CONTENTS

SOURCE

```
The following data set was extracted from the website : https://www.brenda-enzymes.org/index.php
```

ABOUT PROJECT

```
The objective of this classification project is to build predictive models using Logistic Regression, Decision Tree,
Random Forest, and Ensemble Techniques to map the relationships between enzymes and their possible substrates. Enzymes,
as catalysts, facilitate one or more specific reactions, and understanding the associations between enzymes and their
potential substrates holds significant utility in various domains, including bioinformatics, drug discovery, and
biotechnology.
```

IMPORTING LIBRARIES

In [1]:
```python
import warnings
import numpy as np
import pandas as pd
import seaborn as sns
import xgboost as XGB
import notebook_as_pdf
from scipy import stats
import ydata_profiling as P
import matplotlib.pyplot as plt
warnings.filterwarnings ( 'ignore' )
pd.set_option ( "display.max_columns" , None )
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from imblearn.under_sampling import RandomUnderSampler
from sklearn.model_selection import GridSearchCV , KFold
from sklearn.preprocessing import LabelBinarizer , MinMaxScaler , Normalizer , StandardScaler
from sklearn.metrics import accuracy_score , auc , precision_score , confusion_matrix , f1_score , recall_score
from sklearn.ensemble import RandomForestClassifier , AdaBoostClassifier , GradientBoostingClassifier , ExtraTr
```

IMPORTING DATA

In [2]:
```python
DF_Train = pd.read_csv ( 'Enzyme_Train.csv' )
DF_Test = pd.read_csv ( "Enzyme_Test.csv" )
```

In [3]:
```python
DF_Train
```

Out[3]:

	id	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolWt
0	0	323.390782	9.879918	5.875576	5.875576	4.304757	4.304757	2.754513	1.749203	0.000000	11.938294	222.068080
1	1	273.723798	7.259037	4.441467	5.834958	3.285046	4.485235	2.201375	1.289775	45.135471	0.000000	260.029719
2	2	521.643822	10.911303	8.527859	11.050864	6.665291	9.519706	5.824822	1.770579	15.645394	6.606882	382.131027
3	3	567.431166	12.453343	7.089119	12.833709	6.478023	10.978151	7.914542	3.067181	95.639554	0.000000	530.070277
4	4	112.770735	4.414719	2.866236	2.866236	1.875634	1.875634	1.036450	0.727664	17.980451	12.841643	118.062994
...	...	...	...	...	...	...	...	...	...	...	...	...
14833	14833	632.207041	10.911303	6.579933	9.179964	4.653583	6.030052	3.670528	1.770579	32.971529	6.606882	347.063084
14834	14834	62.568425	2.642734	1.446898	1.446898	0.879497	0.879497	0.174620	0.000000	0.000000	0.000000	74.024203
14835	14835	981.327476	10.363081	6.146219	6.146219	4.700576	4.700576	3.064846	2.133897	17.248535	0.000000	297.089560
14836	14836	299.171248	9.949161	6.589761	7.848913	5.276568	5.476436	3.978973	2.299833	45.623794	0.000000	265.959270
14837	14837	785.394062	15.671142	9.896164	10.234264	7.860296	8.522605	5.645502	3.312893	82.448246	5.687386	437.234828

14838 rows × 38 columns

In [4]:
```python
DF_Test
```

Out[4]:

	id	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolWt
0	14838	344.632371	7.283603	4.473966	5.834958	3.412257	4.651530	2.096558	1.116433	49.458581	0.000000	204.079707
1	14839	1432.410201	10.663869	7.079026	8.065215	5.297097	5.297097	3.924155	2.569694	0.000000	29.783175	315.210331
2	14840	83.352608	3.931852	1.774215	1.774215	1.073446	1.073446	0.467830	0.170838	5.969305	6.420822	101.047679
3	14841	150.255712	5.912790	3.548812	3.548812	2.595128	2.595128	1.642813	0.694113	0.000000	0.000000	187.115661
4	14842	1817.276351	24.910940	15.540529	20.047314	12.535886	17.730988	11.979618	4.431173	84.554972	47.360206	891.167638
...	...	...	...	...	...	...	...	...	...	...	...	...
9888	24726	246.422865	4.036581	2.816709	2.816709	1.875634	1.875634	1.235986	0.362743	24.146543	6.420822	146.105528
9889	24727	591.069706	8.770857	5.682461	5.682461	4.050440	4.050440	2.167855	1.770579	0.000000	28.228515	304.240230
9890	24728	378.113435	6.310349	3.402334	4.317724	2.817428	4.071978	1.970236	1.165747	36.705949	0.000000	260.029719
9891	24729	737.653518	9.949161	7.337949	7.337949	4.428511	5.948361	3.972459	2.160881	36.992053	0.000000	314.068426
9892	24730	785.394062	12.170505	7.565385	9.651755	5.842572	8.229500	5.664696	2.587586	49.458581	11.163878	348.045763

9893 rows × 32 columns

STATISTICAL OVERVIEW

DATA DESCRIPTION

In [5]:
```python
DF_Train.describe ( ).transpose ( )
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
id	14838.0	7418.500000	4283.505982	0.000000	3709.250000	7418.500000	11127.750000	14837.000000
BertzCT	14838.0	515.153604	542.456370	0.000000	149.103601	290.987941	652.652585	4069.959780
Chi1	14838.0	9.135189	6.819989	0.000000	4.680739	6.485270	11.170477	69.551167
Chi1n	14838.0	5.854307	4.647064	0.000000	2.844556	4.052701	7.486791	50.174588
Chi1v	14838.0	6.738497	5.866444	0.000000	2.932842	4.392859	8.527859	53.431954
Chi2n	14838.0	4.432570	3.760516	0.000000	1.949719	2.970427	5.788793	32.195368
Chi2v	14838.0	5.253221	4.925065	0.000000	2.034468	3.242775	6.609350	34.579313
Chi3v	14838.0	3.418749	3.436208	0.000000	1.160763	1.948613	4.502070	22.880836
Chi4n	14838.0	1.773472	1.865898	0.000000	0.503897	1.073261	2.534281	16.072810
EState_VSA1	14838.0	29.202823	31.728679	0.000000	5.969305	17.353601	44.876559	363.705954
EState_VSA2	14838.0	10.435316	13.651843	0.000000	0.000000	6.420822	12.841643	99.936429
ExactMolWt	14838.0	292.623087	225.384140	1.007276	148.037173	206.042653	343.090331	2237.318490
FpDensityMorgan1	14838.0	1.236774	5.491284	-666.000000	1.045455	1.250000	1.500000	3.000000
FpDensityMorgan2	14838.0	1.812070	5.495565	-666.000000	1.690909	1.865152	2.062153	3.200000
FpDensityMorgan3	14838.0	2.255470	5.501200	-666.000000	2.100000	2.358491	2.500000	3.400000
HallKierAlpha	14838.0	-1.207776	0.935314	-7.730000	-1.660000	-1.100000	-0.570000	0.820000
HeavyAtomMolWt	14838.0	274.950211	212.678755	0.000000	136.109000	194.276500	326.002000	2035.133000
Kappa3	14838.0	5.874372	45.730226	-104.040000	1.784008	3.261011	5.848400	1512.242231
MaxAbsEStateIndex	14838.0	10.556443	1.559331	0.000000	9.926190	10.421334	11.539743	15.630251
MinEStateIndex	14838.0	-2.119772	2.066415	-6.327514	-4.659604	-1.265370	-0.787037	6.000000
NumHeteroatoms	14838.0	8.584108	7.643769	0.000000	4.000000	6.000000	10.000000	42.000000
PEOE_VSA10	14838.0	11.021644	13.958962	0.000000	0.000000	6.041841	18.311899	97.663462
PEOE_VSA14	14838.0	17.790011	34.561655	0.000000	0.000000	5.969305	15.645394	482.434223
PEOE_VSA6	14838.0	8.962440	19.756727	0.000000	0.000000	0.000000	12.132734	375.425148
PEOE_VSA7	14838.0	11.318811	20.169745	0.000000	0.000000	0.000000	13.847474	211.501279
PEOE_VSA8	14838.0	6.704487	10.865415	0.000000	0.000000	0.000000	6.923737	100.348416
SMR_VSA10	14838.0	15.666766	18.080208	0.000000	5.969305	11.752550	17.721856	80.742293
SMR_VSA5	14838.0	31.066423	33.896638	0.000000	6.420822	20.075376	42.727765	492.729739
SlogP_VSA3	14838.0	13.636941	14.598554	0.000000	4.794537	9.589074	14.912664	115.406157
VSA_EState9	14838.0	49.309959	29.174824	-5.430556	30.000000	41.666667	56.090650	384.450519
fr_COO	14838.0	0.458215	0.667948	0.000000	0.000000	0.000000	1.000000	8.000000
fr_COO2	14838.0	0.459226	0.668111	0.000000	0.000000	0.000000	1.000000	8.000000
EC1	14838.0	0.667745	0.471038	0.000000	0.000000	1.000000	1.000000	1.000000
EC2	14838.0	0.798962	0.400790	0.000000	1.000000	1.000000	1.000000	1.000000
EC3	14838.0	0.313789	0.464047	0.000000	0.000000	0.000000	1.000000	1.000000
EC4	14838.0	0.279081	0.448562	0.000000	0.000000	0.000000	1.000000	1.000000
EC5	14838.0	0.144831	0.351942	0.000000	0.000000	0.000000	0.000000	1.000000
EC6	14838.0	0.151570	0.358616	0.000000	0.000000	0.000000	0.000000	1.000000

In [6]:
```python
DF_Test.describe ( ).transpose ( )
```

Out[6]:

	count	mean	std	min	25%	50%	75%	max
id	9893.0	19784.000000	2856.007440	14838.000000	17311.000000	19784.000000	22257.000000	24730.000000
BertzCT	9893.0	516.411916	544.327795	0.000000	150.255712	289.901774	652.758463	3919.541449
Chi1	9893.0	9.106998	6.754583	0.000000	4.698377	6.447265	10.966946	53.876516
Chi1n	9893.0	5.848047	4.641662	0.000000	2.846050	4.009996	7.490880	43.283925
Chi1v	9893.0	6.733659	5.863886	0.000000	2.934030	4.337841	8.528316	53.990574
Chi2n	9893.0	4.428979	3.770031	0.000000	1.949719	2.930013	5.788793	36.368883
Chi2v	9893.0	5.247994	4.939702	0.000000	2.049137	3.168052	6.516914	44.726398
Chi3v	9893.0	3.401083	3.431766	0.000000	1.171060	1.923982	4.302610	26.736931
Chi4n	9893.0	1.750078	1.837159	0.000000	0.508512	1.058931	2.509394	15.620667
EState_VSA1	9893.0	28.956335	31.470865	0.000000	5.969305	17.282269	44.876559	363.705954
EState_VSA2	9893.0	10.534500	13.768117	0.000000	0.000000	6.420822	12.841643	99.936429
ExactMolWt	9893.0	292.006497	224.667454	15.007276	148.073559	206.021523	342.116212	1888.793611
FpDensityMorgan1	9893.0	1.280917	0.361229	-1.133333	1.043478	1.250000	1.500000	3.000000
FpDensityMorgan2	9893.0	1.861792	0.348650	0.416667	1.692308	1.866667	2.071429	3.200000
FpDensityMorgan3	9893.0	2.304095	0.382551	0.361111	2.100000	2.358491	2.520000	3.375000
HallKierAlpha	9893.0	-1.206781	0.923173	-7.620000	-1.680000	-1.100000	-0.570000	0.830000
HeavyAtomMolWt	9893.0	274.583106	212.321052	14.007000	140.050000	194.125000	320.121000	1758.846000
Kappa3	9893.0	5.280890	37.349006	-104.040000	1.788507	3.261011	5.772640	1512.242231
MaxAbsEStateIndex	9893.0	10.555415	1.572745	0.000000	9.946009	10.418624	11.526326	14.630251
MinEStateIndex	9893.0	-2.098765	2.059608	-6.117075	-4.638889	-1.252751	-0.787037	4.750000
NumHeteroatoms	9893.0	8.590215	7.660447	0.000000	4.000000	6.000000	10.000000	53.000000
PEOE_VSA10	9893.0	11.033200	13.905575	0.000000	0.000000	6.041841	18.311899	122.017202
PEOE_VSA14	9893.0	18.493154	35.832279	0.000000	0.000000	5.969305	17.907916	482.434223
PEOE_VSA6	9893.0	8.917260	19.358857	0.000000	0.000000	0.000000	12.132734	258.844527
PEOE_VSA7	9893.0	11.222822	20.015732	0.000000	0.000000	5.563451	13.344559	271.424271
PEOE_VSA8	9893.0	6.789234	10.778071	0.000000	0.000000	0.000000	6.923737	96.917997
SMR_VSA10	9893.0	15.810550	18.265549	0.000000	5.969305	11.752550	17.744066	79.920667
SMR_VSA5	9893.0	30.919849	33.532665	0.000000	6.420822	19.765380	42.727765	492.729739
SlogP_VSA3	9893.0	13.601593	14.689893	0.000000	4.794537	9.589074	14.383612	115.406157
VSA_EState9	9893.0	49.426755	29.542758	-5.061146	30.000000	41.666667	56.083333	384.450519
fr_COO	9893.0	0.459719	0.676657	0.000000	0.000000	0.000000	1.000000	8.000000
fr_COO2	9893.0	0.461235	0.677194	0.000000	0.000000	0.000000	1.000000	8.000000

```
There are 38 columns and the data values vary from 0 to a wide range of 4069 which suggests

    1. Data Skewness: Skewed distributions in some columns can lead to extreme values, resulting in a wide range of data
       values.

    2. Outliers: Presence of outliers in some columns can significantly impact the data range, stretching it to higher
       or lower values.

Methods to deal with the above issues

    1. Data Normalization

    2. Outlier Handling

    3. Feature Scaling:
```

INFORMATION

In [7]:
```python
DF_Train.info ( )
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14838 entries, 0 to 14837
Data columns (total 38 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 14838 non-null  int64
 1   BertzCT            14838 non-null  float64
 2   Chi1               14838 non-null  float64
 3   Chi1n              14838 non-null  float64
 4   Chi1v              14838 non-null  float64
 5   Chi2n              14838 non-null  float64
 6   Chi2v              14838 non-null  float64
 7   Chi3v              14838 non-null  float64
 8   Chi4n              14838 non-null  float64
 9   EState_VSA1        14838 non-null  float64
 10  EState_VSA2        14838 non-null  float64
 11  ExactMolWt         14838 non-null  float64
 12  FpDensityMorgan1   14838 non-null  float64
 13  FpDensityMorgan2   14838 non-null  float64
 14  FpDensityMorgan3   14838 non-null  float64
 15  HallKierAlpha      14838 non-null  float64
 16  HeavyAtomMolWt     14838 non-null  float64
 17  Kappa3             14838 non-null  float64
 18  MaxAbsEStateIndex  14838 non-null  float64
 19  MinEStateIndex     14838 non-null  float64
 20  NumHeteroatoms     14838 non-null  int64
 21  PEOE_VSA10         14838 non-null  float64
 22  PEOE_VSA14         14838 non-null  float64
 23  PEOE_VSA6          14838 non-null  float64
 24  PEOE_VSA7          14838 non-null  float64
 25  PEOE_VSA8          14838 non-null  float64
 26  SMR_VSA10          14838 non-null  float64
 27  SMR_VSA5           14838 non-null  float64
 28  SlogP_VSA3         14838 non-null  float64
 29  VSA_EState9        14838 non-null  float64
 30  fr_COO             14838 non-null  int64
 31  fr_COO2            14838 non-null  int64
 32  EC1                14838 non-null  int64
 33  EC2                14838 non-null  int64
 34  EC3                14838 non-null  int64
 35  EC4                14838 non-null  int64
 36  EC5                14838 non-null  int64
 37  EC6                14838 non-null  int64
dtypes: float64(28), int64(10)
memory usage: 4.3 MB
```

In [8]:
```python
DF_Test.info ( )
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9893 entries, 0 to 9892
Data columns (total 32 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 9893 non-null   int64
 1   BertzCT            9893 non-null   float64
 2   Chi1               9893 non-null   float64
 3   Chi1n              9893 non-null   float64
 4   Chi1v              9893 non-null   float64
 5   Chi2n              9893 non-null   float64
 6   Chi2v              9893 non-null   float64
 7   Chi3v              9893 non-null   float64
 8   Chi4n              9893 non-null   float64
 9   EState_VSA1        9893 non-null   float64
 10  EState_VSA2        9893 non-null   float64
 11  ExactMolWt         9893 non-null   float64
 12  FpDensityMorgan1   9893 non-null   float64
 13  FpDensityMorgan2   9893 non-null   float64
 14  FpDensityMorgan3   9893 non-null   float64
 15  HallKierAlpha      9893 non-null   float64
 16  HeavyAtomMolWt     9893 non-null   float64
 17  Kappa3             9893 non-null   float64
 18  MaxAbsEStateIndex  9893 non-null   float64
 19  MinEStateIndex     9893 non-null   float64
 20  NumHeteroatoms     9893 non-null   int64
 21  PEOE_VSA10         9893 non-null   float64
 22  PEOE_VSA14         9893 non-null   float64
 23  PEOE_VSA6          9893 non-null   float64
 24  PEOE_VSA7          9893 non-null   float64
 25  PEOE_VSA8          9893 non-null   float64
 26  SMR_VSA10          9893 non-null   float64
 27  SMR_VSA5           9893 non-null   float64
 28  SlogP_VSA3         9893 non-null   float64
 29  VSA_EState9        9893 non-null   float64
 30  fr_COO             9893 non-null   int64
 31  fr_COO2            9893 non-null   int64
dtypes: float64(28), int64(4)
memory usage: 2.4 MB
```

DROPPING UNIQUE COLUMNS

In [9]:
```python
DF_Train.drop ( columns = [ 'id' , 'EC3' , 'EC4' ,'EC5' ,'EC6' ] , inplace = True )
DF_Test.drop ( columns = [ 'id' ] , inplace = True )
DF_Train
```

Out[9]:

	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolWt
0	323.390782	9.879918	5.875576	5.875576	4.304757	4.304757	2.754513	1.749203	0.000000	11.938294	222.068080
1	273.723798	7.259037	4.441467	5.834958	3.285046	4.485235	2.201375	1.289775	45.135471	0.000000	260.029719
2	521.643822	10.911303	8.527859	11.050864	6.665291	9.519706	5.824822	1.770579	15.645394	6.606882	382.131027
3	567.431166	12.453343	7.089119	12.833709	6.478023	10.978151	7.914542	3.067181	95.639554	0.000000	530.070277
4	112.770735	4.414719	2.866236	2.866236	1.875634	1.875634	1.036450	0.727664	17.980451	12.841643	118.062994
...	...	...	...	...	...	...	...	...	...	...	...
14833	632.207041	10.911303	6.579933	9.179964	4.653583	6.030052	3.670528	1.770579	32.971529	6.606882	347.063084
14834	62.568425	2.642734	1.446898	1.446898	0.879497	0.879497	0.174620	0.000000	0.000000	0.000000	74.024203
14835	981.327476	10.363081	6.146219	6.146219	4.700576	4.700576	3.064846	2.133897	17.248535	0.000000	297.089560
14836	299.171248	9.949161	6.589761	7.848913	5.276568	5.476436	3.978973	2.299833	45.623794	0.000000	265.959270
14837	785.394062	15.671142	9.896164	10.234264	7.860296	8.522605	5.645502	3.312893	82.448246	5.687386	437.234828

14838 rows × 33 columns

```
DETAILS OF COLUMNS

    id: An identifier or index for each observation.

    BertzCT: A molecular descriptor representing the topological complexity of a chemical compound.

    Chi1: A molecular descriptor related to the topological shape of a chemical compound.

    Chi1n: A normalized version of Chi1 descriptor.

    Chi1v: A valence-only version of Chi1 descriptor.

    Chi2n: A normalized version of Chi2 descriptor.

    Chi2v: A valence-only version of Chi2 descriptor.

    Chi3v: A valence-only version of Chi3 descriptor.

    Chi4n: A normalized version of Chi4 descriptor.

    EState_VSA1: A molecular descriptor related to atomic contributions to the E-state values of a
chemical compound.

    EState_VSA2: A molecular descriptor related to atomic contributions to the E-state values of a
chemical compound.

    ExactMolWt: The exact molecular weight of a chemical compound.

    FpDensityMorgan1: A fingerprint density descriptor based on Morgan algorithm (radius=1).

    FpDensityMorgan2: A fingerprint density descriptor based on Morgan algorithm (radius=2).

    FpDensityMorgan3: A fingerprint density descriptor based on Morgan algorithm (radius=3).

    HallKierAlpha: A molecular descriptor related to chemical topology.

    HeavyAtomMolWt: The molecular weight of heavy atoms in a chemical compound.

    Kappa3: A molecular descriptor representing the shape of a chemical compound.

    MaxAbsEStateIndex: The maximum absolute E-state value of atoms in a chemical compound.

    MinEStateIndex: The minimum E-state value of atoms in a chemical compound.

    NumHeteroatoms: The number of heteroatoms (non-carbon atoms) in a chemical compound.

    PEOE_VSA10: A molecular descriptor related to partial charges of atoms in a chemical compound.

    PEOE_VSA14: A molecular descriptor related to partial charges of atoms in a chemical compound.

    PEOE_VSA6: A molecular descriptor related to partial charges of atoms in a chemical compound.

    PEOE_VSA7: A molecular descriptor related to partial charges of atoms in a chemical compound.

    PEOE_VSA8: A molecular descriptor related to partial charges of atoms in a chemical compound.

    SMR_VSA10: A molecular descriptor related to solvent-accessible surface area contributions of atoms
in a chemical
    compound.

    SMR_VSA5: A molecular descriptor related to solvent-accessible surface area contributions of atoms
in a chemical
    compound.

    SlogP_VSA3: A molecular descriptor related to the lipophilicity of a chemical compound.

    VSA_EState9: A molecular descriptor related to E-state values of atoms in a chemical compound.

    fr_COO: The number of carboxylic acid groups (COOH) in a chemical compound.

    fr_COO2: The number of ketene groups (RC=C=O) in a chemical compound.

    EC1: Oxidoreductases ( Enzyme Commission )

    EC2: Transferases

    EC3: Hydrolases

    EC4: Lyases

    EC5: Isomerases

    EC6: Ligases


    * Substrate1 (reducing agent) + Substrate2 (oxidizing agent) ⇌ Product1 (oxidized form) + Product2
(reduced form)

    * The Enzyme Commission number (EC number) is a numerical classification scheme for enzymes, based
on the chemical
      reactions they catalyze.
```

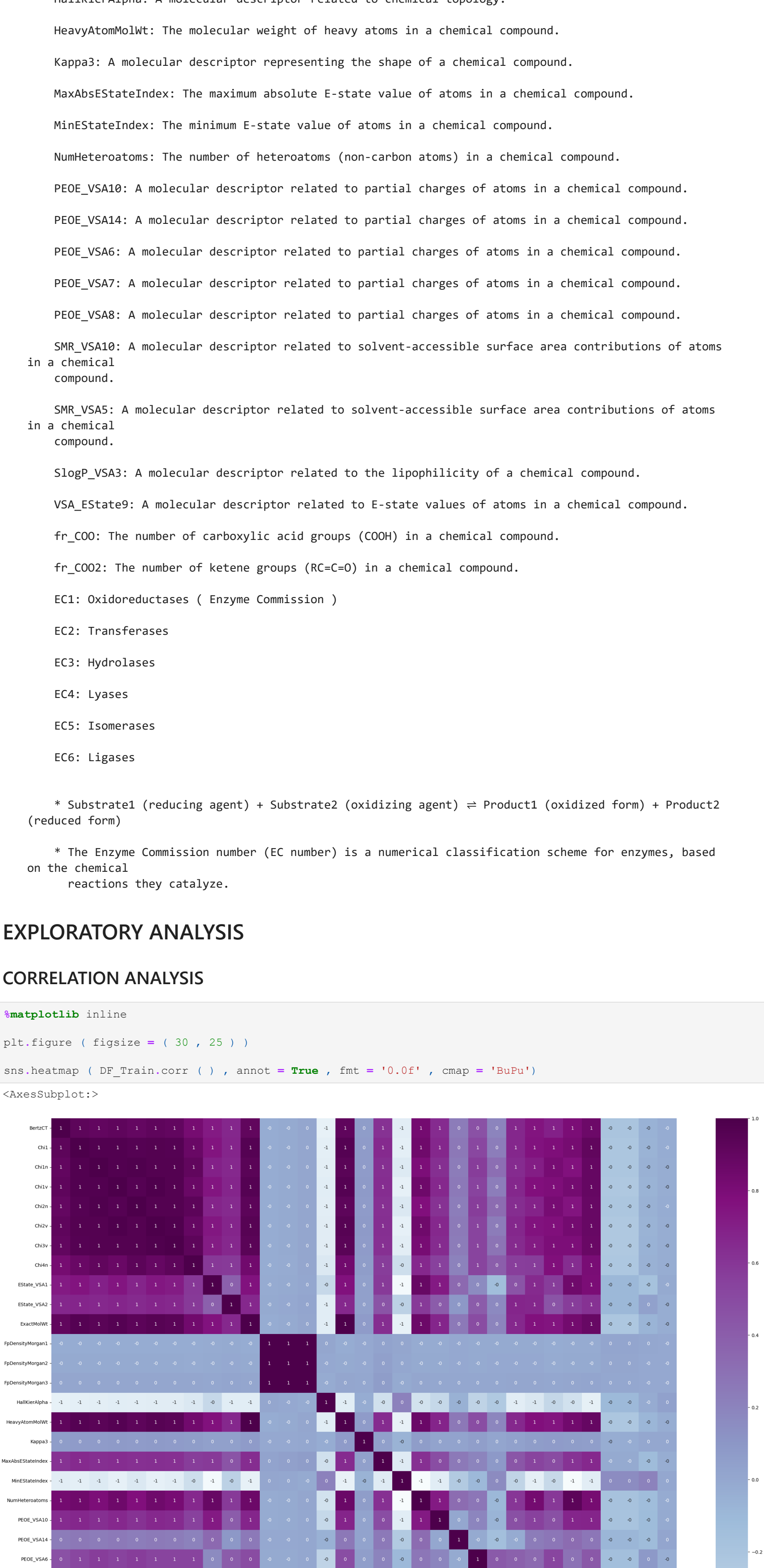
EXPLORATORY ANALYSIS

CORRELATION ANALYSIS

In [10]:
```python
%matplotlib inline
plt.figure ( figsize = ( 30 , 25 ) )
sns.heatmap ( DF_Train.corr ( ) , annot = True , fmt = '0.0f' , cmap = 'BuPu')
```

Out[10]:

<AxesSubplot:>



In [11]:
```python
# Majority of the columns in data set show high correlation with other columns
High_Corr = DF_Train.corr ( ) [ ( DF_Train.corr ( ) >= 0.5 ) | (DF_Train.corr ( ) <= -0.5 ) ].fillna ( "-" )
High_Corr
```

Out[11]:

	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolW
BertzCT	1.0	0.927629	0.906521	0.922312	0.897368	0.919101	0.908613	0.81873	0.72245	0.664584	0.9150
Chi1	0.927629	1.0	0.9735	0.978411	0.960249	0.971049	0.96065	0.879138	0.752059	0.674461	0.96046
Chi1n	0.906521	0.9735	1.0	0.97701	0.983607	0.968451	0.965798	0.914214	0.697719	0.663377	0.94074
Chi1v	0.922312	0.978411	0.97701	1.0	0.962441	0.986538	0.970371	0.877441	0.745145	0.674537	0.96052
Chi2n	0.897368	0.960249	0.983607	0.962441	1.0	0.970987	0.971534	0.934039	0.684409	0.655004	0.93054
Chi2v	0.919101	0.971049	0.968451	0.986538	0.970987	1.0	0.980093	0.889549	0.741071	0.671543	0.95606
Chi3v	0.908613	0.96065	0.965798	0.970371	0.971534	0.980093	1.0	0.925856	0.7173	0.658281	0.94198
Chi4n	0.81873	0.879138	0.914214	0.877441	0.934039	0.889549	0.925856	1.0	0.593447	0.580768	0.84525
EState_VSA1	0.72245	0.752059	0.697719	0.745145	0.684409	0.741071	0.7173	0.593447	1.0	-	0.75899
EState_VSA2	0.664584	0.674461	0.663377	0.674537	0.655004	0.671543	0.658281	0.580768	-	1.0	0.66206
ExactMolWt	0.91501	0.960466	0.940748	0.960524	0.930542	0.956068	0.941985	0.845252	0.758995	0.662065	1
FpDensityMorgan1	-	-	-	-	-	-	-	-	-	-	
FpDensityMorgan2	-	-	-	-	-	-	-	-	-	-	
FpDensityMorgan3	-	-	-	-	-	-	-	-	-	-	
HallKierAlpha	-0.631727	-0.603047	-0.591215	-0.57421	-0.5819	-0.567206	-0.568979	-0.530959	-	-0.579766	-0.57692
HeavyAtomMolWt	0.916533	0.961543	0.939796	0.961587	0.928774	0.956842	0.942007	0.841389	0.766878	0.663346	0.99490
Kappa3	-	-	-	-	-	-	-	-	-	-	
MaxAbsEStateIndex	0.618805	0.653747	0.632353	0.639513	0.623239	0.633889	0.628992	0.572191	0.58332	-	0.64278
MinEStateIndex	-0.584845	-0.607686	-0.542124	-0.626905	-0.526849	-0.623095	-0.584528	-	-0.754944	-	-0.63500
NumHeteroatoms	0.828075	0.852496	0.791468	0.856213	0.77309	0.849558	0.815988	0.660877	0.882927	0.585062	0.86309
PEOE_VSA10	0.646601	0.670871	0.630076	0.66186	0.618925	0.658392	0.644154	0.545786	0.726711	-	0.67436
PEOE_VSA14	-	-	-	-	-	-	-	-	-	-	
PEOE_VSA6	-	0.523459	0.572505	0.542113	0.571197	0.53744	0.535158	0.536849	-	-	
PEOE_VSA7	-	-	-	-	-	-	-	-	-	-	
PEOE_VSA8	0.667995	0.666387	0.648963	0.665717	0.639612	0.665266	0.651354	0.560305	-	0.67729	0.66072
SMR_VSA10	0.752372	0.758098	0.706382	0.770176	0.685918	0.761452	0.724303	0.567885	0.655484	0.667695	0.76472
SMR_VSA5	0.696558	0.783187	0.812716	0.784626	0.813634	0.782354	0.790831	0.781454	0.573015	-	0.75161
SlogP_VSA3	0.801567	0.822502	0.765291	0.830787	0.749101	0.826504	0.792512	0.637785	0.847151	0.568868	0.83434
VSA_EState9	0.860961	0.909979	0.891629	0.890286	0.882566	0.88435	0.87666	0.805501	0.802368	0.61978	0.88708
fr_COO	-	-	-	-	-	-	-	-	-	-	
fr_COO2	-	-	-	-	-	-	-	-	-	-	
EC1	-	-	-	-	-	-	-	-	-	-	
EC2	-	-	-	-	-	-	-	-	-	-	

OUTLIERS

DETECTING OUTLIERS

In [12]:
```python
DF_Train.plot ( kind = 'box' , subplots = True , layout = ( 13 , 3 ), figsize = ( 20 , 100 ) , sharex = True ,
```

TREATING OUTLIERS

    * BertzCT, EState_VSA1, ExactMolWt, HeavyAtomMolWt, Kappa3, PEOE_VSA14, PEOE_VSA7, SMR_VSA5, VSA_EState9

    * Compared to the other columns, the above columns exhibit a higher number of outliers.

In [13]:
```
def Z_Score ( DF ) :

    for i in DF :

        Mean = DF [ i ].mean ( )

        Standard_Deviation = DF [ i ].std ( )

        Threshold = 3

        Lower_Limit = Mean - Threshold * Standard_Deviation

        Upper_Limit = Mean + Threshold * Standard_Deviation

        DF [ i ] = DF [ i ].apply ( lambda x : Lower_Limit if x <= Lower_Limit else Upper_Limit if x >= Upper_L
```

In [14]:
```
Z_Score ( DF_Train )

Z_Score ( DF_Test )
```

In [15]:
```
DF_Train.describe ( ).transpose ( )
```

Out[15]:

	count	mean	std	min	25%	50%	75%	max
BertzCT	14838.0	511.417496	528.308282	0.000000	149.103601	290.987941	652.652585	2142.522715
Chi1	14838.0	9.078324	6.613727	0.000000	4.680739	6.485270	11.170477	29.595156
Chi1n	14838.0	5.795486	4.427432	0.000000	2.844556	4.052701	7.486791	19.795499
Chi1v	14838.0	6.672461	5.633233	0.000000	2.932842	4.392859	8.527859	24.337829
Chi2n	14838.0	4.392154	3.607416	0.000000	1.949719	2.970427	5.788793	15.714119
Chi2v	14838.0	5.206084	4.759467	0.000000	2.034468	3.242775	6.609350	20.028416
Chi3v	14838.0	3.392345	3.336639	0.000000	1.160763	1.948613	4.502070	13.727374
Chi4n	14838.0	1.747693	1.754839	0.000000	0.503897	1.073261	2.534281	7.371166
EState_VSA1	14838.0	29.002691	30.829391	0.000000	5.969305	17.353601	44.876559	124.388862
EState_VSA2	14838.0	10.279135	13.049905	0.000000	0.000000	6.420822	12.841643	51.390845
ExactMolWt	14838.0	290.878762	219.313544	1.007276	148.037173	206.042653	343.090331	968.775509
FpDensityMorgan1	14838.0	1.280632	0.405176	-15.237077	1.045455	1.250000	1.500000	3.000000
FpDensityMorgan2	14838.0	1.855965	0.399327	-14.674625	1.690909	1.865152	2.062153	3.200000
FpDensityMorgan3	14838.0	2.299409	0.425516	-14.248129	2.100000	2.358491	2.500000	3.400000
HallKierAlpha	14838.0	-1.196311	0.888498	-4.013718	-1.660000	-1.100000	-0.570000	0.820000
HeavyAtomMolWt	14838.0	273.479001	207.580719	0.000000	136.109000	194.276500	326.002000	912.986475
Kappa3	14838.0	4.377263	10.391932	-104.040000	1.784008	3.261011	5.848400	143.065052
MaxAbsEStateIndex	14838.0	10.595936	1.389778	5.878450	9.926190	10.421334	11.539743	15.234435
MinEStateIndex	14838.0	-2.120347	2.064516	-6.327514	-4.659604	-1.265370	-0.787037	4.079472
NumHeteroatoms	14838.0	8.579755	7.629306	0.000000	4.000000	6.000000	10.000000	31.515417
PEOE_VSA10	14838.0	10.931896	13.616671	0.000000	0.000000	6.041841	18.311899	52.898530
PEOE_VSA14	14838.0	16.391249	26.899421	0.000000	0.000000	5.969305	15.645394	121.474975
PEOE_VSA6	14838.0	8.174063	15.490359	0.000000	0.000000	0.000000	12.132734	68.232621
PEOE_VSA7	14838.0	10.534276	16.831084	0.000000	0.000000	0.000000	13.847474	71.828045
PEOE_VSA8	14838.0	6.619965	10.502534	0.000000	0.000000	0.000000	6.923737	39.300731
SMR_VSA10	14838.0	15.576051	17.791540	0.000000	5.969305	11.752550	17.721856	69.907389
SMR_VSA5	14838.0	30.506874	31.215741	0.000000	6.420822	20.075376	42.727765	132.756338
SlogP_VSA3	14838.0	13.593901	14.423500	0.000000	4.794537	9.589074	14.912664	57.432602
VSA_EState9	14838.0	49.041284	27.919335	-5.430556	30.000000	41.666667	56.090650	136.834433
fr_COO	14838.0	0.453443	0.644224	0.000000	0.000000	0.000000	1.000000	2.462059
fr_COO2	14838.0	0.454462	0.644424	0.000000	0.000000	0.000000	1.000000	2.463560
EC1	14838.0	0.667745	0.471038	0.000000	0.000000	1.000000	1.000000	1.000000
EC2	14838.0	0.798962	0.400790	0.000000	1.000000	1.000000	1.000000	1.000000

In [16]:
```
DF_Test.describe ( ).transpose ( )
```

Out[16]:

	count	mean	std	min	25%	50%	75%	max
BertzCT	9893.0	512.705964	530.494470	0.000000	150.255712	289.901774	652.758463	2149.395302
Chi1	9893.0	9.052923	6.560760	0.000000	4.698377	6.447265	10.966946	29.370746
Chi1n	9893.0	5.792710	4.435630	0.000000	2.846050	4.009996	7.490880	19.773032
Chi1v	9893.0	6.670749	5.639170	0.000000	2.934030	4.337841	8.528316	24.325316
Chi2n	9893.0	4.389124	3.612860	0.000000	1.949719	2.930013	5.788793	15.739073
Chi2v	9893.0	5.203858	4.779784	0.000000	2.049137	3.168052	6.516914	20.067098
Chi3v	9893.0	3.376261	3.337394	0.000000	1.171060	1.923982	4.302610	13.696381
Chi4n	9893.0	1.726113	1.736116	0.000000	0.508512	1.058931	2.509394	7.261556
EState_VSA1	9893.0	28.779580	30.678336	0.000000	5.969305	17.282269	44.876559	123.368930
EState_VSA2	9893.0	10.393096	13.223511	0.000000	0.000000	6.420822	12.841643	51.838850
ExactMolWt	9893.0	290.263319	218.239184	15.007276	148.073559	206.021523	342.116212	966.008859
FpDensityMorgan1	9893.0	1.275674	0.339622	0.197230	1.043478	1.250000	1.500000	2.364603
FpDensityMorgan2	9893.0	1.861399	0.343692	0.815842	1.692308	1.866667	2.071429	2.907743
FpDensityMorgan3	9893.0	2.306038	0.376176	1.156442	2.100000	2.358491	2.520000	3.375000
HallKierAlpha	9893.0	-1.196846	0.883010	-3.976300	-1.680000	-1.100000	-0.570000	0.830000
HeavyAtomMolWt	9893.0	273.038445	206.675327	14.007000	140.050000	194.125000	320.121000	911.546262
Kappa3	9893.0	4.212828	9.556738	-104.040000	1.788507	3.261011	5.772640	117.327908
MaxAbsEStateIndex	9893.0	10.599168	1.384582	5.837179	9.946009	10.418624	11.526326	14.630251
MinEStateIndex	9893.0	-2.099104	2.058537	-6.117075	-4.638889	-1.252751	-0.787037	4.080059
NumHeteroatoms	9893.0	8.581624	7.627856	0.000000	4.000000	6.000000	10.000000	31.571555
PEOE_VSA10	9893.0	10.945321	13.559669	0.000000	0.000000	6.041841	18.311899	52.749923
PEOE_VSA14	9893.0	17.039309	28.286091	0.000000	0.000000	5.969305	17.907916	125.989992
PEOE_VSA6	9893.0	8.122102	15.468305	0.000000	0.000000	0.000000	12.132734	66.993831
PEOE_VSA7	9893.0	10.463024	16.641499	0.000000	0.000000	5.563451	13.344559	71.270016
PEOE_VSA8	9893.0	6.711386	10.459398	0.000000	0.000000	0.000000	6.923737	39.123446
SMR_VSA10	9893.0	15.732661	18.019654	0.000000	5.969305	11.752550	17.744066	70.607197
SMR_VSA5	9893.0	30.405221	31.013654	0.000000	6.420822	19.765380	42.727765	131.517845
SlogP_VSA3	9893.0	13.548025	14.466325	0.000000	4.794537	9.589074	14.383612	57.671273
VSA_EState9	9893.0	49.110889	28.040610	-5.061146	30.000000	41.666667	56.083333	138.055028
fr_COO	9893.0	0.454134	0.651384	0.000000	0.000000	0.000000	1.000000	2.489690
fr_COO2	9893.0	0.455673	0.652026	0.000000	0.000000	0.000000	1.000000	2.492816

PREDICTIVE ANALYSIS

RANDOM SAMPLING

In [17]:
```
X = DF_Train.drop( ['EC1' , 'EC2'] , axis = 1 )

Y_EC1 = DF_Train [ 'EC1' ]

Y_EC2 = DF_Train [ 'EC2' ]
```

In [18]:
```
print ( "The value counts are : " )

print ( "\nY_EC1 : \n\n{}\n\nY_EC2 : \n\n{}".format ( Y_EC1.value_counts ( ) , Y_EC2.value_counts ( ) ) )
```

```
The value counts are :

Y_EC1 :

1    9908
0    4930
Name: EC1, dtype: int64

Y_EC2 :

1    11855
0     2983
Name: EC2, dtype: int64
```

FEATURE SCALING

In [19]:
```
Scaler = StandardScaler()

X_Scaled = Scaler.fit_transform ( X )

X = pd.DataFrame ( X_Scaled , columns = list ( X.columns ) )

DF_Test_Scaled = Scaler.transform ( DF_Test )

DF_Test = pd.DataFrame ( DF_Test_Scaled , columns = list ( DF_Test.columns ) )

display ( X )

display ( DF_Test )
```

	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolWt	FpDensity
0	-0.355915	0.121206	0.018090	-0.141466	-0.024228	-0.189382	-0.191167	0.000860	-0.940780	0.127144	-0.313765	
1	-0.449930	-0.275087	-0.305835	-0.148677	-0.306908	-0.151461	-0.356949	-0.260955	0.523310	-0.787706	-0.140667	
2	0.019357	0.277157	0.617167	0.777271	0.630150	0.906355	0.729045	0.013042	-0.433280	-0.281410	0.416095	
3	0.106028	0.510322	0.292196	1.093769	0.578236	1.212796	1.355361	0.751939	2.161545	-0.787706	1.090674	
4	-0.754598	-0.705164	-0.661636	-0.675696	-0.697620	-0.699776	-0.706092	-0.581286	-0.357536	0.196369	-0.788011	
...	...	...	...	...	...	...	...	...	...	...	...	
14833	0.228642	0.277157	0.177185	0.445142	0.072472	0.173128	0.083375	0.013042	0.128740	-0.281410	0.256191	
14834	-0.849626	-0.973098	-0.982225	-0.927662	-0.973765	-0.909079	-0.964394	-0.995962	-0.940780	-0.787706	-0.988821	
14835	0.889492	0.194263	0.079221	-0.093421	0.085499	-0.106215	-0.098156	0.220087	-0.381277	-0.787706	0.028320	
14836	-0.401761	0.131676	0.179404	0.208848	0.245174	0.056805	0.175820	0.314649	0.539150	-0.787706	-0.113629	
14837	0.518610	0.996872	0.926229	0.632305	0.961425	0.696850	0.675300	0.891964	1.733649	-0.351873	0.667360	

14838 rows × 31 columns

	BertzCT	Chi1	Chi1n	Chi1v	Chi2n	Chi2v	Chi3v	Chi4n	EState_VSA1	EState_VSA2	ExactMolWt	FpDensityM
0	-0.315707	-0.271372	-0.298495	-0.148677	-0.271643	-0.116520	-0.388364	-0.359738	0.663541	-0.787706	-0.395789	
1	1.743345	0.239744	0.289916	0.247247	0.250865	0.019123	0.159390	0.468435	-0.940780	1.494624	0.110948	
2	-0.810283	-0.778176	-0.908293	-0.869556	-0.920000	-0.868328	-0.876515	-0.898606	-0.747149	-0.295668	-0.865599	
3	-0.683642	-0.478647	-0.507461	-0.554522	-0.498165	-0.548600	-0.524357	-0.600406	-0.940780	-0.787706	-0.473143	
4	2.471858	2.393983	2.201134	2.374357	2.257574	2.631666	2.573716	1.529240	1.801987	2.841578	2.737219	
...	...	...	...	...	...	...	...	...	...	...	...	
9888	-0.501608	-0.762341	-0.672823	-0.684488	-0.697620	-0.699776	-0.646289	-0.789245	-0.157522	-0.295668	-0.660142	
9889	0.150774	-0.046491	-0.025529	-0.175749	-0.094729	-0.242818	-0.366996	0.013042	-0.940780	1.375488	0.060926	
9890	-0.252331	-0.418534	-0.540547	-0.418022	-0.436540	-0.238292	-0.426225	-0.331635	0.249876	-0.787706	-0.140667	
9891	0.428242	0.131676	0.348400	0.118140	0.010079	0.155963	0.173867	0.235464	0.259156	-0.787706	0.105741	
9892	0.518610	0.467556	0.399771	0.528896	0.402079	0.635264	0.681053	0.478631	0.663541	0.067799	0.260672	

9893 rows × 31 columns

LOGISTIC REGRESSION

In [20]:
```
LR = LogisticRegression (  )

LR1 = LR.fit ( X , Y_EC1 )

Result = pd.DataFrame ( )

Result [ 'EC1_Predicted' ] = LR1.predict ( DF_Test )

LR2 = LR.fit ( X , Y_EC2 )

Result [ 'EC2_Predicted' ] = LR2.predict ( DF_Test )

display ( Result.EC1_Predicted.value_counts ( ) )

display ( Result.EC2_Predicted.value_counts ( ) )

Result
```

```
1    8262
0    1631
Name: EC1_Predicted, dtype: int64
1    9891
0       2
Name: EC2_Predicted, dtype: int64
```

Out[20]:

	EC1_Predicted	EC2_Predicted
0	0	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
9888	1	1
9889	1	1
9890	1	1
9891	1	1
9892	0	1

9893 rows × 2 columns

DECISION TREE

In [21]:
```
DT = DecisionTreeClassifier (  )

Parameters = { 'criterion' : [ 'gini' , 'entropy' ],

              'splitter' : [ 'best' ,'random' ],

              'max_depth' : [ None , 5 , 10 ] }

GS = GridSearchCV ( estimator = DT , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC1 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

DT = DecisionTreeClassifier ( **Best_Parameters )

DT.fit ( X , Y_EC1 )

DT_Result = pd.DataFrame (  )

DT_Result [ 'EC1_Predicted' ] = DT.predict ( DF_Test )

display ( DT_Result.EC1_Predicted.value_counts ( ) )

DT_Result
```

```
The best parameters are as follows :

 {'criterion': 'gini', 'max_depth': 5, 'splitter': 'best'}
1    7639
0    2254
Name: EC1_Predicted, dtype: int64
```

Out[21]:

	EC1_Predicted
0	1
1	1
2	1
3	1
4	1
...	...
9888	1
9889	1
9890	0
9891	0
9892	0

9893 rows × 1 columns

In [22]:
```
DT = DecisionTreeClassifier (  )

Parameters = { 'criterion' : [ 'gini' , 'entropy' ],

              'splitter' : [ 'best' ,'random' ],

              'max_depth' : [ None , 5 , 10 ] }

GS = GridSearchCV ( estimator = DT , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC2 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

DT = DecisionTreeClassifier ( **Best_Parameters )

DT.fit ( X , Y_EC2 )

DT_Result [ 'EC2_Predicted' ] = DT.predict ( DF_Test )

display ( DT_Result.EC2_Predicted.value_counts ( ) )

DT_Result
```

```
The best parameters are as follows :

 {'criterion': 'gini', 'max_depth': 5, 'splitter': 'random'}
1    9883
0      10
Name: EC2_Predicted, dtype: int64
```

Out[22]:

	EC1_Predicted	EC2_Predicted
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
9888	1	1
9889	1	1
9890	0	1
9891	0	1
9892	0	1

9893 rows × 2 columns

RANDOM FOREST

In [23]:
```
RF = RandomForestClassifier (  )

Parameters = { 'criterion' : [ 'gini' , 'entropy' ],

              'n_estimators' : [ 100 , 200 , 300 ],

              'max_depth' : [ None , 5 , 10 ] }

GS = GridSearchCV ( estimator = RF , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC1 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

RF = RandomForestClassifier ( **Best_Parameters )

RF.fit ( X , Y_EC1 )

RF_Result = pd.DataFrame (  )

RF_Result [ 'EC1_Predicted' ] = RF.predict ( DF_Test )

display ( RF_Result.EC1_Predicted.value_counts ( ) )

RF_Result
```

```
The best parameters are as follows :

 {'criterion': 'entropy', 'max_depth': 10, 'n_estimators': 200}
1    7938
0    1955
Name: EC1_Predicted, dtype: int64
```

Out[23]:

	EC1_Predicted
0	0
1	1
2	1
3	1
4	1
...	...
9888	1
9889	1
9890	0
9891	0
9892	0

9893 rows × 1 columns

In [24]:
```
RF = RandomForestClassifier (  )

Parameters = { 'criterion' : [ 'gini' , 'entropy' ],

              'n_estimators' : [ 50 , 100 ],

              'max_depth' : [ None , 5 , 10 ] }

GS = GridSearchCV ( estimator = RF , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC2 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

RF = RandomForestClassifier ( **Best_Parameters )

RF.fit ( X , Y_EC2 )

RF_Result [ 'EC2_Predicted' ] = RF.predict ( DF_Test )

display ( RF_Result.EC2_Predicted.value_counts ( ) )

RF_Result
```

```
The best parameters are as follows :

 {'criterion': 'gini', 'max_depth': 10, 'n_estimators': 50}
1    9893
Name: EC2_Predicted, dtype: int64
```

Out[24]:

	EC1_Predicted	EC2_Predicted
0	0	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
9888	1	1
9889	1	1
9890	0	1
9891	0	1
9892	0	1

9893 rows × 2 columns

GRADIENT BOOSTING

In [26]:
```
GB = GradientBoostingClassifier (  )

Parameters = { 'learning_rate' : [ 0.1 , 0.001 ] ,

               'n_estimators' : [ 100 , 200 ]  }

GS = GridSearchCV ( estimator = GB , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC1 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

GB = GradientBoostingClassifier ( **Best_Parameters )

GB.fit ( X , Y_EC1 )

GB_Result = pd.DataFrame (  )

GB_Result [ 'EC1_Predicted' ] = GB.predict ( DF_Test )

display ( GB_Result.EC1_Predicted.value_counts ( ) )

GB_Result
```

```
The best parameters are as follows :

 {'learning_rate': 0.1, 'n_estimators': 100}
1    7817
0    2076
Name: EC1_Predicted, dtype: int64
```

Out[26]:

	EC1_Predicted
0	0
1	1
2	1
3	1
4	1
...	...
9888	1
9889	1
9890	0
9891	0
9892	0

9893 rows × 1 columns

In [28]:
```
GB = GradientBoostingClassifier (  )

Parameters = { 'learning_rate' : [ 0.1 , 0.01 ] ,

               'n_estimators' : [ 100 , 200 ]  }

GS = GridSearchCV ( estimator = GB , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC2 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

GB = GradientBoostingClassifier ( **Best_Parameters )

GB.fit ( X , Y_EC2 )

GB_Result [ 'EC2_Predicted' ] = GB.predict ( DF_Test )

display ( GB_Result.EC2_Predicted.value_counts ( ) )

GB_Result
```

```
The best parameters are as follows :

 {'learning_rate': 0.01, 'n_estimators': 100}
1    9893
Name: EC2_Predicted, dtype: int64
```

Out[28]:

	EC1_Predicted	EC2_Predicted
0	0	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
9888	1	1
9889	1	1
9890	0	1
9891	0	1
9892	0	1

9893 rows × 2 columns

EXTREME GRADIENT BOOSTING

In [30]:
```
from xgboost import XGBClassifier

XGB = XGBClassifier ( random_state = 42 )

Parameters = { 'n_estimators': [ 100, 200 ],

               'learning_rate': [ 0.5 , 0.001],

               'max_depth': [3, 5 ]  }

GS = GridSearchCV ( estimator = XGB , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC1 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

XGB = GradientBoostingClassifier ( **Best_Parameters )

XGB.fit ( X , Y_EC1 )

XGB_Result = pd.DataFrame (  )

XGB_Result [ 'EC1_Predicted' ] = XGB.predict ( DF_Test )

display ( XGB_Result.EC1_Predicted.value_counts ( ) )

XGB_Result
```

```
The best parameters are as follows :

 {'learning_rate': 0.001, 'max_depth': 5, 'n_estimators': 200}
1    9893
Name: EC1_Predicted, dtype: int64
```

Out[30]:

	EC1_Predicted
0	1
1	1
2	1
3	1
4	1
...	...
9888	1
9889	1
9890	1
9891	1
9892	1

9893 rows × 1 columns

In [32]:
```
XGB =  XGBClassifier( random_state = 42 )

Parameters = { 'n_estimators': [ 100, 200 ],

               'learning_rate': [ 0.5 , 0.001],

               'max_depth': [3, 5 ]  }

GS = GridSearchCV ( estimator = XGB , param_grid = Parameters , cv = 5 )

GS.fit ( X , Y_EC2 )

Best_Parameters = GS.best_params_

print ( "The best parameters are as follows : \n\n" , Best_Parameters )

XGB = XGBClassifier ( **Best_Parameters )

XGB.fit ( X , Y_EC2 )

XGB_Result [ 'EC2_Predicted' ] = XGB.predict ( DF_Test )

display ( XGB_Result.EC2_Predicted.value_counts ( ) )

XGB_Result
```

```
The best parameters are as follows :

 {'learning_rate': 0.001, 'max_depth': 3, 'n_estimators': 200}
1    9893
Name: EC2_Predicted, dtype: int64
```

Out[32]:

	EC1_Predicted	EC2_Predicted
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
9888	1	1
9889	1	1
9890	1	1
9891	1	1
9892	1	1

9893 rows × 2 columns

In [ ]: