

Model building

```
#import required labrararies
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','s9','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]:

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	6	s	7	s	8	s	9	s	2	s2	1
0	1	1	-	-	-	0.0	0.0	10	51	64	15	14	1	4.	52	23	81	8.	0	3	2	1	3	23.																	
			00	00	0.0	8.	1.	89.	00.	67	82	70	60	2	66	02	62	95	0	3	8	0.	0	41																	
			7	4															3	2	8	0	6	90																	
1	1	2	0.0	-	10	51	64	15	14	1	4.	52	23	81	8.	0	3	2	1	3	23.																				
			01	0.0	10	8.	2.	91.	03.	67	15	82	14	2	28	07	49	18	0	3	8	0.	0	42																	
			9	00	0.0	67													3	2	8	0	36																		
			3																																						
2	1	3	-	0.0	10	51	64	15	14	1	4.	52	23	81	8.	0	3	2	1	3	23.																				
			0.0	0.0	10	8.	2.	87.	04.	67	35	99	20	2	42	03	23	78	0	3	8	0.	9	34																	
			04	00	0.0	67													3	0	8	0	42																		
			3																																						
3	1	4	0.0	0.0	10	51	64	15	14	1	4.	52	23	81	8.	0	3	2	1	3	23.																				
			00	00	10	8.	2.	82.	01.	67	35	79	87	2	86	08	83	82	0	3	8	0.	8	37																	
			7	0	0.0	67													3	2	8	0	39																		
4	1	5	-	-	10	51	64	15	14	1	4.	52	23	81	8.	0	3	2	1	3	23.																				
			0.0	0.0	10	8.	2.	82.	06.	1	4.	19	04	80	94	0	9	2	1	3	40																				
					0.0	67	37	85	22	4.									0	8.	44																				

i	c	set	set	set	s1	s2	s3	s4	s	.	s1	s1	s1	s1	s	s	s	s	s	s2
d	y	tin	tin	tin					5	.	2	3	4	5	1	1	1	1	2	1
	e	g1	g2	g3						.					6	7	8	9	0	
		01	00						6						0	8	0.	9		
		9	2						2						3	8	0	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]:

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

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

```
#pre-process the dataset
rul=pd.DataFrame (dataset_test.groupby ('id')
['cycle'].max()).reset_index()
rul.columns=['id','max']
rul. head()
```

In [52]:

Out[52]:

	id	max
0	1	192
1	2	287
2	3	179
3	4	189
4	5	269

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

In [53]:

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]:

i	d	c	y	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 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	1 0 0	3 9. 0 6	23. 41 90	3 0 3
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	1 0 0	3 9. 0 0	23. 42 36	3 0 2
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	1 0 0	3 8. 9 5	23. 34 42	3 0 1
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	1 0 0	3 8. 8 8	23. 37 39	3 0 0
4	1	5			- 0.0	- 0.0	10 0.0	51 8. 67	64 2. 37	15 82. 85	14 06. 22	1 4.	.	.	.	23 88. 04	81 33. 80	8. 42 94	3 0 .	9 3	2 3	1 0	3 8.	23. 40 44	2 9 9

i	d	c	y	cl	e	set	tin	g1	set	tin	g2	set	tin	g3	s1	s2	s3	s4	s5	.	s1	s1	s1	s	s	s	s1	s2	s2	t
																					3	4	5	1	1	1	9	0	1	f
						01	00								6									0	8	0.	9			
						9	2								2									3	8	0	0			

5 rows × 27 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	s1	s1	s	s	s	s1	s2	s2	t
																					3	4	5	1	1	1	9	0	1	f
0	1	1				-	-								51	64	15	14	1		23	81	8.	0	3	2	1	3	23.	1
						0.0	0.0			10					8.	1.	89.	00.	4.	.	88.	38.	41	.	3	3	0	9.	41	9
						00	00			0.0					67	82	70	60	6	.	02	62	95	0	2	8	0.	0	90	1
						7	4												2	.				3		0	6			
1	1	2				0.0	-			10					51	64	15	14	1		23	81	8.	0	3	2	1	3	23.	1
						01	0.0			0.0					8.	2.	91.	03.	4.	.	88.	31.	43	.	3	3	0	9.	42	9
						9	3			0.0					67	15	82	14	6	.	07	49	18	0	2	8	0.	0	36	0
																			2	.				3		0	0			
2	1	3				-	0.0			10					51	64	15	14	1		23	81	8.	0	3	2	1	3	23.	1
						0.0	0.0			0.0					8.	2.	87.	04.	4.	.	88.	33.	41	.	3	3	0	8.	34	8
						04	3			0.0					67	35	99	20	6	.	03	23	78	0	0	8	0.	9	42	9
						3													2	.				3		0	5			
3	1	4				0.0	0.0			10					51	64	15	14	1		23	81	8.	0	3	2	1	3	23.	1
						00	0.0			0.0					8.	2.	82.	01.	4.	.	88.	33.	36	.	3	3	0	8.	37	8
						7	0			0.0					67	35	79	87	6	.	08	83	82	0	2	8	0.	8	39	8
																			2	.				3		0	8			
4	1	5				-	-			10					51	64	15	14	1		23	81	8.	0	3	2	1	3	23.	1
						0.0	0.0			0.0					8.	2.	82.	06.	4.	.	88.	33.	42	.	3	3	0	8.	40	8
						01	00			0.0					67	37	85	22	6	.	04	80	94	0	3	8	0.	9	44	7
						9	2												2	.				3		0	0			

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]:

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

```
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']
```