Model building

In [48]:

```
#import required labraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix, accuracy_score
```

In [49]:

```
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]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739
4	1	5	-0.0	-0.0	100.0	518.67	642.37	1582.85	1406.22	14.	...	522.19	2388.04	8133.80	8.4294	0.	393	23	10	38.	23.4044

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21
			0 1 9	0 0 2						6 2						0 3		8 8	0. 0	9 0	

5 rows × 26 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]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	521.66	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	522.28	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	522.42	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	522.86	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	522.19	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044

5 rows × 26 columns

In [51]:

```
pm_truth=pd.read_csv('/content/Aircraft-20221111T140443Z-001.zip',sep='
',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
1	98	2
2	69	3
3	82	4
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
0	1	192
1	2	287
2	3	179
3	4	189
4	5	269

In [53]:

```
pm_truth['rtf']=pm_truth['more']+rul['max']
pm_truth.head()
```

Out[53]:

	more	id	rtf
0	112	1	304
1	98	2	385
2	69	3	248
3	82	4	271
4	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]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s13	s14	s15	s16	s17	s18	s19	s20	s21	ttf
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190	303
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236	302
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442	301
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739	300
4	1	5	-0.0	-0.0	100.0	518.67	642.37	1582.85	1406.22	14.	...	2388.04	8133.80	8.4294	0.	393	23	10	38.	23.4044	299

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s13	s14	s15	s16	s17	s18	s19	s20	s21	ttf
			0019	0002						62					03		88	0.00	90		

5 rows × 27 columns

In [55]:

```
dataset_train['ttf']=dataset_train.groupby
(['id'])['cycle'].transform(max)-dataset_train['cycle']
dataset_train.head()
```

Out[55]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s13	s14	s15	s16	s17	s18	s19	s20	s21	ttf
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	2388.02	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190	191
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	2388.07	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236	190
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	2388.03	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442	189
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	2388.08	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739	188
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	2388.04	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044	187

5 rows × 27 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)
```

```
df_test['label_bc']= df_test['ttf'].apply(lambda x: 1 if x <= period else
0)
df_train.head()
```

Out[56]:

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5	...	s14	s15	s16	s17	s18	s19	s20	s21	ttf	label_bc
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62	...	8138.62	8.4195	0.03	392	2388	100.0	39.06	23.4190	191	0
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62	...	8131.49	8.4318	0.03	392	2388	100.0	39.00	23.4236	190	0
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62	...	8133.23	8.4178	0.03	390	2388	100.0	38.95	23.3442	189	0
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62	...	8133.83	8.3682	0.03	392	2388	100.0	38.88	23.3739	188	0
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62	...	8133.80	8.4294	0.03	393	2388	100.0	38.90	23.4044	187	0

5 rows × 28 columns

In [58]:

```
x=df_train.iloc[:,:-1].values
y=df_train.iloc[:,-1].values
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_sta
te=3)
y
```

Out[58]:

```
array([0, 0, 0, ..., 1, 1, 1])
```

In [59]:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:81
8: 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-regr
ession
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

Out[59]:

```
LogisticRegression()
```

In [65]:

```
#Check the metrics of the model
from sklearn.metrics import accuracy_score
y_predlog=model.predict(x_train)
accuracy_score(y_predlog,y_train)
```

Out[65]:

```
1.0
```

In [61]:

```
y_pred_test=model.predict(x_test)
accuracy_score(y_pred_test,y_test)
```

Out[61]:

```
0.9998384491114701
```

In [62]:

```
from sklearn.metrics import confusion_matrix
cm1=confusion_matrix(y_test,y_pred_test)
cm1
```

Out[62]:

```
array([[5297,    1],
       [   0,  892]])
```

In [64]:

```
#saving the model
import joblib
joblib.dump(model, "engine_model.sav")
```

Out[64]:

```
['engine_model.sav']
```