## *DATA PRE-PROCESSING*

#import required labraries

0.0

0.0

67

37

85

22

4.

19

04

80

94

3 3

0

44

import pandas as pd import numpy as np from sklearn.preprocessing import MinMaxScaler from sklearn.metrics import confusion matrix, accuracy score In [48]: import matplotlib.pyplot as plt plt.style.use('ggplot') %matplotlib inline In [49]: #read the data set dataset train=pd.read csv(r"/content/Dataset-20221111T140216Z-001.zip",sep=' ', header=None).drop([26,27],axis=1) col names=['id', 'cycle', 'setting1','setting2','setting3','s1','s2','s3','s4','s5','s6','s7','s8','s 9','s10','s11','s12','s13','s14','s15','s16','s17','s18','s19','s20','s21'] dataset train.columns=col names print ('Shape of Train dataset: ',dataset train.shape) dataset train.head() Shape of Train dataset: (20631, 26)Out[49]:  $\mathbf{c}$ set set set i s1s1s2s1s1tin tin tin s1s25 d cl 3 4 1 g1 **g2** g323. 51 64 15 14 52 23 81 8. 3 0.0 0.0 10 3 9. 4. 0 0 1 89. 00. 88. 38. 41 41 8. 1. 1. 6 0 8 0. 00 00 0.0 67 82 70 60 66 02 62 95 2 90 3 0 0.0 51 64 15 14 52 23 81 8. 3 23. 10 9. 0.0 4. 3 0 2 2. 91. 03. 2. 88. 31. 43 9 1 1 01 42 8. 0.0 6 0 8 0. 0 00 67 15 82 14 28 07 18 2 36 3 3 0 2 1 0 1 14 52 23 81 8. 3 23. 0.0 51 64 15 10 0.0 3 0 8. **2** 1 3 00 8. 2. 87. 04. 2. 88. 33. 41 9 34 6 0 8 0. 9 04 0.0 42 0 3 67 35 99 20 03 23 78 42 3 3 2 1 0 0.0 0.0 51 15 14 52 23 81 8. 3 23. 64 10 4. 3 0 8. **3** 1 00 00 8. 2. 82. 01. 2. 88. 33. 36 37 6 0 0. 0.0 67 35 79 87 86 08 83 82 2 39 3 51 64 15 14 52 23 81 8. 3 23. 10 2. 2. 82. 06. 88. 33. 42 9 40 4 1 0 2 0.0

 $5 \text{ rows} \times 26 \text{ columns}$ 

In [50]:

dataset\_test=pd.read\_csv('/content/Dataset-20221111T140216Z-001.zip',sep='
',header=None).drop([26,27],axis=1)
dataset\_test.columns=col\_names

# dataset test.head()

print('Shape of Test dataset:',dataset\_train.shape)

dataset\_train.head()

Shape of Test dataset: (20631, 26)

Out[50]:

																				Ou	t[30].
	i d	c y cl e	set tin g1	set tin g2	set tin g3	s1	s2	s3	s4	s 5	•	s1 2	s1 3	s1 4	s1 5	s 1 6	s 1 7	s 1 8	s 1 9	s 2 0	s2 1
0	1	1	0.0 00 7	0.0 00 4	10 0.0	51 8. 67	64 1. 82	15 89. 70	14 00. 60	1 4. 6 2		52 1. 66	23 88. 02	81 38. 62	8. 41 95	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 6	23. 41 90
1	1	2	0.0 01 9	0.0 00 3	10 0.0	51 8. 67	64 2. 15	15 91. 82	14 03. 14	1 4. 6 2		52 2. 28	23 88. 07	81 31. 49	8. 43 18	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 0	23. 42 36
2	1	3	0.0 04 3	0.0 00 3	10 0.0	51 8. 67	64 2. 35	15 87. 99	14 04. 20	1 4. 6 2		52 2. 42	23 88. 03	81 33. 23	8. 41 78	0 0 3	3 9 0	2 3 8 8	1 0 0. 0	3 8. 9 5	23. 34 42
3	1	4	0.0 00 7	0.0 00 0	10 0.0	51 8. 67	64 2. 35	15 82. 79	14 01. 87	1 4. 6 2		52 2. 86	23 88. 08	81 33. 83	8. 36 82	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 8. 8 8	23. 37 39
4	1	5	0.0 01 9	0.0 00 2	10 0.0	51 8. 67	64 2. 37	15 82. 85	14 06. 22	1 4. 6 2		52 2. 19	23 88. 04	81 33. 80	8. 42 94	0 0 3	3 9 3	2 3 8 8	1 0 0. 0	3 8. 9 0	23. 40 44

 $5 \text{ rows} \times 26 \text{ columns}$ 

```
', header=None).drop([1],axis=1)
pm_truth.columns=['more']
pm truth['id']=pm truth.index+1
pm truth. head ()
                                                                        Out[51]:
   more id
 0
    112
         1
     98
         2
 1
 2
     69
         3
     82 4
     91 5
                                                                         In [52]:
#pre-process the dataset
rul=pd.DataFrame (dataset_test.groupby ('id')
['cycle'].max()).reset index()
rul.columns=['id','max']
rul. head()
                                                                        Out[52]:
   id max
       192
       287
 1
   2
       179
 2
   3
 3 4
        189
 4 5
       269
                                                                         In [53]:
pm truth['rtf']=pm truth['more']+rul['max']
pm_truth.head()
                                                                        Out[53]:
```

pm truth=pd.read csv('/content/Aircraft-20221111T140443Z-001.zip',sep='

```
304
      112
       98
            2
                385
 1
       69
 2
            3
                248
                271
 3
       82
       91
            5
                360
                                                                                              In [54]:
#calculate time to failure
pm truth.drop('more', axis=1, inplace=True)
dataset test=dataset test.merge(pm truth,on=['id'],how='left')
dataset test['ttf'] = dataset test['rtf'] - dataset test['cycle']
dataset test.drop('rtf', axis=1, inplace=True)
dataset test.head()
                                                                                            Out[54]:
            set
                 set
                       set
                                                                                              s2
    i
                                                              s1
                                                                   s1
                                                                                    s1
                                                                                        s2
        y
                                                         s1
                                                                        1
                                                                            1
                                                                                1
            tin
                 tin
                       tin
                            s1
                                 s2
                                       s3
                                            s4
                                                                                                  t
    d
       cl
                                                                    5
                                                                                              1
            g1
                  g2
                       g3
        e
                                                                                         3
                                                                                     1
                            51
                                 64
                                      15
                                           14
                                                         23
                                                              81
                                                                   8.
                                                                                             23.
                                                                                                  3
            0.0
                 0.0
                       10
                                                                                3
                                                                                    0
                                                                                        9.
                                                4.
                                                                            9
 0 1
                                      89.
                                           00.
                                                        88.
                                                                   41
                                                                                             41
                                                                                                  0
                            8.
                                 1.
                                                             38.
                                                                        0
                                                                                         0
            00
                 00
                      0.0
                                                 6
                                                                                8
                                                                                    0.
                                      70
                                                                   95
                                                                                             90
                                                                                                  3
                            67
                                 82
                                           60
                                                         02
                                                              62
                                                                                         3
                                                                        0
                                      15
                                                              81
                                                                   8.
            0.0
                            51
                                 64
                                           14
                                                         23
                                                                                             23.
                                                                                                  3
                 0.0
                       10
                                                                                3
                                                                                    0
                                                                                        9.
                                                4.
 1 1
            01
                            8.
                                 2.
                                      91.
                                           03.
                                                        88.
                                                             31.
                                                                   43
                                                                                             42
                  00
                       0.0
                                                 6
                                                                        0
                                                                                         0
                                 15
                            67
                                      82
                                           14
                                                         07
                                                              49
                                                                   18
                                                                                             36
                                                                                                  2
                                                         23
                                                              81
                                                                   8.
                 0.0
                            51
                                 64
                                      15
                                           14
                                                                                             23.
                                                                                                  3
                       10
                                                                                3
            0.0
                                                                                    0
                                                4.
                                                                                         8.
                                           04.
                                                                            9
                                                                                                  0
 2 1
        3
                  00
                            8.
                                 2.
                                      87.
                                                        88.
                                                             33.
                                                                   41
                                                                                             34
            04
                       0.0
                                                 6
                                                                        0
                                                                                8
                                                                                    0.
                                                                                         9
                   3
                            67
                                 35
                                      99
                                           20
                                                         03
                                                              23
                                                                   78
                                                                                             42
                                                                        3
             3
                                                 1
                                                                                         3
                                                                        0
                                                                                     1
                                                              81
                                                                   8.
            0.0
                 0.0
                            51
                                 64
                                      15
                                           14
                                                         23
                                                                                             23.
                                                                                                  3
                                                                                3
                       10
                                                                                    0
                                                 4.
                                                                                         8.
                                                                            9
 3 1
            00
                 00
                                 2.
                                      82.
                                           01.
                                                        88.
                                                             33.
                                                                   36
                                                                                             37
                                                                                                  0
                            8.
                                                                        0
                                                                                8
                       0.0
                                                 6
                                                                                    0.
                                                                                         8
                   0
                            67
                                 35
                                      79
                                           87
                                                        08
                                                              83
                                                                   82
                                                                            2
                                                                                             39
                                                                                                  0
```

82.

06.

8.

0.0

0.0

0.0

2.

88.

33.

8.

23.

id

more

rtf

```
set
          set
               set
y
                                                                    1
7
                                                                 1
                                                                         1
    tin
          tin
               tin
                    s1 s2
                               s3
                                                                                            t
cl
     g1
          g2
                g3
          00
     01
                                          2
                                                                                  0
           2
```

 $5 \text{ rows} \times 27 \text{ columns}$ 

In [55]:

dataset\_train['ttf'] = dataset\_train.groupby
(['id'])['cycle'].transform(max) - dataset\_train['cycle']
dataset\_train.head()

Out[55]:

	i d	c y cl e	set tin g1	set tin g2	set tin g3	s1	s2	s3	s4	s5	•	s1 3	s1 4	s1 5	s 1 6	s 1 7	s 1 8	s1 9	s2 0	s2 1	t t f
0	1	1	0.0 00 7	0.0 00 4	10 0.0	51 8. 67	64 1. 82	15 89. 70	14 00. 60	1 4. 6 2		23 88. 02	81 38. 62	8. 41 95	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 6	23. 41 90	1 9 1
1	1	2	0.0 01 9	0.0 00 3	10 0.0	51 8. 67	64 2. 15	15 91. 82	14 03. 14	1 4. 6 2		23 88. 07	81 31. 49	8. 43 18	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 0	23. 42 36	1 9 0
2	1	3	0.0 04 3	0.0 00 3	10 0.0	51 8. 67	64 2. 35	15 87. 99	14 04. 20	1 4. 6 2		23 88. 03	81 33. 23	8. 41 78	0 0 3	3 9 0	2 3 8 8	1 0 0. 0	3 8. 9 5	23. 34 42	1 8 9
3	1	4	0.0 00 7	0.0 00 0	10 0.0	51 8. 67	64 2. 35	15 82. 79	14 01. 87	1 4. 6 2		23 88. 08	81 33. 83	8. 36 82	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 8. 8 8	23. 37 39	1 8 8
4	1	5	0.0 01 9	0.0 00 2	10 0.0	51 8. 67	64 2. 37	15 82. 85	14 06. 22	1 4. 6 2		23 88. 04	81 33. 80	8. 42 94	0 0 3	3 9 3	2 3 8 8	1 0 0. 0	3 8. 9	23. 40 44	1 8 7

 $5 \text{ rows} \times 27 \text{ columns}$ 

In [56]:

```
df_train=dataset_train.copy()
df_test=dataset_test.copy ()
period=30
df_train['label_bc']=df_train['ttf'].apply(lambda x: 1 if x <= period else
0)</pre>
```

df\_test['label\_bc'] = df\_test['ttf'].apply(lambda x: 1 if x <= period else
0)
df\_train.head()</pre>

-	_																		Ou	t[56]:
	i d	c y cl e	set tin g1	set tin g2	set tin g3	s1	s2	s3	s4	s5	 s1 4	s1 5	s 1 6	s 1 7	s 1 8	s1 9	s2 0	s2 1	t t f	lab el_ bc
0	1	1	0.0 00 7	0.0 00 4	10 0.0	51 8. 67	64 1. 82	15 89. 70	14 00. 60	1 4. 6 2	 81 38. 62	8. 41 95	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 6	23. 41 90	1 9 1	0
1	1	2	0.0 01 9	0.0 00 3	10 0.0	51 8. 67	64 2. 15	15 91. 82	14 03. 14	1 4. 6 2	 81 31. 49	8. 43 18	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 9. 0 0	23. 42 36	1 9 0	0
2	1	3	0.0 04 3	0.0 00 3	10 0.0	51 8. 67	64 2. 35	15 87. 99	14 04. 20	1 4. 6 2	 81 33. 23	8. 41 78	0 0 3	3 9 0	2 3 8 8	1 0 0. 0	3 8. 9 5	23. 34 42	1 8 9	0
3	1	4	0.0 00 7	0.0 00 0	10 0.0	51 8. 67	64 2. 35	15 82. 79	14 01. 87	1 4. 6 2	81 33. 83	8. 36 82	0 0 3	3 9 2	2 3 8 8	1 0 0. 0	3 8. 8 8	23. 37 39	1 8 8	0
4	1	5	0.0 01 9	0.0 00 2	10 0.0	51 8. 67	64 2. 37	15 82. 85	14 06. 22	1 4. 6 2	 81 33. 80	8. 42 94	0 0 3	3 9 3	2 3 8 8	1 0 0. 0	3 8. 9 0	23. 40 44	1 8 7	0

 $5 \; rows \times 28 \; columns$ 

In [57]:

#split the data
x\_train=df\_train.iloc[:,:-1].values
y\_train=df\_train.iloc[:,-1:].values