 5.66s    remaining: 1.89s
750:    learn: 0.1976946        total: 5.67s    remaining: 1.88s
751:    learn: 0.1976728        total: 5.67s    remaining: 1.87s
752:    learn: 0.1976367        total: 5.68s    remaining: 1.86s
753:    learn: 0.1976077        total: 5.69s    remaining: 1.86s
754:    learn: 0.1975477        total: 5.7s     remaining: 1.85s
755:    learn: 0.1975048        total: 5.7s     remaining: 1.84s
756:    learn: 0.1974459        total: 5.71s    remaining: 1.83s
757:    learn: 0.1973757        total: 5.72s    remaining: 1.83s
758:    learn: 0.1973127        total: 5.73s    remaining: 1.82s
759:    learn: 0.1972740        total: 5.74s    remaining: 1.81s
760:    learn: 0.1972096        total: 5.74s    remaining: 1.8s
761:    learn: 0.1971398        total: 5.75s    remaining: 1.79s
762:    learn: 0.1971178        total: 5.76s    remaining: 1.79s
763:    learn: 0.1970431        total: 5.76s    remaining: 1.78s
764:    learn: 0.1969834        total: 5.77s    remaining: 1.77s
765:    learn: 0.1969106        total: 5.78s    remaining: 1.76s
766:    learn: 0.1968160        total: 5.79s    remaining: 1.76s
767:    learn: 0.1967485        total: 5.79s    remaining: 1.75s
768:    learn: 0.1967244        total: 5.8s     remaining: 1.74s
769:    learn: 0.1966938        total: 5.81s    remaining: 1.73s
770:    learn: 0.1966296        total: 5.82s    remaining: 1.73s
771:    learn: 0.1965229        total: 5.82s    remaining: 1.72s
```

```
772:    learn: 0.1964724        total: 5.83s    remaining: 1.71s
773:    learn: 0.1964514        total: 5.84s    remaining: 1.7s
774:    learn: 0.1963649        total: 5.84s    remaining: 1.7s
775:    learn: 0.1963105        total: 5.85s    remaining: 1.69s
776:    learn: 0.1962579        total: 5.86s    remaining: 1.68s
777:    learn: 0.1961944        total: 5.87s    remaining: 1.67s
778:    learn: 0.1961309        total: 5.88s    remaining: 1.67s
779:    learn: 0.1960807        total: 5.88s    remaining: 1.66s
780:    learn: 0.1960543        total: 5.89s    remaining: 1.65s
781:    learn: 0.1959667        total: 5.9s     remaining: 1.64s
782:    learn: 0.1959181        total: 5.91s    remaining: 1.64s
783:    learn: 0.1958692        total: 5.92s    remaining: 1.63s
784:    learn: 0.1958334        total: 5.92s    remaining: 1.62s
785:    learn: 0.1957437        total: 5.93s    remaining: 1.61s
786:    learn: 0.1956722        total: 5.94s    remaining: 1.61s
787:    learn: 0.1956493        total: 5.95s    remaining: 1.6s
788:    learn: 0.1955722        total: 5.95s    remaining: 1.59s
789:    learn: 0.1955400        total: 5.96s    remaining: 1.58s
790:    learn: 0.1954450        total: 5.97s    remaining: 1.58s
791:    learn: 0.1953964        total: 5.97s    remaining: 1.57s
792:    learn: 0.1953385        total: 5.98s    remaining: 1.56s
793:    learn: 0.1952683        total: 5.99s    remaining: 1.55s
794:    learn: 0.1952172        total: 6s       remaining: 1.55s
795:    learn: 0.1951402        total: 6.01s    remaining: 1.54s
796:    learn: 0.1950873        total: 6.02s    remaining: 1.53s
797:    learn: 0.1950478        total: 6.03s    remaining: 1.52s
798:    learn: 0.1949386        total: 6.03s    remaining: 1.52s
799:    learn: 0.1948974        total: 6.04s    remaining: 1.51s
800:    learn: 0.1948138        total: 6.05s    remaining: 1.5s
801:    learn: 0.1947522        total: 6.06s    remaining: 1.5s
802:    learn: 0.1946937        total: 6.06s    remaining: 1.49s
803:    learn: 0.1946510        total: 6.07s    remaining: 1.48s
804:    learn: 0.1946040        total: 6.08s    remaining: 1.47s
805:    learn: 0.1945332        total: 6.09s    remaining: 1.47s
806:    learn: 0.1944475        total: 6.09s    remaining: 1.46s
807:    learn: 0.1944031        total: 6.1s     remaining: 1.45s
808:    learn: 0.1943403        total: 6.11s    remaining: 1.44s
809:    learn: 0.1943061        total: 6.12s    remaining: 1.43s
810:    learn: 0.1942064        total: 6.12s    remaining: 1.43s
811:    learn: 0.1941315        total: 6.13s    remaining: 1.42s
812:    learn: 0.1940511        total: 6.14s    remaining: 1.41s
813:    learn: 0.1940340        total: 6.14s    remaining: 1.4s
814:    learn: 0.1939471        total: 6.15s    remaining: 1.4s
815:    learn: 0.1939064        total: 6.16s    remaining: 1.39s
816:    learn: 0.1938670        total: 6.17s    remaining: 1.38s
817:    learn: 0.1938250        total: 6.17s    remaining: 1.37s
818:    learn: 0.1937853        total: 6.18s    remaining: 1.37s
819:    learn: 0.1937383        total: 6.19s    remaining: 1.36s
```

```
820:    learn: 0.1935879        total: 6.2s     remaining: 1.35s
821:    learn: 0.1935409        total: 6.21s    remaining: 1.34s
822:    learn: 0.1934932        total: 6.22s    remaining: 1.34s
823:    learn: 0.1933936        total: 6.22s    remaining: 1.33s
824:    learn: 0.1932953        total: 6.23s    remaining: 1.32s
825:    learn: 0.1932216        total: 6.24s    remaining: 1.31s
826:    learn: 0.1931494        total: 6.25s    remaining: 1.31s
827:    learn: 0.1930794        total: 6.25s    remaining: 1.3s
828:    learn: 0.1930668        total: 6.26s    remaining: 1.29s
829:    learn: 0.1930472        total: 6.27s    remaining: 1.28s
830:    learn: 0.1930318        total: 6.27s    remaining: 1.28s
831:    learn: 0.1930176        total: 6.28s    remaining: 1.27s
832:    learn: 0.1929686        total: 6.29s    remaining: 1.26s
833:    learn: 0.1928804        total: 6.3s     remaining: 1.25s
834:    learn: 0.1928028        total: 6.3s     remaining: 1.25s
835:    learn: 0.1927789        total: 6.31s    remaining: 1.24s
836:    learn: 0.1927474        total: 6.32s    remaining: 1.23s
837:    learn: 0.1926553        total: 6.32s    remaining: 1.22s
838:    learn: 0.1925323        total: 6.33s    remaining: 1.22s
839:    learn: 0.1924635        total: 6.34s    remaining: 1.21s
840:    learn: 0.1923549        total: 6.35s    remaining: 1.2s
841:    learn: 0.1923082        total: 6.35s    remaining: 1.19s
842:    learn: 0.1922861        total: 6.36s    remaining: 1.18s
843:    learn: 0.1922571        total: 6.37s    remaining: 1.18s
844:    learn: 0.1922126        total: 6.38s    remaining: 1.17s
845:    learn: 0.1921405        total: 6.39s    remaining: 1.16s
846:    learn: 0.1920682        total: 6.39s    remaining: 1.16s
847:    learn: 0.1920260        total: 6.4s     remaining: 1.15s
848:    learn: 0.1919783        total: 6.41s    remaining: 1.14s
849:    learn: 0.1919481        total: 6.42s    remaining: 1.13s
850:    learn: 0.1918988        total: 6.42s    remaining: 1.12s
851:    learn: 0.1918683        total: 6.43s    remaining: 1.12s
852:    learn: 0.1918506        total: 6.44s    remaining: 1.11s
853:    learn: 0.1917967        total: 6.45s    remaining: 1.1s
854:    learn: 0.1917351        total: 6.45s    remaining: 1.09s
855:    learn: 0.1915941        total: 6.46s    remaining: 1.09s
856:    learn: 0.1915367        total: 6.47s    remaining: 1.08s
857:    learn: 0.1915105        total: 6.48s    remaining: 1.07s
858:    learn: 0.1914693        total: 6.48s    remaining: 1.06s
859:    learn: 0.1914122        total: 6.49s    remaining: 1.06s
860:    learn: 0.1913822        total: 6.5s     remaining: 1.05s
861:    learn: 0.1913135        total: 6.51s    remaining: 1.04s
862:    learn: 0.1912729        total: 6.51s    remaining: 1.03s
863:    learn: 0.1912116        total: 6.52s    remaining: 1.03s
864:    learn: 0.1911646        total: 6.53s    remaining: 1.02s
865:    learn: 0.1911063        total: 6.54s    remaining: 1.01s
866:    learn: 0.1910179        total: 6.54s    remaining: 1s
867:    learn: 0.1909211        total: 6.55s    remaining: 996ms
```

```
868:    learn: 0.1908637        total: 6.56s    remaining: 989ms
869:    learn: 0.1908200        total: 6.57s    remaining: 981ms
870:    learn: 0.1907208        total: 6.57s    remaining: 973ms
871:    learn: 0.1906586        total: 6.58s    remaining: 966ms
872:    learn: 0.1905519        total: 6.59s    remaining: 959ms
873:    learn: 0.1904744        total: 6.6s     remaining: 951ms
874:    learn: 0.1904251        total: 6.6s     remaining: 944ms
875:    learn: 0.1903302        total: 6.61s    remaining: 936ms
876:    learn: 0.1902822        total: 6.62s    remaining: 928ms
877:    learn: 0.1902184        total: 6.63s    remaining: 921ms
878:    learn: 0.1901223        total: 6.63s    remaining: 913ms
879:    learn: 0.1900891        total: 6.64s    remaining: 906ms
880:    learn: 0.1900146        total: 6.65s    remaining: 898ms
881:    learn: 0.1899865        total: 6.66s    remaining: 891ms
882:    learn: 0.1898786        total: 6.67s    remaining: 883ms
883:    learn: 0.1898333        total: 6.67s    remaining: 876ms
884:    learn: 0.1897831        total: 6.68s    remaining: 868ms
885:    learn: 0.1897336        total: 6.69s    remaining: 860ms
886:    learn: 0.1896284        total: 6.69s    remaining: 853ms
887:    learn: 0.1895789        total: 6.7s     remaining: 845ms
888:    learn: 0.1895378        total: 6.71s    remaining: 838ms
889:    learn: 0.1894741        total: 6.72s    remaining: 830ms
890:    learn: 0.1893617        total: 6.72s    remaining: 823ms
891:    learn: 0.1892894        total: 6.73s    remaining: 815ms
892:    learn: 0.1892158        total: 6.74s    remaining: 807ms
893:    learn: 0.1891649        total: 6.75s    remaining: 800ms
894:    learn: 0.1890946        total: 6.75s    remaining: 792ms
895:    learn: 0.1890478        total: 6.76s    remaining: 785ms
896:    learn: 0.1890032        total: 6.77s    remaining: 777ms
897:    learn: 0.1889277        total: 6.78s    remaining: 770ms
898:    learn: 0.1888882        total: 6.78s    remaining: 762ms
899:    learn: 0.1888300        total: 6.79s    remaining: 755ms
900:    learn: 0.1887672        total: 6.8s     remaining: 747ms
901:    learn: 0.1887299        total: 6.81s    remaining: 740ms
902:    learn: 0.1886711        total: 6.82s    remaining: 732ms
903:    learn: 0.1886309        total: 6.82s    remaining: 725ms
904:    learn: 0.1886223        total: 6.83s    remaining: 717ms
905:    learn: 0.1885734        total: 6.84s    remaining: 709ms
906:    learn: 0.1885359        total: 6.84s    remaining: 702ms
907:    learn: 0.1884736        total: 6.85s    remaining: 694ms
908:    learn: 0.1883852        total: 6.86s    remaining: 687ms
909:    learn: 0.1883180        total: 6.87s    remaining: 679ms
910:    learn: 0.1882881        total: 6.88s    remaining: 672ms
911:    learn: 0.1882252        total: 6.88s    remaining: 664ms
912:    learn: 0.1881787        total: 6.89s    remaining: 657ms
913:    learn: 0.1881009        total: 6.9s     remaining: 649ms
914:    learn: 0.1880559        total: 6.91s    remaining: 642ms
915:    learn: 0.1880068        total: 6.91s    remaining: 634ms
```

```
916:    learn: 0.1879715        total: 6.92s    remaining: 627ms
917:    learn: 0.1879015        total: 6.93s    remaining: 619ms
918:    learn: 0.1878359        total: 6.94s    remaining: 612ms
919:    learn: 0.1877634        total: 6.95s    remaining: 604ms
920:    learn: 0.1876419        total: 6.95s    remaining: 597ms
921:    learn: 0.1876077        total: 6.96s    remaining: 589ms
922:    learn: 0.1875878        total: 6.97s    remaining: 581ms
923:    learn: 0.1875139        total: 6.98s    remaining: 574ms
924:    learn: 0.1874486        total: 6.99s    remaining: 567ms
925:    learn: 0.1873814        total: 7s       remaining: 559ms
926:    learn: 0.1873383        total: 7s       remaining: 552ms
927:    learn: 0.1873128        total: 7.01s    remaining: 544ms
928:    learn: 0.1872830        total: 7.02s    remaining: 536ms
929:    learn: 0.1871988        total: 7.03s    remaining: 529ms
930:    learn: 0.1871326        total: 7.03s    remaining: 521ms
931:    learn: 0.1870836        total: 7.04s    remaining: 514ms
932:    learn: 0.1870395        total: 7.05s    remaining: 506ms
933:    learn: 0.1869930        total: 7.06s    remaining: 499ms
934:    learn: 0.1869119        total: 7.06s    remaining: 491ms
935:    learn: 0.1868510        total: 7.07s    remaining: 483ms
936:    learn: 0.1868257        total: 7.08s    remaining: 476ms
937:    learn: 0.1867876        total: 7.08s    remaining: 468ms
938:    learn: 0.1866958        total: 7.09s    remaining: 461ms
939:    learn: 0.1866507        total: 7.1s     remaining: 453ms
940:    learn: 0.1865924        total: 7.11s    remaining: 446ms
941:    learn: 0.1865511        total: 7.12s    remaining: 438ms
942:    learn: 0.1865243        total: 7.12s    remaining: 431ms
943:    learn: 0.1864581        total: 7.13s    remaining: 423ms
944:    learn: 0.1864401        total: 7.14s    remaining: 415ms
945:    learn: 0.1863748        total: 7.14s    remaining: 408ms
946:    learn: 0.1863211        total: 7.15s    remaining: 400ms
947:    learn: 0.1862600        total: 7.16s    remaining: 393ms
948:    learn: 0.1861746        total: 7.17s    remaining: 385ms
949:    learn: 0.1860974        total: 7.17s    remaining: 378ms
950:    learn: 0.1860111        total: 7.18s    remaining: 370ms
951:    learn: 0.1859419        total: 7.19s    remaining: 363ms
952:    learn: 0.1858917        total: 7.2s     remaining: 355ms
953:    learn: 0.1858586        total: 7.21s    remaining: 347ms
954:    learn: 0.1858102        total: 7.21s    remaining: 340ms
955:    learn: 0.1857383        total: 7.22s    remaining: 332ms
956:    learn: 0.1857015        total: 7.23s    remaining: 325ms
957:    learn: 0.1856681        total: 7.24s    remaining: 317ms
958:    learn: 0.1855830        total: 7.24s    remaining: 310ms
959:    learn: 0.1855299        total: 7.25s    remaining: 302ms
960:    learn: 0.1854460        total: 7.26s    remaining: 294ms
961:    learn: 0.1854115        total: 7.26s    remaining: 287ms
962:    learn: 0.1853632        total: 7.27s    remaining: 279ms
963:    learn: 0.1853195        total: 7.28s    remaining: 272ms
```

```
964:    learn: 0.1853081        total: 7.29s    remaining: 264ms
965:    learn: 0.1852460        total: 7.29s    remaining: 257ms
966:    learn: 0.1852103        total: 7.3s     remaining: 249ms
967:    learn: 0.1851994        total: 7.31s    remaining: 242ms
968:    learn: 0.1851468        total: 7.31s    remaining: 234ms
969:    learn: 0.1851014        total: 7.32s    remaining: 226ms
970:    learn: 0.1850430        total: 7.33s    remaining: 219ms
971:    learn: 0.1849917        total: 7.34s    remaining: 211ms
972:    learn: 0.1849634        total: 7.34s    remaining: 204ms
973:    learn: 0.1849318        total: 7.35s    remaining: 196ms
974:    learn: 0.1848958        total: 7.36s    remaining: 189ms
975:    learn: 0.1848467        total: 7.37s    remaining: 181ms
976:    learn: 0.1847920        total: 7.38s    remaining: 174ms
977:    learn: 0.1847599        total: 7.38s    remaining: 166ms
978:    learn: 0.1847127        total: 7.39s    remaining: 159ms
979:    learn: 0.1846836        total: 7.4s     remaining: 151ms
980:    learn: 0.1846409        total: 7.41s    remaining: 143ms
981:    learn: 0.1844917        total: 7.42s    remaining: 136ms
982:    learn: 0.1844351        total: 7.42s    remaining: 128ms
983:    learn: 0.1844116        total: 7.43s    remaining: 121ms
984:    learn: 0.1843610        total: 7.44s    remaining: 113ms
985:    learn: 0.1843426        total: 7.44s    remaining: 106ms
986:    learn: 0.1843272        total: 7.45s    remaining: 98.1ms
987:    learn: 0.1842285        total: 7.46s    remaining: 90.6ms
988:    learn: 0.1841554        total: 7.47s    remaining: 83ms
989:    learn: 0.1841295        total: 7.47s    remaining: 75.5ms
990:    learn: 0.1840886        total: 7.48s    remaining: 67.9ms
991:    learn: 0.1840423        total: 7.49s    remaining: 60.4ms
992:    learn: 0.1839959        total: 7.5s     remaining: 52.8ms
993:    learn: 0.1839268        total: 7.5s     remaining: 45.3ms
994:    learn: 0.1838965        total: 7.51s    remaining: 37.7ms
995:    learn: 0.1838360        total: 7.52s    remaining: 30.2ms
996:    learn: 0.1837401        total: 7.52s    remaining: 22.6ms
997:    learn: 0.1836719        total: 7.53s    remaining: 15.1ms
998:    learn: 0.1835967        total: 7.54s    remaining: 7.55ms
999:    learn: 0.1835216        total: 7.55s    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')
```

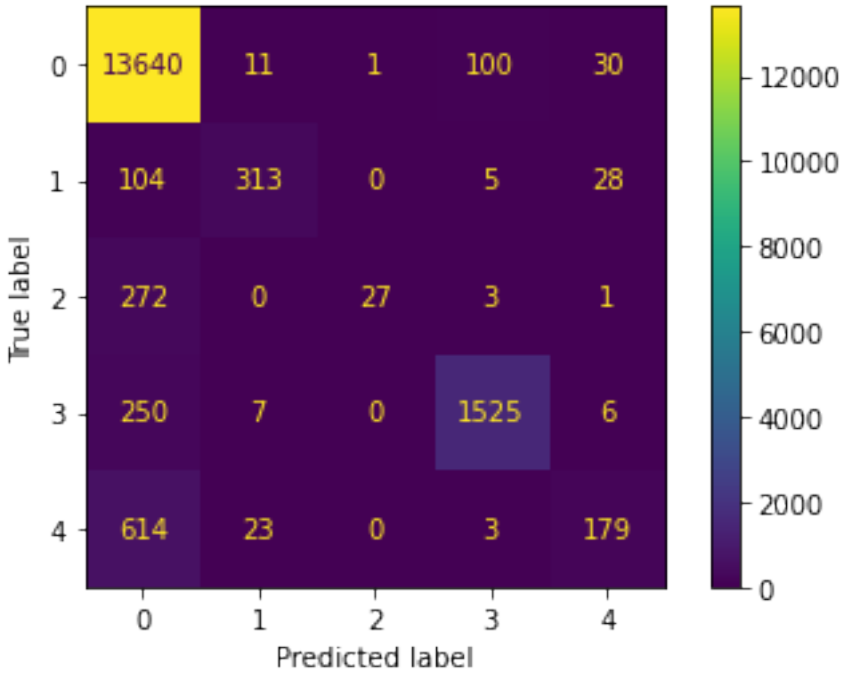
[105]:
```
y_pred = vot_soft.predict(X_test)
```

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

[106]:
```
91.49457472873644
```

[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
0x7fca9336e2b0>
```



[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
```

[125]:
```
              DT          GR      NPHI    RHOB  FACIES
0       133.4417   87.315400  0.468200  2.2995       0
1       132.4196   88.541200  0.458500  2.2981       0
2       133.3569   87.576400  0.454300  2.2950       0
3       134.7392   86.036100  0.482700  2.2907       0
4       135.7694   85.039300  0.536100  2.2856       0
…          …          …        …      …     …
29560   126.6800  102.326070  0.506785  2.4993       0
29561   127.9872  102.490830  0.510428  2.4997       0
29562   127.9657  102.498159  0.510361  2.4999       0
29563   128.9050  102.607440  0.512985  2.5000       0
29564   128.4784  102.560015  0.511792  2.5000       0

[29565 rows x 5 columns]
```

[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
0       133.4417   87.315400  0.468200  2.2995       0
1       132.4196   88.541200  0.458500  2.2981       0
2       133.3569   87.576400  0.454300  2.2950       0
3       134.7392   86.036100  0.482700  2.2907       0
4       135.7694   85.039300  0.536100  2.2856       0
…          …          …        …      …     …
29560   126.6800  102.326070  0.506785  2.4993       0
29561   127.9872  102.490830  0.510428  2.4997       0
29562   127.9657  102.498159  0.510361  2.4999       0
29563   128.9050  102.607440  0.512985  2.5000       0
29564   128.4784  102.560015  0.511792  2.5000       0

[29556 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
0	133.4417	87.315400	0.468200	2.2995	0
1	132.4196	88.541200	0.458500	2.2981	0
2	133.3569	87.576400	0.454300	2.2950	0
3	134.7392	86.036100	0.482700	2.2907	0
4	135.7694	85.039300	0.536100	2.2856	0
…	…	…	…	…	…
29560	126.6800	102.326070	0.506785	2.4993	0
29561	127.9872	102.490830	0.510428	2.4997	0
29562	127.9657	102.498159	0.510361	2.4999	0
29563	128.9050	102.607440	0.512985	2.5000	0
29564	128.4784	102.560015	0.511792	2.5000	0

```
[29556 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

```
[73 rows x 5 columns]
(73, 5)
```

	DT	GR	NPHI	RHOB	FACIES
0	133.4417	87.315400	0.468200	2.2995	0
1	132.4196	88.541200	0.458500	2.2981	0
2	133.3569	87.576400	0.454300	2.2950	0
3	134.7392	86.036100	0.482700	2.2907	0
4	135.7694	85.039300	0.536100	2.2856	0
…	…	…	…	…	…
29560	126.6800	102.326070	0.506785	2.4993	0
29561	127.9872	102.490830	0.510428	2.4997	0
29562	127.9657	102.498159	0.510361	2.4999	0
29563	128.9050	102.607440	0.512985	2.5000	0

```
29564  128.4784  102.560015  0.511792  2.5000       0

[29483 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      133.4417   87.315400  0.468200  2.2995       0
1      132.4196   88.541200  0.458500  2.2981       0
2      133.3569   87.576400  0.454300  2.2950       0
3      134.7392   86.036100  0.482700  2.2907       0
4      135.7694   85.039300  0.536100  2.2856       0
…         …          …        …     …     …
29560  126.6800  102.326070  0.506785  2.4993       0
29561  127.9872  102.490830  0.510428  2.4997       0
29562  127.9657  102.498159  0.510361  2.4999       0
29563  128.9050  102.607440  0.512985  2.5000       0
29564  128.4784  102.560015  0.511792  2.5000       0

[29483 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      0.331038 -0.131149 -0.514977  0.060167       0
1      0.286901 -0.083306 -0.626728  0.054018       0
2      0.327376 -0.120963 -0.675115  0.040404       0
3      0.387068 -0.181081 -0.347926  0.021520       0
4      0.431556 -0.219986  0.267281 -0.000878       0
…         …        …        …        …     …
29478  0.039046  0.454721 -0.070450  0.937637       0
29479  0.095495  0.461151 -0.028483  0.939394       0
29480  0.094567  0.461437 -0.029246  0.940272       0
29481  0.135129  0.465703  0.000976  0.940711       0
29482  0.116707  0.463852 -0.012766  0.940711       0

[29483 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	0.331038	-0.131149	-0.514977	0.060167
1	0.286901	-0.083306	-0.626728	0.054018
2	0.327376	-0.120963	-0.675115	0.040404
3	0.387068	-0.181081	-0.347926	0.021520
4	0.431556	-0.219986	0.267281	-0.000878
…	…	…	…	…
29478	0.039046	0.454721	-0.070450	0.937637
29479	0.095495	0.461151	-0.028483	0.939394
29480	0.094567	0.461437	-0.029246	0.940272
29481	0.135129	0.465703	0.000976	0.940711
29482	0.116707	0.463852	-0.012766	0.940711

[29483 rows x 4 columns]

[135]:
```
y_testing.describe()
```

[135]:

	FACIES
count	29483.000000
mean	0.586643
std	1.227710
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	4.000000

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

[137]:
```
y_predicted
```

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

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

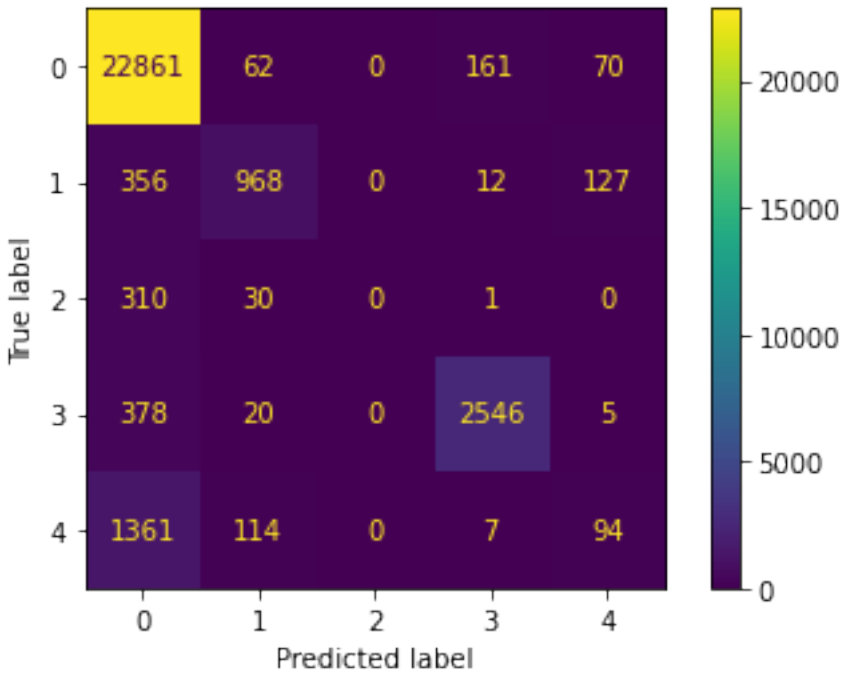
[138]:
```
89.7771597191602
```

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

[139]:
```
array([[22861,    62,     0,   161,    70],
       [  356,   968,     0,    12,   127],
       [  310,    30,     0,     1,     0],
       [  378,    20,     0,  2546,     5],
       [ 1361,   114,     0,     7,    94]])
```

[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
0x7fcaa2edb940>
```



[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)
```

[ ]:
```

```