In [2]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
```

In [3]:
```
hea = pd.read_csv("C:\\Users\Ritik\OneDrive - Indian Institute of Technology (BHU)
hea.head(5)
```

Out[3]:

	Alloy ID	Alloy	Al	Co	Cr	Fe	Ni	Cu	Mn	Ti	...	Annealing_Temp	Ann
0	Alloy 0000	Al0.5NbTaTiV	0.111111	0.0	0.0	0.0	0.0	0.0	0.0	0.222222	...	NaN	
1	Alloy 0001	Al0.75MoNbTiV	0.157895	0.0	0.0	0.0	0.0	0.0	0.0	0.210526	...	NaN	
2	Alloy 0002	Al0.25MoNbTiV	0.058824	0.0	0.0	0.0	0.0	0.0	0.0	0.235294	...	NaN	
3	Alloy 0003	Al0.25NbTaTiV	0.058824	0.0	0.0	0.0	0.0	0.0	0.0	0.235294	...	NaN	
4	Alloy 0004	Al0.2MoTaTiV	0.047619	0.0	0.0	0.0	0.0	0.0	0.0	0.238095	...	NaN	

5 rows × 51 columns

In [4]:
```
hea.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1360 entries, 0 to 1359
Data columns (total 51 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Alloy ID              1360 non-null   object 
 1   Alloy                 1360 non-null   object 
 2   Al                    1360 non-null   float64
 3   Co                    1360 non-null   float64
 4   Cr                    1360 non-null   float64
 5   Fe                    1360 non-null   float64
 6   Ni                    1360 non-null   float64
 7   Cu                    1360 non-null   float64
 8   Mn                    1360 non-null   float64
 9   Ti                    1360 non-null   float64
 10  V                     1360 non-null   float64
 11  Nb                    1360 non-null   float64
 12  Mo                    1360 non-null   float64
 13  Zr                    1360 non-null   float64
 14  Hf                    1360 non-null   float64
 15  Ta                    1360 non-null   float64
 16  W                     1360 non-null   float64
 17  C                     1360 non-null   float64
 18  Mg                    1360 non-null   float64
 19  Zn                    1360 non-null   float64
 20  Si                    1360 non-null   float64
 21  Re                    1360 non-null   float64
 22  N                     1360 non-null   float64
 23  Sc                    1360 non-null   int64  
 24  Li                    1360 non-null   float64
 25  Sn                    1360 non-null   float64
 26  Be                    1360 non-null   int64  
 27  Num_of_Elem           1360 non-null   int64  
 28  Density_calc          1355 non-null   float64
 29  dHmix                 1360 non-null   float64
 30  dSmix                 1360 non-null   float64
 31  dGmix                 1360 non-null   float64
 32  Tm                    1360 non-null   float64
 33  n.Para                1359 non-null   float64
 34  Atom.Size.Diff        1360 non-null   float64
 35  Elect.Diff            1360 non-null   float64
 36  VEC                   1360 non-null   float64
 37  Sythesis_Route        1360 non-null   object 
 38  Hot-Cold_Working      444 non-null    object 
 39  Homogenization_Temp   571 non-null    float64
 40  Homogenization_Time   561 non-null    float64
 41  Annealing_Temp        603 non-null    object 
 42  Annealing_Time_(min)  594 non-null    float64
 43  Quenching             269 non-null    object 
 44  HPR                   56 non-null     object 
 45  Microstructure_       1360 non-null   object 
 46  Multiphase            1355 non-null   object 
 47  IM_Structure          432 non-null    object 
 48  Microstructure        1360 non-null   object 
 49  Phases                1360 non-null   object 
 50  References            1360 non-null   object 
dtypes: float64(35), int64(3), object(13)
memory usage: 542.0+ KB
```

In [5]:
```
hea.drop(["Hot-Cold_Working", "Homogenization_Temp", "Homogenization_Time", "Annea
```

In [6]:
```
hea.corr()
```

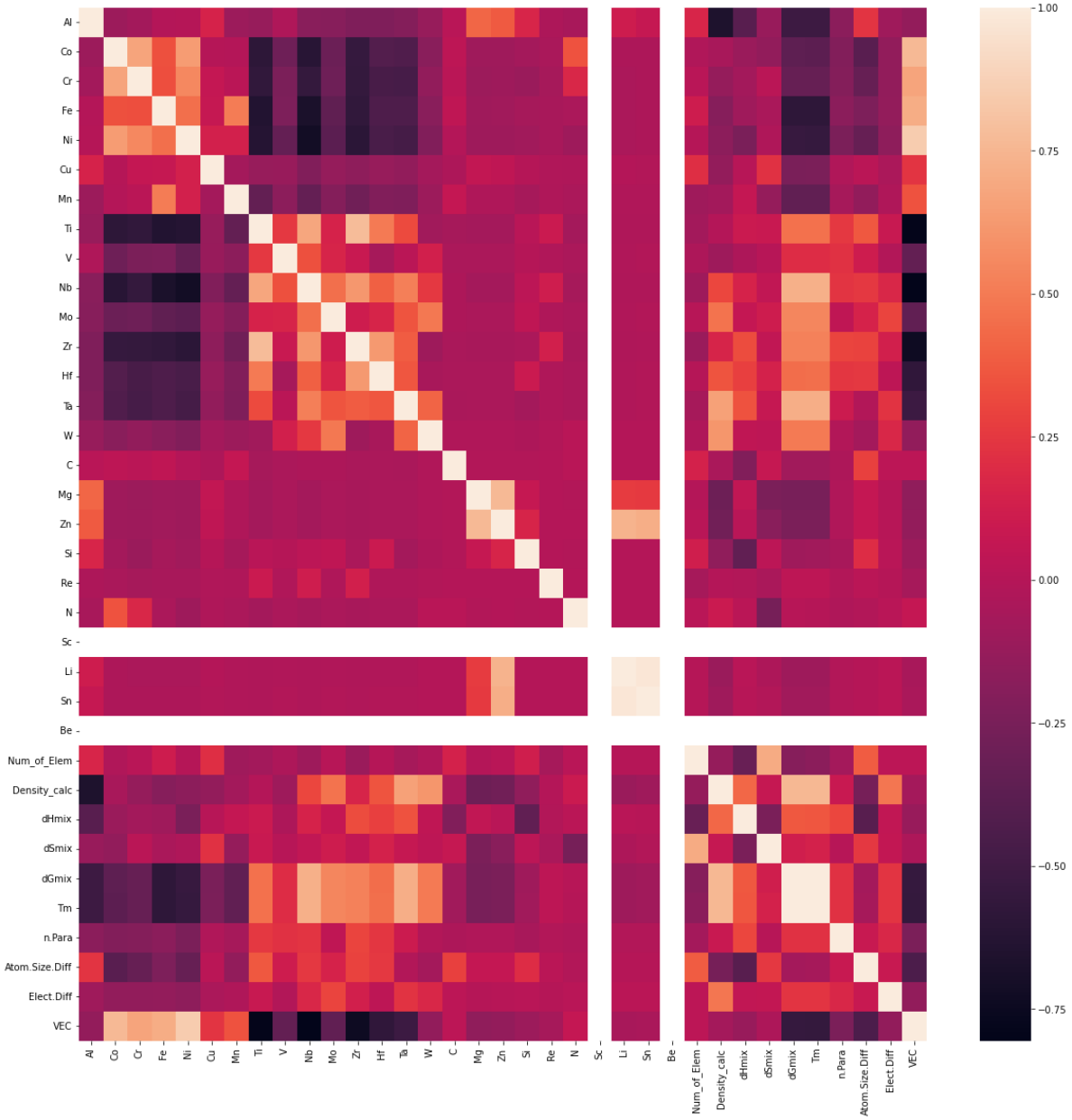
Out[6]:

	Al	Co	Cr	Fe	Ni	Cu	Mn	T
Al	1.000000	-0.106613	-0.075731	-0.003295	0.001423	0.153287	-0.103176	-0.12290
Co	-0.106613	1.000000	0.671023	0.340902	0.639405	-0.001393	-0.007302	-0.59271
Cr	-0.075731	0.671023	1.000000	0.333622	0.551724	0.063945	0.020042	-0.56581
Fe	-0.003295	0.340902	0.333622	1.000000	0.454811	0.073410	0.499551	-0.63865
Ni	0.001423	0.639405	0.551724	0.454811	1.000000	0.132956	0.132578	-0.62675
Cu	0.153287	-0.001393	0.063945	0.073410	0.132956	1.000000	-0.066844	-0.12310
Mn	-0.103176	-0.007302	0.020042	0.499551	0.132578	-0.066844	1.000000	-0.34614
Ti	-0.122905	-0.592717	-0.565814	-0.638655	-0.626758	-0.123103	-0.346147	1.00000
V	-0.023904	-0.295680	-0.234436	-0.229267	-0.337782	-0.119738	-0.164174	0.25237
Nb	-0.180460	-0.614753	-0.555781	-0.672615	-0.719121	-0.214188	-0.336851	0.67671
Mo	-0.185502	-0.305761	-0.292122	-0.357632	-0.381125	-0.133263	-0.203447	0.14781
Zr	-0.222381	-0.546496	-0.553202	-0.569906	-0.596499	-0.163159	-0.283754	0.77871
Hf	-0.220642	-0.415681	-0.466434	-0.432047	-0.455074	-0.124312	-0.215672	0.49600
Ta	-0.201105	-0.430069	-0.467288	-0.438006	-0.474045	-0.144231	-0.220244	0.32214
W	-0.123663	-0.178559	-0.145675	-0.182975	-0.214086	-0.067145	-0.102531	-0.07999
C	0.013720	0.038243	0.022169	0.046254	-0.004096	-0.032855	0.062405	-0.06463
Mg	0.415349	-0.088405	-0.100301	-0.088625	-0.096775	0.057653	-0.023163	-0.06987
Zn	0.375716	-0.086148	-0.097110	-0.085353	-0.094957	0.043804	-0.029536	-0.06808
Si	0.165690	-0.068039	-0.107736	-0.062490	-0.084836	0.004064	-0.062409	0.01541
Re	-0.031602	-0.050720	-0.058455	-0.052303	-0.055906	-0.017010	-0.025974	0.09543
N	-0.052708	0.344500	0.169282	-0.042019	-0.093245	-0.028370	-0.042609	-0.06686
Sc	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
Li	0.108265	-0.035838	-0.041303	-0.036956	-0.039502	-0.005579	-0.018353	-0.02832
Sn	0.070426	-0.030725	-0.035410	-0.031684	-0.033866	-0.010304	-0.015734	-0.02428
Be	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
Num_of_Elem	0.162035	-0.016855	0.018389	0.106226	0.004995	0.208335	-0.089553	-0.07477
Density_calc	-0.660639	-0.052177	-0.134119	-0.196460	-0.170223	-0.141393	-0.076704	-0.00703
dHmix	-0.394763	-0.109673	-0.078287	-0.086898	-0.242074	0.009566	0.063935	0.09261
dSmix	-0.120716	-0.142931	0.021219	-0.047423	-0.028377	0.220454	-0.134256	0.08588
dGmix	-0.512600	-0.363551	-0.326586	-0.581383	-0.544188	-0.245331	-0.351174	0.45750
Tm	-0.510353	-0.368474	-0.327436	-0.581854	-0.545462	-0.239465	-0.353351	0.45870
n.Para	-0.172708	-0.208215	-0.204755	-0.176140	-0.234820	-0.018734	-0.059201	0.24814
Atom.Size.Diff	0.233805	-0.389222	-0.329589	-0.229042	-0.326863	0.020308	-0.144174	0.36786
Elect.Diff	-0.090902	-0.143713	-0.144202	-0.141632	-0.161087	-0.042298	-0.028063	0.08038
VEC	-0.137902	0.761811	0.672652	0.709095	0.848656	0.237102	0.346421	-0.79795

35 rows × 35 columns

In [7]:
```
plt.figure(figsize = (20,20))
sns.heatmap(hea.corr() )
```

Out[7]:
```
<AxesSubplot:>
```



In [8]:
```
hea.drop(["Be", "Sc"] , axis = 1 , inplace = True)
```

In [9]:
```
hea.columns
```

Out[9]:
```
Index(['Alloy ID', 'Alloy ', 'Al', 'Co', 'Cr', 'Fe', 'Ni', 'Cu', 'Mn', 'Ti',
       'V', 'Nb', 'Mo', 'Zr', 'Hf', 'Ta', 'W', 'C', 'Mg', 'Zn', 'Si', 'Re',
       'N', 'Li', 'Sn', 'Num_of_Elem', 'Density_calc', 'dHmix', 'dSmix',
       'dGmix', 'Tm', 'n.Para', 'Atom.Size.Diff', 'Elect.Diff', 'VEC',
       'Sythesis_Route', 'IM_Structure', 'Microstructure', 'Phases'],
      dtype='object')
```

In [10]:
```
hea.drop(["Sythesis_Route", "IM_Structure"] , axis = 1 , inplace = True)
```

In [11]:
```
hea["Microstructure"].describe()
```

Out[11]:
```
count     1360
unique       7
top        BCC
freq       441
Name: Microstructure, dtype: object
```

In [12]:
```
hea["Microstructure"].value_counts()
```

Out[12]:
```
BCC               441
FCC               354
FCC + Im          231
BCC + Im          179
FCC + BCC         102
FCC + BCC + Im     47
Im                  6
Name: Microstructure, dtype: int64
```

In [13]:
```
labels = hea["Phases"]
```

In [14]:
```
labels.describe()
```

Out[14]:
```
count     1360
unique       4
top         Im
freq       463
Name: Phases, dtype: object
```

In [15]:
```
hea.drop(["Alloy ID", "Alloy "], axis = 1 , inplace = True)
```

In [16]:
```
hea.shape
```

Out[16]:
```
(1360, 35)
```

In [17]:
```
para_median = hea['n.Para'].median()
hea["n.Para"].fillna(para_median ,inplace = True)
```

In [18]:
```
dens_med = hea["Density_calc"].median()
hea["Density_calc"].fillna(dens_med , inplace = True)
```

In [19]:
```
hea.drop(["Phases", "Microstructure"], axis = 1 , inplace = True)
```

In [20]:
```
from sklearn.model_selection import train_test_split
train_hea, test_hea , train_labels , test_labels = train_test_split(hea , labels,
```

In [21]:
```
train_hea.shape
```

Out[21]:
```
(1020, 33)
```

In [22]:
```
from sklearn.linear_model import LogisticRegression
pipe = make_pipeline(StandardScaler(), LogisticRegression())
pipe.fit(train_hea,train_labels)
pipe.score(test_hea,test_labels)
```

```
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[22]:
```
0.7529411764705882
```

In [23]:
```
from sklearn.ensemble import RandomForestClassifier
pipe2 = make_pipeline(StandardScaler(), RandomForestClassifier())
pipe2.fit(train_hea, train_labels)
pipe2.score(test_hea, test_labels)
```

Out[23]:
```
0.8529411764705882
```

In [24]:
```
from sklearn.tree import DecisionTreeClassifier
pipe3 = make_pipeline(StandardScaler(), DecisionTreeClassifier())
pipe3.fit(train_hea, train_labels)
pipe3.score(test_hea, test_labels)
```

Out[24]:
```
0.8352941176470589
```

In [25]:
```
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.svm import SVC
svc_model = SVC(C= .1, kernel='linear', gamma= 1)
svc_model.fit(train_hea,train_labels)
prediction = svc_model.predict(test_hea)

print(svc_model.score(train_hea,train_labels))
print(svc_model.score(test_hea, test_labels))
```

```
0.7137254901960784
0.7088235294117647
```

In [26]:
```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
rfe = RFE(model)
fit = rfe.fit(hea, labels)

print("Num Features: %d"% fit.n_features_)
print("Selected Features: %s"% fit.support_)
print("Feature Ranking: %s"% fit.ranking_)
```

```
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
```

```
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
```

```
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Num Features: 16
Selected Features: [False  True  True  True  True False  True  True False  True Fa
lse False
 False False False False False False False False False False False  True
  True  True  True  True  True  True False  True]
Feature Ranking: [ 3  1  1  1  1  7  1  1  6  1  8  2  5  4  9 13 11 14 12 15 18 1
6 17  1
  1  1  1  1  1  1  1 10  1]
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [27]:
```
hea.drop(['Cu' , 'Zr', 'Ta', 'Zn','Si' ,'Re','N' ,'Li', 'Sn','Num_of_Elem' ,'Densi
```

In [28]:
```
from sklearn.model_selection import train_test_split
train_hea, test_hea , train_labels , test_labels = train_test_split(hea , labels,
```

In [29]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
pipe = make_pipeline(StandardScaler(), LogisticRegression())
pipe.fit(train_hea,train_labels)
pipe.score(test_hea,test_labels)
```

```
C:\Users\Ritik\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:814:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[29]:
```
0.7323529411764705
```

In [30]:
```
from sklearn.tree import DecisionTreeClassifier
pipe3 = make_pipeline(StandardScaler(), DecisionTreeClassifier())
pipe3.fit(train_hea, train_labels)
pipe3.score(test_hea, test_labels)
```

Out[30]:
```
0.8235294117647058
```

In [31]:
```
from sklearn.ensemble import RandomForestClassifier
pipe2 = make_pipeline(StandardScaler(), RandomForestClassifier())
pipe2.fit(train_hea, train_labels)
pipe2.score(test_hea, test_labels)
```

Out[31]:
```
0.8441176470588235
```

In [32]:
```
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
svc_model = SVC(C= .1, kernel='linear', gamma= 1)
svc_model.fit(train_hea,train_labels)
prediction = svc_model.predict(test_hea)

print(svc_model.score(train_hea,train_labels))
print(svc_model.score(test_hea, test_labels))
```

```
0.703921568627451
0.711764705882353
```