Team id: PNT2022TMID45954

Project Title: Machine Learning-Based Predictive Analytics for Aircraft Engine

```
from copyreg import pickle
from io import StringIO
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
import pickle
```

[1] ✓ 3.7s Python

```
file = open(r"C:\Users\Andrew David Flavian\Desktop\ibm\IBM-Project-13043-1659508266-main\IBM-Project-13043-1659508266-main\Task\project folder structure\Flask\engine_model.sav",'rb
model = pickle.load(file)
file.close()

file = open(r"C:\Users\Andrew David Flavian\Desktop\ibm\IBM-Project-13043-1659508266-main\IBM-Project-13043-1659508266-main\Task\project folder structure\Flask\engine_model.sav",'rb
rul_data = pickle.load(file)
file.close()



dt =  [1, 7, -0.0000, 0.0002, 100.0 , 518.67, 642.11, 1583.34, 1404.84, 14.62, 21.61, 553.89, 2388.05, 9051.39, 1.30, 47.31, 522.01, 2388.06, 8134.97, 8.3914, 0.03, 391, 2388, 100.0
print(len(dt))
```

[2] ✓ 0.3s Python

```
26
```

```
def predict(data):
    try:
        col_name = ['id','cycle','set1','set2','set3','s1','s2','s3','s4','s5','s6','s7','s8']+['s9','s10','s11','s12','s13','s14','s14','s15','s16','s17','s18','s19','s20']
        test_dataset = pd.DataFrame([data],columns=col_name)
        rul=pd.DataFrame(test_dataset.groupby("id")['cycle'].max()).reset_index()
        rul.columns = ['id','max']
        truth_ds['rtf']=truth_ds['more']+rul["max"]
        truth_ds.head()
        truth_ds['rtf']=truth_ds['more']+rul["max"]
        test_dataset=test_dataset.merge(truth_ds, on= ['id'], how= "left")
        test_dataset[ 'ttf']=test_dataset['rtf'] - test_dataset['cycle']
        test_dataset.drop ('rtf', axis=1, inplace=True)
        df_test = test_dataset.copy()
        period = 30
        df_test['label_bc'] = df_test ['ttf'].apply(lambda x: 1 if x <= period else 0)
        df_test = df_test.dropna()
        if len(df_test.index) == 0 :
            return True
        x_test = df_test.iloc[ : , : -2].values
        y_pred = model.predict(x_test)
        return True if y_pred[0] else False
    except:
        return True
```

[3] ✓ 0.3s Python

```
predict(dt)
```

[4] ✓ 0.4s Python

```
True
```