# *TRAIN THE ML MODEL*

```
#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 [48]:

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

In [ ]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='MNiUYLIWesIem9Eh5uzuzgRGvqx5IYSqSnhJM_pUK1hH',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-
storage.appdomain.cloud')

bucket = 'modelbuilding-donotdelete-pr-gj3r2r4kwyxkto'
object_key = 'Model Building.ipynb'

streaming_body_2 = cos_client.get_object(Bucket=bucket,
Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about
the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It
includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='MNiUYLIWesIem9Eh5uzuzgRGvqx5IYSqSnhJM_pUK1hH',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'modelbuilding-donotdelete-pr-gj3r2r4kwyxkto'
object_key = 'Model Building.ipynb'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

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]:

| | 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 |

|  | id | cycle | setting1 | setting2 | setting3 | s1 | s2 | s3 | s4 | s5 | ... | s12 | s13 | s14 | s15 | s16 | s17 | s18 | s19 | s20 | s21 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **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 [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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  | 019 | 002 |  |  |  |  |  | 62 |  |  |  |  | 03 |  | 88 | 0.0 | 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']
```

# deploy model

In [1]:

```
!pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/en
vs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python
-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/
python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Pytho
n-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.
0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/pyt
hon3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/py
thon3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
```

```
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/p
ython3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/py
thon3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/p
ython3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Pyt
hon-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3
.4)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/
Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watso
n-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Py
thon-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-
machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda
/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm
-watson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/en
vs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->i
bm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/l
ib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-
learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/
lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine
-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/p
ython3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core=
=2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs
/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-
learning) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/l
ib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.
3)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/
python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learni
ng) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/
Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-
learning) (3.0.4)
```

In [2]:

```python
from ibm_watson_machine_learning import APIClient
```

In [3]:

```python
wml_credentials = {
    "url": "https://us-south.ml.cloud.ibm.com",
    "apikey" :"hNzkDg_2UyFOhp8N3SorKaqB14tQpUAT1Ci6RpWIyXTC"
}
```

In [14]:

```python
wml_client = APIClient(wml_credentials)
wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if t
he number of records exceed 50
```

```
------------------------------------  --------------  ---------------------
---
ID                                    NAME            CREATED
c25e08e2-22e3-4932-bf81-77c30fadbe65  model           2022-11-17T16:11:08.2
79Z
29fcbf45-6183-447c-8e34-758b3c2c8945  models          2022-11-17T16:10:23.8
98Z
cfd83c6d-f976-45d7-abd8-194e088f12a6  model building  2022-11-17T16:06:54.8
40Z
79c79f6b-2a3e-4fea-9dbc-bf7631a57192  model building  2022-11-17T15:54:15.9
60Z
------------------------------------  --------------  ---------------------
---
```

In [15]:

```
space_id = "c25e08e2-22e3-4932-bf81-77c30fadbe65"
```

In [16]:

```
wml_client.set.default_space(space_id)
```

Out[16]:

```
'SUCCESS'
```

In [8]:

```
wml_client.software_specifications.list()
```

```
-----------------------------  ------------------------------------  ----
NAME                           ASSET_ID                              TYPE
default_py3.6                  0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12      020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt     069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6        09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12     09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9      0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6          0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                     0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod   1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6              10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl      111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10        125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9             12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6        154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                   1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6         1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6           1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt  1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6           1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod   217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda        26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                       295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8            2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6          2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9          2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6              2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt     32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37           36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10        396b2e83-0953-5b86-9a55-7ce1628a406f  base
```

```
xgboost_0.82-py3.6             39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt     40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10     40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9         4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                 42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                   493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6          49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6             4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6         50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8            52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11     55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                 5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1              5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                     5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc         5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7            632d4b22-10aa-5180-88f0-f52dfb6444d7  base
-----------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' par
ameter.
```

In [9]:

```
MODEL_NAME = 'machine_learning'
DEPLOYMENT_NAME = 'machine_deploy'
MACHINE_MODEL = 'MODEL'
```

In [10]:

```
software_spec_uid =
wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [13]:

```
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:
software_spec_uid
}
```

In [ ]:

```
model_details = wml_client.repository.store_model(
    model=MACHINE_MODEL,
    meta_props=model_props,
    training_data=x_train,
    training_target=y_train
)
```