main_new

August 23, 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
```

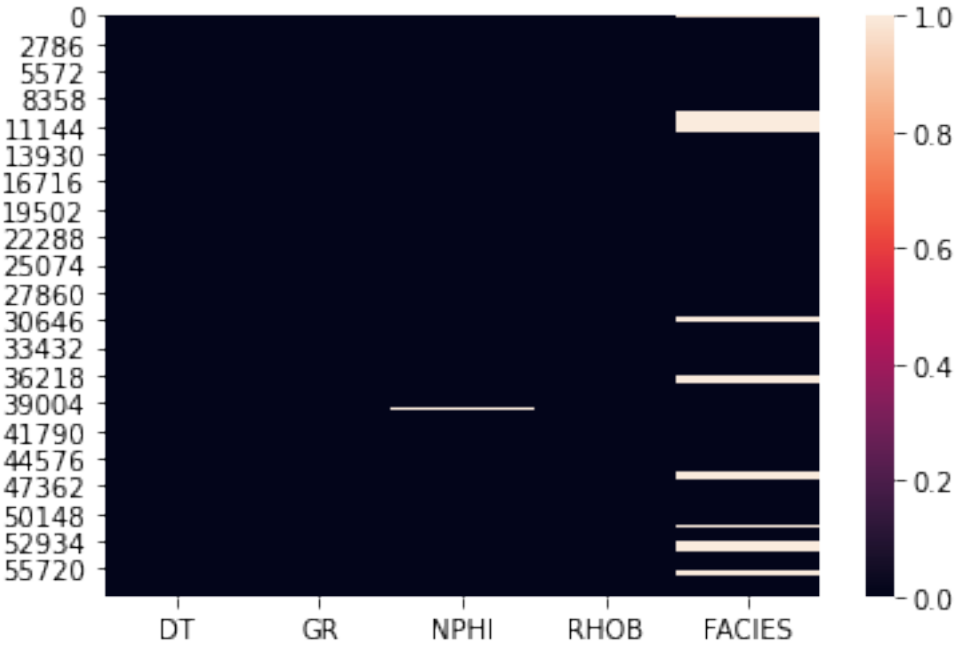
[28]:
```
(58499, 5)
```

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

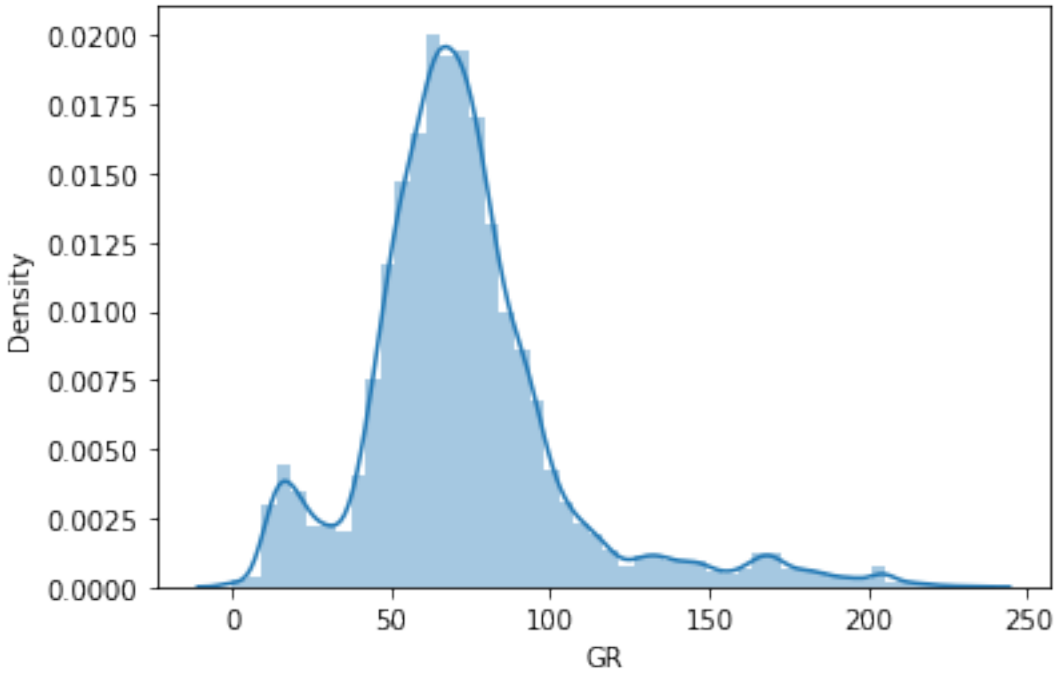
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58499 entries, 0 to 58498
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   DT      58499 non-null  float64
 1   GR      58379 non-null  float64
 2   NPHI    58172 non-null  float64
 3 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



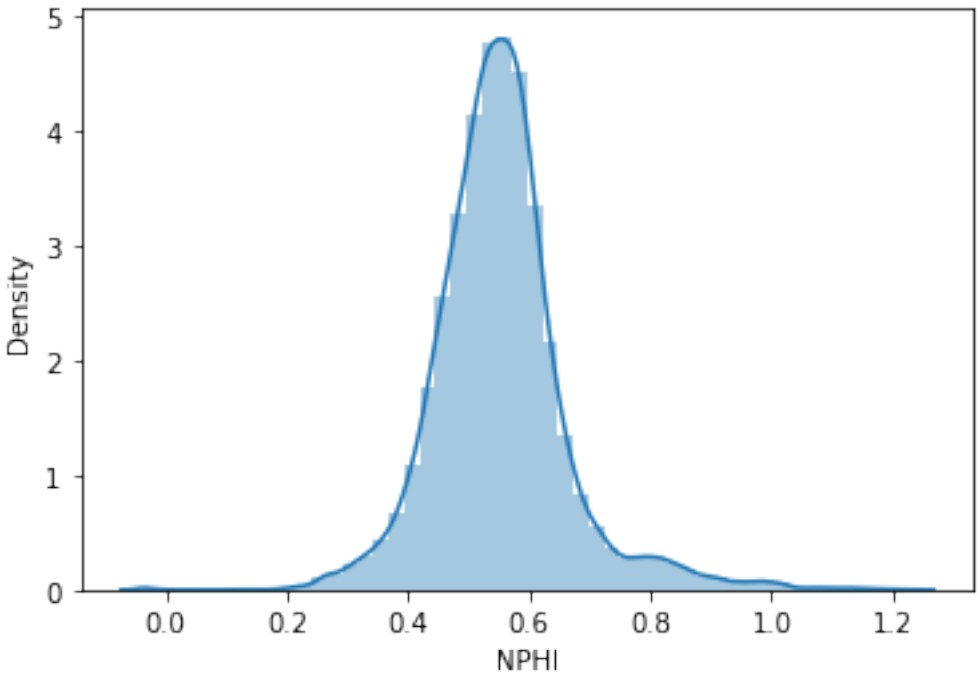
[31]:
```
null_gr = sns.distplot(df.GR.dropna())
```



[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())
```

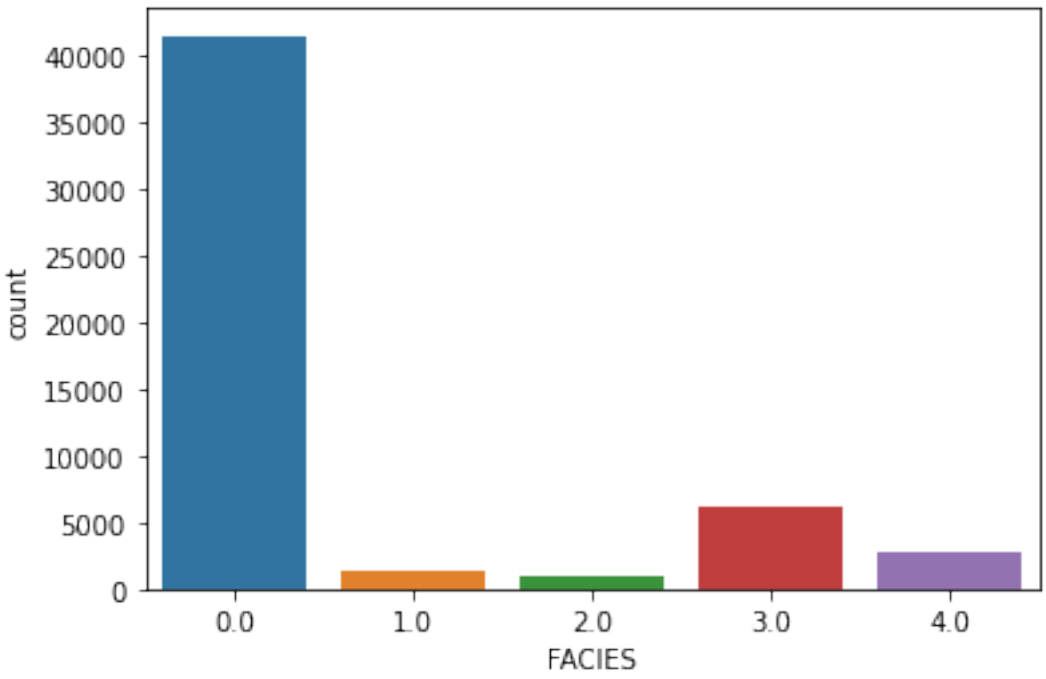


[34]:
```
df.NPHI.describe()
```

[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=4
```

```
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
```

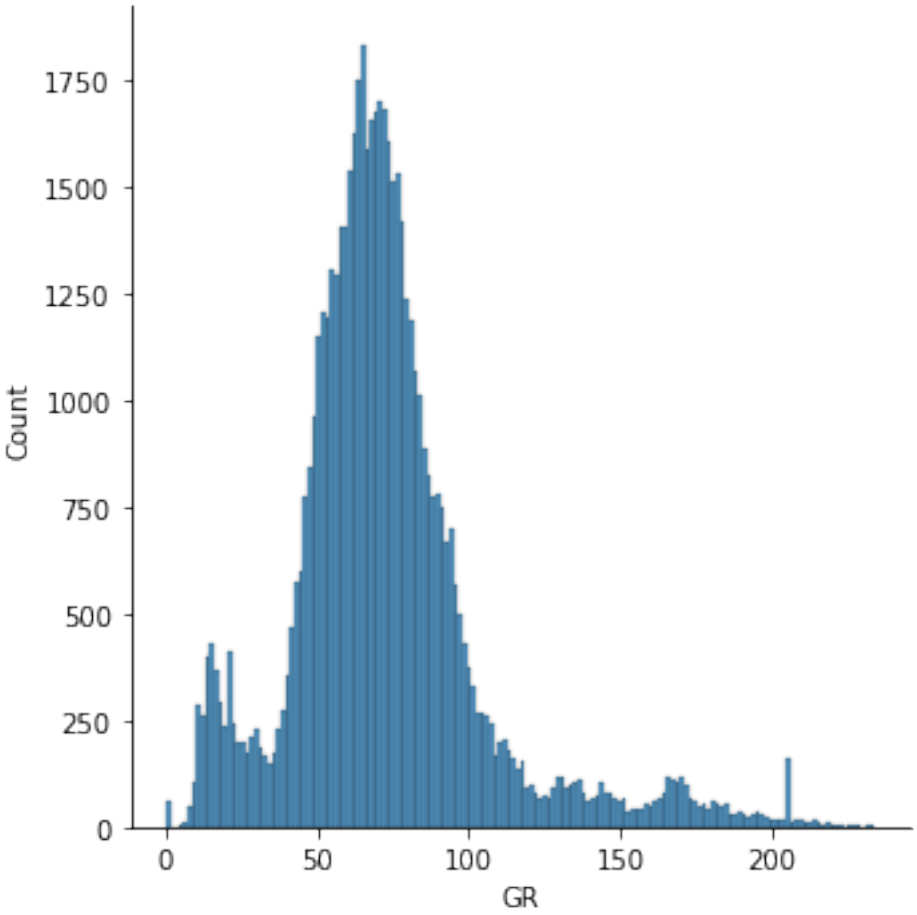
[41]:
```
df.isnull().sum()
```

[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())
```

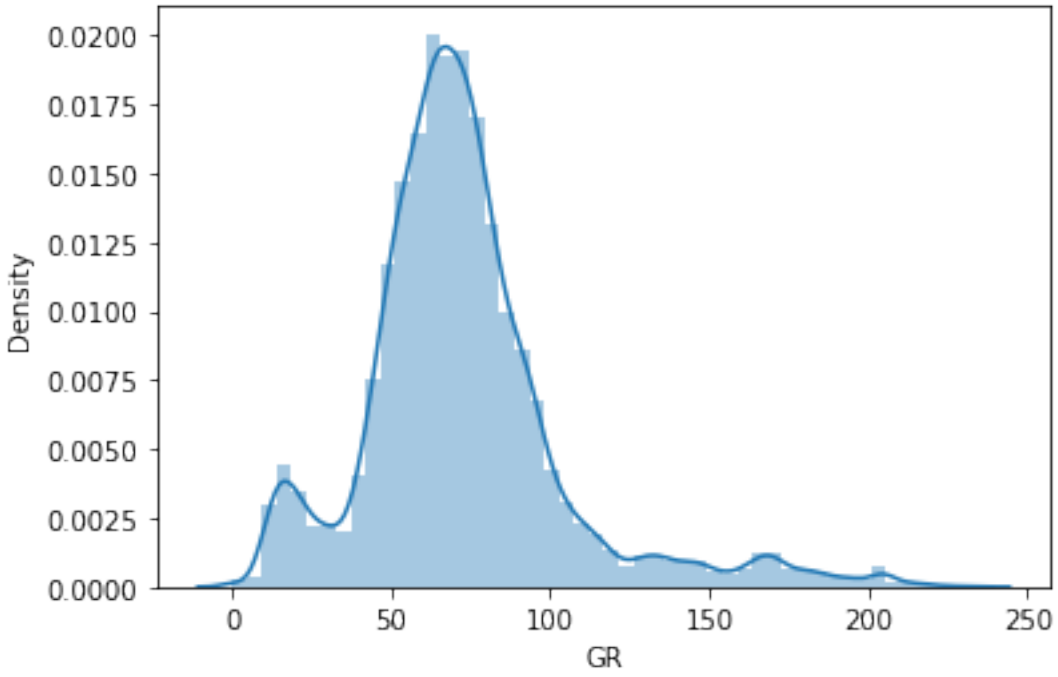
[43]:
```
<seaborn.axisgrid.FacetGrid at 0x7efd5918f370>
```



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

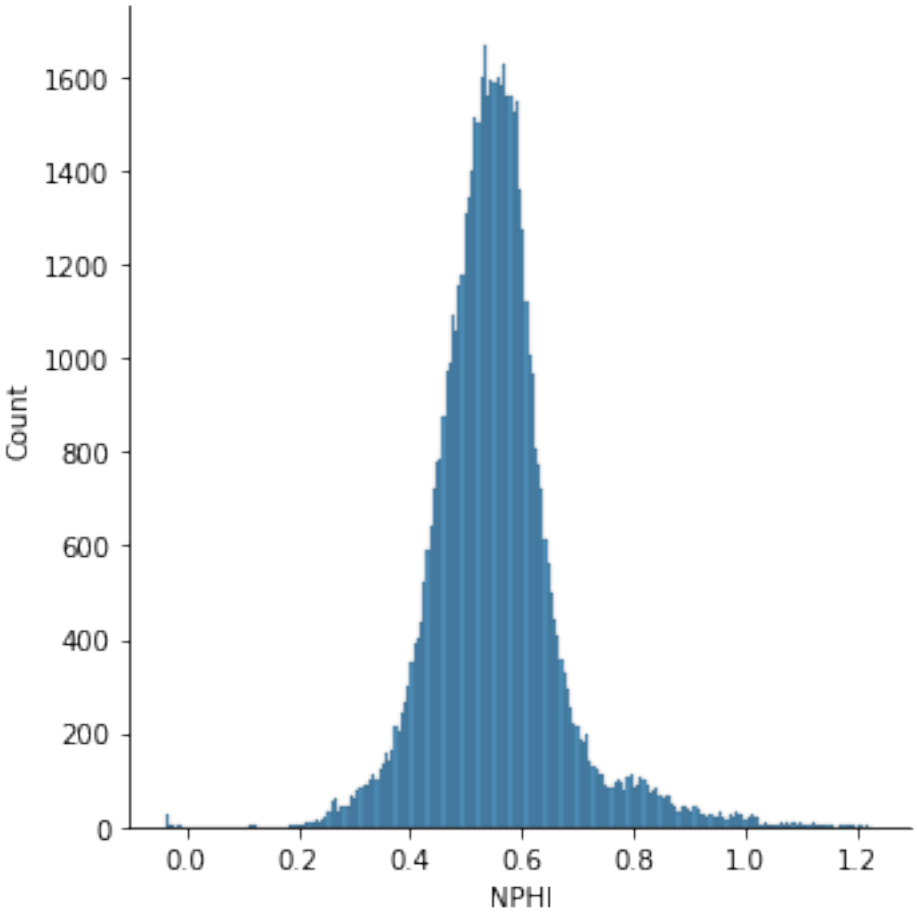
```
WHEN GR WAS NULL
```

[44]:



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

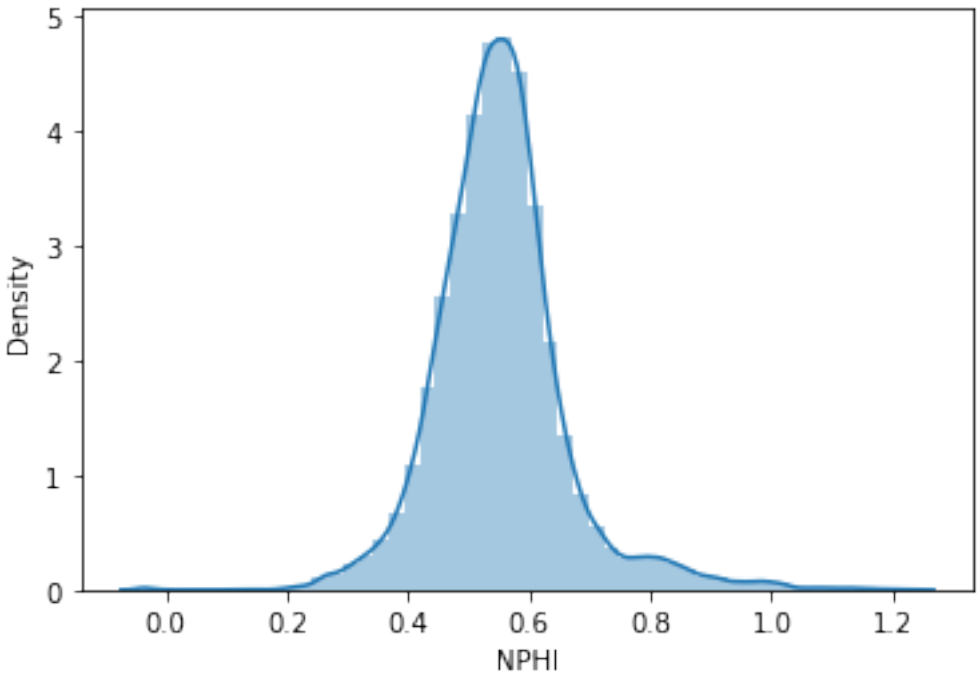
[45]: <seaborn.axisgrid.FacetGrid at 0x7efdb32431f0>



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

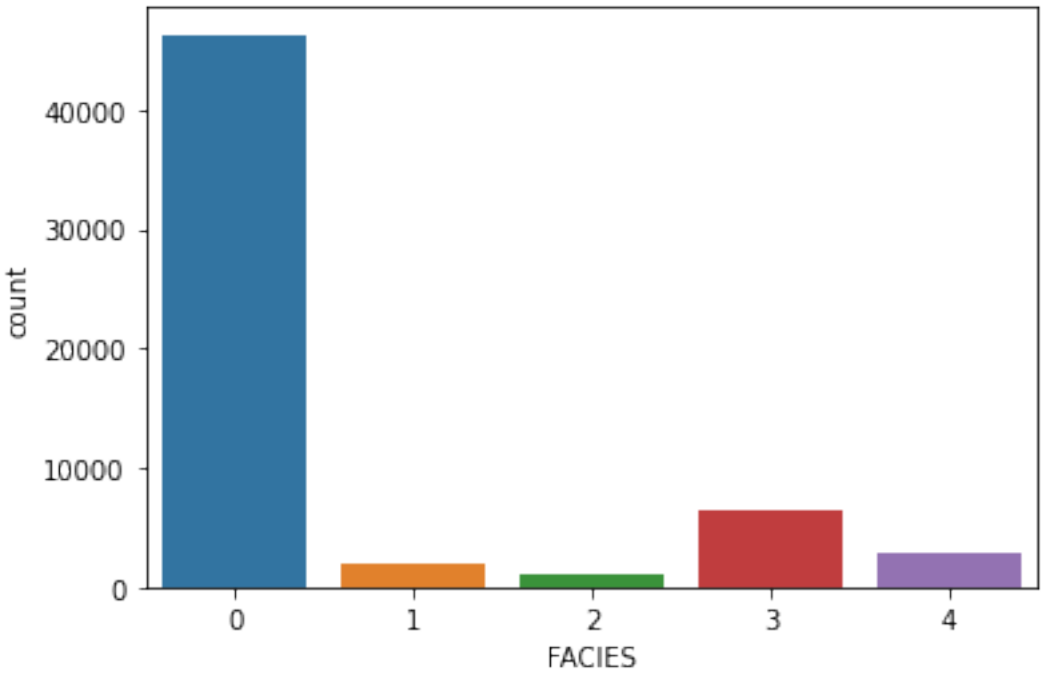
```
WHEN NPHI WAS NULL
```

[46]:



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

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



4 DATA CONDITIONING / OUTLIER REMOVAL

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

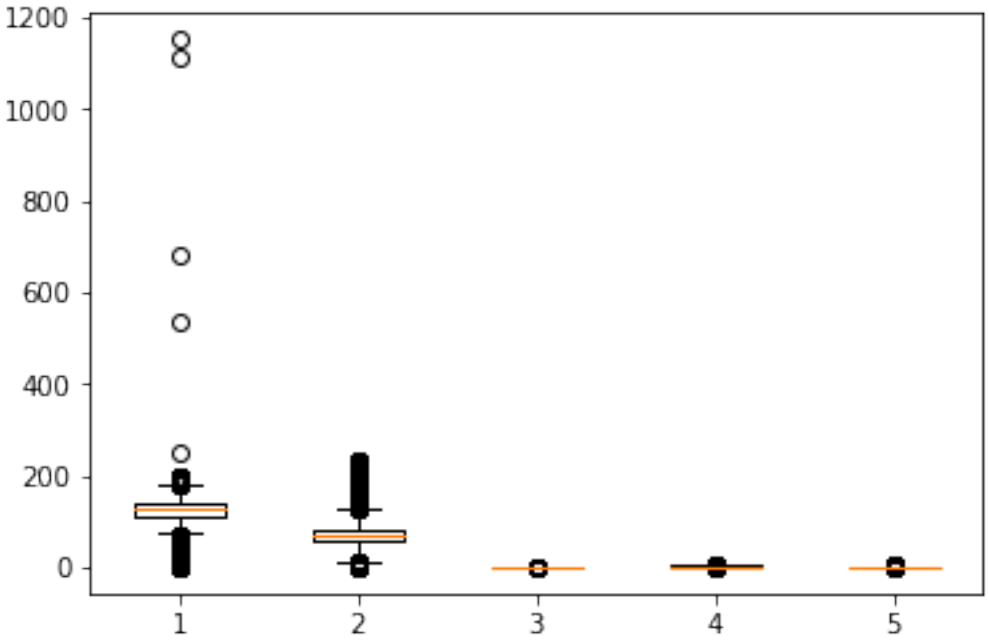
[48]:
```
<bound method NDFrame.head of              DT          GR    NPHI    RHOB
FACIES
0       50.2544   50.212800  0.5340  2.1228       2
1       50.3881   49.750900  0.5316  2.1250       3
2       49.8852   48.251300  0.5126  2.1316       3
3       49.9032   46.821200  0.5137  2.1437       3
4       50.0157   45.346300  0.5472  2.1611       3
…          …          …      …       …     …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  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 0x7efd58d4bac0>,
  <matplotlib.lines.Line2D at 0x7efd58d4be50>,
  <matplotlib.lines.Line2D at 0x7efd58ce6490>,
  <matplotlib.lines.Line2D at 0x7efd58ce6820>,
  <matplotlib.lines.Line2D at 0x7efd58cf2dc0>,
  <matplotlib.lines.Line2D at 0x7efd58cfc190>,
  <matplotlib.lines.Line2D at 0x7efd58d07730>,
  <matplotlib.lines.Line2D at 0x7efd58d07ac0>,
  <matplotlib.lines.Line2D at 0x7efd58c5e0a0>,
  <matplotlib.lines.Line2D at 0x7efd58c5e430>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd58cdb220>,
  <matplotlib.lines.Line2D at 0x7efd58cdb5b0>,
  <matplotlib.lines.Line2D at 0x7efd58ce6bb0>,
  <matplotlib.lines.Line2D at 0x7efd58ce6f40>,
  <matplotlib.lines.Line2D at 0x7efd58cfc520>,
  <matplotlib.lines.Line2D at 0x7efd58cfc8b0>,
  <matplotlib.lines.Line2D at 0x7efd58d07e50>,
  <matplotlib.lines.Line2D at 0x7efd58c51220>,
  <matplotlib.lines.Line2D at 0x7efd58c5e7c0>,
  <matplotlib.lines.Line2D at 0x7efd58c5eb50>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd58d4b730>,
```

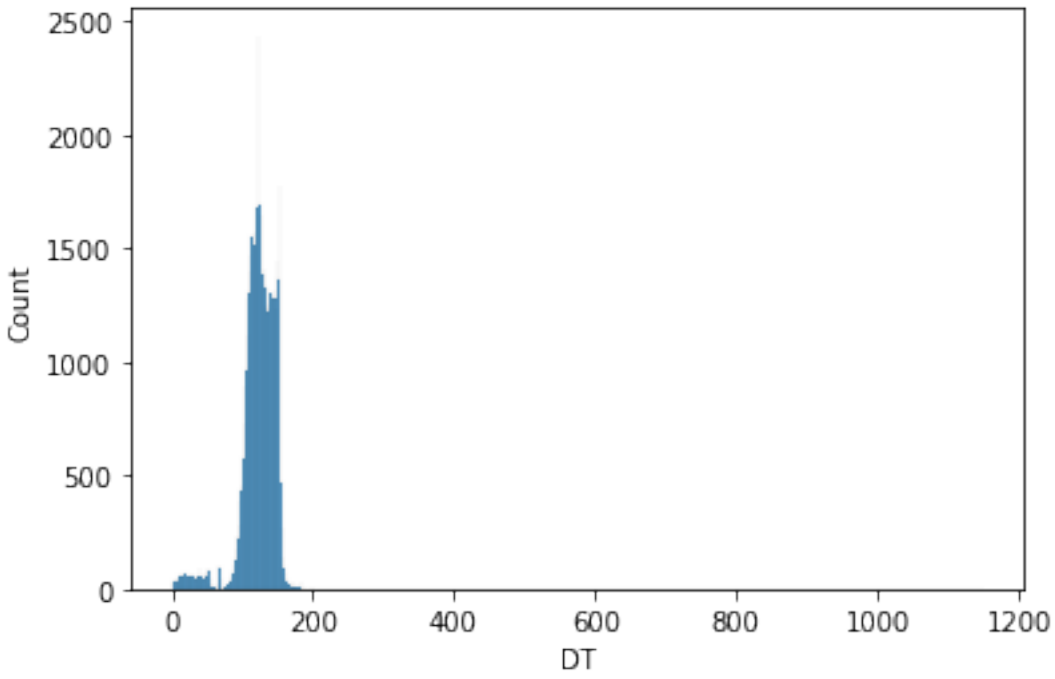
```
 <matplotlib.lines.Line2D at 0x7efd58ce6100>,
 <matplotlib.lines.Line2D at 0x7efd58cf2a30>,
 <matplotlib.lines.Line2D at 0x7efd58d073a0>,
 <matplotlib.lines.Line2D at 0x7efd58c51cd0>],
'medians': [<matplotlib.lines.Line2D at 0x7efd58cdb940>,
 <matplotlib.lines.Line2D at 0x7efd58cf2310>,
 <matplotlib.lines.Line2D at 0x7efd58cfcc40>,
 <matplotlib.lines.Line2D at 0x7efd58c515b0>,
 <matplotlib.lines.Line2D at 0x7efd58c5eee0>],
'fliers': [<matplotlib.lines.Line2D at 0x7efd58cdbcd0>,
 <matplotlib.lines.Line2D at 0x7efd58cf26a0>,
 <matplotlib.lines.Line2D at 0x7efd58cfcfd0>,
 <matplotlib.lines.Line2D at 0x7efd58c51940>,
 <matplotlib.lines.Line2D at 0x7efd58c682e0>],
'means': []}
```



4.2 DT VISUALIZATION

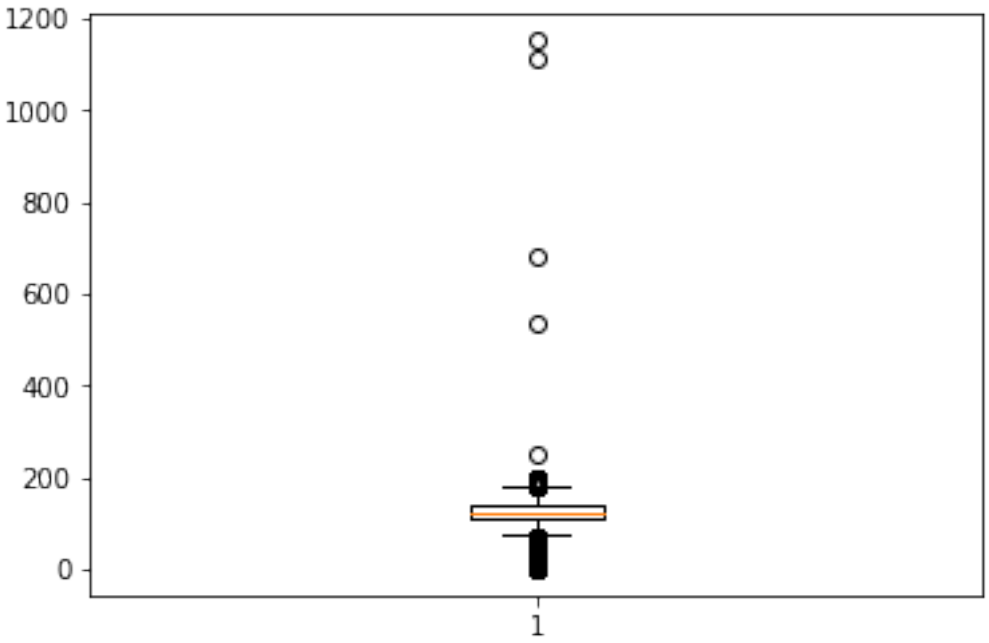
[50]:
```
sns.histplot(df.DT)
```

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



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

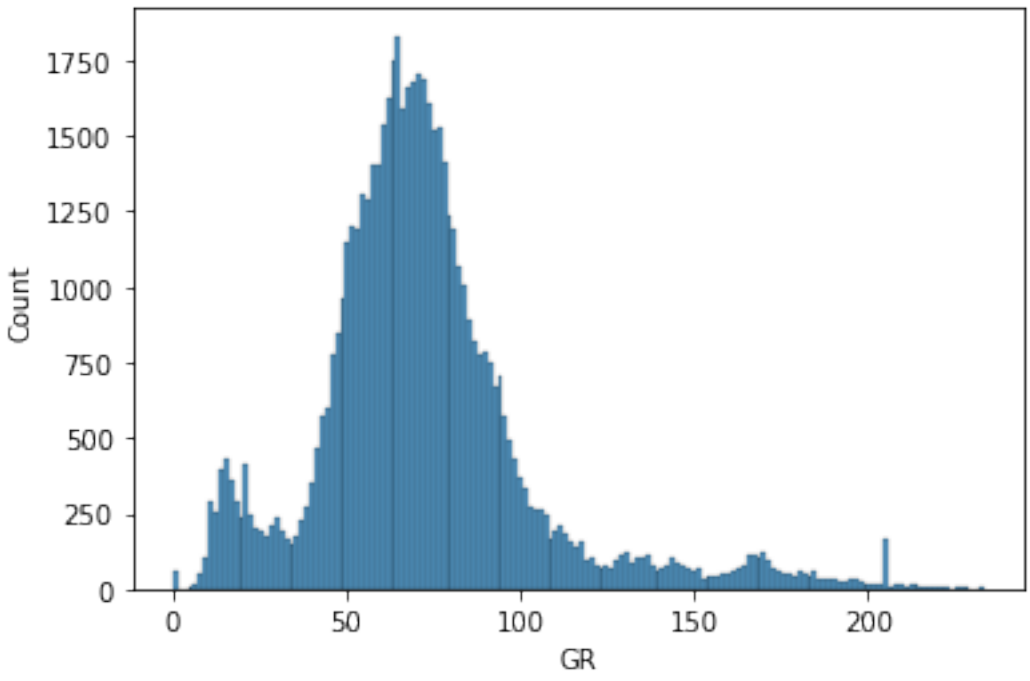
[51]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd57413ac0>,
  <matplotlib.lines.Line2D at 0x7efd57413e50>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd5741e220>,
  <matplotlib.lines.Line2D at 0x7efd5741e5b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd57413730>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd5741e940>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd5741ecd0>],
 'means': []}
```



4.3 GR VISUALIZATION

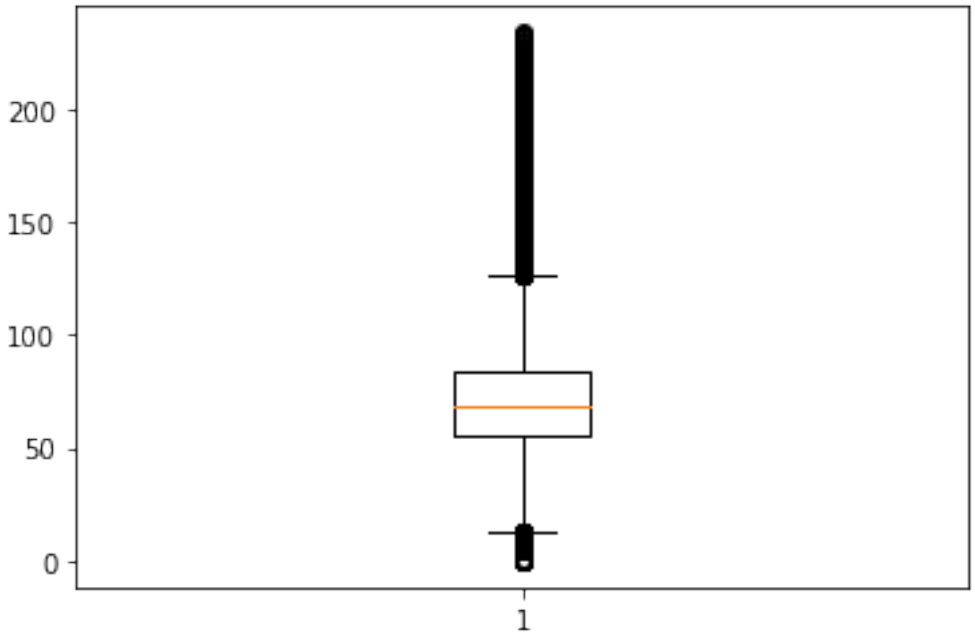
[52]:
```
sns.histplot(df.GR)
```

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



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

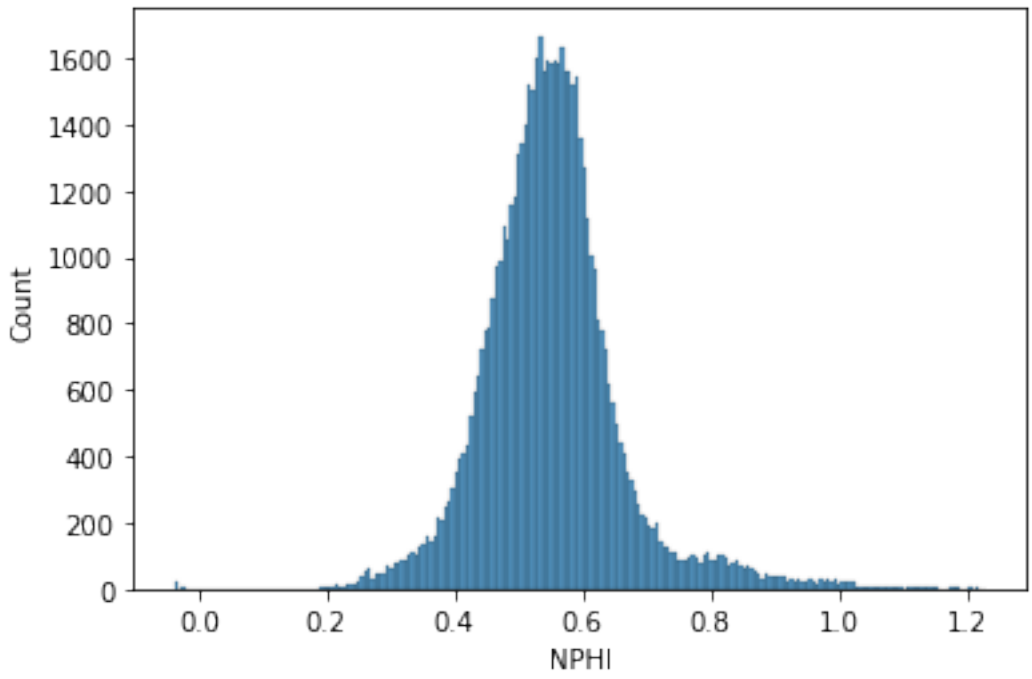
[53]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd50d9cd90>,
  <matplotlib.lines.Line2D at 0x7efd50dab160>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd50dab4f0>,
  <matplotlib.lines.Line2D at 0x7efd50dab880>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd50d9ca00>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd50dabc10>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd50dabfa0>],
 'means': []}
```



4.4 NPHI VISUALIZATION

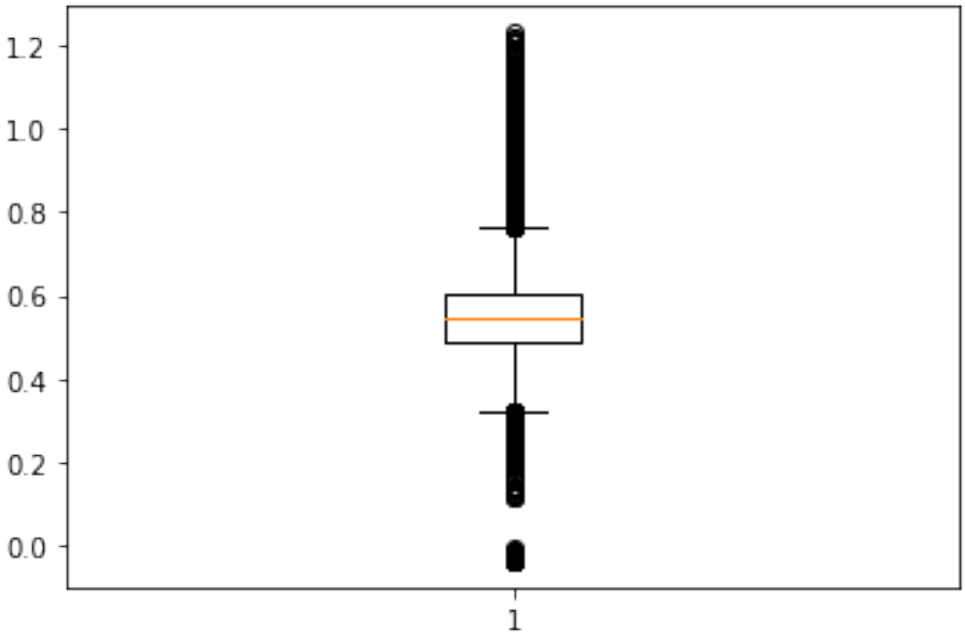
[54]:
```
sns.histplot(df.NPHI)
```

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



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

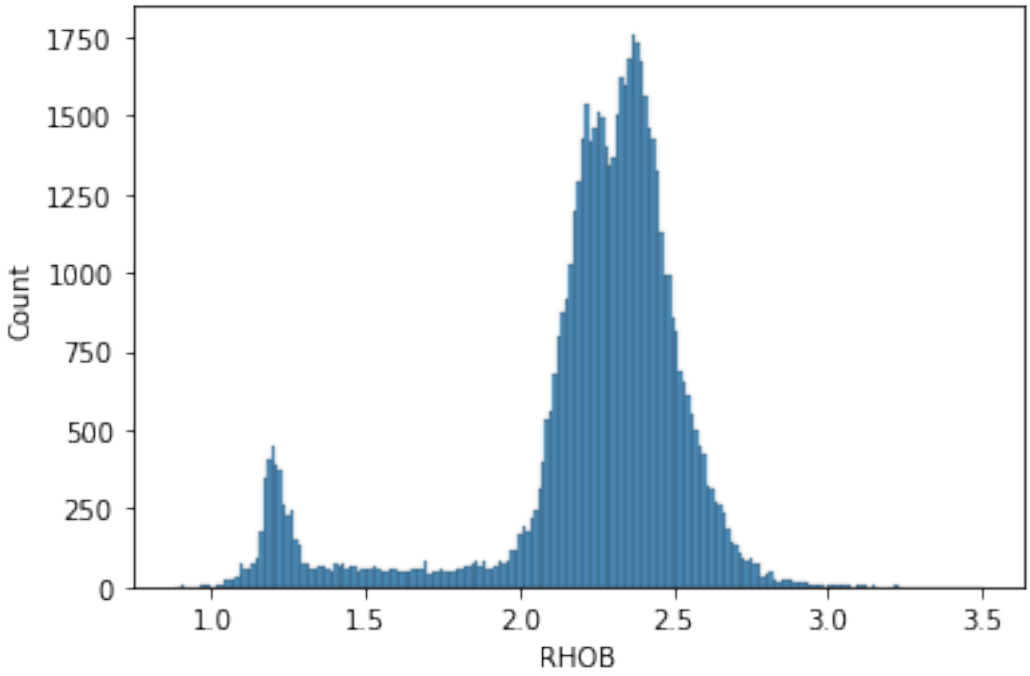
[55]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd50afecd0>,
  <matplotlib.lines.Line2D at 0x7efd50a8c0a0>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd50a8c430>,
  <matplotlib.lines.Line2D at 0x7efd50a8c7c0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd50afe940>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd50a8cb50>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd50a8cee0>],
 'means': []}
```



4.5 RHOB VISUALIZATION

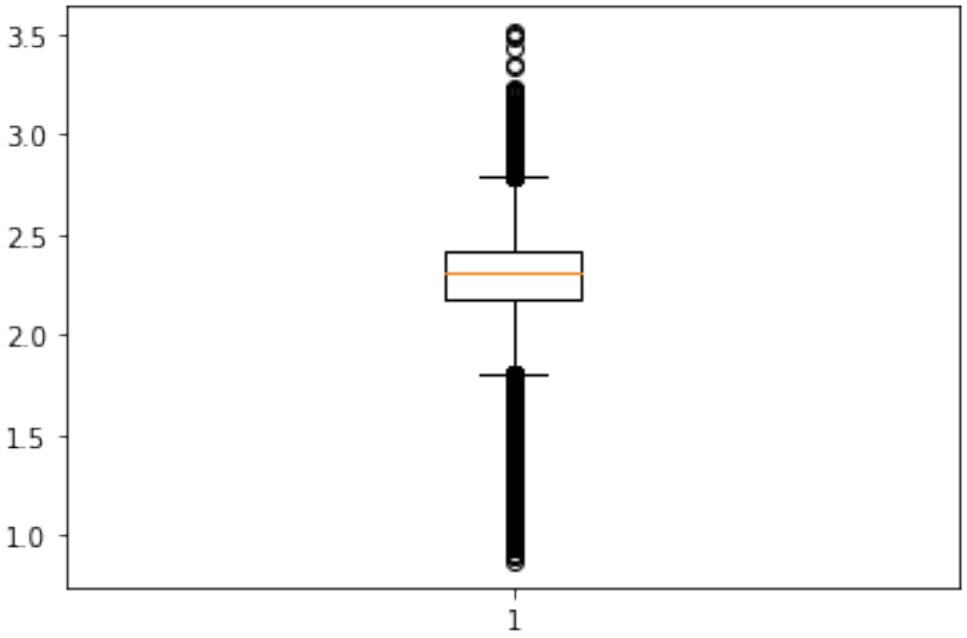
[56]:
```
sns.histplot(df.RHOB)
```

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



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

[57]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd507c9c40>,
  <matplotlib.lines.Line2D at 0x7efd507c9fd0>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd507d63a0>,
  <matplotlib.lines.Line2D at 0x7efd507d6730>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd507c98b0>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd507d6ac0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd507d6e50>],
 'means': []}
```



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

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

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

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

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

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

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


    return df
```

[59]:
```
DATAConditioningStrategy =␣
↪["3_Standard_Deviation","4_Standard_Deviation","InterquartileRange"]
DATAConditioningColumns = ["DT","GR","NPHI","RHOB"]
optionoutlier = 1
df = outliers(DATAConditioningStrategy[optionoutlier] , df,␣
↪DATAConditioningColumns)
```

```
column DT
4 standard deviation outliers -:
              DT       GR    NPHI    RHOB  FACIES
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       2
1       50.3881   49.750900  0.5316  2.1250       3
2       49.8852   48.251300  0.5126  2.1316       3
3       49.9032   46.821200  0.5137  2.1437       3
4       50.0157   45.346300  0.5472  2.1611       3
…          …         …       …       …       …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  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
…         …       …       …       …       …
39777  125.9000  204.7348  0.590867  2.4427       0
39778  125.9000  204.7348  0.590867  2.4315       0
39779  125.9000  204.7348  0.590867  2.4286       0
39780  125.9000  204.7348  0.590867  2.4307       0
39781  125.9000  204.7348  0.600667  2.4320       0

[344 rows x 5 columns]
(344, 5)
             DT          GR    NPHI    RHOB  FACIES
0       50.2544   50.212800  0.5340  2.1228       2
1       50.3881   49.750900  0.5316  2.1250       3
2       49.8852   48.251300  0.5126  2.1316       3
```

```
3       49.9032   46.821200  0.5137  2.1437       3
4       50.0157   45.346300  0.5472  2.1611       3
…         …         …      …       …      …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  0.5331  2.4709       0

[57470 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       2
1       50.3881   49.750900  0.5316  2.1250       3
2       49.8852   48.251300  0.5126  2.1316       3
3       49.9032   46.821200  0.5137  2.1437       3
4       50.0157   45.346300  0.5472  2.1611       3
…         …         …      …       …      …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  0.5331  2.4709       0

[57140 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       2
1       50.3881   49.750900  0.5316  2.1250       3
2       49.8852   48.251300  0.5126  2.1316       3
3       49.9032   46.821200  0.5137  2.1437       3
4       50.0157   45.346300  0.5472  2.1611       3
…          …         …       …       …     …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  0.5331  2.4709       0

[57140 rows x 5 columns]
```

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

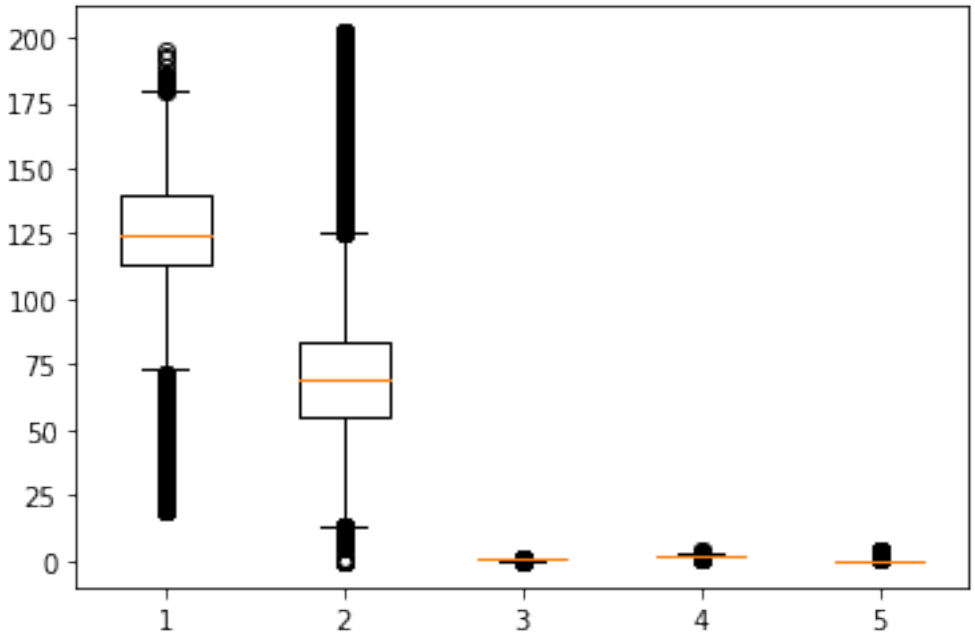
[60]:
```
(57140, 5)
```

4.6 WHOLE DATA AFTER REMOVING OUTLIERS

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

[61]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd507be520>,
  <matplotlib.lines.Line2D at 0x7efd507be8b0>,
  <matplotlib.lines.Line2D at 0x7efd5038be80>,
  <matplotlib.lines.Line2D at 0x7efd50394250>,
  <matplotlib.lines.Line2D at 0x7efd503a2820>,
  <matplotlib.lines.Line2D at 0x7efd503a2bb0>,
  <matplotlib.lines.Line2D at 0x7efd503b8190>,
  <matplotlib.lines.Line2D at 0x7efd503b8520>,
  <matplotlib.lines.Line2D at 0x7efd503c1ac0>,
  <matplotlib.lines.Line2D at 0x7efd503c1e50>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd507bec70>,
  <matplotlib.lines.Line2D at 0x7efd5038b040>,
  <matplotlib.lines.Line2D at 0x7efd503945e0>,
  <matplotlib.lines.Line2D at 0x7efd50394970>,
  <matplotlib.lines.Line2D at 0x7efd503a2f40>,
  <matplotlib.lines.Line2D at 0x7efd503ad310>,
  <matplotlib.lines.Line2D at 0x7efd503b88b0>,
  <matplotlib.lines.Line2D at 0x7efd503b8c40>,
  <matplotlib.lines.Line2D at 0x7efd5074c220>,
  <matplotlib.lines.Line2D at 0x7efd5074c5b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd507be190>,
  <matplotlib.lines.Line2D at 0x7efd5038baf0>,
  <matplotlib.lines.Line2D at 0x7efd503a2490>,
  <matplotlib.lines.Line2D at 0x7efd503addc0>,
  <matplotlib.lines.Line2D at 0x7efd503c1730>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd5038b3d0>,
```

```
 <matplotlib.lines.Line2D at 0x7efd50394d00>,
 <matplotlib.lines.Line2D at 0x7efd503ad6a0>,
 <matplotlib.lines.Line2D at 0x7efd503b8fd0>,
 <matplotlib.lines.Line2D at 0x7efd5074c940>],
'fliers': [<matplotlib.lines.Line2D at 0x7efd5038b760>,
 <matplotlib.lines.Line2D at 0x7efd503a2100>,
 <matplotlib.lines.Line2D at 0x7efd503ada30>,
 <matplotlib.lines.Line2D at 0x7efd503c13a0>,
 <matplotlib.lines.Line2D at 0x7efd5074ccd0>],
'means': []}
```



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

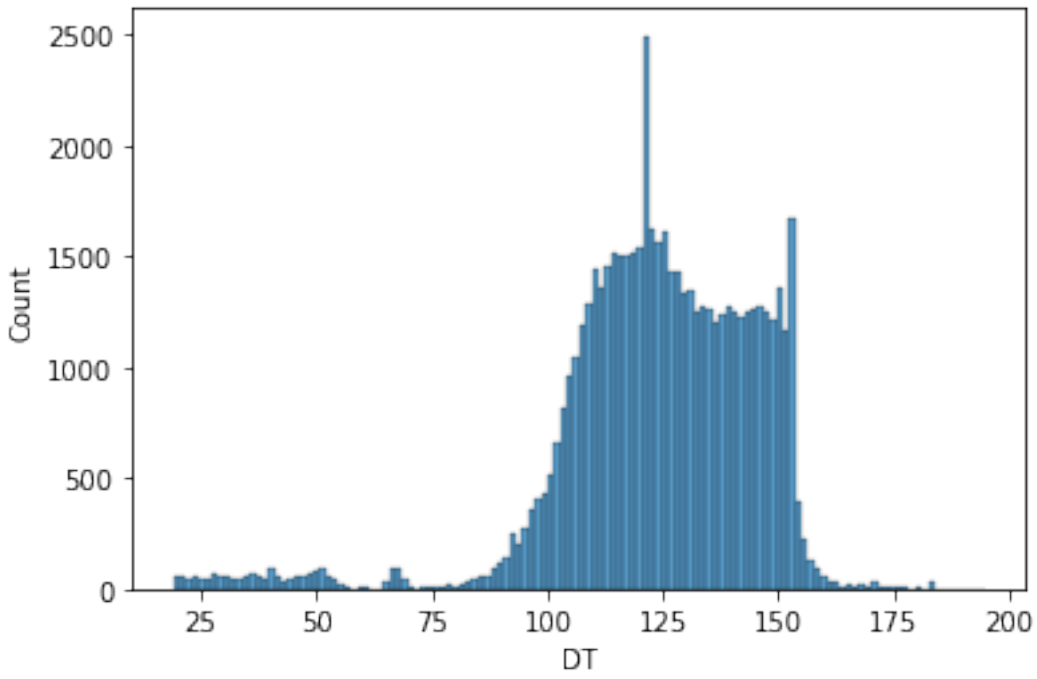
[62]:

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

4.7 DT AFTER REMOVING OUTLIER

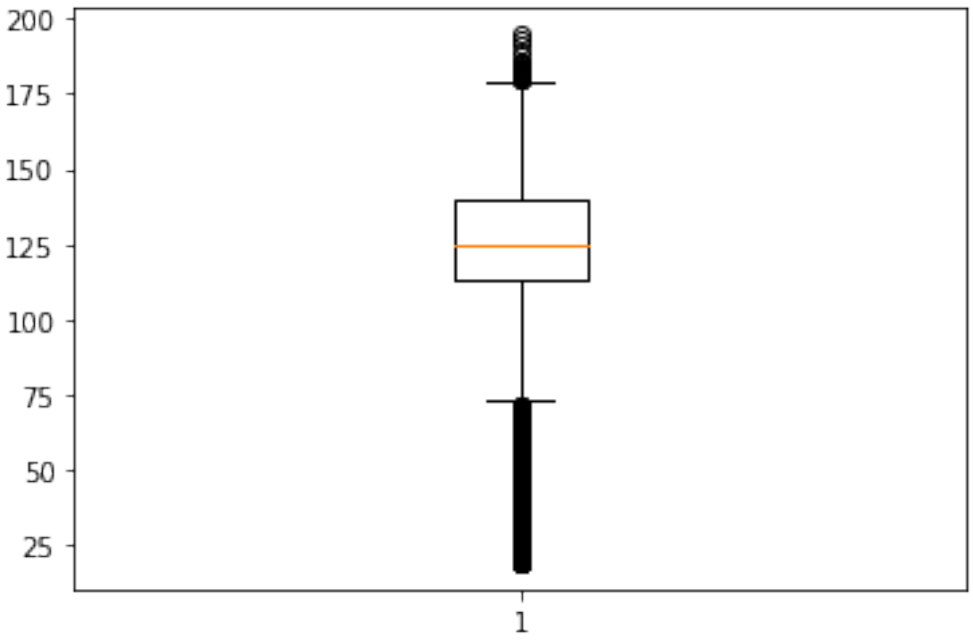
[63]:
```
sns.histplot(df.DT)
```

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



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

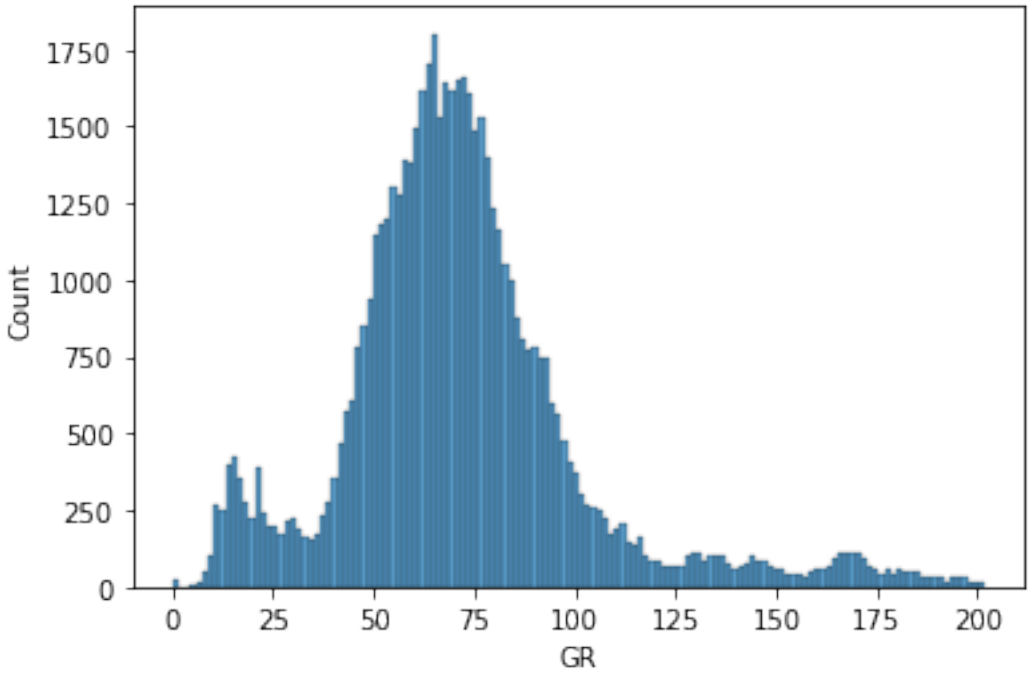
[64]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd50641070>,
  <matplotlib.lines.Line2D at 0x7efd50641400>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd50641790>,
  <matplotlib.lines.Line2D at 0x7efd50641b20>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd50633ca0>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd50641eb0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd505cf280>],
 'means': []}
```



4.8 GR AFTER REMOVING OUTLIER

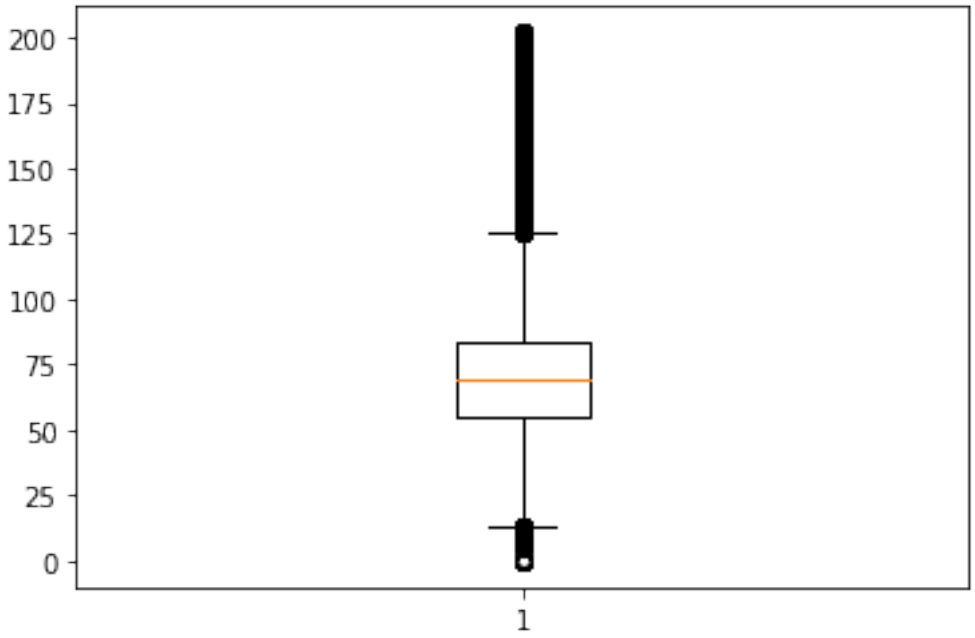
[65]:
```
sns.histplot(df.GR)
```

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



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

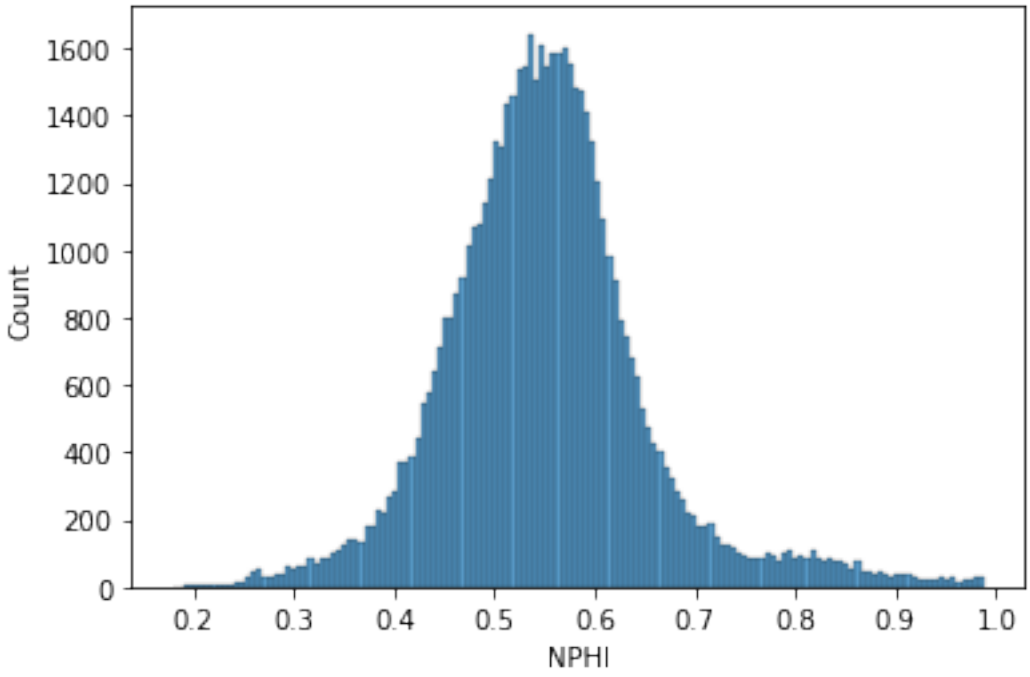
[66]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd503d2b80>,
  <matplotlib.lines.Line2D at 0x7efd503d2f10>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd503dd2e0>,
  <matplotlib.lines.Line2D at 0x7efd503dd670>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd503d27f0>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd503dda00>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd503ddd90>],
 'means': []}
```



4.9 NPHI AFTER REMOVING OUTLIER

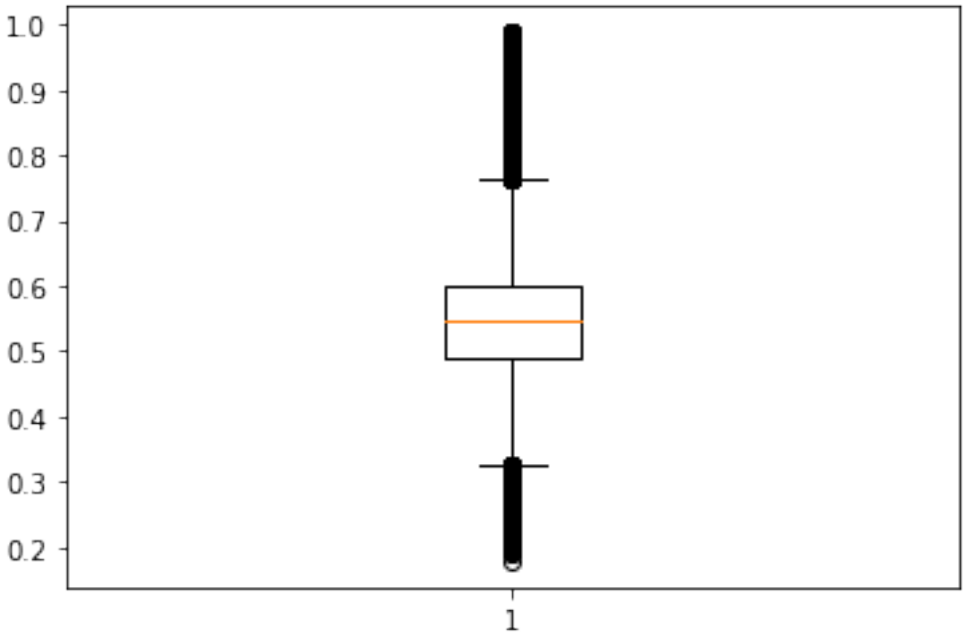
[67]:
```
sns.histplot(df.NPHI)
```

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



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

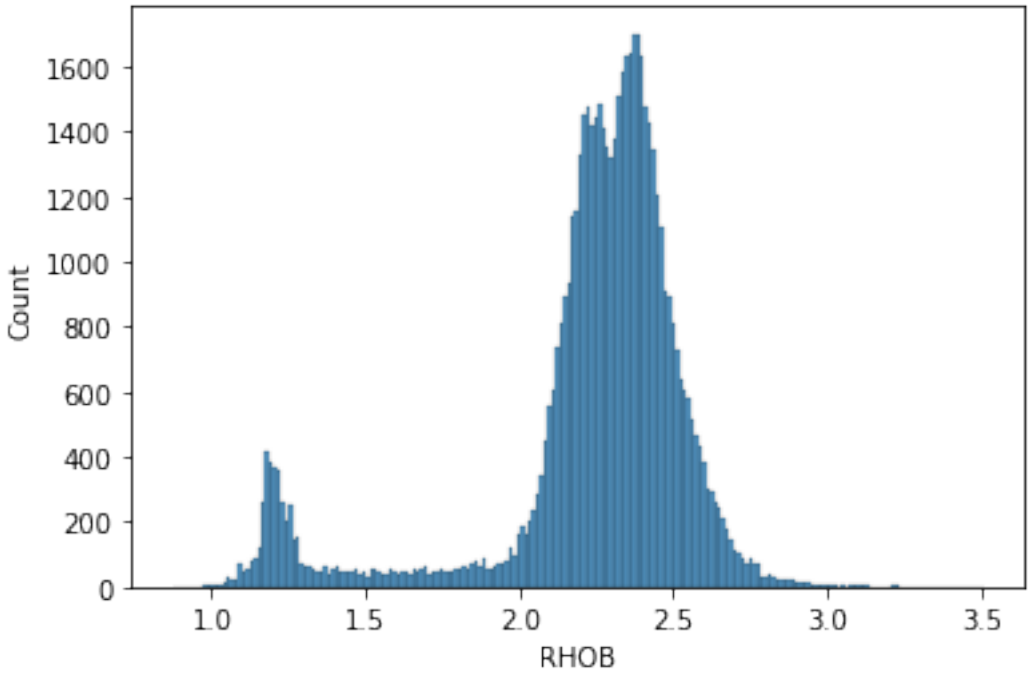
[68]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd501cb8e0>,
  <matplotlib.lines.Line2D at 0x7efd501cbc70>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd501d7040>,
  <matplotlib.lines.Line2D at 0x7efd501d73d0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd501cb550>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd501d7760>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd501d7af0>],
 'means': []}
```



4.10 RHOB AFTER REMOVING OUTLIER

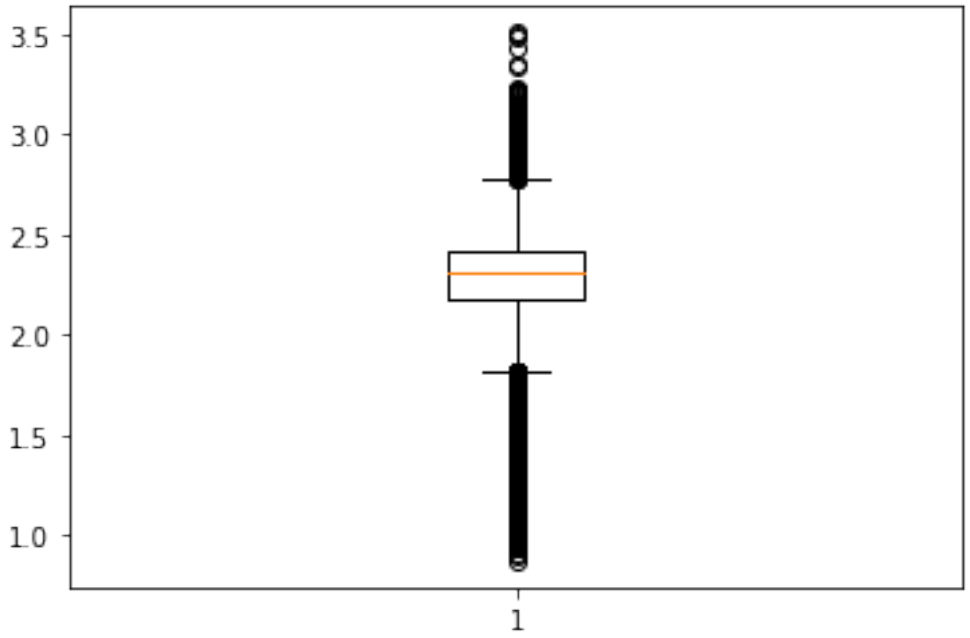
[69]:
```
sns.histplot(df.RHOB)
```

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



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

[70]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7efd4feee070>,
  <matplotlib.lines.Line2D at 0x7efd4feee400>],
 'caps': [<matplotlib.lines.Line2D at 0x7efd4feee790>,
  <matplotlib.lines.Line2D at 0x7efd4feeeb20>],
 'boxes': [<matplotlib.lines.Line2D at 0x7efd4fedfca0>],
 'medians': [<matplotlib.lines.Line2D at 0x7efd4feeeeb0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7efd4fef9280>],
 'means': []}
```



[71]:
```
df
```

[71]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	2
1	50.3881	49.750900	0.5316	2.1250	3
2	49.8852	48.251300	0.5126	2.1316	3
3	49.9032	46.821200	0.5137	2.1437	3
4	50.0157	45.346300	0.5472	2.1611	3
…	…	…	…	…	…
58494	123.7404	130.872833	0.4993	2.4639	0
58495	123.8728	92.579667	0.5313	2.4660	0
58496	123.3722	81.624267	0.5448	2.4714	0

```
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  0.5331  2.4709       0

[57140 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       2
1  50.3881  49.7509  0.5316  2.1250       3
2  49.8852  48.2513  0.5126  2.1316       3
3  49.9032  46.8212  0.5137  2.1437       3
4  50.0157  45.3463  0.5472  2.1611       3
5  50.6831  44.0819  0.5550  2.1740       3
6  51.4311  43.6654  0.5612  2.1707       3
7  52.1678  43.3915  0.5566  2.1595       3
8  52.2883  44.1249  0.5390  2.1534       3
9  51.5991  46.1805  0.5245  2.1551       3
```

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

[73]:
```
(57140, 5)
```

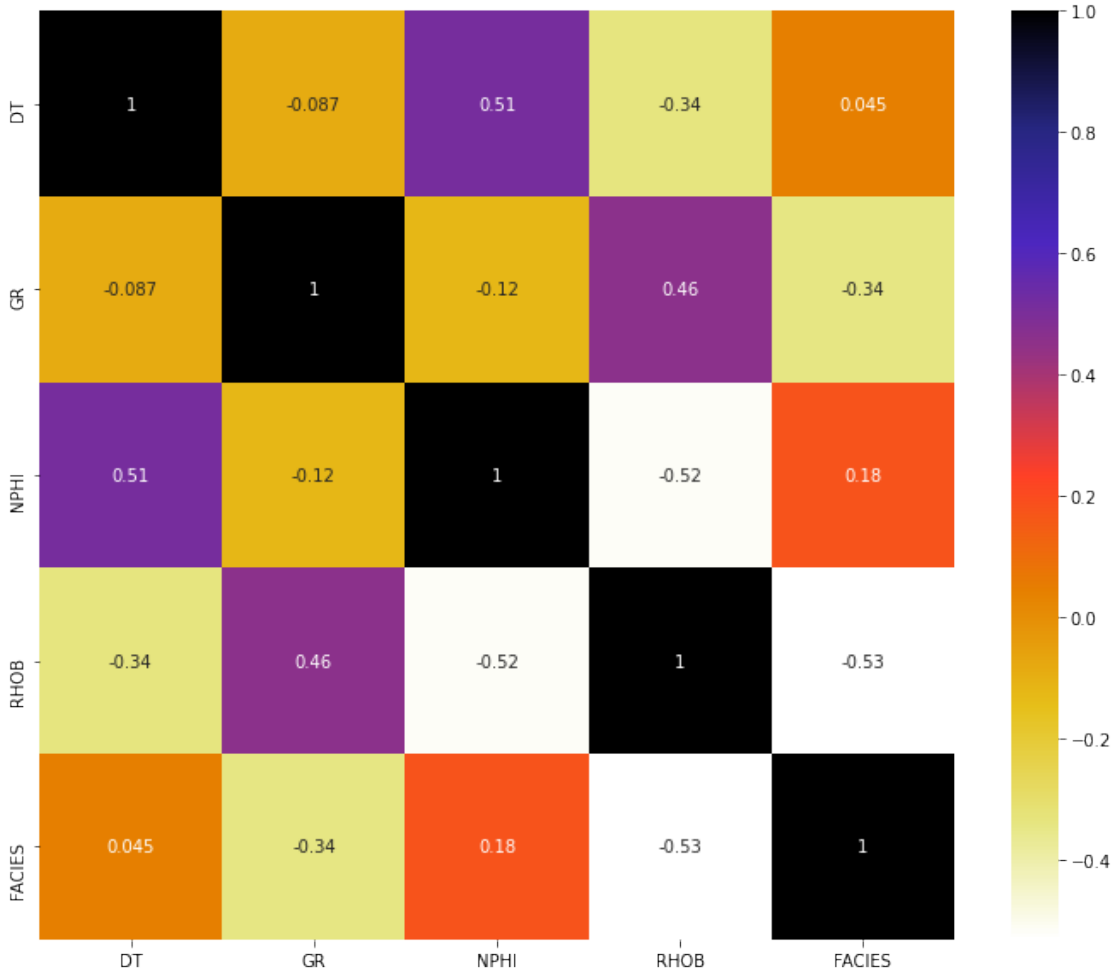
[74]:
```
features = df.shape[1]
features
```

[74]:
```
5
```

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

[75]:
```
DT        489.540543
GR        926.896049
NPHI        0.010355
RHOB        0.116723
FACIES      1.479887
dtype: float64
```

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



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

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

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

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

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

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

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

[78]:
```
FeatureSelectionStrategy=["Variance_Threshold","Absolute_Correlation","Correlation","SelectKBe
optionfeature = 2
df=FeatureSelection(FeatureSelectionStrategy[optionfeature],df)
```

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

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

[80]:
```
df
```

[80]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	2
1	50.3881	49.750900	0.5316	2.1250	3
2	49.8852	48.251300	0.5126	2.1316	3
3	49.9032	46.821200	0.5137	2.1437	3
4	50.0157	45.346300	0.5472	2.1611	3
…	…	…	…	…	…
58494	123.7404	130.872833	0.4993	2.4639	0
58495	123.8728	92.579667	0.5313	2.4660	0
58496	123.3722	81.624267	0.5448	2.4714	0
58497	122.6038	118.991767	0.5364	2.4750	0
58498	122.3045	70.033400	0.5331	2.4709	0

[57140 rows x 5 columns]

[81]:
```
df
```

[81]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	2
1	50.3881	49.750900	0.5316	2.1250	3
2	49.8852	48.251300	0.5126	2.1316	3
3	49.9032	46.821200	0.5137	2.1437	3
4	50.0157	45.346300	0.5472	2.1611	3
…	…	…	…	…	…
58494	123.7404	130.872833	0.4993	2.4639	0
58495	123.8728	92.579667	0.5313	2.4660	0
58496	123.3722	81.624267	0.5448	2.4714	0
58497	122.6038	118.991767	0.5364	2.4750	0
58498	122.3045	70.033400	0.5331	2.4709	0

[57140 rows x 5 columns]

6 SCALING DATA

[82]:
```
df
```

[82]:

	DT	GR	NPHI	RHOB	FACIES
0	50.2544	50.212800	0.5340	2.1228	2
1	50.3881	49.750900	0.5316	2.1250	3
2	49.8852	48.251300	0.5126	2.1316	3
3	49.9032	46.821200	0.5137	2.1437	3

```
4       50.0157   45.346300  0.5472  2.1611       3
…          …         …      …       …     …
58494  123.7404  130.872833  0.4993  2.4639       0
58495  123.8728   92.579667  0.5313  2.4660       0
58496  123.3722   81.624267  0.5448  2.4714       0
58497  122.6038  118.991767  0.5364  2.4750       0
58498  122.3045   70.033400  0.5331  2.4709       0

[57140 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     -2.803623 -0.661857 -0.120266 -0.763591       2
1     -2.798605 -0.678247 -0.142300 -0.754530       3
2     -2.817481 -0.731461 -0.316732 -0.727348       3
3     -2.816805 -0.782208 -0.306633 -0.677512       3
4     -2.812583 -0.834545  0.000918 -0.605848       3
…          …       …        …        …     …
58494 -0.045375  2.200369 -0.438834  0.641269       0
58495 -0.040406  0.841534 -0.145054  0.649918       0
58496 -0.059195  0.452781 -0.021115  0.672158       0
58497 -0.088037  1.778769 -0.098233  0.686985       0
58498 -0.099271  0.041478 -0.128529  0.670099       0

[57140 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	-2.803623	-0.661857	-0.120266	-0.763591	2
1	-2.798605	-0.678247	-0.142300	-0.754530	3
2	-2.817481	-0.731461	-0.316732	-0.727348	3
3	-2.816805	-0.782208	-0.306633	-0.677512	3
4	-2.812583	-0.834545	0.000918	-0.605848	3

[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]:
```
(39998, 4)
```

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

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

[93]:
```
X_test
```

[93]:

	DT	GR	NPHI	RHOB
42651	-0.575488	-0.004989	-1.307322	0.233526
54806	0.528468	-0.145677	-0.432408	-0.069193
36097	1.860476	2.558320	1.036493	-0.472817
31424	-0.045293	0.129513	0.016525	0.049012
26096	-0.840206	0.323031	-0.115676	1.163509
…	…	…	…	…
41755	-0.113703	0.408649	0.379160	0.123970

```
13010  0.716343 -0.412425  0.526050 -0.888797
52553  0.475027  1.503754 -0.245123 -0.623147
39316 -0.768103  1.141689 -0.642644  1.008237
18267 -0.214370 -0.162568 -0.022952  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: 0.877660 using {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.877660 (0.002631) with: {'C': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
0.877660 (0.002631) with: {'C': 100, 'penalty': 'l2', 'solver': 'lbfgs'}
0.875410 (0.002719) with: {'C': 100, 'penalty': 'l2', 'solver': 'liblinear'}
0.877660 (0.002631) with: {'C': 10, 'penalty': 'l2', 'solver': 'newton-cg'}
0.877660 (0.002631) with: {'C': 10, 'penalty': 'l2', 'solver': 'lbfgs'}
0.875410 (0.002719) with: {'C': 10, 'penalty': 'l2', 'solver': 'liblinear'}
0.877610 (0.002643) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'newton-cg'}
0.877619 (0.002643) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'lbfgs'}
0.875410 (0.002719) with: {'C': 1.0, 'penalty': 'l2', 'solver': 'liblinear'}
0.877519 (0.002627) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'newton-cg'}
0.877519 (0.002627) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
0.875210 (0.002672) with: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}
0.876160 (0.002564) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
0.876160 (0.002564) with: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
0.873977 (0.002670) 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: 1.3379866        total: 54ms     remaining: 54s
1:      learn: 1.1629468        total: 61.2ms   remaining: 30.5s
2:      learn: 1.0319283        total: 68.4ms   remaining: 22.7s
3:      learn: 0.9324225        total: 75.7ms   remaining: 18.8s
4:      learn: 0.8512381        total: 82.7ms   remaining: 16.4s
5:      learn: 0.7853871        total: 90.4ms   remaining: 15s
6:      learn: 0.7320989        total: 97.6ms   remaining: 13.8s
7:      learn: 0.6859072        total: 105ms    remaining: 13s
8:      learn: 0.6460036        total: 112ms    remaining: 12.3s
9:      learn: 0.6127546        total: 119ms    remaining: 11.8s
10:     learn: 0.5836265        total: 127ms    remaining: 11.4s
11:     learn: 0.5583323        total: 134ms    remaining: 11.1s
12:     learn: 0.5366990        total: 141ms    remaining: 10.7s
13:     learn: 0.5179303        total: 148ms    remaining: 10.5s
14:     learn: 0.5010923        total: 156ms    remaining: 10.2s
```

```
15:     learn: 0.4853409        total: 163ms    remaining: 10s
16:     learn: 0.4716105        total: 171ms    remaining: 9.87s
17:     learn: 0.4597189        total: 178ms    remaining: 9.7s
18:     learn: 0.4502741        total: 186ms    remaining: 9.58s
19:     learn: 0.4406657        total: 194ms    remaining: 9.49s
20:     learn: 0.4312547        total: 201ms    remaining: 9.38s
21:     learn: 0.4239782        total: 208ms    remaining: 9.27s
22:     learn: 0.4168639        total: 216ms    remaining: 9.18s
23:     learn: 0.4108495        total: 224ms    remaining: 9.09s
24:     learn: 0.4048171        total: 231ms    remaining: 9.03s
25:     learn: 0.4002474        total: 239ms    remaining: 8.94s
26:     learn: 0.3948642        total: 246ms    remaining: 8.87s
27:     learn: 0.3905778        total: 253ms    remaining: 8.8s
28:     learn: 0.3865111        total: 261ms    remaining: 8.73s
29:     learn: 0.3830778        total: 268ms    remaining: 8.66s
30:     learn: 0.3793834        total: 276ms    remaining: 8.61s
31:     learn: 0.3763023        total: 283ms    remaining: 8.55s
32:     learn: 0.3733040        total: 290ms    remaining: 8.51s
33:     learn: 0.3708325        total: 298ms    remaining: 8.46s
34:     learn: 0.3685117        total: 306ms    remaining: 8.43s
35:     learn: 0.3663786        total: 313ms    remaining: 8.38s
36:     learn: 0.3645554        total: 320ms    remaining: 8.33s
37:     learn: 0.3624776        total: 328ms    remaining: 8.3s
38:     learn: 0.3610828        total: 335ms    remaining: 8.25s
39:     learn: 0.3593106        total: 342ms    remaining: 8.21s
40:     learn: 0.3578384        total: 349ms    remaining: 8.17s
41:     learn: 0.3565607        total: 356ms    remaining: 8.13s
42:     learn: 0.3555905        total: 364ms    remaining: 8.1s
43:     learn: 0.3541787        total: 372ms    remaining: 8.08s
44:     learn: 0.3528002        total: 380ms    remaining: 8.07s
45:     learn: 0.3516864        total: 389ms    remaining: 8.06s
46:     learn: 0.3508731        total: 398ms    remaining: 8.07s
47:     learn: 0.3494920        total: 407ms    remaining: 8.07s
48:     learn: 0.3487934        total: 415ms    remaining: 8.05s
49:     learn: 0.3477037        total: 424ms    remaining: 8.05s
50:     learn: 0.3469140        total: 432ms    remaining: 8.05s
51:     learn: 0.3453808        total: 441ms    remaining: 8.04s
52:     learn: 0.3447179        total: 448ms    remaining: 8.01s
53:     learn: 0.3439234        total: 457ms    remaining: 8s
54:     learn: 0.3429559        total: 464ms    remaining: 7.98s
55:     learn: 0.3422897        total: 472ms    remaining: 7.96s
56:     learn: 0.3416474        total: 480ms    remaining: 7.95s
57:     learn: 0.3406864        total: 488ms    remaining: 7.93s
58:     learn: 0.3399309        total: 496ms    remaining: 7.91s
59:     learn: 0.3391858        total: 504ms    remaining: 7.89s
60:     learn: 0.3387536        total: 511ms    remaining: 7.87s
61:     learn: 0.3377070        total: 520ms    remaining: 7.87s
62:     learn: 0.3370588        total: 528ms    remaining: 7.85s
```

```
63:     learn: 0.3363704        total: 535ms    remaining: 7.82s
64:     learn: 0.3356958        total: 542ms    remaining: 7.8s
65:     learn: 0.3353070        total: 549ms    remaining: 7.77s
66:     learn: 0.3349174        total: 557ms    remaining: 7.75s
67:     learn: 0.3345524        total: 564ms    remaining: 7.73s
68:     learn: 0.3341070        total: 572ms    remaining: 7.71s
69:     learn: 0.3337545        total: 580ms    remaining: 7.7s
70:     learn: 0.3332682        total: 588ms    remaining: 7.69s
71:     learn: 0.3329092        total: 595ms    remaining: 7.67s
72:     learn: 0.3324539        total: 603ms    remaining: 7.65s
73:     learn: 0.3320672        total: 610ms    remaining: 7.63s
74:     learn: 0.3317933        total: 617ms    remaining: 7.61s
75:     learn: 0.3314509        total: 625ms    remaining: 7.59s
76:     learn: 0.3309820        total: 632ms    remaining: 7.57s
77:     learn: 0.3304885        total: 640ms    remaining: 7.56s
78:     learn: 0.3301069        total: 647ms    remaining: 7.54s
79:     learn: 0.3296825        total: 654ms    remaining: 7.53s
80:     learn: 0.3293810        total: 662ms    remaining: 7.51s
81:     learn: 0.3290358        total: 669ms    remaining: 7.49s
82:     learn: 0.3285153        total: 676ms    remaining: 7.47s
83:     learn: 0.3280747        total: 684ms    remaining: 7.46s
84:     learn: 0.3276492        total: 692ms    remaining: 7.45s
85:     learn: 0.3273596        total: 700ms    remaining: 7.43s
86:     learn: 0.3267286        total: 707ms    remaining: 7.42s
87:     learn: 0.3262128        total: 715ms    remaining: 7.41s
88:     learn: 0.3260363        total: 722ms    remaining: 7.39s
89:     learn: 0.3253299        total: 729ms    remaining: 7.38s
90:     learn: 0.3249573        total: 737ms    remaining: 7.36s
91:     learn: 0.3246462        total: 744ms    remaining: 7.34s
92:     learn: 0.3242540        total: 751ms    remaining: 7.33s
93:     learn: 0.3238695        total: 759ms    remaining: 7.32s
94:     learn: 0.3234644        total: 768ms    remaining: 7.32s
95:     learn: 0.3232198        total: 776ms    remaining: 7.3s
96:     learn: 0.3229112        total: 783ms    remaining: 7.29s
97:     learn: 0.3227656        total: 791ms    remaining: 7.28s
98:     learn: 0.3224998        total: 799ms    remaining: 7.27s
99:     learn: 0.3222392        total: 806ms    remaining: 7.25s
100:    learn: 0.3218574        total: 813ms    remaining: 7.24s
101:    learn: 0.3215052        total: 820ms    remaining: 7.22s
102:    learn: 0.3212635        total: 828ms    remaining: 7.21s
103:    learn: 0.3210057        total: 836ms    remaining: 7.2s
104:    learn: 0.3207104        total: 843ms    remaining: 7.18s
105:    learn: 0.3204815        total: 851ms    remaining: 7.17s
106:    learn: 0.3202467        total: 858ms    remaining: 7.16s
107:    learn: 0.3199836        total: 865ms    remaining: 7.15s
108:    learn: 0.3196855        total: 872ms    remaining: 7.13s
109:    learn: 0.3193554        total: 880ms    remaining: 7.12s
110:    learn: 0.3191390        total: 887ms    remaining: 7.11s
```

```
111:    learn: 0.3188513        total: 894ms    remaining: 7.09s
112:    learn: 0.3185933        total: 902ms    remaining: 7.08s
113:    learn: 0.3182833        total: 909ms    remaining: 7.06s
114:    learn: 0.3180700        total: 916ms    remaining: 7.05s
115:    learn: 0.3178187        total: 924ms    remaining: 7.04s
116:    learn: 0.3175419        total: 931ms    remaining: 7.03s
117:    learn: 0.3170542        total: 939ms    remaining: 7.01s
118:    learn: 0.3167939        total: 946ms    remaining: 7s
119:    learn: 0.3166898        total: 953ms    remaining: 6.99s
120:    learn: 0.3163538        total: 962ms    remaining: 6.99s
121:    learn: 0.3161846        total: 969ms    remaining: 6.97s
122:    learn: 0.3159116        total: 976ms    remaining: 6.96s
123:    learn: 0.3156031        total: 984ms    remaining: 6.95s
124:    learn: 0.3154736        total: 991ms    remaining: 6.93s
125:    learn: 0.3152449        total: 997ms    remaining: 6.92s
126:    learn: 0.3150685        total: 1s       remaining: 6.91s
127:    learn: 0.3147320        total: 1.01s    remaining: 6.89s
128:    learn: 0.3146055        total: 1.02s    remaining: 6.88s
129:    learn: 0.3144447        total: 1.03s    remaining: 6.87s
130:    learn: 0.3142139        total: 1.03s    remaining: 6.86s
131:    learn: 0.3139351        total: 1.04s    remaining: 6.84s
132:    learn: 0.3138232        total: 1.05s    remaining: 6.83s
133:    learn: 0.3135151        total: 1.05s    remaining: 6.82s
134:    learn: 0.3132271        total: 1.06s    remaining: 6.81s
135:    learn: 0.3129443        total: 1.07s    remaining: 6.8s
136:    learn: 0.3128251        total: 1.08s    remaining: 6.78s
137:    learn: 0.3125505        total: 1.08s    remaining: 6.77s
138:    learn: 0.3123120        total: 1.09s    remaining: 6.75s
139:    learn: 0.3120231        total: 1.1s     remaining: 6.74s
140:    learn: 0.3117023        total: 1.1s     remaining: 6.73s
141:    learn: 0.3113421        total: 1.11s    remaining: 6.73s
142:    learn: 0.3110563        total: 1.12s    remaining: 6.71s
143:    learn: 0.3108357        total: 1.13s    remaining: 6.71s
144:    learn: 0.3106754        total: 1.14s    remaining: 6.7s
145:    learn: 0.3105135        total: 1.14s    remaining: 6.68s
146:    learn: 0.3102814        total: 1.15s    remaining: 6.67s
147:    learn: 0.3100420        total: 1.16s    remaining: 6.67s
148:    learn: 0.3098938        total: 1.17s    remaining: 6.66s
149:    learn: 0.3096807        total: 1.17s    remaining: 6.64s
150:    learn: 0.3092025        total: 1.18s    remaining: 6.64s
151:    learn: 0.3088701        total: 1.19s    remaining: 6.63s
152:    learn: 0.3086877        total: 1.19s    remaining: 6.61s
153:    learn: 0.3084698        total: 1.2s     remaining: 6.6s
154:    learn: 0.3081860        total: 1.21s    remaining: 6.59s
155:    learn: 0.3080581        total: 1.22s    remaining: 6.58s
156:    learn: 0.3079340        total: 1.22s    remaining: 6.57s
157:    learn: 0.3078414        total: 1.23s    remaining: 6.56s
158:    learn: 0.3076427        total: 1.24s    remaining: 6.55s
```

```
159:    learn: 0.3075519        total: 1.25s    remaining: 6.54s
160:    learn: 0.3073687        total: 1.25s    remaining: 6.53s
161:    learn: 0.3069506        total: 1.26s    remaining: 6.52s
162:    learn: 0.3067768        total: 1.27s    remaining: 6.51s
163:    learn: 0.3065772        total: 1.27s    remaining: 6.5s
164:    learn: 0.3063273        total: 1.28s    remaining: 6.49s
165:    learn: 0.3060458        total: 1.29s    remaining: 6.48s
166:    learn: 0.3058541        total: 1.3s     remaining: 6.47s
167:    learn: 0.3056538        total: 1.3s     remaining: 6.45s
168:    learn: 0.3055516        total: 1.31s    remaining: 6.44s
169:    learn: 0.3054550        total: 1.32s    remaining: 6.43s
170:    learn: 0.3052639        total: 1.32s    remaining: 6.42s
171:    learn: 0.3051029        total: 1.33s    remaining: 6.41s
172:    learn: 0.3048459        total: 1.34s    remaining: 6.41s
173:    learn: 0.3046987        total: 1.35s    remaining: 6.4s
174:    learn: 0.3044079        total: 1.36s    remaining: 6.4s
175:    learn: 0.3040922        total: 1.36s    remaining: 6.39s
176:    learn: 0.3038473        total: 1.37s    remaining: 6.38s
177:    learn: 0.3036797        total: 1.38s    remaining: 6.38s
178:    learn: 0.3035071        total: 1.39s    remaining: 6.37s
179:    learn: 0.3034446        total: 1.4s     remaining: 6.36s
180:    learn: 0.3031263        total: 1.4s     remaining: 6.35s
181:    learn: 0.3029916        total: 1.41s    remaining: 6.34s
182:    learn: 0.3028788        total: 1.42s    remaining: 6.33s
183:    learn: 0.3027876        total: 1.43s    remaining: 6.32s
184:    learn: 0.3025922        total: 1.43s    remaining: 6.32s
185:    learn: 0.3023587        total: 1.44s    remaining: 6.31s
186:    learn: 0.3020822        total: 1.45s    remaining: 6.3s
187:    learn: 0.3018802        total: 1.46s    remaining: 6.3s
188:    learn: 0.3017025        total: 1.47s    remaining: 6.29s
189:    learn: 0.3015268        total: 1.47s    remaining: 6.28s
190:    learn: 0.3012035        total: 1.48s    remaining: 6.27s
191:    learn: 0.3007385        total: 1.49s    remaining: 6.26s
192:    learn: 0.3004310        total: 1.5s     remaining: 6.25s
193:    learn: 0.3003090        total: 1.5s     remaining: 6.24s
194:    learn: 0.3001371        total: 1.51s    remaining: 6.23s
195:    learn: 0.3000030        total: 1.52s    remaining: 6.22s
196:    learn: 0.2999337        total: 1.52s    remaining: 6.21s
197:    learn: 0.2997766        total: 1.53s    remaining: 6.21s
198:    learn: 0.2996395        total: 1.54s    remaining: 6.2s
199:    learn: 0.2995349        total: 1.55s    remaining: 6.19s
200:    learn: 0.2994704        total: 1.56s    remaining: 6.18s
201:    learn: 0.2992122        total: 1.56s    remaining: 6.18s
202:    learn: 0.2988785        total: 1.57s    remaining: 6.17s
203:    learn: 0.2987833        total: 1.58s    remaining: 6.16s
204:    learn: 0.2986566        total: 1.59s    remaining: 6.15s
205:    learn: 0.2983585        total: 1.59s    remaining: 6.14s
206:    learn: 0.2982293        total: 1.6s     remaining: 6.13s
```

```
207:    learn: 0.2980764        total: 1.61s    remaining: 6.13s
208:    learn: 0.2979819        total: 1.62s    remaining: 6.12s
209:    learn: 0.2977420        total: 1.62s    remaining: 6.11s
210:    learn: 0.2975648        total: 1.63s    remaining: 6.1s
211:    learn: 0.2974399        total: 1.64s    remaining: 6.09s
212:    learn: 0.2971886        total: 1.65s    remaining: 6.08s
213:    learn: 0.2969474        total: 1.65s    remaining: 6.07s
214:    learn: 0.2967748        total: 1.66s    remaining: 6.07s
215:    learn: 0.2965578        total: 1.67s    remaining: 6.06s
216:    learn: 0.2964196        total: 1.68s    remaining: 6.05s
217:    learn: 0.2963153        total: 1.68s    remaining: 6.04s
218:    learn: 0.2961715        total: 1.69s    remaining: 6.03s
219:    learn: 0.2960194        total: 1.7s     remaining: 6.02s
220:    learn: 0.2958213        total: 1.71s    remaining: 6.01s
221:    learn: 0.2956120        total: 1.71s    remaining: 6s
222:    learn: 0.2952376        total: 1.72s    remaining: 5.99s
223:    learn: 0.2950820        total: 1.73s    remaining: 5.99s
224:    learn: 0.2948585        total: 1.74s    remaining: 5.98s
225:    learn: 0.2947008        total: 1.74s    remaining: 5.97s
226:    learn: 0.2944059        total: 1.75s    remaining: 5.96s
227:    learn: 0.2941847        total: 1.76s    remaining: 5.96s
228:    learn: 0.2939884        total: 1.77s    remaining: 5.95s
229:    learn: 0.2937317        total: 1.77s    remaining: 5.94s
230:    learn: 0.2936406        total: 1.78s    remaining: 5.93s
231:    learn: 0.2935441        total: 1.79s    remaining: 5.92s
232:    learn: 0.2932905        total: 1.8s     remaining: 5.91s
233:    learn: 0.2929802        total: 1.8s     remaining: 5.91s
234:    learn: 0.2928968        total: 1.81s    remaining: 5.9s
235:    learn: 0.2926595        total: 1.82s    remaining: 5.89s
236:    learn: 0.2925535        total: 1.83s    remaining: 5.88s
237:    learn: 0.2923862        total: 1.83s    remaining: 5.87s
238:    learn: 0.2921916        total: 1.84s    remaining: 5.86s
239:    learn: 0.2920157        total: 1.85s    remaining: 5.86s
240:    learn: 0.2918641        total: 1.86s    remaining: 5.85s
241:    learn: 0.2916357        total: 1.86s    remaining: 5.84s
242:    learn: 0.2914777        total: 1.87s    remaining: 5.83s
243:    learn: 0.2911544        total: 1.88s    remaining: 5.82s
244:    learn: 0.2909790        total: 1.89s    remaining: 5.81s
245:    learn: 0.2907741        total: 1.89s    remaining: 5.8s
246:    learn: 0.2906247        total: 1.9s     remaining: 5.8s
247:    learn: 0.2904463        total: 1.91s    remaining: 5.79s
248:    learn: 0.2903623        total: 1.92s    remaining: 5.78s
249:    learn: 0.2902846        total: 1.92s    remaining: 5.77s
250:    learn: 0.2900370        total: 1.93s    remaining: 5.76s
251:    learn: 0.2899138        total: 1.94s    remaining: 5.76s
252:    learn: 0.2897342        total: 1.95s    remaining: 5.75s
253:    learn: 0.2896303        total: 1.96s    remaining: 5.74s
254:    learn: 0.2894996        total: 1.96s    remaining: 5.73s
```

```
255:    learn: 0.2893975        total: 1.97s    remaining: 5.73s
256:    learn: 0.2893023        total: 1.98s    remaining: 5.72s
257:    learn: 0.2891347        total: 1.98s    remaining: 5.71s
258:    learn: 0.2890863        total: 1.99s    remaining: 5.7s
259:    learn: 0.2889392        total: 2s       remaining: 5.69s
260:    learn: 0.2888390        total: 2s       remaining: 5.68s
261:    learn: 0.2887682        total: 2.01s    remaining: 5.67s
262:    learn: 0.2886334        total: 2.02s    remaining: 5.66s
263:    learn: 0.2883602        total: 2.03s    remaining: 5.65s
264:    learn: 0.2880646        total: 2.04s    remaining: 5.65s
265:    learn: 0.2879505        total: 2.04s    remaining: 5.64s
266:    learn: 0.2877939        total: 2.05s    remaining: 5.63s
267:    learn: 0.2876634        total: 2.06s    remaining: 5.62s
268:    learn: 0.2875485        total: 2.06s    remaining: 5.61s
269:    learn: 0.2874937        total: 2.07s    remaining: 5.6s
270:    learn: 0.2872978        total: 2.08s    remaining: 5.6s
271:    learn: 0.2872399        total: 2.09s    remaining: 5.59s
272:    learn: 0.2871813        total: 2.1s     remaining: 5.58s
273:    learn: 0.2869684        total: 2.1s     remaining: 5.57s
274:    learn: 0.2869146        total: 2.11s    remaining: 5.56s
275:    learn: 0.2868002        total: 2.12s    remaining: 5.55s
276:    learn: 0.2867317        total: 2.12s    remaining: 5.54s
277:    learn: 0.2866347        total: 2.13s    remaining: 5.53s
278:    learn: 0.2865329        total: 2.14s    remaining: 5.53s
279:    learn: 0.2863083        total: 2.15s    remaining: 5.52s
280:    learn: 0.2862363        total: 2.15s    remaining: 5.51s
281:    learn: 0.2860988        total: 2.16s    remaining: 5.5s
282:    learn: 0.2860045        total: 2.17s    remaining: 5.49s
283:    learn: 0.2858422        total: 2.17s    remaining: 5.48s
284:    learn: 0.2857088        total: 2.18s    remaining: 5.48s
285:    learn: 0.2856125        total: 2.19s    remaining: 5.46s
286:    learn: 0.2854828        total: 2.2s     remaining: 5.46s
287:    learn: 0.2854296        total: 2.2s     remaining: 5.45s
288:    learn: 0.2853180        total: 2.21s    remaining: 5.44s
289:    learn: 0.2852148        total: 2.22s    remaining: 5.43s
290:    learn: 0.2851561        total: 2.22s    remaining: 5.42s
291:    learn: 0.2849438        total: 2.23s    remaining: 5.41s
292:    learn: 0.2848324        total: 2.24s    remaining: 5.4s
293:    learn: 0.2847122        total: 2.25s    remaining: 5.39s
294:    learn: 0.2846087        total: 2.25s    remaining: 5.38s
295:    learn: 0.2843233        total: 2.26s    remaining: 5.38s
296:    learn: 0.2841439        total: 2.27s    remaining: 5.37s
297:    learn: 0.2839032        total: 2.28s    remaining: 5.37s
298:    learn: 0.2838425        total: 2.29s    remaining: 5.36s
299:    learn: 0.2837701        total: 2.29s    remaining: 5.35s
300:    learn: 0.2836107        total: 2.3s     remaining: 5.34s
301:    learn: 0.2834460        total: 2.31s    remaining: 5.33s
302:    learn: 0.2834191        total: 2.31s    remaining: 5.33s
```

```
303:    learn: 0.2832764        total: 2.32s    remaining: 5.32s
304:    learn: 0.2830771        total: 2.33s    remaining: 5.31s
305:    learn: 0.2828967        total: 2.34s    remaining: 5.31s
306:    learn: 0.2828295        total: 2.35s    remaining: 5.3s
307:    learn: 0.2827167        total: 2.35s    remaining: 5.29s
308:    learn: 0.2825927        total: 2.36s    remaining: 5.28s
309:    learn: 0.2824670        total: 2.37s    remaining: 5.28s
310:    learn: 0.2823342        total: 2.38s    remaining: 5.27s
311:    learn: 0.2821392        total: 2.38s    remaining: 5.26s
312:    learn: 0.2820286        total: 2.39s    remaining: 5.25s
313:    learn: 0.2819301        total: 2.4s     remaining: 5.25s
314:    learn: 0.2817654        total: 2.41s    remaining: 5.24s
315:    learn: 0.2816647        total: 2.42s    remaining: 5.23s
316:    learn: 0.2816350        total: 2.42s    remaining: 5.22s
317:    learn: 0.2814436        total: 2.43s    remaining: 5.21s
318:    learn: 0.2813485        total: 2.44s    remaining: 5.21s
319:    learn: 0.2812003        total: 2.45s    remaining: 5.2s
320:    learn: 0.2811422        total: 2.45s    remaining: 5.19s
321:    learn: 0.2810597        total: 2.46s    remaining: 5.18s
322:    learn: 0.2806856        total: 2.47s    remaining: 5.17s
323:    learn: 0.2805422        total: 2.48s    remaining: 5.17s
324:    learn: 0.2804687        total: 2.48s    remaining: 5.16s
325:    learn: 0.2803749        total: 2.49s    remaining: 5.15s
326:    learn: 0.2802394        total: 2.5s     remaining: 5.14s
327:    learn: 0.2801796        total: 2.5s     remaining: 5.13s
328:    learn: 0.2800828        total: 2.51s    remaining: 5.12s
329:    learn: 0.2799945        total: 2.52s    remaining: 5.12s
330:    learn: 0.2799123        total: 2.53s    remaining: 5.11s
331:    learn: 0.2797786        total: 2.54s    remaining: 5.1s
332:    learn: 0.2797258        total: 2.54s    remaining: 5.09s
333:    learn: 0.2795476        total: 2.55s    remaining: 5.09s
334:    learn: 0.2794557        total: 2.56s    remaining: 5.08s
335:    learn: 0.2793389        total: 2.56s    remaining: 5.07s
336:    learn: 0.2792048        total: 2.57s    remaining: 5.06s
337:    learn: 0.2790901        total: 2.58s    remaining: 5.05s
338:    learn: 0.2789249        total: 2.59s    remaining: 5.05s
339:    learn: 0.2788251        total: 2.6s     remaining: 5.04s
340:    learn: 0.2786801        total: 2.6s     remaining: 5.03s
341:    learn: 0.2784752        total: 2.61s    remaining: 5.02s
342:    learn: 0.2783179        total: 2.62s    remaining: 5.02s
343:    learn: 0.2781507        total: 2.63s    remaining: 5.01s
344:    learn: 0.2780993        total: 2.63s    remaining: 5s
345:    learn: 0.2780323        total: 2.64s    remaining: 4.99s
346:    learn: 0.2779317        total: 2.65s    remaining: 4.98s
347:    learn: 0.2778121        total: 2.65s    remaining: 4.97s
348:    learn: 0.2777047        total: 2.66s    remaining: 4.97s
349:    learn: 0.2775535        total: 2.67s    remaining: 4.96s
350:    learn: 0.2773645        total: 2.68s    remaining: 4.95s
```

```
351:    learn: 0.2771848        total: 2.69s    remaining: 4.94s
352:    learn: 0.2770078        total: 2.69s    remaining: 4.94s
353:    learn: 0.2769627        total: 2.7s     remaining: 4.93s
354:    learn: 0.2768266        total: 2.71s    remaining: 4.92s
355:    learn: 0.2766265        total: 2.71s    remaining: 4.91s
356:    learn: 0.2765308        total: 2.72s    remaining: 4.9s
357:    learn: 0.2763595        total: 2.73s    remaining: 4.9s
358:    learn: 0.2762800        total: 2.74s    remaining: 4.9s
359:    learn: 0.2762059        total: 2.75s    remaining: 4.89s
360:    learn: 0.2760642        total: 2.76s    remaining: 4.88s
361:    learn: 0.2758808        total: 2.76s    remaining: 4.87s
362:    learn: 0.2756919        total: 2.77s    remaining: 4.86s
363:    learn: 0.2755950        total: 2.78s    remaining: 4.86s
364:    learn: 0.2754755        total: 2.79s    remaining: 4.85s
365:    learn: 0.2753811        total: 2.79s    remaining: 4.84s
366:    learn: 0.2753415        total: 2.8s     remaining: 4.83s
367:    learn: 0.2752057        total: 2.81s    remaining: 4.82s
368:    learn: 0.2751101        total: 2.81s    remaining: 4.81s
369:    learn: 0.2749902        total: 2.82s    remaining: 4.8s
370:    learn: 0.2748836        total: 2.83s    remaining: 4.8s
371:    learn: 0.2747845        total: 2.84s    remaining: 4.79s
372:    learn: 0.2746280        total: 2.84s    remaining: 4.78s
373:    learn: 0.2746074        total: 2.85s    remaining: 4.77s
374:    learn: 0.2745211        total: 2.86s    remaining: 4.76s
375:    learn: 0.2744139        total: 2.87s    remaining: 4.76s
376:    learn: 0.2743423        total: 2.87s    remaining: 4.75s
377:    learn: 0.2741605        total: 2.88s    remaining: 4.74s
378:    learn: 0.2740890        total: 2.89s    remaining: 4.73s
379:    learn: 0.2740222        total: 2.9s     remaining: 4.72s
380:    learn: 0.2739548        total: 2.9s     remaining: 4.71s
381:    learn: 0.2738998        total: 2.91s    remaining: 4.71s
382:    learn: 0.2738104        total: 2.92s    remaining: 4.7s
383:    learn: 0.2736776        total: 2.93s    remaining: 4.7s
384:    learn: 0.2735818        total: 2.93s    remaining: 4.69s
385:    learn: 0.2734350        total: 2.94s    remaining: 4.68s
386:    learn: 0.2731996        total: 2.95s    remaining: 4.67s
387:    learn: 0.2731152        total: 2.96s    remaining: 4.66s
388:    learn: 0.2729615        total: 2.96s    remaining: 4.65s
389:    learn: 0.2728568        total: 2.97s    remaining: 4.65s
390:    learn: 0.2727674        total: 2.98s    remaining: 4.64s
391:    learn: 0.2726242        total: 2.98s    remaining: 4.63s
392:    learn: 0.2725703        total: 2.99s    remaining: 4.62s
393:    learn: 0.2724866        total: 3s       remaining: 4.62s
394:    learn: 0.2723185        total: 3.01s    remaining: 4.61s
395:    learn: 0.2721251        total: 3.02s    remaining: 4.6s
396:    learn: 0.2720314        total: 3.02s    remaining: 4.59s
397:    learn: 0.2718489        total: 3.03s    remaining: 4.58s
398:    learn: 0.2715912        total: 3.04s    remaining: 4.58s
```

```
399:    learn: 0.2715370        total: 3.04s    remaining: 4.57s
400:    learn: 0.2714473        total: 3.05s    remaining: 4.56s
401:    learn: 0.2713676        total: 3.06s    remaining: 4.55s
402:    learn: 0.2713123        total: 3.07s    remaining: 4.54s
403:    learn: 0.2711768        total: 3.07s    remaining: 4.53s
404:    learn: 0.2711442        total: 3.08s    remaining: 4.53s
405:    learn: 0.2710982        total: 3.09s    remaining: 4.52s
406:    learn: 0.2709339        total: 3.1s     remaining: 4.51s
407:    learn: 0.2707510        total: 3.1s     remaining: 4.51s
408:    learn: 0.2706651        total: 3.11s    remaining: 4.5s
409:    learn: 0.2705372        total: 3.12s    remaining: 4.49s
410:    learn: 0.2703747        total: 3.13s    remaining: 4.48s
411:    learn: 0.2702618        total: 3.13s    remaining: 4.47s
412:    learn: 0.2700903        total: 3.14s    remaining: 4.47s
413:    learn: 0.2699547        total: 3.15s    remaining: 4.46s
414:    learn: 0.2698805        total: 3.16s    remaining: 4.45s
415:    learn: 0.2698208        total: 3.17s    remaining: 4.44s
416:    learn: 0.2697193        total: 3.17s    remaining: 4.44s
417:    learn: 0.2696734        total: 3.18s    remaining: 4.43s
418:    learn: 0.2695965        total: 3.19s    remaining: 4.42s
419:    learn: 0.2695523        total: 3.19s    remaining: 4.41s
420:    learn: 0.2694732        total: 3.2s     remaining: 4.4s
421:    learn: 0.2693970        total: 3.21s    remaining: 4.39s
422:    learn: 0.2692339        total: 3.22s    remaining: 4.39s
423:    learn: 0.2691280        total: 3.23s    remaining: 4.38s
424:    learn: 0.2690069        total: 3.23s    remaining: 4.37s
425:    learn: 0.2689698        total: 3.24s    remaining: 4.37s
426:    learn: 0.2688704        total: 3.25s    remaining: 4.36s
427:    learn: 0.2687216        total: 3.26s    remaining: 4.35s
428:    learn: 0.2686494        total: 3.26s    remaining: 4.34s
429:    learn: 0.2686343        total: 3.27s    remaining: 4.34s
430:    learn: 0.2684711        total: 3.28s    remaining: 4.33s
431:    learn: 0.2684492        total: 3.29s    remaining: 4.32s
432:    learn: 0.2682538        total: 3.29s    remaining: 4.31s
433:    learn: 0.2681412        total: 3.3s     remaining: 4.31s
434:    learn: 0.2679848        total: 3.31s    remaining: 4.3s
435:    learn: 0.2679128        total: 3.32s    remaining: 4.29s
436:    learn: 0.2677697        total: 3.33s    remaining: 4.29s
437:    learn: 0.2677214        total: 3.33s    remaining: 4.28s
438:    learn: 0.2676480        total: 3.34s    remaining: 4.27s
439:    learn: 0.2675775        total: 3.35s    remaining: 4.26s
440:    learn: 0.2674252        total: 3.36s    remaining: 4.26s
441:    learn: 0.2673185        total: 3.37s    remaining: 4.25s
442:    learn: 0.2671441        total: 3.38s    remaining: 4.24s
443:    learn: 0.2669481        total: 3.38s    remaining: 4.24s
444:    learn: 0.2667661        total: 3.39s    remaining: 4.23s
445:    learn: 0.2666731        total: 3.4s     remaining: 4.22s
446:    learn: 0.2665686        total: 3.4s     remaining: 4.21s
```

```
447:    learn: 0.2664755        total: 3.41s    remaining: 4.2s
448:    learn: 0.2663772        total: 3.42s    remaining: 4.2s
449:    learn: 0.2662563        total: 3.43s    remaining: 4.19s
450:    learn: 0.2661997        total: 3.43s    remaining: 4.18s
451:    learn: 0.2661308        total: 3.44s    remaining: 4.17s
452:    learn: 0.2659972        total: 3.45s    remaining: 4.17s
453:    learn: 0.2659452        total: 3.46s    remaining: 4.16s
454:    learn: 0.2658157        total: 3.46s    remaining: 4.15s
455:    learn: 0.2657104        total: 3.47s    remaining: 4.14s
456:    learn: 0.2655687        total: 3.48s    remaining: 4.13s
457:    learn: 0.2654697        total: 3.49s    remaining: 4.13s
458:    learn: 0.2653428        total: 3.5s     remaining: 4.12s
459:    learn: 0.2652493        total: 3.5s     remaining: 4.11s
460:    learn: 0.2651329        total: 3.51s    remaining: 4.11s
461:    learn: 0.2650281        total: 3.52s    remaining: 4.1s
462:    learn: 0.2649503        total: 3.53s    remaining: 4.09s
463:    learn: 0.2648554        total: 3.53s    remaining: 4.08s
464:    learn: 0.2647904        total: 3.54s    remaining: 4.07s
465:    learn: 0.2646860        total: 3.55s    remaining: 4.07s
466:    learn: 0.2645238        total: 3.56s    remaining: 4.06s
467:    learn: 0.2644677        total: 3.56s    remaining: 4.05s
468:    learn: 0.2644021        total: 3.57s    remaining: 4.04s
469:    learn: 0.2643269        total: 3.58s    remaining: 4.03s
470:    learn: 0.2642437        total: 3.58s    remaining: 4.03s
471:    learn: 0.2640994        total: 3.59s    remaining: 4.02s
472:    learn: 0.2639635        total: 3.6s     remaining: 4.01s
473:    learn: 0.2638524        total: 3.61s    remaining: 4s
474:    learn: 0.2638131        total: 3.61s    remaining: 4s
475:    learn: 0.2637532        total: 3.62s    remaining: 3.99s
476:    learn: 0.2636818        total: 3.63s    remaining: 3.98s
477:    learn: 0.2635738        total: 3.64s    remaining: 3.97s
478:    learn: 0.2635466        total: 3.64s    remaining: 3.96s
479:    learn: 0.2635027        total: 3.65s    remaining: 3.96s
480:    learn: 0.2633766        total: 3.66s    remaining: 3.95s
481:    learn: 0.2632529        total: 3.67s    remaining: 3.94s
482:    learn: 0.2631791        total: 3.67s    remaining: 3.93s
483:    learn: 0.2631311        total: 3.68s    remaining: 3.92s
484:    learn: 0.2630556        total: 3.69s    remaining: 3.92s
485:    learn: 0.2629462        total: 3.7s     remaining: 3.91s
486:    learn: 0.2628667        total: 3.7s     remaining: 3.9s
487:    learn: 0.2627838        total: 3.71s    remaining: 3.89s
488:    learn: 0.2627442        total: 3.72s    remaining: 3.88s
489:    learn: 0.2627140        total: 3.73s    remaining: 3.88s
490:    learn: 0.2625980        total: 3.73s    remaining: 3.87s
491:    learn: 0.2625185        total: 3.74s    remaining: 3.86s
492:    learn: 0.2624342        total: 3.75s    remaining: 3.85s
493:    learn: 0.2623986        total: 3.75s    remaining: 3.85s
494:    learn: 0.2622996        total: 3.76s    remaining: 3.84s
```

```
495:    learn: 0.2622389        total: 3.77s    remaining: 3.83s
496:    learn: 0.2621410        total: 3.78s    remaining: 3.82s
497:    learn: 0.2620643        total: 3.78s    remaining: 3.81s
498:    learn: 0.2619698        total: 3.79s    remaining: 3.81s
499:    learn: 0.2619091        total: 3.8s     remaining: 3.8s
500:    learn: 0.2618412        total: 3.81s    remaining: 3.79s
501:    learn: 0.2616134        total: 3.81s    remaining: 3.78s
502:    learn: 0.2615327        total: 3.82s    remaining: 3.77s
503:    learn: 0.2614042        total: 3.83s    remaining: 3.77s
504:    learn: 0.2613577        total: 3.83s    remaining: 3.76s
505:    learn: 0.2612906        total: 3.84s    remaining: 3.75s
506:    learn: 0.2612019        total: 3.85s    remaining: 3.74s
507:    learn: 0.2611275        total: 3.86s    remaining: 3.73s
508:    learn: 0.2610486        total: 3.86s    remaining: 3.73s
509:    learn: 0.2609381        total: 3.87s    remaining: 3.72s
510:    learn: 0.2608610        total: 3.88s    remaining: 3.71s
511:    learn: 0.2607812        total: 3.89s    remaining: 3.71s
512:    learn: 0.2606889        total: 3.9s     remaining: 3.7s
513:    learn: 0.2606198        total: 3.9s     remaining: 3.69s
514:    learn: 0.2605012        total: 3.91s    remaining: 3.68s
515:    learn: 0.2603590        total: 3.92s    remaining: 3.68s
516:    learn: 0.2602507        total: 3.93s    remaining: 3.67s
517:    learn: 0.2601635        total: 3.94s    remaining: 3.66s
518:    learn: 0.2600866        total: 3.94s    remaining: 3.65s
519:    learn: 0.2600342        total: 3.95s    remaining: 3.65s
520:    learn: 0.2599889        total: 3.96s    remaining: 3.64s
521:    learn: 0.2598739        total: 3.96s    remaining: 3.63s
522:    learn: 0.2598229        total: 3.97s    remaining: 3.62s
523:    learn: 0.2597798        total: 3.98s    remaining: 3.61s
524:    learn: 0.2596204        total: 3.99s    remaining: 3.61s
525:    learn: 0.2595216        total: 3.99s    remaining: 3.6s
526:    learn: 0.2594760        total: 4s       remaining: 3.59s
527:    learn: 0.2593831        total: 4.01s    remaining: 3.58s
528:    learn: 0.2592598        total: 4.01s    remaining: 3.57s
529:    learn: 0.2591620        total: 4.02s    remaining: 3.57s
530:    learn: 0.2590982        total: 4.03s    remaining: 3.56s
531:    learn: 0.2589366        total: 4.04s    remaining: 3.55s
532:    learn: 0.2588241        total: 4.04s    remaining: 3.54s
533:    learn: 0.2586980        total: 4.05s    remaining: 3.54s
534:    learn: 0.2585799        total: 4.06s    remaining: 3.53s
535:    learn: 0.2585128        total: 4.07s    remaining: 3.52s
536:    learn: 0.2584854        total: 4.08s    remaining: 3.51s
537:    learn: 0.2583866        total: 4.08s    remaining: 3.51s
538:    learn: 0.2583490        total: 4.09s    remaining: 3.5s
539:    learn: 0.2582692        total: 4.1s     remaining: 3.49s
540:    learn: 0.2582089        total: 4.11s    remaining: 3.48s
541:    learn: 0.2580421        total: 4.11s    remaining: 3.48s
542:    learn: 0.2579957        total: 4.12s    remaining: 3.47s
```

```
543:    learn: 0.2579280        total: 4.13s    remaining: 3.46s
544:    learn: 0.2578227        total: 4.13s    remaining: 3.45s
545:    learn: 0.2577363        total: 4.14s    remaining: 3.44s
546:    learn: 0.2576146        total: 4.15s    remaining: 3.44s
547:    learn: 0.2575824        total: 4.16s    remaining: 3.43s
548:    learn: 0.2574884        total: 4.17s    remaining: 3.42s
549:    learn: 0.2573977        total: 4.17s    remaining: 3.41s
550:    learn: 0.2573460        total: 4.18s    remaining: 3.41s
551:    learn: 0.2572974        total: 4.19s    remaining: 3.4s
552:    learn: 0.2572084        total: 4.2s     remaining: 3.39s
553:    learn: 0.2571171        total: 4.2s     remaining: 3.38s
554:    learn: 0.2570363        total: 4.21s    remaining: 3.38s
555:    learn: 0.2569413        total: 4.22s    remaining: 3.37s
556:    learn: 0.2568277        total: 4.23s    remaining: 3.36s
557:    learn: 0.2567347        total: 4.24s    remaining: 3.36s
558:    learn: 0.2566790        total: 4.24s    remaining: 3.35s
559:    learn: 0.2564978        total: 4.25s    remaining: 3.34s
560:    learn: 0.2563875        total: 4.26s    remaining: 3.33s
561:    learn: 0.2562755        total: 4.27s    remaining: 3.33s
562:    learn: 0.2562320        total: 4.28s    remaining: 3.32s
563:    learn: 0.2561290        total: 4.29s    remaining: 3.31s
564:    learn: 0.2559529        total: 4.3s     remaining: 3.31s
565:    learn: 0.2558752        total: 4.3s     remaining: 3.3s
566:    learn: 0.2557987        total: 4.31s    remaining: 3.29s
567:    learn: 0.2557429        total: 4.32s    remaining: 3.29s
568:    learn: 0.2556387        total: 4.33s    remaining: 3.28s
569:    learn: 0.2555791        total: 4.33s    remaining: 3.27s
570:    learn: 0.2554833        total: 4.34s    remaining: 3.26s
571:    learn: 0.2554149        total: 4.35s    remaining: 3.25s
572:    learn: 0.2553730        total: 4.36s    remaining: 3.25s
573:    learn: 0.2553181        total: 4.36s    remaining: 3.24s
574:    learn: 0.2552577        total: 4.37s    remaining: 3.23s
575:    learn: 0.2551989        total: 4.38s    remaining: 3.22s
576:    learn: 0.2550614        total: 4.39s    remaining: 3.21s
577:    learn: 0.2549901        total: 4.39s    remaining: 3.21s
578:    learn: 0.2549443        total: 4.4s     remaining: 3.2s
579:    learn: 0.2548686        total: 4.41s    remaining: 3.19s
580:    learn: 0.2547628        total: 4.42s    remaining: 3.18s
581:    learn: 0.2546871        total: 4.42s    remaining: 3.18s
582:    learn: 0.2545694        total: 4.43s    remaining: 3.17s
583:    learn: 0.2544708        total: 4.44s    remaining: 3.16s
584:    learn: 0.2544167        total: 4.45s    remaining: 3.15s
585:    learn: 0.2543251        total: 4.45s    remaining: 3.15s
586:    learn: 0.2541328        total: 4.46s    remaining: 3.14s
587:    learn: 0.2540413        total: 4.47s    remaining: 3.13s
588:    learn: 0.2539794        total: 4.48s    remaining: 3.13s
589:    learn: 0.2539332        total: 4.49s    remaining: 3.12s
590:    learn: 0.2538267        total: 4.49s    remaining: 3.11s
```

```
591:    learn: 0.2537850        total: 4.5s     remaining: 3.1s
592:    learn: 0.2536604        total: 4.51s    remaining: 3.09s
593:    learn: 0.2535307        total: 4.52s    remaining: 3.09s
594:    learn: 0.2534960        total: 4.52s    remaining: 3.08s
595:    learn: 0.2534150        total: 4.53s    remaining: 3.07s
596:    learn: 0.2532974        total: 4.54s    remaining: 3.06s
597:    learn: 0.2532718        total: 4.54s    remaining: 3.06s
598:    learn: 0.2531975        total: 4.55s    remaining: 3.05s
599:    learn: 0.2531433        total: 4.56s    remaining: 3.04s
600:    learn: 0.2530987        total: 4.57s    remaining: 3.03s
601:    learn: 0.2530236        total: 4.58s    remaining: 3.02s
602:    learn: 0.2529594        total: 4.58s    remaining: 3.02s
603:    learn: 0.2528604        total: 4.59s    remaining: 3.01s
604:    learn: 0.2527809        total: 4.6s     remaining: 3s
605:    learn: 0.2527136        total: 4.6s     remaining: 2.99s
606:    learn: 0.2526509        total: 4.61s    remaining: 2.98s
607:    learn: 0.2525609        total: 4.62s    remaining: 2.98s
608:    learn: 0.2524903        total: 4.63s    remaining: 2.97s
609:    learn: 0.2524149        total: 4.63s    remaining: 2.96s
610:    learn: 0.2523628        total: 4.64s    remaining: 2.95s
611:    learn: 0.2522800        total: 4.65s    remaining: 2.95s
612:    learn: 0.2521952        total: 4.66s    remaining: 2.94s
613:    learn: 0.2520861        total: 4.66s    remaining: 2.93s
614:    learn: 0.2519779        total: 4.67s    remaining: 2.92s
615:    learn: 0.2518622        total: 4.68s    remaining: 2.92s
616:    learn: 0.2518109        total: 4.69s    remaining: 2.91s
617:    learn: 0.2517450        total: 4.69s    remaining: 2.9s
618:    learn: 0.2516997        total: 4.7s     remaining: 2.89s
619:    learn: 0.2516343        total: 4.71s    remaining: 2.89s
620:    learn: 0.2516080        total: 4.71s    remaining: 2.88s
621:    learn: 0.2515378        total: 4.72s    remaining: 2.87s
622:    learn: 0.2514635        total: 4.73s    remaining: 2.86s
623:    learn: 0.2514105        total: 4.74s    remaining: 2.85s
624:    learn: 0.2513373        total: 4.75s    remaining: 2.85s
625:    learn: 0.2512395        total: 4.75s    remaining: 2.84s
626:    learn: 0.2512103        total: 4.76s    remaining: 2.83s
627:    learn: 0.2511467        total: 4.77s    remaining: 2.82s
628:    learn: 0.2510757        total: 4.77s    remaining: 2.81s
629:    learn: 0.2510504        total: 4.78s    remaining: 2.81s
630:    learn: 0.2509978        total: 4.79s    remaining: 2.8s
631:    learn: 0.2509216        total: 4.8s     remaining: 2.79s
632:    learn: 0.2508374        total: 4.8s     remaining: 2.79s
633:    learn: 0.2507506        total: 4.81s    remaining: 2.78s
634:    learn: 0.2506682        total: 4.82s    remaining: 2.77s
635:    learn: 0.2506382        total: 4.83s    remaining: 2.76s
636:    learn: 0.2505775        total: 4.83s    remaining: 2.75s
637:    learn: 0.2505133        total: 4.84s    remaining: 2.75s
638:    learn: 0.2504828        total: 4.85s    remaining: 2.74s
```

```
639:    learn: 0.2503835        total: 4.86s    remaining: 2.73s
640:    learn: 0.2502881        total: 4.86s    remaining: 2.72s
641:    learn: 0.2502153        total: 4.87s    remaining: 2.72s
642:    learn: 0.2501345        total: 4.88s    remaining: 2.71s
643:    learn: 0.2500177        total: 4.89s    remaining: 2.7s
644:    learn: 0.2499869        total: 4.89s    remaining: 2.69s
645:    learn: 0.2499174        total: 4.9s     remaining: 2.69s
646:    learn: 0.2498406        total: 4.91s    remaining: 2.68s
647:    learn: 0.2497184        total: 4.92s    remaining: 2.67s
648:    learn: 0.2496460        total: 4.92s    remaining: 2.66s
649:    learn: 0.2495841        total: 4.93s    remaining: 2.65s
650:    learn: 0.2494848        total: 4.94s    remaining: 2.65s
651:    learn: 0.2493982        total: 4.94s    remaining: 2.64s
652:    learn: 0.2493221        total: 4.95s    remaining: 2.63s
653:    learn: 0.2492545        total: 4.96s    remaining: 2.62s
654:    learn: 0.2492282        total: 4.97s    remaining: 2.62s
655:    learn: 0.2492035        total: 4.97s    remaining: 2.61s
656:    learn: 0.2491251        total: 4.98s    remaining: 2.6s
657:    learn: 0.2490212        total: 4.99s    remaining: 2.59s
658:    learn: 0.2489457        total: 5s       remaining: 2.58s
659:    learn: 0.2488720        total: 5s       remaining: 2.58s
660:    learn: 0.2487760        total: 5.01s    remaining: 2.57s
661:    learn: 0.2487314        total: 5.02s    remaining: 2.56s
662:    learn: 0.2485755        total: 5.03s    remaining: 2.55s
663:    learn: 0.2485261        total: 5.03s    remaining: 2.55s
664:    learn: 0.2484915        total: 5.04s    remaining: 2.54s
665:    learn: 0.2484249        total: 5.05s    remaining: 2.53s
666:    learn: 0.2483742        total: 5.06s    remaining: 2.52s
667:    learn: 0.2483028        total: 5.06s    remaining: 2.52s
668:    learn: 0.2482228        total: 5.07s    remaining: 2.51s
669:    learn: 0.2481703        total: 5.08s    remaining: 2.5s
670:    learn: 0.2480730        total: 5.09s    remaining: 2.49s
671:    learn: 0.2479865        total: 5.09s    remaining: 2.49s
672:    learn: 0.2478955        total: 5.1s     remaining: 2.48s
673:    learn: 0.2477964        total: 5.11s    remaining: 2.47s
674:    learn: 0.2477327        total: 5.12s    remaining: 2.46s
675:    learn: 0.2476652        total: 5.13s    remaining: 2.46s
676:    learn: 0.2475798        total: 5.13s    remaining: 2.45s
677:    learn: 0.2475275        total: 5.14s    remaining: 2.44s
678:    learn: 0.2474632        total: 5.15s    remaining: 2.43s
679:    learn: 0.2474131        total: 5.16s    remaining: 2.43s
680:    learn: 0.2473781        total: 5.16s    remaining: 2.42s
681:    learn: 0.2473414        total: 5.17s    remaining: 2.41s
682:    learn: 0.2473021        total: 5.18s    remaining: 2.4s
683:    learn: 0.2472110        total: 5.19s    remaining: 2.4s
684:    learn: 0.2471606        total: 5.19s    remaining: 2.39s
685:    learn: 0.2470858        total: 5.2s     remaining: 2.38s
686:    learn: 0.2470344        total: 5.21s    remaining: 2.37s
```

```
687:    learn: 0.2469815        total: 5.22s    remaining: 2.37s
688:    learn: 0.2468282        total: 5.23s    remaining: 2.36s
689:    learn: 0.2468002        total: 5.23s    remaining: 2.35s
690:    learn: 0.2466897        total: 5.24s    remaining: 2.34s
691:    learn: 0.2466176        total: 5.25s    remaining: 2.34s
692:    learn: 0.2465773        total: 5.26s    remaining: 2.33s
693:    learn: 0.2464733        total: 5.27s    remaining: 2.32s
694:    learn: 0.2464192        total: 5.28s    remaining: 2.31s
695:    learn: 0.2463325        total: 5.28s    remaining: 2.31s
696:    learn: 0.2462587        total: 5.29s    remaining: 2.3s
697:    learn: 0.2461783        total: 5.3s     remaining: 2.29s
698:    learn: 0.2460542        total: 5.3s     remaining: 2.28s
699:    learn: 0.2460010        total: 5.31s    remaining: 2.28s
700:    learn: 0.2459332        total: 5.32s    remaining: 2.27s
701:    learn: 0.2458208        total: 5.33s    remaining: 2.26s
702:    learn: 0.2457946        total: 5.33s    remaining: 2.25s
703:    learn: 0.2457297        total: 5.34s    remaining: 2.25s
704:    learn: 0.2456276        total: 5.35s    remaining: 2.24s
705:    learn: 0.2455102        total: 5.36s    remaining: 2.23s
706:    learn: 0.2454660        total: 5.37s    remaining: 2.22s
707:    learn: 0.2453557        total: 5.37s    remaining: 2.21s
708:    learn: 0.2453003        total: 5.38s    remaining: 2.21s
709:    learn: 0.2452285        total: 5.39s    remaining: 2.2s
710:    learn: 0.2451386        total: 5.39s    remaining: 2.19s
711:    learn: 0.2450980        total: 5.4s     remaining: 2.19s
712:    learn: 0.2449816        total: 5.41s    remaining: 2.18s
713:    learn: 0.2448702        total: 5.42s    remaining: 2.17s
714:    learn: 0.2448089        total: 5.42s    remaining: 2.16s
715:    learn: 0.2447162        total: 5.43s    remaining: 2.15s
716:    learn: 0.2446358        total: 5.44s    remaining: 2.15s
717:    learn: 0.2445681        total: 5.45s    remaining: 2.14s
718:    learn: 0.2444621        total: 5.46s    remaining: 2.13s
719:    learn: 0.2443842        total: 5.46s    remaining: 2.13s
720:    learn: 0.2442862        total: 5.47s    remaining: 2.12s
721:    learn: 0.2441680        total: 5.48s    remaining: 2.11s
722:    learn: 0.2441028        total: 5.49s    remaining: 2.1s
723:    learn: 0.2440395        total: 5.5s     remaining: 2.09s
724:    learn: 0.2439629        total: 5.5s     remaining: 2.09s
725:    learn: 0.2439167        total: 5.51s    remaining: 2.08s
726:    learn: 0.2438415        total: 5.52s    remaining: 2.07s
727:    learn: 0.2437722        total: 5.52s    remaining: 2.06s
728:    learn: 0.2437210        total: 5.53s    remaining: 2.06s
729:    learn: 0.2436845        total: 5.54s    remaining: 2.05s
730:    learn: 0.2436351        total: 5.55s    remaining: 2.04s
731:    learn: 0.2435013        total: 5.55s    remaining: 2.03s
732:    learn: 0.2434133        total: 5.56s    remaining: 2.02s
733:    learn: 0.2433338        total: 5.57s    remaining: 2.02s
734:    learn: 0.2432645        total: 5.58s    remaining: 2.01s
```

```
735:    learn: 0.2432152        total: 5.58s    remaining: 2s
736:    learn: 0.2431784        total: 5.59s    remaining: 2s
737:    learn: 0.2430920        total: 5.6s     remaining: 1.99s
738:    learn: 0.2430269        total: 5.61s    remaining: 1.98s
739:    learn: 0.2429774        total: 5.61s    remaining: 1.97s
740:    learn: 0.2428955        total: 5.62s    remaining: 1.97s
741:    learn: 0.2427522        total: 5.63s    remaining: 1.96s
742:    learn: 0.2427221        total: 5.64s    remaining: 1.95s
743:    learn: 0.2426554        total: 5.65s    remaining: 1.94s
744:    learn: 0.2426063        total: 5.66s    remaining: 1.94s
745:    learn: 0.2425689        total: 5.66s    remaining: 1.93s
746:    learn: 0.2424839        total: 5.67s    remaining: 1.92s
747:    learn: 0.2423727        total: 5.68s    remaining: 1.91s
748:    learn: 0.2423126        total: 5.69s    remaining: 1.91s
749:    learn: 0.2422774        total: 5.69s    remaining: 1.9s
750:    learn: 0.2422398        total: 5.7s     remaining: 1.89s
751:    learn: 0.2421432        total: 5.71s    remaining: 1.88s
752:    learn: 0.2420607        total: 5.71s    remaining: 1.87s
753:    learn: 0.2419898        total: 5.72s    remaining: 1.87s
754:    learn: 0.2419038        total: 5.73s    remaining: 1.86s
755:    learn: 0.2418619        total: 5.74s    remaining: 1.85s
756:    learn: 0.2417782        total: 5.75s    remaining: 1.84s
757:    learn: 0.2417512        total: 5.75s    remaining: 1.84s
758:    learn: 0.2416958        total: 5.76s    remaining: 1.83s
759:    learn: 0.2416352        total: 5.77s    remaining: 1.82s
760:    learn: 0.2415404        total: 5.77s    remaining: 1.81s
761:    learn: 0.2414559        total: 5.78s    remaining: 1.8s
762:    learn: 0.2414113        total: 5.79s    remaining: 1.8s
763:    learn: 0.2413430        total: 5.79s    remaining: 1.79s
764:    learn: 0.2412581        total: 5.8s     remaining: 1.78s
765:    learn: 0.2412011        total: 5.81s    remaining: 1.77s
766:    learn: 0.2411704        total: 5.82s    remaining: 1.77s
767:    learn: 0.2410891        total: 5.83s    remaining: 1.76s
768:    learn: 0.2410102        total: 5.84s    remaining: 1.75s
769:    learn: 0.2409573        total: 5.84s    remaining: 1.75s
770:    learn: 0.2408509        total: 5.85s    remaining: 1.74s
771:    learn: 0.2407937        total: 5.86s    remaining: 1.73s
772:    learn: 0.2407418        total: 5.87s    remaining: 1.72s
773:    learn: 0.2407036        total: 5.87s    remaining: 1.71s
774:    learn: 0.2406555        total: 5.88s    remaining: 1.71s
775:    learn: 0.2405979        total: 5.89s    remaining: 1.7s
776:    learn: 0.2405323        total: 5.89s    remaining: 1.69s
777:    learn: 0.2404858        total: 5.9s     remaining: 1.68s
778:    learn: 0.2404314        total: 5.91s    remaining: 1.68s
779:    learn: 0.2403472        total: 5.92s    remaining: 1.67s
780:    learn: 0.2402917        total: 5.92s    remaining: 1.66s
781:    learn: 0.2402277        total: 5.93s    remaining: 1.65s
782:    learn: 0.2401784        total: 5.94s    remaining: 1.65s
```

```
783:    learn: 0.2401033        total: 5.95s    remaining: 1.64s
784:    learn: 0.2400614        total: 5.95s    remaining: 1.63s
785:    learn: 0.2399870        total: 5.96s    remaining: 1.62s
786:    learn: 0.2398988        total: 5.97s    remaining: 1.61s
787:    learn: 0.2398065        total: 5.97s    remaining: 1.61s
788:    learn: 0.2397505        total: 5.98s    remaining: 1.6s
789:    learn: 0.2396933        total: 5.99s    remaining: 1.59s
790:    learn: 0.2395890        total: 6s       remaining: 1.58s
791:    learn: 0.2394818        total: 6.01s    remaining: 1.58s
792:    learn: 0.2394190        total: 6.01s    remaining: 1.57s
793:    learn: 0.2393150        total: 6.02s    remaining: 1.56s
794:    learn: 0.2392703        total: 6.03s    remaining: 1.55s
795:    learn: 0.2391980        total: 6.04s    remaining: 1.55s
796:    learn: 0.2391705        total: 6.05s    remaining: 1.54s
797:    learn: 0.2391108        total: 6.05s    remaining: 1.53s
798:    learn: 0.2390182        total: 6.06s    remaining: 1.52s
799:    learn: 0.2389851        total: 6.07s    remaining: 1.52s
800:    learn: 0.2388930        total: 6.08s    remaining: 1.51s
801:    learn: 0.2388357        total: 6.08s    remaining: 1.5s
802:    learn: 0.2387489        total: 6.09s    remaining: 1.49s
803:    learn: 0.2386977        total: 6.1s     remaining: 1.49s
804:    learn: 0.2386389        total: 6.11s    remaining: 1.48s
805:    learn: 0.2385319        total: 6.12s    remaining: 1.47s
806:    learn: 0.2384250        total: 6.12s    remaining: 1.46s
807:    learn: 0.2383672        total: 6.13s    remaining: 1.46s
808:    learn: 0.2383289        total: 6.14s    remaining: 1.45s
809:    learn: 0.2382537        total: 6.15s    remaining: 1.44s
810:    learn: 0.2381266        total: 6.16s    remaining: 1.43s
811:    learn: 0.2380317        total: 6.16s    remaining: 1.43s
812:    learn: 0.2379319        total: 6.17s    remaining: 1.42s
813:    learn: 0.2378978        total: 6.18s    remaining: 1.41s
814:    learn: 0.2378543        total: 6.19s    remaining: 1.4s
815:    learn: 0.2378028        total: 6.2s     remaining: 1.4s
816:    learn: 0.2377452        total: 6.2s     remaining: 1.39s
817:    learn: 0.2376736        total: 6.21s    remaining: 1.38s
818:    learn: 0.2375351        total: 6.22s    remaining: 1.37s
819:    learn: 0.2374697        total: 6.23s    remaining: 1.37s
820:    learn: 0.2374365        total: 6.24s    remaining: 1.36s
821:    learn: 0.2373272        total: 6.24s    remaining: 1.35s
822:    learn: 0.2372314        total: 6.25s    remaining: 1.34s
823:    learn: 0.2372088        total: 6.26s    remaining: 1.34s
824:    learn: 0.2371919        total: 6.26s    remaining: 1.33s
825:    learn: 0.2371528        total: 6.27s    remaining: 1.32s
826:    learn: 0.2370851        total: 6.28s    remaining: 1.31s
827:    learn: 0.2370050        total: 6.29s    remaining: 1.3s
828:    learn: 0.2369530        total: 6.29s    remaining: 1.3s
829:    learn: 0.2369005        total: 6.3s     remaining: 1.29s
830:    learn: 0.2367587        total: 6.31s    remaining: 1.28s
```

```
831:    learn: 0.2367161        total: 6.32s    remaining: 1.27s
832:    learn: 0.2366820        total: 6.32s    remaining: 1.27s
833:    learn: 0.2366410        total: 6.33s    remaining: 1.26s
834:    learn: 0.2365952        total: 6.34s    remaining: 1.25s
835:    learn: 0.2365420        total: 6.34s    remaining: 1.24s
836:    learn: 0.2364650        total: 6.35s    remaining: 1.24s
837:    learn: 0.2364289        total: 6.36s    remaining: 1.23s
838:    learn: 0.2363332        total: 6.37s    remaining: 1.22s
839:    learn: 0.2362635        total: 6.38s    remaining: 1.21s
840:    learn: 0.2361920        total: 6.38s    remaining: 1.21s
841:    learn: 0.2361132        total: 6.39s    remaining: 1.2s
842:    learn: 0.2360761        total: 6.4s     remaining: 1.19s
843:    learn: 0.2360175        total: 6.4s     remaining: 1.18s
844:    learn: 0.2359516        total: 6.41s    remaining: 1.18s
845:    learn: 0.2358883        total: 6.42s    remaining: 1.17s
846:    learn: 0.2358624        total: 6.43s    remaining: 1.16s
847:    learn: 0.2357948        total: 6.44s    remaining: 1.15s
848:    learn: 0.2357759        total: 6.44s    remaining: 1.15s
849:    learn: 0.2357291        total: 6.45s    remaining: 1.14s
850:    learn: 0.2356578        total: 6.46s    remaining: 1.13s
851:    learn: 0.2355918        total: 6.47s    remaining: 1.12s
852:    learn: 0.2355139        total: 6.47s    remaining: 1.11s
853:    learn: 0.2354562        total: 6.48s    remaining: 1.11s
854:    learn: 0.2354232        total: 6.49s    remaining: 1.1s
855:    learn: 0.2353145        total: 6.5s     remaining: 1.09s
856:    learn: 0.2352619        total: 6.5s     remaining: 1.08s
857:    learn: 0.2352086        total: 6.51s    remaining: 1.08s
858:    learn: 0.2351695        total: 6.52s    remaining: 1.07s
859:    learn: 0.2351305        total: 6.53s    remaining: 1.06s
860:    learn: 0.2350502        total: 6.53s    remaining: 1.05s
861:    learn: 0.2349821        total: 6.54s    remaining: 1.05s
862:    learn: 0.2349175        total: 6.55s    remaining: 1.04s
863:    learn: 0.2348624        total: 6.56s    remaining: 1.03s
864:    learn: 0.2348082        total: 6.56s    remaining: 1.02s
865:    learn: 0.2347562        total: 6.57s    remaining: 1.02s
866:    learn: 0.2346959        total: 6.58s    remaining: 1.01s
867:    learn: 0.2346472        total: 6.58s    remaining: 1s
868:    learn: 0.2345931        total: 6.59s    remaining: 994ms
869:    learn: 0.2345503        total: 6.6s     remaining: 986ms
870:    learn: 0.2344721        total: 6.61s    remaining: 979ms
871:    learn: 0.2344212        total: 6.62s    remaining: 971ms
872:    learn: 0.2343557        total: 6.63s    remaining: 964ms
873:    learn: 0.2342924        total: 6.63s    remaining: 956ms
874:    learn: 0.2342732        total: 6.64s    remaining: 948ms
875:    learn: 0.2341240        total: 6.65s    remaining: 941ms
876:    learn: 0.2340940        total: 6.65s    remaining: 933ms
877:    learn: 0.2340432        total: 6.66s    remaining: 926ms
878:    learn: 0.2340047        total: 6.67s    remaining: 918ms
```

```
879:    learn: 0.2339632        total: 6.67s    remaining: 910ms
880:    learn: 0.2338829        total: 6.68s    remaining: 903ms
881:    learn: 0.2338410        total: 6.69s    remaining: 895ms
882:    learn: 0.2337382        total: 6.7s     remaining: 888ms
883:    learn: 0.2337161        total: 6.71s    remaining: 880ms
884:    learn: 0.2336259        total: 6.71s    remaining: 872ms
885:    learn: 0.2335765        total: 6.72s    remaining: 865ms
886:    learn: 0.2334832        total: 6.73s    remaining: 857ms
887:    learn: 0.2334451        total: 6.74s    remaining: 850ms
888:    learn: 0.2333624        total: 6.74s    remaining: 842ms
889:    learn: 0.2332851        total: 6.75s    remaining: 834ms
890:    learn: 0.2332121        total: 6.76s    remaining: 827ms
891:    learn: 0.2331126        total: 6.77s    remaining: 819ms
892:    learn: 0.2330437        total: 6.77s    remaining: 812ms
893:    learn: 0.2329806        total: 6.78s    remaining: 804ms
894:    learn: 0.2328931        total: 6.79s    remaining: 796ms
895:    learn: 0.2328159        total: 6.8s     remaining: 789ms
896:    learn: 0.2327521        total: 6.8s     remaining: 781ms
897:    learn: 0.2326644        total: 6.81s    remaining: 774ms
898:    learn: 0.2326096        total: 6.82s    remaining: 766ms
899:    learn: 0.2325526        total: 6.83s    remaining: 759ms
900:    learn: 0.2325114        total: 6.83s    remaining: 751ms
901:    learn: 0.2324336        total: 6.84s    remaining: 743ms
902:    learn: 0.2323384        total: 6.85s    remaining: 736ms
903:    learn: 0.2322677        total: 6.86s    remaining: 728ms
904:    learn: 0.2322328        total: 6.86s    remaining: 721ms
905:    learn: 0.2322039        total: 6.87s    remaining: 713ms
906:    learn: 0.2321203        total: 6.88s    remaining: 705ms
907:    learn: 0.2320671        total: 6.89s    remaining: 698ms
908:    learn: 0.2319280        total: 6.89s    remaining: 690ms
909:    learn: 0.2318942        total: 6.9s     remaining: 683ms
910:    learn: 0.2318689        total: 6.91s    remaining: 675ms
911:    learn: 0.2317624        total: 6.92s    remaining: 668ms
912:    learn: 0.2316842        total: 6.93s    remaining: 660ms
913:    learn: 0.2316184        total: 6.93s    remaining: 652ms
914:    learn: 0.2314896        total: 6.94s    remaining: 645ms
915:    learn: 0.2314688        total: 6.95s    remaining: 637ms
916:    learn: 0.2314099        total: 6.96s    remaining: 630ms
917:    learn: 0.2313468        total: 6.96s    remaining: 622ms
918:    learn: 0.2313144        total: 6.97s    remaining: 614ms
919:    learn: 0.2312914        total: 6.98s    remaining: 607ms
920:    learn: 0.2312407        total: 6.98s    remaining: 599ms
921:    learn: 0.2312135        total: 6.99s    remaining: 592ms
922:    learn: 0.2311912        total: 7s       remaining: 584ms
923:    learn: 0.2311100        total: 7.01s    remaining: 576ms
924:    learn: 0.2310625        total: 7.02s    remaining: 569ms
925:    learn: 0.2309807        total: 7.02s    remaining: 561ms
926:    learn: 0.2309411        total: 7.03s    remaining: 554ms
```

```
927:    learn: 0.2308876        total: 7.04s    remaining: 546ms
928:    learn: 0.2308109        total: 7.05s    remaining: 539ms
929:    learn: 0.2307344        total: 7.05s    remaining: 531ms
930:    learn: 0.2307143        total: 7.06s    remaining: 523ms
931:    learn: 0.2306580        total: 7.07s    remaining: 516ms
932:    learn: 0.2305934        total: 7.08s    remaining: 508ms
933:    learn: 0.2305455        total: 7.09s    remaining: 501ms
934:    learn: 0.2304997        total: 7.09s    remaining: 493ms
935:    learn: 0.2304641        total: 7.1s     remaining: 486ms
936:    learn: 0.2304107        total: 7.11s    remaining: 478ms
937:    learn: 0.2303863        total: 7.12s    remaining: 470ms
938:    learn: 0.2303519        total: 7.12s    remaining: 463ms
939:    learn: 0.2302582        total: 7.13s    remaining: 455ms
940:    learn: 0.2302023        total: 7.14s    remaining: 448ms
941:    learn: 0.2301642        total: 7.15s    remaining: 440ms
942:    learn: 0.2301271        total: 7.16s    remaining: 433ms
943:    learn: 0.2300482        total: 7.16s    remaining: 425ms
944:    learn: 0.2300011        total: 7.17s    remaining: 417ms
945:    learn: 0.2299670        total: 7.18s    remaining: 410ms
946:    learn: 0.2298987        total: 7.19s    remaining: 402ms
947:    learn: 0.2298143        total: 7.2s     remaining: 395ms
948:    learn: 0.2297749        total: 7.2s     remaining: 387ms
949:    learn: 0.2296857        total: 7.21s    remaining: 380ms
950:    learn: 0.2296166        total: 7.22s    remaining: 372ms
951:    learn: 0.2295877        total: 7.23s    remaining: 364ms
952:    learn: 0.2295127        total: 7.24s    remaining: 357ms
953:    learn: 0.2294792        total: 7.24s    remaining: 349ms
954:    learn: 0.2294254        total: 7.25s    remaining: 342ms
955:    learn: 0.2293847        total: 7.26s    remaining: 334ms
956:    learn: 0.2293468        total: 7.26s    remaining: 326ms
957:    learn: 0.2292948        total: 7.27s    remaining: 319ms
958:    learn: 0.2292384        total: 7.28s    remaining: 311ms
959:    learn: 0.2291672        total: 7.29s    remaining: 304ms
960:    learn: 0.2291234        total: 7.29s    remaining: 296ms
961:    learn: 0.2290274        total: 7.3s     remaining: 288ms
962:    learn: 0.2289454        total: 7.31s    remaining: 281ms
963:    learn: 0.2289060        total: 7.32s    remaining: 273ms
964:    learn: 0.2288306        total: 7.33s    remaining: 266ms
965:    learn: 0.2287294        total: 7.33s    remaining: 258ms
966:    learn: 0.2286653        total: 7.34s    remaining: 250ms
967:    learn: 0.2285951        total: 7.35s    remaining: 243ms
968:    learn: 0.2285522        total: 7.35s    remaining: 235ms
969:    learn: 0.2284793        total: 7.36s    remaining: 228ms
970:    learn: 0.2284124        total: 7.37s    remaining: 220ms
971:    learn: 0.2283616        total: 7.38s    remaining: 213ms
972:    learn: 0.2282761        total: 7.39s    remaining: 205ms
973:    learn: 0.2282065        total: 7.39s    remaining: 197ms
974:    learn: 0.2281337        total: 7.41s    remaining: 190ms
```

```
975:    learn: 0.2280900        total: 7.41s    remaining: 182ms
976:    learn: 0.2280425        total: 7.42s    remaining: 175ms
977:    learn: 0.2279934        total: 7.43s    remaining: 167ms
978:    learn: 0.2279027        total: 7.43s    remaining: 159ms
979:    learn: 0.2278510        total: 7.44s    remaining: 152ms
980:    learn: 0.2278093        total: 7.45s    remaining: 144ms
981:    learn: 0.2277361        total: 7.46s    remaining: 137ms
982:    learn: 0.2276875        total: 7.46s    remaining: 129ms
983:    learn: 0.2276050        total: 7.47s    remaining: 122ms
984:    learn: 0.2275649        total: 7.48s    remaining: 114ms
985:    learn: 0.2275214        total: 7.49s    remaining: 106ms
986:    learn: 0.2274013        total: 7.49s    remaining: 98.7ms
987:    learn: 0.2273376        total: 7.5s     remaining: 91.1ms
988:    learn: 0.2272987        total: 7.51s    remaining: 83.5ms
989:    learn: 0.2272336        total: 7.52s    remaining: 75.9ms
990:    learn: 0.2271730        total: 7.52s    remaining: 68.3ms
991:    learn: 0.2271089        total: 7.53s    remaining: 60.7ms
992:    learn: 0.2270758        total: 7.54s    remaining: 53.1ms
993:    learn: 0.2270244        total: 7.55s    remaining: 45.6ms
994:    learn: 0.2269744        total: 7.55s    remaining: 38ms
995:    learn: 0.2268983        total: 7.56s    remaining: 30.4ms
996:    learn: 0.2268620        total: 7.57s    remaining: 22.8ms
997:    learn: 0.2268171        total: 7.58s    remaining: 15.2ms
998:    learn: 0.2267345        total: 7.59s    remaining: 7.59ms
999:    learn: 0.2266770        total: 7.59s    remaining: 0us
```

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

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

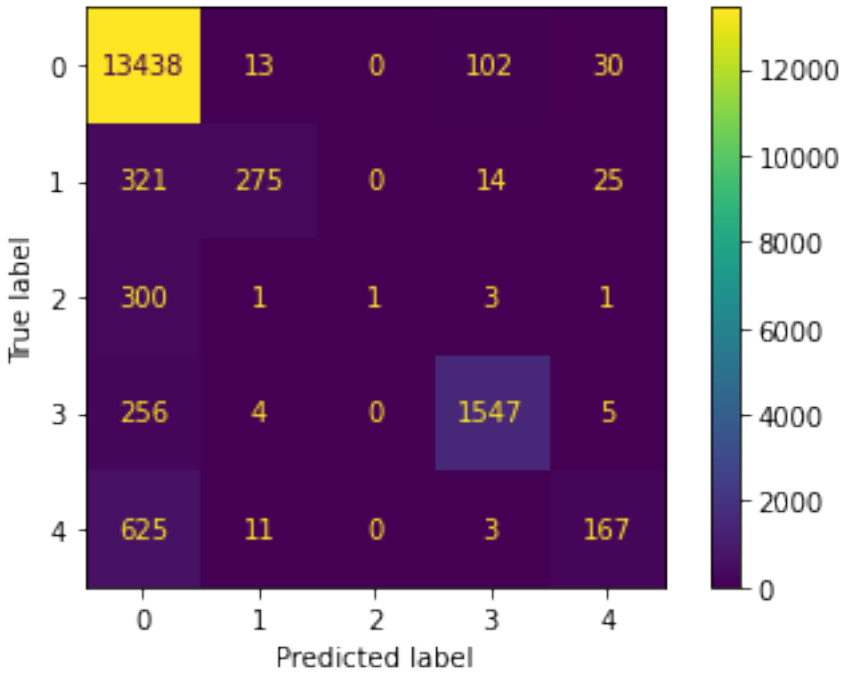
[105]:
```
y_pred = vot_soft.predict(X_test)
```

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

[106]:
```
90.00116672500292
```

[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
0x7efd4fe94fd0>
```



[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   78.920533  0.530600  2.4993       0
29561   127.9872   84.668033  0.500300  2.4997       1
29562   127.9657   96.542533  0.524533  2.4999       1
29563   128.9050   95.799167  0.501867  2.5000       0
29564   128.4784  122.594467  0.606433  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   78.920533  0.530600  2.4993       0
29561  127.9872   84.668033  0.500300  2.4997       1
29562  127.9657   96.542533  0.524533  2.4999       1
29563  128.9050   95.799167  0.501867  2.5000       0
29564  128.4784  122.594467  0.606433  2.5000       0

[29565 rows x 5 columns]
column GR
4 standard deviation outliers -:
             DT         GR      NPHI    RHOB  FACIES
15080  129.1290   919.1449  0.537300  2.2511       0
15081  125.8534  1995.1610  0.556267  2.2460       0
15082  122.6853  1994.9180  0.536833  2.2596       0
15083  122.0640   918.8965  0.499700  2.2751       0
(4, 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   78.920533  0.530600  2.4993       0
29561  127.9872   84.668033  0.500300  2.4997       1
29562  127.9657   96.542533  0.524533  2.4999       1
29563  128.9050   95.799167  0.501867  2.5000       0
29564  128.4784  122.594467  0.606433  2.5000       0

[29561 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   78.920533  0.530600  2.4993       0
29561  127.9872   84.668033  0.500300  2.4997       1
29562  127.9657   96.542533  0.524533  2.4999       1
29563  128.9050   95.799167  0.501867  2.5000       0
29564  128.4784  122.594467  0.606433  2.5000       0

[29488 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   78.920533  0.530600  2.4993       0
29561  127.9872   84.668033  0.500300  2.4997       1
29562  127.9657   96.542533  0.524533  2.4999       1
29563  128.9050   95.799167  0.501867  2.5000       0
29564  128.4784  122.594467  0.606433  2.5000       0
```

```
[29488 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	0.330860	-0.111518	-0.517281	0.059721	0
1	0.286739	-0.065275	-0.629032	0.053573	0
2	0.327199	-0.101672	-0.677419	0.039960	0
3	0.386868	-0.159779	-0.350230	0.021078	0
4	0.431338	-0.197383	0.264977	-0.001317	0
…	…	…	…	…	…
29483	0.038981	-0.428212	0.201613	0.937095	0
29484	0.095408	-0.211389	-0.147465	0.938852	1
29485	0.094480	0.236573	0.131720	0.939730	1
29486	0.135027	0.208530	-0.129416	0.940169	0
29487	0.116612	1.219376	1.075269	0.940169	0

```
[29488 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	0.330860	-0.111518	-0.517281	0.059721
1	0.286739	-0.065275	-0.629032	0.053573
2	0.327199	-0.101672	-0.677419	0.039960

```
3      0.386868 -0.159779 -0.350230  0.021078
4      0.431338 -0.197383  0.264977 -0.001317
…         …        …        …        …
29483  0.038981 -0.428212  0.201613  0.937095
29484  0.095408 -0.211389 -0.147465  0.938852
29485  0.094480  0.236573  0.131720  0.939730
29486  0.135027  0.208530 -0.129416  0.940169
29487  0.116612  1.219376  1.075269  0.940169

[29488 rows x 4 columns]
```

[135]:
```
y_testing
```

[135]:
```
       FACIES
0           0
1           0
2           0
3           0
4           0
…         …
29483       0
29484       1
29485       1
29486       0
29487       0

[29488 rows x 1 columns]
```

[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]:
```
89.16169289202386
```

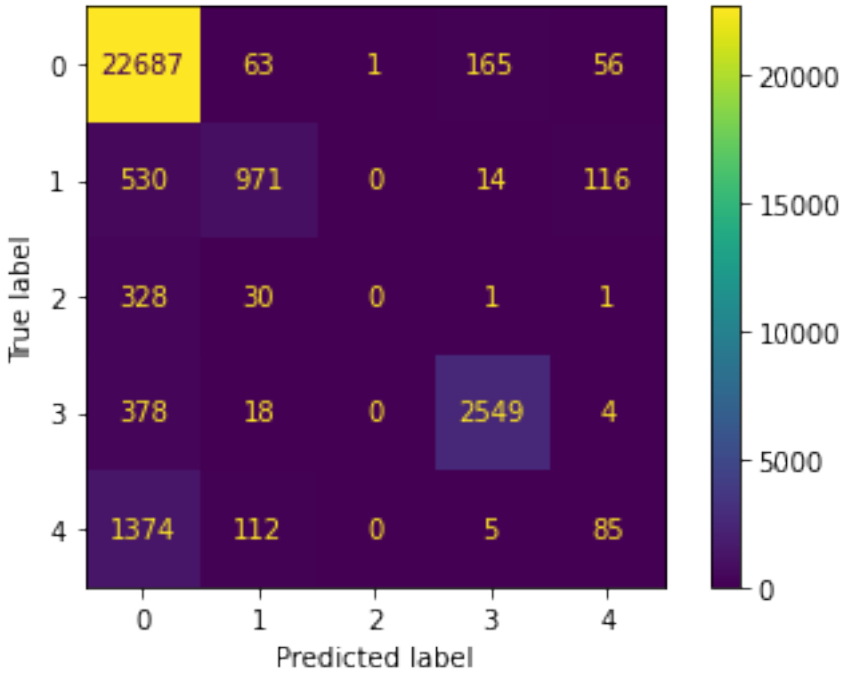
[139]:
```
confusion_matrix(y_testing, y_predicted)
```

[139]:
```
array([[22687,    63,     1,   165,    56],
       [  530,   971,     0,    14,   116],
       [  328,    30,     0,     1,     1],
       [  378,    18,     0,  2549,     4],
       [ 1374,   112,     0,     5,    85]])
```

[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
0x7efd4fe2daf0>
```



[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)
```

[ ]:
```

```