```
import os
import sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen
from urllib.parse import unquote, urlparse
from urllib.error import HTTPError
from zipfile import ZipFile
import tarfile
import shutil

CHUNK_SIZE = 40960
DATA_SOURCE_MAPPING = ':https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F36672%2F55876%2Fbundle%2Farchive.zip%3FX-Goog-Algorithm

KAGGLE_INPUT_PATH='/kaggle/input'
KAGGLE_WORKING_PATH='/kaggle/working'
KAGGLE_SYMLINK='kaggle'

!umount /kaggle/input/ 2> /dev/null
shutil.rmtree('/kaggle/input', ignore_errors=True)
os.makedirs(KAGGLE_INPUT_PATH, 0o777, exist_ok=True)
os.makedirs(KAGGLE_WORKING_PATH, 0o777, exist_ok=True)

try:
  os.symlink(KAGGLE_INPUT_PATH, os.path.join("..", 'input'), target_is_directory=True)
except FileExistsError:
  pass
try:
  os.symlink(KAGGLE_WORKING_PATH, os.path.join("..", 'working'), target_is_directory=True)
except FileExistsError:
  pass

for data_source_mapping in DATA_SOURCE_MAPPING.split(','):
    directory, download_url_encoded = data_source_mapping.split(':')
    download_url = unquote(download_url_encoded)
    filename = urlparse(download_url).path
    destination_path = os.path.join(KAGGLE_INPUT_PATH, directory)
    try:
        with urlopen(download_url) as fileres, NamedTemporaryFile() as tfile:
            total_length = fileres.headers['content-length']
            print(f'Downloading {directory}, {total_length} bytes compressed')
            dl = 0
            data = fileres.read(CHUNK_SIZE)
            while len(data) > 0:
                dl += len(data)
                tfile.write(data)
                done = int(50 * dl / int(total_length))
                sys.stdout.write(f"\r[{'=' * done}{' ' * (50-done)}] {dl} bytes downloaded")
                sys.stdout.flush()
                data = fileres.read(CHUNK_SIZE)
            if filename.endswith('.zip'):
              with ZipFile(tfile) as zfile:
                zfile.extractall(destination_path)
            else:
              with tarfile.open(tfile.name) as tarfile:
                tarfile.extractall(destination_path)
            print(f'\nDownloaded and uncompressed: {directory}')
    except HTTPError as e:
        print(f'Failed to load (likely expired) {download_url} to path {destination_path}')
        continue
    except OSError as e:
        print(f'Failed to load {download_url} to path {destination_path}')
        continue

print('Data source import complete.')
```

```
Downloading , 1383996 bytes compressed
[==================================================] 1383996 bytes downloaded
Downloaded and uncompressed: 
Data source import complete.
```

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import confusion_matrix,accuracy_scor

from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM, Activation
from keras.callbacks import EarlyStopping

import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
dataset_train=pd.read_csv('../input/PM_train.txt',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','
dataset_train.columns=col_names
print('Shape of Train dataset: ',dataset_train.shape)
dataset_train.head()
```

```
Shape of Train dataset:  (20631, 26)
```

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62

5 rows × 26 columns

```
dataset_test=pd.read_csv('../input/PM_test.txt',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)
```

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62

5 rows × 26 columns

```
pm_truth=pd.read_csv('../input/PM_truth.txt',sep=' ',header=None).drop([1],axis=1)
pm_truth.columns=['more']
pm_truth['id']=pm_truth.index+1
pm_truth.head()
```

	more	id
0	112	1
1	98	2
2	69	3
3	82	4
4	91	5

Next steps: Generate code with pm_truth View recommended plots

```
# generate column max for test data
rul = pd.DataFrame(dataset_test.groupby('id')['cycle'].max()).reset_index()
rul.columns = ['id', 'max']
rul.head()
```

	id	max
0	1	31
1	2	49
2	3	126
3	4	106
4	5	98

Next steps: Generate code with `rul` View recommended plots

```
# run to failure
pm_truth['rtf']=pm_truth['more']+rul['max']
pm_truth.head()
```

	more	id	rtf
0	112	1	143
1	98	2	147
2	69	3	195
3	82	4	188
4	91	5	189

Next steps: Generate code with `pm_truth` View recommended plots

```
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()
```

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5
0	1	1	0.0023	0.0003	100.0	518.67	643.02	1585.29	1398.21	14.62
1	1	2	-0.0027	-0.0003	100.0	518.67	641.71	1588.45	1395.42	14.62
2	1	3	0.0003	0.0001	100.0	518.67	642.46	1586.94	1401.34	14.62
3	1	4	0.0042	0.0000	100.0	518.67	642.44	1584.12	1406.42	14.62
4	1	5	0.0014	0.0000	100.0	518.67	642.51	1587.19	1401.92	14.62

5 rows × 27 columns

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

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62

5 rows × 27 columns

```
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()
```

	id	cycle	setting1	setting2	setting3	s1	s2	s3	s4	s5
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70	1400.60	14.62
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82	1403.14	14.62
2	1	3	-0.0043	0.0003	100.0	518.67	642.35	1587.99	1404.20	14.62
3	1	4	0.0007	0.0000	100.0	518.67	642.35	1582.79	1401.87	14.62
4	1	5	-0.0019	-0.0002	100.0	518.67	642.37	1582.85	1406.22	14.62

5 rows × 28 columns

```
features_col_name=['setting1', 'setting2', 'setting3', 's1', 's2', 's3', 's4', 's5', 's6', 's7', 's8', 's9', 's10', 's11',
                   's12', 's13', 's14', 's15', 's16', 's17', 's18', 's19', 's20', 's21']
target_col_name='label_bc'
```

```
sc=MinMaxScaler()
df_train[features_col_name]=sc.fit_transform(df_train[features_col_name])
df_test[features_col_name]=sc.transform(df_test[features_col_name])


def gen_sequence(id_df, seq_length, seq_cols):
    df_zeros=pd.DataFrame(np.zeros((seq_length-1,id_df.shape[1])),columns=id_df.columns)
    id_df=df_zeros.append(id_df,ignore_index=True)
    data_array = id_df[seq_cols].values
    num_elements = data_array.shape[0]
    lstm_array=[]
    for start, stop in zip(range(0, num_elements-seq_length), range(seq_length, num_elements)):
        lstm_array.append(data_array[start:stop, :])
    return np.array(lstm_array)

# function to generate labels
def gen_label(id_df, seq_length, seq_cols,label):
    df_zeros=pd.DataFrame(np.zeros((seq_length-1,id_df.shape[1])),columns=id_df.columns)
    id_df=df_zeros.append(id_df,ignore_index=True)
    data_array = id_df[seq_cols].values
    num_elements = data_array.shape[0]
    y_label=[]
    for start, stop in zip(range(0, num_elements-seq_length), range(seq_length, num_elements)):
        y_label.append(id_df[label][stop])
    return np.array(y_label)


# timestamp or window size
seq_length=50
seq_cols=features_col_name


# generate X_train
X_train=np.concatenate(list(list(gen_sequence(df_train[df_train['id']==id], seq_length, seq_cols)) for id in df_train['id'].unique()))
print(X_train.shape)
# generate y_train
y_train=np.concatenate(list(list(gen_label(df_train[df_train['id']==id], 50, seq_cols,'label_bc')) for id in df_train['id'].unique()))
print(y_train.shape)
```

```
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
```

```
# generate X_test
X_test=np.concatenate(list(list(gen_sequence(df_test[df_test['id']==id], seq_length, seq_cols)) for id in df_test['id'].unique()))
print(X_test.shape)
# generate y_test
y_test=np.concatenate(list(list(gen_label(df_test[df_test['id']==id], 50, seq_cols,'label_bc')) for id in df_test['id'].unique()))
print(y_test.shape)
```

```
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
  <ipython-input-13-ce390dc8d101>:14: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a fut
    id_df=df_zeros.append(id_df,ignore_index=True)
```

```
nb_features =X_train.shape[2]
timestamp=seq_length

model = Sequential()

model.add(LSTM(
         input_shape=(timestamp, nb_features),
         units=100,
         return_sequences=True))
model.add(Dropout(0.2))

model.add(LSTM(
          units=50,
          return_sequences=False))
model.add(Dropout(0.2))

model.add(Dense(units=1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50, 100)           50000

 dropout (Dropout)           (None, 50, 100)           0

 lstm_1 (LSTM)               (None, 50)                30200

 dropout_1 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 80251 (313.48 KB)
Trainable params: 80251 (313.48 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
# fit the network
model.fit(X_train, y_train, epochs=10, batch_size=200, validation_split=0.05, verbose=1,
          callbacks = [EarlyStopping(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto')])
```

```
Epoch 1/10
98/98 [==============================] - 31s 274ms/step - loss: 0.2123 - accuracy: 0.9131 - val_loss: 0.0990 - val_accuracy: 0.9542
Epoch 2/10
98/98 [==============================] - 25s 259ms/step - loss: 0.0811 - accuracy: 0.9679 - val_loss: 0.0891 - val_accuracy: 0.9572
Epoch 3/10
98/98 [==============================] - 32s 330ms/step - loss: 0.0717 - accuracy: 0.9715 - val_loss: 0.0994 - val_accuracy: 0.9533
<keras.src.callbacks.History at 0x7c51ec393820>
```

```
# training metrics
scores = model.evaluate(X_train, y_train, verbose=1, batch_size=200)
print('Accurracy: {}'.format(scores[1]))
```

```
103/103 [==============================] - 11s 105ms/step - loss: 0.0815 - accuracy: 0.9642
Accurracy: 0.9642491936683655
```

```
def prob_failure(machine_id):
    machine_df=df_test[df_test.id==machine_id]
    machine_test=gen_sequence(machine_df,seq_length,seq_cols)
    m_pred=model.predict(machine_test)
    failure_prob=list(m_pred[-1]*100)[0]
    return failure_prob
```

```
machine_id=16
print('Probability that machine will fail within 30 days: ',prob_failure(machine_id))
```

```
<ipython-input-13-ce390dc8d101>:3: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future
  id_df=df_zeros.append(id_df,ignore_index=True)
4/4 [==============================] - 1s 21ms/step
Probability that machine will fail within 30 days:  0.051617444
```