In [23]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score,f1_score

import warnings
warnings.filterwarnings(action='ignore')
```

In [24]:
```
data=pd.read_csv('data-ori.csv')
data.head()
```

Out[24]:

	HAEMATOCRIT	HAEMOGLOBINS	ERYTHROCYTE	LEUCOCYTE	THROMBOCYTE	MCH	MCHC	MCV	AGE	SEX	SOURCE
0	35.1	11.8	4.65	6.3	310	25.4	33.6	75.5	1	F	out
1	43.5	14.8	5.39	12.7	334	27.5	34.0	80.7	1	F	out
2	33.5	11.3	4.74	13.2	305	23.8	33.7	70.7	1	F	out
3	39.1	13.7	4.98	10.5	366	27.5	35.0	78.5	1	F	out
4	30.9	9.9	4.23	22.1	333	23.4	32.0	73.0	1	M	out

In [25]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4412 entries, 0 to 4411
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   HAEMATOCRIT   4412 non-null   float64
 1   HAEMOGLOBINS  4412 non-null   float64
 2   ERYTHROCYTE   4412 non-null   float64
 3   LEUCOCYTE     4412 non-null   float64
 4   THROMBOCYTE   4412 non-null   int64  
 5   MCH           4412 non-null   float64
 6   MCHC          4412 non-null   float64
 7   MCV           4412 non-null   float64
 8   AGE           4412 non-null   int64  
 9   SEX           4412 non-null   object 
 10  SOURCE        4412 non-null   object 
dtypes: float64(7), int64(2), object(2)
memory usage: 379.3+ KB
```

In [26]:
```
data.isna().sum()
```

Out[26]:
```
HAEMATOCRIT     0
HAEMOGLOBINS    0
ERYTHROCYTE     0
LEUCOCYTE       0
THROMBOCYTE     0
MCH             0
MCHC            0
MCV             0
AGE             0
SEX             0
SOURCE          0
dtype: int64
```

Preprocessing

In [35]:
```
def preprocess_input(df):
    df=df.copy()

    #Binary Encoding
    df['SEX']=df['SEX'].replace({'F':0, 'M':1})
```

```
    #split df into X and y
    y=df['SOURCE']
    X=df.drop('SOURCE',axis=1)

    #Train_test_split
    X_train,X_test,y_train,y_test =train_test_split(X,y,train_size=0.7,shuffle=True,random_state=1)


    #Scale X
    scaler= StandardScaler()
    scaler.fit(X_train)
    X_train=pd.DataFrame(scaler.transform(X_train),index=X_train.index, columns=X_train.columns)
    X_test=pd.DataFrame(scaler.transform(X_test),index=X_test.index, columns=X_test.columns)

    return X_train,X_test,y_train,y_test
```

In [36]:
```
X_train,X_test,y_train,y_test=preprocess_input(data)
```

In [37]:
```
X_train
```

Out[37]:

	HAEMATOCRIT	HAEMOGLOBINS	ERYTHROCYTE	LEUCOCYTE	THROMBOCYTE	MCH	MCHC	MCV	AGE	SEX
2275	1.521571	0.918324	4.205082	-0.507635	-0.368181	-3.543672	-1.886871	-3.523787	0.113088	-1.043023
4093	-0.590225	-0.613524	-0.673427	-0.468456	-0.184678	0.106741	-0.267275	0.263954	1.411455	-1.043023
1727	-1.512041	-1.618799	-1.463160	0.099634	1.195966	-0.228501	-0.996093	0.191392	-0.350615	-1.043023
615	0.817639	0.870454	0.765924	-0.488046	-0.140986	0.032243	0.380563	-0.127881	-1.138910	0.958752
1610	0.348351	0.391752	-0.036546	-0.311742	0.260973	0.665478	0.299583	0.670302	-0.443356	-1.043023
...	...	...	...	...	...	...	...	...	...	...
2895	0.029906	0.056660	-0.240348	-0.781885	0.531859	0.479233	0.137624	0.510665	0.576790	-1.043023
2763	-0.003615	-0.278432	-0.533314	-0.703528	0.182329	0.479233	-1.158053	1.149211	0.484050	-1.043023
905	1.320448	1.444897	0.753186	-0.488046	0.907604	0.926222	0.704482	0.742864	-0.953429	0.958752
3980	-0.539944	-0.565653	-0.558789	-0.703528	-0.420611	-0.005006	-0.267275	0.089805	1.318715	0.958752
235	-0.288540	-0.230562	-0.011071	0.334706	0.732839	-0.451995	0.218603	-0.606791	-1.741723	-1.043023

3088 rows × 10 columns

In [38]:
```
y_train
```

Out[38]:
```
2275    out
4093     in
1727     in
615     out
1610    out
       ...
2895    out
2763    out
905     out
3980    out
235     out
Name: SOURCE, Length: 3088, dtype: object
```

In [39]:
```
y_train.value_counts()
```

Out[39]:
```
out    1834
in     1254
Name: SOURCE, dtype: int64
```

Training

In [40]:
```
models = {
    "Logistic Regression": LogisticRegression(),
    "      Decision Tree": DecisionTreeClassifier(),
    "     Neural Network": MLPClassifier(),
    "      Random Forest": RandomForestClassifier(),
    "  Gradient Boosting": GradientBoostingClassifier()
}

for name, model in models.items():
    model.fit(X_train, y_train)
    print(name + " trained.")
```

```
Logistic Regression trained.
      Decision Tree trained.
     Neural Network trained.
      Random Forest trained.
  Gradient Boosting trained.
```

Result

In [41]:
```
for name,model in models.items():
    y_pred = model.predict(X_test)
    acc=accuracy_score(y_test,y_pred)
    print(name+ "Accuracy: {:.2f}%".format(acc*100))
```

```
Logistic RegressionAccuracy: 71.15%
      Decision TreeAccuracy: 66.77%
     Neural NetworkAccuracy: 74.17%
      Random ForestAccuracy: 74.02%
  Gradient BoostingAccuracy: 73.64%
```

In [42]:
```
for name,model in models.items():
    y_pred = model.predict(X_test)
```

```
    f1=f1_score(y_test,y_pred,pos_label='in')
    print(name+ "F1-Score: {:.5f}%".format(f1))
```

```
Logistic RegressionF1-Score: 0.59705%
      Decision TreeF1-Score: 0.59259%
     Neural NetworkF1-Score: 0.66732%
      Random ForestF1-Score: 0.66208%
  Gradient BoostingF1-Score: 0.64783%
```

In [ ]: