In [2]:
```
import joblib
import pandas as pd
from sklearn.metrics import f1_score
```

Function-1

In [3]:
```
def final_fun_1( x ):
    """
    This function performs data preprocessing
    and makes final predictions and
    returns the class labels
    """

    # Load preprocessing models
    median_imputer = joblib.load("../median_imputer.pkl")
    mice_imputer = joblib.load("../mice_imputer.pkl")

    # Drop features with max null values
    x = x.drop(['br_000', 'bq_000', 'bp_000', 'bo_000', 'ab_000', 'cr_000', 'bn_000', 'cd_000'] , axis=1)

    # Specify features whose missing values are imputed using Median Imputer
    median_features = ['ak_000','ca_000','dm_000','df_000','dg_000', \
                       'dh_000','dl_000','dj_000','dk_000','eb_000', \
                       'di_000','ac_000','bx_000','cc_000']

    # Median Imputation
    x[median_features] = median_imputer.transform(x[median_features])

    # MICE Imputation
    x = pd.DataFrame(data = mice_imputer.transform(x) , columns= x.columns )

    # Load our Best Model
    model = joblib.load("../gbdt_model.pkl")

    # Predict class label
    y = model.predict(x)

    return y
```

Function-2

In [11]:
```
def final_fun_2( x , y ):
    """
    This function performs data preprocessing,
    makes final predictions and returns the computed
    performance metric and class labels
    """

    # Load preprocessing models
    median_imputer = joblib.load("../median_imputer.pkl")
    mice_imputer = joblib.load("../mice_imputer.pkl")

    # Drop features with max null values
    x = x.drop(['br_000', 'bq_000', 'bp_000', 'bo_000', 'ab_000', 'cr_000', 'bn_000', 'cd_000'] , axis=1)

    # Specify features whose missing values are imputed using Median Imputer
    median_features = ['ak_000','ca_000','dm_000','df_000','dg_000', \
                       'dh_000','dl_000','dj_000','dk_000','eb_000', \
                       'di_000','ac_000','bx_000','cc_000']

    # Median Imputation
    x[median_features] = median_imputer.transform(x[median_features])

    # MICE Imputation
    x = pd.DataFrame(data = mice_imputer.transform(x) , columns= x.columns )

    # Load our Best Model
    model = joblib.load("../gbdt_model.pkl")

    # Predict Class Labels
    y_pred = model.predict(x)

    # Return Performance Metric
    return f1_score(y,y_pred,average='macro') , y_pred
```

Testing both functions on test dataset

In [12]:
```
data = pd.read_csv("../aps_failure_test_set.csv",skiprows=20,na_values=["na"])

def get_correct_label(y):
    """
    This function converts the class labels
    from 'neg' and 'pos' to 0 and 1 respectively
    """
    return y.replace(['neg','pos'],[0,1])

print(data['class'].unique())
data['class'] = get_correct_label(data['class'])
print(data['class'].unique())

y_test = data['class']
x_test = data.drop('class',axis=1)
```

```
['neg' 'pos']
[0 1]
```

In [13]:
```
F1_test , y_pred = final_fun_2(x_test,y_test)
print("Macro-F1 Score: ",F1_test)
```

```
Macro-F1 Score:  0.8817895552269138
```